APPLICATION OF LEARNING ALGORITHMS TO NONLINEAR FILTERING AND MARKOV CHAIN MONTE CARLO METHODS

By

ANAND RADHAKRISHNAN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2019

To my parents and my wife

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my advisor, Professor Sean Meyn, who has been a source of continued wisdom, guidance and support throughout my PhD over the last six years. Right from the time, I was a student in his Stochastic methods class, I was lucky to receive his able mentorship that has made my sailing smooth. With limited prior research experience, it would not have been possible to step into complex arena of stochastic control and Markov chains research, if not for his amazing patience and cheerful personality. I would like to thank Professor Jose Principe, Professor Kamran Mohseni and Professor James Hobert, for agreeing to be part of my supervisory committee and for their critical evaluation of my work. I would also like to thank Professor Eric Moulines for the warm hospitality he bestowed on me during my month long stay at École Polytechnique, Paris in the summer of 2016. This stay opened up avenues for collaborative research and I also got the opportunity to attend the first Data Science Summer School (DS3) in 2017. I would like to thank all the professors at IIT Bombay and College of Engineering, Trivandrum for helping me build a solid foundation, which acted as a springboard for conducting my PhD research.

I would like to thank my lab mates and colleagues, Adithya Devraj, Joel Mathias, Shuhang Chen, Neil Cammardella, Yue Chen and Surya Dhulipala for helping me with my research and creating a friendly environment to work in.

When I first landed in Gainesville in Fall 2012, I never imagined it would be my home for the next seven years (still counting!). After the initial struggle of settling in a new environment, I was made to feel at home by my amazing friends and roommates. I am thankful to Kishore Rajasekar, Shivashankar Halan, Pratyush Chakraborty, Joji Jacob, Sarath Francis, Gayathri Srinivasan, Ravi Venkatraman, Ramya Sivakumar, Yogesh Deshmukh, Thames Harrison for making my life relatively stress-free. I would like to thank Dr. Ravi Ahuja and my team members at Optym, Gainesville for the opportunity to intern there.

I would like to thank my parents, Lalitha and Radhakrishnan for the immense support, both financial and emotional, that they provided me throughout my life. The freedom of choice

that they granted, allowed me to pursue my dream. I am also indebted to my grandparents for their unconditional love and affection. My cousin Dr. Sriram Ganapathy deserves special mention as he was the first person to inspire me to pursue a PhD. I would also like to thank my friend Dr.Krishnakumar Gopalakrishnan for his constant support in matters of research and life in general.

I would like to express my love and gratitude to my wife Dharani Balasubramanian, who has stood by my side for the last four years. This PhD would not have been completed without the tons of emotional support that she gave me to keep me going. Thank you for sticking with me through thick and thin.

TABLE OF CONTENTS

LIST OF TABLES

Table page

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

APPLICATION OF LEARNING ALGORITHMS TO NONLINEAR FILTERING AND MARKOV CHAIN MONTE CARLO METHODS

By

Anand Radhakrishnan

December 2019

Chair: Sean P. Meyn
Major: Electrical and Computer Engineering

This dissertation broadly focuses on the development of Monte Carlo based methods for applications in nonlinear filtering and variance reduction in simulation algorithms. Poisson's equation is a central theme in stochastic optimal control and Markov chain theory. In this dissertation, we study a particular version of Poisson's equation associated to the Langevin diffusion.

Feedback particle filter (FPF) is a Monte Carlo based approximation to the nonlinear filter based on mean field optimal control techniques. It is known that the gradient to the solution of a particular version of Poisson's equation plays a crucial role in the FPF. It was recently discovered that the same object can be used to reduce variance in common discrete-time Markov Chain Monte Carlo (MCMC) algorithms, even though the solution is defined with respect to a Markov model in continuous time. In the feedback particle filter, the gradient to the solution is interpreted as the innovations gain. In MCMC algorithms, it appears in the objective function to minimize the asymptotic variance.

The main contributions of this dissertation are as follows: a new formulation of the TD-learning algorithm, called the differential TD-learning is proposed to approximate the gradient of the solution to Poisson's equation directly. This requires the difficult choice of an appropriate parameterized family of functions within which the approximation lies. In addition to considering a finite dimensional family of functions, the basis selection problem is addressed by using a reproducing kernel Hilbert space (RKHS). In the RKHS setting, the

objective function is represented in the form of an empirical risk minimization (ERM) problem that allows us to apply a recent variation of RKHS theory to find the optimal approximation. Both applications are discussed in detail and illustrated with numerical experiments.

CHAPTER 1
INTRODUCTION

In a broad sense, this dissertation explores the development of reinforcement learning based techniques suited to applications in nonlinear filtering (or nonlinear state estimation) and Markov chain Monte Carlo (MCMC) algorithms. During the initial development of this dissertation, a special class of approximate nonlinear filters called the feedback particle filter (FPF) was the primary focus. Later, by lucky coincidence, it was observed that similar techniques could be applied to obtain interesting results in MCMC algorithms as well. Hence, the chapter on MCMC forms a smaller portion of the dissertation.

Question:lucky coincidence maybe too casual?

The goal of this chapter is a cursory introduction of the elements that are key to the problems considered in this dissertation. In Section 1.1, we describe the two major goals undertaken - in the areas of nonlinear filtering and MCMC. Subsequently, in Section 1.2, we introduce the various tools used, motivate the solution approaches adopted, without delving much into the technical details.

1.1 Goals of the Dissertation

1.1.1 Nonlinear Filtering/State Estimation

Consider a dynamic system evolving in time according to a given mathematical model. A complete characterization of the system is given by its states. Uncertainties in the system model or external disturbances that affect the state are modeled as process noise, and indirect observations of the state, corrupted by measurement noise are available. The observation model and the noise statistics are assumed to be known. The state dynamics and observations may be in either continuous or discrete time depending on the system properties. Additionally, the state dynamics are assumed to be Markovian, i.e. in rough terms, the probability of the current state just depends on the previous state, and not on the entire history. These assumptions will be made more precise in Chapter 4.

A generic block diagram of a state estimator is depicted in Fig. 1-1. The goal of any filtering/state estimation problem is to recursively estimate the states of the system based on

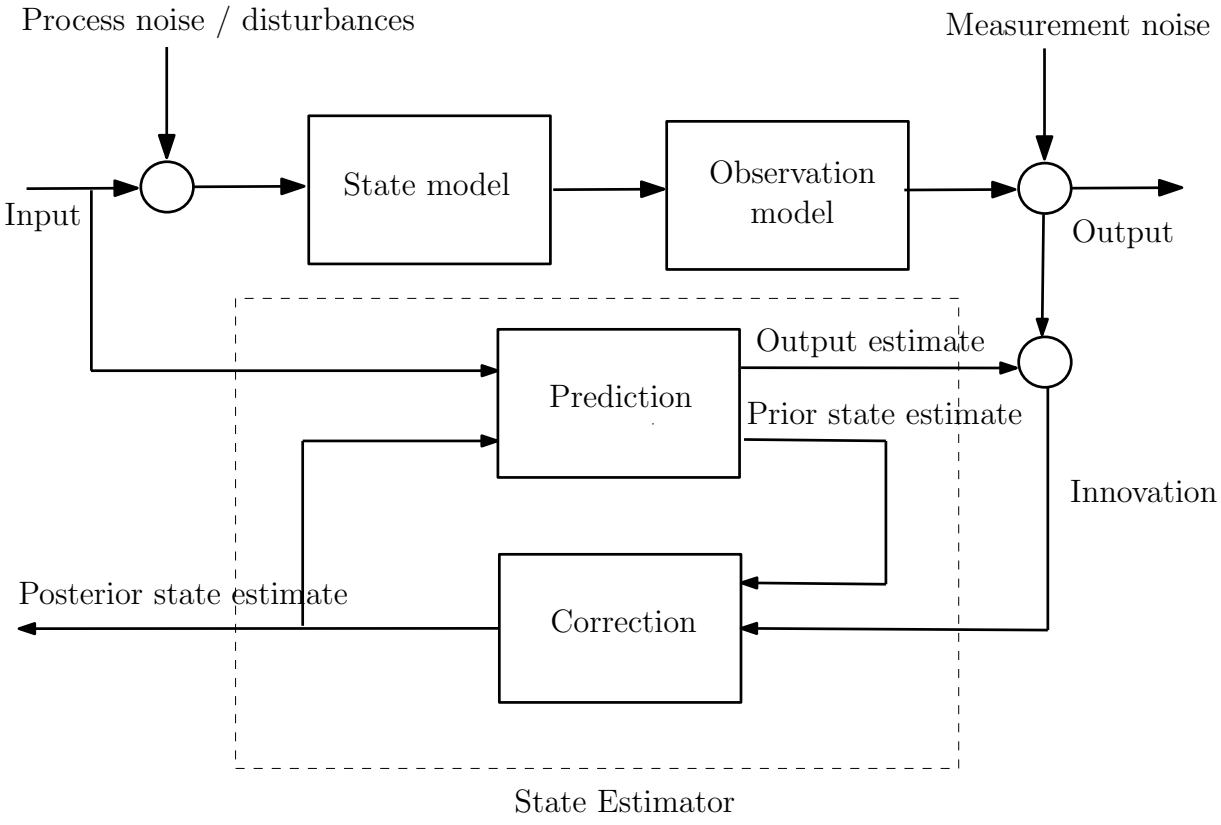


Figure 1-1. Block diagram of a state estimator

noisy partial observations. The top row of the block diagram, with the state and observation models, indicates the actual evolution of the system. The prediction and correction blocks form part of the state estimator. Given the current estimate of the state, the prediction block gives the best state estimate, called the prior estimate for the next time instant using the state model. The correction block updates the prior after receiving the most recent observation, and outputs the posterior estimate. All filtering approaches share this basic structure, although in some cases, the prediction and correction steps may be combined.

Question:Need to verify if the block diagram is correct

Initial applications of filtering included satellite orbit determination, aircraft navigation and tracking [3]. More recently, filtering has found applications in diverse areas such as machine learning [4], queueing networks, mathematical finance [5] and data assimilation problems for weather forecasting [6].

Question:A simple example may be good here.

When the system dynamics are linear and the noise quantities are Gaussian, the problem is simpler and the well-known Kalman filter is the optimal solution. The Kalman filter gives a linear SDE (stochastic differential equation) for the conditional mean of the state and a Riccati type ODE for the state covariance. These two quantities completely characterize the posterior estimate. However, in practice, the linear state dynamics with Gaussian disturbances assumption is often violated. For example, in weather forecasting the states evolve via a complex system of fluid mechanics equations that are nonlinear. The optimal solution is given by a set of SDEs, like the Zakai's equation or the Kushner-Stratonovich equation. The posterior estimates are in the form of conditional distributions of the state, given the entire history of observations. A detailed discussion of the nonlinear filtering theory can be found in [7].

Linear approximations like the extended Kalman filter (EKF) were studied for application to nonlinear systems. They perform well as long as the state or observation dynamics do not deviate significantly from linearity. Later with the advent of modern computing, Monte Carlo based methods like the conventional bootstrap particle filter gained popularity. The underlying principle here is to approximate the posterior distribution using empirical samples called particles. Budhiraja et al. [8] provide a comprehensive survey of numerical methods for nonlinear filtering problems.

The main focus of this dissertation is a class of controlled particle system algorithms called the feedback particle filter (FPF). Feedback particle filter was originally formulated for the continuous-time nonlinear filtering problem in the Euclidean setting [1]. They have since been extended to Riemannian manifolds and matrix Lie groups [9]. The FPF is similar in its feedback structure to the Kalman filter and in its empirical approximation approach to the standard particle filter. In other respects, they are significantly different. A crucial component of the FPF is the gain function, which is analogous to the Kalman gain in the Kalman filter. The optimal gain function in the FPF is obtained as the gradient of the solution to a particular version of Poisson's equation [1, 10]. Obtaining an analytical solution to the Poisson's equation

is often difficult and hence, approximation is required. In this dissertation, our main focus is on developing algorithms to approximate the gain function. A detailed discussion of the FPF theory and gain approximation algorithms is reserved for Chapter 4.

1.1.2 Markov Chain Monte Carlo (MCMC) Algorithms

The second application of interest is Markov chain Monte Carlo (MCMC) algorithms. MCMC algorithms have a long history of being applied to problems in Bayesian statistics []. Question:citation needed

In standard Monte Carlo methods, expectation of a function f of a random variable X distributed according to a density ρ is approximated empirically as,

$$\mathsf{E}_{X\sim\rho}[f(X)] := \int_{\mathsf{X}} f(x)\rho(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(X_i) \tag{1-1}$$

where each X_i is distributed according to ρ and N is sufficiently large. As is often the case, it may be difficult to generate a sequence of samples $\{X_i\}_1^N$ according to the desired target distribution ρ. Methodologies such as rejection sampling and importance sampling make use of an easy-to-sample surrogate distribution to sample from the original target distribution. But, if ρ is high-dimensional, it is difficult to find a closely matching simple surrogate distribution.

MCMC algorithms provide an alternative solution in this situation. They are a special class of Monte Carlo methods in which the samples X_i are the states of an ergodic Markov chain. Given the target density ρ, the problem reduces to designing an appropriate transition kernel for a Markov chain that has ρ as its invariant density. The Langevin diffusion is a continuous-time Markov process, which can be thought of as a perturbed gradient flow with respect to a potential function. It forms the basis of many MCMC algorithms. The Gibbs algorithm [11] and Metropolis-Hastings (M-H) algorithm [12] are other popular discrete-time MCMC techniques. They have been widely applied for problems in Bayesian inference, statistical physics, computation biology etc.

The asymptotic convergence of the empirical averages of the form (1-1) to the true expected value is guaranteed under general conditions by law of large numbers. The main drawback of these techniques as compared to standard Monte Carlo sampling which provides

independent and identically distributed (i.i.d.) samples, is that the successive samples of the Markov chain are correlated to each other. This results in slower convergence of the algorithms to the target density. The central limit theorem states that,

$$\sqrt{N}\Big(\frac{1}{N}\sum_{i=1}^{N} f(X_i) - \mathsf{E}_{X\sim\rho}[f(X)]\Big) \xrightarrow{d} \mathcal{N}(0,\sigma_\infty^2), \qquad \text{as } N\to\infty, \tag{1-2}$$

where $\mathcal{N}(0,\sigma_\infty^2)$ refers to the Gaussian distribution with zero mean and variance σ_∞^2. Asymptotic variance σ_∞^2 is a measure quantifying the rate of convergence. Lower its value, faster is the convergence of the Markov chain to its invariant distribution and hence, the goal is to minimize it. Asymptotic variance can be expressed in terms of the solution to Poisson's equation [13].

Control variates, which are zero-mean terms added to the function f, have been used to reduce the asymptotic variance of the estimates without adding any bias. Henderson, in his dissertation [14] notes that the best choice of control variates can be constructed using the solution to Poisson's equation. They also feature prominently in Chapter 11 of the book [13] with the objective of constructing reduced-variance estimators for network models. Control variates constructed using the fluid value function have been shown to produce a 100-fold reduction in variance over the standard estimator for the KSRS queueing model in the examples considered in the book chapter.

In this dissertation, we demonstrate that the same algorithms we propose for approximating the FPF gain function, find additional application in improving the performance of popular MCMC algorithms. A detailed discussion on MCMC, including numerical examples demonstrating asymptotic variance reduction is provided in Chapter 5.

1.2 Tools Used in the Dissertation

Now, that the main application areas of the dissertation have been described briefly, we introduce the various tools that help us achieve our goal. In Section 1.2.1, a preliminary description of the Poisson's equation, that is crucial to both our applications of interest is given.

1.2.1 Poisson's Equation

In its most general form, Poisson's equation is a second-order differential equation of the form,

$$\mathcal{D}h := -f,$$

where $\mathcal{D}$ is a second-order differential operator. Usually, $f \in C^2$ is given and is "centered" by subtracting its mean. The function h is unknown and is called the solution to the Poisson's equation. In physics, the operator $\mathcal{D}$ is often taken to be the Laplacian. Poisson's equation appears widely in the context of Markov chains and stochastic optimal control. In the context of a continuous-time diffusion process, the operator $\mathcal{D}$ refers to the infinitesimal generator, also called the differential generator. .

Anand: If $f \equiv 0$, then h is precisely harmonic functions [15]

Question:The states evolve in the form of a controlled Markov chain based on a given policy.

Poisson's equation is central to average-cost optimal control theory. In this case, f is a one-step cost function and h is called a relative value function. Relative value function gives the infinite-horizon expected cost when starting from a given state under this/a stationary policy. Approximate solutions to the equation lead to direct performance bounds of the control algorithm [13]. Explicit bounds on the solution h have been obtained under general conditions of the chain in [].

Question:citation needed

Our interest lies in a particular version of Poisson's equation associated to the Langevin diffusion process. Langevin diffusion is discussed in greater detail in Sections section 2.1 and **??**. Gradient of the solution to this equation is the optimal choice of the gain function associated with the FPF [1]. In MCMC algorithms, as noted by Henderson [14] and later by Dellaportas et al. [16], the optimal control variates can be constructed from this solution. Thus, Poisson's equation and its solution are central to the goals of this dissertation.

Obtaining a closed form solution is difficult outside of special cases and this motivates the study of approximation algorithms. Finding an approximate solution falls within the framework of reinforcement learning and in particular, temporal difference (TD) learning. In this dissertation, we develop variants of the TD learning algorithm that can approximate the

gradient of the solution to Poisson's equation directly. Other approaches include the Markov semigroup approximation by Taghvaei et al. [17].

1.2.2 Reinforcement Learning and TD Learning

In this section, a beginner level introduction to reinforcement learning algorithms is provided. Reinforcement learning algorithms have gained popularity over the last decade having achieved major successes in a wide variety of applications like AlphaGo, backgammon etc. In a general setting, such algorithms involve learning what actions to take in a given situation, so as to maximize a numerical reward (or equivalently minimize a numerical cost) over a (possibly infinite) time-horizon. The learned set of actions, called a policy is a mapping from the state space to the action space. In a stochastic setting, this mapping is expressed in terms of probability of taking a particular action in a given state. The learning is performed purely based on interactions with the environment without any prior knowledge of the system model. A whole variety of algorithms including Q-learning [18] and temporal difference (TD) learning [19] belong to this category. A slightly different class that makes use of model information is called approximate dynamic programming. Although, the end objective is to obtain optimal policies, a central theme in all these algorithms is value function approximation. This aspect is what makes these algorithms an attractive choice for our objective.

Anand:need references

Sutton and Barto write in their monograph [20], "If one had to identify one idea as central and novel to reinforcement learning, it would undoubtedly be temporal difference learning". Originally introduced by Sutton in [19], TD learning algorithms address the problem of policy evaluation associated with discrete-time stochastic optimal control problems called Markov decision processes (MDPs). In other words, for a given fixed policy, the algorithm computes estimates of the value function through an iterative procedure. A large body of prior research is available that studies the asymptotic convergence properties of these algorithms. Most of them, however are restricted to either the discounted-cost case or an undiscounted-cost (average-cost) setting for a finite state space Markov chain with an absorbing state. Both

these assumptions are violated in the context of approximating the solution to Poisson's equation of interest to us. Hence, the need to develop new algorithms.

A least-squares approach to TD learning, introduced by Bradtke et al. [21], has been discussed in the context of value function estimation for a fixed policy in [13]. An appropriate learning objective is chosen based on the context. For example, in optimal control, the goal is to optimize performance over the class of policies, whereas in MCMC, minimizing the asymptotic variance is the chosen criterion. Least squares TD (LSTD) learning aims to obtain approximations to the true value function from within a parameterized family of functions. The learning problem is thus reduced to finding the best approximation based on a chosen criterion from within this class. Section 11.5.2 of [13] presents the LSTD learning algorithm for the discounted-cost problem. Recursive update rules are presented for the parameter weights and they have been shown to converge asymptotically to the optimal values. In Section 11.5.4, the results are generalized to the average-cost setting, where we need to solve the Poisson's equation. However, the application of LSTD algorithm in this case is justified only if a regenerating state for the Markov chain exists. This rules out its application to state spaces in higher dimensions.

In this dissertation, we follow the key ideas presented in Section 11.5.2 to develop a new class of algorithms called differential LSTD learning that approximates the gradient of the solution to Poisson's equation directly. The algorithm design enables the construction of a Monte Carlo based technique that scales well in high dimensions. This is achieved by introducing an implicit discounting factor, that ensures the existence of a regenerating state for the process. For the special case of Langevin diffusion, using the self-adjoint property of the differential generator, we obtain a simple and elegant version of the algorithm in Section 2.3.1. A more general version of the algorithm that can be applied to any continuous-time diffusion processes is presented in Section 2.3.3. A similar algorithm for discrete-time examples with applications to optimal control is presented in [22].

1.2.3 Reproducing Kernel Hilbert space (RKHS)

Traditionally, TD learning has been tested on spaces spanned by a carefully chosen finite set of basis functions. However, there are no standard approaches to choosing the basis. If any insight about the structure of the true value function is available, this can be exploited in choosing an appropriate function class. In optimal control problems, this insight is available in the form of fluid value functions. But this is not true for Poisson's equation of interest. One of the novelties of this dissertation is the use of reproducing kernel Hilbert space (RKHS) as an approximating function space for TD learning algorithms. This simplifies the choice of problem dependent basis functions by making use of the information about particle locations effectively.

Reproducing kernel Hilbert spaces form an important class of function spaces in learning theory [2, 23]. They are complete Hilbert spaces uniquely characterized by a symmetric and positive definite kernel function. The use of RKHS for function approximation provides us with greater flexibility and potentially a richer function class. Although optimization in an RKHS is typically infinite dimensional, they are endowed with useful properties that help us reduce the problem to finding a finite set of parameters via the representer theorem [24, 25]. A brief background of the RKHS theory along with properties that make them useful in this dissertation is provided in Chapter 3. The differential TD learning algorithm specific to the Langevin diffusion is adapted to accommodate an RKHS as the approximating function space in Chapter 3.

1.3 Dissertation Outline

The various topics discussed in this chapter may seem distinct and disconnected. This dissertation attempts to tie them together as it progresses. To summarize, we restrict our attention to approximating solutions of a particular version of the Poisson's equation. Differential TD learning algorithms using a finitely parameterized family of functions and an RKHS are studied and presented. Approximations obtained using these algorithms are tested and have been shown to improve the performance of the FPF and MCMC algorithms.

The main contributions of this dissertation that form the remaining chapters can be summarized as - i) development of differential LSTD learning algorithm, similar to LSTD in [13] to approximate the gradient of the solution to Poisson's equation directly in Chapter 2 , ii) refinement of this algorithm with a simpler resolution [26] in Section 2.3.1, iii) address the problem of basis selection by the use of RKHS in Chapter 3, iv) applications to nonlinear filtering in Chapter 4 and MCMC algorithms in Chapter 5. Finally, conclusions and scope for future work are included in Chapter 6.

Anand:very complex sentence construction

1.4 Notation

Question:X or Φ?

$\boldsymbol{X} := \{X_t : t \geq 0\}$ is a stochastic process, evolving on the state space X, which is taken to be $\mathbb{R}^d$, unless specified. A subscript notation X_t is often used to denote $X(t)$, where t is an index of time. The symbol ρ denotes a probability density on X, and $L^2(\mathsf{X}, \rho)$ is the Hilbert space of square integrable functions with the usual inner product, $\langle \phi, \psi \rangle_{L^2} := \int \phi(x)\psi(x)\rho(x)\mathrm{d}x$, where ϕ and ψ are scalar valued functions in $L^2(\mathsf{X}, \rho)$; written L^2 when there is no risk of ambiguity. The associated norm is denoted $\|\phi\|_{L^2}^2 := \langle \phi, \phi \rangle_{L^2}$. The inner product is extended to vector valued functions $f, g : \mathsf{X} \to \mathbb{R}^d$ by defining, $\langle f, g \rangle_{L^2} := \sum_{k=1}^d \langle f_k, g_k \rangle_{L^2}$. The expectation of a function f with respect to ρ is written as $\mathsf{E}_{X\sim\rho}[f(X)]$ and the notation E_x is used as shorthand for the conditional expectation $\mathsf{E}_x[f(X_t)] := \mathsf{E}[f(X_t)|X_0 = x]$. We use $\|\cdot\|$ without the subscript to refer to the Euclidean norm and $\|\cdot\|_\infty$ refers to the maximum or uniform norm.

Another Hilbert space $\mathcal{H}$ will appear in the application of reproducing kernel Hilbert space theory with its inner product denoted $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, and induced norm $\|\cdot\|_{\mathcal{H}}$. C^k is used to denote the space of k-times continuously differentiable functions $f\colon \mathsf{X} \to \mathbb{R}$, ∇f denotes the gradient of a function $f \in C^1$, and Δf the Laplacian for $f \in C^2$. If $\mathsf{X} = \mathbb{R}$, the notation f' and f'' are used to denote the first and second derivatives respectively.

CHAPTER 2
DIFFERENTIAL TD LEARNING

A basic introduction of the essential elements of this dissertation was provided in Chapter 1. In this chapter, a more formal mathematical description of the concepts is aimed at. The outline of this chapter is:

(i) In Section 2.1, a prerequisite introduction to the Langevin diffusion and its associated Poisson's equation is given. The intention is to motivate why solving this equation is important to the two applications of interest.

(ii) The standard least squares temporal difference (LSTD) learning algorithm is described next. This algorithm forms the basis of techniques to approximate the solution to Poisson's equation. A detailed derivation is presented, as a precursor to deriving a new variant.

(iii) We derive the differential LSTD (∇-LSTD) learning algorithm that tries to approximate the gradient of the solution to Poisson's equation directly rather than the function itself. It is shown in Section 2.3.1, that in the special case of Langevin diffusion, the ∇-LSTD-L algorithm provides a simple and elegant solution. This is achieved by using the self-adjoint property of its differential generator. This is the major contribution in this chapter.

(iv) A more general version of the ∇-LSTD algorithm, that can be applied to a broad class of continuous-time diffusion processes is also derived along the lines of the standard LSTD algorithm. This algorithm with its application to gain function approximation in the FPF was published as a conference proceeding [27]. A discrete-time analog of the algorithm has been successfully applied to problems in optimal control.

2.1 Langevin Diffusion and Poisson's Equation

In this section, we introduce the Langevin diffusion and its associated Poisson's equation with only the requisite amount of detail, so that the broad motivation is not obfuscated by technical details. A more elaborate discussion is reserved for Section 5.1 in the context of MCMC algorithms.

2.1.1 Langevin Diffusion

The Langevin diffusion may be regarded as a d-dimensional gradient flow perturbed with "noise", described by the SDE,

$$\mathrm{d}\Phi_t = \underbrace{-\nabla U(\Phi_t)\,\mathrm{d}t}_{\text{drift}} + \underbrace{\sqrt{2}\,\mathrm{d}W_t}_{\text{diffusion}}, \tag{2-1}$$

where $\boldsymbol{W} = \{W_t : t \geq 0\}$ is a standard Brownian motion on $\mathbb{R}^d$. The potential function $U : \mathsf{X} \to \mathbb{R}$ is continuously differentiable. Under suitable regularity conditions, this diffusion is reversible and has a unique invariant density $\rho = e^{-U+\Lambda}$, where Λ is a normalizing constant so that ρ integrates to unity [28]. The SDE in (2-1) can be thought of as composed of a deterministic drift term and a stochastic diffusion term. The intuition is that the drift term moves the process along the direction in which the density ρ increases. In this sense, it is a biased random walk. In practice, as simulating path solutions to this SDE is difficult, discretized versions of the equation based on Euler-Mauryama scheme are used (2-2):

$$\Phi_n = \Phi_{n-1} - \nabla U(\Phi_{n-1})\delta_n + \sqrt{2\delta_{n-1}}W_{n-1}, \tag{2-2}$$

where $\{\delta_n\}_{n\geq 1}$ is a sequence of step sizes and $\{W_n\}_{n\geq 1}$ is a sequence of i.i.d. standard Gaussian random variables. In this dissertation, only implementations using a constant step size parameter $\delta_n \equiv \delta$ are considered.

The Langevin diffusion is associated with a differential generator $\mathcal{D}$ (also called infinitesimal generator), which is defined as,

$$\begin{aligned} \mathcal{D}f &:= \lim_{t\to 0} \frac{\mathsf{E}[f(\Phi_t) - f(x)|\Phi_0 = x]}{t} \\ &= -\nabla U \cdot \nabla f + \Delta f, \qquad f \in C^2, \end{aligned} \tag{2-3}$$

where ∇ denotes the gradient and Δ is the Laplacian. The differential generator can be thought of as the derivative operator in an expected sense. Under conditions on U, the SDE (2-1) defines a strong Markov semigroup $\{P_t\}_{t\geq 0}$. The generator $\mathcal{D}$ can be written in terms of

the semigroup as,

$$\mathcal{D} = \lim_{t \to 0} \frac{P_t - I}{t}. \tag{2-4}$$

2.1.2 Poisson's Equation

Let $c\colon \mathsf{X} \to \mathbb{R}$ be a function of interest, and

$$\eta := \mathsf{E}_{\Phi \sim \rho}[c(\Phi)] = \int_{\mathsf{X}} c(x)\rho(x)\mathsf{d}x = \langle c, 1 \rangle_{L^2}. \tag{2-5}$$

A function $h \in C^2$ is said to be the solution to Poisson's equation with forcing function c if it satisfies,

$$\mathcal{D}h := -\tilde{c}, \qquad \tilde{c} = c - \eta. \tag{2-6}$$

Additionally, h can also be expressed in the following integral form:

$$h(x) = \int_0^\infty \mathsf{E}_x[\tilde{c}(\Phi_t)]\mathsf{d}t, \tag{2-7}$$

where $h(x)$ can be interpreted as the infinite-horizon expected normalized cost with the initial state $\Phi_0 = x$. The notation $\mathsf{E}_x[\tilde{c}(\Phi_t)]$ is shorthand for the conditional expectation $\mathsf{E}[\tilde{c}(\Phi_t)|\Phi_0 = x]$, with x as the initial state. The function h is also called the relative value function in average-cost optimal control.

The existence of a solution h in a weak sense holds under very weak assumptions on U and c [15, 29]. Glynn et al. in [15] provide Lyapunov bounds for the solution h. Representations for the gradient of h and its bounds are obtained in [30, 31]. The existence of a smooth solution $h \in C^2$ has been established under stronger conditions in [32], subject to growth conditions on c similar to those used in [15]. In the remaining part of this dissertation, existence of h is assumed.

Analogous to h, a discounted-cost value function h^γ is defined as,

$$h^\gamma(x) := \int_0^\infty \exp(-\gamma t)\mathsf{E}_x[c(\Phi_t)]\mathsf{d}t, \tag{2-8}$$

with a discount rate $\gamma > 0$ and $c(x)$ is the one-step cost function at the state x. The discounted-cost value function is often more popular in applications where the future costs incurred are assigned lower weights than the immediate costs. It may be noted that as the value of $\gamma \to 0$, it closely approximates an average-cost problem.

2.1.3 Relevance to our Applications

The solution h of Poisson's equation associated to the Langevin diffusion (2-6) is crucial in each of the applications we consider in this dissertation. In the feedback particle filter (FPF), the innovations gain function K at each t is obtained as the gradient of h [1]:

$$\mathsf{K}(x) = \nabla h(x)\,, \quad x \in \mathsf{X}\,. \tag{2-9}$$

A detailed analysis of Poisson's equation appearing in the FPF is performed in [30]. The FPF is described in detail in Chapter 4.

In MCMC algorithms, as mentioned in Chapter 1, the asymptotic variance is a measure of convergence. In this context, ρ is the target density, and c is the function whose expectation needs to be computed. The expected value is approximated using the empirical mean η_N:

$$\eta_N := \frac{1}{N}\sum_{n=0}^{N-1} c\,(\Phi_n), \tag{2-10}$$

where $\Phi_n \sim \rho$, is obtained by sampling from an ergodic Markov chain with ρ as its invariant density. One such Markov chain is the discretization in (2-2), also known as the unadjusted Langevin algorithm (ULA) or Langevin Monte Carlo (LMC). Under general conditions, the mean estimates will obey a Central Limit Theorem (CLT) of the form,

$$\sqrt{N}\Big(\eta_N - \eta\Big) \xrightarrow{d} \mathcal{N}(0, \sigma_\infty^2), \tag{2-11}$$

where the convergence is in distribution [28, 33]. Here, σ_∞^2 denotes the asymptotic variance that has a representation in terms of h [15, 33, 34]. It has been noted in [14, 16] that estimates for the solution h can be used to construct algorithms that provide estimators with

improved asymptotic variance. More details about the application to MCMC algorithms is in Chapter 5.

Computation of h is typically intractable, especially in higher dimensions and hence, approximation approaches are required. Algorithms based on reinforcement learning provide a suitable approach. Section 2.2 reviews the least-squares TD (LSTD) learning algorithm that attempts to obtain the best approximation in an $L^2(\rho)$-norm sense. The LSTD algorithm has limitations, which curtails its scope in using it for our applications. To overcome this, a new class of algorithms, called differential LSTD (∇-LSTD) learning, based on the idea of approximating the gradient of h directly is proposed. The algorithm for the case of Langevin diffusion is derived in Section 2.3.1. A more general version that can be applied to a continuous-time diffusion process is described in Section 2.3.3.

2.2 Least Squares Temporal Difference (LSTD) Learning

2.2.1 LSTD for Discounted Cost Value Function

In this section, the least squares temporal difference (LSTD) learning algorithm to approximate the discounted-cost value function (2-8) of a continuous-time diffusion process is presented. Theory for TD learning in the discounted cost setting is largely complete, however theory and algorithms for the average cost case is more fragmented. The LSTD algorithm was first proposed for a finite-state-action space Markov decision process (MDP) in the context of stochastic optimal control by Bradtke et al. in [21]. It is noted in [21, 35] that the LSTD algorithm, although more computationally expensive than the conventional TD learning algorithm of Sutton [19], is statistically more efficient and also avoids the choice of a tunable step-size parameter.

The derivation of the algorithm here closely follows the one in Section 11.5.2 in [13]. The key difference is that the book section discusses the discrete-time case. Consider a one-dimensional continuous-time diffusion on $\mathbb{R}$:

$$\mathrm{d}\Phi_t = a(\Phi_t)\mathrm{d}t + \sigma(\Phi_t)\mathrm{d}B_t, \tag{2-12}$$

where $\boldsymbol{B}$ is standard Brownian motion and $a : \mathbb{R} \to \mathbb{R}$ is a Lipschitz-continuous function. The differential generator $\mathcal{D}$ is defined for functions $f : \mathbb{R} \to \mathbb{R}$ as:

$$\mathcal{D}f = af' + \frac{\sigma^2}{2} f'', \qquad f \in C^2. \tag{2-13}$$

It may be noted that the Langevin diffusion (2-1) is a special case of the diffusion (2-12) with $a(x) = -\nabla U(x)$ and $\sigma(x) \equiv \sqrt{2}$, and the associated differential generator is defined in (2-3).

Consider a parameterized family of approximations $\{h^\theta : \theta \in \mathbb{R}^\ell\}$. In the case of a linear parameterization, where we have ℓ functions on X as the basis, denoted $\{\psi_i : 0 \le i \le \ell\}$, the parameterized family $\{h^\theta\}$ becomes,

Question:Using X and $\mathbb{R}$ interchangably, which one is better?

$$h^\theta(x) := \sum_{i=1}^{\ell} \theta_i \psi_i(x) = \theta^\intercal \psi(x), \qquad x \in \mathsf{X} \tag{2-14}$$

The goal in LSTD learning algorithm is to minimize the approximation error in $L^2(\rho)$ norm:

$$\begin{aligned} \mathcal{E}(\theta) &:= \|h^\gamma - h^\theta\|_{L^2}^2 \\ &= \int_{\mathsf{X}} (h^\gamma(x) - h^\theta(x))^2 \rho(x) \mathrm{d}x \\ &= \mathsf{E}_{\Phi \sim \rho}[|h^\gamma(\Phi) - h^\theta(\Phi)|^2]. \end{aligned} \tag{2-15}$$

Among the conventional TD algorithms, only TD(1) has an interpretation as a norm minimization problem. It is evident from the definition of $\mathcal{E}(\theta)$ that it penalizes the approximation error more strongly for states with larger invariant probability $\rho(x)$. As noted in [13], these are the states that are visited more often and hence, a better approximation of h^γ is desired at these points on X. A more important benefit of using the $L^2(\rho)$ norm is that it allows the construction of an algorithm using Monte Carlo methods.

The necessary conditions for optimality can be obtained by taking the gradient of (2-15) with respect to θ and equating it to zero,

$$
\begin{aligned}
0 &= -\nabla_\theta \mathcal{E}(\theta) \\
&= -2\|(h^\gamma - h^\theta)\nabla_\theta h^\theta\|_{L^2} \\
&= -2\mathsf{E}_{\Phi\sim\rho}[(h^\gamma(\Phi) - h^\theta(\Phi))\nabla_\theta h^\theta(\Phi)]
\end{aligned} \tag{2-16}
$$

Question:Under what conditions can we exchange the derivative and expectation for a nonlinear parameterization?

In general, a solution to (2-16) can be obtained using recursive updates of the parameter θ by stochastic approximation techniques. However, for the linear parameterization (2-14), the optimal θ^* admits a closed form expression:

$$
\theta^* := M^{-1}b, \tag{2-17}
$$

where M and b are defined as,

$$
M := \mathsf{E}_{\Phi\sim\rho}[\psi(\Phi)\psi(\Phi)^\intercal], \qquad b := \mathsf{E}_{\Phi\sim\rho}[h^\gamma(\Phi)\psi(\Phi)]. \tag{2-18}
$$

The expression for b is not computable as it involves the unknown function h^γ that we are trying to approximate. The challenge now is to find an alternate observable representation for b.

A resolution is obtained using the definition of h^γ (2-8) and the generalized resolvent kernel of [36–38]: For a measurable function $G\colon \mathbb{R} \to \mathbb{R}$, and measurable functions f in some domain, the resolvent kernel R_G is an operator defined as,

$$
R_G f\,(x) := \int_0^\infty \mathsf{E}_x\Big[\exp\Big(-\int_0^t G(\Phi_s)\,\mathrm{d}s\Big) f(\Phi_t)\Big]\mathrm{d}t. \tag{2-19}
$$

In [36, 37] it is assumed that $G > 0$ everywhere. These conditions are relaxed in [38, 39]. An adjoint operator $R_G^\dagger$ is defined such that the following holds:

$$
\langle R_G f,\, g\rangle_{L^2} = \langle f,\, R_G^\dagger g\rangle_{L^2}, \tag{2-20}
$$

where f and g are in $L^2(\rho)$.

The discounted-cost value function h^γ in (2-8) can now be represented in terms of R_G (2-19) with $G \equiv \gamma$ as,

$$h^\gamma = R_\gamma c. \tag{2-21}$$

If the value function h^γ is C^2, then it solves the discounted-cost optimality equation,

$$\gamma h^\gamma - \mathcal{D}h^\gamma = c. \tag{2-22}$$

This is analogous to Poisson's equation that arises in the average-cost case. Using (2-21) and (2-22), resolvent kernel R_γ can be shown to satisfy the following inverse formula:

$$R_\gamma c = (I_\gamma - \mathcal{D})^{-1} c, \tag{2-23}$$

where I_γ refers to multiplication by γ. The inverse $(I_\gamma - \mathcal{D})^{-1}$ exists on some domain under conditions provided in [38]. The following transformation can be applied to b, using (2-21) followed by an adjoint trick,

$$\begin{aligned} b = \mathsf{E}_{\Phi\sim\rho}[h^\gamma(\Phi)\psi(\Phi)] &= \langle h^\gamma,\, \psi\rangle_{L^2} \\ &= \langle R_\gamma c,\, \psi\rangle_{L^2} \\ &= \langle c,\, R_\gamma^\dagger \psi\rangle_{L^2}, \end{aligned} \tag{2-24}$$

where $R_\gamma^\dagger$ denotes the adjoint of R_γ. Now, all that remains is to obtain an expression for the adjoint operator $R_\gamma^\dagger$ in terms of observable quantities. This is achieved by the application of Lemma 1.

Lemma 1. *Let $\mathbf{\Phi} = \{\Phi_t : t \in \mathbb{R}\}$ denote a stationary version of a continuous-time diffusion process. For measurable functions f, g with at most exponential growths we have,*

$$\langle R_\gamma f,\, g\rangle_{L^2} = \langle f,\, R_\gamma^\dagger g\rangle_{L^2} = \mathsf{E}_{\Phi\sim\rho}\left[f(\Phi_t)\varphi_g(t)\right], \quad t \in \mathbb{R}, \tag{2-25}$$

wherein $\boldsymbol{\varphi}_g$ is the stationary process:

$$\varphi_g(r) := \int_0^\infty \exp(-\gamma(t-r))g(\Phi_{r-t})dt, \tag{2-26}$$

Consequently,

$$R_\gamma^\dagger g(x) = \mathsf{E}[\varphi_g(t)|\Phi_t = x] \tag{2-27}$$

Proof. This is achieved by applying the stationarity property of the process Φ_t, which gives,

$$\mathsf{E}_{\Phi\sim\rho}[f(\Phi_t)g(\Phi_0)] = \mathsf{E}_{\Phi\sim\rho}[f(\Phi_0)g(\Phi_{-t})]. \tag{2-28}$$

The proof is obtained by rewriting b in its integral form as,

$$\begin{aligned}
\langle R_\gamma f,\, g\rangle_{L^2} &= \int_0^\infty \exp(-\gamma t)\Big(\int_{\mathsf{X}} \mathsf{E}[f(\Phi_t)|\Phi_0 = x]g(x)\rho(x)\mathsf{d}x\Big)\mathsf{d}t \\
&= \int_0^\infty \exp(-\gamma t)\Big(\int_{\mathsf{X}} \mathsf{E}[f(\Phi_t)g(\Phi_0)|\Phi_0 = x]\rho(x)\mathsf{d}x\Big)\mathsf{d}t \\
&= \int_0^\infty \exp(-\gamma t)\mathsf{E}_{\Phi\sim\rho}[f(\Phi_t)g(\Phi_0)]\mathsf{d}t \\
&= \int_0^\infty \exp(-\gamma t)\mathsf{E}_{\Phi\sim\rho}[f(\Phi_0)g(\Phi_{-t})]\mathsf{d}t \qquad \text{(Applying the stationarity property of } \Phi) \\
&= \mathsf{E}_{\Phi\sim\rho}\Big[f(\Phi_0)\underbrace{\int_0^\infty \exp(-\gamma t)g(\Phi_{-t})\mathsf{d}t}_{\varphi_g(0)}\Big] \qquad \text{(Applying Fubini's theorem and absolute integrability)} \\
&= \langle f, R_\gamma^\dagger g\rangle_{L^2}.
\end{aligned} \tag{2-29}$$

It can be seen from (2-29) that the adjoint $R_\gamma^\dagger$ operating on a function g in $L^2(\rho)$ takes the form:

$$R_\gamma^\dagger g(x) := \int_0^\infty \exp(-\gamma t)\mathsf{E}_x[g(\Phi_{-t})]\mathsf{d}t \tag{2-30}$$

■

Thus, the adjoint $R_\gamma^\dagger$ can be interpreted as the resolvent for the time-reversed process $\{\Phi_{-t}\}$. If we denote φ_ψ as,

$$\varphi_\psi(r) := \int_0^\infty \exp(-\gamma(t-r))\psi(\Phi_{r-t})\mathsf{d}t, \tag{2-31}$$

b in (2-24) can be written as,

$$b = \mathsf{E}[c(\Phi_0)\varphi_\psi(0)]. \tag{2-32}$$

The function φ_ψ is called the eligibility vector in TD learning. A slightly more general proof using the generalized resolvent kernel R_G appears in the derivation of ∇-LSTD learning in Section 2.3.3.

This representation for b (2-32), combined with the representation for M in (2-18) lends itself to application of Monte Carlo methods in their computation. Monte carlo approximations to M and b can be obtained using the following integral forms:

$$M \approx \frac{1}{T}\int_0^T \psi(\Phi_t)\psi^\intercal(\Phi_t)\mathrm{d}t \tag{2-33a}$$

$$b \approx \frac{1}{T}\int_0^T c(\Phi_t)\varphi_\psi(t)\mathrm{d}t \tag{2-33b}$$

A recursive formulation of the algorithm can be provided in terms of the three ODEs:

$$\frac{d}{dt}\varphi_\psi(t) = -\gamma\,\varphi_\psi(t) + \psi(\Phi_t) \tag{2-34a}$$

$$\frac{d}{dt}b(t) = c(\Phi_t)\varphi_\psi(t) \tag{2-34b}$$

$$\frac{d}{dt}M(t) = \psi(\Phi_t)\psi^\intercal(\Phi_t) \tag{2-34c}$$

$$\theta(t) := M(t)^{-1}b(t) \tag{2-34d}$$

Equations (2-34a–2-34d) summarize the LSTD algorithm. The system of ODEs is initialized with $\varphi_\psi(0), b(0) \in \mathbb{R}^\ell$, and a positive definite $\ell \times \ell$ matrix $M(0)$. The computational complexity arising due to matrix inversion operation in (2-34d) can be reduced by applying the matrix inversion lemma, as pointed out in [13].

The ODE in (2-34a) that governs the evolution of the eligibility vector $\varphi_\psi(t)$ is equivalent to the one that appears in TD(λ) algorithm [19], with the exponential "forgetting factor" $\lambda = 1$. Convergence of the parameter estimates $\theta(t)$ to θ^* in the limit as t goes to ∞ has been shown by the application of law of large numbers.

Anand:needs reading

A linear parameterization for h^θ is not essential for the LSTD algorithm. For a nonlinear parameterization, LSTD can be implemented as a stochastic approximation recursion. A

discussion of nonlinear parameterization is skipped here and reserved for Section 2.3.3 while discussing the differential TD learning algorithm.

2.2.2 LSTD for Poisson's Equation

The LSTD algorithm was presented in the context of the discounted-cost value function in Section 2.2. To approximate the average-cost value function h (2-6), the common practice is to use a discounted formulation as a proxy. The discount rate γ is usually set very close to zero with the intention of mimicking the average cost problem. However, it is known the variance of the algorithm diverges as $\gamma \to 0$. In the paper by Tsitsikilis et al. [40], a variant of the TD(λ) learning algorithm for the average-cost case is presented for the case of finite state space Markov chains. It is mentioned in the conclusions that extensions to a general state space is easily possible, but it only considers the case where the exponential weighting factor $\lambda < 1$. Only when $\lambda = 1$, the algorithm can be interpreted as minimizing the $L^2(\rho)$ norm of the approximation error $\mathcal{E}$ (2-15).

The LSTD algorithm for the average cost case, that involves Poisson's equation is also discussed in Section 11.5.4 of [13]. However, this algorithm yields asymptotically unbiased estimators only with the underlying assumption of the existence of a regenerating state. In informal terms, a regenerating state of a Markov chain is one that is visited infinitely often and when visited, the chain can be thought of as forgetting the past, i.e. once the regenerating state is reached, the future transitions of the chain are statistically independent from its past and hence, its entire history may be discarded. This requirement impedes the applicability of the algorithm, if the dimension of the state space is greater than one. This motivates the development of algorithms that can overcome these shortcomings in Sections 2.3.1 and 2.3.3.

2.3 Differential TD (∇-LSTD) Learning

In this Section, we describe the differential LSTD (∇-LSTD) learning algorithm, which tries to approximate the gradient of the solution to Poisson's equation (2-6) directly. The development of this algorithm is motivated by two factors - i) the shortcomings of the LSTD algorithm for dimensions > 1, and ii) applications that only require the gradient of the solution

like the FPF. In conventional applications, where we are interested in h instead of ∇h, in addition to obtaining the optimal parameter values, we also need to add an optimal constant term. Keeping the same notation in the previous section, the goal of the ∇-LSTD learning algorithm is:

$$\theta^* = \arg\min_{\theta} \|\nabla h - \nabla h^\theta\|_{L^2}^2. \tag{2-35}$$

2.3.1 ∇-LSTD Learning for Langevin Diffusion (∇-LSTD-L)

In this Section, we first focus on the special case of differential TD learning algorithm for the Langevin diffusion (2-1), denoted as ∇-LSTD-L in the remainder of this dissertation. A more generic version of ∇-LSTD learning that is applicable to any continuous-time diffiusion process is described in Section 2.3.3. For the Langevin diffusion, a property of its differential generator (2-3), described in Prop. 2.1 allows the construction of a simple algorithm. The advantages of this simpler resolution will be more evident after the general version is presented in Section 2.3.3. Prop. 2.1 is a corollary to the result in [41, 42] and can be proved by a simple application of the integration by parts formula:

Proposition 2.1. *For a Langevin diffusion with differential generator $\mathcal{D}$ (2-3), suppose that $f, g\colon \mathbb{R}^d \to \mathbb{R}$ are in $C^2 \cap L^2$, and that their first and second partial derivatives also lie in L^2. Then,*

$$\langle \nabla f, \nabla g \rangle_{L^2} = \sum_{k=1}^{d} \Big\langle \frac{\partial f}{\partial x_k}, \frac{\partial g}{\partial x_k} \Big\rangle_{L^2} = -\langle f, \mathcal{D} g \rangle_{L^2} = -\langle \mathcal{D} f, g \rangle_{L^2}. \tag{2-36}$$

■

Proof. We can assume that f and g have compact support. The extension to arbitrary functions satisfying the assumptions of the proposition is obtained by approximation in $L^2(\rho)$. In the following, $\partial_k f$ is used as shorthand for $\frac{\partial f}{\partial x_k}$.

Question: I found this comment in one of the notes you wrote . What does this comment mean?

Consider first the scalar case, where $d = 1$:

$$
\begin{aligned}
\sum_{k=1}^{d}\langle\partial_k f,\partial_k g\rangle_{L^2} &= \langle f',g'\rangle_{L^2}\\
&= \int_{-\infty}^{\infty} f'(x)g'(x)\rho(x)dx\\
&= \int_{-\infty}^{\infty} (g'(x)\rho(x))f'(x)dx\\
&= -\int_{-\infty}^{\infty} (g'(x)\rho'(x)+g''(x)\rho(x))f(x)dx, \qquad (g'\rho f\big|_{-\infty}^{\infty}=0)\\
&= -\int_{-\infty}^{\infty} \Big(\frac{g'(x)\rho'(x)}{\rho(x)}+g''(x)\Big)f(x)\rho(x)dx\\
&= -\int_{-\infty}^{\infty} (-U'(x)g'(x)+g''(x))f(x)\rho(x)dx, \qquad (U(x)=-\log\rho(x))\\
&= -\int_{-\infty}^{\infty} \mathcal{D}g(x)f(x)\rho(x)dx\\
&= -\langle f,\mathcal{D}g\rangle
\end{aligned}
\tag{2-37}
$$

It follows by symmetry that $\langle f',g'\rangle_{L^2} = -\langle \mathcal{D}f, g\rangle_{L^2}$.

In the multidimensional case, we make use of the following version of integration by parts:

$$
\int\dots\int\left(\int_{-\infty}^{\infty} g\,\partial_k f\,dx_k\right)dx_{\bar{k}} = -\int\dots\int\left(\int_{-\infty}^{\infty} f\,\partial_k g\,dx_k\right)dx_{\bar{k}} \tag{2-38}
$$

Applying the above relation, with g replaced by $g'\rho$, we get,

$$
\begin{aligned}
\langle\partial_k f,\partial_k g\rangle_{L^2} &= \int\dots\int (\partial_k f(x))\,(\partial_k g(x)\rho(x))\,dx\\
&= -\int\dots\int f(x)\{\partial_k^2 g(x)-\partial_k U(x)\partial_k g(x)\}\rho(x)dx
\end{aligned}
\tag{2-39}
$$

Summing over k gives the desired conclusion:

$$
\sum_{k=1}^{d}\langle\partial_k f,\partial_k g\rangle_{L^2} = -\langle f,\mathcal{D}g\rangle = -\langle\mathcal{D}g,f\rangle_{L^2}. \tag{2-40}
$$

■

The equality on the right side is a result of the self-adjoint property of the Langevin generator $\mathcal{D}$. This can also be proved using the reversibility property of the diffusion.

2.3.2 Linear Parameterization

If we assume a linear parameterization of the form in (2-14) with $\theta \in \mathbb{R}^\ell$ as the parameters and $\{\psi_i \in C^2 : 1 \le i \le \ell\}$ as the basis functions, the optimization problem described by (2-35) takes the quadratic form in (2-41). Then an adjoint argument provided by Prop. 2.1 leads to a representation that can be applied for computation.

Lemma 2. *The norm appearing in* (2-35) *is a quadratic form,*

$$\|\nabla h - \nabla h^\theta\|_{L^2}^2 = \theta^\intercal M\theta - 2b^\intercal\theta + k, \tag{2-41}$$

in which for each $1 \le i, j \le \ell$*,*

$$M_{i,j} = \langle \nabla\psi_i, \nabla\psi_j\rangle_{L^2}, \quad b_i = \langle \tilde{c}, \psi_i\rangle_{L^2}, \tag{2-42}$$

and $k = \|\nabla h\|_{L^2}^2$*. Consequently, the optimizer* (2-35) *is any solution to*

Question:any solution or the solution?

$$M\theta^* = b. \tag{2-43}$$

■

We assume henceforth that the basis is linearly independent in $L^2(\rho)$, so that M is invertible, and hence $\theta^* = M^{-1}b$. Using Prop. 2.1 and Poisson's equation (2-6), b_i in (2-42) has an alternate representation in terms of known functions:

$$\begin{aligned} b_i &= \langle \nabla h, \nabla\psi_i\rangle_{L^2} \\ &= -\langle \mathcal{D}h, \psi_i\rangle_{L^2} \\ &= \langle \tilde{c}, \psi_i\rangle_{L^2}. \end{aligned} \tag{2-44}$$

The expressions for M and b in (2-42), permit the construction of Monte Carlo based approximations to implement the ∇-LSTD-L algorithm. The centered function $\tilde{c}$ can be approximated using its empirical equivalent $\tilde{c}_T$ defined as,

$$\tilde{c}_T(x) := c(x) - \frac{1}{T}\int_0^T c(\Phi_t)dt, \tag{2-45}$$

where Φ_t is distributed according to the density ρ. The matrix M and vector b can be approximated using the following integral forms,

$$M \approx M_T := \frac{1}{T}\int_0^T \big(\nabla\psi(\Phi_t)\big)\big(\nabla\psi(\Phi_t)\big)^\intercal dt, \tag{2-46}$$

$$b \approx b_T := \frac{1}{T}\int_0^T \tilde{c}_T(\Phi_t)\psi(\Phi_t)dt, \tag{2-47}$$

and $\theta_T = M_T^{-1}b_T$. We shall discuss the benefits and limitations of this algorithm after presenting the more generic version in Section 2.3.3.

2.3.3 ∇-LSTD Learning for a General Diffusion (∇-LSTD)

In this Section, we present the differential TD learning algorithm for a general continuous-time diffusion process. This generic algorithm is denoted as ∇-LSTD in the remainder of this dissertation. The ideas involved in the derivation are similar to those in the LSTD algorithm. A derivation of the ∇-LSTD algorithm for a discrete-time MDP with examples from optimal control is provided in [22]. This algorithm is discussed in the context of gain function approximation for the FPF in Chapter 4 and in [27].

The presentation of the algorithm here is restricted to the scalar case, where $\mathsf{X} = \mathbb{R}$. However, under general conditions, it is expected that the algorithm can be extended to higher dimensions. Consider a general continuous-time diffusion process defined in (2-12) with the definition for its associated differential generator $\mathcal{D}$ in (2-13). Denote by h the solution to Poisson's equation (2-6) with forcing function $c \in C^1$. Differentiating each side of (2-6) with respect to x, we obtain

Question:Should I say anything more?

$$\begin{aligned}\frac{d}{dx}(\mathcal{D}h) &= a'h' + ah'' + \frac{\sigma^2}{2}h''' \\ &= a'h' + \mathcal{D}h' = -c'.\end{aligned} \tag{2-48}$$

In operator theoretic notation, this can be written as,

$$(I_{-a'} - \mathcal{D})h' = c'. \tag{2-49}$$

It is required that c' has at most exponential growth [31]. We say that a function $f\colon \mathbb{R} \to \mathbb{R}$ has at most exponential growth if

$$\sup_x \frac{\log(1+|f(x)|)}{1+|x|} < \infty \tag{2-50}$$

Additionally, if a satisfies regularity assumptions in Prop. 2.2, the derivative h' has the following representation in terms of the resolvent kernel R_G (2-19) with $G \equiv -a'$.

$$h' = R_{-a'}c' \tag{2-51}$$

Notice that the resolvent representation is obtained here for h' as opposed to h in (2-21). The remainder of this derivation proceeds by taking $a' = -U''$, which is true for the special case of Langevin diffusion. However, nowhere is it required that a' takes this form. The regularity conditions on U are described in Prop. 2.2.

Proposition 2.2. *Suppose that $U\colon \mathbb{R} \to \mathbb{R}$ satisfies the following assumptions:*

(i) *U is C^2 with $\sup_x |U''(x)| < \infty$.*

(ii) *For some $\varepsilon > 0$,*

$$U''(x) \geq \varepsilon, \qquad \text{for } \ |x| \geq \varepsilon^{-1}. \tag{2-52}$$

Suppose moreover that c' is continuous, and has at most exponential growth. Then $R_{U''}c'$ is finite valued, and for any $n \geq 1$ we have

$$\int \left|R_{U''}c'\,(x)\right| \exp(n|x|)\,\rho(x)\,dx < \infty \tag{2-53}$$

■

Now, the derivation of the algorithm follows the same steps involved in the derivation of LSTD in Section 2.2. An adjoint $R^\dagger_{U''}$ is required that satisfies (2-20) for $G \equiv U''$. Lemma 3 provides the expression for the adjoint $R^\dagger_{U''}$.

Lemma 3. *Let $\mathbf{\Phi} = \{\Phi_t : t \in \mathbb{R}\}$ denote a stationary version of the Langevin diffusion. For measurable functions f, g with at most exponential growths we have,*

$$\langle g, R_{U''} f \rangle_{L^2} = \mathsf{E}[f(\Phi_t)\varphi_g(t)]\,, \quad t \in \mathbb{R}, \tag{2-54}$$

wherein φ_g is the stationary process:

$$\varphi_g(t) = \int_{-\infty}^{t} \exp\Big(-\int_r^t U''(\Phi_s)\ ds\Big) g(\Phi_r)\ dr \tag{2-55}$$

Consequently,

$$R^{\dagger}_{U''} g(x) = \mathsf{E}[\varphi_g(t) | \Phi_t = x] \tag{2-56}$$

Proof. For ease of notation, we denote

$$\mathcal{U}_r^t := \int_r^t U''(\Phi_s)\, \mathsf{d}s. \tag{2-57}$$

Based on this definition, and using (2-19),

$$\begin{aligned}
&\langle g, R_{U''} f \rangle_{L^2} \\
&= \mathsf{E}[g(\Phi_0) R_{U''} f(\Phi_0)] \\
&= \mathsf{E}\Big[[g(\Phi_0) \int_0^{\infty} \mathsf{E}\big[\exp\big(-\mathcal{U}_0^{\tau}\big) f(\Phi_\tau)) \mid \Phi_0\big]\, d\tau\Big] \\
&= \int_0^{\infty} \mathsf{E}\Big[f(\Phi_\tau) \exp\big(-\mathcal{U}_0^{\tau}\big) g(\Phi_0)\Big]\ \mathsf{d}\tau.
\end{aligned}$$

Interchanging the expectation and the integral requires Fubini's theorem, which is justified by Prop. 2.2. The change of variables $r = t - \tau$ gives

$$\langle g, R_{U''} f \rangle_{L^2} = \int_{-\infty}^{t} \mathsf{E}\Big[f(\Phi_{t-r}) \exp\big(-\mathcal{U}_0^{t-r}\big) g(\Phi_0)\Big]\ \mathsf{d}r, \tag{2-58}$$

and applying stationarity,

$$\langle g, R_{U''} f \rangle_{L^2} = \int_{-\infty}^{t} \mathsf{E}\Big[f(\Phi_t) \exp\big(-\mathcal{U}_r^{t}\big) g(\Phi_r))\Big]\ \mathsf{d}r. \tag{2-59}$$

The proof of (2-54) is complete via a second application of Fubini's theorem. ∎

It is worthwhile to note that the derivation of the LSTD algorithm is a special case of this lemma with $G \equiv \gamma$. This leads to an interesting interpretation of the exponential term $\exp(-\mathcal{U}_r^t)$ as implicitly introducing a "discounting factor" to the average-cost problem. As a result of this discounting, the existence of a regenerating state to the diffusion is guaranteed, which makes it applicable to problems in higher dimensions.

Question:Can I say $U'' = \gamma$?

Question:Can you check this statement?

The ∇-LSTD learning algorithm provides an unbiased and asymptotically consistent estimate of θ^*. The algorithm is defined by the set of ODEs:

$$\frac{d}{dt}\varphi(t) = -U''(\Phi_t)\varphi(t) + \psi'(\Phi_t) \tag{2-60a}$$

$$\frac{d}{dt}b(t) = \varphi(t)c'(\Phi_t) \tag{2-60b}$$

$$\frac{d}{dt}M(t) = \psi'(\Phi_t)\psi'^{\intercal}(\Phi_t) \tag{2-60c}$$

The vector $\varphi(t)$ is analogous to the eligibility vector in TD learning [13, 43]. Existence of a steady state solution to (2-60a) of the form in (2-55) is guaranteed under a Lyapunov drift condition in [31]. The estimates of θ^* are generated via $\theta(t) = M(t)^{-1}b(t)$. The ODE is initialized with $\varphi(0), b(0) \in \mathbb{R}^\ell$, and $M(0) > 0$ a $\ell \times \ell$ matrix. It is interesting to note that the eligibility vector term φ_t is absent in the ∇-LSTD-L algorithm.

Anand:need to place this somewhere else

2.3.4 Nonlinear Parameterization

Consider a more general case of nonlinear parameterization with an ℓ- dimensional function class $\mathcal{H} := \{h^\theta : \theta \in \mathbb{R}^\ell\}$. In this case, the optimization problem (2-35) may not be convex. Let ψ_θ denote the gradient of h^θ with respect to θ, i.e. $\nabla_\theta h^\theta = \psi_\theta$. For a nonlinear parameterization, following the first order optimality conditions for an optimizer θ^* of (2-35), we have,

$$\begin{aligned} 0 &= -\nabla_\theta \mathcal{E} \\ &= \mathsf{E}\big[\big(h'(\Phi) - h'_{\theta^*}(\Phi)\big)\psi'_{\theta^*}(\Phi)\big] \\ &= \mathsf{E}\big[h'(\Phi)\psi'_{\theta^*}(\Phi)\big] - \mathsf{E}\big[h'_{\theta^*}(\Phi)\psi'_{\theta^*}(\Phi)\big] \end{aligned} \tag{2-61}$$

Using Lemma 3, we can write an alternate representation for $\mathsf{E}\big[h'(\Phi)\psi'_\theta(\Phi)\big]$ as below:

$$\mathsf{E}\big[h'(\Phi)\psi'_\theta(\Phi)\big] = \langle R_{U''}c', \psi'_\theta\rangle_{L^2} = \langle c', R^\dagger_{U''}\psi'_\theta\rangle_{L^2}$$

Obtaining solutions to equations of the form (2-61) fall within the framework of stochastic approximation techniques. Recursive estimates for the optimizer θ^* can be obtained by using a gradient descent like stochastic approximation algorithm:

$$\begin{aligned} d_t &= \varphi_{\psi_{\theta_t}}(t)c'(\Phi_t) - h'_{\theta_t}(\Phi_t)\psi'_{\theta_t}(\Phi_t), \\ \theta_t &= \theta_{t-1} - \alpha_t M_t^{-1} d_t. \end{aligned} \tag{2-62}$$

Here, $\{\alpha_t\}$ is a positive gain sequence, subject to the conditions,

Question:Better notation for $\varphi_{\psi_{\theta_t}}$?

$$\sum_t \alpha_t = \infty, \qquad \sum_t \alpha_t^2 < \infty. \tag{2-63}$$

A common choice is $\alpha_t = 1/(t+1)$. By using a matrix gain M_t, inspired by Newton-Raphson methods, convergence of the algorithm can be improved. The optimal choice of the matrix gain M^* is,

$$\begin{aligned} M^* &:= -\nabla^2_\theta \mathcal{E}|_{\theta=\theta^*} \\ &\approx -\langle \psi'_{\theta^*}, \psi'^{T}_{\theta^*}\rangle_{L^2}, \end{aligned} \tag{2-64}$$

and approximated as,

Question:is the sign and notation for M correct? Can you also check the comments made?

$$M_T := -\frac{1}{T}\int_0^T \psi'_{\theta_t}(\Phi_t)\psi'^{T}_{\theta_t}(\Phi_t)dt \tag{2-65}$$

More recently, it has been shown that a better estimate of M^* can be obtained via a two-time-scale stochastic approximation algorithm. To keep the discussion simple, such algorithms are skipped here. Interested readers are referred to the work by Devraj et al. [44].

2.4 Summary and Conclusions

Both variants of the ∇-LSTD algorithm presented in Sections 2.3.1 and 2.3.3 approximate the gradient of the solution to Poisson's equation directly. The ∇-LSTD-L algorithm in Section 2.3.1 provides a greatly simplified representation for b compared to the ∇-LSTD algorithm. The major improvement is that as the eligibility vector φ does not appear in the

algorithm, simulating the SDE associated to the diffusion can be avoided. This enables us to relax the assumption that the density ρ is known at least in its unnormalized form or that the samples Φ_t are even generated by a Langevin diffusion. The sampling methodology is insignificant as long as the samples are distributed according to ρ. This fact is very useful in constructing "plug and play" algorithms for FPF gain function approximation and asymptotic variance reduction. The ∇-LSTD-L algorithm works with only the particle locations and the forcing function c evaluated at these locations as its inputs. In practice, it has been observed that the variance of the parameter estimates obtained using the ∇-LSTD-L algorithm is lower than those obtained from the ∇-LSTD algorithm. This is illustrated in Chapter 4 while presenting numerical examples. The main limitation of this method is that Prop. 2.1 holds only for the Langevin diffusion and hence, this algorithm cannot be extended to a class of more general diffusions.

Technically, ∇-LSTD algorithm belongs to the category of approximate dynamic programming rather than reinforcement learning, as it assumes knowledge of the model for simulation. However, they are seen to achieve considerable variance reduction as compared to the standard LSTD algorithm. They also have a wider scope of application to problems in optimal control in comparison to the Langevin-specific variant.

To summarize, in this chapter, we introduced Poisson's equation and motivated why the solution to Poisson's equation is central to this dissertation. The main conclusions from this chapter are as follows:

(i) Poisson's equation is central to the ergodic theory of Markov chains and finds applications in average-cost optimal control, performance evaluation, variance reduction etc.

(ii) Langevin diffusion is a continuous-time diffusion process that forms an integral part of our two applications of interest - the feedback particle filter and Markov chain Monte Carlo algorithms.

(iii) Obtaining analytical solutions to Poisson's equation for the Langevin diffusion is difficult outside of special cases and hence, approximation algorithms are required.

(iv) The derivation of the standard LSTD learning algorithm to approximate the discounted cost value function is presented. Its limitations to address the average cost problem is discussed.

(v) To address the limitations of LSTD, a new class of algorithms called differential TD learning that directly approximates the gradient of the solution is introduced. For the special case of Poisson's equation associated to the Langevin diffusion, adjoint property of the differential generator allows a simple formulation that does not require a simulation of the model.

(vi) A more generic variant of the ∇-LSTD algorithm is presented for a general continuous-time diffusion. This is computationally expensive and assumes the knowledge of the model. Extensions of this algorithm to state spaces in higher dimensions is possible, but requires more work.

(vii) Numerical examples comparing the various algorithms in the context of the FPF and MCMC are presented in Chapters 4 and 5 respectively.

CHAPTER 3
REPRODUCING KERNEL HILBERT SPACES (RKHS) FOR DIFFERENTIAL TD LEARNING

Question:Can you read this section till the beginning of 3.1?

In Chapter 2, the standard LSTD algorithm was reviewed and two versions of the differential TD learning algorithm were presented. An important step in all these algorithms is the selection of a suitable approximating function class, characterized by a linear or nonlinear parameterization. It is noted in [13] that any insight about the true value function can aid in this step. For example, if the value function is known to obey certain properties, say convexity, then the basis functions chosen can also be convex. In certain examples, the value function at a state can provide useful information about the values at "neighboring" states. In the case of stochastic optimal control, the fluid value functions provide a natural starting point to approximate the solution to the average-cost value function. However, such choices are heavily problem-dependent and no standard methods exist. In most traditional examples of TD learning, a predetermined set of basis functions is used to define the function class. This limits the learning capability in two ways:

(i) The chosen function class may not be rich enough to give a good approximation of the value function.

(ii) The approximation algorithms discussed in this dissertation are based on Monte Carlo methods with finite sample size. A predetermined set of basis functions will not be able to exploit information about the distribution of these samples in the state space.

Hence, it is a better idea to choose a function class that adapts dynamically to suit the problem and the sample distribution. Kernel methods provide an alternative approach to basis selection.

This chapter aims to adapt the ∇-LSTD-L algorithm proposed in Chapter 2 in a reproducing kernel Hilbert space (RKHS) setting. The remainder of this chapter is organized as follows - In Section 3.1, a very concise introduction to kernel functions and the RKHS theory is provided. A more detailed discussion of its properties including Mercer's theorem can be found in Appendix A. Gaussian kernels have found wide acceptance in machine learning and function

approximation and some of the useful properties of the RKHS induced by the Gaussian kernel is discussed in Appendix B. Function approximation problems arising in machine learning are usually cast in the form of a regularized empirical risk minimization (ERM) problem - a brief introduction is provided in Section 3.2. Obtaining solutions to such problems are made possible via the classical representer theorem. The theorem is stated in 3.2.1 and the proof is provided in Appendix C. In Section 3.2, the minimum-norm problem in $L^2(\rho)$ (2-35) of approximating the gradient of the solution to Poisson's equation is reformulated as an equivalent ERM problem to apply the RKHS machinery. As the ERM formulated consists of gradient terms, a generalized version of the representer theorem is required to obtain the optimal solution. A simplified sub-optimal solution that lies on a subspace of the original RKHS is also presented. We provide a short review of an error analysis approach for a generic least squares regression problem in an RKHS in Appendix D. Error analysis for ERMs with loss functions having gradient terms is a topic for future research.

Question:is this statement about error analysis okay?

3.1 RKHS Basics

Kernel methods have been quite well-studied for over five decades now. They provide a flexible and mathematically-elegant framework for a range of estimation problems by relating them to a function reconstruction problem in a higher dimensional feature space. By controlling the features of this space, the properties of the candidate functions can also be implicitly controlled. The mathematical properties of kernel functions were first investigated by Moore [45] and Mercer [46] in the early twentieth century. Although initial applications were in time series analysis, detection, filtering and prediction problems, more recently they have been found to be incredibly useful in machine learning [47], especially after the advent of support vector machines (SVM) for classification and regression problems [48, 49]. In this dissertation, the goal is to employ kernel methods as an alternative to finite dimensional basis selection for the ∇-LSTD-L learning algorithm.

Anand:From Bhujwalla's thesis, need to rephrase

3.1.1 Reproducing Kernels

Before defining an RKHS, a positive definite kernel needs to be defined. A function $K : \mathsf{X} \times \mathsf{X} \to \mathbb{R}$ which for all $N \in \mathbb{N}$ and all $x^1, x^2, \cdots, x^N \in \mathsf{X}$ gives rise to a positive definite Gram matrix is a called a positive definite kernel. That is,

$$\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K(x^i, x^j) \geq 0, \qquad \forall \alpha_i \in \mathbb{R}, \forall x^i \in \mathsf{X} \tag{3-1}$$

Let us consider a function f which is a linear combination of the form,

$$f(\cdot) = \sum_{i=1}^{N} \alpha_i K(x^i, \cdot), \tag{3-2}$$

where $N \in \mathbb{N}, \alpha_i \in \mathbb{R}, \{x^i \in \mathsf{X} : 1 \leq i \leq N\}$ are arbitrary. Let $\mathcal{H}^0$ denote the vector space (pre-Hilbert space) spanned by all functions f of this form. Evidently, the function g defined as,

$$g(\cdot) = \sum_{j=1}^{M} \beta_j K(x^j, \cdot), \tag{3-3}$$

where each $\beta_i \in \mathbb{R}$ and $\{x^j\}_1^M$ are arbitrary points, also belongs to this vector space. An inner product of functions $f, g \in \mathcal{H}^0$ is defined as,

$$\langle f,\, g\rangle := \sum_{i=1}^{N}\sum_{j=1}^{M} \alpha_i \beta_j K(x^i, x^j). \tag{3-4}$$

It is interesting to note that the inner product is independent of the expansions used for f and g, i.e.

$$\langle f, g\rangle = \sum_{i=1}^{N} \alpha_i g(x^i) = \sum_{j=1}^{N} \beta_j f(x^j) \tag{3-5}$$

The reproducing property of the kernel K follows from the definition of the inner product (3-4),

$$\langle K(x, \cdot), f(\cdot)\rangle = f(x) \qquad \forall x \in \mathsf{X}, \forall f \in \mathcal{H}^0. \tag{3-6}$$

Owing to this property, the kernel function K is called the reproducing kernel. It is also called the evaluation functional as it evaluates the function f at a point x. The symmetry of the kernel K guarantees that the inner product definition satisfies the commutative property, i.e. if

$K_x = K(x, \cdot)$ we have,

$$\langle K_x, K_{x'} \rangle = K(x, x') = K(x', x) = \langle K_{x'}, K_x \rangle. \tag{3-7}$$

The space obtained by the completion of functions of the form f defined in (3-2), that are endowed with the inner product defined in (3-4) (denoted as $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ here onward to distinguish from the normal dot product definition) yields a Hilbert space $\mathcal{H}$, called the reproducing kernel Hilbert space (RKHS). The corresponding norm is defined as $\|f\|_{\mathcal{H}} := \sqrt{\langle f, f \rangle_{\mathcal{H}}}$.

The Moore-Aronszajn theorem states that every symmetric, positive definite kernel $K : \mathsf{X} \times \mathsf{X} \to \mathbb{R}$ defines an RKHS $\mathcal{H}$ for which K is the reproducing kernel. A Hilbert function space $\mathcal{H}$ that has a reproducing kernel K is always an RKHS and conversely, every RKHS has a (unique) reproducing kernel. While all RKHS are Hilbert spaces, the converse is not true. For example, the space of square-integrable L^2 functions is a Hilbert space, but not an RKHS.

The reproducing kernel Hilbert spaces have the remarkable property that norm convergence implies pointwise convergence. For any function $f \in \mathcal{H}$ and $\{f_n\} \subset \mathcal{H}$ be a sequence with $\|f_n - f\|_{\mathcal{H}} \to 0$ for $n \to \infty$, then for all $x \in \mathsf{X}$, we have,

$$\lim_{n \to \infty} f_n(x) = \lim_{n \to \infty} \langle K_x, f_n \rangle_{\mathcal{H}} = \langle K_x, f \rangle_{\mathcal{H}} = f(x) \tag{3-8}$$

If K is a continuous, symmetric and positive definite kernel satisfying,

$$\kappa = \sup_{x \in \mathsf{X}} \sqrt{K(x, x)} < \infty, \tag{3-9}$$

then, Prop. 3.1 holds. Additionally, if K is continuous, then every function in $\mathcal{H}$ is continuous.

Proposition 3.1. *If the kernel K is uniformly bounded by (3-9), then any $f \in \mathcal{H}$ is also bounded.*

Proof. If $f \in \mathcal{H}$, for all $x \in \mathsf{X}$,

$$\begin{aligned} |f(x)| &= |\langle K(x,\cdot), f\rangle_{\mathcal{H}}| \\ &\leq \|K(x,\cdot)\|_{\mathcal{H}}\|f\|_{\mathcal{H}}, \qquad \text{(By Cauchy-Schwarz inequality)} \\ &\leq \kappa\|f\|_{\mathcal{H}} \end{aligned} \tag{3-10}$$

As the above inequality holds for all $x \in \mathsf{X}$,

$$\|f\|_\infty := \max_{x\in\mathsf{X}} |f(x)| \leq \kappa\|f\|_{\mathcal{H}} \tag{3-11}$$

■

The RKHS also has a feature map representation. Let $\Phi : \mathsf{X} \to \mathbb{R}^{\mathsf{X}}$ be a feature map that maps each point from X into a function mapping X to $\mathbb{R}$, i.e. $\mathbb{R}^{\mathsf{X}} := \{f : \mathsf{X} \to \mathbb{R}\}$. The map Φ is non-unique, but a canonical feature map exists and is given by $\Phi(x) = K(x,\cdot)$,

$$\begin{aligned} K(x,x') &= \langle \Phi(x), \Phi(x')\rangle \\ &= \langle K_x, K_{x'}\rangle_{\mathcal{H}} \end{aligned} \tag{3-12}$$

3.1.2 Examples of Reproducing Kernel Functions

It is evident from the construction of the RKHS that the properties of the functions in $\mathcal{H}$ are governed by the properties of its kernel K. Therefore, by choosing different kernel functions, different function spaces can be generated. Typically, K depends on a variable hyperparameter, which can be tuned to control the scaling. Some commonly used kernel functions are:

1. Linear kernel: The simplest example is a linear kernel in one dimension:

$$K(x,x') := xx', \qquad x, x' \in \mathsf{X}. \tag{3-13}$$

2. Polynomial kernel: It has been shown by Aronszajn [23] that sums and products of kernel functions are also kernels. Polynomial kernels are obtained by taking combinations of linear

kernels:

$$K(x, x') = (xx' + a)^b, \qquad x, x' \in \mathsf{X}, a \in \mathbb{R}, b \in \mathbb{N}. \tag{3-14}$$

3. Radial basis function kernels: Both linear and polynomial kernels depend on the absolute location of the inputs. Radial basis functions are a class of kernels that are translation-invariant, i.e. their values only depend on the relative positions of the inputs, and not the absolute values themselves. They are essentially a measure of similarity, in the sense that the proximity of the points in the input space determines the value of the kernel. A class of radial basis function kernels may be defined as:

$$K(x, x') := g(\gamma_1, \|x - x'\|^{\gamma_2}) \tag{3-15}$$

The most popular example of a radial basis kernel is the exponential kernel defined as $K(x, x') := \exp(-\gamma_1 \|x - x'\|^{\gamma_2})$, of which the standard Gaussian kernel is a special case:

$$K_\varepsilon := \exp\Big\{-\frac{\|x - x'\|^2}{4\varepsilon}\Big\}, \tag{3-16}$$

where ε is the variance parameter that controls the width of the function. The Gaussian kernel enjoys a number of desirable properties that makes it widely used. It is also the choice of kernel in this dissertation. The properties have been investigated in [50, 51] and a brief review is provided in Appendix B.

More composite kernel functions can be constructed by taking various combinations of reproducing kernels.

3.2 Empirical Risk Minimization (ERM)

In this dissertation, kernel methods and RKHS are employed with the ultimate objective of acting as an approximating function space to be used in the ∇-LSTD learning algorithms presented in Chapter 2. As mentioned earlier, RKHS has a rich history of being used for learning functions from observed data. Such class of problems termed empirical risk minimization (ERM) have been well studied in statistical learning theory literature. In order to utilize the RKHS theory, the $L^2(\rho)$ norm-minimization problem in (2-35) needs to be fit in

an ERM framework. In this section, a primer on ERM is provided, followed by the necessary transformations that lead to an ERM formulation for the ∇-LSTD learning problem.

In an ideal setting, the goal of a learning problem is to find an approximation of a function $f_\rho : \mathsf{X} \to Y$, when only a pair of values $\boldsymbol{z} = (x^i, y^i)_{i=1}^N$ drawn from an unknown probability measure ρ on $\mathsf{X} \times Y$ is available. Usually, X is a compact domain and $Y = \mathbb{R}$. For a least-squares regression problem, the best target function often called the regression function is the minimizer of:

$$f_\rho := \arg\min_f \int_{\mathsf{X}\times Y} (f(x) - y)^2 d\rho, \tag{3-17}$$

it takes the form:

$$f_\rho(x) = \int_Y y d\rho(y|x). \tag{3-18}$$

In practice, it is not possible to obtain f_ρ due to two reasons. First, ρ is unknown and only a finite number of sample points are available for estimation and therefore, it is not possible to carry out the integration (3-18). Secondly, since the approximating function space that is chosen is typically a small subspace, an exact match for f_ρ may not be obtained within this space. Hence, we focus on minimizing the empirical risk on observed data:

$$f_{\boldsymbol{z}} := \arg\min_f \frac{1}{N} \sum_{i=1}^{N} (y^i - f(x^i))^2. \tag{3-19}$$

Assuming, N is large enough and $\boldsymbol{z} \sim \rho$, (3-19) can be thought of as approximating (3-17) quite well. In a non-parametric setting, this problem is ill-posed and can have infinitely many solutions. This is resolved by adding an extra regularization term to the objective function. A common approach followed is to minimize,

$$f_{\boldsymbol{z},\lambda} := \arg\min_f \frac{1}{N} \sum_{i=1}^{N} (y^i - f(x^i))^2 + \lambda \|Af\|^2_{\mathcal{L}^2_\rho(\mathsf{X})}, \tag{3-20}$$

where A is an operator and $\mathcal{L}^2_\rho(\mathsf{X})$ is the Hilbert space of square integrable functions on X with measure ρ_{X} on X and $\lambda > 0$ is the regularization parameter. The key objective in an ERM problem is not just to obtain a good fit on the observed data, but to generalize well on

"unseen" data as well. If the standard L^2 norm is used for regularization, (3-20) is called the ridge regression. To obtain sparse solutions, L^1-regularization is used, which forms the LASSO technique. Regularization is used to prevent overfitting and achieve better generalization. In this dissertation, we consider a Tikhonov regularization scheme [52] associated with Mercer kernels.

Anand:ridge and LASSO needs citations

3.2.1 ERM in an RKHS Setting

Now that the utility of regularized empirical risk minimization has been motivated, we consider a more general formulation of an ERM in an RKHS setting. A loss function $L\colon \mathsf{X} \times \mathbb{R} \to \mathbb{R}$ is given. It is not necessary that the loss function takes the mean-squared error form in (3-20). Given an RKHS $\mathcal{H}$, the goal is to find the function $f_\lambda^* \in \mathcal{H}$ that solves the infinite-dimensional regularized optimization problem:

$$f_\lambda^* := \arg\min_{f \in \mathcal{H}} \underbrace{\frac{1}{N}\sum_{i=1}^{N} L(y^i, f(x^i))}_{\text{Empirical risk}} + \underbrace{\Omega(\|f\|_{\mathcal{H}})}_{\text{Regularization}} . \tag{3-21}$$

The objective function in (3-21) consists of an empirical risk term, that minimizes the error on observed data, and a regularization term that controls the properties of the model. It is important to note that the regularization term is independent of the observed data sequence $\mathbf{z}$. The function $\Omega : [0, \infty] \to \mathbb{R}$ is strictly monotonically increasing. Often, Ω is chosen to be proportional to a quadratic function of the norm $\|\cdot\|_{\mathcal{H}}$, such as $\lambda\|f\|_{\mathcal{H}}^2$. The value of the regularization parameter λ controls the trade-off between the empirical error term and the smoothness of the estimated function. In other words, a larger value for λ reduces the variance of the estimates by reducing its dependency on $\mathbf{z}$, but at the cost of introducing bias to the model. Regularization is necessary to ensure the well-posedness of the problem and increasing λ improves the numerical stability of the algorithm.

Anand:A few words about ways to choose the best λ.

The remarkable theme in the RKHS literature is that while the optimization problem in (3-21) is infinite dimensional, the optimizer lies in an identifiable finite-dimensional subspace. The result is due to the classical version of the representer theorem of Kimeldorf and Wahba

[24]. The original form of the theorem considered the mean-squared loss $L(y^i, f(x^i)) = (y^i - f(x^i))^2$. An extension to non-quadratic loss functions was provided in [53]. The classical version of the representer theorem is stated next and its proof presented in [25] is provided in Appendix C.

Theorem 3.1 (Representer Theorem)**.** *The minimizer $f^*_\lambda \in \mathcal{H}$ of the regularized risk (3-21) for an arbitrary loss function $L(y^i, f(x^i))$ lies in the span of kernel functions centered at the sample points, $K(x^i, \cdot)$,*

$$f^*_\lambda(x) = \sum_{i=1}^{N} \beta_i K(x^i, x)$$

■

Schölkopf et al. in [25] present two more generalized versions of the representer theorem, called the semiparametric representer theorem and biased regularization. Semiparametric representer theorem is particularly useful as it allows a combination of finitely parameterized family of functions and an RKHS to be used as the approximating function space. The proof for this extension is straight forward.

Theorem 3.2 (Semiparametric Representer Theorem)**.** *In addition to assumptions in Theorem 3.1, if a set of M real-valued functions $\{\psi_j\}_{j=1}^M : \mathsf{X} \to \mathbb{R}$ is given, with the property that they are linearly independent, then any $\tilde{f} := f + g$, where $f \in \mathcal{H}$ and $g \in$ span $\{\psi_j\}$, then the minimizer of the regularized risk functional,*

$$\tilde{f}^*_\lambda(x) := \arg\min_{\tilde{f}} \frac{1}{N} \sum_{i=1}^{N} L(y^i, \tilde{f}(x^i)) + \Omega(\|f\|_{\mathcal{H}})$$

admits a representation of the form

$$\tilde{f}(x) = \sum_{i=1}^{N} \beta_i K(x^i, x) + \sum_{j=1}^{M} \gamma_j \psi_j(x)$$

with $\gamma_j \in \mathbb{R}$ for all $j \leq M$.

Anand:Read spline smoothing problem in Bhujwalla's thesis and see if this can be applied for gradient regularization.

3.3 Kernel Methods for Differential TD-Learning

In this Section, the goal is to construct an ERM problem in an RKHS setting, such that it closely approximates the objective function in the ∇-LSTD learning algorithm (**??**). The

loss function L will be chosen so that the optimizing function approximately solves (**??**); the infinite dimensional variant of (**??**).

$$\|\nabla h - \nabla g\|_{L^2}^2 := \sum_{k=1}^{d} \Big\| \frac{\partial}{\partial x_k^i}(h-g) \Big\|_{L^2}^2 . \tag{3-22}$$

The fact that the original objective function aims to minimize the approximation error in $L^2(\rho)$ norm helps in the construction of an equivalent ERM.

Candidates for the loss function L are discussed here: If the values $\{x^i\} \subset \mathsf{X}$ are sampled randomly and independently according to ρ, then by the law of large numbers, for large N,

$$\frac{1}{N}\sum_{i=1}^{N} L(x^i, g(x^i), \nabla g\,(x^i)) \approx \int L(x, g(x), \nabla g\,(x))\rho(x)\,\mathsf{d}x. \tag{3-23}$$

The independence assumption is not essential; the samples $\{x^i\}$ may very well be generated by an ergodic Markov chain with ρ as its invariant density.

In view of (3-22), the ideal loss function is thus

$$L^*(x, g(x), \nabla g\,(x)) = \|\nabla h\,(x) - \nabla g\,(x)\|^2\,, \quad x \in \mathsf{X}\,. \tag{3-24}$$

The difficulty with this definition is that L^* in its current form is not computable as ∇h is unknown. This is resolved by applying the special structure of the differential generator $\mathcal{D}$, stated in Prop. 2.1:

Proposition 3.2. *If h is a solution to Poisson's equation satisfying $h \in C^2 \cap L^2$, then*

$$\|\nabla h - \nabla g\|_{L^2}^2 = \|\nabla h\|_{L^2}^2 + \int L^\bullet(x, g(x), \nabla g\,(x))\rho(x)\,\mathsf{d}x$$

where $L^\bullet(x, g(x), \nabla g\,(x)) = \|\nabla g(x)\|^2 - 2\tilde{c}(x)g(x)$. ■

Proof of Prop. 3.2

Expanding L^* gives,

$$\begin{aligned} L^*(x, g(x), \nabla g(x)) &= \|\nabla h(x) - \nabla g(x)\|^2 \\ &= \|\nabla h(x)\|^2 + \|\nabla g(x)\|^2 - 2\nabla h(x) \cdot \nabla g(x) \end{aligned}$$

To establish the proposition, it remains to show that,

$$\int \nabla h(x) \cdot \nabla g(x) \rho(x) \mathsf{d}x = \int \tilde{c}(x) g(x) \rho(x) \mathsf{d}x.$$

This follows from an application of Prop. 2.1:

$$\begin{aligned} \langle \nabla h, \nabla g \rangle_{L^2} &:= \int \nabla h(x) \cdot \nabla g(x) \rho(x) dx \\ &= -\int (\mathcal{D}h(x)) g(x) \rho(x) dx \\ &= \int \tilde{c}(x) g(x) \rho(x) dx \qquad \text{(from (??))} \end{aligned}$$

■

Prop. 3.2 motivates the ERM introduced in this dissertation:

$$g^* := \arg\min_{g \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \Big[\underbrace{\|\nabla g(x^i)\|^2 - 2\tilde{c}_N(x^i) g(x^i)}_{L^\bullet(x^i, g(x^i), \nabla g(x^i))} \Big] + \lambda \|g\|_{\mathcal{H}}^2 \tag{3-25}$$

in which the centered function $\tilde{c}$ is also approximated:

$$\tilde{c}_N(x) = c(x) - \frac{1}{N} \sum_{i=1}^{N} c(x^i)\,, \quad x \in \mathsf{X}\,.$$

The term $\|\nabla h\|^2$ can be conveniently dropped from (3-25) as it does not affect the optimization problem.

3.3.1 Extended Representer Theorem

It may be seen that the ERM in (3-25) falls in the standard form in (3-21) except that the loss function $L^\bullet$ depends on the gradient of the estimator g. Problems of this form where learning with gradients is required The classical version of the representer theorem (Theorem 3.1) is valid under the assumption that the loss function L depends only on g. In this dissertation, as we are interested in loss functions of the form

$$\min_{g \in \mathcal{H}} \Big\{ \frac{1}{N} \sum_{i=1}^{N} L(x^i, g(x^i), \nabla g\,(x^i)) + \lambda \|g\|_{\mathcal{H}}^2 \Big\}, \tag{3-26}$$

where the function L includes ∇g, it cannot be applied directly. The extension to include differential loss (including gradient terms) appeared in [54]. The main motivation in this work is to extend the RKHS theory to applications that may have gradient data or unlabeled data available for improving learning ability. Two types of problems are presented - one in which the loss function includes a term with gradients of the estimator function evaluated at the unlabeled points, called semi-supervised learning and a second where, gradient values at sample points are available and the algorithm is required to learn the function values and the gradients simultaneously, called the Hermite learning algorithm. Potential applications discussed are in image processing, where gradient fitting is used to preserve edges and discontinuities and avoiding staircasing effects [55].

Similar to the standard representer theorem, that guarantees a finite dimensional solution to a potentially infinite dimensional ERM problem, an extended version is presented in [54] that can be applied to cases where the loss functions include partial derivatives of the estimator. This version is stated here followed by its proof.

Theorem 3.3 (Extended Representer Theorem)**.** *Suppose that $L(x, g(x), \nabla g)$ is a convex function on $\mathbb{R}^{d+1}$ for each $x \in \mathsf{X}$. Then the optimizer g^* of (3-26) over $g \in \mathcal{H}$ exists, is unique, and has the following representation:*

Anand:change x_k^i to x^k

$$g^*(x) = \sum_{i=1}^{N} \Big[\beta_i^{0*} K(x^i, x) + \sum_{k=1}^{d} \beta_i^{k*} \frac{\partial K}{\partial x_k}(x^i, x) \Big], \tag{3-27}$$

where $\{\beta_i^{k*} : i = 1, \cdots, N,\ k = 0, \cdots, d\}$ are real numbers. ■

Proof. The proof provided here is a special case of the proof for a more general version of the extended representer theorem appearing in [54]. The partial derivative reproducing property of the kernel K stated and proved in [54, Theorem 1], presented as Lemma 4 is used in the proof.

Lemma 4. *If $K : \mathsf{X} \times \mathsf{X} \to \mathbb{R}$ is a Mercer kernel such that $K \in C^2$, then the following statements hold:*

(i) *For any $x \in \mathsf{X}$, $\frac{\partial K}{\partial x_k} \in \mathcal{H}$.*

(ii) *A partial derivative reproducing property holds:*

$$\frac{\partial f(x)}{\partial x_k} = \left\langle \frac{\partial K(x,\cdot)}{\partial x_k}, f(\cdot) \right\rangle_{\mathcal{H}} \qquad \forall x \in \mathsf{X}, \forall f \in \mathcal{H}. \tag{3-28}$$

By Lemma 4, $\frac{\partial K}{\partial x_k}(x^i, .) \in \mathcal{H}$ for each k. Therefore, any f of the form:

$$f := \sum_{i=1}^{N} \Big[\beta_i^0 K(x^i, .) + \sum_{k=1}^{d} \beta_i^k \frac{\partial K}{\partial x_k}(x^i, .) \Big],$$

is in $\mathcal{H}$. The rest of the proof proceeds using arguments similar to those in the proof of Theorem 3.1.

Let us denote by $\mathcal{H}_{\|}$, the linear subspace of $\mathcal{H}$ made up of all such functions f,

$$\mathcal{H}_{\|} := \Big\{ f \in \mathcal{H} \,|\, f = \sum_{i=1}^{N} \Big[\beta_i^0 K(x^i, .) + \sum_{k=1}^{d} \beta_i^k \frac{\partial K}{\partial x_k}(x^i, .) \Big] \Big\},$$

where N ranges over $\mathbb{Z}_+$, and each $\beta_i^k \in \mathbb{R}$ for each i and $k = 0$ to d. Let $\mathcal{H}_{\perp}$ be the subspace of $\mathcal{H}$ orthogonal to $\mathcal{H}_{\|}$:

$$\mathcal{H}_{\perp} := \{ \tilde{f} \in \mathcal{H} \,|\, \langle \tilde{f}, f \rangle_{\mathcal{H}} = 0, \forall f \in \mathcal{H}_{\|} \}$$

We can see that every $g \in \mathcal{H}$ can be uniquely decomposed into a component lying within $\mathcal{H}_{\|}$, denoted by $g_{\|}$ and a component lying within $\mathcal{H}_{\perp}$, denoted by $g_{\perp}$. The $\mathcal{H}$-norm can be written as,

$$\|g\|_{\mathcal{H}}^2 = \|g_{\|} + g_{\perp}\|_{\mathcal{H}}^2 = \|g_{\|}\|_{\mathcal{H}}^2 + \|g_{\perp}\|_{\mathcal{H}}^2 \geq \|g_{\|}\|^2 \tag{3-29}$$

This again implies that the regularization term in (3-26) is minimized if g lies in the subspace $\mathcal{H}_{\|}$. Furthermore, using the reproducing property of the kernel K,

$$\begin{aligned} g(x^i) &= \langle g \,, K(x^i, .) \rangle_{\mathcal{H}} \\ &= \langle g_{\|} \,, K(x^i, .) \rangle_{\mathcal{H}} + \langle g_{\perp} \,, K(x^i, .) \rangle_{\mathcal{H}} \\ &= \langle g_{\|} \,, K(x^i, .) \rangle_{\mathcal{H}} \\ &= g_{\|}(x^i), \end{aligned} \tag{3-30}$$

and partial derivative reproducing property, we have for all $k = 1, \cdots, d$,

$$\begin{aligned}\frac{\partial g(x^i)}{\partial x_k^i} &= \left\langle g\,, \frac{\partial K}{\partial x_k^i}(x^i, .)\right\rangle_{\mathcal{H}} \\ &= \left\langle g_{\|}\,, \frac{\partial K}{\partial x_k}(x^i, .)\right\rangle_{\mathcal{H}} + \left\langle g_{\perp}\,, \frac{\partial K}{\partial x_k}(x^i, .)\right\rangle_{\mathcal{H}} \\ &= \langle g_{\|}\,, \frac{\partial K}{\partial x_k}(x^i, .)\rangle_{\mathcal{H}} \\ &= \frac{\partial g_{\|}(x^i)}{\partial x_k}.\end{aligned} \tag{3-31}$$

From equations (3-30, 3-31), it is clear that the empirical error term depends only on the component $g_{\|}$,

$$L(x^i, g(x^i), \nabla g(x^i)) = L(x^i, g_{\|}(x^i), \nabla g_{\|}(x^i)),$$

and from (3-29), $g_{\|}$ minimizes the regularization term. Hence, the regularized objective function is minimized if g^* lies within $\mathcal{H}_{\|}$ and takes the form,

$$g^*(y) = \sum_{i=1}^{N}\Big[\beta_i^{0*} K(x^i, y) + \sum_{k=1}^{d} \beta_i^{k*} \frac{\partial}{\partial x_k} K(x^i, y)\Big]$$

■

3.3.2 Optimal ERM Solution (∇-LSTD-RKHS-Opt)

We first provide a full description of the solution to (3-25) for the one-dimensional model in this section. The optimizer is obtained as the solution to a system of linear equations, extensions to higher dimensions is straight-forward and provided towards the end of the section. The regularized ERM problem (3-25) in one-dimension reduces to the following:

$$g^* := \arg\min_{g \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \Big\{ (g'(x^i))^2 - 2\tilde{c}_N(x^i) g(x^i) \Big\} + \lambda \|g\|_{\mathcal{H}}^2, \tag{3-32}$$

where $g' = dg/dx$. Using the extended form of the Representer Theorem (Theorem 3.3), the optimizer g^* has the following form:

$$g_\beta(y) = \sum_{i=1}^{N}\Big[\beta_i^0 K(x^i, y) + \beta_i^1 \frac{\partial K}{\partial x}(x^i, y)\Big], \qquad y \in \mathsf{X} \tag{3-33}$$

Theorem 3.3 states that the solution to the infinite dimensional optimization problem reduces to a quadratic program over $\mathbb{R}^{2N}$ when there are N samples. A formula for the optimizer $\beta^* \in \mathbb{R}^{2N}$ is obtained as follows. Introduce vectorized notation:

$$\boldsymbol{\beta}^\intercal := [\beta_1^0, \ldots, \beta_N^0\,,\, \beta_1^1, \ldots, \beta_N^1], \quad \boldsymbol{\zeta}^\intercal := [\tilde{c}_N(x^1), \ldots, \tilde{c}_N(x^N)],$$

and introduce matrices $M_{00}, M_{10}, M_{01}, M_{11}$, whose $\{i,j\}^{th}$ entries are defined as follows:

$$M_{00}(i,j) := K(x^i, x^j) \qquad M_{10}(i,j) := \frac{\partial K}{\partial x}(x^i, x^j)$$
$$M_{01}(i,j) := \frac{\partial K}{\partial y}(x^i, x^j) \qquad M_{11}(i,j) := \frac{\partial^2 K}{\partial x \partial y}(x^i, x^j).$$

Here, $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ refer to the partial derivatives with respect to the first variable and second variable of $K(x,y)$ respectively.

Proposition 3.3. *The optimal parameter vector β^* is,*

$$\beta^* = M^{-1} b \tag{3-34}$$

$$\textit{where,} \quad M := \frac{1}{N} \left[\begin{array}{c} M_{01} \\ \hline M_{11} \end{array}\right] [M_{10} \,|M_{11}] + \lambda \left[\begin{array}{c|c} M_{00} & M_{01} \\ \hline M_{10} & M_{11} \end{array}\right]$$
$$b := \frac{1}{N} \left[\begin{array}{c} M_{00} \\ \hline M_{10} \end{array}\right] \boldsymbol{\zeta}$$

■

The derivation of the optimal parameter β^* is analogous to the derivation of the optimal θ^* in the finite basis case in Lemma 2. The proof of Prop. 3.3 follows from elementary calculus.

Following the above derivation, for an arbitrary dimension d, using the following matrix notation with $1 \leq k, l \leq d$ and $1 \leq i, j \leq N$:

$$M_{m0}(i,j) := \frac{\partial K}{\partial x_k}(x^i, x^j), \quad M_{0n}(i,j) := \frac{\partial K}{\partial y_l}(x^i, x^j), \quad M_{mn}(i,j) := \frac{\partial^2 K}{\partial x_k \partial y_l}(x^i, x^j).$$

$$\text{where,} \quad M := \frac{1}{N} \begin{bmatrix} M_{01} \\ \vdots \\ M_{d1} \end{bmatrix} [M_{10} \ \dots M_{1d}] + \lambda \begin{bmatrix} M_{00} & \dots & M_{0d} \\ \vdots & \vdots & \vdots \\ M_{d0} & \dots & M_{dd} \end{bmatrix}$$
$$b := \frac{1}{N} \begin{bmatrix} M_{00} \\ \vdots \\ M_{d0} \end{bmatrix} \boldsymbol{\zeta} \tag{3-35}$$

Each $M_{kl} \in \mathbb{R}^{N \times N}$ and therefore M is a $(d+1)N \times (d+1)N$ matrix and b is a $(d+1)N$ vector. The optimal parameter vector $\beta^* := [(\beta_i^0)_{i=1}^N, (\beta_i^1)_{i=1}^N, \dots, (\beta_i^d)_{i=1}^N] \in \mathbb{R}^{(d+1)N}$ is in the same form as (3-34). Although, β^* is easily computable, the solution scales poorly with dimensions. Moreover, there are more parameters $(d+1)N$ than there are sample points N, which is usually not preferred in machine learning. This could potentially lead to overfitting and other numerical issues. Ideally, we prefer an algorithm that scales well with dimensions. This leads to the following two extensions:

- A reduced complexity solution (3-36).
- A differential regularizer (3-38).

3.3.3 Reduced Complexity Solution (∇-LSTD-RKHS-Simple)

Theorem 3.3 states that the optimizer of the ERM with gradient terms in the loss function lies in a subspace of the RKHS $\mathcal{H}$ spanned by the kernel functions (K_{x^i}) and its partial derivatives $(\{\partial_k K_{x^i}\}_{k=1}^d)$ centered at the sample points. The main source of complexity is the presence of the $d \times N$ kernel partial derivative terms of the solution . To obtain a reduced complexity solution, g is restricted to a subspace of the following form:

$$g(y) := \sum_{i=1}^{N} \beta_i K(x^i, y). \tag{3-36}$$

Proposition 3.4. *The parameter vector $\beta^\circ \in \mathbb{R}^N$ that minimizes (3-32) over all functions g of the form (3-36) is given by,*

$$\beta^\circ := M^{\circ -1} b^\circ \,, \tag{3-37}$$

where $M^\circ := N^{-1}M_{01}M_{10} + \lambda M_{00}$ *and* $b^\circ := N^{-1}M_{00}\,\boldsymbol{\zeta}$. ■

Solutions of the form (3-36) are the optimal solution given by the standard representer theorem if the loss function did not have gradient terms. Although, (3-37) is clearly a sub-optimal solution to (3-32), β° is computationally simpler to obtain than β^* as there are only N parameters to estimate. In a surprising observation, it was found that their performance is nearly the same in numerical examples using Gaussian kernels for $d \leq 5$. More details are contained in Section **??**.

3.3.4 Differential Regularizer Formulation

Anand:need to check if it is possible

An alternate approach is to add a regularizing term that penalizes the $\mathcal{H}$-norm of the gradient ∇g rather than g itself. The modified objective for the one-dimensional model becomes,

$$\tilde{g} := \arg\min_{g\in\mathcal{H}} \frac{1}{N}\sum_{i}^{N}\Big\{(g'(x^i))^2 - 2\tilde{c}_N(x^i)g(x^i)\Big\} + \lambda\|g'\|_{\mathcal{H}}^2 \tag{3-38}$$

This suits our objective better as h' is the function for which we seek an approximation. Bhujwalla et al. in [56] have argued that a differential regularizer term adds more flexibility to the hyperparameter selection for the kernel function without compromising the smoothness of the approximation.

Unfortunately the representer theorem is not easily adapted to obtain a finite dimensional solution to the optimization problem (3-38), even in the differential generalization of [54]. The method proposed in [56] is an "extended representer" through the addition of an arbitrary number of kernel functions centered at newly introduced samples.

In our case, to obtain a solution we return to the original loss function (3-24), which in this one dimensional setting becomes $L^*(x, g, g') = [h'(x) - g'(x)]^2$. The associated ERM is then

$$g^* := \arg\min_{g\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N}[h'(x^i) - g'(x^i)]^2 + \lambda\|g'\|_{\mathcal{H}}^2\,. \tag{3-39}$$

Since this does not depend upon g, we can apply the classical representer theorem to conclude that there are scalars $\{\beta_i^*\}$ such that the optimal solution has the form $g^{*\prime} = g_{\beta^*}$, where

$$g^{*\prime} = \sum_{i=1}^{N} \beta_i^* K(x^i, y)\,, \qquad y \in \mathsf{X}\,.$$

We cannot use the definition (3-39) to compute β^* since h is not known. However, following the Law of Large Numbers approximation arguments, we can obtain the value of β by substituting g_β of this form into the right hand side of this variant of (3-32):

Anand:Law of Large Numbers needs capitalization?

$$g^* := \arg\min_{g\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N}\Big\{(g'(x^i))^2 - 2\tilde{c}_N(x^i)g(x^i)\Big\} + \lambda\|g'\|_{\mathcal{H}}^2,$$

and then solving the quadratic optimization problem to obtain β^*.

This approach requires a kernel for which we can easily compute

$$K_-(x,y) = \int_{-\infty}^{y} K(x,r)\,\mathrm{d}r$$

so that

$$g_\beta = \sum_{i=1}^{N} \beta_i K_-(x^i, \cdot\,)$$

It is not clear if the optimal solution has this form for dimension greater than one – this is another topic of current research.

3.4 Algorithm Design and Error Analysis

After choosing an appropriate loss function (3-25) to suit our objective, the next choice in the algorithm is that of a suitable kernel function K. In the case of ∇-LSTD learning, desirable properties of the RKHS used as the approximating function space will help us make an informed choice. More discussion on choosing the kernel in the context of ∇-LSTD learning is reserved for Section **??**. An equally important step in the algorithm design is the choice of kernel hyperparameters and the regularization parameter λ. The importance of a good choice of λ is illustrated with examples in [47]. Methods such as ordinary cross-validation and generalized cross-validation are suggested. As mentioned before, a large value of λ introduces bias and a small value of λ contributes to the variance of the error. A general criteria chosen

Anand:section referencing required

to obtain the optimal λ^* is to minimize the sum of the bias and the variance terms. Also, it may be seen that a good choice of λ depends on the number of samples N. $\lambda(N)$ is chosen to be a decaying function of N such as $\propto \frac{1}{N}$ or $\frac{1}{\sqrt{N}}$.

It is of interest to analyze the quality of the estimates obtained. A large body of literature is available that tries to obtain the error bounds arising in ERM problems using RKHS. The analysis approaches can be largely classified as those based on i) complexity of the hypothesis space (like covering numbers [57–59], Vapnik-Chervonenkis dimension [60], Rademacher-complexity [61, 62] and ii) notions of algorithm stability [63, 64] etc. A majority of these works provides PAC (probably almost correct) style bounds on the error for finite N. Our interest is in obtaining asymptotic bounds to the expected approximation error. Additionally, there are no results to the best of our knowledge that provide meaningful bounds for ERMs with gradient terms in the loss function. Paper by Bousquet et al. [64] is of particular interest as it derives performance bounds for a wide variety of algorithms including regularized least-squares regression in an RKHS. A short desciption of this result is provided in Appendix D. Extensions of this result to analyze ∇-LSTD learning on RKHS is an area for future research.

3.5 Conclusions

At the end of Chapter 2, it was mentioned that one of the major difficulties in ∇-LSTD learning is the choice of a parameterized family. Different versions of the algorithm to accommodate linear and nonlinear parameterizations were presented. However, these methods were not conducive to scaling for higher dimensions. Also, they made inefficient use of the sample information in constructing an approximation function space. This chapter has addressed most of these shortcomings. The major contributions in this chapter can be summarized as follows:

(i) By choosing a reproducing kernel Hilbert space as the approximating function space, we have eliminated the need to carefully choose a finite set of basis functions. In addition to being easily scalable to problems in higher dimensions, they also effectively use the sample

distribution. Another advantage of using an RKHS is that the optimizer is searched from within a potentially richer infinite dimensional function space.

(ii) By combining the TD learning theory with RKHS theory, we are able to express the minimum norm objective function in ∇-LSTD learning as an empirical risk minimization problem (ERM). The optimal solution to this ERM is obtained via recent extensions of RKHS theory.

(iii) As the optimal solution does not scale well with dimensions, a reduced complexity solution is proposed. We also propose a differential regularizer approach for which theory is currently absent, opening avenues for future research.

(iv) We also discuss the problem of choosing the optimal regularization parameter λ. A short review of existing error analysis approaches is provided. Extensions to bound errors for loss functions with gradient terms is part of future research.

(v) In Chapters 4 and 5, applications of these algorithms are presented in the context of gain function approximation in the feedback particle filter and asymptotic variance reduction in MCMC algorithms.

CHAPTER 4
APPLICATIONS TO NONLINEAR FILTERING

Applications to nonlinear filtering was the main motivation behind the development of the ∇-LSTD learning algorithms in Chapters 2 and 3. In this chapter, the problem of nonlinear filtering and the associated theory is described in detail in Section 4.1. A brief survey on approximations to the nonlinear filter, including the extended Kalman Filter (EKF) and particle filters, is provided in Section 4.2. Feedback particle filter (FPF), which is the main focus of the dissertation is formally introduced in Section 4.3. A critical component of the FPF is the gain function, which is obtained as the gradient of the solution to Poisson's equation associated to the Langevin diffusion. A sketch of the derivation of the optimal gain function that guarantees asymptotic exactness of the FPF to the nonlinear filter is provided in Appendix **??**. Galerkin-based approximation methods and an algorithm based on approximating the Markov semigroup of the Langevin diffusion [17], which was developed in parallel research are presented in Sections 4.4.1 and 4.4.2 respectively. Due to the gradient representation of the gain, ∇-LSTD algorithms described in Chapter 2 and its RKHS based variants from Chapter 3 offer a natural solution to approximating it. Two enhancements to the RKHS based ∇-LSTD learning algorithm are proposed to improve their performance in the FPF in Section 4.5. Finally, Section 4.7 contains a number of numerical experiments that compare the performance of the various algorithms for gain approximation and for filtering problems.

Question:Should I skip this from the Appendix

4.1 Introduction to Nonlinear Filtering

A preliminary introduction to nonlinear filtering was provided in Chapter 1. A schematic diagram of a state estimator appears in Fig. 1-1. Our goal in this section is to make things more precise through a more formal description of the problem. First, we begin with a motivating application. Nonlinear filtering has its origins in tracking problems in satellite and aircraft navigation. The key goal of filtering is to obtain recursive estimates of the state of a stochastic dynamical system based on partial noisy observations. A typical example in a tracking application is the simultaneous estimation of position and velocity of a moving

target. Here, position and velocity of the target constitute the state of the system and the observations are modeled as nonlinear functions of the state made in the presence of noise. More recently, filtering has found applications in diverse areas such as machine learning [4], queueing networks, mathematical finance [5] and data assimilation problems for weather forecasting [6].

A filtering problem can be formulated in continuous or discrete-time and continuous or finite state-space depending on the application of interest. The simplest dynamical model for filtering in discrete-time and state space is the Hidden Markov Model (HMM). In this dissertation however, we are primarily interested in a filtering problem in continuous-time with states evolving on a Euclidean space. For simplicity, let us restrict ourselves to the scalar filtering problem:

$$\begin{aligned} \text{State model}: \quad \mathrm{d}X_t &= a(X_t)\mathrm{d}t + \sigma_B \mathrm{d}B_t, \\ \text{Observation model}: \quad \mathrm{d}Z_t &= c(X_t)\mathrm{d}t + \sigma_W \mathrm{d}W_t, \end{aligned} \tag{4-1}$$

where $\boldsymbol{X} = \{X_t\}$ is the scalar state process, $\boldsymbol{Z} = \{Z_t\}$ is the scalar observation process, a and c are C^1 functions, and $\boldsymbol{B} = \{B_t\}$, $\boldsymbol{W} = \{W_t\}$ are mutually independent standard Brownian motions. The state model describes the evolution of the hidden state of the system. The uncertainties in the state model and the external disturbances that affect the dynamics are modeled as the state noise term $\boldsymbol{B}$. Indirect observations, in the form of nonlinear functions of the state corrupted by noise $\boldsymbol{W}$ are available via the observation model. The goal of the filtering problem is to approximate the posterior density ρ_t^* of X_t, given the past observation history $\mathcal{Z}_t := \sigma(Z_s : s \leq t)$. For any measurable set $A \subset \mathbb{R}$, the posterior density ρ_t^* is defined as,

$$\int_A \rho_t^*(x)\mathrm{d}x := \mathsf{P}\{X_t \in A \mid \mathcal{Z}_t\}. \tag{4-2}$$

A huge body of literature has been devoted to the study of such problems. The theory of nonlinear filtering has been described in [7, 65]. A more accessible derivation of the nonlinear filter, from a change of measure standpoint is provided in [3]. In this section, without going into a lot of detail, we state some of the important results.

4.1.1 Zakai and Kushner-Stratonovich equations

Zakai's equation [66] and Kushner-Stratonovich (K-S) [67, 68] equations are the key results in this area. The Kushner-Stratonovich equation provides a stochastic PDE describing the evolution of ρ_t^*:

$$\mathrm{d}\rho_t^*(x) = \mathcal{D}^\dagger \rho_t^*(x)\mathrm{d}t + \frac{1}{\sigma_W^2}(c(x) - \hat{c}_t)(\mathrm{d}Z_t - \hat{c}_t\mathrm{d}t)\rho_t^*(x), \tag{4-3}$$

where $\mathcal{D}^\dagger \rho := -(d(\rho a)/dx) + (\sigma_B^2/2)(d^2\rho/dx^2)$ is the adjoint operator to the differential generator of the SDE describing the state model in (4-1) and $\hat{c}_t := \int c(x)\rho_t^*(x)dt$. For a nonlinear observation function $c(x)$, a moment closure problem arises in the K-S equation and hence, it cannot generally be reduced to stochastic PDEs so that they could be numerically integrated. In some cases, it is convenient to use the Zakai's equation, which is a linear stochastic PDE that describes the evolution of the unnormalized posterior density $\tilde{\rho}_t^*$:

$$\mathrm{d}\tilde{\rho}_t^*(x) = \mathcal{D}^\dagger \tilde{\rho}_t^*(x)\mathrm{d}t + \frac{1}{\sigma_W^2}c(x)\mathrm{d}Z_t\tilde{\rho}_t^*(x), \tag{4-4}$$

Question:need the definition of $\mathcal{D}^\dagger$. Is it the adjoint to the differential generator?

4.1.2 Kalman-Bucy Filter

Solution to the equations (4-3, 4-4) are in general, infinite dimensional. In the special case where the functions $a(x)$ and $c(x)$ are linear, i.e. $a(x) = Ax$ and $c(x) = Cx$ and the prior distribution ρ_0 is Gaussian, it is guaranteed that the posterior density ρ_t^* also remains Gaussian for all t. A closed-form solution can be obtained to the filtering problem in this case. The posterior density ρ_t^* is completely characterized by the conditional mean μ_t and the state covariance Σ_t. The optimal filter is given by the classical Kalman-Bucy filter [69], which is described by the set of equations (4-5, 4-6), for a scalar system:

$$\text{Conditional mean:} \quad \mathrm{d}\mu_t = A\mu_t\mathrm{d}t + \underbrace{\frac{\Sigma_t C}{\sigma_W^2}}_{\text{Kalman gain}}(\mathrm{d}Z_t - C\mu_t\mathrm{d}t), \tag{4-5}$$

$$\text{State covariance:} \quad \frac{\mathrm{d}}{\mathrm{d}t}\Sigma_t = 2A\Sigma_t + \sigma_B^2 - \frac{\Sigma_t^2 C^2}{\sigma_W^2}. \tag{4-6}$$

Here, the conditional mean μ_t evolves according to the SDE in (4-5) and (4-6) is an ODE called the continuous-time Riccati equation. The state covariance Σ_t evolves independent of the observations Z_t and the conditional mean μ_t and as a result, it can be solved offline. The term $\frac{\Sigma_t C}{\sigma_W^2}$ is called the Kalman gain, denoted as $\mathsf{K}_{\mathsf{kal}}$.

4.2 Approximations to the Nonlinear Filter

In a general nonlinear setting, excepting special cases like the Beneš filter [70], the optimal nonlinear filter cannot expressed in terms of a finite set of parameters. Equations (4-3, 4-4) show that the posterior density can be computed in a recursive fashion. This leads to the development of discretization schemes to approximate the nonlinear filter. However, numerical approximations to the solution of the K-S equation using an Euler-type discretization are not accurate or robust.

As noted in Section 1.1.1, typically a filtering problem can be separated into two steps - prediction step, where the posterior density ρ_t^* at time t is propagated according to the state model, without accounting for the current observations, and the correction step, where the estimates are corrected after receiving the latest observations. As the state dimension increases, numerical solutions to both these steps become prohibitively expensive and approximate solutions are sought. Approaches to approximating the prediction step and the correction step can be chosen independently and then combined. In a broad sense, two approaches can be taken, one where an exact solution is obtained under the assumptions of linearity and another, where a finite-dimensional approximation of the Kushner-Stratonovich equation is used. Budhiraja et al. in [8] provide a comprehensive survey of approximation techniques for nonlinear filtering, with a particular focus on particle filtering based algorithms. A tutorial on a number of variants of particle filtering methods, particularly in the discrete-time setting is given in [71].

4.2.1 Extended Kalman Filter

Extended Kalman filter (EKF) [72] is based on the principle of local linearization of the state and observation models around the mean μ_t. Consequently, the resulting posterior

densities are approximated as Gaussians. In the EKF, the conditional mean and state covariance estimates for the system in (4-1) are governed by the following equations:

$$\mathrm{d}\mu_t = a(\mu_t)\mathrm{d}t + \frac{\Sigma_t C(\mu_t)}{\Sigma_W^2}(\mathrm{d}Z_t - c(\mu_t)\mathrm{d}t), \tag{4-7}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\Sigma_t = 2A(\mu_t)\Sigma_t + \sigma_B^2 - \frac{\Sigma_t^2 C^2(\mu_t)}{\sigma_W^2}, \tag{4-8}$$

where $A = \frac{da}{dx}$ and $C = \frac{dc}{dx}$. In higher dimensional state spaces, A and C are the Jacobian matrices. These are fairly easy to implement for moderate state dimensions. The performance of the EKF deteriorates if the state and observation models deviate significantly from linearity or if the noise variances are high and as a result, they suffer from severe divergence and instability problems. Furthermore, the EKF fails to capture the multi-modal features of the posterior distribution.

4.2.2 Particle Filters

Particle filters are Monte Carlo based approximations to the nonlinear filter [73]. They belong to the second category mentioned, wherein without relying on the linearization of dynamics, the posterior is approximated using a finite number of samples called particles. They are based on the idea of sequential importance sampling (SIS). Although, the discrete-time variant of the filter is more common, the derivation of the continuous-time particle filter is provided in [3]. The basic idea is this: a large number of particles $\{X_0^i\}$ drawn independently from the same prior distribution ρ_0^* are propagated through time according to the state model (4-1). Each particle is associated with an importance weight. Initialized with equal weights, the weight corresponding to each particle is updated based on the observations as follows:

Anand:check this reference

$$\mathrm{d}w_t^i = w_t^i\left(\frac{1}{\sigma_W^2}\right)(c(X_t^i) - \bar{c}_t)(\mathrm{d}Z_t - \bar{c}_t\mathrm{d}t), \tag{4-9}$$

where $\bar{c}_t := \sum_{i=1}^N w_t^i c(X_t^i)$. The particle locations $\{X_t^i\}$ along with the importance weights $\{w_t^i\}$ are used to come up with an empirical estimate of the posterior distribution as:

$$\rho_t^*(x) \approx \rho_t^{(N)}(x) = \sum_{i=1}^N w_t^i \delta(x - X_t^i), \tag{4-10}$$

where δ is the Dirac-delta function.

Particle filters are easy to implement and are ideally suited for a parallel computing architecture. The accuracy of the estimates improves as N increases. They do not require linearization of the model or discretization of the filter SDEs and are often seen to outperform the EKF in highly nonlinear examples. However, they are known to suffer from particle degeneracy, where after a few iterations, only a handful of particles remain with significant weights, thus reducing the effective sample size. A proposed remedy is frequent resampling of particles when the effective particle size falls below a predetermined threshold. After the resampling step, the importance weights of the new particles are reinitialized to be equal. Certain resampling schemes may again result in loss of diversity or sample impoverishment due to the replication of the same particles. A host of resampling schemes aimed at resolving these issues is presented in [8, 71].

4.3 Feedback Particle Filter (FPF)

In this section, we introduce the feedback particle filter (FPF), which is the preferred approximation of the nonlinear filter in this dissertation. In Section 4.2, several approximations to the nonlinear filter, including the EKF and the particle filter were presented. In addition to providing an overview of such techniques, one of the purposes behind giving a detailed exposition was to enable us to compare and contrast the similarities (and dissimilarities) of these techniques with the FPF.

FPF was first introduced in [74] as an alternative approach to particle filtering, inspired by mean-field optimal control techniques. Along the lines of particle filtering, the FPF is constructed as a collection of N particles, each of which evolves according to a stochastic differential equation (SDE). The state evolution of the i^{th} particle at time t, denoted X_t^i mimics that of the state model:

$$\mathrm{d}X_t^i = \underbrace{a(X_t^i)\mathrm{d}t + \sigma_B \mathrm{d}B_t^i}_{\text{Prediction}} + \underbrace{\mathrm{d}U_t^i}_{\text{Correction}}, \tag{4-11}$$

in which each $\boldsymbol{B}^i$ is a standard Brownian motion. The initial conditions $\{X_0^i : 1 \leq i \leq N\}$ are assumed i.i.d., with common prior distribution ρ_0^*. The primitives $\boldsymbol{B}^i$ and $\{X_0^j : j \geq 1\}$ are mutually independent, and also independent of $(\boldsymbol{X}, \boldsymbol{Z})$. The particles are all coupled via the control input term U_t^i corresponding to the i^{th} particle. The conditional distribution of a particle X_t^i given $\mathcal{Z}_t$ is given by:

$$\int_A \rho_t(x)\mathrm{d}x = P\{X_t^i \in A \mid \mathcal{Z}_t\}, \tag{4-12}$$

and an empirical approximation of ρ_t is obtained as:

$$\rho_t^{(N)}(A) := \sum_{i=1}^{N} \mathbb{1}_A\{X_t^i \in A\}, \tag{4-13}$$

where $A \in \mathcal{B}$ (Borel measurable subsets of $\mathbb{R}$) measurable set in $\mathbb{R}$. The particles are conditionally independent given the observations, so that for large N, the approximation $\rho_t^{(N)} \sim \rho_t$ holds in a weak sense.

Question:This was ρ_t^* before, but I guess it should be this.

The main difference in (4-13) compared to the estimate in (4-10) is that all the particles are weighted equally in the FPF. Such unweighted approaches hold the promise of avoiding the resampling step and overcoming the curse of dimensionality that the standard particle filter suffers. Similar unweighted particle filter approaches were proposed in [75, 76]. Daum et al. discuss the limitations of the conventional particle filter and present a filtering scheme for the continuous-discrete time case, called the information flow filter. A detailed comparison of the FPF with the information flow filter is provided in [77]. Neural particle filter (NPF), another unweighted approach is presented in [78]. A more extensive list of filtering techniques based on interacting particle systems can be found in [42]. The FPF has found success in applications such as physical activity recognition [79, 80], satellite navigation [81, 82] etc.

It may also be noted that prediction and correction operations are performed in a single step in (4-11). The correction step is implemented via the control input term U_t^i. For each i,

the control input $\boldsymbol{U}^i$ is constructed so that for any t and any $A \in \mathcal{B}$,

$$\mathsf{P}\{X_t^i \in A \mid \mathcal{Z}_t\} = \mathsf{P}\{X_t \in A \mid \mathcal{Z}_t\} = \int_A \rho_t^*(x)\mathsf{d}x.$$

The problem of choosing the optimal U_t^i such that the posterior density ρ coincides with the true posterior ρ_t^* is cast as an optimal control problem in [1, 74]. The Kullback-Leibler divergence between the true posterior ρ^* and the FPF mean-field estimate ρ is used as the cost function. An explicit formula for U_t^i, derived in [1] and it is used to control the dynamics of the i^{th} particle:

$$\mathsf{d}X_t^i = a(X_t^i)\mathsf{d}t + \sigma_B \mathsf{d}B_t^i + \underbrace{\mathsf{K}_t(X_t^i)\mathsf{d}I_t^i + \Omega(X_t^i, t)\mathsf{d}t}_{\text{optimal control input } U_t^i}, \tag{4-14}$$

where $\Omega(x,t) := \frac{1}{2}\sigma_W^2 \mathsf{K}_t(x)\frac{\partial \mathsf{K}_t(x)}{\partial x}$ and $\{I_t^i : t \geq 0\}$ is analogous to the innovations process corresponding to the i^{th} particle,

$$\mathsf{d}I_t^i := \mathsf{d}Z_t - \frac{1}{2}(c(X_t^i) + \hat{c}_t)\mathsf{d}t, \tag{4-15}$$

with $\hat{c}_t := \mathsf{E}[c(X_t^i) \mid \mathcal{Z}_t] = \int c(x)\rho_t(x)\,dx$ and approximated using $\hat{c}_t^{(N)} := \frac{1}{N}\sum_{i=1}^N c(X_t^i)$. The filter is expressed in the Stratonovich form as:

$$\mathsf{d}X_t^i = a(X_t^i)\mathsf{d}t + \sigma_B \mathsf{d}B_t^i + \mathsf{K}_t(X_t^i) \circ \mathsf{d}I_t^i, \tag{4-16}$$

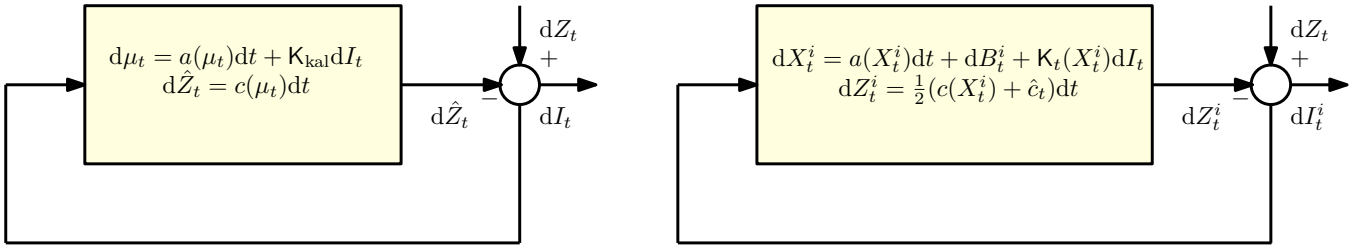
where, the symbol "$\circ$" indicates that the SDE is in Stratonovich form. The main subject of interest in this dissertation is the gain function K_t, that multiplies the innovations term I_t in (4-16). A detailed description of the gain function is provided next in Section 4.3.1.

4.3.1 FPF Gain Function

It is evident from (4-16), that the FPF exhibits a gain $\times$ innovation error structure that is a hallmark feature of the Kalman-Bucy filter. This feedback control structure was also present in the K-S equation (4-3) describing the nonlinear filter, but conspicuously absent in the standard formulation of the particle filter . The FPF restores this feedback structure, which contributes to the robustness of the filter via the self-correcting property. The block diagrams

Anand:bootstrap PF?

in Fig. 4-1 illustrate the similarity in structure of the FPF with the Kalman filter. However, unlike the Kalman-Bucy filter where the gain is a constant, the FPF gain function K_t depends on the state X_t. For a linear-Gaussian system, the optimal gain has been shown to coincide with the Kalman gain.



a) Kalman filter b) Feedback particle filter (FPF)

Figure 4-1. Schematic block diagrams comparing the Kalman filter and the feedback particle filter (FPF) [1]

To implement the FPF, the gain function K_t needs to be computed for each t. To simplify notation, consider a fixed time $t \geq 0$, and suppress dependency on t. In particular, let K denote the gain function that appears as K_t in (4-11), and let ρ denote the conditional density ρ_t. It has been proved in [1] that the optimal gain is obtained as the solution to the following Euler-Lagrange boundary value problem (EL-BVP):

$$\begin{aligned} \nabla \cdot \Big(\frac{1}{\rho(x)} \nabla(\rho(x)\mathsf{K}(x)) \Big) &= -\frac{1}{\sigma_W^2} \nabla c, \\ \lim_{x \to \infty} \mathsf{K}(x)\rho(x) &= 0. \end{aligned} \tag{4-17}$$

Denoting $U = -\log(\rho) \in C^1$, it is easy to show by simple integration that the EL-BVP in (4-17) can be equivalently characterized in the standard form of Poisson's equation for the Langevin diffusion (4-18).

$$\mathcal{D}h = -\nabla U \cdot \nabla h + \Delta h = -\tilde{c}, \qquad \tilde{c} = c - \int c(x)\rho(x)\,\mathrm{d}x, \tag{4-18}$$

where the gain K coincides with the gradient of h:

$$\mathsf{K}(x) = \nabla h(x)\,, \quad x \in \mathsf{X}\,. \tag{4-19}$$

The formula for the gain is thus obtained via a solution to the Poisson's equation: Assuming ∇U is globally Lipschitz continuous, and under some regularity conditions on c, the solution h and its gradient exists, as discussed in Chapter 2. In the remainder, it is assumed that these regularity conditions are satisfied and that the gain K exists and is unique. Theorem 3.3 in [1] states that provided the prior densities match, i.e. $\rho_0^*(x) = \rho_0(x)$, the FPF in (4-16) with the gain K obtained as a solution to (4-17) is consistent with the optimal nonlinear filter (4-3), i.e.

$$\rho_t(x) = \rho_t^*(x). \tag{4-20}$$

An estimate of ρ_t given by the empirical distribution of the particles $\rho_t^{(N)}$ approximates ρ_t^* as $N \to \infty$.

4.4 FPF Gain Approximation

The major challenge in the implementation of the FPF is the computation of the gain function. The exact computation of K is intractable outside of these particular cases.

(i) When U is quadratic and c is linear, then K is a constant independent of x, and can be interpreted as a Kalman gain [83].

(ii) In the general scalar case, when $d = 1$, K has an explicit solution:

$$\mathsf{K}(x) = -\frac{1}{\rho(x)} \int_{-\infty}^{x} (c(y) - \hat{c})\rho(y)\mathrm{d}y. \tag{4-21}$$

This motivates the use of approximation techniques to obtain an estimate of K that is optimal in some meaningful metric. In the first part of this section, we discuss Galerkin-based gain approximation algorithms. An algorithm based on approximation of the transition kernel of the Langevin diffusion was developed in parallel research by Taghvaei et al. [84] and error analysis of this method was studied in [85]. This algorithm has been used in this dissertation as an important benchmark for comparing the performance in numerical simulations. A short review is provided in Section 4.4.2.

As motivated in Chapters 2 and 3, approximation algorithms based on ∇-LSTD learning form the core topic of this dissertation. It is evident from (4-19), that the objective function

used in (2-35) suits the objective of approximating the gain K very well:

$$\|\nabla h - \nabla h^\theta\|_{L^2}^2 = \|\mathsf{K} - \mathsf{K}_\theta\|_{L^2}^2. \tag{4-22}$$

Thus, ∇-LSTD learning algorithms can be directly applied for gain approximation. Two refinements of the RKHS based ∇-LSTD learning algorithm to enable online gain estimation, are proposed in Section 4.5.

4.4.1 Galerkin-based Methods

Galerkin-based algorithms are a result of a weak formulation of the EL-BVP (4-17). A function h is called a weak solution to the Poisson's equation (4-17) if

$$\int \nabla h(x) \cdot \nabla \varphi(x) \rho(x) dx = \int \tilde{c}(x) \varphi(x) \rho(x) dx, \forall \varphi \in H_1^0, \tag{4-23}$$

where $\varphi(x)$ are called test functions. This formulation is called a Galerkin relaxation. A finite dimensional solution to (4-23) is obtained by choosing an ℓ-dimensional function class $\mathcal{H} := \{h^\theta : \theta \in \mathbb{R}^\ell\}$ to approximate h and a collection of ℓ test functions $\{\varphi_j\}_{j=1}^\ell$. In prior work [1, 42], $\mathcal{H}$ is considered to be a linearly parameterized family of functions of the form $h^\theta := \sum_{j=1}^\ell \theta_j \psi_j$, where $\{\psi_j\}$ are called the basis functions. It is assumed that each ψ_j is continuously differentiable, with gradient denoted $\nabla \psi_j$. These functions are used to define the approximation of the filter gain

Anand:Need to verify H_1^0.

Anand:reference?

$$K_\theta = \nabla h^\theta = \sum_{j=1}^\ell \theta_j \nabla \psi_j. \tag{4-24}$$

Then, the Galerkin relaxation of Poisson's equation (4-23) is defined to be the set of ℓ equations in the parameter θ. A particular case is when the set of test functions and basis functions chosen are the same. The solution θ in this case is given by:

$$\begin{aligned} \theta &= M^{-1} b, \\ &\text{where,} \\ M &:= \langle \nabla \psi, \nabla \psi \rangle, \quad b := \langle \tilde{c}, \psi \rangle. \end{aligned} \tag{4-25}$$

Empirical estimates for M and b can be computed using particles. If the particles are distributed according to ρ, then the solution (4-25) is the same as the optimal solution obtained using ∇-LSTD learning for a linear parameterization (2-43). This is a matter of coincidence rather than design, and in general, Galerkin formulations do not result in norm minimization.

Another Galerkin formulation is obtained in terms of the Bellman error. Recall that the FPF gain is the derivative of the solution to Poisson's equation (4-18). For the linear parameterization, the following formulation results in ℓ linear equations as before:

$$\begin{aligned} 0 &= \langle \underbrace{\mathcal{D}h^\theta + \tilde{c}}_{\text{Bellman Error}} , \varphi_j \rangle \\ &= \int (-\nabla U(x) \cdot \nabla h^\theta(x) + \Delta h^\theta(x) + \tilde{c}(x)) \varphi_j(x) \mathsf{d}x, \qquad 1 \le j \le \ell. \end{aligned} \tag{4-26}$$

A commonly used estimate for K that is easily computable is the constant gain approximation, denoted as K^*. This is obtained by choosing $\{\psi_j(x) = x_j\}_{j=1}^d$ as the basis and test functions in the Galerkin relaxation. This results in $M = I_d$ and,

$$\widehat{\mathsf{K}}_j^* := b = \langle \tilde{c}, x_j \rangle \tag{4-27}$$

$$\approx \frac{1}{N} \sum_{i=1}^N [c(x_j^i) - \hat{c}] x_j^i, \qquad 1 \le j \le d, \tag{4-28}$$

where $\widehat{\mathsf{K}}_j^*$ is the j^{th} component of the gain. Alternately, the constant gain can also be interpreted as the minimizer of

$$\widehat{\mathsf{K}}^* := \arg\min_{\widehat{\mathsf{K}} \in \mathbb{R}^d} \|\mathsf{K} - \widehat{\mathsf{K}}\|_{L^2}^2 \tag{4-29}$$

where the minimum is over deterministic vectors. The solution is evidently the mean, $\widehat{\mathsf{K}}^* = \mathsf{E}[\mathsf{K}]$. The proof of the representation (4-28) can also be obtained by applying (2-36):

$$\widehat{\mathsf{K}}_j^* = \langle \mathsf{K}^*, e_j \rangle_{L^2} = \langle \tilde{c}, \psi_j \rangle_{L^2}, \qquad 1 \le j \le d,$$

where $\{e_j\}$ are the standard basis elements in $\mathbb{R}^d$. This approximation has been successfully tested in [86]. The constant gain is an important component of the refinements to the RKHS based ∇-LSTD algorithm proposed in Section 4.5.

The Galerkin methods provide a good algorithmic framework for approximating the gain. However, in general it is not easy to obtain performance guarantees. In this dissertation, we obtain approximations by solving the minimum norm problem (2-35) using ∇-TD learning. Under general conditions we can be assured that the approximation is the minimum-norm optimal solution.

4.4.2 Markov Semigroup Approximation

In this section, a very concise overview of the Markov semigroup approximation for the FPF gain function is provided. The algorithm was initially presented in [84], followed by error analysis in [85]. While the approach is entirely different, it is likely that concepts from this concurrent work can provide valuable insights to design an improved solution that combines the merits of both the approaches.

The algorithm is based on a semigroup formulation of the Poisson's equation for Langevin diffusion (2-6). The function h can be expressed as the solution to the following fixed-point equation:

$$h = P_\epsilon h + \int_0^\epsilon P_s(h - \hat{h})\mathrm{d}s, \tag{4-30}$$

where P_ϵ refers to the transition semigroup of the Langevin diffusion, defined in (2-4). The main step is the use of an approximate semigroup T in place of P_ϵ. An empirical solution to the fixed-point equation at the particle locations is obtained through successive approximation. The gain is subsequently obtained by taking the gradient. A summary of the algorithm to approximate the gain for a fixed time t is tabulated in Algorithm 13:

Algorithm 1. *Markov semigroup gain function approximation algorithm*

Require: $\{x^i\}_{i=1}^N, \{c(x^i)\}_{i=1}^N, h_{prev}, \epsilon, L.$

Ensure: $\{\mathsf{K}_i\}_{i=1}^N$

1: Calculate $K_{ij} = \exp(-\|x^i - x^j\|^2/4\epsilon)$ for $i, j = 1$ to N ▷ Gaussian kernel

2: Calculate $\kappa_{ij} = \frac{K_{ij}}{\sqrt{\sum_k K_{ik}}\sqrt{\sum_k K_{jk}}}$ for $i, j = 1$ to N

3: Calculate $d_i = \sum_j \kappa_{ij}$ for $i = 1$ to N

4: Calculate $\mathrm{T}_{ij} = \frac{\kappa_{ij}}{d_i}$ for $i, j = 1$ to N ▷ Approximation of the Markov kernel for Langevin diffusion

5: Calculate $\pi_i = \frac{d_i}{\sum_j d_j}$ for $i = 1$ to N

6: Calculate $\hat{c} = \sum_{i=1}^{N} \pi_i c(x^i)$

7: Initialize $h = h_{prev}$

8: **for** $t = 1$ to L **do**

9: $h_i = \sum_{j=1}^{N} \mathrm{T}_{ij} h_j + \epsilon(c - \hat{c})$ for $i = 1$ to N ▷ Successive approximation

10: **end for**

11: Calculate $r_i = h_i + \epsilon c_i$ for $i = 1$ to N

12: Calculate $s_{ij} = \frac{1}{2\epsilon}\mathrm{T}_{ij}(r_j - \sum_{k=1}^{N} \mathrm{T}_{ik} r_k)$ for $i, j = 1$ to N

13: Calulate $\mathsf{K}_i = \sum_j s_{ij} x^j$ for $i = 1$ to N ▷ FPF gain function approximation

end

Similar to the RKHS-based ∇-LSTD learning, the Markov semigroup approximation provides a basis-independent solution to FPF gain function approximation. The only hyperparameter that needs to be tuned is the time-step parameter ϵ and it has been remarked in [84] that the approximation is valid for small values of ϵ. However, the algorithm may lead to two potential difficulties. Solving for the Poisson's equation on the particles may result in overfitting, and numerical issues may be magnified in the subsequent gradient approximation. The RKHS method addresses these difficulties via regularization. Another potential drawback is the use of successive approximation, which might require a large number of iterations to converge.

Question:I had it before we discussed about it, I can remove this.

In this dissertation, we use this algorithm acts as a benchmark for comparing the performance of ∇-LSTD learning based techniques, as this is the only known basis-independent technique to approximate K. Numerical examples that compare the performance are discussed in Section 4.7.

4.5 Enhanced ∇-LSTD Algorithms for FPF Gain Approximation

In this section, two new enhancements to the RKHS based ∇-LSTD learning algorithm described in Chapter 3 are presented.

(i) Dynamic regularization - The online filtering problem is considered here. To obtain the estimates of the posterior using the FPF, we need to compute the gain function K_t for all t. This necessitates computationally simple algorithms that can update the gain function in an iterative fashion. The basic idea is to modify the ERM to include a term with the gain function at the previous instant.

(ii) Utilizing the constant gain - To obtain good approximations using the RKHS based ∇-LSTD algorithm, a suitable choice of the hyperparameters λ and ε are required. As described in Chapter 3, there are no standard rules to choose the optimal values and the selection is usually done by cross-validation in a supervised learning setting. However, this may not always be possible in gain function approximation and any additional information that can aid in this selection or make the algorithm robust to the values of hyperparameters is useful. We propose to use the easily computable constant gain approximation (4-29) as an additional constraint in the ERM problem.

It is possible to apply both the enhancements independently or simultaneously to improve performance.

4.5.1 Dynamic Regularization - ∇-LSTD-RKHS with Memory

In a discrete implementation of the FPF, it is assumed that time is sampled with constant inter-sampling time δ. The gain updates are performed at $t = n\delta$, and the FPF uses K_n rather than K_t. It is expected that $\mathsf{K}_n = \mathsf{K}_{t_n} \approx \mathsf{K}_{t_{n-1}}$ if $\delta \approx 0$. This is the motivation for the dynamic regularization developed in this section. Given an additional regularization parameter λ_1, the proposed ERM is defined as in (3-19), with modified loss function:

$$\begin{aligned} g_n^* &:= \arg\min_{g\in\mathcal{H}} \frac{1}{N}\sum_{j=1}^{N} L_n(x_n^j, g, \nabla g) + \lambda\|g\|_{\mathcal{H}}^2 \\ L_n(x, g, \nabla g) &:= \|\nabla g(x)\|^2 - 2\tilde{c}_N(x)g(x) + \lambda_1\|\nabla g(x) - \nabla g_{n-1}(x)\|^2 \end{aligned} \tag{4-31}$$

The extended representer theorem (Theorem 3.3) again leads to a solution of the form (3-27), and we then take $\mathsf{K}_{t_n}(x_n^j) = \nabla g_n^*(x_n^j)$. A reduced complexity approximation for the gain function at step n can be obtained using (3-36) as

$$g_n^*(.) := \sum_{j=1}^{N} \beta_{j,n} K(x_n^j, .)\,. \tag{4-32}$$

Substituting (4-32) into (4-31), and using the vector/matrix notation as defined in (3-34) gives,

$$\beta_n^* = \arg\min_{\beta\in\mathbb{R}^N} \frac{1}{N}\Big[(1+\lambda_1)\beta^{\intercal}\Big(\sum_{k=1}^{d} M_{0k}^{\intercal} M_{0k}\Big)\beta - \beta^{\intercal}(2\lambda_1 \sum_{k=1}^{d} M_{0k}^{\intercal}\mathsf{K}_{n-1,k} + M_{00}\boldsymbol{\zeta}\Big)\Big] + \lambda\beta^{\intercal} M_{00}\beta \tag{4-33}$$

This is a quadratic optimization problem with solution

$$\begin{aligned} &\beta_n^* = M^{-1}b \\ \text{with}\quad &M = (1+\lambda_1)\sum_{k=1}^{d} M_{0k}^{\intercal} M_{0k} + \lambda N M_{00} \\ &b = M_{00}\boldsymbol{\zeta} + \lambda_1 \sum_{k=1}^{d} M_{0k}^{\intercal}\mathsf{K}_{n-1,k} \end{aligned} \tag{4-34}$$

4.5.2 Utilizing the Constant Gain Approximation (∇-LSTD-RKHS-OM)

The constant gain approximation (4-28) has been shown to work well in applications [86]. It is also easy to compute, so it is natural to impose the constraint at time $t = n\delta$,

$$\nabla g = \widehat{\mathsf{K}}_{t_n}^* + \nabla\tilde{g}$$

in which $\tilde{g} \in \mathcal{H} \cap C^1$, and the mean of $\nabla\tilde{g}$ under the density ρ_t is equal to zero. It is not difficult to introduce this additional constraint in any of the ERM formulations. This algorithm with the additional constraint is termed the ∇-LSTD-RKHS optimal mean (OM) algorithm.

To simplify notation, dependency on n (or t) is suppressed. The constrained optimization problem is defined as follows:

$$\begin{aligned}\tilde{g}^* := \arg\min_{\tilde{g}\in\mathcal{H}} \quad & \|\nabla h - \widehat{\mathsf{K}}^* - \nabla\tilde{g}\|_{L_2}^2 \\ \text{s.t.} \quad & \langle \partial_{x_k}\tilde{g}, 1\rangle_{L_2} = 0, \quad 1 \le k \le d\end{aligned}$$

where $\widehat{\mathsf{K}}^*$ is defined in (4-28), and "1" is the constant function, identically equal to unity.

The solution can be obtained by finding a saddle point for the Lagrangian, with Lagrange multiplier $\mu \in \mathbb{R}^d$:

$$L(\tilde{g},\mu) := \|\nabla h - \widehat{\mathsf{K}}^* - \nabla\tilde{g}\|_{L_2}^2 + \langle \mu, \nabla\tilde{g}\rangle_{L_2} \tag{4-35}$$

Expanding the quadratic, and applying Prop. 2.1 as in previous ERM formulations gives

$$L(\tilde{g},\mu) = \|\nabla h - \widehat{\mathsf{K}}^*\|_{L_2}^2 + \|\nabla\tilde{g}\|_{L_2}^2 - 2\langle\tilde{c},\tilde{g}\rangle_{L_2} + 2\langle\widehat{\mathsf{K}}^*, \nabla\tilde{g}\rangle_{L_2} + \langle\mu,\nabla\tilde{g}\rangle_{L_2} \tag{4-36}$$

The pair $(\tilde{g}^*, \mu^*)$ are obtained through the max-min problem:

$$\max_{\mu}\min_{\tilde{g}} L(\tilde{g},\mu) \tag{4-37}$$

As in each previous setting, this is approximated by a regularized ERM. An empirical saddle-point problem is defined as follows,

$$\begin{aligned}(\mu^*,\tilde{g}^*) &= \arg\max_{\mu}\Big(\arg\min_{\tilde{g}\in\mathcal{H}}\Big[\frac{1}{N}\sum_{i=1}^{N} L(x^i,\mu,g,\nabla g) + \lambda\|g\|_{\mathcal{H}}^2\Big]\Big) \\ L(x,\tilde{g},\nabla\tilde{g},\mu) &= \|\nabla\tilde{g}\,(x)\|^2 - 2\tilde{c}_N(x)\tilde{g}(x) + \nabla\tilde{g}\,(x)\cdot[2\widehat{\mathsf{K}}^* + \mu]\end{aligned} \tag{4-38}$$

The extended representer theorem (Theorem 3.3) again leads to a solution of the form (3-27) for the optimizer $\tilde{g}^*$. A closed form expression is possible because this reduces to a quadratic program in β^*. An explicit solution is presented here only for the reduced complexity approximation, in which the optimization is performed over the finite-dimensional subspace (3-36). Let κ denote the matrix whose k^{th} column is equal to $K_{x_k}\mathbf{1}$, with $\mathbf{1}$ the column vector consisting of ones. A suboptimal solution over the subspace (3-36) is obtained, similar to the

computation leading to (4-32):

$$\beta^* := \arg\min_{\beta\in\mathbb{R}^N} \frac{1}{N}\Big[\beta^{\intercal}\Big(\sum_{k=1}^{d} M_{0k}^{\intercal} M_{0k}\Big)\beta - 2\beta^{\intercal} M_{00}\boldsymbol{\zeta} + 2\beta^{\intercal}\kappa\widehat{\mathsf{K}}^* + \beta^{\intercal}\kappa\mu\Big] + \lambda\beta^{\intercal} M_{00}\beta \tag{4-39}$$

Taking derivatives with respect to β and μ and equating to zero gives $N+d$ linear equations in $N+d$ unknowns (4-40):

$$\begin{aligned} 0 &= 2\Big(\frac{1}{N}\sum_{k=1}^{d} M_{0k}^{\intercal} M_{0k} + \lambda M_{00}\Big)\beta^* + \frac{\kappa\mu^*}{N} + \frac{2}{N}\Big(\kappa\widehat{\mathsf{K}}^* - M_{00}\boldsymbol{\zeta}\Big) \\ 0 &= \kappa^{\intercal}\beta^* \end{aligned} \tag{4-40}$$

The gain K is then computed as $\mathsf{K} = \widehat{\mathsf{K}}^* + \nabla\tilde{g}^*$.

4.5.3 Summary of all ∇-LSTD algorithms

Algorithm 2. *∇-LSTD algorithm with Langevin SDE simulation*

Require: $\{x^i\}_{i=1}^N, \{c(x^i)\}_{i=1}^N, \{\psi_j\}_{j=1}^\ell$, δ, T.

Ensure: smooth posterior $\rho(x)$, gain at particle locations $\{\mathsf{K}\}_{i=1}^N$

1: Compute smooth posterior density $\rho(x)$ from $\{x^i\}_{i=1}^N$ using Expectation-Maximization (Appendix E) or kernel density estimation algorithms.

2: Simulate the discretized Langevin SDE (2-2) for a chosen δ upto time T.

3: Calculate φ, M, b according to (2-60c).

4: Calculate $\theta^* = M^{-1}b$.

5: Calculate $\nabla h(x) = \sum_{j=1}^{\ell} \theta^* \nabla\psi(x)$ ▷ Gradient approximation

6: Calculate $\mathsf{K}_i = \nabla h(x^i)$ for $i = 1$ to N. ▷ FPF gain function approximation

end

Algorithm 3. *∇-LSTD-L algorithm with finite basis*

Require: $\{x^i\}_{i=1}^N, \{c(x^i)\}_{i=1}^N, \{\psi_j\}_{j=1}^\ell$.

Ensure: $\{\mathsf{K}_i\}_{i=1}^N$

1: Calculate M and b according to (2-47)

2: Calculate $\theta^* = M^{-1}b$

3: Calculate $\nabla h(x) = \sum_{j=1}^{\ell} \theta^* \nabla\psi(x)$ ▷ Gradient approximation

4: Calculate $\mathsf{K}_i = \nabla h(x^i)$ for $i = 1$ to N. ▷ FPF gain function approximation

end

Algorithm 4. *∇-LSTD-RKHS algorithms for gain function approximation*

Require: $\{x^i\}_{i=1}^N, \{c(x^i)\}_{i=1}^N, \beta_{prev}, \varepsilon, \lambda, \lambda_1$.

Ensure: $\{\mathsf{K}_i\}_{i=1}^N$

1: Calculate $K_{ij} = \exp(-\|x^i - x^j\|^2/4\varepsilon)$ for $i, j = 1$ to N ▷ Gaussian kernel

2: Calculate $\partial_{x_k} K_{ij} = -\frac{(x_k^i - x_k^j)}{2\varepsilon} K_{ij}$ for all $i, j = 1$ to N and $k = 1$ to d ▷ Gaussian kernel derivatives

3: Calculate M and b by solving one of the following: (3-35) for ∇-LSTD-RKHS-Opt, (3-37) for ∇-LSTD-RKHS-Simple, (4-34) for ∇-LSTD-RKHS with memory or (4-40) for ∇-LSTD-RKHS-OM.

4: Calculate $\beta^* = M^{-1}b$

5: Calulate $\mathrm{K}_i = \sum_{j=1}^N \beta_j^{0*} K_{ij} + \sum_{j=1}^N \sum_{k=1}^d \beta_j^{k*} \partial_{x_k} K_{ij}$ for $i = 1$ to N ▷ FPF gain function approximation

end

Question:Could you check this?

Question:Should I keep this? I can remove the section and just have it as a paragraph instead

4.6 Complexity Comparison of the Algorithms

The Markov semigroup approximation attempts to solve an approximate problem exactly, whereas the ∇-LSTD-RKHS method tries to find the best approximate solutions without altering the problem. Due to their different approaches, there is no direct way of comparing the performances using analysis. However, it is firmly believed that the ∇-LSTD-RKHS method produces asymptotically unbiased approximations.

4.7 Numerical Experiments

We begin by presenting algorithms to obtain a smooth approximation of the empirical posterior estimate in Section 4.7.1. In Section 4.7.2, a short discussion on the potential numerical issues the true FPF gain and its approximations may exhibit is provided, followed by tricks to resolve these problems in a practical implementation. These issues mainly impact the ∇-LSTD algorithms that require an SDE simulation. Subsequently, we present a number

of numerical experiments to compare the performance of the various gain approximation algorithms discussed in this dissertation. The experiments can be broadly classified as:

- Those concerning approximating the gain for a fixed time t
- Those concerning an online filtering problem.

The second set of experiments is more challenging as it requires continuous estimation of the gain at the sampling instants. Extended Kalman filter (EKF) and sequential importance sampling (SIS) particle filter are also used for performance comparison in the filtering examples.

4.7.1 Smooth approximations of the posterior

First, we are concerned with the gain approximation for a fixed t. The general version of the ∇-LSTD learning algorithm in Section 2.3.3 requires the simulation of the Langevin SDE (2-1). In the FPF, estimates to the posterior density ρ are only available in the form of its empirical equivalent $\rho^{(N)}$. To simulate the Langevin SDE, one needs to obtain a smooth potential function $U = -\log \rho$. This necessitates the smoothing of the empirical estimate $\rho^{(N)}$. The contribution of the work in [27] is the approximation of the FPF gain, along with a smooth approximation of the empirical distributions. Therefore, before choosing a parameterized family of continuous gain functions $\mathcal{K} = \{\mathsf{K}_\theta : \theta \in \mathbb{R}^\ell\}$, it is required to specify a parameterized family of smooth and continuous densities, denoted $\mathcal{P} = \{\rho_\alpha : \alpha \in \mathbb{R}^m\}$, where each α_i for $1 \le i \le m$ is a parameter vector that specifies a member density from within the family $\mathcal{P}$. The gain approximation therefore consists of two steps:

1. Obtain $\rho \in \mathcal{P}$ that most closely approximates $\rho^{(N)}$.
2. Obtain an approximation within $\mathcal{K}$ for the FPF gain using the ∇-LSTD learning algorithm in Section 2.3.3, based on the solution to Step 1.

The choice of $\mathcal{P}$ will depend on the application. Most of the examples in this dissertation consider a finite-dimensional class of univariate Gaussian mixture models (GMM). An

m-component Gaussian mixture density has the following general form:

$$\rho(x) = \sum_{i=1}^{m} w_i \rho_i(x), \qquad \sum_{i=1}^{m} w_i \quad = 1, \tag{4-41}$$

where each ρ_i is a Gaussian density with mean μ_i and standard deviation σ_i, and w_i as its weight. In this case, the parameter vector $\alpha_i := [\mu_i, \sigma_i, w_i]$. The density estimation problem in Step 1 is defined as follows: Given the set of N particles $\{x^i\}_{i=1}^N$, and a family $\mathcal{P}$ of probability density functions, the goal is to find $\rho \in \mathcal{P}$ that is most likely to have generated the given particles. In this dissertation, the Expectation Maximization (EM) algorithm is used to obtain the parameter estimates corresponding to the maximum a posteriori (MAP) optimal density. The EM algorithm is an iterative procedure to obtain the maximum-likelihood estimates of an analytically intractable cost function. The steps involved in the algorithm for GMM estimation is provided in Appendix E. For the nonlinear oscillator example, where the diffusion is on the unit circle, we consider a mixture of von Mises densities [87]. Kernel density estimators (KDE) offer a non-parametric alternative to the EM algorithm. This has not been explored in this dissertation.

The choice of $\mathcal{K}$ is also problem-specific – examples are given in Section **??**. The EM algorithm followed by the simulation of the Langevin SDE for implementing the ∇-LSTD learning algorithm adds a layer of additional complexity to the FPF. For online estimation in a filtering setting, the EM algorithm at each instant t can be initialized with the ML parameters obtained from the previous instant for faster convergence.

4.7.2 Numerical Issues with the Gain

In this section, we highlight some of the numerical issues associated with the optimal FPF gain and how the approximation method described in Section 4.7.1 can be adapted to ensure the stability of the filter. For $d = 1$, the optimal gain takes the integral form in (4-21):

$$\mathsf{K}(x) = -\frac{1}{\rho(x)} \int_{-\infty}^{x} (c(y) - \hat{c})\rho(y)\mathrm{d}y$$

If $c(x) = x$ and ρ is a Gaussian mixture model of the form (4-41), the gain $\mathsf{K}(x)$ is given by,

$$\mathsf{K}(x) = \frac{1}{\sigma_W^2 \rho(x)} \Big(\sum_{i=1}^{m} w_i (\hat{c} - c(\mu_i)) F_i(x) + \sum_{i=1}^{m} w_i \sigma_i^2 \rho_i(x) \Big), \tag{4-42}$$

where $F_i = \int_{-\infty}^{x} \rho_i(x) \mathrm{d}x$ is the Gaussian CDF. The second term with the Gaussian PDFs is well-behaved and approaches to a constant that is equal to the Kalman gain corresponding to ρ_i with the largest σ_i, as $x \to \infty$. But, the first term becomes numerically unstable and results in high magnitudes of gain in regions where $\rho(x)$ is small. This issue can be clearly seen if we consider the bimodal densities ρ_1 and ρ_2 shown in Fig. 4-2. In spite of being nearly similar, their corresponding gains K_1 and K_2 are very different, with the peak value of K_2 nearly 7 times the peak value of K_1 owing to the "deeper valley" of ρ_2. High magnitudes of FPF gain affect the particles that lie in this region of the state space, leading to numerical instabilities in filtering.

Anand:need to come back and verify

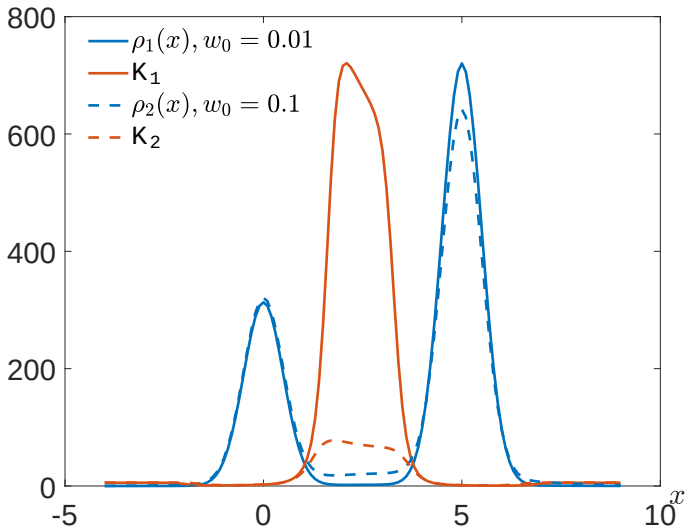


Figure 4-2. True FPF gains K_1 for ρ_1 with $w_0 = 0.01$ and K_2 for ρ_2 with $w_0 = 0.1$.

These numerical issues also impact the performance of ∇-LSTD learning algorithm. The rate of convergence of the parameters depends on properties of the stochastic process $\mathbf{\Phi}$ described by Langevin's diffusion (2-1), and the associated stochastic process that defines the eligibility vectors in (2-60a). While a stationary version of this process exists in all of the

examples considered here, it will be seen that in some examples the sample paths take on large values, which contributes to high variance. These issues are most pronounced when the mixture density has a very shallow "valley", which is precisely the situation that leads to very large values of the true gain K.

One of the practical approaches to prevent K from taking extremely high values is to introduce a third Gaussian density ρ_0 to "fill the well". The third density $\rho_0(x)$ is chosen as $\mathcal{N}(\mu_0, \sigma_0^2)$, where μ_0 and σ_0 correspond to the sample mean and standard deviation of the entire particle population. The mixture probability w_0 of ρ_0 in the overall distribution ρ is forced to be greater than a threshold value. This is further illustrated by the sharp decline in the peak magnitude of K_1 compared to K_2 in Fig. 4-2.

4.7.3 Gain Function Approximation for a Fixed t

In each of the following experiments the density $\rho \in \mathcal{P}$ was chosen with $m = 2$,

$$\rho = 0.5\mathcal{N}(-1, 0.4472) + 0.5\mathcal{N}(1, 0.4472), \tag{4-43}$$

such that each component has a variance of 0.2. The observation function is linear, with $c(x) \equiv x$ and $\sigma_W = 1$.

Linear parameterization

It is reasonable to search for a basis that offers flexibility in regions where the density ρ takes on non-negligible values. We consider three different choices for basis functions as follows:

(i) A polynomial basis, such that $\psi_n(x) = x^n$ for $1 \leq n \leq \ell$.

(ii) A Fourier basis composed of sines and cosines of harmonic frequencies, $\psi_n(x) = \sin(nx), \psi_{n+1}(x) = \cos(nx)$ for $1 \leq n \leq \ell/2$.

(iii) Polynomials weighted by the component densities ρ_i, defined as:

$$\{\psi_n(x) : x \in \mathbb{R},\ 1 \leq n \leq \ell\} = \{x^k \rho_i(x) : 1 \leq k \leq \ell/2,\ i = 1, 2\} \tag{4-44}$$

It is not difficult to show that the true gain is nearly constant for large x, with $\lim_{|x|\to\infty} \mathsf{K}(x) = \mathsf{K}_{\mathsf{kal}}$. The limit corresponds to the Kalman-like gain $\mathsf{K}_{\mathsf{kal}}$ obtained for the model in which ρ is replaced by ρ_i (the Gaussian density with the highest variance). The function $\psi_n(x) = x$ is added to the basis (for choices ii and iii, it is already included in i), such that $\nabla\psi(x) \equiv 1$ is present to account for this constant asymptotic gain value. The class $\mathcal{K}$ is defined using this basis:

$$K_\theta = \theta^{\intercal}\nabla\psi = \sum_{n=1}^{\ell} \theta_n \nabla\psi_n \,, \quad \theta \in \mathbb{R}^\ell$$

Nonlinear parameterization

In a nonlinear parameterization setting, the following form is chosen for the class $\mathcal{K}$.

$$K_\theta(x) = K_0 + \sum_{i=1}^{l} \xi_i^\theta(x) \tag{4-45}$$

Of the various nonlinear parameterizations tested, the following produced the best results,

$$\xi_i^\theta(x) = \frac{a_i}{(x - b_i)^2 + c_i^2} \tag{4-46}$$

The coefficients $\alpha_i = \{a_i, b_i, c_i\}, K_0$ constitute the parameters to be estimated. In the simulation results surveyed here $l = 3$ and hence, $\theta \in \mathbb{R}^\ell$ with $\ell = 10$.

Optimal nonlinear parameterization can be obtained by running ∇-TD learning using stochastic approximation techniques, discussed in Section 2.3.4. Two such techniques were used, namely stochastic Newton Raphson and approximation with Polyak averaging [88]. The scalar gain term γ_t is set to $1/(t+1)^\beta$. For stochastic Newton Raphson, β is chosen to be 1, and $\beta = 0.6$ was chosen for Polyak averaging.

Basis-free algorithms

The following basis-free approximation approaches were compared:

(i) RKHS based methods, both optimal and simplified versions of Section 1.2.3.

(ii) RKHS method with optimal mean, using the constant gain approximation (RKHS-OM) of Section 4.5.2.

(iii) Markov semigroup approximation method of Section 4.4.2.

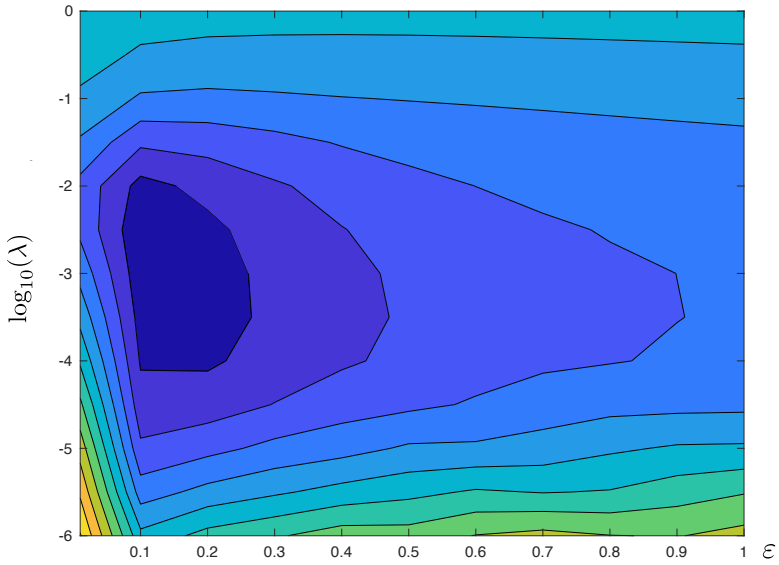


Figure 4-3. Contour plots of average over 100 trials of $\log(\|\mathsf{K} - \widehat{\mathsf{K}}\|_{L^2}^2)$ with λ and ε with $N = 500$.

For each of the RKHS methods, the standard Gaussian kernel (3-16) was used, centered at the particle locations $\{x^i\}_1^N$. Fig. 4-3 illustrates the sensitivity in gain approximation to the hyperparameters λ and ε for $N = 500$. The contour plots in Fig. 4-3 correspond to the log of the average mean square error in the gain approximation $\|\mathsf{K} - \widehat{\mathsf{K}}\|_{L^2}^2$; estimated by observations over 100 independent trials. It is evident that a larger value for λ prevents overfitting and a smaller value for ε provides more flexibility. With increase in N, the best choices of λ and ε show a declining trend. Based on these results from sensitivity analysis, $\lambda = 10^{-2}$ and $\varepsilon = 0.1$ was chosen for the RKHS methods.

Results and discussion

Anand:Change ρ_1 in the legend to ρ

Fig. 4-4 compares the performance of the various choices of basis functions for different simulation times $T = 10^4, 10^5, 10^6$ of the Langevin SDE. An Euler-discretization scheme was implemented with time-step size of $\delta = 0.01$. A 10-dimensional basis was chosen in each case. The true FPF gain computed using the integral formula (4-21) is also plotted for comparison. The histogram of the particles $\{x^i\}$ obtained from the SDE and the true density ρ are shown as the shaded region. As expected, the accuracy of the approximation improves as T increases.

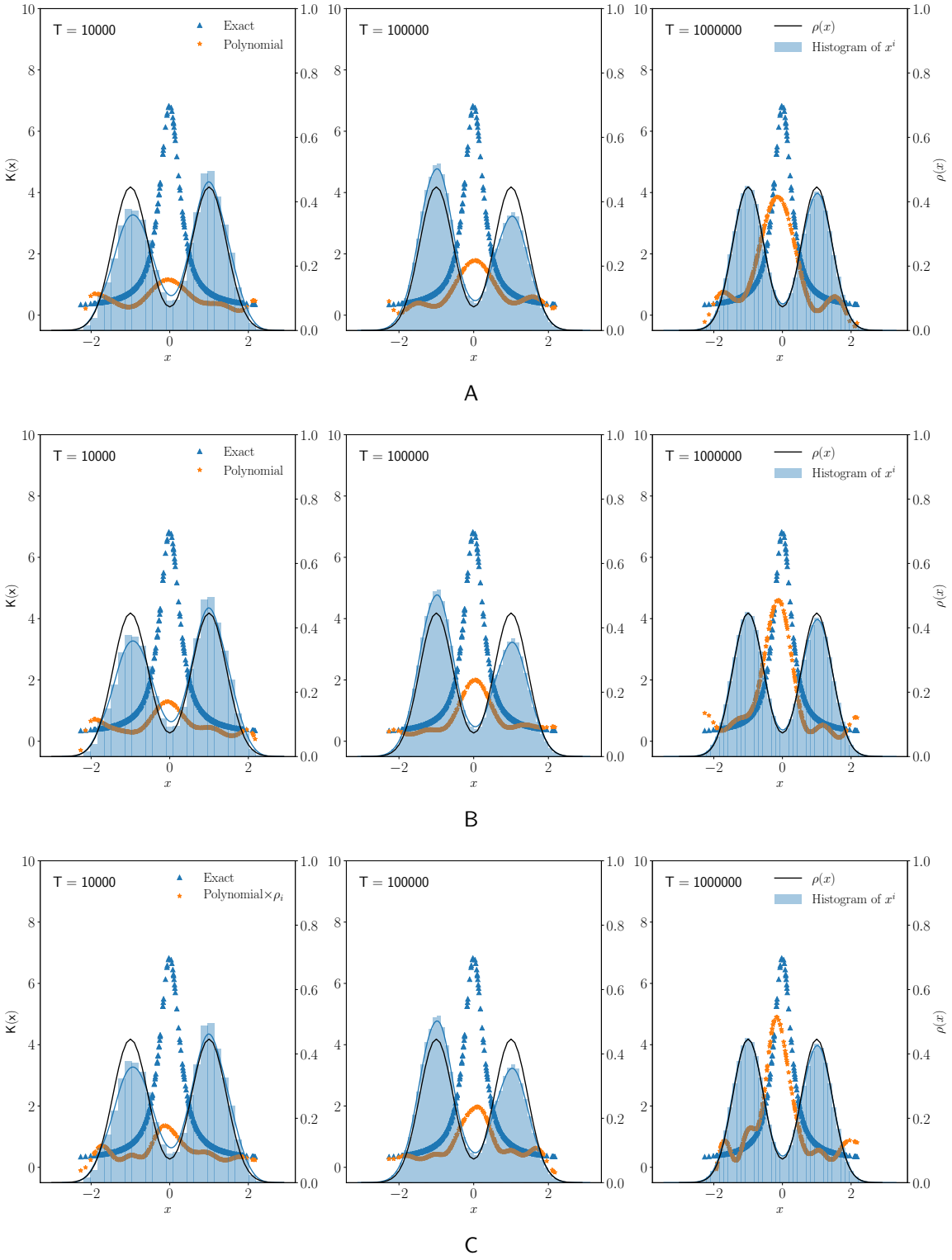


Figure 4-4. Gain function approximations using ∇-LSTD learning algorithm (Section 2.3.3) using a linear parameterization [27] A) Polynomial basis, B) Fourier basis consisting of sines and cosine functions, C) Polynomials weighted by ρ_i.

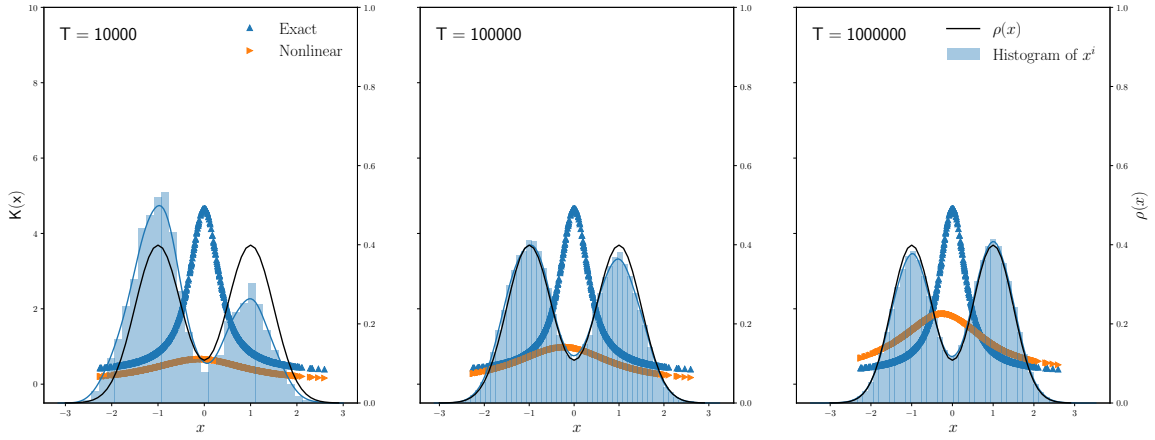


Figure 4-5. Gain function approximations using a 9-dimensional nonlinear parameterization of the form using ∇-LSTD-L algorithm (Section 2.3.3)

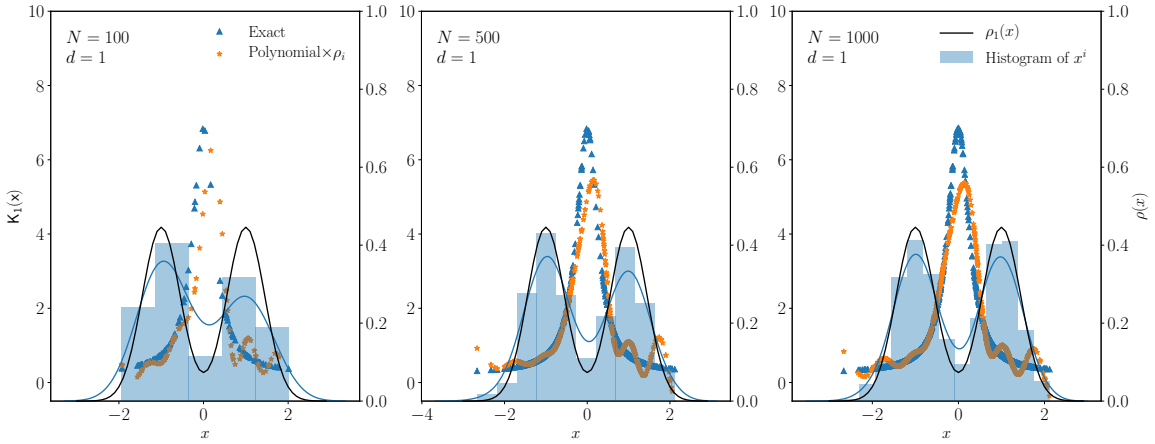
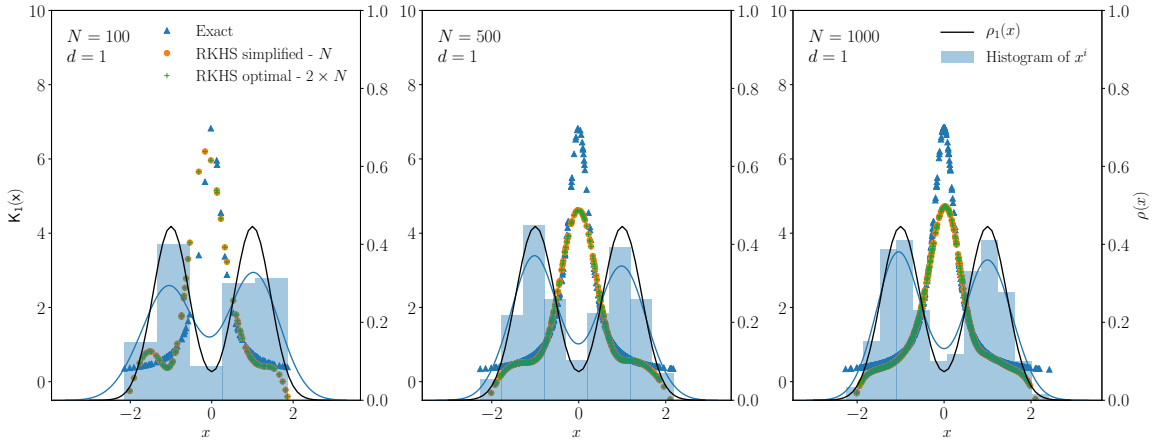


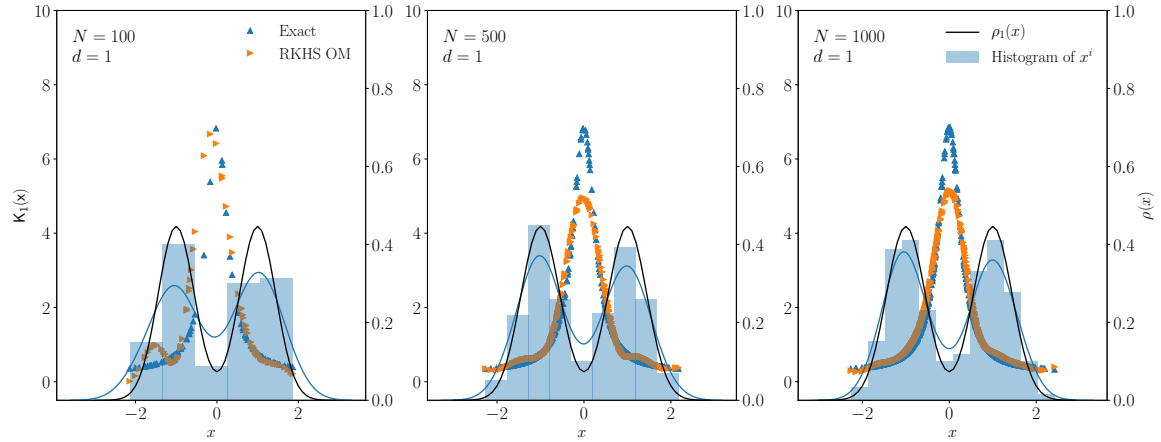
Figure 4-6. Gain function approximations using a 10-dimensional polynomial$\times\rho_i$ basis using ∇-LSTD algorithm (Section 2.3.1)

For $T = 10^6$, the histogram converges to ρ and as a result, the gain approximation matches well with the true value. Although, the approximation is not tight everywhere, the accuracy is typically better at values of x for which $\rho(x)$ is large. All three choices of basis functions perform nearly similar. Polynomials weighted by the component densities ρ_i is a better choice because it ensures that the gain decays at large values of x.

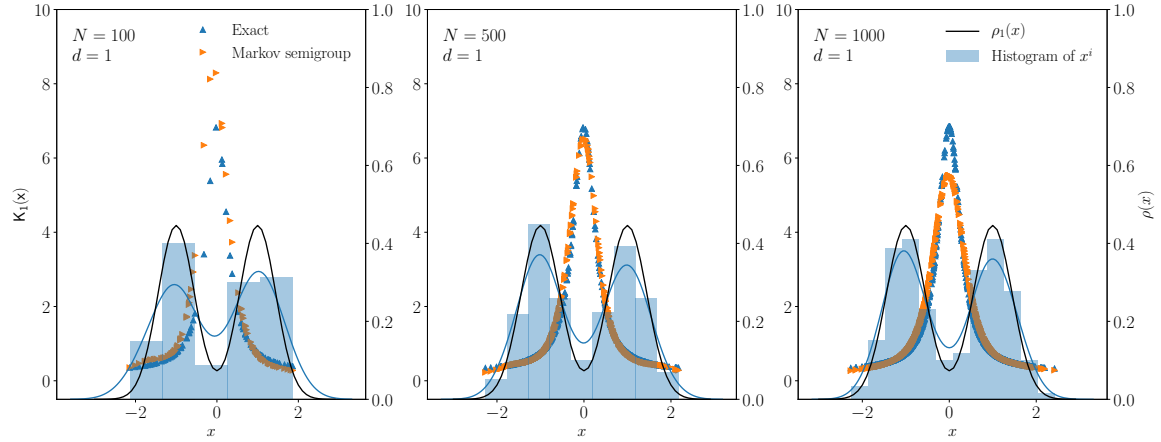
Fig. 4-7 compares the performance of the various basis-free approaches for the same density ρ for $N = 100, 500, 1000$. The particles $\{x^i\}_1^N$ distributed according to ρ can be thought of as being available from the FPF. The parameter ε was set to 0.1 in the



A



B



C

Figure 4-7. Gain function approximations using basis-free approaches - A) ∇-LSTD-RKHS-Opt and ∇-LSTD-Simple with $2N$ and N parameters respectively [26], B) ∇-LSTD-RKHS-OM algorithm [89], C) Markov semigroup approximation [84].

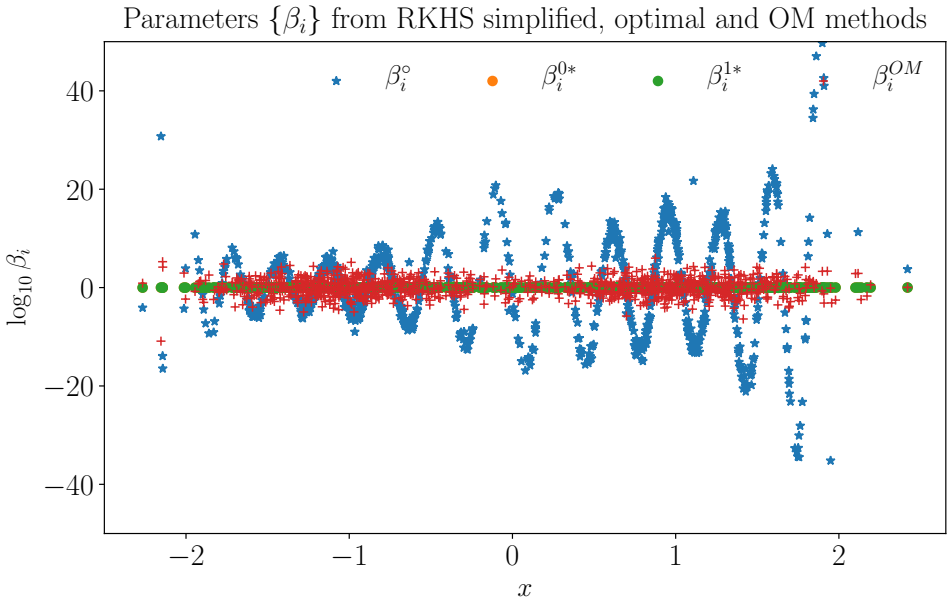


Figure 4-8. Magnitudes of $\{\beta_i\}$ obtained from A) ∇-LSTD-RKHS-Optimal, B) ∇-LSTD-RKHS-Simple and C) ∇-LSTD-RKHS-OM algorithms

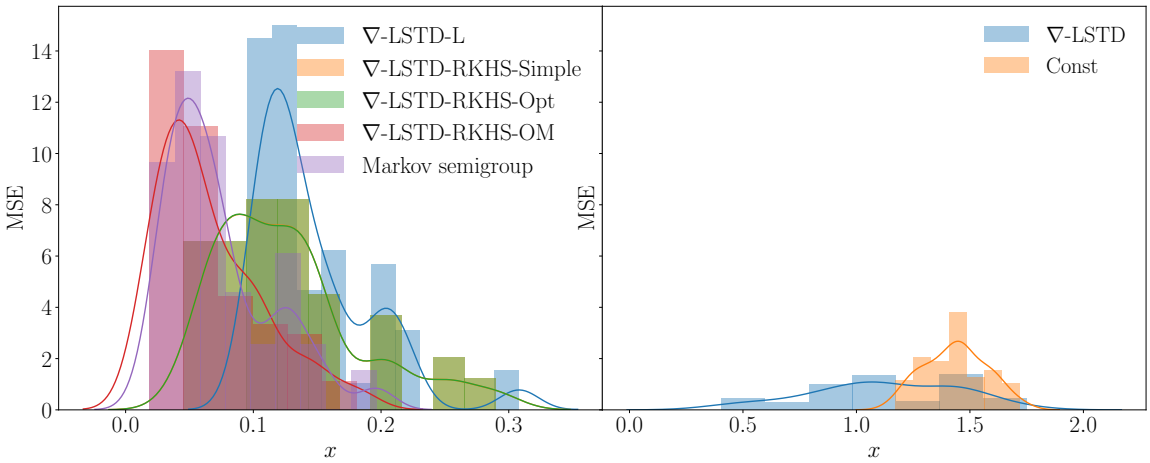


Figure 4-9. Histograms of MSE obtained using various methods for $d = 1, N = 1000$ over 100 independent trials.

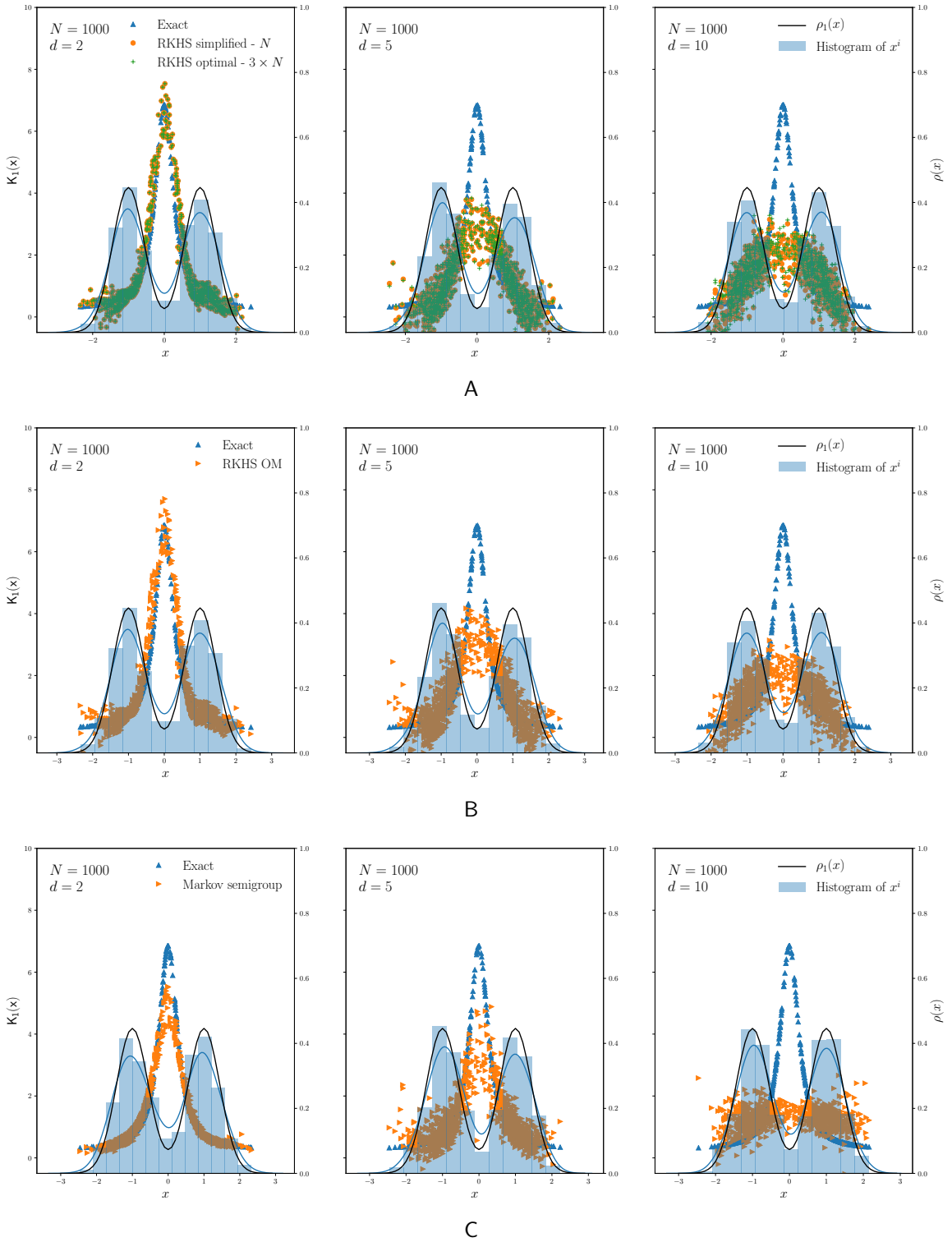
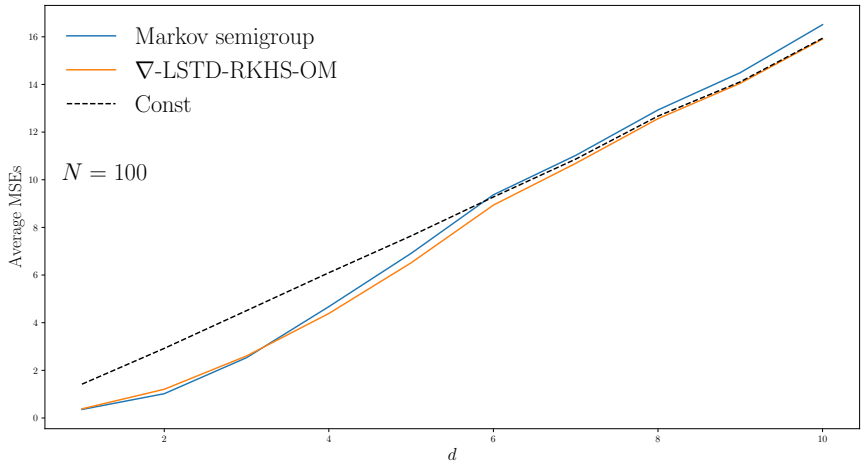
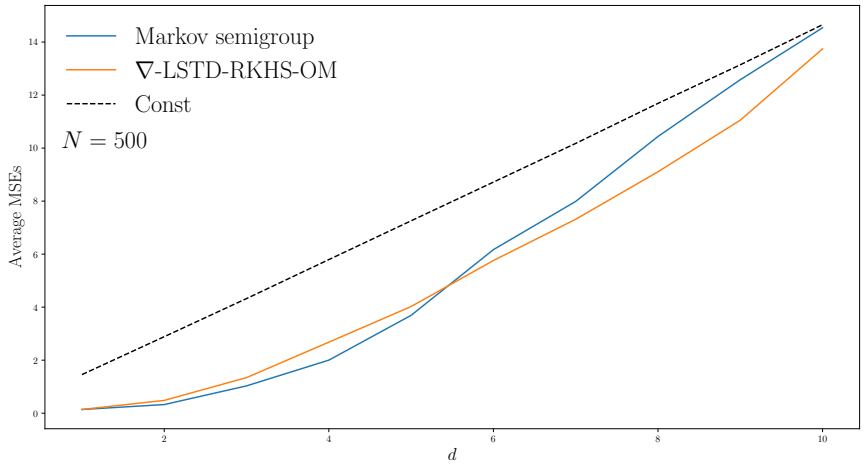


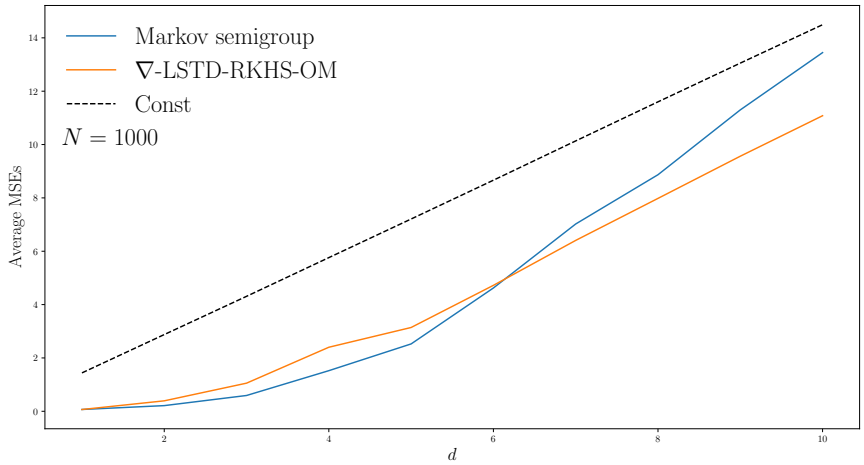
Figure 4-10. Gain function approximations A) ∇-LSTD-RKHS-Opt and ∇-LSTD-RKHS-Simple with $2N$ and N parameters respectively [26], B) ∇-LSTD-RKHS-OM method [89], C) Markov semigroup approximation [84].



A



B



C

Figure 4-11. Gain function approximations A) RKHS optimal and RKHS simplified with $2N$ and N parameters respectively [26], B) RKHS-OM method [89], C) Markov semigroup approximation [84].

Markov semigroup approximation method (the best value found in [17]). Fig. 4-6 shows the approximations obtained using the same weighted polynomial basis, but using the Langevin-specific ∇-LSTD algorithm in Section 2.3.1. The same seed was used for random number generation in each of the implementations. It is important to recall that these methods do not require simulating the Langevin SDE. It can be seen from Figures 4-4, 4-6 and 4-7 that these methods show superior performance over those requiring the SDE simulation. Also, the number of particles required for a good approximation of the gain reduces dramatically from $T = 10^6$ to $N = 500$. Moreover, they make intelligent use of the FPF particles, whereas the other methods derive no useful information from them.

Another surprising observation in Fig. 4-7 A is that the optimal RKHS solution with $2N$ parameters and the reduced complexity solution with N parameters produce an identical gain estimate. This holds true even for higher dimensional examples with $d \leq 5$ (See Fig. 4-11). The RKHS-OM solution is better, especially at particle locations closer to the boundaries. Fig. 4-8 compares the magnitudes of the optimal parameter estimates obtained from the three algorithms. Parameters β_i° corresponding to the reduced complexity solution tend to take higher values than β_i^* or β_i^{OM} and are susceptible to numerical instabilities. RKHS-OM algorithm is the preferred choice, as its computational complexity is comparable to the reduced complexity solution, but is much more stable numerically.

The mean square error in the approximation is computed as:

$$\mathcal{E} := \frac{1}{N} \sum_{i=1}^{N} \|\mathsf{K}(x^i) - \hat{\mathsf{K}}(x^i)\|^2, \tag{4-47}$$

where $\hat{\mathsf{K}}$ denotes the approximate gain. Fig. 4-9 displays the histograms of the MSE obtained for $N = 1000$ for 100 independent trials. As expected, the RKHS-OM and the Markov semigroup approximation methods have the lowest approximation error. All the methods however, produce significant improvement over the constant gain.

Higher dimensional example

It is of interest to test how the gain approximation methods scale with the system dimension d. We considered higher dimensional examples with $1 < d \leq 10$. The d dimensional probability density is composed as a product of d independent 2-component Gaussian mixtures along each dimension. Let ρ_i be the density in the i^{th} dimension and let ρ be defined as

$$\rho(x) = \prod_{i=1}^{d} \rho_i(x_i), \tag{4-48}$$

where $x := [x_1, x_2, \cdots, x_d] \in \mathbb{R}^d$. Taking $\log$ on both sides, the potential function is expressed as a sum:

$$U(x) = \sum_{i=1}^{d} U_i(x_i) \tag{4-49}$$

where $U = -\log(\rho)$. Then,

$$\nabla U = \Big[\frac{\partial U_1}{\partial x_1}, \frac{\partial U_2}{\partial x_2}, \cdots, \frac{\partial U_d}{\partial x_d}\Big] \tag{4-50}$$

The FPF gain K consists of d components and can be written in terms of the solution to Poisson's equation h as:

$$\begin{aligned} \mathsf{K} &= \Big[\frac{\partial h}{\partial x_1}, \frac{\partial h}{\partial x_2}, \cdots \frac{\partial h}{\partial x_d}\Big] \\ &= \Big[\mathsf{K}_1, \mathsf{K}_2, \cdots, \mathsf{K}_d\Big], \end{aligned} \tag{4-51}$$

and,

$$\Delta h = \sum_{i=1}^{d} \frac{\partial^2 h}{\partial x_i^2} = \sum_{i=1}^{d} \frac{\partial \mathsf{K}_i}{\partial x_i} \tag{4-52}$$

Suppose, it is possible to decompose the observation function in the form, $c(x) = c_1(x_1) + c_2(x_2) + \cdots + c_d(x_d)$, then the Poisson's equation to obtain the FPF gain function becomes:

$$-\sum_{i=1}^{d} \frac{\partial U_i}{\partial x_i} \mathsf{K}_i(x_i) + \sum_{i=1}^{d} \frac{\partial \mathsf{K}_i(x_i)}{\partial x_i} = -\sum_{i=1}^{d} \tilde{c}_i(x_i) \tag{4-53}$$

This can be split into d independent Poisson's equations as follows:

$$-\frac{\partial U_i}{\partial x_i} \mathsf{K}_i(x_i) + \frac{\partial \mathsf{K}_i(x_i)}{\partial x_i} = -\tilde{c}_i(x_i) \qquad 1 \leq i \leq d. \tag{4-54}$$

In particular, we consider $c(x) = C^{\intercal} x$, where $C = \mathbf{1}$.

Fig. 4-11 compares the performance of the various methods for $d = 2, 5, 10$ for $N = 1000$. The performance degrades with d as expected. This is partly because $N = 1000$ is insufficient for larger values of d.

4.7.4 Nonlinear Oscillator

We consider next the nonlinear oscillator example introduced in [1]. The state evolves on the unit circle, and the observations are nonlinear:

$$\begin{aligned} \mathsf{d}\vartheta &= \omega \mathsf{d}t + \sigma_B \mathsf{d}B_t \mod 2\pi, \\ \mathsf{d}Z_t &= c(\vartheta)\mathsf{d}t + \sigma_W \mathsf{d}W_t \end{aligned}$$

The parameter ω is the mean angular velocity, and $\boldsymbol{B}$ and $\boldsymbol{W}$ are mutually independent standard Brownian motions. The observation function is $c(\vartheta) = \frac{1}{2}[1 + \cos(\vartheta)]$. Nonlinear oscillators have important applications including neuroscience.

The feedback particle filter for this model is given by :

$$\mathsf{d}\vartheta_t^i = \omega \mathsf{d}t + \sigma_B \mathsf{d}B_t^i + \mathsf{K}(\vartheta_t^i) \circ \mathsf{d}I^i(t) \mod 2\pi\,,$$

with $\mathsf{d}I^i(t) = \mathsf{d}Z_t - \frac{1}{2}(c(\vartheta_t^i) + \hat{c}_t)\mathsf{d}t$. The gain function $\mathsf{K}(\vartheta, t)$ at each instant t is obtained as the solution to (2-6):

$$-U'(\vartheta)\mathsf{K}(\vartheta) + \mathsf{K}'(\vartheta) = -\frac{\tilde{c}(\vartheta)}{\sigma_W^2}$$

The conditional density $\rho(\vartheta, t)$ at any instant is modeled as a mixture of von Mises densities:

$$\rho(\vartheta) = \sum_{i=1}^{m} w_i \rho_i(\vartheta)$$

Each component of the mixture $\rho_i(\vartheta)$ is given as follows:

$$\rho_i(\vartheta, t) = \beta^{-1} \exp\big(\kappa_i \cos(\vartheta - \mu_i)\big),$$

where β is a normalizing constant, and μ_i is the mean of the density. This reduces to a uniform density for $\kappa_i = 0$, and the variance vanishes as $\kappa_i \to \infty$ [87]. By choosing a family of a mixture of von Mises densities, it is possible to model any form of circular density from

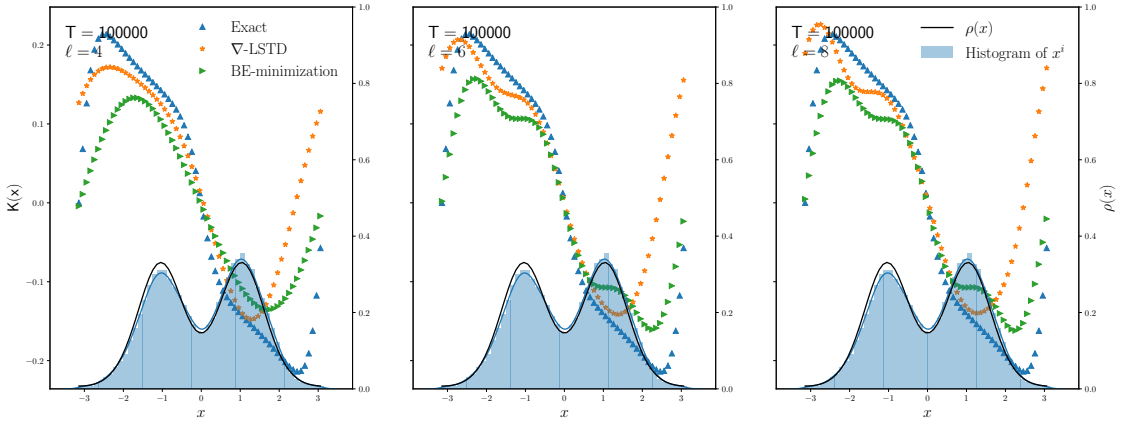


Figure 4-12. Comparison of ∇-LSTD learning with Bellman error minimization for $4, 6$ and 8 dimensional Fourier basis for a nonlinear oscillator model

uniform to a bimodal Gaussian mixture. Poisson's equation can be numerically solved in the scalar case for this mixture density $\rho(\vartheta)$, so that the gain function K is numerically computed.

Linear parameterization

We consider a mixture of von Mises densities with the following parameters: $\mu_1 = -\pi/3$, $\mu_2 = \pi/3$, $\kappa_1 = \kappa_2 = 3$, $w_1 = 0.5$, $w_2 = 0.5$.

The ∇-LSTD learning algorithm is applied to this nonlinear oscillator problem. A linearly parameterized family using sines and cosines is the chosen basis. This is a reasonable choice because for a uniform density, the gain $\mathsf{K} = -\frac{\sin\vartheta}{2\sigma_W^2}$.

For comparison, we also considered an approximation computed by minimizing the $L^2(\rho)-$norm of the Bellman error:

$$\min_\theta \|\mathcal{E}(\theta)\|_{L^2}^2 = \min_\theta \langle \mathcal{D}h^\theta + \tilde{c}\rangle_{L^2} = \min_\theta \|\mathcal{D}h^\theta + \tilde{c}\|_{L^2}^2 \tag{4-55}$$

This minimization problem can be solved using Monte-Carlo methods.

Fig. 4-12 compares the performance of the ∇-LSTD learning algorithm and the Bellman error minimization algorithm with the numerically computed exact gain function for basis functions of dimensions 4, 6 and 8. It can be observed that ∇-LSTD learning gives a better approximation than the BE minimization algorithm in regions of high values of $\rho(\vartheta)$.

4.7.5 Filtering Experiments

The examples surveyed in this section are the following:

- A parameter estimation problem with linear observations and bimodal prior
- A nonlinear multidimensional ship dynamics model.

4.7.6 Parameter Estimation

Anand:needs more work in terms of better plots

A simple example where the state $X_t \in \mathbb{R}$ remains at its initial value is considered first. State-observation model:

$$\mathsf{d}X_t = 0, \qquad X_0 \sim \rho_0$$

$$\mathsf{d}Z_t = X_t \, \mathsf{d}t + \sigma_W \, \mathsf{d}W_t,$$

where the measurement noise standard deviation is $\sigma_W = 1$. In this simple example, the filtering problem reduces to parameter estimation. It is observed in [71] that conventional particle filters are not well suited for such parameter estimation problems, where there is no state dynamics or process noise.

Online gain estimation

To compute the exact gain using (4-21), a smooth approximation to the posterior density ρ_t is required for all t. This was done by assuming that the particles are generated from a 3-component Gaussian mixture density. Based on this model, the parameters for the density were obtained using the EM algorithm.

The constant gain approximation FPF version was compared previously in [86] with the extended Kalman filter (EKF) and the sequential importance resampling (SIR) particle filter. The prior density for the EKF was chosen to be a Gaussian with matching mean and covariance to the actual bimodal prior. The SIS-PF was initialized with 1000 particles drawn i.i.d from ρ_0.

Many variants of SIS-PF were tested in these experiments. Particle degeneracy was observed in implementations without resampling. Resampling at regular intervals was implemented to address this, but numerical issues persisted due to lack of diversity of particles.

The comparisons shown here are for SIS-PF with periodic resampling at every third time interval.

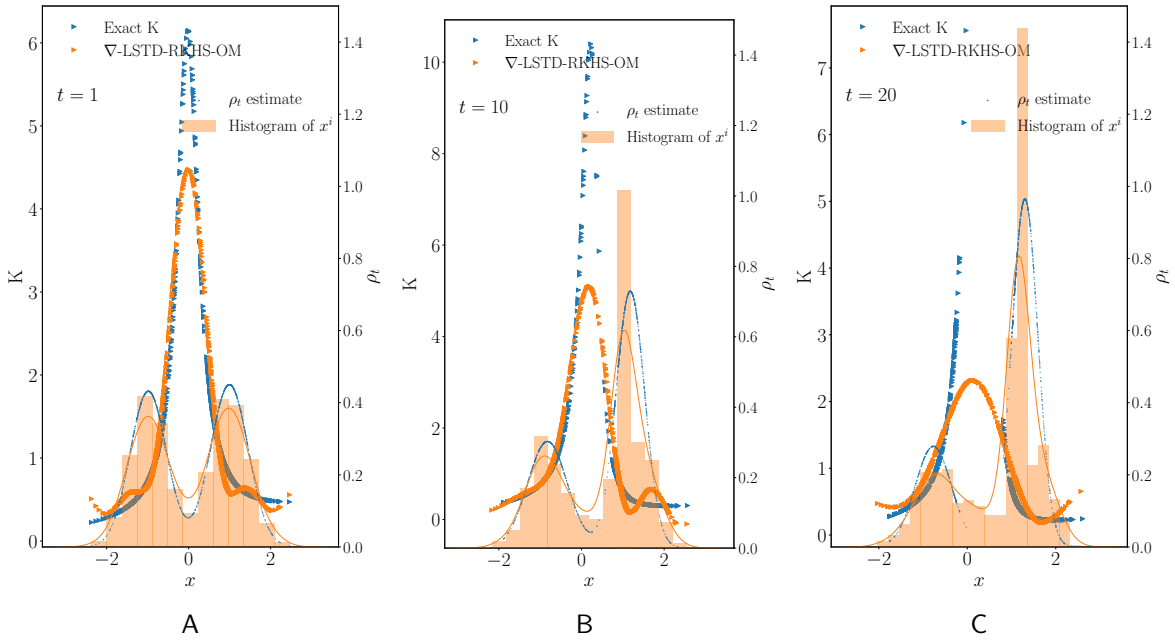


Figure 4-13. Gain approximations and posterior estimates at $t = 1, 10$ and 20 respectively using i) exact computation ii) ∇-LSTD-RKHS-OM method

Fig. 4-15 displays the trajectories of the state estimates (obtained as the conditional mean) in a typical run. The RKHS-OM implementations perform better than the other algorithms in most experiments. Although the RKHS-OM estimates show fluctuations in the beginning, they tend to closely track the FPF estimates obtained using the exact gain later on. The constant gain FPF implementation shows a slower response and its estimates match the EKF estimates towards the end. The SIS-PF suffers from particle degeneracy midway through the simulation, and hence estimates show almost no change towards the end.

The conditional mean is a crude indicator of success. The goal of this paper is to estimate the entire posterior density for each t.

Fig. **??** shows the estimates of the posterior density ρ_t obtained at $t = 1$ for each implementation. In the FPF versions run using the exact gain and RKHS-OM methods, ρ_t evolves to a unimodal density. Both estimates have a sharp spike close to the state value

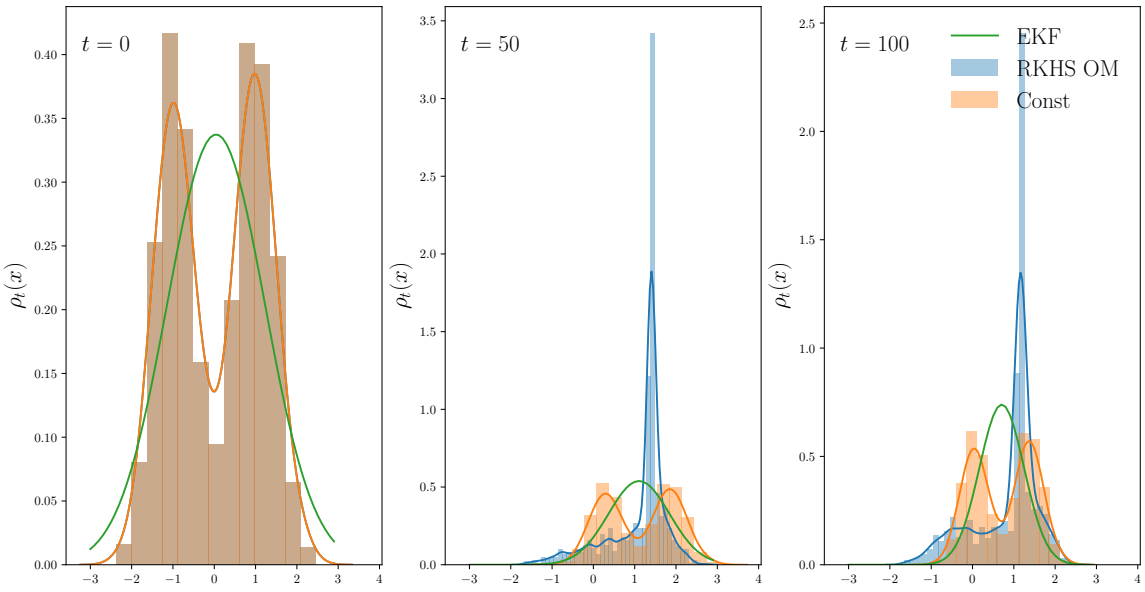


Figure 4-14. Posterior estimates $\rho_t^{(N)}$ using EKF, FPF with constant gain and FPF with ∇-LSTD-RKHS-OM gain approximations at i) $t = 0$,ii) $t = 50$, iii) $t = 100$

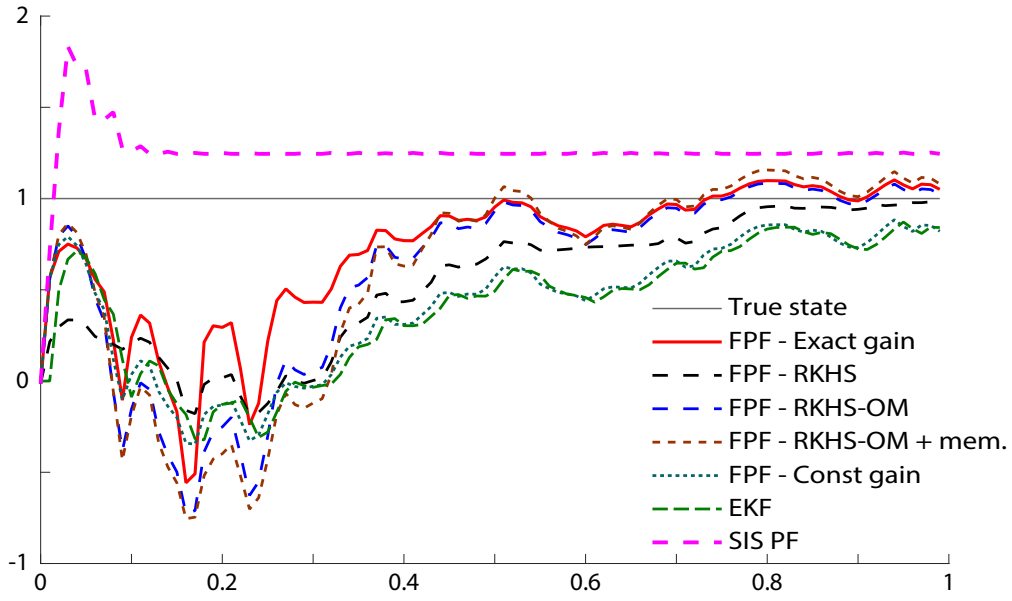


Figure 4-15. State estimate trajectories from the various filters

$X_t \equiv 1$. The histogram for the constant gain FPF shows bimodal behavior at $t = 1$. The state covariances of the EKF and the constant gain implementation are similar and much larger than the others. In conclusion, the RKHS methods provide significantly better posterior estimates.

4.7.7 Ship dynamics example

The performance of the FPF was tested for a nonlinear multidimensional dynamical system. This example, originally described in [8], has been tested with the constant gain implementation of the FPF in [86].

The model (4-56) underlying this example describes the motion of a ship in two dimensions. It has a constant radial and angular velocity when the ship is within a certain distance from the origin. Outside of this region, a restoring force pushes it back towards the origin. The two dimensional state process is modeled by the SDE

$$\begin{aligned} \mathsf{d}X_{t,1} &= -X_{t,2}\,\mathsf{d}t + a_1(X_{t,1}, X_{t,2})\,\mathsf{d}t + \sigma_1\,\mathsf{d}B_{t,1} \\ \mathsf{d}X_{t,2} &= X_{t,1}\,\mathsf{d}t + a_2(X_{t,1}, X_{t,2})\,\mathsf{d}t + \sigma_2\,\mathsf{d}B_{t,2} \end{aligned} \tag{4-56}$$

where $\{B_{t,1}\}$ and $\{B_{t,2}\}$ are independent standard Brownian motions, σ_1 and σ_2 are constants,

$$\begin{aligned} a_i(x) &:= \varsigma \frac{x_i}{|x|^2} - \Theta \frac{x_i}{|x|} \mathbf{1}_{(\varrho,\infty)}(|x|), \\ x &= [x_1\ x_2]^\intercal \in \mathbb{R}^2, \quad i = 1, 2, \end{aligned}$$

where $|x| = \sqrt{x_1^2 + x_2^2}$, and $\mathbf{1}_{(\varrho,\infty)}$ denotes the indicator function for the set (ϱ, ∞). The parameter values are chosen to be $\varsigma = 2, \Theta = 50, \varrho = 9$ and $\sigma_1 = \sigma_2 = 0.4$ as in [8, 86].

A discrete time observation model is used in [8], which may be regarded as an approximation of the SDE

$$\mathsf{d}Z_t = c(X_t)\mathsf{d}t + \mathsf{d}W_t, \qquad c(x) = \arctan(x_2/x_1)\,.$$

In the experiments that follow, the SDE is approximated using the same sampling time $\delta = 0.05$ that is used in [8], so the setup is unchanged. The value $\sigma_W = 2.5$ was used, which is consistent with prior work. As always, the state disturbance, measurement noise, and FPF initial conditions were taken to be mutually independent.

As in [8, 86], 100 independent trials were performed for an overall run time of $8.25s$ each, with each trial driven by independent process and measurement noise. For each of the 100 trials, the initial state x_0 was set to $(0.5\,, -0.5)^T$.

Two different priors were compared in these experiments, determined by matrices $\Sigma_1 = I_{2\times 2}$ and $\Sigma_2 = 5I_{2\times 2}$: filters were initialized with Gaussian prior $N(x_0, \Sigma_i)$ for $i = 1, 2$. For the particle-based methods, 500 particles were drawn independently from the prior density. The sequential importance sampling (SIS) particle filter was implemented with deterministic resampling at every third time instant. For the RKHS based methods, $\lambda = 10^{-1}$ and $\varepsilon = 2$ were chosen. Exact computation of the gain requires solving a PDE in two dimensions and is omitted here.

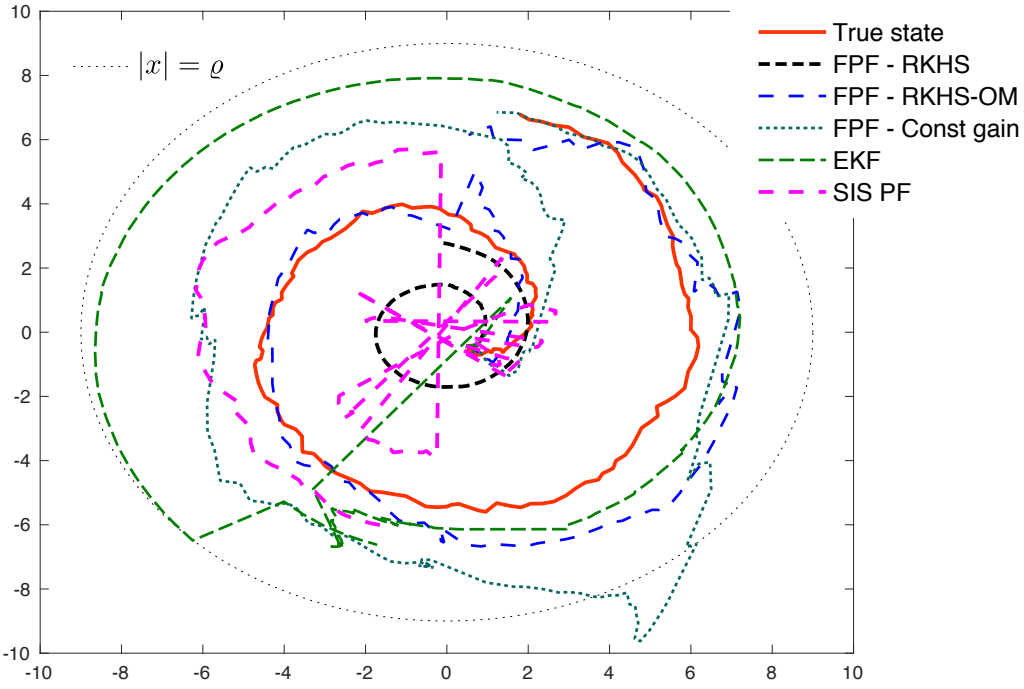


Figure 4-16. Ship trajectory estimates in phase space.

Fig. 4-16 displays a typical state trajectory along with the estimates obtained using the each of the five filters using Σ_2. The EKF estimates are erratic as reported in [8], and are often seen to diverge. The particle methods exhibit superior performance.

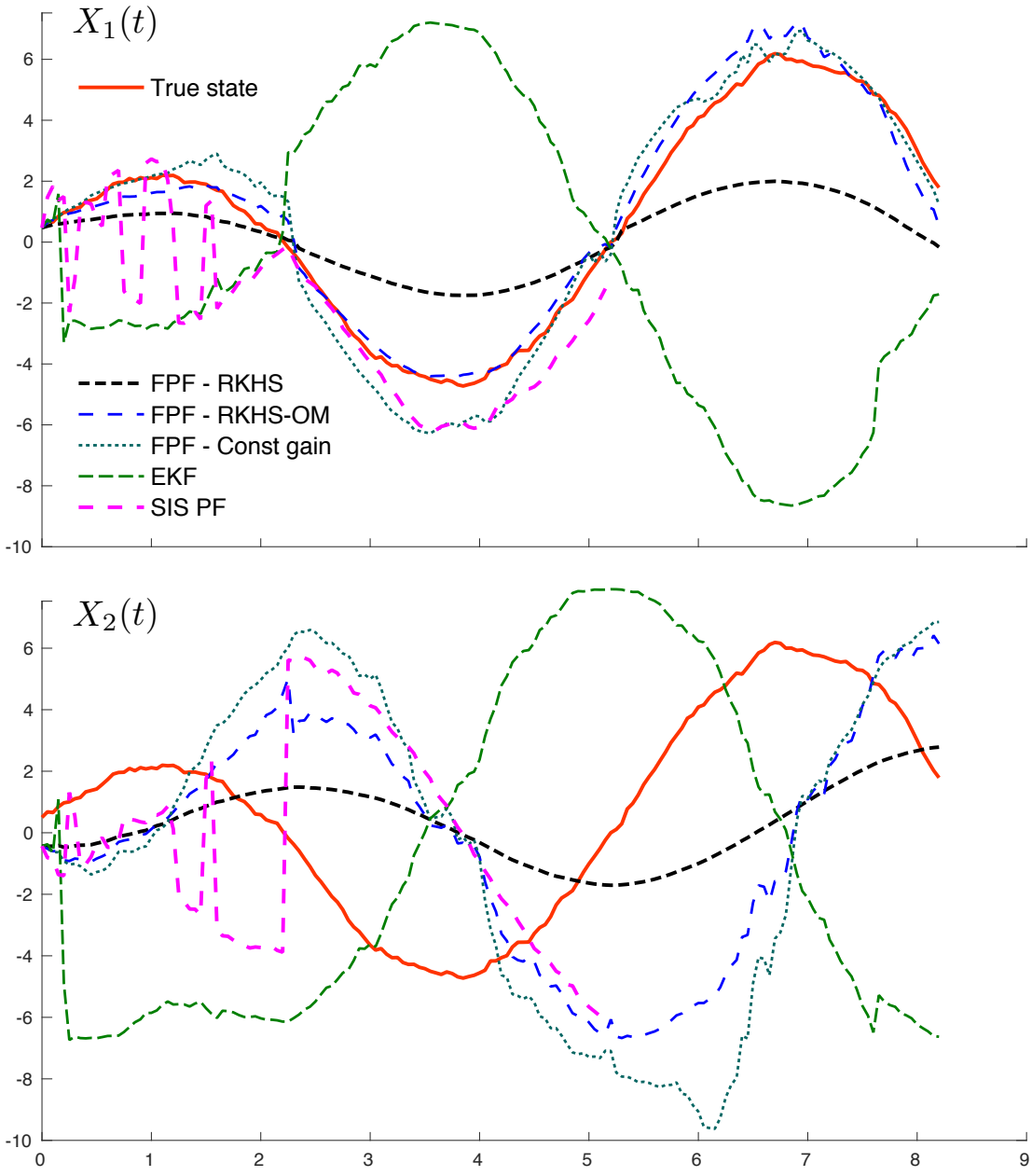


Figure 4-17. State estimates X_1 and X_2 from the various filters

The phase-space plot masks the significant delay observed in these experiments. Individual state estimates for X_1 and X_2 are shown in Fig. 4.7.7. Estimates of X_1 closely follow the actual state trajectory, whereas estimates of the second state show significant phase lag. The FPF-RKHS method without the optimal mean is seen to perform poorly. The SIS PF estimates show large fluctuations. The FPF RKHS-OM method provides the most reliable estimates.

The root-mean-square-error (RMSE) over 100 independent trials was used to compare the approaches:

$$\text{RMSE} := \frac{1}{100}\frac{1}{165}\sum_{j=1}^{100}\sum_{n=1}^{165}|X^j(n\delta) - \hat{X}^j(n\delta)|$$

where X^j, $j = 1, \cdots, 100$ represents the signal trajectory, $X^j(n\delta)$ is the true state at time $n\delta$, $\hat{X}^j(n\delta)$ is the state estimate obtained as the conditional mean and 165 is the total number of time instances for each simulation. Table 4-1 summarizes the results obtained.

The FPF-RKHS-OM provides the lowest RMSE value among the four filters. The EKF performs the worst as the estimates tend to diverge from the true state trajectory in many runs.

Another metric discussed in [8] is the number of times each filter "loses its track". This is obtained by setting a maximum tolerance value for the norm of the estimation error at each instant. If at any instant, the filter estimates produce larger than this tolerance limit, it is considered as having "lost the track". This tolerance limit was set to 10 in these experiments. The EKF lost track over 93 times in the 100 trials, whereas FPF-RKHS-OM lost track only 4 times.

Table 4-1. Comparison of various filters

Type of filter	Σ_1	Σ_2	Lost track (Σ_2)
FPF RKHS-OM	0.9023	1.6254	4 times
FPF RKHS mem.	0.9162	1.9408	7 times
FPF const. gain	1.3060	2.3231	14 times
SIR PF	3.1481	4.2648	57 times
EKF	6.5203	18.441	93 times

4.8 Conclusions

In this Chapter, we introduced the feedback particle filter as an alternate approximation method to nonlinear filtering. Gain approximation is an integral part of this approach to nonlinear filtering. A number of ∇-LSTD learning based gain approximation algorithms were discussed. The generic ∇-LSTD learning scheme is statistically and computationally inefficient as it requires the additional steps of smoothing the posterior density and simulating the Langevin SDE. The ∇-LSTD-Langevin algorithm solves some of these issues. The problem of appropriate basis selection which was addressed in Chapter 3 was illustrated through numerical examples. The RKHS based ∇-LSTD algorithm offers a basis-independent solution and is seen to perform much better than any finite parameterization. This algorithm is also seen to scale easily to higher dimensional systems. The problem of hyperparameter selection in the RKHS methods was also addressed via the RKHS OM method, where the additional information about the constant gain approximation reduces the sensitivity of the solution to the parameters ε and λ. For online estimation of the gain in filtering problems, an enhancement that takes into account that the posterior densities and FPF gain evolve with known dynamics was also presented.

Numerical examples demonstrate the superiority of the RKHS based algorithms over the finite parameterization methods and they are on par with the Markov semigroup approximation method that was developed in parallel research. It is hypothesized that the RKHS based methods offer a more robust and computationally efficient solution, as it avoids the need for successive approximation techniques. In the filtering examples, the FPF implementations with RKHS based gain approximation perform much better than EKF or the constant gain FPF. It also overcomes the resampling and particle degeneracy issues associated with the conventional particle filters.

There remain many open questions. Alternatives to the EM algorithm like kernel density estimation must be explored for density approximation based on particles, and it will be valuable to apply variance reduction techniques to improve the ∇-TD learning algorithms. In

terms of mathematical challenges, the most important open problem is robustness of the filter to modeling error, including the impact of an imperfect FPF gain.

CHAPTER 5
APPLICATION TO MARKOV CHAIN MONTE CARLO ALGORITHMS

One of the interesting applications of ∇-LSTD learning is in the context of Markov chain Monte Carlo(MCMC) simulations. In this chapter, a basic introduction to MCMC algorithms with a particular emphasis on Langevin diffusion and its discrete variants is provided first in Section 5.1. We then define asymptotic variance as a measure of convergence and motivate how ∇-LSTD learning is perfectly suited to obtain MCMC estimates with minimal asymptotic variance. In Section 5.2, we introduce the very popular Metropolis-Hastings (MH) algorithm and present an extension of the asymptotic variance reduction methods to the class of reversible Markov chains in Section 5.3. A majority of the existing research that aims to improve the convergence of MCMC estimates focus on minimizing the sample variance, as it is easier to implement. In Section 5.4, through a motivating example, we prove why this is not the correct objective for MCMC. In recent research, it has been found that under certain assumptions, the convergence rates of ULA and RWM are related and the same asymptotic variance reduction techniques can be applied to RWM [90]. Applications to numerical examples involving logistic regression (logit) and probit models are discussed in Section 5.5.

Anand:Need to rephrase after reading [90]

5.1 Langevin Diffusion for MCMC

In many scenarios, it is of interest to compute the expectation of a function c with respect to a target distribution ρ:

$$\eta = \int c(x)\rho(x)\mathrm{d}x\,. \tag{5-1}$$

Typically, computing integrals of the form (5-1) analytically is difficult when ρ is in high dimensions and Markov chain Monte Carlo (MCMC) methods are used instead. MCMC provides numerical algorithms to obtain estimates of η. It can be approximated using time averages of the following form:

$$\eta_N = \frac{1}{N}\sum_{n=0}^{N-1} c(\Phi_n), \tag{5-2}$$

in which $\mathbf{\Phi}$ is a Markov chain whose steady-state distribution has density ρ [33, 34]. In this chapter, we initially focus on a formulation in continuous time and then extend the idea to discrete-time MCMC algorithms.

The Langevin Diffusion, introduced in Section 2.1 is the grandmother of all MCMC algorithms. It forms the basis for various other popular MCMC algorithms such as the unadjusted Langevin algorithm (ULA) and Metropolis-adjusted Langevin algorithm (MALA). The diffusion obeys the stochastic differential equation in (2-1):

$$\mathrm{d}\Phi_t = -\nabla U(\Phi_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t, \tag{5-3}$$

where $\boldsymbol{W} = \{W_t : t \geq 0\}$ is a standard Brownian motion on $\mathbb{R}^d$. Under general conditions, this diffusion is reversible, with unique invariant density $\rho = e^{-U+\Lambda}$, where Λ is a normalizing constant so that ρ integrates to unity. For a large T, the mean η can be approximated as,

$$\eta \approx \eta_T := \frac{1}{T}\int_0^T c(\Phi_t)\,dt. \tag{5-4}$$

To compare the different MCMC algorithms it is convenient to consider an asymptotic setting. Under general conditions, the estimates will obey a Central Limit Theorem (CLT) of the form:

$$\sqrt{T}\tilde{\eta}_T \xrightarrow{d} N(0, \sigma_\infty^2) \tag{5-5}$$

where $\tilde{\eta}_T = \eta_T - \eta$, and the convergence is in distribution. Under further mild assumptions, the variance of the Gaussian limit is given by the so-called asymptotic variance:

$$\sigma_\infty^2 = \lim_{T\to\infty} \mathsf{E}\left[\left(\frac{1}{\sqrt{T}}\int_0^T (c(\Phi_t) - \eta)dt\right)^2\right] \tag{5-6}$$

For example, these conclusions hold for a Markov chain that is V-uniformly ergodic, provided $c^2 \in L_\infty^V$ [15, 33]. It has been shown in [15, 33] that the asymptotic variance σ_∞^2 has the following general representation in terms of the solution to Poisson's equation h (2-6),

Anand:c^2 or c?

$$\sigma_\infty^2 = 2\langle h, \tilde{c}\rangle_{L^2}. \tag{5-7}$$

The representation (5-7) is valid for any diffusion that is V-uniformly ergodic. For the special case of the Langevin diffusion, by application of Prop. 2.1, $\sigma_\infty^2 = 2\|\nabla h\|_{L^2}^2$.

In practice, discretized versions of the equation based on Euler-Mauryama scheme are used (2-2):

$$\Phi_n = \Phi_{n-1} - \nabla U(\Phi_{n-1})\delta_n + \sqrt{2\delta_{n-1}} W_{n-1},$$

where $\{\delta_n\}_{n\geq 1}$ is a sequence of step sizes and $\{W_n\}_{n\geq 1}$ is a sequence of i.i.d. standard Gaussian random variables. This implementation is called the unadjusted Langevin algorithm (ULA) [91]. In the numerical experiments, the step size δ is chosen to be a constant, independent of n.

The primary goal in the design of MCMC algorithms is the faster convergence of the Markov chain to its invariant distribution (i.e. the target distribution). The asymptotic variance is a measure of this convergence and hence, it is ideal to minimize σ_∞^2. Several approaches have been proposed to improve the convergence rate including constructing an irreversible Markov chain with the same invariant density [41, 92], using an optimal scaling parameter [93] etc. Most of these approaches alter the transition kernel of the Markov chain and hence, do not work well on samples already obtained. In the remainder of this paper, we restrict our discussion to post-hoc schemes for reversible Markov chains, i.e. samples from an ergodic Markov chain are available and variance reduction is achieved by post-processing. These scheme require no changes to the sampling methodology.

Control variates introduced in [13, 94–96] are zero-mean terms, which when added to the estimator can produce a significant reduction in the asymptotic variance without adding bias. One of the advantages of this scheme is that they work post-hoc, and are independent of the MCMC algorithm used.

Anand:a few more words on original application

Dellaportas et al. in [16] have employed control variates for particular examples of reversible Markov chains like the Gibbs sampler. More recently, Brosse et al. in [97] show that the asymptotic variances of ULA, MALA and RWM are close to the asymptotic variance of the Langevin diffusion and construct control variates that work for all the three methods. In this

dissertation, we borrow heavily from both the above and demonstrate that using RKHS based ∇-LSTD learning algorithms, the variance reduction is improved many-fold.

The basic idea is described here. Let $\psi\colon \mathbb{R}^d \to \mathbb{R}^\ell$ denote a C^2 function, regarded as a family of ℓ basis functions, and $\theta \in \mathbb{R}^\ell$ denote the parameters,

$$c^\theta := c + \underbrace{\mathcal{D}h^\theta}_{\text{Control variate}}, \qquad \text{where} \quad h^\theta := \sum_{i=1}^{\ell} \theta_i \psi_i = \theta^\intercal \psi \tag{5-8}$$

and $\mathcal{D}$ denotes the differential generator of the Langevin diffusion (2-3). Under the assumptions imposed it will follow that the steady-state means of c and c^θ coincide for any $\theta \in \mathbb{R}^\ell$, so that we obtain a family of asymptotically unbiased estimators:

Anand:Do we need to show $Ph = h$?

$$\eta_T^\theta = \frac{1}{T} \int_0^T c^\theta(\Phi_t)\, dt \tag{5-9}$$

It is then of interest to find the parameter with minimal asymptotic variance:

$$\theta^* = \arg\min_\theta (\sigma_\infty^\theta)^2 \tag{5-10}$$

Here, $(\sigma_\infty^\theta)^2$ corresponds to the asymptotic variance of the new estimate (5-9), i.e.

$$\sqrt{T}(\eta_T^\theta - \eta) \xrightarrow{d} N(0, (\sigma_\infty^\theta)^2).$$

Proposition 5.1. *Suppose that c and each ψ_i lie in $L_\infty^{\sqrt{V}}$. Then, equation (5-9) defines an asymptotically unbiased estimate of η. Its asymptotic variance is*

$$(\sigma_\infty^\theta)^2 = 2\|\nabla h - \nabla h^\theta\|_{L^2}^2 \tag{5-11}$$

Proof. Applying the differential generator $\mathcal{D}$ on $h - h^\theta$, we have

$$\begin{aligned} \mathcal{D}(h - h^\theta) &= -\tilde{c} - \mathcal{D}h^\theta \\ &= -c + \eta + c - c^\theta \qquad \text{From (5-8)} \\ &= -c^\theta + \eta := -\tilde{c}^\theta \end{aligned} \tag{5-12}$$

Hence $h - h^\theta$ is the solution to Poisson's equation with forcing function c^θ. Analogous to (5-7), the asymptotic variance for the new estimator in (5-9) is represented as $(\sigma_\infty^\theta)^2 = 2\langle h - h^\theta, \tilde{c}^\theta \rangle_{L^2}$, and applying Prop. 2.1 gives (5-11). ■

An important observtion here is that the asymptotic variance $(\sigma_\infty^\theta)^2$ for the new estimator (5-8), with control variates (5-11) has the same form as (2-35), and hence any of the variants of the ∇-LSTD algorithm can be applied. Numerical examples using these algorithms are discussed in Section 5.5.

5.2 Metropolis-Hastings Algorithm

Although Langevin diffusion offers a very intuitive sampling algorithm, it requires the computation of the gradient of the potential function U, which may be expensive in high dimensions. Hence, other discrete time algorithms are preferred. Metropolis-Hastings algorithm is a popular MCMC technique to simulate multivariate distributions. It was originally proposed by Metropolis in 1953 and extended to a more general case by Hastings in 1970 [12]. The only requirement in Metropolis-Hastings is that it should be possible to compute the value of a function f proportional to the target density ρ. The function f could potentially be the unnormalized form of ρ.

The algorithm requires the choice of a simple "proposal" distribution g, from which samples can be drawn easily. The function $g(x, y)$ gives the conditional distribution for the next sample y, given the current sample x. The algorithm can be summarized in the following steps:

- Choose an arbitrary point $\Phi_0 = x$ to be the first sample.
- Choose a proposal distribution g. Pick a new candidate sample point $x' \sim g(x, .)$.
- Compute the acceptance ratio $\alpha(x, x') = \min\left[\frac{f(x')\, g(x', x)}{f(x)\, g(x, x')}, 1\right]$.
- With probability α, accept x' and set $\Phi_1 = x'$; if rejected set $\Phi_1 = x$.
- Repeat above for $\Phi_2, \ldots \Phi_N$ for a fixed N.

The transition kernel P_{mh} for a Metropolis-Hastings chain can be written as:

$$P_{mh}(x, dy) := g(x, y)\alpha(x, y)dy + \left[1 - \int_{\mathsf{X}} g(x, y)\alpha(x, y)dy\right] \delta_x(dy), \tag{5-13}$$

and with the acceptance ratio α as defined, the reversible Markov chain with kernel P_{mh} has its invariant density as ρ.

A special case of this algorithm is obtained if g is chosen to be a Gaussian centered at x with δ as the variance parameter. This also simplifies the expression for α as $g(x, x') = g(x', x)$. This variant is known as the random walk Metropolis (RWM) algorithm. For a small value of δ, it has been shown by Brosse et al. in [90] that the following relation between P_{mh} and the differential generator $\mathcal{D}$ of the Langevin diffusion holds,

Anand:variance parameter is equivalent to the time step in ULA?

$$\lim_{\delta \to 0} \delta^{-1}(P_{mh} - I) = \mathcal{D}$$

Thus for small values of δ, the control variates can be obtained as $\delta \mathcal{D} h^{\theta}$. But this approximation is not valid for larger step-sizes and hence this method is practically limited in use.

Anand:Do we need this statement? Or is there a hack for larger δ?

5.3 Control Variates for a Reversible Markov Chain

In this section, we extend the idea of control variates to the more general case of reversible Markov chains, including RWM. The main difficulty is that Prop. 2.1 is absent here. However using reversibility, we obtain an expression for the optimal parameter values. A derivation similar to this is presented in [16]. Consider a discrete time Markov chain with a transition kernel P. Let h denote the solution to the Poisson's equation of this Markov chain,

$$(P - I)h = -\tilde{c},$$

where I refers to the identity operator. In discrete time, the asymptotic variance has the following representation in terms of h,

$$\sigma_{\infty}^2 = 2\langle h, \tilde{c} \rangle_{L^2} - \langle \tilde{c}, \tilde{c} \rangle_{L^2}. \tag{5-14}$$

A linearly parameterized class of functions h^θ is defined as in (5-8) and a new function c^θ with the control variates is defined as

$$c^\theta := c + (P - I)h^\theta \tag{5-15}$$

Denoting $\tilde{h}^\theta := h - h^\theta$, a simple extension of (5-12) is obtained:

$$(P - I)\tilde{h}^\theta = -\tilde{c}^\theta. \tag{5-16}$$

Following from (5-14), the asymptotic variance corresponding to this new estimator c^θ has the following representation:

$$\begin{aligned}(\sigma_\infty^\theta)^2 :=& 2\langle \tilde{h}^\theta, \tilde{c}^\theta \rangle_{L^2} - \langle \tilde{c}^\theta, \tilde{c}^\theta \rangle_{L^2} \\ =& -2\langle \tilde{h}^\theta, (P - I)\tilde{h}^\theta \rangle_{L^2} - \langle (P - I)\tilde{h}^\theta, (P - I)\tilde{h}^\theta \rangle_{L^2} \\ =& \langle \tilde{h}^\theta, \tilde{h}^\theta \rangle_{L^2} - \langle P\tilde{h}^\theta, P\tilde{h}^\theta \rangle_{L^2}\end{aligned} \tag{5-17}$$

To eliminate the unknown term h from (5-17), the self-adjoint property of the transition kernel of a reversible Markov chain is used (Prop. 5.2).

Proposition 5.2. *For a reversible Markov chain the following holds:*

$$\langle Pf, g \rangle_{L^2} = \langle f, Pg \rangle_{L^2}, \qquad f, g \in L^2(\rho)$$

Proof. For a reversible Markov chain, the detailed balance equations hold with respect to its unique invariant measure ρ:

$$\rho(dx)P(x, dy) = \rho(dy)P(y, dx). \tag{5-18}$$

From the definition of the transition kernel P,

$$\begin{aligned}Pf(x) &= \mathsf{E}[f(\Phi_{k+1})|\Phi_k = x] \\ &= \int P(x, dy) f(y)\end{aligned} \tag{5-19}$$

Using the definition of the inner product,

$$\begin{aligned}
\langle Pf, g\rangle_{L^2} &:= \int (Pf(x))g(x)\rho(dx) \\
&= \int \left(\int P(x, dy)f(y)\right) g(x)\rho(dx) \qquad \text{(From (5-19))} \\
&= \int \left(\int P(y, dx)g(x)\right) f(y)\rho(dy) \qquad \text{(Applying (5-18))} \\
&= \int (Pg(y))f(y)\rho(dy) \\
&= \langle f, Pg\rangle_{L^2}.
\end{aligned}$$

■

Proposition 5.3. *The optimal parameter vector $\theta^* \in \mathbb{R}^\ell$ that minimizes (5-21) is given by*

$$\begin{aligned}
\theta^* &:= M^{-1}b \qquad \textit{where,} \\
M &:= \langle \psi, \psi\rangle_{L^2} - \langle P\psi, P\psi\rangle_{L^2} \\
b &:= \langle \tilde{c}, \psi\rangle_{L^2} + \langle \tilde{c}, P\psi\rangle_{L^2}.
\end{aligned} \tag{5-20}$$

The optimal control variate is then given by,

$$(P - I)h^\theta = \theta^{*\intercal}(P\psi) - \theta^{*\intercal}\psi.$$

Proof. Expanding the expression for $(\sigma^\theta_\infty)^2$ (5-17) gives,

$$\begin{aligned}
(\sigma^\theta_\infty)^2 &= \langle \tilde{h}^\theta, \tilde{h}^\theta\rangle_{L^2} - \langle P\tilde{h}^\theta, P\tilde{h}^\theta\rangle_{L^2} \\
&= \langle h, h\rangle_{L^2} - 2\langle h, h^\theta\rangle_{L^2} + \langle h^\theta, h^\theta\rangle_{L^2} - \langle Ph, Ph\rangle_{L^2} + 2\langle Ph, Ph^\theta\rangle_{L^2} - \langle Ph^\theta, Ph^\theta\rangle_{L^2}
\end{aligned} \tag{5-21}$$

Substituting for h^θ in (5-17) and applying the first order necessary conditions for optimality by taking the derivative with respect to θ,

$$\begin{aligned}
0 &= -\langle h, \psi\rangle_{L^2} + \theta^{*\intercal}\langle \psi, \psi\rangle_{L^2} + \langle Ph, P\psi\rangle_{L^2} - \theta^{*\intercal}\langle P\psi, P\psi\rangle_{L^2} \\
&= -\langle h, \psi\rangle_{L^2} + \theta^{*\intercal}\langle \psi, \psi\rangle_{L^2} + \langle Ph, P\psi\rangle_{L^2} - \theta^{*\intercal}\langle P\psi, P\psi\rangle_{L^2} + \langle Ph, \psi\rangle_{L^2} - \langle Ph, \psi\rangle_{L^2} \\
&= -\langle \tilde{c}, \psi\rangle_{L^2} + \langle Ph, (P - I)\psi\rangle_{L^2} + \theta^{*\intercal}\Big(\langle \psi, \psi\rangle_{L^2} - \langle P\psi, P\psi\rangle_{L^2}\Big) \qquad \text{(Using } (P - I)h = -\tilde{c}\text{)}
\end{aligned} \tag{5-22}$$

Applying Prop. 5.2,

$$\langle Ph, (P-I)\psi\rangle_{L^2} = \langle (P-I)h, P\psi\rangle_{L^2} = -\langle \tilde{c}, P\psi\rangle_{L^2}$$

Denoting M and b as

$$M := \langle \psi, \psi\rangle_{L^2} - \langle P\psi, P\psi\rangle_{L^2}, \qquad b := \langle \tilde{c}, \psi\rangle_{L^2} + \langle \tilde{c}, P\psi\rangle_{L^2}.$$

completes the proof. ■

It can be shown that M is a symmetric positive definite matrix. The Monte Carlo estimates of M are designed to respect this constraint:

$$\begin{aligned}
\hat{M}_N &:= \frac{1}{N}\sum_{n=0}^{N-1}\Big(\psi(\Phi_n)\cdot\psi^{\intercal}(\Phi_n) - \psi(\Phi_{n+1})\cdot\psi^{\intercal}(\Phi_{n+1})\Big) \\
M_N &:= \frac{1}{2}(\hat{M}_N + \hat{M}_N^{\intercal}) \\
b_N &:= \frac{1}{N}\sum_{n=0}^{N-1}\tilde{c}(\Phi_n)\Big(\psi(\Phi_n) + \psi(\Phi_{n+1})\Big).
\end{aligned}$$

Similar results for optimal control variates for reversible Markov chains have been obtained in [16]. Although optimal parameter weight estimates $\theta_N = M_N^{-1}b_N$ are easy to obtain, the main difficulty is to compute the term $P\psi$ which is part of the control variate defined in Prop. 5.3. In [16], numerical examples are presented for a class of conjugate random-scan Gibbs samplers for which $P\psi$ is explicitly computable in closed form. In this dissertation, we substitute for $(P-I)\psi$ with $\delta\mathcal{D}\psi$, which is easy to compute. Recent results in [90] justifies this substitution for small values of δ.

5.4 Sample Variance v Asymptotic Variance

The basic idea behind using control variates to minimize asymptotic variance of the MCMC estimates was discussed in Section 5.1. It requires approximating the value function h, which is non-trivial in most cases. A large body of prior research [98, 99] attempts to achieve better estimates by solving an easier problem. In [99], a parameterized family of control variates is considered as in this dissertation. However, the optimal parameter is chosen to

minimize the sample variance rather than the asymptotic variance. This is well-motivated only if the samples are i.i.d. The unbiased estimator in [99] is based on a function c^ϑ of the same form as (5-8):

$$c^\vartheta := c + \mathcal{D}h^\vartheta, \qquad \text{where} \quad h^\vartheta = \sum_{i=1}^{\ell} \vartheta_i \psi_i = \vartheta^{\intercal} \psi.$$

The quantity minimized is the sample variance, i.e. $(\sigma^\vartheta)^2 = \mathsf{E}[(\tilde{c}^\vartheta)^2] = \langle \tilde{c}^\vartheta, \tilde{c}^\vartheta \rangle_{L^2}$, where $\tilde{c}^\vartheta = c^\vartheta - \eta$ resulting in

$$\vartheta^* = \arg\min_{\vartheta \in \mathbb{R}^\ell} (\sigma^\vartheta)^2 = M^{-1} b \tag{5-23}$$

where $M = \langle \mathcal{D}\psi, \mathcal{D}\psi \rangle_{L^2}$ and $b = -\langle \tilde{c}, \mathcal{D}\psi \rangle_{L^2}$. This is termed the zero-variance (ZV) algorithm in [99]. It is interesting to observe that the variance minimization algorithm is equivalent to minimizing the Bellman error technique presented in (4-55).

The important question to be addressed is: does minimizing the sample variance minimize (perhaps approximately) minimize the asymptotic variance? Using a simple numerical example, it can be shown that this is not always the case.

Consider X $\equiv \mathbb{R}$, $c(x) \equiv x$, ρ is a univariate 2-component Gaussian mixture density defined in (4-43) and shown by the shaded region in Fig. 5-2. Two sets of basis functions are chosen for ψ (similar to those defined in Section 4.7.3):

(i) a family of polynomials of the form $\psi_n(x) = x^n$ for $1 \le n \le \ell$.

(ii) a family of polynomial functions weighted by the Gaussian density components ρ_i,

$$\{\psi_n(x) : x \in \mathbb{R},\ 1 \le n \le \ell\} = \{x^k \rho_i(x) : 1 \le k \le \ell/2,\ i = 1, 2\}, \tag{5-24}$$

where $\rho_i \sim N(\mu_i, \sigma_i^2)$. The rationale behind this choice of basis functions is that the contribution of each ψ_n is local to a particular mode. This allows us to approximate h with good accuracy in the region around μ_1 and μ_2, where most of the samples of the Langevin diffusion are concentrated.

Anand:Wrong text in C

The resulting asymptotic variances for the parameters, θ^* (5-10) and ϑ^* (5-23) are compared against the basis dimension ℓ. The optimality gap for $c(x) = x$ shown in parts A and

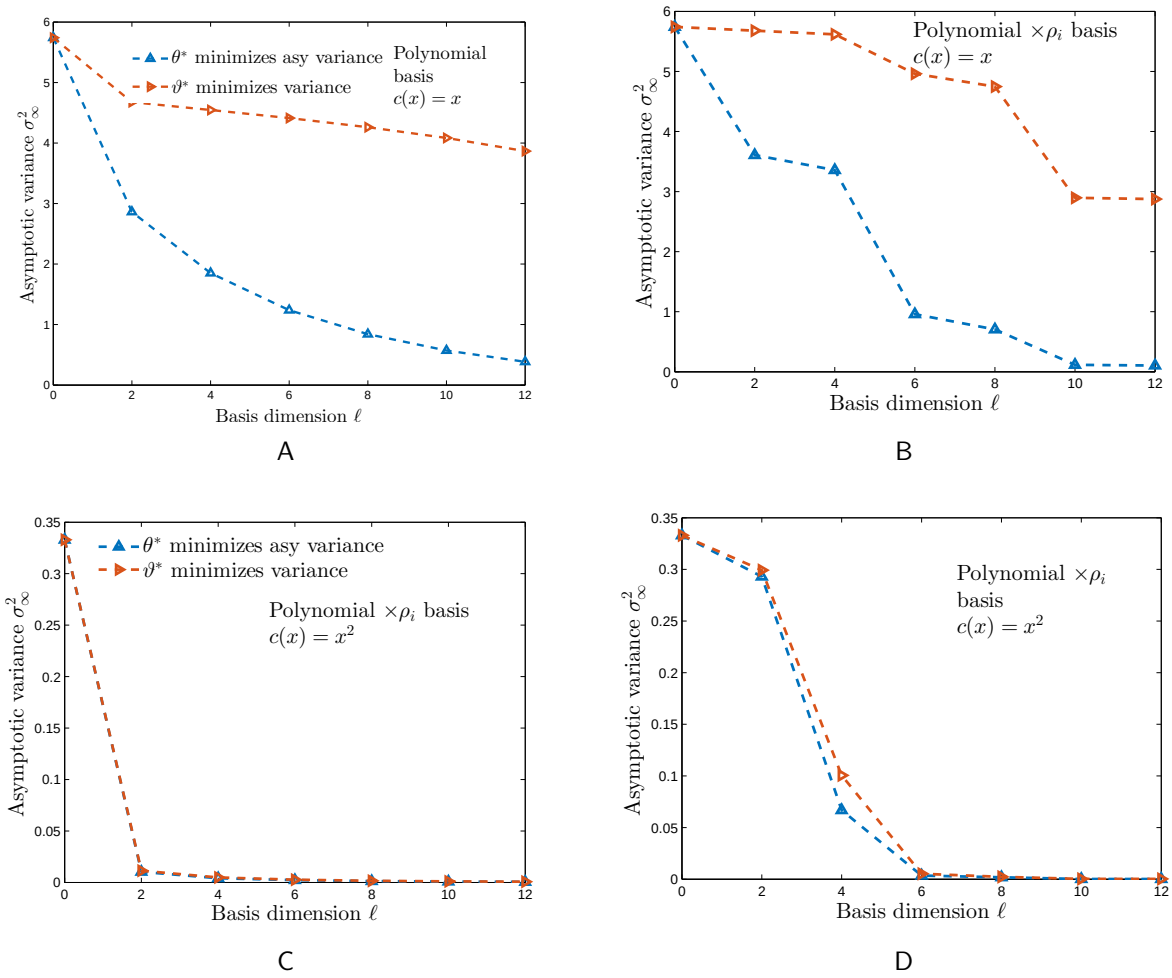


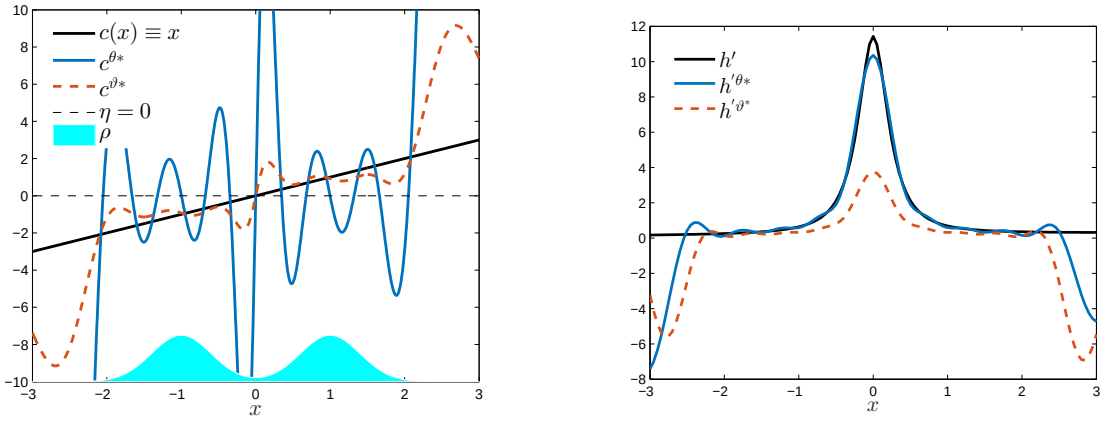
Figure 5-1. Comparison of $(\sigma^{\theta^*}_\infty)^2$ and $(\sigma^{\vartheta^*}_\infty)^2$ for $\theta^*, \vartheta^* \in \mathbb{R}^\ell$, $0 \le \ell \le 12$ and $c(x) = x, x^2$, for i) a polynomial basis (in A and C) and ii) a weighted polynomial basis (in B and D)

B of Fig. 5-1 is considerable. The discrepancy is clear from a close look at the two functions c^{θ^*} and c^{ϑ^*} that are plotted in Fig. 5-2. The key difference is that the "local mean" of c^{θ^*} is nearly η in each well of the density ρ (shaded region of Fig. 5-2):

$$\int_{-\infty}^{0} c^{\theta^*}(x)\rho(x)\,dx \approx \int_{0}^{\infty} c^{\theta^*}(x)\rho(x)\,dx \approx \eta = 0.$$

This helps reduce the asymptotic variance for this slowly mixing diffusion. The function c^{ϑ^*} does not share this property and produces biased estimates of η in each well. The approximations h'^{ϑ^*} and h'^{θ^*} are plotted along with the exact solution h' obtained analytically

in Fig. 5-2. The function h'^{ϑ^*} is a poor approximation to h', and that results in much higher asymptotic variance. In summary, to minimize the asymptotic variance, it is important to approximate h' quite well and the ∇-LSTD objective function aims to minimize the approximation error in the mean-square sense. The same experiment was repeated for $c(x) \equiv \sin x$ with similar results. It is conjectured that this phenomenon is more pronounced in multimodal densities. However, for $c(x) = x^2$, both the methods achieve nearly similar reduction in asymptotic variance.



A B

Figure 5-2. i) Modified estimators using control variates c^{θ^*} and c^{ϑ^*}, ii) Approximations h'^{θ^*} and h'^{ϑ^*} plotted with true gradient h' for $c(x) = x$ with a polynomial $\times \rho_i$ basis.

Examining the autocorrelation functions offers another perspective to verify if the two objectives are equivalent. In the discrete time case, the asymptotic variance of the estimates of η can be written as the infinite sum of all the autocorrelation functions,

$$\begin{aligned} \sigma_\infty^2 &:= \sum_{n=-\infty}^{\infty} R(n), \qquad R(n) = \mathsf{E}[\tilde{c}(\Phi_0)\tilde{c}(\Phi_n)] \\ &= 2\sum_{n=0}^{\infty} R(n) - R(0). \end{aligned} \tag{5-25}$$

The expression for σ_∞^2 in (5-14) can be derived from this infinite summation form. On inspecting the two objectives, it is easy to see that while c^{θ^*} tries to minimize σ_∞^2, c^{ϑ^*} minimizes just the term $R(0)$ in it. This is evident from the the autocorrelation function $R(n)$

in (5-25) plotted in Fig. 5-3. Autocorrelation function corresponding to the three different estimators c, c^{θ^*} and c^{ϑ^*} for $c(x) = x$, are plotted for n upto 100. Although, c^{ϑ^*} has the least correlation value at $n = 0$ (sample variance), it shows a slow decay as n increases, c^{θ^*} shows a much sharper decay, resulting in a much lower overall asymptotic variance. The plot clearly illustrates that minimizing the sample variance is not always equivalent to minimizing the asymptotic variance.

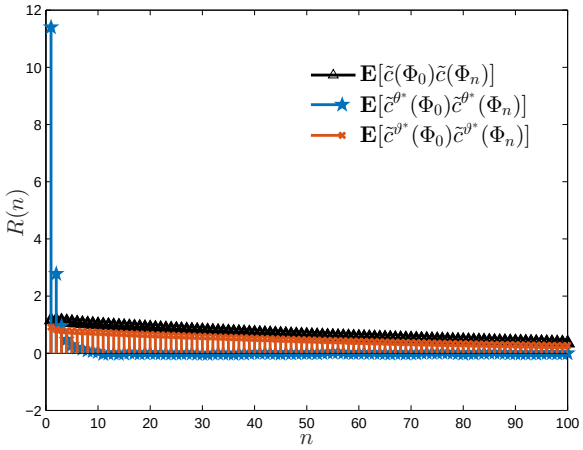


Figure 5-3. Autocorrelation functions $R(n)$ corresponding to the three estimators c, c^{θ^*} and c^{ϑ^*} for $n = 0$ to 100 for ULA with step size $\delta = 0.05$

5.5 Numerical Examples

In this section we present a survey of the numerical experiments performed based on the algorithms presented in this paper. We illustrate the remarkable reduction in asymptotic variance that is achieved by the various ∇-LSTD based algorithms, first for estimating the mean of a simple functions $c(x)$ with respect to a univariate Gaussian mixture target density and then in a maximum-likelihood parameter estimation problem using logistic regression.

5.5.1 ULA and RWM for a Univariate Gaussian Mixture Target Density

In this section, we first look at how the control variates help in reducing the asymptotic variance for both the ULA and RWM algorithms. For a simple demonstration of the idea, we consider the same one dimensional bimodal target density defined as mixture of Gaussians (4-43). This gives a symmetric density ρ as shown by the shaded region in Fig. 5-2. The linear

function $c(x) \equiv x$ is used in the simulation experiments; symmetry of ρ and c being an odd function imply $\eta = 0$.

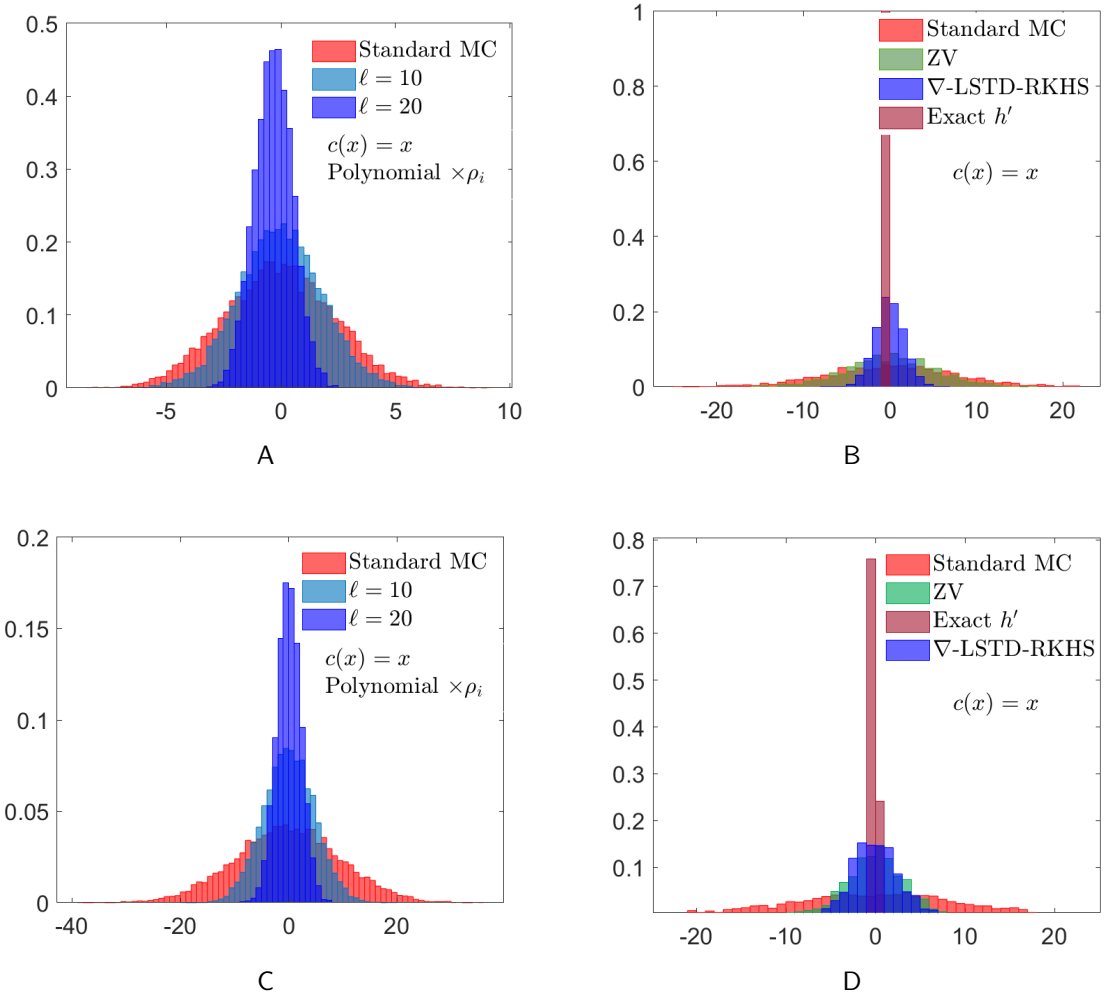


Figure 5-4. Histogram of $\sqrt{N}(\eta_N^i - \eta)$ for the various control variate schemes - A) and B) ULA with $\delta = 0.05$ and $c(x) = x$ with finite basis $\ell = 10, 20$ in A) and other approximation schemes in B), and C) and D) for RWM with $\delta = 0.05$ with finite basis $\ell = 10, 20$ in C) and other approximation schemes in D).

Both the ULA and RWM algorithms were simulated using Euler discretization with a step size of $\delta = 0.05$. Part A in Fig. 5-4 shows the histograms of the estimates obtained over 10^4 independent trials for $\ell = 10, 20$ compared to the standard MC method. A total number of 10^5 samples were used in each trial. The empirical variance values observed on the histograms are very close to the analytical variances, although a small bias is seen in the estimate in some

cases. This bias may be attributed to the Euler-discretization of the Langevin diffusion which alters the invariant distribution as mentioned in [100].

Part B of Fig. 5-4 presents a comparison of the performance of the control variates obtained by the other approximation schemes. The histograms were obtained from 1000 independent trials and the number of samples used in each trial was $N = 10^5$. The exact h' obtained analytically (4-21) is used to illustrate that it produces a near-zero asymptotic variance. The RKHS based ∇-LSTD algorithm produces the lowest asymptotic variance among approximation methods. Compared to the standard estimator, the variance minimization method (ZV) [99] produced nearly the same variance. Markov semigroup approximation method [17] and the constant approximation to h' were also tested, but they produced variances larger the the standard estimator.

The same set of experiments is repeated for the RWM algorithm with the results displayed in parts C and D of Fig. 5-4. The caveat in this case is that, it cannot be proved that the control variate thus computed is optimal for the RWM algorithm, as Prop. 2.1 is specific to the Langevin diffusion. However, significant reduction in the variance is observed in simulation results.

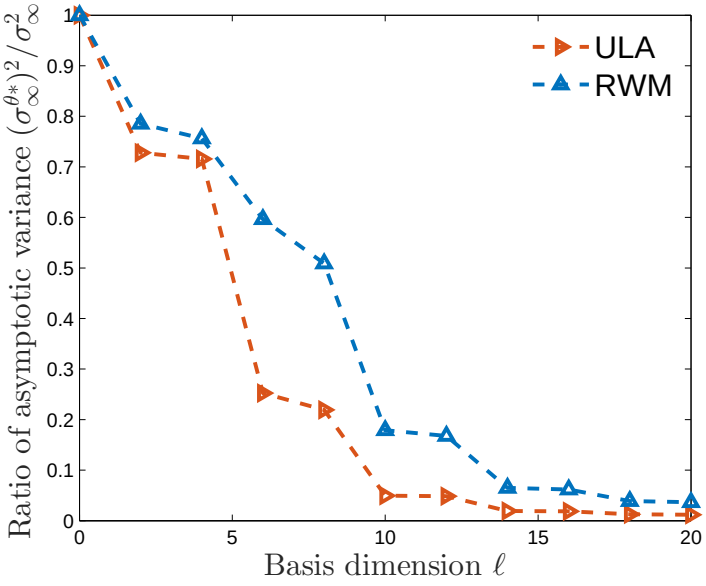


Figure 5-5. Asymptotic variance reduction comparison between ULA and RWM algorithms for $c(x) = x$ and $1 \leq \ell \leq 20$

Fig. 5-5 shows the relative reduction in variance for both the ULA and RWM algorithms. The value corresponding to $\ell = 0$ is normalized to 1 in both the cases. It may be seen that significant variance reduction is achieved for RWM, although at a reduced rate than for ULA.

The algorithm can be applied to a wider class of densities with similar results; we have seen positive results for heavy-tailed densities also.

5.5.2 Logistic Regression - Swiss Bank Notes Example

In practice, MCMC algorithms find wide applications in Bayesian inference problems, where we are interested in estimating the mean with respect to a posterior distribution. In this section, we discuss a particular example of Bayesian logistic regression or the logit model. A detailed description of the problem in the context of the Swiss bank notes example is given in [99] and a brief summary is provided here.

The goal is to learn the regression coefficients that can be used to classify the Swiss bank notes dataset into genuine and counterfeit. Given in the dataset are the measurements for four covariates - the length of the bill, the width of the left and the right edges and the bottom margin width for 200 bank notes of which 100 each are counterfeit and genuine. This is essentially a binary classification problem in a supervised learning setting.

Let $X \in \mathbb{R}^{N_n \times N_d}$ correspond to the covariate measurements of N_n bank notes, with N_d denoting the number of covariates with $N_n = 200$ and $N_d = 4$. Let $\{Y_i \in \{0,1\}, 1 \leq i \leq N_n\}$ correspond to the labels for each note being genuine or counterfeit respectively. Let $\boldsymbol{\Theta} \in \mathbb{R}^{N_d}$ be the regression coefficients that are learned from the given data to obtain a good classifier. We are interested in finding the best estimates for the regression coefficients, i.e. the coefficients that maximize the posterior probability ρ defined as,

$$\begin{aligned}\boldsymbol{\Theta}^* &:= \underset{\Theta \in \mathbb{R}^{N_d}}{\arg\max}\, \rho(\boldsymbol{\Theta} | \{X_i, Y_i\}_{i=1}^{N_n}) \\ &= \underset{\Theta \in \mathbb{R}^{N_d}}{\arg\max} \exp\left(\underbrace{\sum_{i=1}^{N_n} \{Y_i \boldsymbol{\Theta}^T X_i - \log(1 + e^{\boldsymbol{\Theta}^T X_i})\}}_{\text{Likelihood}} - \underbrace{2^{-1} \boldsymbol{\Theta}^T \Sigma^{-1} \boldsymbol{\Theta}}_{\text{prior}} \right),\end{aligned} \tag{5-26}$$

where the parameter vector $\mathbf{\Theta} = [\Theta_1, \ldots, \Theta_{N_d}]$ has a zero-mean Gaussian prior with a covariance matrix Σ.

In problems such as this, the maximum likelihood estimate is rarely available in closed-form and is hard to compute. Instead, we try to compute the Bayesian estimate:

$$\mathbf{\Theta}_{bayes} := \int_{\Theta \in \mathbb{R}^{N_d}} \mathbf{\Theta}\, \rho(\mathbf{\Theta} | \{X_i, Y_i\}_{i=1}^{N_n}) \mathrm{d}\Theta. \tag{5-27}$$

Obtaining the Bayesian estimator (5-27) requires sampling $\mathbf{\Theta}$ values from the posterior distribution and then computing the empirical mean $\hat{\mathbf{\Theta}}$. Samples from this posterior density may be obtained using any reversible MCMC technique, in particular we focus on the RWM algorithm. In the following, it is assumed that N samples generated by the RWM algorithm are available.

The goal as before is to compute the optimal control variates that minimize the asymptotic variance of each of the estimates $\hat{\mathbf{\Theta}}$. Finding the optimal control variates boils down to finding approximate solutions to the Poisson's equations corresponding to each of the four regression coefficients. The equations to be solved are,

$$\mathcal{D} h_k = -\tilde{\Theta}_k = -(\Theta_k - \hat{\Theta}_k), \qquad 1 \leq k \leq N_d. \tag{5-28}$$

The approximation h_k^θ is chosen to belong to a family of linearly parameterized functions as before,

$$h_k^\theta := \theta_k^\intercal \psi = \sum_{j=1}^{\ell} \theta_{kj} \psi_j,$$

where $\psi_j : \mathbb{R}^{N_d} \to \mathbb{R}$ is the chosen set of basis functions and $\theta_{kj} \in \mathbb{R}$ are the parameter values. The optimal parameter values are obtained by minimizing $\|\nabla h_k - \nabla h_k^\theta\|_{L^2}^2$. For a general basis,

the optimal parameter θ_k^* can be derived along the same lines as before,

$$\begin{aligned} \theta_k^* &= M^{-1} b_k, \qquad \text{where,} \\ M &:= \langle \nabla\psi, \nabla\psi \rangle_{L^2} \approx \frac{1}{N} \sum_{i=0}^{N-1} \nabla\psi(\boldsymbol{\Theta}^i) \nabla\psi^T(\boldsymbol{\Theta}^i) \\ b_k &:= \langle \tilde{\Theta}_k, \psi \rangle_{L^2} \approx \frac{1}{N} \sum_{i=0}^{N-1} \tilde{\Theta}_k^i \psi(\boldsymbol{\Theta}^i), \end{aligned} \tag{5-29}$$

where each $\boldsymbol{\Theta}^{\boldsymbol{i}}$ corresponds to the i^{th} sample of the MCMC algorithm

Polynomials of degree 1 and 2 are chosen as the basis functions in [99] and we use the same basis for comparing the algorithm performances. Additionally, we also compare the results against the ∇-LSTD-RKHS algorithm with Gaussian kernels.

Linear polynomial basis

First we choose $\ell = 4$ and ψ_j to be polynomial of degree 1, i.e.

$$\psi^T := [\Theta_1 \quad \Theta_2 \quad \Theta_3 \quad \Theta_4] := \boldsymbol{\Theta}^T \tag{5-30}$$

For this choice of ψ, $\nabla\psi = I$ and $\Delta\psi = 0$. Hence, θ_k^* has the simple expression,

$$\theta_k^* = \frac{1}{N} \sum_{i=0}^{N-1} \tilde{\Theta}_k^i \boldsymbol{\Theta}^i \tag{5-31}$$

The control variate $\mathcal{D}h_k^\theta$ is explicitly computable and is given by,

$$\mathcal{D}h_k^\theta(\boldsymbol{\Theta}^i) = -\nabla \log(\rho(\boldsymbol{\Theta}^i | \{X_j, Y_j\}_{j=1}^N)) \cdot \theta_k^* \qquad \forall k, i$$

and the new estimator of the regression coefficients will be,

$$\bar{\Theta}_k^i := \Theta_k^i + \mathcal{D}h_k^\theta(\boldsymbol{\Theta}^i).$$

The ZV-MCMC algorithm, proposed in [99] minimizes the ordinary variance instead of the asymptotic variance. In general, it is an easier optimization problem to solve, but in this specific example, it is interesting to note that θ_k^* using (5-31) is simpler to compute and the

solution is numerically more stable (as inverting an ill-conditioned matrix is not required) than the ZV method.

Quadratic basis

A quadratic polynomial basis of the form,

$$\psi^T := [\Theta_1,\, \Theta_2,\, \Theta_3,\, \Theta_4,\, \Theta_1^2,\, \Theta_1\Theta_2,\, \Theta_1\Theta_3,\, \Theta_1\Theta_4,\, \Theta_2^2,\, \Theta_2\Theta_3,\, \Theta_2\Theta_4,\, \Theta_3^2,\, \Theta_3\Theta_4,\, \Theta_4^2] \quad (5\text{-}32)$$

is also considered similar to [99]. In this case, $\nabla\psi \in \mathbb{R}^{14\times 4}$ and $\theta_k \in \mathbb{R}^{14}$. The optimal parameter values are given by (5-29).

∇-LSTD-RKHS algorithm

We also investigate the performance of the RKHS based ∇-LSTD learning algorithm for this example. The parameter values of $\lambda = 10^{-7}$ and $\varepsilon = 2$ were found to produce the best results. As it becomes prohibitively expensive to place a Gaussian kernel at each of the RWM samples, they were placed at 200 randomly chosen samples.

Results

Simulations were performed using the Swiss bank note dataset provided in [99]. We performed 1000 independent trials each with 10^5 samples and 10^4 samples used for burn-in. The box plots of the estimates of the four regression coefficients $\boldsymbol{\Theta}$ shown in Fig. 5-6 indicate that significant variance reduction is achieved using all the control variate methods. It may be observed that for a linear basis, both the ZV-L and ∇-LSTD-L learning produce nearly the same asymptotic variance. The ∇-LSTD method is able to produce estimates for the four regression coefficients, whose variance values are $10-65$ times smaller than the standard RWM sampler. Using the quadratic polynomial basis, ZV-Q method outperforms the ∇-LSTD-Q method slightly, a variance reduction factor of $100 - 200$ over the standard estimator is still obtained. The ∇-LSTD-RKHS method produces the best results with variances that are lower by a factor $20 - 50$ over the ZV-Q method. The ∇-LSTD-RKHS method however produced a much larger number of outliers. By a more careful choice of the values for λ, ε and by placing the kernel functions at more well-chosen samples, better results may be obtained.

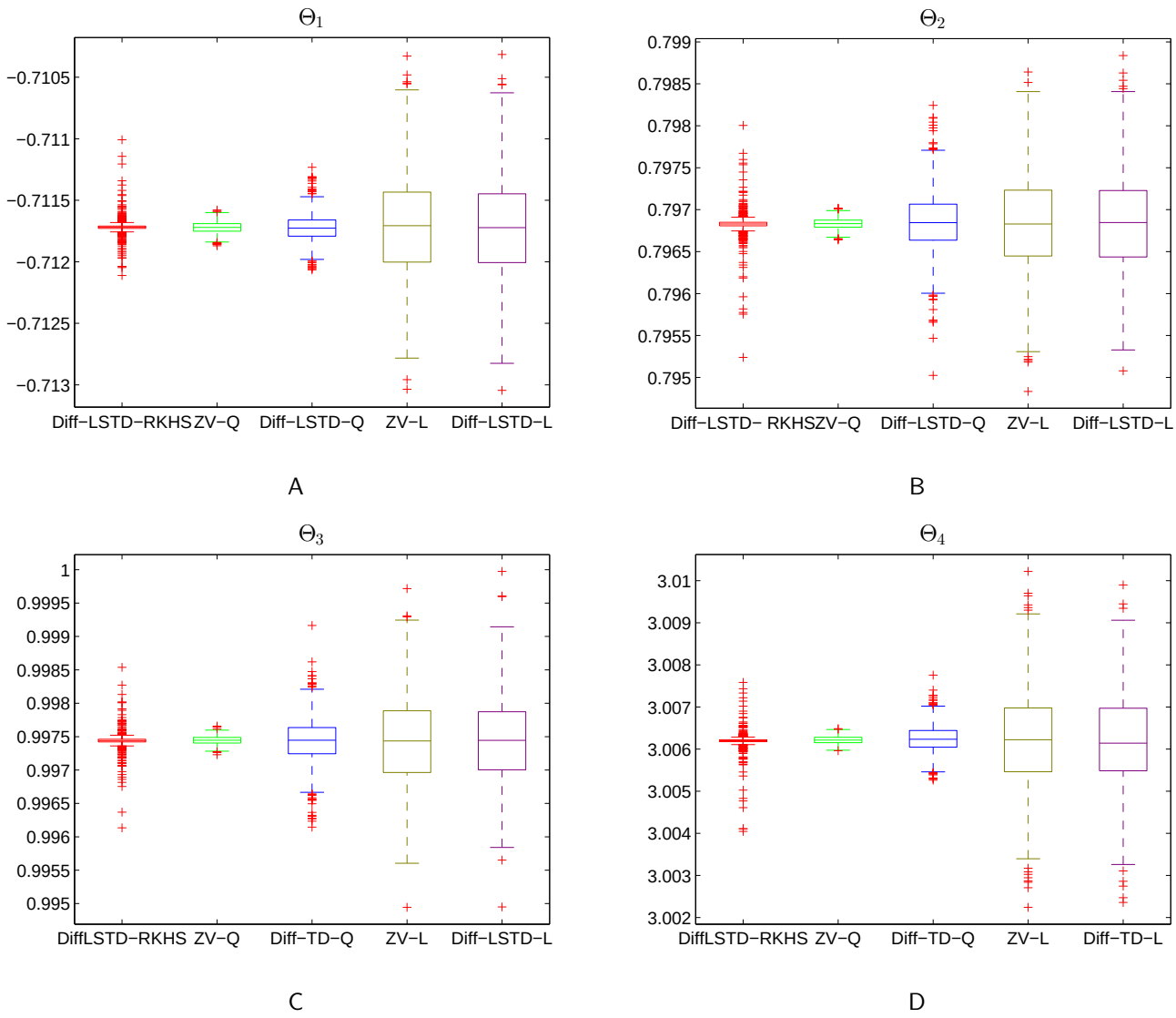


Figure 5-6. Boxplots of estimates of Θ obtained over 1000 trials using linear and quadratic polynomial basis using the ZV-MCMC and ∇-LSTD and the ∇-LSTD-RKHS algorithms).

One might also be interested in the ordinary variance of the samples within a single run. The box plots in Fig. 5-7 compare the variance values of the samples obtained within a run using the ZV-Q method and the ∇-LSTD-RKHS learning method. It may be observed that in spite of having outliers, the mean sample variance using the RKHS method is about one order of magnitude lower than the ZV-Q method.

The same set of experiments were tried out with similar results for the logistic regression example using MALA and ULA sampling. Similar results were also obtained for the probit model Vaso constriction example discussed in [99]. The plots for these simulations are provided in the appendix.

Anand:Should I include those plots?

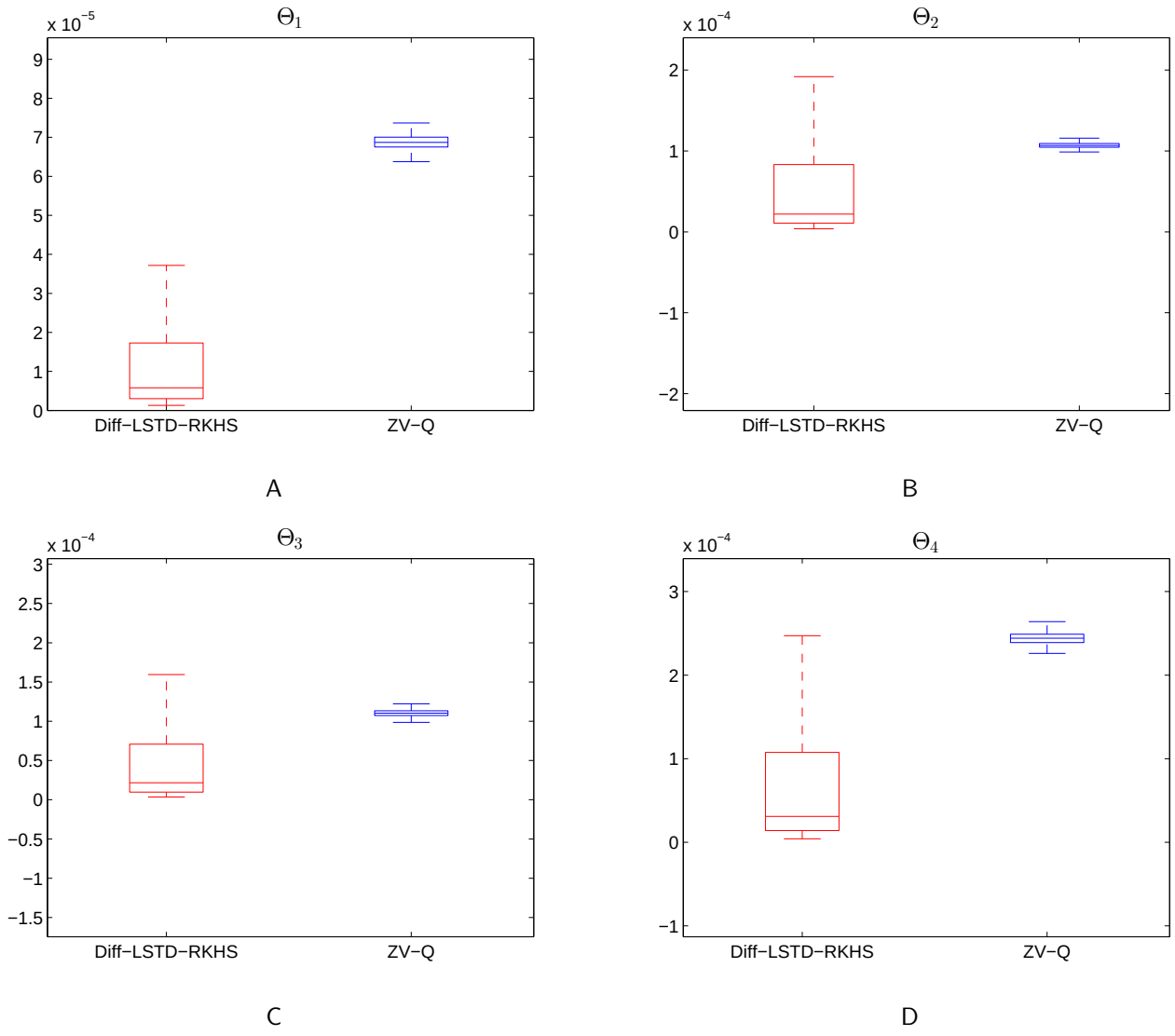


Figure 5-7. Boxplots of the in-trial variances of estimates of $\boldsymbol{\Theta}$ obtained over 1000 trials using the ∇-LSTD-RKHS and ZV with quadratic polynomials

5.6 Conclusions

In this Chapter, we provided a brief discussion on MCMC algorithms, particularly the ULA, which has an associated continuous time Langevin diffusion process. MCMC algorithms

have been widely used in Bayesian inference, as a means of approximating expectations using numerical integration. Asymptotic variance was defined as a measure of convergence of these algorithms. We proved that the simpler task of minimizing the sample variance is not always equivalent to minimizing the asymptotic variance, in fact in certain examples minimizing one quantity produces undesirable results on the other. For the Langevin based algorithms, using Prop. 2.1, we are able to express the asymptotic variance minimization problem in a form that fits into the ∇-LSTD learning objective function. This immediately opens up the application of the various versions of ∇-LSTD learning algorithms discussed in the previous chapters to be applied to this problem. The effectiveness of these techniques is however not limited to ULA, they are also seen to produce significant variance reduction in MALA and RWM algorithms. Through recent research, theoretical justification has been sought towards this explaining this effectiveness.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

The broad objectives that this dissertation set out to address have been met. The development of ∇-LSTD learning based algorithms was mainly motivated by its application to FPF. Prior to this dissertation, the FPF gain estimation was an open problem. Only constant gain and Galerkin approximations were used in practice. We developed a new class of differential TD learning algorithms, whose applications are not restricted to FPF gain approximation. The discrete-time and finite state space analog of the generic ∇-LSTD algorithm presented here has been applied to optimal control problems like speed scaling etc. However, the original algorithm was inefficient both statistically and computationally. Statistically, it suffered from high variance issues, and computationally, an additional layer of complexity was introduced which required simulation of the Langevin SDE. This algorithm was also not friendly for online filtering problems as it requires simulating the SDE for times 10^5 or 10^6 at each time step. Prop. 2.1 provided an important breakthrough and helped us develop a ∇-LSTD version exclusive for Langevin diffusion in Chapter 2. This algorithm yields a more computationally efficient method by reducing particle sizes to $N = 500$ or 1000 from 10^6. The need to obtain a smooth approximation for the empirical posterior is also avoided. Thus, it is more "plug and play" in nature, when applied to a filtering problem.

However, one major difficulty that remained unsolved is the extension to problems with higher dimensional state spaces. An appropriate choice of a parameterized family of functions is difficult without much insight about the structure of the solution. Basis-independent versions of ∇-LSTD learning algorithms were developed in an RKHS setting. Using a recent extension of the classic representer theorem that includes gradient terms in the loss function, we are able to obtain the best approximations from within an infinite dimensional Hilbert space. The ∇-LSTD-RKHS learning algorithms allow easy extensions to higher dimensions. Performance comparison of all these different methods was done in the context of gain function approximation and filtering examples.

It was also observed that the same algorithms could be applied to minimize the asymptotic variance of MCMC algorithms. For the Langevin diffusion, the asymptotic variance minimization takes an objective function that exactly fits into the framework of ∇-LSTD learning. Through recent research, we provide theoretical justification to apply the control variates methods to non-Langevin based algorithms like RWM as well.

In spite of this, there are still open research problems, in establishing relevant theory, developing more efficient algorithms, as well as in finding practical applications:

(i) Investigate why reduced complexity solution is as good as the optimal?

(ii) Error analysis of the ∇-LSTD-RKHS method, which would provide valuable insights on choices of hyper parameters λ and ε.

(iii) A more thorough comparison of the various gain approximation algorithms is also needed.

In terms of algorithm development, we should aim for:

(i) Developing a ∇-LSTD-RKHS algorithm with a differential regularizer. Currently, this is limited by the scope of representer theorem.

(ii) Developing an algorithm based on semiparametric representer theorem, that allows the use of additional parameterized functions into the approximating class.

In terms of practical applications of the work, we need to investigate:

(i) Real time filtering problem - Potential application to battery SOC estimation is being explored.

(ii) Applications to control in the form of POMDPs.

(iii) Extensions of the results for Langevin diffusion to other MCMC algorithms.

APPENDIX A
ORTHONORMAL BASIS FUNCTIONS AND MERCER'S THEOREM

Given an RKHS $\mathcal{H}$, it is useful to understand the set of functions that belong to the Hilbert space. This can be studied using Mercer's theorem [46]. Mercer's theorem forms the connection between the theory of reproducing kernels and integral operators. Consider the integral operator (Hilbert-Schmidt operator) $L_K : L_\mu^2(x) \to L_\mu^2(x)$ defined by,

$$(L_K f)(x) = \int_{\mathsf{X}} K(x,t) f(t) d\mu(t) \tag{A-1}$$

where μ is a finite Borel measure and X is a compact set. The linear map $L_K^{1/2}$, which denotes the square root of L_K is a Hilbert isomorphism between $L_\mu^2(\mathsf{X})$ and $\mathcal{H}$ as illustrated in Fig. A-1.

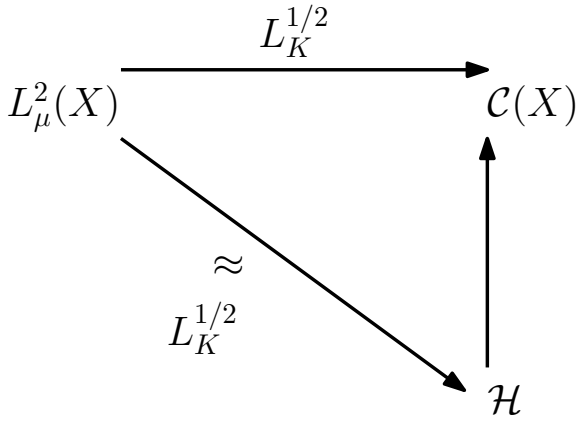


Figure A-1. Diagram illustrating the isomorphic transformations between $\mathcal{H}$ and L_μ^2.

L_K is a self-adjoint, compact operator with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$, with the corresponding normalized eigenfunctions $\{\phi_n\}_{n=1}^{\infty}$ forming an orthonormal basis for $L_\mu^2(\mathsf{X})$. Mercer's theorem states that

$$K(x,x') = \sum_{n=1}^{\infty} \lambda_n \phi_n(x) \phi_n(x'), \tag{A-2}$$

where the series converges absolutely for each $x, x' \in \mathsf{X}$. The set $\{\sqrt{\lambda_n}\phi_n\}_{n=1}^{\infty}$ forms an orthonormal basis for $\mathcal{H}$. However, finding an eigenfunction feature representation for a kernel is challenging, except in special cases.

APPENDIX B
PROPERTIES OF THE GAUSSIAN KERNEL - RKHS

Of the many kernels being used, Gaussian kernel is the most widely used and often gives the best performance [50]. The study of the properties of the Gaussan kernel has received a lot of attention [50, 51, 101]. The Gaussian kernel is a translation-invariant kernel given by,

$$K_\epsilon(x, x') := \exp(-\|x - x'\|^2/4\epsilon) \qquad \forall x, x' \in \mathsf{X}, \tag{B-1}$$

where ϵ is the variance hyperparameter that defines the width of the kernel. An illustration of the Gaussian kernel for $\epsilon = 0.125$ and the corresponding Fourier transform is given in Fig. B-1. It may be seen that the Fourier transform decays exponentially fast for large values of ω. The lack of high frequency components in the Gaussian kernel indicates that the functions belonging to the induced RKHS are smooth. It may be noted that the hyperparameter ϵ has an inverse relationship with the spread of the spectra of the kernel, i.e. lower the value of ϵ, higher frequency components are more prominent in the Fourier transform.

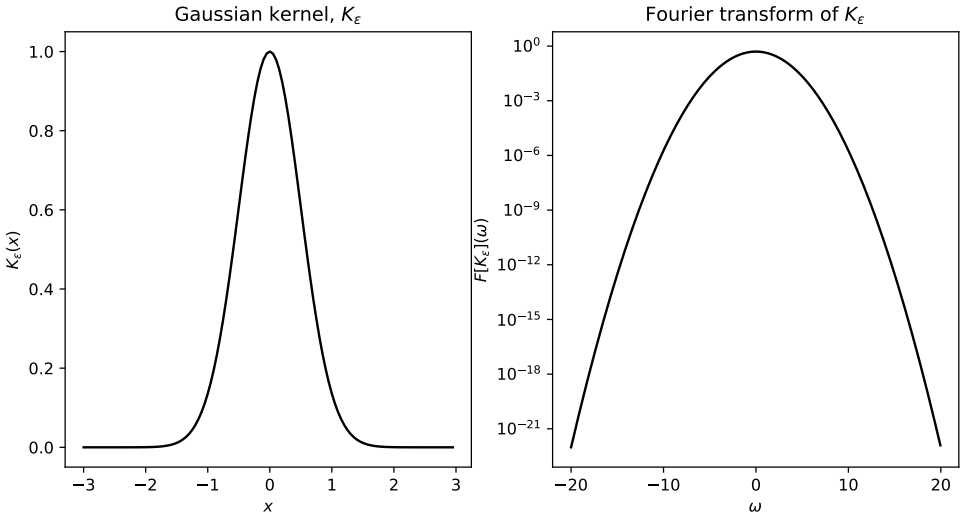


Figure B-1. Gaussian kernel with $\epsilon = 0.125$ and its Fourier transform [2].

The Gaussian kernel can be interpreted as a similarity measure of the two points passed to it. If the two points are identical, $K_\epsilon(x, x) = 1$ and if they are far apart, the value tends to

0. A simple analysis of the two extreme choices for ϵ helps us understand the behavior of the kernel better. As $\epsilon \to 0$, K_ϵ decays rapidly and increasingly tends to a dirac function centered at x and all points on X are independent of each other. On the other hand as $\epsilon \to \infty$, K_ϵ tends to 1 everywhere and the points influence each other equally. In both these cases, the learning capability of the RKHS induced is poor. Hence, the choice of ϵ can be thought of as a complexity-flexibility or bias-variance trade-off.

Steinwart et al. in [51] tries to answer questions like - which functions are contained in the RKHS induced by the Gaussian kernel, how the corresponding norms can be computed, and how the RKHS of different widths correlate to each other. In particular, RKHS of Gaussian kernels always have countable orthonormal bases. Theorem 3 of the paper gives the orthonormal basis functions for $\mathcal{H}$ defined by the Gaussian kernel K_ϵ. It states that for $\epsilon > 0$ and $n \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$, the sequence of functions $\{\phi_n : \mathbb{R} \to \mathbb{R}\}$ defined by,

$$\phi_n(x) := \sqrt{\frac{1}{(2\epsilon)^n n!}} x^n \exp(-x^2/4\epsilon). \tag{B-2}$$

is an orthonormal basis for $\mathcal{H}$.

Minh in [50] gives several properties of the RKHS induced by Gaussian kernels. Theorem 1 of the paper states that the RKHS $\mathcal{H}$ induced by the standard Gaussian kernel is infinite dimensional, i.e. $\dim(\mathcal{H}) = \infty$ and

$$\mathcal{H} := \left\{ f = \exp(-x^2/4\epsilon) \sum_{n=0}^{\infty} w_n x^n : \|f\|_{\mathcal{H}}^2 = \sum_{k=0}^{\infty} (2\epsilon)^k k! \sum_{n=0}^{k} w_n^2 < \infty \right\} \tag{B-3}$$

Some of the salient properties of the Gaussian kernel RKHS are summarized below:

1. $\mathcal{H}$ induced by the standard Gaussian kernel does not contain any polynomial on X, including the non-zero constant function.
2. If X is compact, $\mathcal{H}$ induced by the Gaussian kernel is dense in the space of $\mathcal{C}(\mathrm{X})$ of continuous functions on X. This means that given a continuous function $h(x)$, for all

$\varepsilon > 0$, we can find a function $g(x) \in \mathcal{H}$ such that

$$\|h(x) - g(x)\|_\infty \leq \epsilon \qquad \forall x \in \mathsf{X} \tag{B-4}$$

3. Let $K_\epsilon(x, x') = \exp(-\frac{\|x-x'\|^2}{4\epsilon})$. The Hilbert space $\mathcal{H}_\epsilon$ induced by K_ϵ on X contains the function $\exp(-\frac{c\|x\|^2}{4\epsilon})$ if and only if $0 < c < 2$. For example, $\exp(-\frac{\|x\|^2}{2\epsilon}) \in \mathcal{H}_{\epsilon/2}$, but $\exp(-\frac{\|x\|^2}{2\epsilon}) \notin \mathcal{H}_\epsilon$. As $c = 0$ is excluded, it validates the first property that constant functions do not belong to $\mathcal{H}$.
4. The functions in $\mathcal{H}$ that are smooth are not necessarily integrable, as $\mathcal{H} \notin L^1(\mathbb{R}^n)$ for any $\epsilon > 0$. This implies that L^1 norm optimization or regularization is infeasible in $\mathcal{H}$. This could still be done on subsets of finite linear combinations of the basis functions.
5. Partial derivatives of the Gaussian kernel denoted as $\frac{\partial^n}{\partial x^n} K_x \in \mathcal{H}$. As a corollary, it can be shown that $t^n K_x(t) \in \mathcal{H}$. Additionally, for any polynomial $p(t)$, $p(t)K_x(t) \in \mathcal{H}$. An expression for the Hilbert space norm for kernel derivative of any order d is also provided in [50].

The paper by Michelli et al. [101] sets out to identify kernels with universal approximating property, i.e. given any compact set X, any function $f \in \mathcal{C}(\mathsf{X})$, there is a function $g \in \mathcal{H}$ such that $\|f - g\|_\infty \leq \varepsilon$ holds for any $\varepsilon > 0$. Thus for any choice of compact set X, the space $\mathcal{H}$ is dense in $\mathcal{C}(\mathsf{X})$ in the maximum norm. A kernel that satisfies this property is called the universal kernel. The paper discusses the characterization of universal kernels in terms of feature map representation of the kernel K. It provides necessary and sufficient condition for K to have the universal approximation property in terms of its features. Under conditions provided in Theorem 17 of the paper, the standard Gaussian kernel is shown to be universal.

APPENDIX C
PROOF OF REPRESENTER THEOREM Theorem **??**

Proof. From the definition of the RKHS $\mathcal{H}$, any f of the form:

$$f := \sum_{i=1}^{N} \beta_i K(x_i, \cdot),$$

is in $\mathcal{H}$. Denote the linear subspace of $\mathcal{H}$ made up of all such functions f that are finite linear combinations of the kernel functions centered at x_i,

$$\mathcal{H}_{\shortparallel} := \Big\{ f \in \mathcal{H} \,|\, f = \sum_{i=1}^{N} \beta_i K(x_i, \cdot) \Big\},$$

where N ranges over $\mathbb{Z}_+$, and each $\beta_i \in \mathbb{R}$ for each i. Denote by $\mathcal{H}_\perp$, the subspace of $\mathcal{H}$ orthogonal to $\mathcal{H}_{\shortparallel}$:

$$\mathcal{H}_\perp := \{ \tilde{f} \in \mathcal{H} \,|\, \langle \tilde{f}, f \rangle_{\mathcal{H}} = 0, \, \forall f \in \mathcal{H}_{\shortparallel} \}$$

We can see that every $f \in \mathcal{H}$ can be uniquely decomposed into a component lying within $\mathcal{H}_{\shortparallel}$, denoted by $f_{\shortparallel}$ and a component lying within $\mathcal{H}_\perp$, denoted by $f_\perp$. The $\mathcal{H}$-norm can be written as,

$$\Omega(\|f\|_{\mathcal{H}}) = \Omega(\|f_{\shortparallel} + f_\perp\|_{\mathcal{H}}) = \Omega(\sqrt{\|f_{\shortparallel}\|_{\mathcal{H}}^2 + \|f_\perp\|_{\mathcal{H}}^2}) \geq \Omega(\|f_{\shortparallel}\|_{\mathcal{H}})$$

The inequality is obtained from the fact that $\|f_\perp\|_{\mathcal{H}} \geq 0$ and the function $\Omega(\cdot)$ is a strict monotone. This implies that the regularization term in (**??**) is minimized if f lies in the subspace $\mathcal{H}_{\shortparallel}$. Additionally, using reproducing property of the kernel K,

$$\begin{aligned} f(x_i) &= \langle f \,, K(x_i, .) \rangle_{\mathcal{H}} \\ &= \langle f_{\shortparallel} \,, K(x_i, .) \rangle_{\mathcal{H}} + \langle f_\perp \,, K(x_i, .) \rangle_{\mathcal{H}} \\ &= \langle f_{\shortparallel} \,, K(x_i, .) \rangle_{\mathcal{H}} \\ &= f_{\shortparallel}(x_i). \end{aligned}$$

Therefore,

$$L(x_i, f(x_i)) = L(x_i, f_{\shortparallel}(x_i))$$

The empirical error term depends only on the component $f_{\shortparallel}$. Hence, the regularized objective function is minimized if f_λ^* lies within $\mathcal{H}_{\shortparallel}$ and takes the form,

$$f_\lambda^*(x) = \sum_{i=1}^{N} \beta_i^* K(x_i, x)$$

This concludes the proof. ■

APPENDIX D
GENERALIZATION ERROR BOUNDS FOR LEAST-SQUARES REGRESSION ON RKHS

Let $Z = X \times Y$ be the space of inputs and outputs and let $S = \{z_1 = (x_1, y_1), \cdots, z_N = (x_N, y_N)\}$ be a dataset of size N drawn i.i.d. from an unknown distribution ρ_{XY}. Let $f : X \to Y$ be a function that maps from the input space X to output space Y. A learning algorithm tries to learn the function f_S from the given dataset S.

The generalization error or expected risk is defined as:

$$R(f) := \mathsf{E}_{z\sim\rho_{XY}}[c(f, z)]$$

The empirical error of a function f measured on the training set S is:

$$R_N(f) := \frac{1}{N}\sum_{i=1}^{N} c(f, z_i)$$

Our aim is to obtain bounds on the random variable $R(f_S) - R_N(f_S)$.

If $S = \{z_1, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots z_N\}$, let us define a modified training set $S_i := \{z_1, \ldots, z_{i-1}, z'_i, z_{i+1}, \ldots, z_N\}$ where the i^{th} training sample z_i is replaced by z'_i.

Uniform stability: A notion of uniform stability is defined for the algorithm as follows. If the training set S is defined as above and S_i be the training set where i^{th} sample is removed, then the algorithm is β-stable if the following holds:

$$\|c(f_S, z) - c(f_{S_i}, z)\|_\infty \leq \beta, \qquad \forall S \in Z^N, \forall z'_i, z \in Z$$

This condition implies stability of the algorithm in the sense that if a training sample from the original set is replaced by a new sample, the difference in cost is smaller than some constant β.

Lemma 5. *For any symmetric learning algorithm we have for all* $1 \leq i \leq N$*:*

$$\mathsf{E}_{S\sim\rho_{XY}^N}[R(f_S) - R_N(f_S)] = \mathsf{E}_{S,z'_i\sim\rho_{XY}^{N+1}}[c(f_S, z'_i) - c(f_{S_i}, z'_i)]$$

Proof.

$$\mathsf{E}_{S\sim\rho_{XY}^N}[R_N(f_S)] = \frac{1}{N}\sum_{i=1}^{N} \mathsf{E}_{S\sim\rho_{XY}^N}[c(f_S, z_i)] = \mathsf{E}_{S\sim\rho_{XY}^N}[c(f_S, z_i)],\ \forall i \in \{1, \ldots, N\}$$

The above is true by symmetry and the i.i.d assumption. Now by simply renaming z_i as z_i',

$$\mathsf{E}_{S\sim\rho_{XY}^N}[R_N(f_S)] = \mathsf{E}_{S_i\sim\rho_{XY}^N}[c(f_{S_i}, z_i')]$$

The expected risk term can be written as,

$$\mathsf{E}_{S\sim\rho_{XY}^N}[R(f_S)] = \mathsf{E}_{S,z_i'\sim\rho_{XY}^{N+1}}[c(f_S, z_i')]$$

Using this and the fact that the algorithm is β-stable,

$$\begin{aligned}\mathsf{E}_{S\sim\rho_{XY}^N}[R(f_S) - R_N(f_S)] &= \mathsf{E}_{S,z_i'\sim\rho_{XY}^{N+1}}[c(f_S, z_i') - c(f_{S_i}, z_i')] \\ &\leq \mathsf{E}_{S,z_i'\sim\rho_{XY}^{N+1}}[\beta] \\ &= \beta\end{aligned}$$

■

Now, the next step is to prove that our algorithm of interest is β-stable for some value of β.

D.1 Application to regularization in Hilbert spaces

Let $\mathcal{H}$ be an RKHS induced by a kernel function K. Let us assume that K is continuous and bounded, so that

$$\kappa := \sup_{x\in X}\sqrt{K(x,x)} < \infty$$

and therefore, by Cauchy-Schwarz inequality,

$$|f(x)| \leq \kappa\|f\|_{\mathcal{H}},\ \forall x \in X, \forall f \in \mathcal{H} \tag{D-1}$$

σ-**admissibility**: A cost function $c(f(x), y)$ defined on $\mathcal{H} \times Y$ is σ-admissible with respect to $\mathcal{H}$ if c is convex with respect to its first argument and the following condition holds

$$|c(y_1, y') - c(y_2, y')| \leq \sigma|y_1 - y_2|,\ \forall y_1, y_2 \in \mathcal{D}, \forall y' \in Y,$$

where $\mathcal{D} = \{y : \exists f \in \mathcal{H}, \exists x \in X, f(x) = y\}$ is the domain of the first argument of c.

Theorem D.1. *If $l(f,z) = c(f(x),y)$ is σ-admissible with respect to $\mathcal{H}$, then the learning algorithm defined by*

$$A_S = \arg\min_{g\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} l(g,z_i) + \lambda\|g\|_{\mathcal{H}}^2$$

has uniform stability β with respect to l with

$$\beta \leq \frac{\kappa^2\sigma^2}{2\lambda N}$$

Proof. The proof uses Lemma 20 in the paper that gives bounds on a general regularization term $N(g)$ that appears in the ERM. Here, I just state the result without the proof.

Lemma 6. *Consider two ERM formulations:*

$$\textit{Problem 1:} R_r(g) := \frac{1}{N}\sum_{j=1}^{N} l(g,z_j) + \lambda N(g),$$
$$\textit{Problem 2:} R_r^{\setminus i}(g) := \frac{1}{N}\sum_{j\neq i} l(g,z_j) + \lambda N(g)$$

Let f be the minimizer of R_r in $\mathcal{H}$ and let $f^{\setminus i}$ be the minimizer of $R_r^{\setminus i}$ in $\mathcal{H}$ and denote $\Delta f = f^{\setminus i} - f$. Then for any $f \in [0,1]$,

$$N(f) - N(f+t\Delta f) + N(f^{\setminus i}) - N(f^{\setminus i} - t\Delta f) \leq \frac{t\sigma}{\lambda N}|\Delta f(x_i)|$$

For regularization in the RKHS, where $N(g) = \|g\|_{\mathcal{H}}^2$,

$$2\|\Delta f\|_{\mathcal{H}}^2 \leq \frac{\sigma}{\lambda N}|\Delta f(x_i)|$$

Using (D-1),

$$\Delta f(x_i) \leq \kappa\|\Delta f\|_{\mathcal{H}},$$

so that,

$$\|\Delta f\|_{\mathcal{H}} \leq \frac{\sigma\kappa}{2\lambda N}$$

By the σ-admissibility of l,

$$|l(f,z) - l(f^{\setminus i},z)| \leq \sigma|f(x) - f^{\setminus i}(x)| = \sigma|\Delta f(x)|$$

Using (D-1),we have

$$|l(f,z) - l(f^{\setminus i},z)| \leq \sigma\kappa\|\Delta f\|_{\mathcal{H}} = \frac{\kappa^2\sigma^2}{2\lambda N}$$

■

One additional condition that is required is to bound the loss function. If we have an apriori bound on the target values y_i, then the boundedness condition is satisfied.

Let f_ρ denote the *regression function*,

$$f_\rho := \arg\min \mathsf{E}_{\rho_{XY}}[(f(x)-y)^2]$$

For a more general problem

$$f_\rho := \arg\min_f \mathsf{E}_{\rho_{XY}}[c(f(x),y)]$$

$$f_S := \arg\min_{f\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} c(f(x_i),y_i) + \lambda\|f\|_{\mathcal{H}}^2$$

APPENDIX E
EXPECTATION MAXIMIZATION (EM) ALGORITHM FOR GAUSSIAN MIXTURES

- E Step :

$$w^{(k)}(j|n) = \frac{w_j^{(k)} p_j(x^i; \mu_j^{(k)}, \sigma_j^{(k)})}{\sum_{j=0}^{m-1} w_j^{(k)} p_j(x^i; \mu_j^{(k)}, \sigma_j^{(k)})} \tag{E-1}$$

- M Step :

$$\begin{aligned} \mu_j^{(k+1)} &= \frac{\sum_{n=1}^{N} w^{(k)}(j|n) x^i}{\sum_{n=1}^{N} w^{(k)}(j|n)} \\ \sigma_j^{(k+1)} &= \sqrt{\frac{\sum_{n=1}^{N} w^{(k)}(j|n) \|x^i - \mu_j^{(k+1)}\|^2}{\sum_{n=1}^{N} w^{(k)}(j|n)}} \\ w_j^{(k+1)} &= \frac{1}{N} \sum_{n=1}^{N} w^{(k)}(j|n) \end{aligned} \tag{E-2}$$

REFERENCES

[1] T. Yang, P. Mehta, and S. Meyn, "Feedback particle filter," vol. 58, no. 10, pp. 2465–2480, Oct 2013.

[2] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[3] A. Kutschireiter, S. Carlo Surace, and J.-P. Pfister, "The Hitchhikers Guide to Nonlinear Filtering," *arXiv e-prints*, p. arXiv:1903.09247, Mar 2019.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[5] D. Brigo and B. Hanzon, "On three filtering problems arising in mathematical finance," *arXiv e-prints*, p. arXiv:0812.4050, Dec 2008.

[6] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, vol. 99, no. C5, pp. 10 143–10 162, 1994. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/94JC00572

[7] A. Bain and D. Crisan, *Fundamentals of stochastic filtering*. Springer, 2008, vol. 60.

[8] A. Budhiraja, L. Chen, and C. Lee, "A survey of numerical methods for nonlinear filtering problems," *Physica D: Nonlinear Phenomena*, vol. 230, no. 1-2, pp. 27 – 36, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/B6TVK-4M2WNVT-2/2/1e50b641f6d484aacab2c9aa19a0db3b

[9] C. Zhang, A. Taghvaei, and P. G. Mehta, "Feedback particle filter on matrix lie groups," in *2016 American Control Conference (ACC)*, July 2016, pp. 2723–2728.

[10] R. Laugesen, P. G. Mehta, S. P. Meyn, and M. Raginsky, "Poisson's equation in nonlinear filtering," in *Proc. 53rd IEEE Conference on Decision and Control*, Dec 2014, pp. 4185–4190.

[11] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," vol. 82, pp. 528–540, 1987.

[12] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970. [Online]. Available: http://www.jstor.org/stable/2334940

[13] S. P. Meyn, *Control Techniques for Complex Networks*. Cambridge: Cambridge University Press, 2007, pre-publication edition available online.

[14] S. G. Henderson and S. University., *Variance reduction via an approximating Markov process [microform]*, 1997.

[15] P. W. Glynn and S. P. Meyn, "A Liapounov bound for solutions of the Poisson equation," *Ann. Appl. Probab.*, vol. 24, no. 2, pp. 916–931, 1996.

[16] P. Dellaportas and I. Kontoyiannis, "Control variates for estimation based on reversible Markov chain Monte Carlo samplers," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 74, no. 1, pp. 133–161, 2012. [Online]. Available: http://www.jstor.org/stable/41430932

[17] A. Taghvaei and P. G. Mehta, "Gain function approximation in the feedback particle filter," in *IEEE Conference on Decision and Control*, Dec 2016, pp. 5446–5452.

[18] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[19] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.

[20] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press. On-line edition at http://www.cs.ualberta.ca/~sutton/book/the-book.html, 1998.

[21] S. J. Bradtke and A. G. Barto, "Linear least-squares algorithms for temporal difference learning," *Mach. Learn.*, vol. 22, no. 1-3, pp. 33–57, 1996.

[22] A. M. Devraj and S. P. Meyn, "Differential TD learning for value function approximation," *ArXiv e-prints and 55th IEEE Conf. on Decision and Control*, pp. 6347–6354, April 2016.

[23] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950. [Online]. Available: http://www.jstor.org/stable/1990404

[24] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *Journal of Mathematical Analysis and Applications*, vol. 33, no. 1, pp. 82 – 95, 1971. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0022247X71901843

[25] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Computational Learning Theory*, D. Helmbold and B. Williamson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 416–426.

[26] A. Radhakrishnan and S. Meyn, "Feedback particle filter design using a differential-loss reproducing kernel Hilbert space," in *American Control Conference (ACC)*, June 2018, pp. 329–336.

[27] A. Radhakrishnan, A. Devraj, and S. Meyn, "Learning techniques for feedback particle filter design," in *55th Conference on Decision and Control*, Dec 2016, pp. 5453–5459.

[28] R. N. Bhattacharya, "On the functional central limit theorem and the law of the iterated logarithm for markov processes," *Zeitschrift für Wahrscheinlichkeitstheorie*

und Verwandte Gebiete, vol. 60, no. 2, pp. 185–201, Jun 1982. [Online]. Available: https://doi.org/10.1007/BF00531822

[29] I. Kontoyiannis and S. Meyn, "Geometric ergodicity and the spectral gap of non-reversible Markov chains," vol. 154, no. 1-2, pp. 327–339, 2012, 10.1007/s00440-011-0373-4. [Online]. Available: http://dx.doi.org/10.1007/s00440-011-0373-4

[30] R. S. Laugesen, P. G. Mehta, S. P. Meyn, and M. Raginsky, "Poisson's equation in nonlinear filtering," vol. 53, no. 1, pp. 501–525, 2015. [Online]. Available: http://dx.doi.org/10.1137/13094743X

[31] A. Devraj, I. Kontoyiannis, and S. Meyn, "Geometric Ergodicity in a Weighted Sobolev Space: Part 2, Markovian diffusions," *In preparation*, 2017.

[32] E. Pardoux and Y. Veretennikov, "On the poisson equation and diffusion approximation. i," vol. 29, no. 3, pp. 1061–1085, 07 2001. [Online]. Available: http://dx.doi.org/10.1214/aop/1015345596

[33] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*, 2nd ed. Cambridge: Cambridge University Press, 2009, published in the Cambridge Mathematical Library. 1993 edition online.

[34] S. Asmussen and P. W. Glynn, *Stochastic Simulation: Algorithms and Analysis*, ser. Stochastic Modelling and Applied Probability. New York: Springer-Verlag, 2007, vol. 57.

[35] J. A. Boyan, "Technical update: Least-squares temporal difference learning," vol. 49, no. 2-3, pp. 233–246, 2002.

[36] J. Neveu, "Potentiel Markovien récurrent des chaînes de Harris," *Ann. Inst. Fourier, Grenoble*, vol. 22, pp. 7–130, 1972.

[37] S. P. Meyn and R. L. Tweedie, "Generalized resolvents and Harris recurrence of Markov processes," *Contemporary Mathematics*, vol. 149, pp. 227–250, 1993.

[38] A. Devraj, I. Kontoyiannis, and S. Meyn, "Geometric Ergodicity in a Weighted Sobolev Space," *ArXiv e-prints, and Submitted for publication*, 2017.

[39] I. Kontoyiannis and S. P. Meyn, "Spectral theory and limit theorems for geometrically ergodic Markov processes," vol. 13, pp. 304–362, 2003.

[40] J. N. Tsitsiklis and B. V. Roy, "Average cost temporal-difference learning," *Automatica*, vol. 35, no. 11, pp. 1799 – 1808, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109899000990

[41] C.-R. Hwang, R. Normand, and S.-J. Wu, "Variance reduction for diffusions," 2014.

[42] T. Yang, R. S. Laugesen, P. G. Mehta, and S. P. Meyn, "Multivariable feedback particle filter," *Automatica*, vol. 71, pp. 10–23, 9 2016.

[43] D. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Cambridge, Mass: Atena Scientific, 1996.

[44] A. M. Devraj and S. P. Meyn, "Fastest convergence for Q-learning," *ArXiv e-prints*, Jul. 2017.

[45] E. H. Moore, "On properly positive hermitian matrices," *Bull. Amer. Math. Soc.*, vol. 23, 1916. [Online]. Available: https://ci.nii.ac.jp/naid/10029628562/en/

[46] J. Mercer and A. R. Forsyth, "Xvi. functions of positive and negative type, and their connection the theory of integral equations," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 209, no. 441-458, pp. 415–446, 1909. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1909.0016

[47] G. Wahba, *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, 1990. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611970128

[48] C. Cortes and V. Vapnik, "Support-vector networks," vol. 20, no. 3, pp. 273–297, 1995.

[49] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. MIT Press, 1997, pp. 155–161. [Online]. Available: http://papers.nips.cc/paper/1238-support-vector-regression-machines.pdf

[50] H. Q. Minh, "Some properties of gaussian reproducing kernel hilbert spaces and their implications for function approximation and learning theory," *Constructive Approximation*, vol. 32, no. 2, pp. 307–338, Oct 2010. [Online]. Available: https://doi.org/10.1007/s00365-009-9080-0

[51] I. Steinwart, D. Hush, and C. Scovel, "An explicit description of the reproducing kernel hilbert spaces of gaussian rbf kernels," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4635–4643, Oct 2006.

[52] R. Willoughby, "Solutions of ill-posed problems (a. n. tikhonov and v. y. arsenin)," *SIAM Review*, vol. 21, no. 2, pp. 266–267, 1979. [Online]. Available: https://doi.org/10.1137/1021044

[53] D. D. Cox and F. O'Sullivan, "Asymptotic analysis of penalized likelihood and related estimators," *The Annals of Statistics*, vol. 18, no. 4, pp. 1676–1695, 1990. [Online]. Available: http://www.jstor.org/stable/2241881

[54] D.-X. Zhou, "Derivative reproducing properties for kernel methods in learning theory," *Journal of Computational and Applied Mathematics*, vol. 220, no. 1, pp. 456 – 463, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042707004657

[55] S. Didas, S. Setzer, and G. Steidl, "Combined 2 data and gradient fitting in conjunction with 1 regularization," *Advances in Computational Mathematics*, vol. 30, no. 1, pp. 79–99, Jan 2009. [Online]. Available: https://doi.org/10.1007/s10444-007-9061-4

[56] Y. Bhujwalla, V. Laurain, and M. Gilson, "An rkhs approach to systematic kernel selection in nonlinear system identification," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 3898–3903.

[57] D.-X. Zhou, "The covering number in learning theory," *Journal of Complexity*, vol. 18, no. 3, pp. 739 – 767, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885064X02906357

[58] Ding-Xuan Zhou, "Capacity of reproducing kernel spaces in learning theory," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1743–1752, July 2003.

[59] S. Smale and D.-X. Zhou, "Estimating the approximation error in learning theory," *Analysis and Applications*, vol. 01, no. 01, pp. 17–41, 2003. [Online]. Available: https://doi.org/10.1142/S0219530503000089

[60] F. Girosi, "Approximation error bounds that use vc-bounds," in *in: Proc. Internat. Conf. Artilicial Neural Networks*, 1995, pp. 295–302.

[61] C. Cortes, M. Mohri, and A. Rostamizadeh, "Generalization bounds for learning kernels," in *Proceedings of the 27th Annual International Conference on Machine Learning (ICML 2010)*, 2010. [Online]. Available: http://www.cs.nyu.edu/~mohri/pub/lk.pdf

[62] C. Micchelli, M. Pontil, Q. Wu, and D.-X. Zhou, "Error bounds for learning the kernel," *Analysis and Applications*, 09 2016.

[63] O. Bousquet and A. Elisseeff, "Algorithmic stability and generalization performance," in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2001, pp. 196–202. [Online]. Available: http://papers.nips.cc/paper/1854-algorithmic-stability-and-generalization-performance.pdf

[64] ——, "Stability and generalization," *J. Mach. Learn. Res.*, vol. 2, pp. 499–526, Mar. 2002. [Online]. Available: https://doi.org/10.1162/153244302760200704

[65] G. Kallianpur, *Stochastic filtering theory*, ser. Applications of Mathematics. New York: Springer-Verlag, 1980, vol. 13.

[66] M. Zakai, "On the optimal filtering of diffusion processes," *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 11, no. 3, pp. 230–243, Sep 1969. [Online]. Available: https://doi.org/10.1007/BF00536382

[67] H. J. Kushner, *Stochastic Stability and Control*. New York: Academic Press, 1967.

[68] R. L. Stratonovich, "Conditional Markov processes," *SIAM Theory Probab. Appl.*, vol. 5, pp. 156–178, 1960.

[69] R. E. Kalman, "When is a linear control system optimal?" *Journal of Basic Engineering*, vol. 86, p. 51, 1964.

[70] V. E. BeneŠ, "Exact finite-dimensional filters for certain diffusions with nonlinear drift," *Stochastics*, vol. 5, no. 1-2, pp. 65–92, 1981. [Online]. Available: https://doi.org/10.1080/17442508108833174

[71] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb 2002.

[72] A. Jazwinski, *Stochastic processes and filtering theory*, ser. Mathematics in science and engineering. New York, NY [u.a.]: Acad. Press, 1970, no. 64. [Online]. Available: http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+021832242&sourceid=fbw_bibsonomy

[73] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[74] T. Yang, P. G. Mehta, and S. P. Meyn, "A mean-field control-oriented approach to particle filtering," July 2011, pp. 2037–2043.

[75] S. K. Mitter and N. J. Newton, "A variational approach to nonlinear estimation," vol. 42, no. 5, pp. 1813–1833, 2003.

[76] D. Crisan and J. Xiong, "Approximate mckean-vlasov representations for a class of spdes," 2005.

[77] T. Yang, H. A. P. Blom, and P. G. Mehta, "The continuous-discrete time feedback particle filter," *American Control Conference*, pp. 648–653, June 2014.

[78] A. Kutschireiter, S. Surace, H. Sprekeler, and J.-P. Pfister, "Nonlinear bayesian filtering and learning: A neuronal dynamics for perception," *Scientific Reports*, vol. 7, 08 2017.

[79] A. K. Tilton, E. T. Hsiao-Wecksler, and P. G. Mehta, "Filtering with rhythms: Application to estimation of gait cycle," June 2012, pp. 3433–3438.

[80] A. K. Tilton, P. G. Mehta, and S. P. Meyn, "Multi-dimensional feedback particle filter for coupled oscillators," in *Proc. American Control Conference (ACC)*, 2013, pp. 2415–2421.

[81] K. Berntorp and P. Grover, "Data-driven gain computation in the feedback particle filter," in *2016 American Control Conference (ACC)*, July 2016, pp. 2711–2716.

[82] K. Berntorp, "Feedback particle filter: Application and evaluation," in *2015 18th International Conference on Information Fusion (Fusion)*, July 2015, pp. 1633–1640.

[83] T. Yang, R. S. Laugesen, P. G. Mehta, and S. P. Meyn, "Multivariable feedback particle filter," *ArXiv e-prints*, 2013, submitted for publication.

[84] A. Taghvaei and P. G. Mehta, "Gain function approximation in the feedback particle filter," *ArXiv and to appear, IEEE CDC*, Mar. 2016.

[85] A. Taghvaei, P. G. Mehta, and S. P. Meyn, "Error estimates for the kernel gain function approximation in the feedback particle filter," in *Proc. of the American Control Conference and arXiv*, 2017.

[86] A. K. Tilton, S. Ghiotto, and P. G. Mehta, "A comparative study of nonlinear filtering techniques," in *Proceedings of the 16th International Conference on Information Fusion*, July 2013, pp. 1827–1834.

[87] M. Evans, N. Hastings, and B. Peacock, "Ch. 41: von Mises Distribution," in *Statistical Distributions*, 3rd ed. Wiley, 2000.

[88] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Delhi, India and Cambridge, UK: Hindustan Book Agency and Cambridge University Press (jointly), 2008.

[89] A. Radhakrishnan and S. Meyn, "Gain function tracking in the feedback particle filter," in *2019 American Control Conference (ACC)*, July 2019, pp. 5352–5359.

[90] N. Brosse, A. Durmus, S. Meyn, and E. Moulines, "Diffusion approximations and control variates for MCMC," *ArXiv e-prints*, Aug. 2018.

[91] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine Learning*, vol. 50, no. 1, pp. 5–43, Jan 2003. [Online]. Available: https://doi.org/10.1023/A:1020281327116

[92] A. B. Duncan, T. Lelièvre, and G. A. Pavliotis, "Variance reduction using nonreversible langevin samplers," *Journal of Statistical Physics*, vol. 163, no. 3, pp. 457–491, May 2016. [Online]. Available: https://doi.org/10.1007/s10955-016-1491-2

[93] G. O. Roberts and J. S. Rosenthal, "Optimal scaling for various metropolis-hastings algorithms," *Statistical Science*, vol. 16, no. 4, pp. 351–367, 2001. [Online]. Available: http://www.jstor.org/stable/3182776

[94] S. Henderson, "Variance reduction via an approximating Markov process," Ph.D. dissertation, Stanford University, Stanford, California, USA, 1997.

[95] S. G. Henderson, S. P. Meyn, and V. B. Tadić, "Performance evaluation and policy selection in multiclass networks," *Discrete Event Dynamic Systems*, vol. 13, no. 1, pp. 149–189, Jan 2003. [Online]. Available: https://doi.org/10.1023/A:1022197004856

[96] S. Kim and S. G. Henderson, "Adaptive control variates for finite-horizon simulation," vol. 32, no. 3, pp. 508–527, 2007.

[97] N. Brosse, A. Durmus, S. Meyn, and E. Moulines, "Diffusion approximations and control variates for MCMC," *ArXiv e-prints*, Aug. 2018.

[98] C. J. Oates, M. Girolami, and N. Chopin, "Control functionals for monte carlo integration," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 79, no. 3, pp. 695–718, 2017. [Online]. Available: http://dx.doi.org/10.1111/rssb.12185

[99] T. Papamarkou, A. Mira, and M. Girolami, "Zero variance differential geometric markov chain monte carlo algorithms," *Bayesian Anal.*, vol. 9, no. 1, pp. 97–128, 03 2014. [Online]. Available: https://doi.org/10.1214/13-BA848

[100] G. O. Roberts and R. L. Tweedie, "Exponential convergence of Langevin distributions and their discrete approximations," *Bernoulli*, pp. 341–363, 1996.

[101] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *J. Mach. Learn. Res.*, vol. 7, pp. 2651–2667, Dec. 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1248547.1248642

BIOGRAPHICAL SKETCH

Anand Radhakrishnan was born in Kerala, India. He received the Bachelor of Technology degree in electronics and instrumentation from the University of Kerala, India in 2006, and the Master of Technology (M. Tech.) degree in control and computing from the Indian Institute of Technology, Bombay in 2010. He was with Samsung Research India Pvt Ltd. as a senior software engineer specializing in mobile communications technology between 2010 and 2012. In Spring of 2014, he joined the PhD program in the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. He was advised by Prof. Sean. P. Meyn. He received an M.S. degree in electrical and computer Engineering from the University of Florida in Spring 2019. During his PhD, he also interned at Broadcom Corporation, San Diego and Optym, Gainesville in 2013 and 2019 repsectively. He received his PhD degree in Fall 2019. He has authored papers published in Conference on Decision and Control (CDC) and American Control Conference (ACC) proceedings. His research interests include stochastic processes, state estimation, machine learning and Markov chain Monte Carlo (MCMC) algorithms.