Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

STATIONARY PROBABILITY DISTRIBUTIONS OF STOCHASTIC GRADIENT DESCENT AND THE SUCCESS AND FAILURE OF THE DIFFUSION APPROXIMATION

by
William Joseph McCann

In this thesis, Stochastic Gradient Descent (SGD), an optimization method originally popular due to its computational efficiency, is analyzed using Markov chain methods. We compute both numerically, and in some cases analytically, the stationary probability distributions (invariant measures) for the SGD Markov operator over all step sizes or learning rates. The stationary probability distributions provide insight into how the long-time behavior of SGD samples the objective function minimum.

A key focus of this thesis is to provide a systematic study in one dimension comparing the exact SGD stationary distributions to the Fokker-Planck diffusion approximation equations — which are commonly used in the literature to characterize the SGD probability distribution in the limit of small step sizes/learning rates. While various error estimates for the diffusion approximation have recently been established, they are often in a weak sense and not in a strong maximum norm. Our study shows that the diffusion approximation converges with a slow rate in the maximum norm to the true stationary distribution. In addition to large quantitative errors, the exact SGD probability distribution exhibits fundamentally different behavior to the diffusion approximation: they can have compact or singular supports; and there can be multiple invariant measures for non-convex objective functions (when the diffusion approximation only has one).

Finally, we use the Markov operator to establish additional results: (1) we show that for quadratic objective functions the SGD expected value is the objective function minimum for any step size. This has the practical implication that time average SGD solutions converge to the minimum even when the SGD iterates never reach or access

the minimum. (2) We provide a simple approach to formally derive Fokker-Planck diffusion approximations using only basic calculus (e.g., integration by parts and Taylor expansions), which may be of interest to the engineering community. (3) We observe that the stationary distributions of the Markov operator lead to additional Fokker-Planck equations with simpler diffusion coefficients than what is currently in the literature.

STATIONARY PROBABILITY DISTRIBUTIONS OF STOCHASTIC GRADIENT DESCENT AND THE SUCCESS AND FAILURE OF THE DIFFUSION APPROXIMATION

by
William Joseph McCann

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Masters of Science in Data Science with a Concentration in Statistics

Department of Mathematical Sciences

May 2021

APPROVAL PAGE

STATIONARY PROBABILITY DISTRIBUTIONS OF STOCHASTIC GRADIENT DESCENT AND THE SUCCESS AND FAILURE OF THE DIFFUSION APPROXIMATION

William Joseph McCann

Dr. David Shirokoff, Thesis Advisor — Date
Associate Professor of Mathematical Sciences, New Jersey Institute of Technology

---

Dr. James Maclaurin, Committee Member — Date
Assistant Professor of Mathematical Sciences, New Jersey Institute of Technology

---

Dr. Cristina Frederick, Committee Member — Date
Assistant Professor of Mathematical Sciences, New Jersey Institute of Technology

---

Dr. Ji Meng Loh, Committee Member — Date
Associate Professor of Mathematical Sciences, New Jersey Institute of Technology

BIOGRAPHICAL SKETCH

Author: William Joseph McCann

Degree: Master of Science

Date: May 2021

Undergraduate and Graduate Education:

- Bachelors of Science in Computer Science and Applied Mathematics, New Jersey Institute of Technology, 2016

Major: Data Science with a Concentration in Statistics

Sarge: *Simmons was the one that led us to you after he stealthily avoided capture.*
Grif: *Avoided capture!? They knocked him out first and picked me at random!*
Sarge: *Yes. A randomness that Simmons used to save the day!*

Red vs. Blue, Episode 92: Where Credit is Due

ACKNOWLEDGMENT

I would like to thank Professor David Shirokoff for all the help he has provided me on not only this project, but throughout my entire college career. I will be forever grateful for him entertaining my questions that had absolutely nothing to do with course material during his office hours while I was a first-semester freshman. His discussions with me helped fuel my love for mathematics and give me the passion to get where I am today.

I would also like to thank Professor James Maclaurin for his assistance and expertise over the past year in statistical processes, as well as fellow committee members Professor Christina Frederick and Professor Ji Ming Loh for their work to help me complete my thesis.

I would like to thank Professor David Horntrop, Professor Roy Goodman, and Daniel Pavlick for guiding me throughout my college career and helping show me opportunities that I would have never have thought I could do on my own. I would like to thank Professor Casey Diekman for spending a year working with me and showing me what a research project is like, as well as just being an overall cool guy who was a blast to spend time with.

I would like to thank my parents, sister, grandparents, and family for constantly supporting me and always being interested in the advanced theoretical mathematics I was working on.

Finally, I would like to thank my friends, including but not limited to Anuj, Ariana, George, Devin, Brenden, Maria, Stefan, Kayla, Ishani, Aarati, Alekhya, Chris, Navya, John, Jeremy, Anna, Abby, Joe, Conor, Jules, Dan, Raffay, Birra, Amulya and Ravi for making every day an exciting an adventure worth living.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Stochastic Gradient Descent

Stochastic gradient descent (SGD) is an algorithm designed to minimize objective functions of the form

$$F(\vec{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\vec{x}), \tag{1.1}$$

where $F(\vec{x})$ can be written as a sum of individual $f_i(\vec{x})$'s.

Minimizing functions of the form (1.1) is a crucial task in the training of deep-neural networks for supervised learning. In the case of supervised learning, the functions $f_i(\vec{x})$'s arise naturally as *loss functions* related to fitting given training data with an interpolant (so that the interpolant *learns* the data, see §1.2.3).

It is also relevant to discuss the main advantage that stochastic gradient descent has when compared to other methods of optimization, such as Newton's Method or standard Gradient Descent. When optimizing models to fit billions of records of data (represented by an individual f_j), computing the full gradient for Gradient Descent or Hessian for Newton's method requires evaluating the gradient or Hessian of F in (1.1) — and becomes costly when n is large. As such, even though SGD may require more iterations to converge (or become close) to a minima, and may need to be understood in a probabilistic sense, SGD provides approaches that avoid computing a full gradient or Hessian (which may provide reductions in computational time).

More interestingly, it has been shown that SGD has a regularizing effect in which the random process appears to intrinsically lower model overfitting[18].

Remark. *Stochastic gradient descent includes gradient descent as a special case when $n = 1$, or equivalently when SGD is viewed as having only one mini-batch.*

The simplest version of SGD takes the following form: Given initial data $\vec{x}_0$ and fixed *step size* $\eta > 0$,

$$\text{(SGD)} \qquad \vec{x}_{k+1} = \vec{x}_k - \eta \nabla f_{i_k}(\vec{x}_k), \text{ where,} \tag{1.2}$$
$$i_k \in \{1, 2, \ldots, n\} \text{ is drawn with a uniform distribution.}$$

In Equation (1.2), η is the *learning rate* parameter or *step size*, while the random variables i_k are drawn uniformly among the integers $[n]$ where

$$[n] := \{1, 2, \ldots, n\}.$$

Such a choice of i_k ensures that the average sampling of ∇f_i provides an *unbiased approximation* to ∇F, that is:

$$\mathbb{E}[\nabla f_{i_k}] = \sum_{i=1}^{n} \frac{1}{n} f_i = \nabla F. \tag{1.3}$$

An unbiased sampling guarantees that the expected value of an SGD step agrees with the deterministic gradient decent step, i.e., starting at a value of $\vec{x}_k$, $\mathbb{E}[\vec{x}_{k+1}] = \vec{x}_k - \eta \mathbb{E}[\nabla F(\vec{x}_{i_k})] = \vec{x}_k - \eta \nabla F(\vec{x}_k)$.

In practice, the step direction ∇f_{i_k} in (1.2) is often replaced with a *minibatch* of f_i's. Each minibatch consists of at least one, but possibly several f_i's. Formally, stochastic gradient descent with minibatches (SGD-MB) is defined via subsets (aka minibatches) $\{B_1, B_2, \ldots, B_d\}$, where $B_j \subseteq [n]$ for each $1 \leq i \leq d$. Each minibatch B_i has an associated gradient

$$g_i(\vec{x}) := \frac{1}{|B_i|} \sum_{k \in B_i} \nabla f_k(\vec{x}), \tag{1.4}$$

which are collectively assumed to satisfy an unbiased approximation $\mathbb{E}[\nabla g_j] = \nabla F$. The notation $|B_i|$ in (1.4) denotes the number of elements in the set B_i. The SGD-MB dynamics, are: Given initial data $\vec{x}_0$, and $\eta > 0$,

$$\text{(SGD-MB)} \qquad \vec{x}_{k+1} = \vec{x}_k - \eta \nabla g_{i_k}(\vec{x}_k), \text{ where,} \tag{1.5}$$
$$i_k \in \{1, 2, \ldots, d\} \text{ is drawn with a uniform distribution.}$$

As an example, the subsets $\{B_1, B_2, \ldots, B_d\}$ could be taken as a (disjoint) partition of $[n]$ with each B_j $(1 \leq j \leq d)$ having the same size $b := |B_1| = \ldots = |B_d|$ (in which case b would divide n). While the SGD in (1.2) and SGD-MB in (1.5) may produce different dynamics and stationary probability distributions, structurally, the two share the same dynamical equations once one relabels the variables $d \leftarrow n$, and $g_i \leftarrow f_i$.

The SGD in (1.2) is also not restricted to selecting the random variables i_k (which determine the choice of descent direction) with equal probability among the set $[n]$. Rather, we can have a set of selection probabilities $\{p_1, p_2, \ldots, p_n\}$ which represent the probability of selecting $\{f_1, f_2, \ldots, f_n\}$ (or B_j in SGD-MB) as descent directions. Namely, the p_j's satisfy

$$\sum_{j=1}^{n} p_j = 1, \quad \text{and } p_j > 0, \text{ for } 1 \leq j \leq n, \tag{1.6}$$

and are constrained to provide an unbiased approximation,

$$\mathbb{E}[f_{j_k}] = \sum_{j=1}^{n} p_j f_j = F. \tag{1.7}$$

Note that withough loss of generality one can take $p_j > 0$ positive; a value of $p_j = 0$ would imply that the corresponding f_j does not contribute to the dynamics or objective function.

Modifying the distribution from which i_k samples the set $[n]$ yields a *Weighted Stochastic Gradient Descent* (WSGD). In terms of the algorithm, WSGD does not

change the structural form of the update equation, however it can lead to different dynamics. Algorithmically, WSGD can be written as:

$$\text{(WSG)} \qquad \vec{x}_{k+1} = \vec{x}_k - \eta \nabla g_{i_k}(\vec{x}_k), \text{ where,} \tag{1.8}$$
$$i_k \in \{1, 2, \ldots, d\} \text{ is drawn with a distribution } \{p_1, p_2, \ldots, p_n\}.$$

Note that the WSGD (1.8) as written uses minibatch gradients f_{i_k} (so it is, more correctly, a weighted minibatch SGD). The methods presented within this thesis generalize to include both WSGD and SGD-MB.

1.2 Motivating Examples for SGD from Supervised Learning

Artificial Intelligence, and in particular machine learning, has exploded in popularity during recent years. In essence, machine learning (ML) is the process of applying statistical models to data in order to make predictions about, or understand features of, future data. While early traces of the field date back to the $1950s$, the advent of cheap, high powered computation has enabled the practical use of ML in problems today.

When discussing machine learning, there are two broad categories of models used: *supervised* and *unsupervised* [22]. In *supervised* learning models, each data point used in the training/learning process has a corresponding true output that the model is supposed to emulate. Common supervised methods include Linear Regression, Decision Trees, Logistic Regression, Support Vector Machines, and certain variations of Neural Networks. In *unsupervised* learning, the model is not given a correct output for each piece of input data, and needs to make inferences based on qualities and features of the data collection itself. Some examples of unsupervised learning methods are K-Means Clustering, Hierarchical Clustering, and Principle Component Analysis.

In order to provide an adequate insight into the type of problems that stochastic gradient descent might be used for, we will provide two simple examples that can be found in the field of machine learning.

1.2.1 The Supervised Learning Problem

A standard supervised learning problem [43] is to approximate (learn) a function $y = g(\vec{x})$ from a collection of input data $x_i \in \mathbb{R}^d$ and corresponding labels $y_i \in \mathbb{R}$ where $y_i = g(\vec{x}_i)$ for $1 \leq i \leq n$. For instance, $\vec{x}_i$ could be vectors representing images, and the values of y_i could be labels (e.g., $y_i = 1$ if the image contains a narwhal, and 0 otherwise). In practice y_i may be vector data and not restricted to scalars. Together, the data points $(\vec{x}_i, y_i)$ for $1 \leq i \leq n$ are referred to as the *training data*.

Aside from the known training data $(\vec{x}_i, y_i)$, where $y_i = g(\vec{x}_i)$, one has no other knowledge of the function $g(\vec{x})$. Mathematically, supervised learning is then equivalent to constructing an interpolation function for $g(\vec{x})$. The standard approach is to consider a family of interpolation functions $G(\vec{x}, \vec{\beta})$ parameterized by unknown—to be determined—weights, $\vec{\beta}$. The $\vec{\beta}$ values are chosen so that $G(\vec{x}, \vec{\beta})$ agrees with $g(\vec{x})$ on the known training data, e.g., $G(\vec{x}_i, \vec{\beta}) \approx y_i$ for $1 \leq j \leq n$. The most common approach to choose $\vec{\beta}$ is then to minimize the mismatch of $G(\vec{x}, \vec{\beta})$

$$\text{minimize } F(\vec{\beta}) = \frac{1}{n} \sum_{j=1}^{n} \left(y_i - G(\vec{x}_i, \vec{\beta}) \right)^2. \tag{1.9}$$

Note that $F(\vec{\beta})$ in (1.9) is (up to a factor of n^{-1}) the ℓ^2 norm squared of $\|\vec{y} - \vec{G}(\vec{x}, \vec{b})\|^2$ characterizing the mismatch of $G(\vec{x}, \vec{\beta})$ on the training data (the vector $\vec{G}$ is $\vec{G}_i = G(\vec{x}_i, \vec{\beta})$). One could of course use loss functions other than ℓ^2.

As one can see, the number of terms in (1.9) may be large in practical problems. Fortunately, however, (1.9) fits within the structure of problems for which SGD may be used. Namely with batch size 1, we would at every iteration of SGD, randomly

select one data point and minimize the resulting expression from considering only that data point, e.g., move in a step direction of $\nabla_{\vec{\beta}} f_i$ where $f_i(\vec{\beta}) = (y_i - G(\vec{x}_i, \vec{\beta}))^2$.

From the above observations, we see that SGD (as well as SGD-MB and WSGD) can all be applied to any objective of the form (1.9) (even if the loss function is not ℓ^2). We now provide two concrete examples for functions $G(\vec{x}, \vec{\beta})$ that may be used in practice.

1.2.2 Least Squares Linear Regression

For Least Squares Linear Regression we attempt to fit a linear function to our data such that we minimize the Residual Sum of Squares Error (RSS) between the data and predicted values. In general, if we have d input dimensions, then we are trying to find a $\vec{\beta} \in \mathbb{R}^d$ that solves the following minimization problem:

$$\text{minimize } F(\beta) := \frac{1}{n} \sum_{i=0}^{n} \underbrace{(\vec{x_i}^T \vec{\beta} - y_i)^2}_{f_i}. \tag{1.10}$$

Namely, the function $G(\vec{x}, \vec{\beta}) = \vec{x}^T \vec{\beta}$. Note that the factor of $1/n$ in front of the summation is added simply to write (1.10) in the form of (1.2) (re-scaling an objective function $F \to \alpha F$ for $\alpha > 0$ has the superficial effective of modifying the time step $\eta \to \alpha\eta$ in SGD). The function $F(\vec{\beta})$ is perhaps more often written as $F(\vec{\beta}) = \frac{1}{n}\|\boldsymbol{A}\vec{\beta} - \vec{y}\|^2$, where

$$\boldsymbol{A} = \begin{pmatrix} \vec{x}_1^T \\ \vdots \\ \vec{x}_n^T \end{pmatrix}, \qquad \vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

The Least Squares Regression problem is a convex optimization problem with several standard solutions (e.g., one can solve the *normal equations*, or write the solution via a QR factorization or singular value decomposition of $\boldsymbol{A}$ [44]). However, we can still consider how (1.10) would look as an objective function for the SGD in

(1.2). As written in (1.10), the objective $F(\vec{\beta})$ is a summation of $f_i = (\vec{x}_i^T\vec{\beta} - y_i)^2$ which are defined by each data point. Each step of SGD moves in a direction given by one of the f_i's.

Example 1. *Consider a toy problem where $n = 3$ with data $(\vec{x}, y) \in \mathbb{R}^2 \times \mathbb{R}$ given by:*

$$\left\{ \left([1,2]^T, 1\right), \left([-1,0]^T, 0\right), \left([2,-3]^T, 1\right) \right\}. \tag{1.11}$$

The least squares objective function is:

$$F(\vec{\beta}) = \frac{1}{3}\left(1 - [1,2]^T\vec{\beta}\right)^2 + \frac{1}{3}\left(0 - [-1,0]^T\vec{\beta}\right)^2 + \frac{1}{3}\left(1 - [2,-3]^T\vec{\beta}\right)^2. \tag{1.12}$$

Every iteration of SGD, randomly selects one of the following 3 functions:

$$f_1(\vec{\beta}) = \left(1 - [1,2]^T\vec{\beta}\right)^2, \quad f_2(\vec{\beta}) = \left(0 - [-1,0]^T\vec{\beta}\right)^2,$$
$$f_3(\vec{\beta}) = \left(1 - [2,-3]^T\vec{\beta}\right)^2,$$

and moves in a direction of $\vec{b} \to \vec{\beta} - \eta\nabla f_i(\vec{\beta})$. Alternatively, SGD can be viewed as choosing at random one of the f_i's, and taking a step direction to minimize the individual f_i (before randomly selecting the next one to "optimize").

1.2.3 A Single Layer Neural Network Example

We now consider fitting a single layer neural network model $G(\vec{x}, \vec{\beta})$ to our data. In the context of supervised learning, the neural network $G(\vec{x}, \vec{\beta})$ is taken to be a nested composition of activation functions, often either a sigmoid or ReLU (see §1.2.4) with linear affine functions.

As an example model, we take $G(\vec{x}, \vec{\beta})$ to be a single layer neural network:

$$G(\vec{x}, \vec{\beta}) = \sigma\left(w_5\sigma(w_1x_1 + w_2x_2 + b_1) + w_6\sigma(w_3x_1 + w_4x_2 + b_2) + b_3\right), \tag{1.13}$$

where the parameters $\vec{\beta} = (w_1, w_2, w_3, w_4, w_5, w_6, b_1, b_2, b_3)^T$, $\vec{x} \in \mathbb{R}^2$ and $\sigma : \mathbb{R} \to \mathbb{R}$ is the activation function. Given three data-points (1.11), the supervised learning problem is then to find weight values w_j and bias values b_j that minimize the following expression

$$\text{minimize } \sum_{i=1}^{3} \left(y_i - \sigma\left(w_5\sigma(w_1x_{i,1} + w_2x_{i,2} + b_1) + w_6\sigma(w_3x_{i,1} + w_4x_{i,2} + b_2) + b_3\right)\right)^2,$$

$$\text{subject to } \vec{w} \in \mathbb{R}^6, \vec{b} \in \mathbb{R}^3.$$

We notate that $x_{i,j}$ is the j^{th} component of the i^{th} data point.

1.2.4 Common Activation Functions in Machine Learning

Each type of machine learning model handles data using its own mixture of various mathematical functions, in a way that can be thought of as a type of cooking: many different recipes to cook require different ingredients to be handled in different ways. However, just like how many common ingredients are shared between recipes, there are numerous functions common to the various ML models.

For the following models, we stick to the convention that $\vec{x} \in \mathbb{R}^d$ represents a piece of input data, $\vec{y} \in \mathbb{R}^m$ represents corresponding output data, $\boldsymbol{W}$ represent some matrix of weights/coefficients to the input variables, and $\vec{b}$ represents an affine shift in the model, often called *bias* or *activation energy*. The notation of $\vec{x}$ as input data, $\boldsymbol{W}, \vec{b}$ as model coefficients, and $\vec{y}$ as output data are not representative of the notation that we will use in subsequent sections, they are more consistent with other literature.

The Perceptron. First introduced in 1958 by Frank Rosenblatt[39], the perceptron is a function that formed the basis of what would eventually become neural networks. The perceptron is essentially a Heaviside step function composed with a linear function, $\sigma : \mathbb{R}^d \to \{0, 1\}$, given by:

$$\sigma(\vec{x}) = \begin{cases} 1, & \vec{w}^T\vec{x} + b > 0, \\ 0, & \text{else} \end{cases}. \tag{1.14}$$

Note that in this case, $\vec{w} \in \mathbb{R}^d$ is a vector and $b \in \mathbb{R}$ is a scalar. Perceptrons are then composed into "layers" which can then have their outputs mapped into another layer of perceptrons: in modern terminology this is considered an example of a neural network.

While perceptrons initially were thought to be quite powerful, their capabilities were shown to be much more limited than initially expected in *Perceptrons: an introduction to computational geometry* in 1969[36]. The most infamous example of said limitations is the proof that the single layer set of perceptrons cannot correctly separate the binary XOR data set, and in fact can only correctly classify linearly separable data. Geometrically, the perceptron is the characteristic function for an open halfspace defined by the plane $\vec{w}^T\vec{x} + b = 0$.

The Sigmoid Function. One of the main problems with the perceptron function is that it is neither continuous, nor differentiable at the activation boundary (i.e., the interface $\vec{w}^T\vec{x}+b = 0$ separating output values of 0 from 1). Even worse, the derivative is 0 where-ever it exists — which can be problematic when training (or learning) the weights via optimization algorithms that make use of the function gradient. As such, a smooth alternative was proposed: a sigmoid, S-shaped curve. There are several

versions of the sigmoid function, $\vec{f} : \mathbb{R}^d \to \mathbb{R}^m$, two common ones are:

$$\text{(Logistic function)} \qquad \vec{\sigma}(\vec{x}) = \left(1 + \exp\left(\boldsymbol{W}\tilde{\mathrm{x}} + \tilde{\mathrm{b}}\right)\right)^{-1}, \tag{1.15}$$

$$\text{(Hyperbolic tangent)} \qquad \vec{\sigma}(\vec{x}) = \tanh\left(\boldsymbol{W}\vec{x} + \vec{b}\right). \tag{1.16}$$

In (1.15) and (1.16), the notation exp and tanh etc., is understood as applying the functions component-wise to each component of a vector, e.g., $\vec{y} = \exp(\tilde{\mathrm{x}})$ means that $y_j = \exp(\mathrm{x_j})$ for each $1 \leq j \leq d$. In words, (1.15) and (1.16) are S-shaped functions applied to each component individually of the vector $\boldsymbol{W}\vec{x} + \vec{b}$.

For the past decade, sigmoid functions have been a popular choice for an activation function, however they have started to fall out of popularity in recent years to rectified linear units (ReLUs). Sigmoid functions have a bounded range of output values and become "flat" for large values of their input arguments. As a result, sigmoids may have very small gradient values (similar to the perceptron) — which can cause issues in optimization algorithms that rely on computing the gradients of the sigmoid.

The Rectified Linear Unit (ReLU). ReLUs are an attempt to fix some of the pitfalls of sigmoid functions. This activation function is defined as $\vec{\sigma} : \mathbb{R}^d \to \mathbb{R}^m$:

$$\vec{\sigma}(\vec{x}) = \max(0, \boldsymbol{W}\vec{x} + \vec{b}), \tag{1.17}$$

where just like the sigmoid, the max is applied component-wise.

The single variable ReLU, $\sigma(x) = \max\{0, x\}$, has a constant derivative of 1 at all positive points and is unbounded as $x \to \infty$. This fixes the vanishing gradient on values of $x > 0$, even though the issue still persists on values of $x < 0$. The function is also very simple to compute, not needing to perform any numerical methods for

exponentiation on hyperbolic functions (which can become costly in applications such as ML where many functional evaluations are needed).

As noted prior though, the vanishing gradient issue still occurs on the left side but even more dramatically; this can lead to many un-utilized parameters in the model network, and a reduction of model strength by extension. As such, some people have suggested some modified variations that do not have a flat left hand side. For example there is the *Gaussian Error Linear Unit* defined as $f : \mathbb{R} \to \mathbb{R}$:

$$\sigma(x) = x\Phi(x), \tag{1.18}$$

where $\Phi(x)$ is the cumulative distribution function for the normal distribution.

CHAPTER 2

ANALYSIS OF SGD WITH MARKOV CHAIN METHODS

Table 2.1 Table Of Notation Used In Following Sections

Notation Table		
Symbol	Name	Description
F	Objective Function	Function to be minimized as per problem requirements
f_j	Splitting Function / Loss Function	Component of objective function to be selected by SGD. By definition $F = \sum_{j=1}^{n} p_j f_j$. In practice represents one possible batch of data during an SGD iteration.
p_j	Selection Probability	Probability of selecting splitting function f_j at any iteration of SGD. Generally selected to be $p_j = \frac{1}{n}$.
n	Number of Splittings	Variable representing the number of splitting functions f_j that F is decomposed into. Can be thought of in practice as the number of possible SGD batches.
$\vec{x} \in \mathbb{R}^d$	Model Parameters	Parameters of F to be varied in order to minimize F.
$\vec{G}$	Objective Gradient	$\vec{G} = \nabla F$.

Continuation of 2.1		
Symbol	Name	Description
$\vec{g_j}$	Splitting Gradient / Loss Gradient	$\vec{g_j} = \nabla f_j$.
$\rho_m(\vec{x})$	Probability Distribution at Step m	Probability distribution for the m'th iteration x_m of SGD.
$P(\vec{y}, A)$	Markov Transition Kernel	Generalization of Markov matrix, it encodes the Markov transition probabilities $P(\vec{y}, A) = \Pr(\vec{x}_m \in A \vert \vec{x}_{m-1} = \vec{y})$.
P	Markov Operator	Generalization of multiplication by a Markov matrix. An Operator who's application, i.e. integration against $P(\vec{y}, A)$, advances one iteration of the probability distribution. $\rho_{m+1}(\vec{x}) = (P\rho_m)(\vec{x})$.
$\rho(\vec{x})$	Invariant Measure/Stationary Probability Distribution/Steady State	The probability distribution for $\rho_m(x)$ after a "long" number of iterations. The distribution for which $\rho(\vec{x}) = (P\rho)(\vec{x})$.

2.1 The Markov Operator for Stochastic Gradient Descent

In this section we view stochastic gradient descent as a Markov chain and introduce (i) the Markov transition kernel, which generalizes the notion of a Markov matrix to the infinite dimensional state space; and (ii) the Markov operator, which is the generalization of matrix multiplication used to (exactly) time step the SGD probability distributions.

Standard gradient descent is an entirely deterministic procedure in which we take an input state from our state space (generally $\mathbb{R}^d$) and apply consecutive maps. Stochastic gradient descent is similar, however there are a set of mappings which are selected via probability weights. As such, it is natural to observe that stochastic gradient descent, as well as normal gradient descent, are Markov chains—each state has a fixed probability to transition to another known state.

For our purposes of stochastic gradient descent, each function $f_i(\vec{x})$ corresponds to its own map $\varphi_i(\vec{x})$. For notational convenience, we rewrite WSGE (without loss of generality, restricting to minibatch sizes of 1) as

$$\text{(WSGD')} \qquad \vec{x}_{m+1} = \varphi_{i_m}(\vec{x}_m), \text{ where } \varphi_j(\vec{y}) := \vec{y} - \eta \nabla \vec{f_j}(\vec{y}),$$
$$i_m \in \{1, 2, \ldots, d\} \text{ is drawn with a distribution } \{p_1, p_2, \ldots, p_n\}.$$

The SGD (or more generally WSGD' (1.8)) defines a Markov chain $\{\vec{x}_0, \vec{x}_1, \ldots\}$ for the evolution $\vec{x}_m$ on an infinite state space $X = \mathbb{R}^d$ (e.g., since the possible values $\vec{x}_m$ can take are uncountable). The fact that SGD defines a Markov chain follows from the simple observation that the probability of $\vec{x}_{m+1} = \vec{y}$ depends only on the previous value of $\vec{x}_m$ (i.e., SGD satisfies the *Markov property*).

In contrast to the finite dimensional state space setting where the transition probabilities for a Markov chain are characterized via a Markov matrix, here one has a more general Markov *transition kernel*. The transition kernel $P(\vec{y}, A)$ is defined

abstractly as

$$P(\vec{y}, A) := \Pr(x_m \in A \mid x_{m-1} = \vec{y}), \quad \text{for any}^1 \text{ set } A \subseteq \mathbb{R}^d, \text{ and } \vec{y} \in \mathbb{R}^d. \tag{2.1}$$

Here Pr is the probability of an event occurring, $\vec{y} \in \mathbb{R}^d$ is any point in the state space, and $A \subseteq \mathbb{R}^d$ is any (measurable) set. In words, $P(\vec{y}, A)$ is the probability that $\vec{x}_m \in A$ given that $\vec{x}_{m-1} = \vec{y}$. When the state space is finite, $P(\vec{y}, A)$ reduces to a Markov matrix. The Markov property guarantees that the transition kernel is independent of m.

The transition kernel $P(\vec{y}, A)$ can then be determined for both the deterministic (iterative) map, $n = 1$ and the full WSGD'. In the case when $n = 1$ in WSGD', i.e., $\vec{x}_{m+1} = \varphi(\vec{x}_m)$ where $\varphi(\vec{y}) := \vec{y} - \eta \nabla \vec{f}(\vec{y})$ then

$$\text{(WSGD' with } n = 1) \qquad P(\vec{y}, A) = \begin{cases} 1 & \text{when } \varphi(\vec{y}) \in A, \\ 0, & \text{when } \varphi(\vec{y}) \notin A. \end{cases} \tag{2.2}$$

This yields the following alternative form for $P(\vec{y}, A)$ when $n = 1$:

$$\text{(WSGD' with } n = 1) \qquad P(\vec{y}, A) = \int_A \delta(\vec{x} - \varphi(\vec{y}))\, \mathrm{d}\vec{x}, \quad \text{or,} \tag{2.3}$$

$$P(\vec{y},\, \mathrm{d}\vec{x}) = \delta(\vec{x} - \varphi(\vec{y}))\, \mathrm{d}\vec{x}. \tag{2.4}$$

More generally, the full WSGE' transition kernel has the form:

$$\text{(WSGD' any } n) \qquad P(\vec{y}, A) = \int_A \sum_{j=1}^{n} p_j \delta(\vec{x} - \varphi_j(\vec{y}))\, \mathrm{d}\vec{x}, \quad \text{or,} \tag{2.5}$$

$$P(\vec{y},\, \mathrm{d}\vec{x}) = \sum_{j=1}^{n} p_j \delta(\vec{x} - \varphi_j(\vec{y}))\, \mathrm{d}\vec{x}. \tag{2.6}$$

In addition to the transition kernels, we also introduce the probability distribution $\rho_m(\vec{x})$ for the variable $\vec{x}_m$ at iteration m. Intuitively, $\rho_m(\vec{x})\, \mathrm{d}\vec{x}$ is the probability of $\vec{x}_m$ being in a box $\mathrm{d}\vec{x}$ at $\vec{x}$, i.e. given a measurable set $A \subseteq \mathbb{R}^d$,

$$\int_A \rho_m(\vec{x})\, \mathrm{d}\vec{x} = \Pr(\vec{x}_m \in A). \tag{2.7}$$

In (2.7) we have abused the notation somewhat — in general $\rho_m(\vec{x})\,\mathrm{d}\vec{x}$ is the probability measure for $\vec{x}_m$ and can be a non-classical function such as a Dirac mass.

The transition kernels can then be used to characterize how the probability distributions $\rho_m(\vec{x})$ of SGD evolve in time. Formally, this is done through the introduction of the Markov operator P characterizing the (discrete-in-time) evolution of the probability distribution through integration against $P(\vec{y},\,\mathrm{d}\vec{x})$, that is:

$$\rho_{m+1}(\vec{x})\,\mathrm{d}\vec{x} = \int_{\mathbb{R}^d} P(\vec{y},[x,x+\,\mathrm{d}\vec{x}])\rho_m(\vec{y})\,\mathrm{d}\vec{y}, \tag{2.8}$$

which is a Chapman-Kolmogorov Equation [40], and hence

$$\rho_{m+1}(\vec{x}) = \int_{\mathbb{R}^d} \sum_{j=1}^{n} p_j \delta\Big(\vec{x}-\varphi_j(\vec{y})\Big)\rho_m(\vec{y})\,\mathrm{d}\vec{y}. \tag{2.9}$$

Equation (2.9) characterizes the exact evolution of the probability distribution for WSGD. Equation (2.8) can also be recast in operator form (by substituting (2.6)) as

$$\rho_{m+1}(\vec{x}) = (P\rho_m)(\vec{x}), \quad \text{where} \tag{2.10}$$

$$(P\rho)(\vec{x}) := \int_{\mathbb{R}^d} \sum_{j=1}^{n} p_j \delta\Big(\vec{x}-\varphi_j(\vec{y})\Big)\rho(\vec{y})\,\mathrm{d}\vec{y}. \tag{2.11}$$

Here P is referred to as a Markov operator (when $n=1$, P is also referred to as the Perron-Frobenius operator [27]). If we state that $\rho_m(\vec{x})$ is the initial state probability distribution, then the continuous version of multiplying by a Markov matrix to get the next state is the Markov operator given as (2.10)–. This can be thought of intuitively as to get the probability at the point $\vec{x}$ at step $m+1$, you add the probability of all points at step m where the maps $\varphi_j(\vec{y}) = \vec{x}$.

2.2 Formal Derivation of the Fokker-Plank Approximations to SGD

In this section we perform a formal derivation of the Fokker-Plank approximation to equation (2.9) (equivalently (2.10)–(2.10)), by expanding about powers of η.

There have been numerous works in the literature that establish and/or study partial differential equation (PDE) approximations for the SGD probability evolution ([7, 8, 33] examined Fokker-Plank/variational models for SGD; see [29, 2, 20, 13, 12] for rigorous derivations of the diffusion approximation and higher order PDEs; see [42] for the closely related problem of approximating SGD via a stochastic ODE). The advantage of the approach here is that we start with the exact Markov operator and formally derive the PDE approximations using only basic calculus: (i) integration by parts; and (ii) Taylor expansions. The derivation as presented is not rigorous — only formal (see [13] for a rigorous approach starting from the Markov operator). In addition to deriving the diffusion approximation for the time evolution, we also obtain an additional PDE for the stationary probability distributions that has a different diffusion coefficient than the diffusion approximation.

We begin by multiplying (2.9) by a smooth test function that vanishes at infinity $\Psi(\vec{x})$ and integrate to get that

$$\begin{aligned}
\langle \Psi, \rho_{m+1}, \rangle &= \int_{\mathbb{R}^d} \Psi(\vec{x}) \rho_{m+1}(\vec{x}) d\vec{x} \\
&= \sum_{j=1}^{n} \int_{\mathbb{R}^d} \Psi(\vec{x}) p_j \int_{\mathbb{R}^d} \delta\left(\vec{x} - \vec{\varphi_j}(\vec{x})\right) \rho_m(\vec{y}) d\vec{y} d\vec{x} \\
&= \sum_{j=1}^{n} p_j \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \Psi(\vec{x}) \delta\left(\vec{x} - \vec{\varphi_j}(\vec{x})\right) \rho_m(\vec{y}) d\vec{x} d\vec{y} \\
&= \sum_{j=1}^{n} p_j \int_{\mathbb{R}^d} \Psi(\vec{y} - \eta \vec{g_j}(\vec{y})) \rho_m(\vec{y}) d\vec{y} \\
&= \langle P^\dagger \Psi, \rho_m(\vec{x}) \rangle
\end{aligned} \tag{2.12}$$

where we can define the adjoint of P as

$$
\begin{aligned}
P^{\dagger}\Psi(\vec{y}) &= \sum_{j=1}^{n} p_j \Psi\left(\vec{y} - \eta \vec{g_j}(\vec{y})\right) \\
&= \sum_{j=1}^{n} p_j \left[\Psi(\vec{y}) - \eta \vec{g_j}(\vec{y}) \cdot \nabla \Psi(\vec{y}) + \frac{\eta^2}{2} \vec{g_j}^T(\vec{y}) [H\Psi(\vec{y})] \vec{g_j}(\vec{y}) - \ldots \right].
\end{aligned} \tag{2.13}
$$

Note that in the previous step we Taylor expanded the adjoint operator. By definition, we have that

$$
\sum_{j=1}^{n} p_j = 1, \quad \sum_{j=1}^{n} p_j f_j(\vec{y}) = F(\vec{y}), \tag{2.14}
$$

which means that we can then simplify our expression down to the following infinite sum

$$
P^{\dagger}\Psi(\vec{y}) = \Psi(\vec{y}) - \eta \nabla F(\vec{y}) \cdot \nabla \Psi(\vec{y}) + \frac{\eta^2}{2!} \sum_{j=1}^{n} p_j \vec{g_j}^T(\vec{y}) [H\Psi(\vec{y})] \vec{g_j}(\vec{y}) \ldots \tag{2.15}
$$

where H represents taking the Hessian. We will state that $\vec{g}_j = [g_j^1, g_j^2, \ldots, g_j^d]^T$ in order to refer to the components of g_j (where $\vec{g}_j = \nabla f_j$ is short hand for the gradients). We now will notate the r^{th} order truncation of P as $\mathcal{P}_r$. The first three $\mathcal{P}_r$ operators are the following:

$$
\begin{aligned}
&\mathcal{P}_0^{\dagger} = \boldsymbol{I}, \\
&\mathcal{P}_1^{\dagger} = \boldsymbol{I} - \eta \sum_{j=1}^{n} p_j \sum_{i=1}^{N} g_j^i \partial_i = \boldsymbol{I} - \eta [\nabla F(\vec{y})] \cdot \nabla, \\
&\mathcal{P}_2^{\dagger} = \boldsymbol{I} - \eta [\nabla F(\vec{y})] \cdot \nabla + \frac{\eta^2}{2} \sum_{j=1}^{n} p_j \sum_{k,l=1}^{d} g_j^l g_j^k \partial_l \partial_k, \\
&\mathcal{P}_3^{\dagger} = \boldsymbol{I} - \eta [\nabla F(\vec{y})] \cdot \nabla + \frac{\eta^2}{2} \sum_{j=1}^{n} p_j \sum_{k,l=1}^{d} g_j^l g_j^k \partial_l \partial_k - \frac{\eta^3}{3!} \sum_{j=1}^{n} p_j \sum_{l,k,p=1}^{d} g_j^l g_j^k g_j^p \partial_l \partial_k \partial_p,
\end{aligned} \tag{2.16}
$$

where $\boldsymbol{I}$ represents the identity operator and ∂_k represents $\frac{\partial}{\partial x_k}$. Suppose that we want to approximate $P^\dagger$ by these truncations, we will find that

$$\langle \Psi, \rho_{m+1} \rangle = \langle P^\dagger \Psi, \rho_m \rangle \approx \langle \mathcal{P}_r^\dagger \Psi, \rho_m \rangle = \langle \Psi, \mathcal{P}_r \rho_m \rangle \tag{2.17}$$

therefore we can say that

$$\rho_{m+1} \approx \mathcal{P}_r(\rho_m). \tag{2.18}$$

To provide a few examples of these truncations, denote

$$\begin{aligned}
\mathcal{P}_1(\rho) &= \rho + \eta \nabla \cdot \big(\nabla F(\vec{x})\rho\big), \\
\mathcal{P}_2(\rho) &= \rho + \eta \nabla \cdot \big(\nabla F(\vec{x})\rho\big) + \frac{\eta^2}{2} \sum_{j=1}^{n} p_j \sum_{k,l=1}^{d} \partial_l \partial_k \big(g_j^l(\vec{x}) g_j^k(\vec{x}) \rho\big), \\
\mathcal{P}_3(\rho) &= \rho + \eta \nabla \cdot \big(\nabla F(\vec{x})\rho\big) + \frac{\eta^2}{2} \sum_{j=1}^{n} p_j \sum_{k,l=1}^{d} \partial_l \partial_k \big(g_j^l(\vec{x}) g_j^k(\vec{x}) \rho\big) \\
&\quad + \frac{\eta^3}{3!} \sum_{j=1}^{n} p_j \sum_{l,k,p=1}^{d} \partial_l \partial_k \partial_p (g_j^l(\vec{x}) g_j^k(\vec{x}) g_j^p(\vec{x}) \rho).
\end{aligned} \tag{2.19}$$

Now let us consider what would happen if ρ was the solution to the following linear partial differential equation

$$\partial_t \rho = \mathcal{A} \rho, \tag{2.20}$$

and we sample at time steps of $m \cdot \eta$, we would find that

$$\rho_{m+1} = e^{\mathcal{A}\eta} \rho. \tag{2.21}$$

If we compare this to (2.18) we will find that

$$e^{\mathcal{A}\eta} = \mathcal{P}_r \implies \mathcal{A} = \frac{1}{\eta}\log \mathcal{P}_r \tag{2.22}$$

It is convenient to define $\mathcal{G} := (\mathcal{P}_r - I)/\eta$, so that $\mathcal{P}_r = I + \eta\mathcal{G}$, and $\mathcal{A} = \eta^{-1}\log(I + \eta\mathcal{G})$:

$$\frac{\partial \rho}{\partial t} = \frac{1}{\eta}\log(I + \eta\mathcal{G})\rho \tag{2.23}$$

Using $\epsilon^{-1}\log(1+\epsilon r) = r - \frac{1}{2}\epsilon r^2 + \frac{1}{3}\epsilon^2 r^3 + \ldots$, one has:

$$\frac{\partial \rho}{\partial t} = \mathcal{G}\rho - \frac{\eta}{2}\mathcal{G}^2\rho + \frac{1}{3}\eta^2\mathcal{G}^3\rho + O(\eta^3), \tag{2.24}$$

where

$$\begin{aligned}\mathcal{G}\rho = \nabla\cdot\big(\nabla F(\vec{x})\rho\big) + \frac{\eta}{2}\sum_{j=1}^{n} p_j \sum_{k,l=1}^{d} \partial_l\partial_k\big(g_j^l(\vec{x})g_j^k(\vec{x})\rho\big) \\ + \frac{\eta^2}{3!}p_j \sum_{l,k,p=1}^{d} \partial_l\partial_k\partial_p(g_j^l(\vec{x})g_j^k(\vec{x})g_j^p(\vec{x})\rho).\end{aligned} \tag{2.25}$$

Note that this was recieved by truncating the perturbed Taylor Expansion (2.24) in order to get approximate partial differential equations.

2.3 Fokker-Planck Approximations for the Stationary Distribution

For a time varying partial differential equation, we define the stationary solutions (also known as the *invariant measure* or *steady state*), to be the solution to the equation to which there is no change over time

$$\frac{\partial u}{\partial t} = 0. \tag{2.26}$$

This can be thought of as the result that is retrieved after a long period of time, assuming convergence, or a fixed point solution to the equation. From our above sections, we know that we can formulate our Markov operator as a partial differential equation that has steady states. We can also look at the steady states of the equation that occur when

$$\mathcal{P}\rho = \rho \tag{2.27}$$

directly from our Markov operator. Note that PDE approximations to (2.27) do *not* necessarily provide, in a systematic fashion, the same steady states PDEs as (2.26).

We introduce the following notation in order to make the equations simpler to write

$$D(\vec{x}) = \sum_{j=1}^{n} p_j \vec{g_j} \vec{g_j}^T - \vec{G}\vec{G}^T, \quad u(\vec{x}) = \nabla\left(F + \frac{\eta}{4}(\vec{G})^2\right), \tag{2.28}$$

where $\vec{G} = \nabla F$. In 1 dimension this simplifies to

$$D(x) = \sum_{j=1}^{n} p_j g_j^2 - G^2, \quad u(x) = \frac{\partial}{\partial x}\left(F + \frac{\eta}{4}(G)^2\right). \tag{2.29}$$

We also use the constant that $\beta^{-1} = \frac{\eta}{2}$. From 2.19 we can state that if $\mathcal{P}\rho = \rho$ then

$$0 = \nabla \cdot (\nabla F \rho) + \beta^{-1} \sum_{j=1}^{n} p_j \sum_{k,l=1}^{d} \partial_l \partial_k \big(g_j^l(\vec{x}) g_j^k(\vec{x}) \rho\big). \tag{2.30}$$

We will refer to (2.30) as the *Markov PDE*. This PDE is not time variational as we received it directly from the truncations of the Markov operator. In addition,

from (2.25) we can find our other version of the equation by taking the steady state equation for the PDE, which gives us

$$0 = \nabla \cdot \left(u\rho + \beta^{-1} \nabla \cdot (D(\vec{x})\rho) \right). \tag{2.31}$$

In the literature, (2.31) is referred to as the *diffusion approximation*, and we will often refer to it as the diffusion equation.

Finally, We note that the following PDE:

$$0 = \nabla \cdot \left(\nabla F \rho + \beta^{-1} \nabla \cdot (D(\vec{x})\rho) \right), \tag{2.32}$$

has been used in various forms throughout the literature as a model PDE, often without rigorous justification. Eqaution (2.32) does, however, agree with the Fokker-Planck equations truncated to $\mathcal{O}(\eta)$. We refer to (2.32) as the *model equation*.

2.4 Fokker-Planck Solutions in 1-D

In one dimension at steady state, we get that our Model ODE, Markov ODE, and Diffusion Equation reduce to a simple form that can be exactly solved via simple differential equations methods. Integrating out one derivative term from each equation, and setting the constant term to 0 such that the resultant solution is a proper probability distribution, we find that our solutions are all of the form

$$\rho(x) = Ze^{-I(x)}. \tag{2.33}$$

Here Z, and $I(x)$ are to be determined and depend on the format of the equation. Note that each stationary PDE equation is of the form

$$0 = \frac{\partial}{\partial x} \left[U(x)\rho + \frac{\eta}{2} \frac{\partial}{\partial x} (D(x)\rho) \right] \tag{2.34}$$

with their respective versions of $U, D(x)$. We now observe the follow special case:

Assumption: $D(x) > 0$ for all $x \in \mathbb{R}$. Note that we first integrate out the collective $\frac{\partial}{\partial x}$ term in (2.34) and set the constant of integration to 0. We can do this as we know that the solution $\rho(x)$ that we get from these models satisfy the following properties:

1. $\int_{-\infty}^{\infty} \rho(x)dx = 1$
2. $\forall x \in \mathbb{R}$, $\rho(x) \geq 0$.

From (2.34) and with the assumption that $D > 0$ we can solve our ODEs using simple integration for $I(x)$

$$I(x) = \int_0^x \frac{2}{\eta} \frac{U + \frac{\eta}{2} D'}{D} dx = \log(D) + \int_0^x \frac{2U}{\eta D} dx. \tag{2.35}$$

Since $D > 0$ we can simplify our expression further to

$$\rho = \frac{Z}{D} e^{-\int_0^x \frac{2U}{\eta D} dx} \tag{2.36}$$

where Z is the constant that normalizes our expression. This expression is particularly interesting because it shows that our approximations will believe there to be support for all real numbers. As we will see in future sections this is not true, we can even come up a splitting for *any* potential objective function we would like that would have an infinitely supported approximation.

Theorem 1. *For any objective function $F : \mathbb{R} \to \mathbb{R}$, there exists a splitting of $F(x)$ for which the ODE approximations will provide a solution with infinite support.*

Proof. Since (2.36) is infinitely supported when $D(x) > 0$, we need to find a splitting of F that will always provide a non-zero value of D. Suppose that we were to split $F(x)$ into two functions

$$F(x) = \frac{1}{2}f_1(x) + \frac{1}{2}f_2(x), \tag{2.37}$$

where $p_1 = p_2 = \frac{1}{2}$ for the sake of simplicity. We can compute $D(x)$ through some algebraic manipulation

$$D(x) = \frac{1}{2}\left((f_1')^2 + (f_2')^2\right) - \frac{1}{4}(f_1' + f_2')^2 \tag{2.38}$$

$$= \frac{1}{4}(f_1')^2 - \frac{1}{2}f_1'f_2' + \frac{1}{4}(f_2')^2 \tag{2.39}$$

$$= \frac{1}{4}\left(f_1' - f_2'\right)^2. \tag{2.40}$$

As such if we select the following splitting functions, then we will find that our value of $D(x)$ evaluates down to a constant

$$f_1(x) = F(x) + x \tag{2.41}$$

$$f_2(x) = F(x) - x. \tag{2.42}$$

Since in this situation we have that $D(x) = 1$, we know that our solution to the ODEs will be infinitely supported. Q.E.D.

Something to notice from this proof is that we can easily construct splittings for any given objective function solely by shifting our objective by a linear term. From this we will get a resulting constant that then allows us to have this infinitely supported exponential function as our distribution.

Another particularly interesting observation is that if we do not simplify our integral expression, we can actually let it become the following form

$$\begin{aligned}\rho(x) &= Z^{-1}\, D(x)^{-1}\exp\left(-\frac{2}{\eta}\int \frac{U(x)}{D(x)}\mathrm{d}x\right),\\ &= Z^{-1}\exp\left(-\frac{2}{\eta}\int D(x)^{-1}\frac{d}{dx}[\Phi(x)+\frac{\eta}{2}D(x)]\mathrm{d}x\right),\end{aligned} \tag{2.43}$$

where $U(x) = \Phi'(x)$. This is particularly interesting as it shares the same form as some equations from statistical physics

$$\text{(Gibb's measure)} \qquad \rho(x) = Z^{-1}e^{-\beta\, V(x)}, \qquad \text{where,} \tag{2.44}$$

$$\text{(Inverse temperature } \beta) \qquad \beta^{-1} := \frac{\eta}{2}, \tag{2.45}$$

$$\text{(Free energy)} \qquad V(x) := \int D^{-1}(x)\frac{d}{dx}\Big(\Phi(x)+\beta^{-1}D(x)\Big)\mathrm{d}x, \tag{2.46}$$

$$\text{(Partition function)} \qquad Z := \int_{-\infty}^{\infty} e^{-\beta V(x)}\mathrm{d}x. \tag{2.47}$$

It is interesting to observe that when $V(x)$ is smooth, the point x^* with highest probability satisfies:

$$x^* = \operatorname{argmin} V(x), \tag{2.48}$$

$$V'(x^*) = 0, \quad \Longleftrightarrow \quad \frac{d}{dx}\Big(\Phi(x)+\beta^{-1}D(x)\Big) = 0. \tag{2.49}$$

Hence, we are led to the following observation:

Remark. *In the limit as $0 < \eta \ll 1$, the points x^* with highest probability (e.g., maximize ρ) are critical points of the original objective function $F(x)$, i.e.,* $\frac{d}{dx}F(x^*) = 0$ *only if* $\frac{d}{dx}D(x^*) = 0$.

2.4.1 Lyapunov Entropy Functional for Diffusion Approximation

The diffusion approximation in 1d always exhibits a Lyapunov functional — which has a unique minimum given by (2.43). For the time variational version PDE of our model equations, given as

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial x}\left[U(x)\rho + \frac{\eta}{2}\frac{\partial}{\partial x}\left(D\rho\right)\right], \tag{2.50}$$

we can show and find the existance of a Lyapunov entropy function that is minimized over time. Note that this is only for the Diffusion equation (2.31) and the Model equation (2.32), as the Markov ODE (2.30) is derived intrinsically through steady states.

The PDE (2.50) has the following variational form:

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial x}\Big(D(x)\rho(x)\frac{\partial}{\partial x}\frac{\delta \mathcal{E}_\beta}{\delta \rho}\Big), \tag{2.51}$$

where

$$\mathcal{E}_\beta(\rho) = \int V(x)\rho(x) + \beta^{-1}\rho(x)\log\rho(x)\,\mathrm{d}x. \tag{2.52}$$

This follows since

$$\begin{aligned}
\frac{\delta \mathcal{E}_\beta}{\delta \rho} &= V(x) + \beta^{-1}\log\rho(x) + \beta^{-1}, && (2.53)\\
\implies \frac{\partial \rho}{\partial t} &= \frac{\partial}{\partial x}\Big(D(x)\rho(x)\frac{\partial}{\partial x}\big(V + \beta^{-1}\log\rho + \beta^{-1}\big)\Big), && (2.54)\\
&= \frac{\partial}{\partial x}\Big(D(x)\rho(x)\big(D^{-1}\frac{d}{dx}\big(\Phi(x) + \beta^{-1}D(x)\big) + \beta^{-1}\rho^{-1}\rho_x\big)\Big), && (2.55)\\
&= \frac{\partial}{\partial x}\Big(\Phi'(x)\rho(x) + \beta^{-1}\rho D'(x) + \beta^{-1}D(x)\rho_x\big)\Big), && (2.56)\\
&= \frac{\partial}{\partial x}\Big(\Phi'(x)\rho(x) + \beta^{-1}\partial_x\big(\rho D(x)\big)\Big), && (2.57)
\end{aligned}$$

which is exactly the PDE (2.50) with $\beta^{-1} = \eta/2$. Further, this variational structure guarantees $\mathcal{E}_\beta(\rho)$ is a Lyapunov functional:

$$\frac{d}{dt}\mathcal{E}_\beta(\rho) = -\int_{-\infty}^{\infty} D(x)\rho(x)\Big(\frac{d}{dx}\frac{\delta \mathcal{E}_\beta}{\delta \rho}\Big)^2 \,\mathrm{d}x, \tag{2.58}$$

$$= -\int_{-\infty}^{\infty} D(x)\rho(x,t)\Big(\frac{d}{dx}\big(V(x) + \beta^{-1}\log\rho(x,t)\big)\Big)^2 \,\mathrm{d}x, \tag{2.59}$$

$$\leq 0. \tag{2.60}$$

The calculation above is actually not completely rigorous — it made use of integration by parts and the assumption that the boundary term vanishes, i.e.,

$$\Big[D(x)\rho(x)\frac{\delta \mathcal{E}_\beta}{\delta \rho}\frac{d}{dx}\Big(\frac{\delta \mathcal{E}_\beta}{\delta \rho}\Big)\Big]_{-\infty}^{\infty} = 0, \quad \forall\, t \geq 0.$$

It also assumed that the energy $\mathcal{E}_\beta < \infty$ was finite for all $\rho(x,t)$.

2.5 Numerical Methods for the Stationary Probability Distributions

Since stochastic gradient descent is an intrinsically random dynamical system that has the form of an infinite dimensional Markov Operator we can observe the properties of Markov matrices and operators in relation to SGD. In particular we are interested in the existence, computation, and approximation of an stationary probability distribution (invariant measure) of SGD. When the Markov operator satisfies suitable conditions (for instance has a spectral gap), the invariant measure is the probability distribution of where the value $\vec{x}_m$ will sample after in infinite amount of time.

However, calculations of these stationary probability distributions (to which a system may have more than one) analytically is either needlessly difficult or impossible, which is why for each particular system we will turn to numerical computations in order to make observations about the properties and dynamics of the true distributions. Our particular choice of numerical method is Ulam's Method for calculating invariant measure because of its simplicity to implement.

2.5.1 Ulam's Method for Computing Stationary Distributions

We will approximate the Continuous Markov Operator P as a finite matrix $\boldsymbol{P}$ acting upon a finite distribution vector $\vec{p}$ [31]. As such we are approximating the relation

$$\rho_{m+1}(\vec{x}) = P\rho_m(\vec{x}) \tag{2.61}$$

with the matrix vector product

$$\vec{p}_{m+1} = \boldsymbol{P}\vec{p}_m. \tag{2.62}$$

Here $\vec{p}$ is defined to be the probability on the interval $[-L, L]$ with a finite grid spacing given by H. As such that we can consider that the j^{th} component of $\vec{p}$, notated as $\vec{p}_m^j$, to represent the probability of x being in the cube with a corner at $-L + jH$. In terms of probability this would be written as

$$\vec{p}_m^j \approx \int_{-L+jH}^{-L+(j+1)H} \rho_m(\vec{x})\,\mathrm{d}\vec{x}. \tag{2.63}$$

For our purposes we only consider the usage of Ulam's Method in $1D$, however it could be extended to higher dimensions as well.

We now need to build the matrix $\boldsymbol{P}$. Notice that the element $\boldsymbol{P}^{ij}$ represents the proportion probability from bin $\vec{p}_m^i$ that moves to $\vec{p}_{m+1}^j$. From this we can numerically calculate the values of our matrix through repeated application of our map on the intervals from the following algorithm. For $\boldsymbol{P}^{ij}$ we first apply the map $\varphi(x)$ to a grid of k points on the interval $[-L + iH, -L + (i + 1)H]$. To then get $\boldsymbol{P}^{ij}$ we calculate the proportion of points that land in the interval $[-L + jH, -L + (j + 1)H]$. This gives us the matrix for one map $\varphi(x)$, the total Markov matrix for SGD is then the sum of the matrices for each of the maps $\varphi_1(x), \ldots, \varphi_n(x)$.

Once we compute our matrix, we can get the Stationary Probability Distribution by solving for $\vec{p}$ in which

$$\boldsymbol{P}\vec{p} = \vec{p}. \tag{2.64}$$

This is equivalent to solving for the eigenvector of $\boldsymbol{P}$ for an eigenvalue $\lambda = 1$. As long as the domain Ω is large enough so that Ω is a *trapping region* for each map $\varphi_j(x)$ and $1 \leq j \leq n$ (i.e., $\varphi(x) \in \Omega$ for each $\vec{x} \in \Omega$) then $\boldsymbol{P}$ is (exactly) a Markov matrix. Hence, by construction, the largest eigenvalue of $\boldsymbol{P}$ is 1.

2.5.2 Ulam's Method: Code Validation via the Logistic Map

While we perform particle simulations in later sections, in this section we compare our Ulam's method computations to a known invariant measure: the invariant measure of the logistic map. This will provide a way to systematically validate the code.

The logistic map is the map $x_{n+1} = \varphi(x_n; r)$ where $\varphi(x; r) = rx(1-x)$ with $x \in [0, 1]$. It is known that for larger values of r, the system will become chaotic, with $r = 4$ being total chaos. However, despite the system being chaotic, there is a known, analytic invariant measure providing the probability (which is sampled over long times by the iterates x_n of the map). The invariant measure is given by [21]:

$$\rho(x) = \frac{1}{\pi\sqrt{x(1-x)}}.$$

Figure 2.1 plots the invariat measure computed via Ulam's method against the known analytic result; the right subfigure contains a convergence plot (in L^1) as the mesh $H \to 0$ (generally Ulam's method exhibits slow convergence in H).

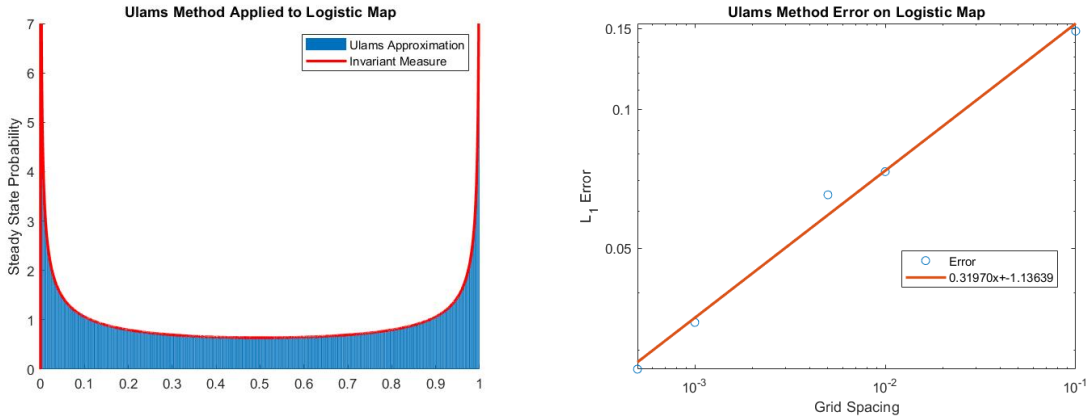


Figure 2.1 Convergence plot of Ulam's method to the invariant measure of the logistic map. Visually the invariant measure fits quite well even for larger values of H the grid spacing. Error decays at a low rate as Ulam's method is not even an $\mathcal{O}(H)$[9] method and there are singularities on the interval boundaries.

CHAPTER 3

STOCHASTIC GRADIENT DESCENT FOR A QUADRATIC OBJECTIVE FUNCTION

In this section we numerically compute the exact stationary probabilities to SGD for a quadratic test problem, as well as the Fokker-Planck solutions. Although simple, the quadratic test problem is important because (i) it corresponds to the least square minimization problem which is important in practice and has widespread application; and (ii) the quadratic can be used as a local model for nonquadratic objective functions in the vicinity of a local minimum, and hence can shed light on more complex problems.

In this section we choose a quadratic cost function $F(x) = \frac{1}{2}x^2$ with a quadratic splitting into two functions parameterized by $a \in \mathbb{R}_+$ and $b \in \mathbb{R}$:

$$f_1(x) = \frac{1}{2}(1+b)\,x^2 + ax, \quad f_2(x) = \frac{1}{2}(1-b)\,x^2 - ax. \tag{3.1}$$

We will choose minibatches with size $k = 1$, which means that at each iteration we have a $\frac{1}{2}$ chance of choosing between the maps

$$\varphi_1(x_i) = x_i - \eta((1+b)x_i + a) = (1 - \eta(1+b))x_i - \eta a \tag{3.2}$$

$$\varphi_2(x_i) = x_i - \eta((1-b)x_i - a) = (1 - \eta(1-b))x_i + \eta a. \tag{3.3}$$

3.1 Symmetries and Scalings of the Quadratic Problem

With the above choice, SGD becomes:

$$x_{n+1} = (1 - \eta(1+b))x_n - \eta a, \quad \text{with probability } \frac{1}{2}, \tag{3.4}$$

$$x_{n+1} = (1 - \eta(1-b))x_n + \eta a, \quad \text{with probability } \frac{1}{2}. \tag{3.5}$$

We first discuss the two simple symmetries that can be observed by a change of variables. First we observe that the dynamics (3.4)–(3.5) are invariant under the change $(a, b) \to (-a, -b)$ which swaps $(\phi_1, \phi_2) \to (\phi_2, \phi_1)$. The dynamics would not be invariant if the probabilities were not both 1/2.

Second, each of the equations (3.4)–(3.5) remain invariant under the change $(x, b) \to (-x, -b)$.

$$\frac{1}{2}(1+b)\,x^2 + ax \to \frac{1}{2}(1-b)\,x^2 - ax \tag{3.6}$$

$$\frac{1}{2}(1-b)\,x^2 - ax \to \frac{1}{2}(1+b)\,x^2 + ax \tag{3.7}$$

Due to the two symmetries, it is sufficient that when we are studying the behavior of solutions in later sections that we focus solely on the cases when $a \geq 0$, $b \geq 0$ and $\eta \geq 0$. Note that $\eta > 0$ is a problem restriction: when $\eta = 0$, the dynamics (3.4)–(3.5) just reduce to the (trivial) steady state $x_n = x_0$ for all n.

The values of a can further be broken into two cases: $a = 0$ and $a > 0$.

Case: $a = 0$ When $a = 0$, both equations (3.4)–(3.5) are gradient descents on a quadratic with effective time steps of $\eta(1 \pm b)$. This case is simple as both splitting equations simplify down to

$$f_1(x) = \frac{1}{2}(1+b)x^2, \quad f_2(x) = \frac{1}{2}(1-b)x^2. \tag{3.8}$$

which are just two parabolas centered around the origin. For this situation we note that $x = 0$ is a fixed point (although perhaps unstable) of the dynamics, which implies that $\rho(x) = \delta(x)$ is a stationary probability distribution for all b, η values. We do not consider this case any further.

Case: $a > 0$ In this case, we can rescale $x_n = a\tilde{x}_n$; the new dynamics on $\tilde{x}_n$ have a effectively set to $a = 1$. Therefore, without loss of generality we can take $a = 1$ from the outset.

In the subsequent studies, we will eventually take $a = 1$; the study of the quadratic stationary distributions is then reduced to cases where $\eta > 0$, $b \geq 0$.

3.2 Solution to Diffusion ODE for a Quadratic Problem

Now we solve the diffusion approximation ODE steady states for the choice of quadratic functions in (3.1). The diffusion approximation makes use of

$$v(x) = \frac{1}{2}\big(f_1'(x) + f_2'(x)\big) = \frac{(1+b)x + 1 + (1-b)x - 1}{2} = x,$$

and

$$\begin{aligned} M(x) &= \frac{(f_1'(x))^2 + (f_2'(x))^2}{2}, \\ &= \frac{((1+b)x+1)^2 + ((1-b)x-1)^2}{2} \\ &= (1+b^2)x^2 + 2bx + 1. \end{aligned} \tag{3.9}$$

The diffusion coefficient $D(x) = M(x) - v(x)^2$. We also have associated derivatives $v'(x) = 1$ and $M'(x) = 2(1+b^2)x + 2b$. Now plugging in to our general diffusion approximation solution (2.33)–(2.35) and integrating yields:.

$$\rho(x) = \begin{cases} \frac{C}{(bx+1)^{\alpha+1}} e^{\frac{-\beta}{(bx+1)}}, & x \geq -\frac{1}{b} \\ 0, & x < -\frac{1}{b}. \end{cases} \tag{3.10}$$

We defined temporary variables $\beta = \frac{a\left(\frac{2}{a}+1\right)}{b^2}$ and $\alpha = \frac{2}{b^2\eta} + \frac{1}{b^2} + 1$.

With respect to x (3.10) is the well-known *inverse gamma distribution*. That means that our coefficient is given by $C = \frac{\beta^\alpha}{\Gamma(\alpha)}$ which we also must scale by a factor of b in order to make it appropriately integrate to 1. From here we will also for brevity drop the piece-wise definition, however it is understood that points outside the definition we set to 0. With all of this our final distribution is

$$(b \neq 0) \qquad \rho(x) = \frac{b\beta^\alpha}{\Gamma(\alpha)(bx+1)^{\alpha+1}} e^{\frac{-\beta}{(bx+1)}}. \tag{3.11}$$

The coefficients α and β in (3.11) become singular when $b = 0$. In this case, (2.33)–(2.35) can be reapplied from the start with $b = 0$ to obtain the final solution

$$(b = 0) \qquad \rho(x) = \sqrt{\frac{\beta}{\pi}} e^{-\beta x^2}, \tag{3.12}$$

where $\beta = \frac{2+\eta}{2\eta}$.

When $b = 0$, $\rho(x)$ has infinite support on both sides, and as $\eta \to 0$ we find $\rho(x) \to \delta(x)$ and as $\eta \to \infty$, $\rho(x) \to \sqrt{\frac{1}{2\pi}} e^{-\frac{1}{2}x^2}$.

We now look at the more interesting case of when $b \neq 0$. To keep consistent with the literature for the inverse gamma distribution we have that

$$\beta = \frac{3\eta}{b^2}, \alpha = \frac{1}{b^2}(2/\eta + 1) + 1, u = bx + 1 \tag{3.13}$$

so our distribution is

$$\rho(u) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} u^{-\alpha-1} e^{-\frac{\beta}{u}}. \tag{3.14}$$

Since $\rho(u)$ is a common and well understood distribution, we can see that $\rho(x)$ has infinite support on the region $(-\frac{1}{b}, \infty)$.

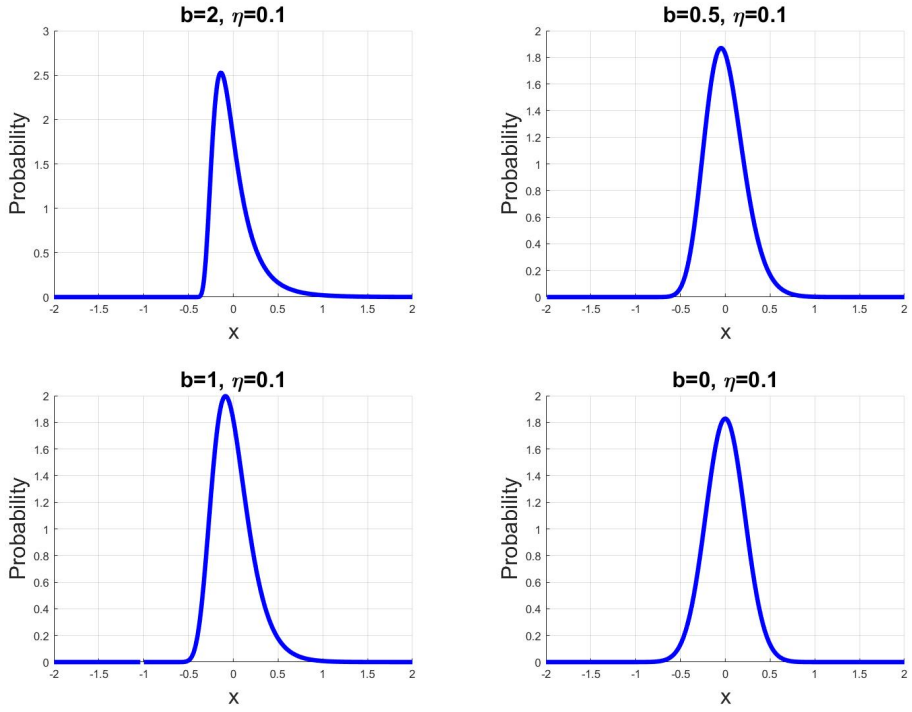


Figure 3.1 Figures showing differences in Diffusion Equation solution behavior when you vary values of b

3.3 Stationary Probability Distributions of the Markov Operator

In this section we examine the stationary probability distributions to the Markov operator for varying (b, η). In most cases, the distributions are computed numerically via Ulam's method, however for specific (b, η) parameter values we will provide exact formulas for the distributions.

In this section we also provide boundaries of different solution behaviors. We provide conditions on (b, η) such that there exists a trapping region, which implies the

stationary distribution is finitely supported. We also explain conditions on regions such that the stationary distribution is smooth. Finally we prove the expected value of the stationary distribution for all values of (b, η) to be equivalently 0.

In this section we let $a = 1$ in (3.1), so that

$$f_1(x) = \frac{1}{2}(1+b)x^2 + x, \qquad f_2(x) = \frac{1}{2}(1-b)x^2 - x, \tag{3.15}$$

with corresponding gradient descents

$$\varphi_1(x) = (1-(1+b)\eta)x - \eta, \qquad \varphi_2(x) = (1-(1-b)\eta)x + \eta, \tag{3.16}$$

and associated inverse functions

$$\varphi_1^{-1}(x) = \frac{x+\eta}{1-(1+b)\eta}, \quad \text{when } 1-(1+b)\eta \neq 0, \tag{3.17}$$

$$\varphi_2^{-1}(x) = \frac{x-\eta}{1-(1-b)\eta}, \quad \text{when } 1-(1-b)\eta \neq 0. \tag{3.18}$$

In this case, the Markov operator equation (2.9) becomes:

$$\rho_{n+1}(\vec{x}) = \frac{1}{2}\int_{\mathbb{R}} \left[\delta(x-\varphi_1(y)) + \delta(x-\varphi_2(y))\right] \rho_n(y)\,\mathrm{d}y. \tag{3.19}$$

The stationary solutions to (3.19) are then obtained when $\rho_{n+1} = \rho_n =: \rho$. We can derive an alternative equation for $\rho(x)$ by integrating the δ-distribution out in (2.9) to obtain

$$\rho(x) = \frac{1}{2}\left(\frac{\rho(\varphi_1^{-1}(x))}{|1-(1+b)\eta|}\right) + \frac{1}{2}\left(\frac{\rho(\varphi_2^{-1}(x))}{|1-(1-b)\eta|}\right), \quad \text{when } (1\pm b)\eta \neq 1. \tag{3.20}$$

Note that (3.20) conserves probability — the action of φ_j^{-1} composed with ρ_n (in the right hand side) contains a horizontal dialation and shift, which is balanced by a vertical dialation to conserve the total probability. Furthermore, (3.20) is understood in a weak sense, and does not need to hold pointwise for every value of x: Namely

(3.20) is understood in the sense that for any interval (a, b)

$$\int_a^b \rho(x)\,\mathrm{d}x = \int_a^b \frac{1}{2}\left(\frac{\rho(\varphi_1^{-1}(x))}{|1-(1+b)\eta|}\right) + \frac{1}{2}\left(\frac{\rho(\varphi_2^{-1}(x))}{|1-(1-b)\eta|}\right)\,\mathrm{d}x. \tag{3.21}$$

The case when equation (3.20) fails, i.e., $(1 \pm b)\eta = 1$, is discussed further in §3.3.2.

In the remaining subsections, we investigate both numerically and analytically the stationary probability distributions to (3.19) for varying $\eta > 0$ and $b \geq 0$. We first present several exact stationary distributions for special η, b values. These solutions highlight that stationary distributions can be both classical (L^1) functions or singular probability distributions (with Dirac masses). We then characterize the support of ρ in different parameter regimes, showing that ρ generally has compact support for sufficiently small η values. Finally, we provide a phase diagram (for b vs η) with numerically computed stationary distributions.

3.3.1 Exact Stationary Probability Distribution

In this subsection we present exact stationary probability distributions for specific b and η values.

Case 1: $(\eta, b) = (\frac{1}{2}, 0)$. One particularly interesting situation arises when we have $\eta = \frac{1}{2}$ and $b = 0$, for which $\varphi_1^{-1}(x) = 2x + 1$ and $\varphi_2^{-1}(x) = 2x - 1$ and hence (3.20) becomes:

$$\rho(x) = \rho(2x + 1) + \rho(2x - 1). \tag{3.22}$$

By direct calculation we see that the uniform distribution:

$$\rho(x) = \begin{cases} \frac{1}{2} & \text{when } |x| < 1, \\ \frac{1}{4} & \text{when } |x| = 1, \\ 0, & \text{when } |x| > 1. \end{cases} \tag{3.23}$$

satisfies (3.22) (exactly for every x).

Case 2: $(\eta, b) = (\frac{3}{2}, 0)$. This case is nearly identical to Case 1, however we now have to be mindful of the absolute value in the denominator of equation (3.20). Note that here $\varphi_1^{-1}(x) = -2x + 3$ and $\varphi_2^{-1}(x) = -2x - 3$, so that (3.20) becomes:

$$\rho(x) = \frac{1}{2|(1 - \frac{3}{2})|} \left(\rho(\varphi_1^{-1}(x)) + \rho(\varphi_2^{-1}(x))\right), \tag{3.24}$$

$$\rho(x) = \rho(-2x + 3) + \rho(-2x - 3). \tag{3.25}$$

By direct computation again we see that a uniform distribution on the range $(-3, 3)$ satisfies this equation:

$$\rho(x) = \begin{cases} \frac{1}{6} & \text{when } |x| < 3, \\ \frac{1}{12} & \text{when } |x| = 3, \\ 0, & \text{when } |x| > 3. \end{cases} \tag{3.26}$$

This solution is interesting as $\{-1, 1\}$ are the minima of the functions in (3.15), so while inside the range $[-1, 1]$, we cannot guarantee the iterates of SGD stay there.

Case 3: $(\eta, b) = (1, 0)$. In the unique situation in which $b = 0, \eta = 1$ we find that the dynamics are

$$\begin{aligned} \varphi_1(x) &= (1 - (1 + b)\eta)x - \eta = -1 \\ \varphi_2(x) &= (1 - (1 - b)\eta)x + \eta = 1. \end{aligned} \tag{3.27}$$

This means that regardless of x_0, every value of SGD for $m \geq 1$ satisfies $x_m \in \{-1, 1\}$, i.e., the state space is two dimensional and the Markov operator reduces down to a two dimensional Markov matrix. To make the correspondence precise, introduce the vector $\vec{p}_m$ such that:

$$\vec{p}_m = \begin{pmatrix} p_m^1 \\ p_m^2 \end{pmatrix} := \begin{pmatrix} \Pr(x_m = -1) \\ \Pr(x_m = 1). \end{pmatrix}, \qquad \rho_m(x) = p_m^1 \delta(x+1) + p_m^2 \delta(x-1). \quad (3.28)$$

The Markov operator then becomes:

$$\vec{p}_{m+1} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \vec{p}_m. \quad (3.29)$$

The stationary distribution in vector form is $\vec{p} = (1/2, 1/2)^T$, or equivalently:

$$\rho(x) = \frac{1}{2} \left(\delta(x-1) + \delta(x+1) \right). \quad (3.30)$$

3.3.2 Death–Respawn Markov Dynamics when $\eta(1 \pm b) = 1$

The Markov dynamics simplify to a state space which is (effectively) countable and a corresponding Markov chain that can be characterized as a "death" and "respawn" model in two special cases. Specifically, when $\eta = (1+b)^{-1}$ $(b \geq 0)$ the dynamics become:

$$\varphi_1(x) = (1 - (1+b)\eta)x - \eta = -\frac{1}{1+b}, \quad (3.31)$$

$$\varphi_2(x) = (1 - (1-b)\eta)x + \eta = \frac{2b}{1+b}x + \frac{1}{1+b}. \quad (3.32)$$

Similarly, when $\eta = (1-b)^{-1}$, $(0 < b < 1)$ the dynamics become

$$\begin{aligned}\varphi_1(x) &= (1-(1+b)\eta)x - \eta = -\frac{2b}{1-b}x - \frac{1}{1-b},\\ \varphi_2(x) &= (1-(1-b)\eta)x + \eta = \frac{1}{1-b}.\end{aligned} \tag{3.33}$$

In both cases, one of the two maps resets x to a constant, so that at every time step there is a probability of 0.5 that the point x_m "dies" and respawns at a point x_0 (either $-(1+b)^{-1}$ or $(1-b)^{-1}$). In this case the stationary probability will be supported on a discrete (countable) set of points with Dirac masses (with weights that can be determined explicitly).

3.3.3 Stationary Distributions with Compact Support

In this subsection we establish bounds for the support of stationary distributions $\rho(x)$ — which roughly speaking are the points where $\rho(x)$ is "non-zero". A point x is in the *support* of the probability $\rho(x)$, written as $x \in \text{supp}(\rho)$, if for every $\epsilon > 0$ the integral is (strictly) positive:

$$\int_{x-\epsilon}^{x+\epsilon} \rho(x)\,\mathrm{d}x > 0. \tag{3.34}$$

To establish bounds on the support, we solve for trapping regions $U \subset \mathbb{R}^d$ of the SGD dynamics. A trapping region is a set such that if $\vec{x}_m \in U$ enters, then the SGD dynamics $\{x_{m+1}, x_{m+2}, \ldots\}$ can never exit. Formally, they are defined as follows.

Definition 1. *(Trapping region for SGD) A set U is a trapping region for the SGD dynamics (or equivalently WSGD) if:*

$$\varphi_j(U) \subseteq U, \quad \textit{for all } 1 \le j \le n.$$

The notation $\varphi(U) := \{\varphi(\vec{x}) : \vec{x} \in U\}$.

Trapping regions are significant since they trap the probability dynamics under the Markov operator, and provide bounds on the support of stationary distributions.

Proposition 1. *(Trapping regions trap supp(ρ_m)) Let U be a closed set and a trapping region, and suppose that supp(ρ_m) $\subseteq U$. Then ρ_{m+1} defined by the Markov dynamics* (3.19) *(and more generally* (2.8)*) has supp(ρ_{m+1}) $\subseteq U$.*

Proof. If $\text{supp}(\rho_m) \subseteq U$, then by direct evaluation of the integral in (3.19), $\text{supp}(\rho_{m+1})$ is contained in $\cup_j \varphi_j(U)$ for $j = 1, 2$ in (3.19) (and more generally $1 \leq j \leq n$ for (2.8)). Since each $\varphi_j(U) \subseteq U$ (U is a trapping region) we have that $\text{supp}(\rho_{m+1}) \subseteq U$. Q.E.D.

We now determine the smallest closed trapping regions of the form $U = [A, B]$ (with $B > A$) for the dynamics (3.16). Note that the case $(\eta, b) = (1, 0)$ is handled in §3.3.1 which solves the stationary probability exactly, so we disregard it here.

Case 1: $b = 0, 0 < \eta < 1.$ Then $U = [-1, 1]$ is a trapping region.

Here $0 < (1 - \eta) < 1$ which is the x-coefficient in $\varphi_1(x) = (1 - \eta)x - \eta$ and $\varphi_2(x) = (1 - \eta)x + \eta$ and squeezes any set $\varphi_1(U)$, $\varphi_2(U)$. Hence, a set $U = [-L, L]$ under either mapping $\varphi_1(U)$ or $\varphi_2(U)$ has has a right boundary of $(1 - \eta)L + \eta$, so that we require $(1-\eta)L+\eta \leq L$. Thus any value of $L \geq 1$ yields a trapping region — the smallest such value is $L = 1$. Note that the dynamics are invariant under $x \to -x$ so we get the left boundary for free. It is interesting that the trapping region bound is $L = 1$ for all values of $0 < \eta < 1$.

Case 2: $b = 0, 1 < \eta < 2.$ Then $U = [-L, L]$, for $L = \frac{\eta}{2-\eta}$ is a trapping region. Since $\eta > 1$, the coefficient in front of x will be negative. Therefore the largest value on the right for will occur at the end of our support interval of opposite sign. For $U = [-L, L]$, we want to solve the equation $-L = \varphi_1^{-1}(L)$:

$$-L = \frac{L}{1 - \eta} + \frac{\eta}{1 - \eta}. \tag{3.35}$$

The solution to this equation has the form $L = \frac{\eta}{2-\eta}$. It is interesting that the set U approaches infinity as $\eta \to 2$.

Case 3: $b = 0, \eta \geq 2$. There is no trapping region of the form $U = [-L, L]$ for $L > 0$. Suppose that there was a bounded region $U = [-L, L]$ of support when $\eta \geq 2$. We know that one of our dynamics equations would be of the form $\varphi(x)_1 = (1-\eta)x + \eta$. Since $\eta \geq 2$ we know that the coefficient in front of our value of x will be ≤ -1. For all possible values of η, we will attain $\max_{x \in U} \varphi_1(x)$ when $x = -L$. Plugging in we find $\varphi_1(-L) = (\eta - 1)L + \eta \geq (2-1)L + 2 = L + 2$, holds $\forall L \in \mathbb{R}_+$. Since $L + 2 > L$ there is not a finite trapping region interval. For these parameter values, we observe numerical that the stationary probability appear to have (unbounded) support on $\mathbb{R}$.

Case 4: General case. The set $U = [-L, L]$ with $L > 0$ is (in general) a trapping region if $(-L, L)$ satisfy:

$$-L \leq \varphi_j(-L) \leq L, \quad \text{and} \quad -L \leq \varphi_j(L) \leq L, \quad \text{hold for } j = 1, 2. \tag{3.36}$$

Since $\varphi_j(x)$ for $j = 1, 2$ are both linear functions, (3.36) constitutes 8 linear inequalities, for which there is a feasible solution only for certain b and η values. Cases 1–3 summarize the "tightest" solutions when $b = 0$. Note that even in the non-symmetric case if the trapping region is non symmetric, we could contain it within a larger symmetric region.

We begin by looking at the dynamic equation $\varphi_1(x)$. First, we consider the case in which $[1-(1+b)\eta] > 0$. In this case the sign of the input is preserved upon scaling, so we only need to consider 1 active inequality in that $-L \leq \varphi_1(-L)$. Solving the inequality we have that

$$
\begin{aligned}
-L &\leq [1-(1+b)\eta](-L) - \eta, \quad \text{when } \eta > \frac{1}{1+b} \\
\eta &\leq [1-1+(1+b)\eta]L, \\
L &\geq \frac{1}{1+b}.
\end{aligned} \tag{3.37}
$$

Solving for equality, we can see that taking $L = (1+b)^{-1}$ yields a trapping region whenever $\eta < \frac{1}{1+b}$ (note that we satisfy the inequality $\varphi_2(L) \leq L$ for free).

The case when $[1-(1+b)\eta] = 0$ is handled in §3.3.2.

We now look at the more interesting cases of when $[1-(1+b)\eta] < 0$. For this, we now must solve the active inequality that $-L \leq \varphi_1(L)$ as now at each iteration the input value changes sign upon scaling. Solving the inequality we find that our equation becomes

$$
\begin{aligned}
-L &\leq [1-(1+b)\eta](L) - \eta \\
\eta &\leq [2-(1+b)\eta](L) \\
L &\geq \frac{\eta}{2-(1+b)\eta}.
\end{aligned} \tag{3.38}
$$

So in this case our interval of support $\left[-\frac{\eta}{2-(1+b)\eta}, \frac{\eta}{2-(1+b)\eta}\right]$. Note, however, since we assumed $L > 0$, that if the coefficient $[2-(1+b)\eta] \leq 0$, we will get that our inequality cannot be satisfied. Therefore we find that if $\eta \geq \frac{2}{1+b}$ then we cannot be sure that there exists a finite trapping region of the form $[-L, L]$. Note that we conjecture that the stationary distribution in this region has infinite support provided it exists.

Next we observe the cases of $\varphi_2(x)$, of which the first few cases will be similar. The difference now though is that since $b > 0$ we had that $1+b$ was strictly positive, however now we can have this term be negative for $1-b$ which leads to some new cases.

First, assume that $[1-(1-b)\eta] > 0$. In this situation our maximum arises when $x = L$, so we only need to solve the active inequality that $\varphi_2(L) \leq L$,

$$\begin{aligned} L &\geq [1-(1-b)\eta]L + \eta \\ -\eta &\geq -(1-b)\eta L \\ L &\geq \frac{1}{1-b} \end{aligned} \tag{3.39}$$

which gives us a region of $\left[-\frac{1}{1-b}, \frac{1}{1-b}\right]$. However, we now must also notice that if $[1-(1-b)\eta] = \gamma \geq 1$ then we have the inequality

$$0 \geq (\gamma - 1)L + \eta, \tag{3.40}$$

which is impossible to satisfy. As such, we know that in this case we cannot be sure there exists a finite trapping region of format $[-L, L]$. In terms of the parameters (b, η), this will occur if $b \geq 1$.

As prior, when $[1-(1-b)\eta] = 0$ we have a trivial case in which $\varphi_2(x)$ evaluates to a constant. In this situation the invariant measure is finitely supported in any interval that contains $\varphi_2(x)$.

Finally we consider what happens when $[1-(1-b)\eta] < 0$. We proceed similar to how we proceeded before, and consider the active condition that $\varphi_2(-L) \leq L$ since the negative coefficient will swap the sign of the input. Solving the inequality we find that

$$\begin{aligned} L &\geq [1-(1-b)\eta](-L) + \eta \\ \eta &\leq [2-(1-b)\eta](L) \\ L &\geq \frac{\eta}{2-(1-b)\eta}. \end{aligned} \tag{3.41}$$

Similar to the previous case, this is guaranteed to be finitely supported when $[2-(1-b)\eta] < 0$. In the case this condition is not satisfied we again cannot be sure that we have a region of finite support, which is when $\frac{2}{1-b} \leq \eta$.

3.3.4 Necessary Conditions for a Continuous Distribution

In this section we discuss a transition in the Markov operator dynamics that plays a role in separating probability distributions for which $\rho(x)$ is continuous on all of $\mathbb{R}$ and those which are non-continuous or have singular Dirac masses.

Case 1: $b = 0, \frac{1}{2} < \eta < 1.$ The Markov dynamics reduce to

$$\rho_{n+1}(x) = \frac{1}{2(1-\eta)} \left(\rho_n(\varphi_1^{-1}(x)) + \rho_n(\varphi_2^{-1}(x)) \right), \tag{3.42}$$

where the coefficient on the right hand side is $\gamma := \frac{1}{2(1-\eta)} > 1$. Moreover both φ_1^{-1} and φ_2^{-1} have a linear term $\sim (1-\eta)^{-1}x$ which acts to squeeze the x-axis by a factor of $(1-\eta) < 1$. Therefore, each term on the right hand side of equation (3.42) undergoes two actions: (i) the probability ρ_n is squeezed by a factor $(1-\eta)$; and (ii) stretched by a factor of $\gamma > 1$. Moreover, the dynamics admit a trapping region $U = [-1, 1]$.

Proposition 2. *(Necessary condition for a continuous $\rho(x)$) Suppose that $\rho(x)$ is a continuous probability density function on all of $\mathbb{R}$ with support in $[-1, 1]$. Then $\rho(x)$ cannot be a stationary solution of* (3.42) *in a strong point-wise sense.*

Proof. Assume that $\rho(x)$ is a stationary distribution of the equation

$$\rho(x) = \gamma \left(\rho(\varphi_1^{-1}(x)) + \rho(\varphi_2^{-1}(x)) \right), \tag{3.43}$$

for $\gamma > 1$. Define $\rho_{max} := \rho(x^*)$ (which is bounded) where $x^* = \text{argmax}_{|x|\leq 1}\rho(x)$ (x^* exists by the extreme value theorem since $\rho(x) = 0$ outside of $[-1, 1]$ and continuous everywhere). Then equation (3.43) fails to hold at $x = \varphi_1(x^*)$ since:

$$\gamma\left(\rho(\varphi_1^{-1}(x)) + \rho(\varphi_2^{-1}(x))\right) \geq \gamma\rho_{max} > \rho_{\max} \geq \rho(\varphi_1(x^*)), \tag{3.44}$$

Q.E.D.

Case 2: General case. We showed that in Case 1 our steady state solution becomes "unstable" in that we get that the probability measures grow unbounded under the Markov dynamics, implying the failure of strong continuous solutions. In the event that $0 < b$ we will have that depending on our choice of η only one of our two functions φ_1 or φ_2 will behave in this way (note that the proof of Prop. 2 only required one such term). We use the equation (3.19) and see that the coefficients

$$-1 < \frac{1}{2(1-(1\pm b)\eta)} < 1 \tag{3.45}$$

determine the amplification of each term. Let $c = 1 \pm b$. If we adjust this to be an inequality for η in terms of b we find that $|2(1 - c\eta)| > 1$:

$$\implies \quad 2 - 2c\eta < -1, \quad \text{or} \quad 1 < 2 - 2c\eta, \tag{3.46}$$

$$\implies \quad c\eta > \frac{3}{2}, \quad \text{or} \quad c\eta < \frac{1}{2} \tag{3.47}$$

A necessary condition for continuous stationary solutions is then

$$\left\{(1+b)\eta > \frac{3}{2}, \text{ or } (1+b)\eta < \frac{1}{2}\right\} \quad \text{and} \quad \left\{(1-b)\eta > \frac{3}{2}, \text{ or } \quad (1-b)\eta < \frac{1}{2}\right\}. \tag{3.48}$$

With these inequalities we find that there exists a band of non-continuous solutions that appears to be centered around the curves defined by the Death-Respawn Dynamics from §3.3.2. This band is visualized in the next section Figure 3.2

3.3.5 Phase Plot of Different True Solution Behaviors

In order to show how different values of b and η affect the behavior of the true solution that we compute with Ulam's method, we create a phase diagram of several different examples.

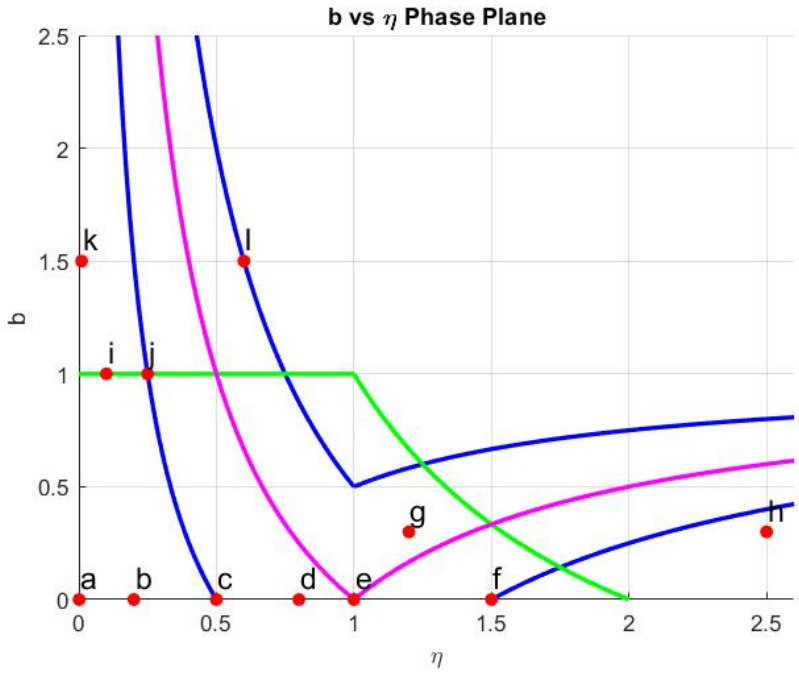
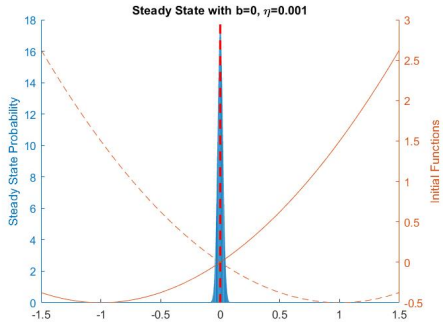
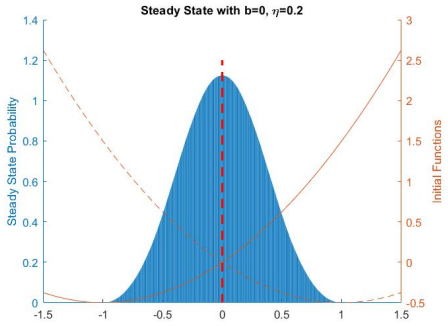


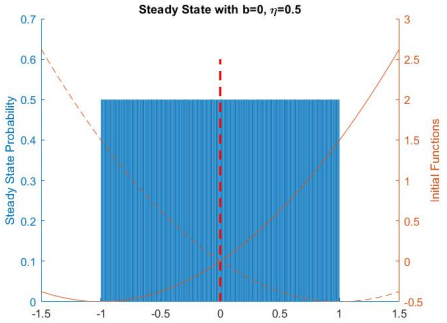
Figure 3.2 Phase plot showing the various behaviors of the stationary probability distributions to (3.19) (true solutions) with respect to different values of η and b. The magenta lines correspond to parameter values exhibiting death-respawn dynamics in §3.3.2. The blue curves around the magenta curve defines the boundary of parameter values for $\rho(x)$ to be discontinuous function as defined in the necessary condition §3.3.4. The region defined to the bottom-left of the green curves represents the boundary for which the SGD dynamics admit a trapping region $U = [-L, L]$ for a finite value of $L \geq 0$ as outline in §3.3.3. The letters on the scatter plot points correspond to the associated figure labeled in Fig. 3.2.


Steady State with b=0, η=0.001
Steady State Probability
Initial Functions

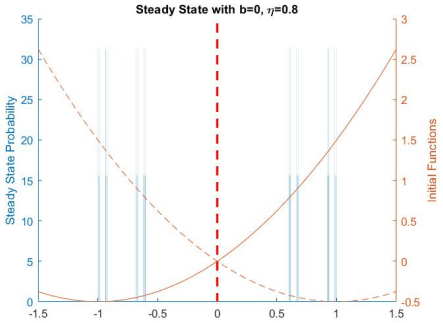
(a)


Steady State with b=0, η=0.2
Steady State Probability
Initial Functions

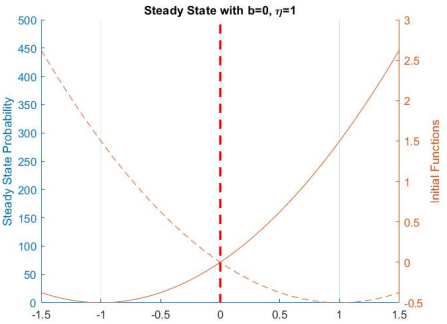
(b)


Steady State with b=0, η=0.5
Steady State Probability
Initial Functions

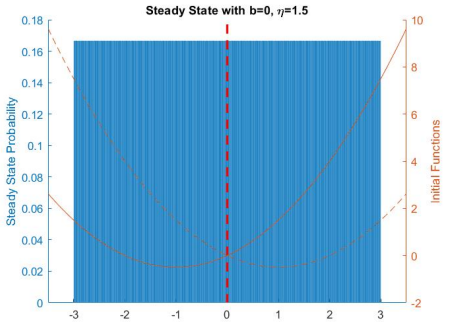
(c)


Steady State with b=0, η=0.8
Steady State Probability
Initial Functions

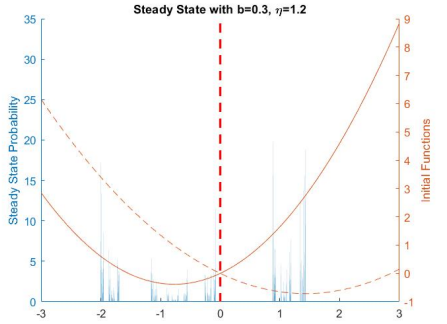
(d)


Steady State with b=0, η=1
Steady State Probability
Initial Functions

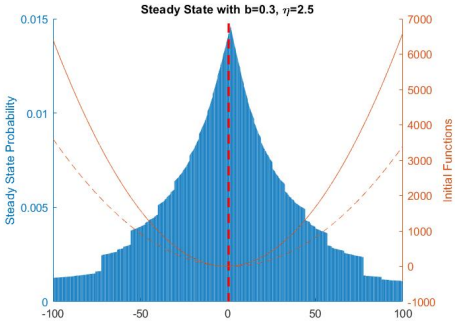
(e)


Steady State with b=0, η=1.5
Steady State Probability
Initial Functions

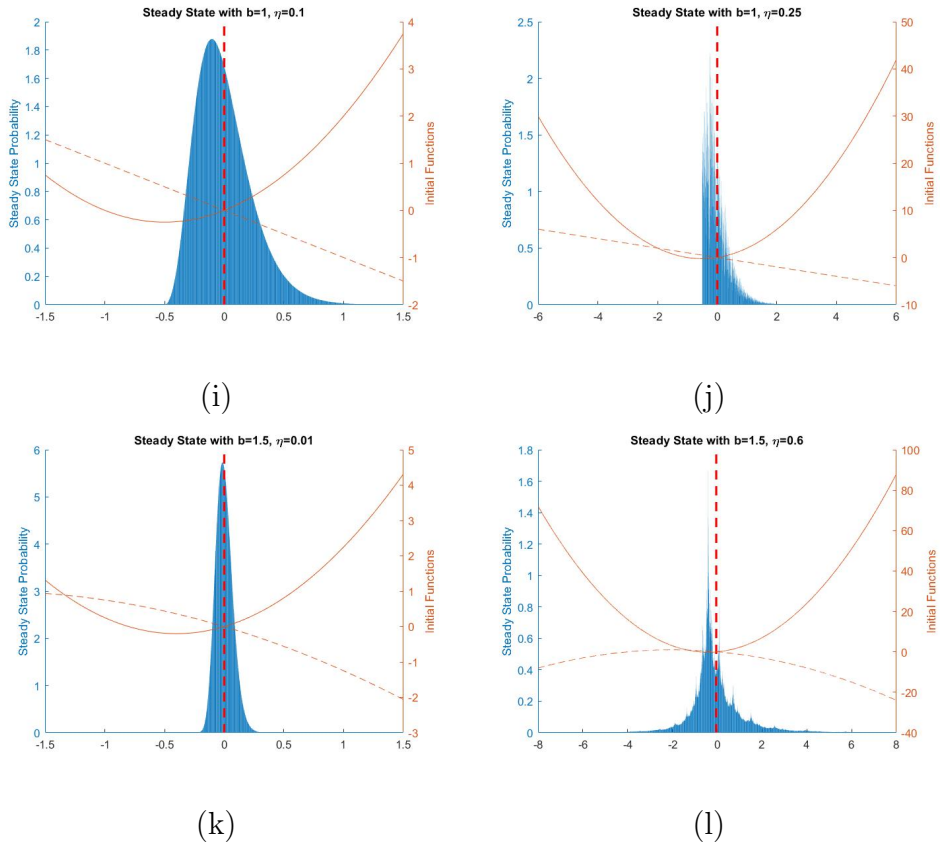
(f)


Steady State with b=0.3, η=1.2
Steady State Probability
Initial Functions

(g)


Steady State with b=0.3, η=2.5
Steady State Probability
Initial Functions

(h)



(i) (j)

(k) (l)

Figure 3.2 Plots of the various invariant measures for different values of b, η in the quadratic case. Notice that as η passes a critical line as shown in the phase plots that the solutions become unstable.

3.4 Convergence Study of the Diffusion Approximation for Stationary Probabilities

Now that we have a closed form expression that solves the steady state of the diffusion approximation, it becomes a necessary question to ask how well do these solutions approximate the true steady state distribution that is provided by the Markov Operator? In this section we compare the true solutions of the Stationary Distributions to the approximations that we compute by solving the ODE equations.

For our convergence study we select a 2 function splitting of a 1-D quadratic equation, specifics of which are described in more detail in further sections, and

compare the results of what our differential equations believe the steady states to be, versus what we gather as the true steady state solution. To compute the true solution we use Ulam's Method to compute the invariant measure of the Markov operator.

Since we can think of the invariant measure of our problem as the probability distribution of where some initial data point will lie given a long period of time, we can perform a particle simulation of several thousand points, which due to the Law of Large Numbers, should agree with the invariant measure that we get from Ulam's Method.

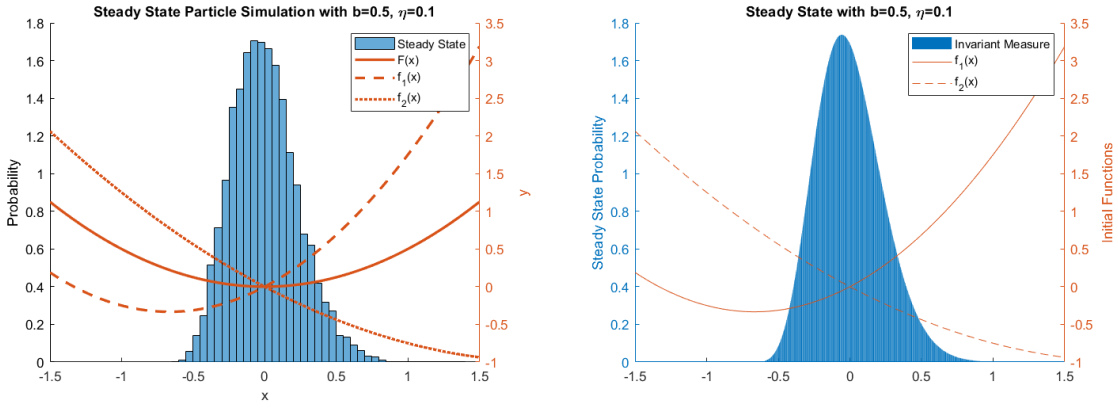


Figure 3.3 Figures showing a comparison between the invariant measures computed by Ulam's method on the right and particle simulations on the left. As you increase the number of particles run, the shape of the particle solution becomes smoother to match what is computed by Ulam's method.

We now compare the exact result computed by Ulam's method to the approximation equations derived above to see how well they match the true invariant measure.

As we can see in the above figures that the approximation ODEs do not really do that good of a job approximating the true behavior of the solution for values of η that are "large". These equations do hold in the limiting case, however the non-limiting cases leave more to be desired.

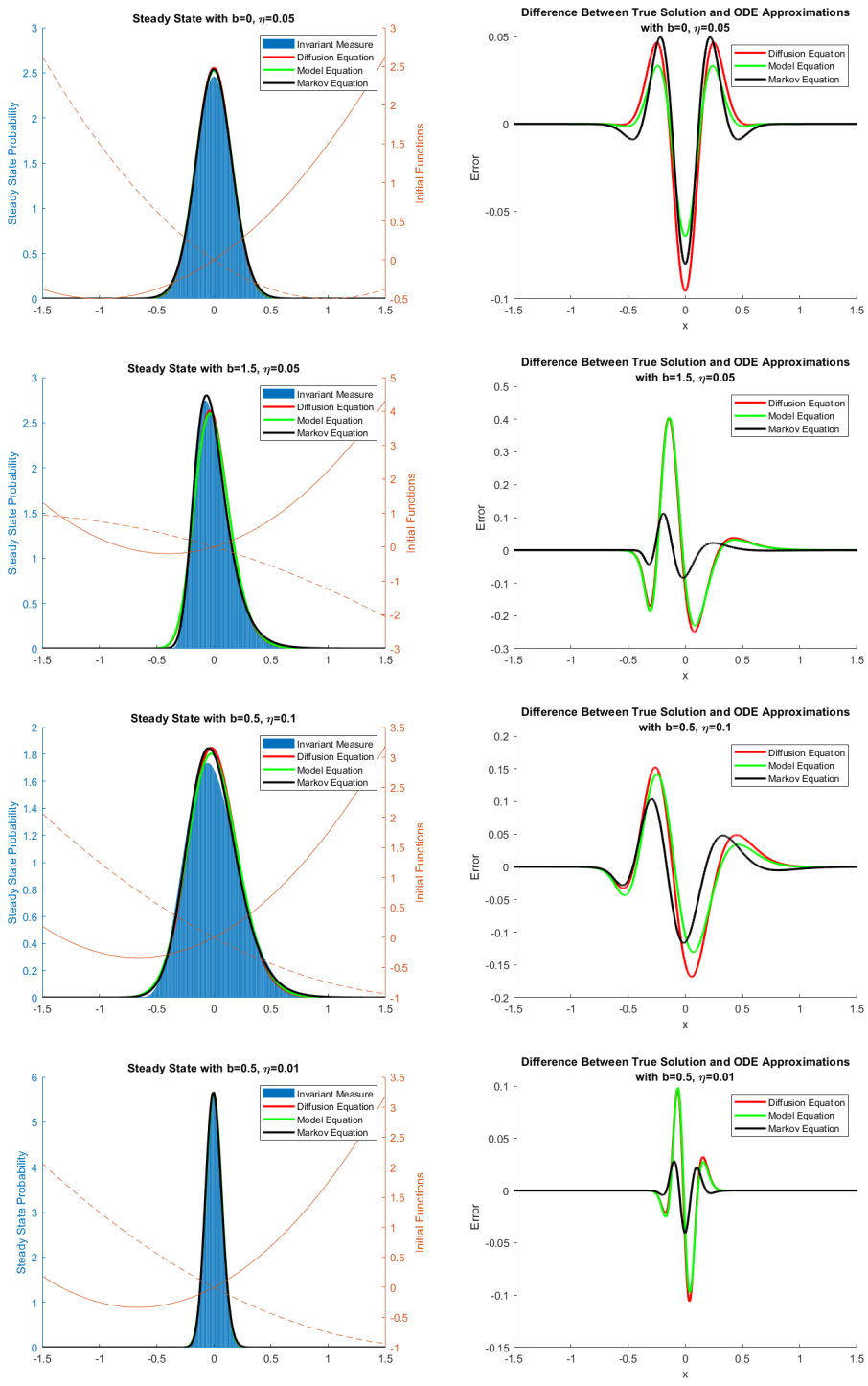


Figure 3.4 Plots of the various computed ODE solutions vs. the true invariant measure computed by Ulam's method on the right. The figures on the left are the associated error values of the true solution vs the approximation solution. Visually we can see that the solutions are quite different then the true values.

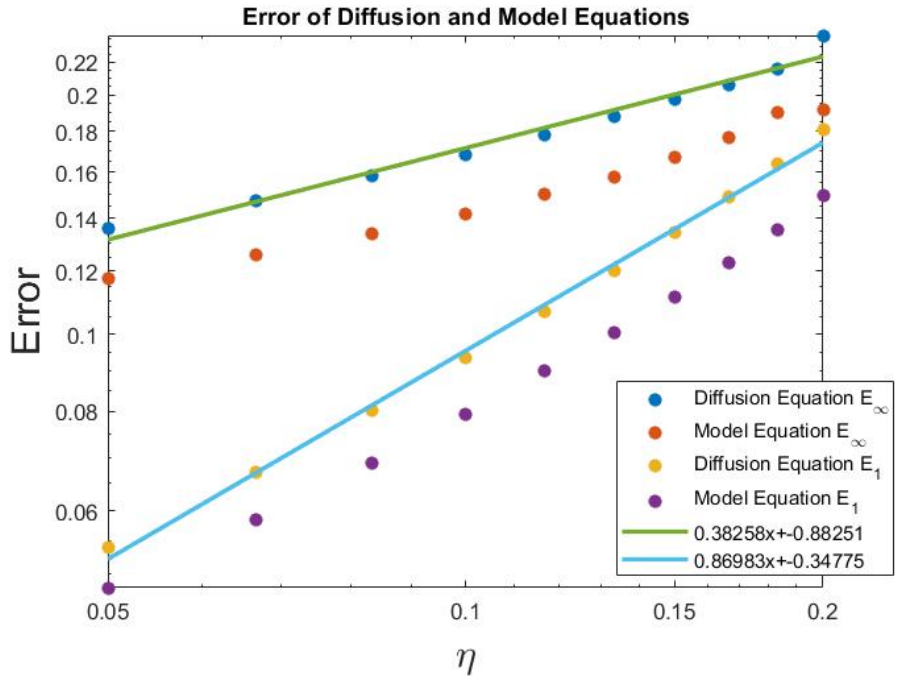


Figure 3.5 Plot of convergence of the Diffusion and Model equations as η grows smaller.

Plotting the error of the the Model and Diffusion Equations we find that the error converges, but quite slowly (the numerics suggest a rate slower than $\sqrt{\eta}$) in the L_∞ sense and (slower than η) in the L_1 sense. As stated above, the slow L_∞ convergence produces noticeable observed differences in the diffusion approximation probability distribution (even when $0.1 < \eta < 1$).

3.5 Expected Value of Generalized Quadratic Problem

Practically it is desirable to run SGD with large step sizes of η to speed up convergence to the optima. In this section we prove the expected value of our parameters converges to the optimum value of our objective function regardless of the step size.

To compute the expected value of the invariant measure of our stochastic, we first consider the most generalized variation of stochastic gradient descent in the quadratic case.

Let our cost function be $f(x) = x^2$ with a set of n quadratic splitting functions $\mathscr{F}$ chosen with equal probability such that

$$x^2 = \sum_{f_i \in \mathscr{F}} f_i(x) = \sum_{f_i \in \mathscr{F}} (b_i x^2 + a_i x) \tag{3.49}$$

in which we have probability p_i selection chance for each function. Notice that $\sum_{i=0}^{n} p_i b_i = 1$ and $\sum_{i=0}^{n} p_i a_i = 0$, which comes from the fact that the expected value of the function that we select is x^2.

Theorem 2. *The expected value of the invariant probability measure $\rho(x)$ of stochastic gradient descent performed by selecting any $f_i \in \mathscr{F}$ with weighted probability p_i is $E(x) = 0$*

Proof. We begin by first creating the $\varphi_i(x)$ functions that we will randomly choose from at every iteration. Following the standard form from equation 1.5, we know that

$$\varphi_i(x) = x - \eta(\nabla f_i) = (1 - 2b_i\eta)x - a_i\eta, \tag{3.50}$$

with associated inverse function

$$\varphi_i(x)^{-1} = \frac{x}{(1 - 2b_i\eta)} + \frac{a_i\eta}{(1 - 2b_i\eta)}. \tag{3.51}$$

With these dynamics equations, we can now apply the Perron-Frobenious operator to generalize equation 3.19 for our set of n dynamics equations. Remember that we are selecting each possible $\varphi_i(x)$ with equal probability $p = \frac{1}{n}$. If we say that the invariant measure is the function $\rho_n(x)$ such that $\rho_{n+1}(x) = \rho_n(x) = \rho(x)$, then we get the following equation

$$\rho(x) = \sum_{i=0}^{n} p_i \frac{\rho(\varphi_i(x)^{-1})}{|1 - 2b_i\eta|} = \sum_{i=0}^{n} p_i \frac{\rho\left(\frac{x}{(1-2b_i\eta)} + \frac{a_i\eta}{(1-2b_i\eta)}\right)}{|1 - 2b_i\eta|}. \tag{3.52}$$

For ease of notation, let us define the following constants

$$\gamma_i = \frac{1}{(1-2b_i\eta)},\ \delta_i = \frac{a_i\eta}{(1-2b_i\eta)} = a_i\eta\gamma_i, \tag{3.53}$$

which simplifies our summation to

$$\rho(x) = \sum_{i=0}^{n} p_i|\gamma_i|\rho(\gamma_i x + \delta_i). \tag{3.54}$$

Multiplying both sides of the equation by x and integrating from $-\infty$ to ∞ we get

$$\int_{-\infty}^{\infty} x\rho(x)dx = \sum_{i=0}^{n} \int_{-\infty}^{\infty} p_i|\gamma_i|x\rho(\gamma_i x + \delta_i)dx. \tag{3.55}$$

$\int_{-\infty}^{\infty} x\rho(x)dx = E(\rho(x))$ by definition. We will say that $\mu = E(\rho(x))$ for simplicity purposes.

Note that because we are working with finitely sized sums, we can confidently switch the summation and the integral. We now perform the "greatest trick in mathematics" and multiply each term by $\frac{\gamma_i}{\gamma_i}$ and add $\delta_i - \delta_i$ to get the following equation.

$$\mu = \sum_{i=0}^{n} \frac{p_i|\gamma_i|}{\gamma_i} \int_{-\infty}^{\infty} (\gamma_i x + \delta_i - \delta_i)\rho(\gamma_i x + \delta_i)dx. \tag{3.56}$$

We can now perform some algebraic manipulations to get these integrals into some easily integrable forms.

$$\mu = \sum_{i=0}^{n} \frac{p_i|\gamma_i|}{\gamma_i} \int_{-\infty}^{\infty} (\gamma_i x + \delta_i - \delta_i)\rho(\gamma_i x + \delta_i)dx$$
$$\mu = \sum_{i=0}^{n} \frac{p_i|\gamma_i|}{\gamma_i} \int_{-\infty}^{\infty} (\gamma_i x + \delta_i)\rho(\gamma_i x + \delta_i)dx - \sum_{i=0}^{n} \frac{p_i|\gamma_i|\delta_i}{\gamma_i} \int_{-\infty}^{\infty} \rho(\gamma_i x + \delta_i)dx \tag{3.57}$$
$$\mu = \sum_{i=0}^{n} \frac{p_i|\gamma_i|}{\gamma_i^2}\mu - \sum_{i=0}^{n} \frac{p_i|\gamma_i|\delta_i}{\gamma_i|\gamma_i|} = \sum_{i=0}^{n} \frac{p_i|\gamma_i|}{\gamma_i^2}\mu - \sum_{i=0}^{n} \frac{p_i\delta_i}{\gamma_i}.$$

We now can simply rearrange the equation to solve for μ, and replacing γ_i, δ_i with their original values we find

$$\begin{aligned}
\mu &= -\frac{\sum_{i=0}^{n} \frac{p_i\delta_i}{\gamma_i}}{1 - \sum_{i=0}^{n} \frac{p_i|\gamma_i|}{\gamma_i^2}} \\
\mu &= -\frac{\sum_{i=0}^{n} p_i \frac{\frac{a_i\eta}{(1-2b_i\eta)}}{\frac{1}{(1-2b_i\eta)}}}{1 - \sum_{i=0}^{n} p_i \frac{|\gamma_i|}{(\gamma_i)^2}} \\
\mu &= -\frac{\eta \sum_{i=0}^{n} p_i a_i}{1 - \sum_{i=0}^{n} p_i \frac{\operatorname{sign}\gamma_i}{(\gamma_i)}}
\end{aligned} \tag{3.58}$$

Since we know that the weighted sum of $\sum_{i=0}^{n} p_i a_i = 0$, we can say that

$$\mu = 0 \tag{3.59}$$

Q.E.D.

This is quite surprising as it shows that our expected value, assuming we are choosing in a uniform way, **does not depend on our choice of step size** η. So for large step sizes of η we can estimate our parameter minima by taking the time average of the values we have.

CHAPTER 4

SGD ON DOUBLE WELL POLYNOMIAL

In this chapter we observe the effects of SGD dynamics on problems in which the objective function that we are trying to minimize in non-convex. We offer several different cases to show how depending on the splitting functions SGD can select either a optimal but narrow minima or a sub-optimal but wide minima.

We begin by describing a motivating toy problem for the following section. Residual sum of squares error, often denoted as RSS, is a standard way of quantifying model error. Suppose we have some model that takes in k a data points x with each $x \in \mathbb{R}^n$ with associated true output value $y \in \mathbb{R}^m$, and performs some function $f(x|\beta)$ to predict the output y with a parameter vector $\beta \in \mathbb{R}^p$. The RSS is defined as

$$RSS = \sum_{i=1}^{k} \|f(x_i; \beta) - y_i\|^2 \tag{4.1}$$

Suppose that we are working with a single parameter model $\beta \in \mathbb{R}$, with each input $x \in \mathbb{R}^3$ and output $y \in \mathbb{R}$. We define our model function as follows

$$f(x; \beta) = x_0 + x_1\beta + x_2\beta^2 \tag{4.2}$$

which means that our associated RSS function is given as

$$RSS = \sum_{i=1}^{k} (x_{i,0} + x_{i,1}\beta + x_{i,2}\beta^2 - y_i)^2. \tag{4.3}$$

Notice that when adding together all datapoints, our RSS function effectively becomes

$$RSS = a\beta^4 + b\beta^3 + c\beta^2 + d\beta + e \tag{4.4}$$

where a, b, c, d, e are constants related to the problem and the data. As such, if one wanted to perform complete gradient descent in order to minimize this error term, this would be the final minimized polynomial.

This example may appear a bit contrived, however it is a necessary exercise as many practical problems in the field of Data Science are highly non-convex, so analyzing how SGD approaches problems that do not have one single local minimizer are greatly valuable. A good example of a practical non-convex problem is the Neural Network toy problem provided in Section 1.2.3. In particular, in order to analyze the regularizing effects of SGD, we how to observe how SGD selects between wide, shallow minima compared to narrow, deep minima.

As such we will observe the results of the ODE models and steady states for problems that have two local minima; we refer to the problems as "Double Well" problems. For simplicity, we restrict ourselves to double well polynomials. In general, a double well polynomial $p(x)$ where $x \in \mathbb{R}$ is of the form:

$$\int_0^x \beta_0 t^{k_0}(\beta_1 t + \alpha_1)^{k_1}(\beta_2 t + \alpha_2)^{k_2} q(t) dt. \tag{4.5}$$

Where β_j, α_j, k_j are constants and $q(x)$ is a polynomial with no real roots. This creates a polynomial with 3 critical points at $-\frac{\alpha_j}{\beta_j}$ (provided $\alpha_0 = 0$), such that there are two local minima provided β_0 is chosen such that the leading coefficient is positive, and $k_0 + k_1 + k_2 + \deg(q)$ is odd.

4.1 Double Well with Comparable Depths

In this section we observe a particular case of the non-convex objective function in which the two wells that we are trying to optimize have comparable depths between

the two of them. As such there is not an extreme difference in which one well can be thought of as incredibly better than the other. We take this objective function and perform different splittings on it to observe what occurs in these scenarios.

Our first example of a double well function that we will study is one that does not have any particularly extreme differences between the two wells, rather one is deeper and narrower, but not significantly. We choose the function:

$$p(x) = \int_0^x 20t\left(t+\frac{1}{2}\right)\left(t-\frac{1}{2}\right)\left(t^2-t+\frac{3}{5}\right)dt \tag{4.6}$$

which evaluates to

$$p(x) = 2F(x) = \frac{10}{3}x^6 - 4x^5 + \frac{7}{4}x^4 + \frac{5}{3}x^3 - \frac{3}{2}x^2. \tag{4.7}$$

Unlike the simple quadratic splitting, we now have a case in which there are 7 parameters that we can vary for our various splittings. For a 2 function splitting set we can have an $f_1(x), f_2(x)$ given as:

$$\begin{aligned} f_1(x) &= a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 \\ f_2(x) &= -a_1x + \left(-\frac{3}{2} - a_2\right)x^2 + \left(\frac{5}{3} - a_3\right)x^3 + \left(\frac{7}{4} - a_4\right)x^4 \\ &\quad + \left(-4 - a_5\right)x^5 + \left(\frac{10}{3} - a_6\right)x^6, \end{aligned} \tag{4.8}$$

to which we also consider our choice in η as an additional seventh parameter. Realistically we cannot consider every situation involving these parameter changes, however we will observe several interesting splitting cases.

The first example that we consider is one derived by performing a splitting $f_1(x), f_2(x)$ in which both splitting functions have only one local minima and are monotonic on each side. This means that at every iteration SGD will approach one

specific well that does not depend on the parameter value $\vec{x}$. We then also observe the example that arises when we have one splitting function that is not convex and has two local minima as well; this provides different behavior as now SGD can become stuck in local minima while iterating.

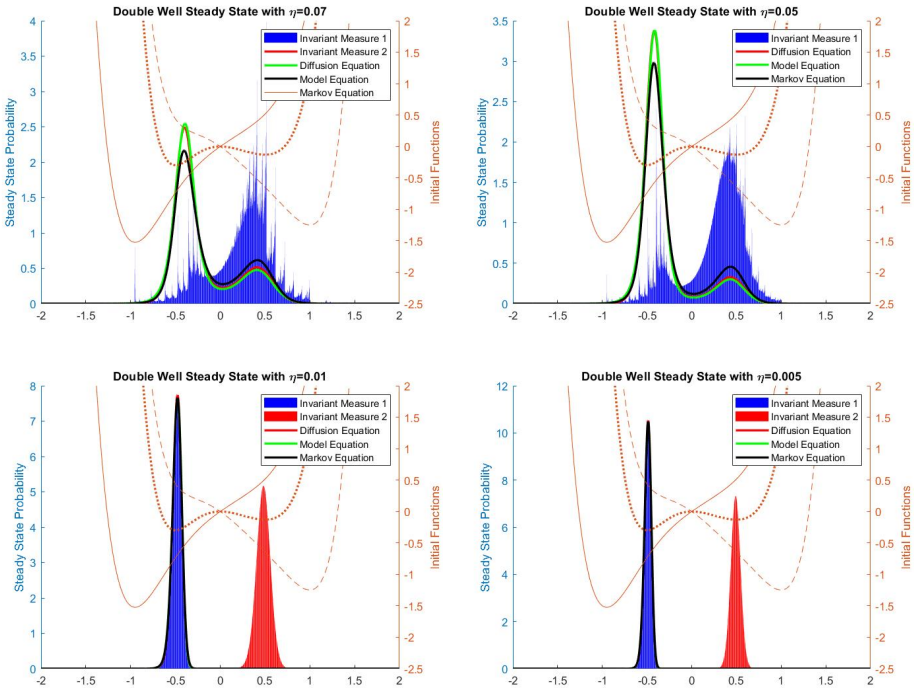


Figure 4.1 Figures in which a double well is split between two functions that have only one minima, with parameters $[a_1, \ldots, a_6]$ being $[1, -.5, 1, 0, 0, 1]$ for various values of η. For our larger values of η we only have one invariant measure which is very non-smooth. This single steady state appears to select the shallower well with greater probability, while the ODE approximations choose the narrower deep well. For smaller values of η there is a split between the invariant measures, one per well, to which the ODE approximations model the deeper well but ignore the state in the shallower well.

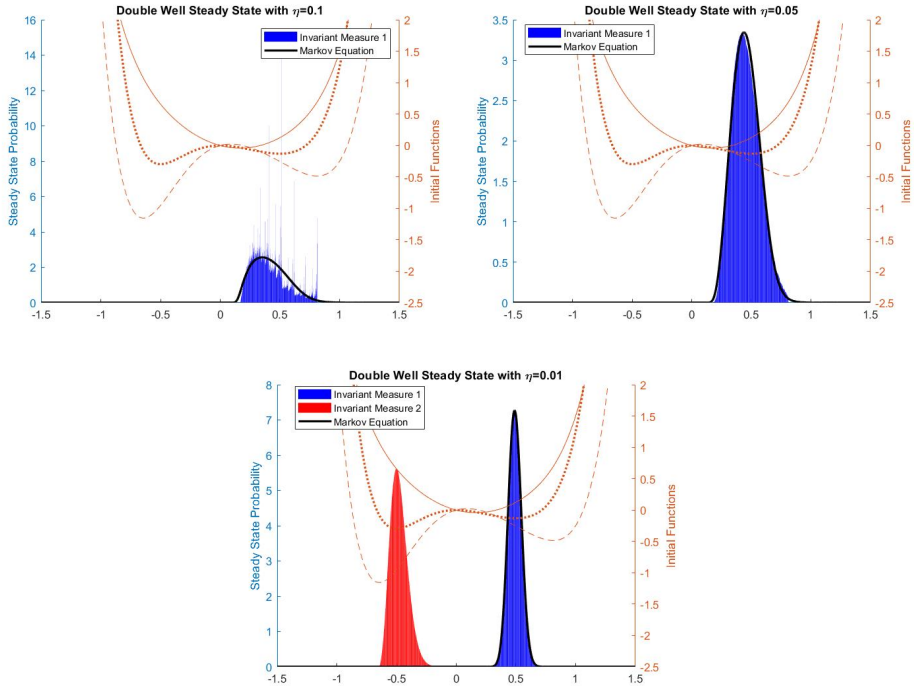


Figure 4.2 Figures in which a double well is split between one function that has only one minima, with parameters $[a_1, \ldots, a_6]$ being $[-.5, 1.5, 0, 0, -.5, 1]$ for various values of η. Interestingly, for larger values of η, both the true steady states and the ODE approximations completely ignore the deep well. This is likely due to the fact that if one reaches the minima of the single minima function, you are located inside the shallow well portion of the double minima splitting function. For small enough values of η we find that there is one invariant measure per well, with the ODE solutions picking the shallower well.

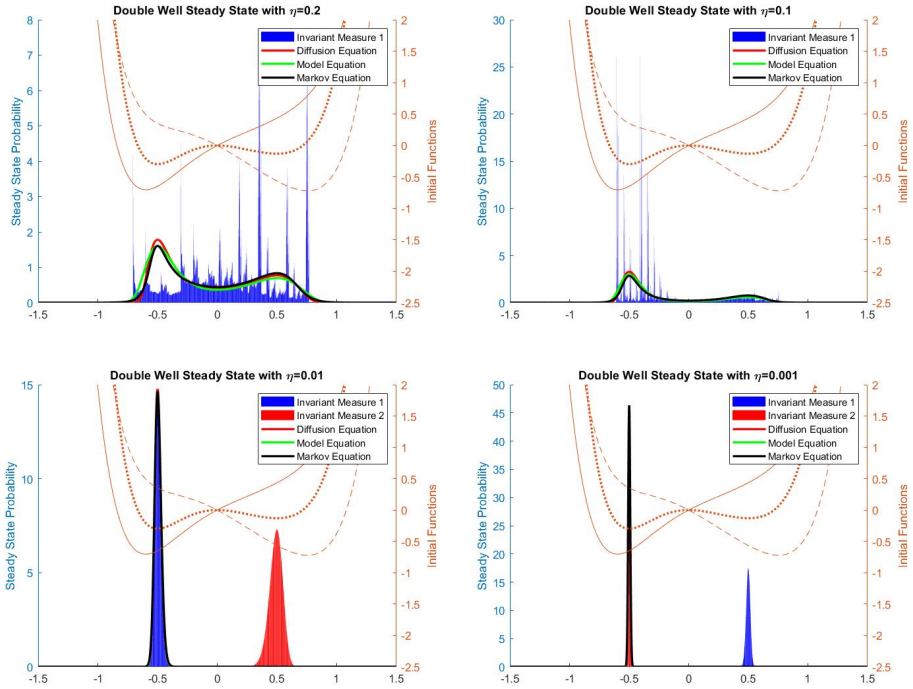


Figure 4.3 Figures in which the splitting functions are $F(x) \pm x$ for various values of η. From Theorem 1 we know that we will have infinite support on our ODE solutions, even though this may not be the case for smaller η values steady states. Despite the solution being very non-smooth, the ODE solutions seem to do an alright job at approximating the true steady state solutions when there is one steady state. When there are two steady states the ODE solutions pick the deeper well.

CHAPTER 5

DISCUSSION AND FURTHER WORK

In this chapter we provide some insight into the work performed in this thesis document, as well as providing several unanswered questions and tasks that could provide interesting work in the future.

5.1 Effectiveness of ODE Approximations to Stationary Distributions

Through this thesis we have compared the solutions of the steady state ODEs to the true invariant measures that we have computed through Ulam's method. As such we have observed that, while in the limit of $\eta \to 0$ these approximations are correct at predicting the behavior of SGD, for larger values of η the approximations we calculated have a few noticeable issues. Here we collect a list of several problems that arise when attempting to model the stationary probability distribution using the Diffusion, Model, or Markov equation.

1. Steady state diffusion approximations converge slowly in the L_1 norm, and even slower in the L_∞ norm. Visually, the diffusion approximations may not appear close to the true invariant measures (even for moderate to small values of η).

2. Steady state diffusion approximations may provide solutions with infinite support, even when the true invariant measure have finite support. Several examples of this phenomenon can be found in the quadratic splitting case (see §3.3.3).

3. It is possible for the steady state diffusion approximations to have a unique solution, even when there are multiple invariant measures to the exact Markov dynamics (see §4). This has the implication that the diffusion approximation

(in cases when the function $F(x)$ may be non-convex) may not hold for infinitely long times (c.f. [12]) .

4. Steady State diffusion approximations will be continuous (and even smooth), when the true invariant measure may not be (see §3.3.4).

As such, we believe that there are more areas that can be researched in the future to help provide a better understanding of SGD and how it can be approximated using continuous time models.

5.2 Future Questions to Answer

The first questions relevant to answer are direct questions related to the dynamics of the invariant measure that are extensions of what was discussed in this thesis document. We pose the following questions or problems for future work:

1. Do similar behavior of dynamics in the quadratic case appear when extending to higher dimensional parameter vectors $\vec{x}$?
2. How do the systems behave under more splittings?
3. Provide more examples of the double well and how the ODE models perform in these problems.
4. What value of η provides the boundary between one and two invariant measures in double well problems?
5. In which values of η does both wells in the double well become stable?
6. What is the behavior in the case when both splitting functions are non-convex?
7. How does a more extreme example of splittings behave in different cases?
8. Does the blue line in Fig. 3.2 represent the end of meaningful approximations via PDE/ODEs?

9. When there are multiple roots of $D(x)$, how does that impact the ODE approximations?

10. Does there exist a Lyapunov energy function for higher dimensional equations?

11. Does the $\mathcal{O}(\eta^2)$ ODE approximations perform noticeably better for larger values of η?

We hope that in the future these problems can provide interesting problems and results for ourselves and any other future researchers.

BIBLIOGRAPHY

[1] Alnur Ali, Edgar Dobriban, and Ryan Tibshirani. The implicit regularization of stochastic gradient flow for least squares. In *International Conference on Machine Learning*, pages 233–244. PMLR, 2020.

[2] Jing An, Jianfeng Lu, and Lexing Ying. Stochastic modified equations for the asynchronous stochastic gradient descent. *Information and Inference: A Journal of the IMA*, 9(4):851–873, 2020.

[3] Anders Andreassen and Ethan Dyer. Asymptotics of wide convolutional neural networks. *arXiv preprint arXiv:2008.08675*, 2020.

[4] Aritz Bercher, Lukas Gonon, Arnulf Jentzen, and Diyora Salimova. Weak error analysis for stochastic gradient descent optimization algorithms. *arXiv preprint arXiv:2007.02723*, 2020.

[5] Jeremy Bernstein, Arash Vahdat, Yisong Yue, and Ming-Yu Liu. On the distance between two neural networks and the stability of learning. *arXiv preprint arXiv:2002.03432*, 2020.

[6] Christopher Bose, Gary Froyland, Cecilia González-Tokman, and Rua Murray. Ulam's method for lasota–yorke maps with holes. *SIAM Journal on Applied Dynamical Systems*, 13(2):1010–1032, 2014.

[7] Pratik Chaudhari, Adam Oberman, Stanley Osher, Stefano Soatto, and Guillaume Carlier. Deep relaxation: partial differential equations for optimizing deep neural networks. *Research in the Mathematical Sciences*, 5(3):1–30, 2018.

[8] Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE, 2018.

[9] Jiu Ding, Tien Yien Li, and Aihui Zhou. Finite approximations of markov operators. *Journal of Computational and Applied Mathematics*, 147(1):137–152, 2002.

[10] Zhiyan Ding and Qin Li. Langevin monte carlo: random coordinate descent and variance reduction. *arXiv preprint arXiv:2007.14209*, 2020.

[11] Carlos Esteve, Borjan Geshkovski, Dario Pighin, and Enrique Zuazua. Large-time asymptotics in deep learning. *arXiv preprint arXiv:2008.02491*, 2020.

[12] Y. Feng, T. Gao, L. Li, J.-G. Liu, and Y. Lu. Uniform-in-time weak error analysis for stochastic gradient descent algorithms via diffusion approximation. *Comm. Math. Sci.*, 18:163–188, 2020.

[13] Y. Feng, L. Li, and J.-G. Liu. Semigroups of stochastic gradient descent and online principle component analysis: properties and diffusion approximations. *Comm. Math. Sci.*, 16:777–789, 2018.

[14] Gary Froyland. Ulam's method for random interval maps. *Nonlinearity*, 12(4):1029, 1999.

[15] Caroline Geiersbach and Winnifried Wollner. A stochastic gradient method with mesh refinement for pde-constrained optimization under uncertainty. *SIAM Journal on Scientific Computing*, 42(5):A2750–A2772, 2020.

[16] Susanne Gerber, Simon Olsson, Frank Noé, and Illia Horenko. A scalable approach to the computation of invariant measures for high-dimensional markovian systems. *Scientific reports*, 8(1):1–9, 2018.

[17] A Golmakani, CE Koudjinan, S Luzzatto, and Paweł Pilarczyk. Rigorous numerics for critical orbits in the quadratic family. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(7):073143, 2020.

[18] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. PMLR, 2016.

[19] Catherine F Higham and Desmond J Higham. Deep learning: An introduction for applied mathematicians. *SIAM Review*, 61(4):860–891, 2019.

[20] W. Hu, C.J. Li, L. Li, and J.-G. Liu. On the diffusion approximation of nonconvex stochastic gradient descent. *Annals of Mathematical Sciences and Applications*, 4:3–32, 2019.

[21] Michael V Jakobson. Absolutely continuous invariant measures for one-parameter families of one-dimensional maps. *Communications in Mathematical Physics*, 81(1):39–88, 1981.

[22] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*, volume 112. Springer, 2013.

[23] Nikolas Kantas, Panos Parpas, and Grigorios A Pavliotis. The sharp, the flat and the shallow: Can weakly interacting agents learn to escape bad minima? *arXiv preprint arXiv:1905.04121*, 2019.

[24] Tamara G Kolda and David Hong. Stochastic gradients for large-scale tensor decomposition. *SIAM Journal on Mathematics of Data Science*, 2(4):1066–1095, 2020.

[25] Lingkai Kong and Molei Tao. Stochasticity of deterministic gradient descent: Large learning rate for multiscale objective function. *Advances in Neural Information Processing Systems*, 33, 2020.

[26] Juan Kuntz, Philipp Thomas, Guy-Bart Stan, and Mauricio Barahona. Stationary distributions of continuous-time markov chains: a review of theory and truncation-based approximations. *SIAM Review*, 63(1):3–64, 2021.

[27] Andrzej Lasota and Michael C Mackey. *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*, volume 97. Springer Science & Business Media, 2013.

[28] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

[29] Qianxiao Li, Tai Cheng, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pages 2101–2110. PMLR, 2017.

[30] Ruilin Li, Xin Wang, Hongyuan Zha, and Molei Tao. Improving sampling accuracy of stochastic gradient mcmc methods via non-uniform subsampling of gradients. *arXiv preprint arXiv:2002.08949*, 2020.

[31] Tien-Yien Li. Finite approximation for the frobenius-perron operator. a solution to ulam's conjecture. *Journal of Approximation theory*, 17(2):177–186, 1976.

[32] Chao Ma, Stephan Wojtowytsch, Lei Wu, et al. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. *arXiv preprint arXiv:2009.10713*, 2020.

[33] Stephan Mandt, Matthew Hoffman, and David Blei. A variational analysis of stochastic gradient algorithms. In *International conference on machine learning*, pages 354–363. PMLR, 2016.

[34] Stephan Mandt, Matthew D Hoffman, David M Blei, et al. Continuous-time limit of stochastic gradient descent revisited. *NIPS-2015*, 2015.

[35] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[36] Marvin Minsky and Seymour A Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT press, 1969.

[37] Deanna Needell and Rachel Ward. Batched stochastic gradient descent with weighted sampling. In *International Conference Approximation Theory*, pages 279–306. Springer, 2016.

[38] Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Advances in neural information processing systems*, 27:1017–1025, 2014.

[39] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[40] Sheldon M Ross. *Applied probability models with optimization applications*. Courier Corporation, 2013.

[41] Umut Şimşekli, Mert Gürbüzbalaban, Thanh Huy Nguyen, Gaël Richard, and Levent Sagun. On the heavy-tailed theory of stochastic gradient descent for deep neural networks. *arXiv preprint arXiv:1912.00018*, 2019.

[42] Justin Sirignano and Konstantinos Spiliopoulos. Stochastic gradient descent in continuous time: A central limit theorem. *Stochastic Systems*, 10(2):124–151, 2020.

[43] Gilbert Strang. *Linear Algebra and Learning from Data*. Wellesley - Cambridge Press, Wellesley MA, 2019.

[44] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. SIAM, Philadelphia PA, 2000.

[45] Yazhen Wang and Shang Wu. Asymptotic analysis via stochastic differential equations of gradient descent algorithms in statistical and computational paradigms. *Journal of Machine Learning Research*, 21(199):1–103, 2020.

[46] E Weinan, Chao Ma, and Lei Wu. Machine learning from a continuous viewpoint, i. *Science China Mathematics*, 63(11):2233–2266, 2020.

[47] Lei Wu, Chao Ma, and Weinan E. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8289–8298, 2018.

[48] Xiaoxia Wu, Edgar Dobriban, Tongzheng Ren, Shanshan Wu, Zhiyuan Li, Suriya Gunasekar, Rachel Ward, and Qiang Liu. Implicit regularization of normalization methods. *arXiv preprint arXiv:1911.07956*, 2019.

[49] Xiaoxia Wu, Simon S Du, and Rachel Ward. Global convergence of adaptive gradient methods for an over-parameterized neural network. *arXiv preprint arXiv:1902.07111*, 2019.

[50] Yuege Xie, Xiaoxia Wu, and Rachel Ward. Linear convergence of adaptive stochastic gradient descent. In *International Conference on Artificial Intelligence and Statistics*, pages 1475–1485. PMLR, 2020.

[51] Yao Zhang, Andrew M Saxe, Madhu S Advani, and Alpha A Lee. Energy–entropy competition and the effectiveness of stochastic gradient descent in machine learning. *Molecular Physics*, 116(21-22):3214–3223, 2018.