Theoretical Connection between the Stochastic Method and the Barrier Method for Linear Programs

A report submitted for the course
COMP8755 Individual Computing Project
- 12pt (S1 2022)

By:
Haoran Huang

Supervisor:
Professor. Stephen Gould

Australian National University

School of Computing
College of Engineering and Computer Science (CECS)
The Australian National University

May, 2022

Abstract

Linear programs can be used to describe the forward pass of a declarative node in a deep declarative network; however, the gradient of the linear program vanishes in the backward pass. Therefore, we would like to investigate the connection between two approximations of the linear program which both enable gradient calculation – the stochastic method and the barrier method. We provide proof for a well-known theorem, the differentiability of the Fenchel conjugate function and build the connection between the stochastic and the barrier methods for linear programs based on the theorem. Then we performed experiments that verify such a connection. We found that there exists an equivalency between the stochastic method and the barrier function method in the approximation and gradient calculation of a linear program.

Acknowledgement

I would like to express my deepest gratitude and appreciation to Prof. Stephen Gould. His guidance and advice helped me finish my project. I would also like to give special thanks to my parents who supported me unconditionally throughout my postgraduate study.

Declaration:

I declare that this work:

- upholds the principles of academic integrity, as defined in the University Academic Misconduct Rules;
- is original, except where collaboration (for example group work) has been authorised in writing by the course convener in the class summary and/or Wattle site;
- is produced for the purposes of this assessment task and has not been submitted for assessment in any other context, except where authorised in writing by the course convener;
- gives appropriate acknowledgement of the ideas, scholarship and intellectual property of others insofar as these have been used;
- in no part involves copying, cheating, collusion, fabrication, plagiarism or recycling.

May, Haoran Huang

Contents

List of Figures

Chapter 1

Introduction

Starting from the multilayer perceptron model [10], deep learning models have been evolving as computation power improves and new algorithms are brought up. In particular, the backpropagation algorithm is essential in that it enables end-to-end learning. As new theories and ideas are adopted, models like convolutional neural networks[11], recurrent neural networks[18] and transformers[19] have been developed and improved benchmarks for tasks in computer vision, natural language processing and other fields. Traditionally, the node in a neural network is defined by explicit functions as shown in fig. 1.1. Gould et al. embedded the idea of differentiable optimization into the deep learning model and proposed the deep declarative network (DDN)[6]. In the DDN, the forward pass of a node can be defined by optimization problems. Such a node is called the declarative node, while the traditional node can be referred to as the imperative node. Declarative nodes are more expressive than imperative nodes as functions can be transformed into optimization problems, the converse is not necessarily true. Gould et al. also introduce differentiating algorithm [5, 6] for the backpropagation of the declarative node making DDN an end-to-end learnable model.

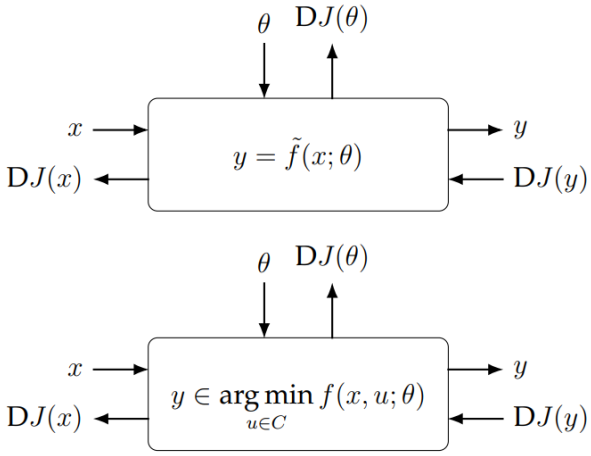

Figure 1.1: ([6]) Illustration of imperative node and declarative node

However, when the optimization problem is not strongly convex, the gradient vanishes. Consequently, the backward gradient flow gets stuck and the DDN fails to learn the parameters. One common example is the linear program (LP) problem. The outputs of an LP problem always lie in the set of vertices of the polytope defined by the inequality constraints. And the output change abruptly as the parameter in the objective changes, so the gradient of the LP problem is either zero or not defined.

There are two well-developed methods to approximate the LP problem and hence deal with the zero gradients, the stochastic method and the barrier method. To better understand and apply these methods, their theoretical connection is studied. We first introduce and explain the differentiability of the Fenchel conjugate function[17, 3, 1] theorem and build the connection between the stochastic and the barrier methods for linear programs based on the theorem. Then we design experiments that verify such a connection in the forward and backward pass of an LP declarative node. We find that for any stochastic method, there exists a barrier function method that approximates the linear program in the same way. For certain stochastic methods, we can derive closed-form barrier methods that are equivalent (i.e the stochastic method with Gumbel noise is equivalent to the negative entropy barrier method). In general, it is hard to find a close-formed barrier function from a stochastic method due to expectation calculation in the stochastic method.

Chapter 2

Related Works

The idea of DDN [6] was built on differentiable optimization which generates the backward gradient flow and enables the end-to-end learning architect of DDN. Therefore, the algorithm for gradient calculation and approximation is essential to the DDN model. Gould et al.[5] provide a formula to differentiate parametrized argmin and argmax problems. This method works for problems consisting of strongly convex functions which have an invertible Hessian matrix. It does not work for the LP problem but can compute the gradient of LPs approximated by barrier methods. Abernethy et al [2] introduce stochastic smoothing techniques for smoothing the LP and conclude properties of perturbed optimal and optimizer. Based on those properties, Berthet et al.[3] implement a practical algorithm to differentiate the LP approximated by stochastic methods using *Monte Carlo estimation*. These algorithms for differentiating approximated LPs are used and implemented in the experiments of our research.

The perturbation is essential in the stochastic method and was studied by many researchers. The interest in perturbation arises from the conflict between making good assumptions and computing efficiently as perturbation is efficient to compute and serves as a regularizer (Hanza et al. [8]).

Firstly, perturbation severs as an alternative representation of a probability model (i.e. the perturb-and-MAP model). Secondly, we can learn the properties of the machine learning algorithm through the connection between perturbation and regularization.

Papandreou and Yuille [15], the perturbation models probability distribution. The Gumbel perturbation resembles the Gibbs distribution. This result helps us go from the expectation of perturbed maximum over a noise distribution to a closed-form function so that we can find the conjugate function of it, which is its counterpart barrier function.

Prior Knowledge is required in almost all machine learning tasks. The best solution often varies from the best decision made upon this believed knowledge (Hanzan et al.[8]). Prior knowledge can be given as a regularization or a prior

distribution in statistical machine learning. Injecting perturbations into our models can be efficient and beneficial as it brings in the prior knowledge and uncertainties on the knowledge.

The connection between the perturbation and the regularizer inspired our study, and we extend such a connection to the stochastic method and the barrier method for LPs.

	Prior Knowledge	Regularization	Perturbation
OLO	How the Learner receives a reward?	Follow the Regularized Leader(FTRL)	Follow the Perturbed Leader(FTPL)
SFP	Expected payoff of a choice of play	Deterministic Perturbation	Perturbation

Figure 2.1: A table shows prior knowledge, regularization and perturbation in different scenarios

Abernethy et al.[1] study the online learning algorithm where the agent makes choices over a fixed action set and the omniscient adversary determines the score of that choice. The agent can only make choices based on previous knowledge of the adversary, but the adversary might change its strategy. To cope with this, follow the perturbed leader (FTPL) and follow the regularized leader are invited and found equivalent.

Hofbauer et al.[9] investigated the global convergence of games under the stochastic fictitious play (SFP) model. In SFP, a perturbation is added to the expected payoff compared to standard fictitious play where the player acts according to their belief on opponents (e.g., min-max algorithm). In the study of discrete choice of an agent. Given any form of the perturbation vector (regardless of the distribution), Hofbauer et al. [9] claimed that a deterministic regularizer for the choice probability can be derived. This random choice theorem lies the foundation of the theoretical connection between perturbation and regularization. In our study, this connection is restudied, and a detailed proof is provided with applications in the LP problems.

Rockefeller[17] offers elaborate theories on convex optimization, especially on Fenchel duality which helps us prove the theory. Boyd [4] and Mordukhovich and Nam [12] also provide materials on convex analysis.

Chapter 3

Problem Definition

3.1 Linear Program

Let $f : \mathbb{R}^n \to \mathbb{R}$ and Ω be a nonempty subset of $\mathbb{R}^n$, consider the optimization problem P:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & x \in \Omega \end{aligned} \tag{3.1}$$

where $f(x)$ is the objective function and Ω is the feasible region.

We provide the definition of the global optimal solution.

Definition 3.1.1. *([12]) Any $\bar{x} \in \Omega$ is called a feasible solution of problem P. $\bar{x} \in \Omega,\ f(\bar{x}) \leq f(x)\ \forall x \in \Omega$ is called a global optimal solution of P.*

The absolute minimizer is also defined in [12], which can used to check if an *argmin* or *argmax* operation generates a solution.

Definition 3.1.2. *([12]) Let f be a proper (definition 4.1.1) function. An element $x \in dom(f)$ is called an absolute minimizer of f if $f(\bar{x}) \leq f(x),\ \forall x \in dom(f)$.*

The LP problem is a subclass of optimization problems where the objective and constraint functions are all linear [4].

The inequality form of Linear program is written as [4]:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^T x \\ \text{subject to} \quad & Ax \preceq b \end{aligned} \tag{3.2}$$

where $c \in \mathbb{R}^n$ is the linear coefficient of the objective function, $x \in \mathbb{R}^n$ is the variable, $A \in \mathbb{R}^{m \times n}$ represents the constraint functions in a matrix form, $b \in \mathbb{R}^m$ is the constants in the constraints function in a vector form. The feasible region

is a polytope defined by halspaces: $a_i^T x \leq b_i,\ \forall 1 \leq i \leq m$, where $a_i \in \mathbb{R}^n$ is the i-th row of A and $b_i \in \mathbb{R}$ is the i-th element in b.

The linear program has a wide range of applications in optimization. It can be used to describe and solve practical problems like the network flow problem and the profit maximization problem [16]. In our study, we are interested in LP used as a declarative node. We can see from fig. 3.1, the feasible points with the same objective value lie in a hyperplane orthogonal to c and at least one of the vertices minimizes the objective. Therefore, the LP is a piece-wise constant function of the objective parameter c and hence the gradient is either zero or non exist[3]. To enable learning on the LP declarative node, we need approximation methods for the LP problem which can provide backward gradient flow.

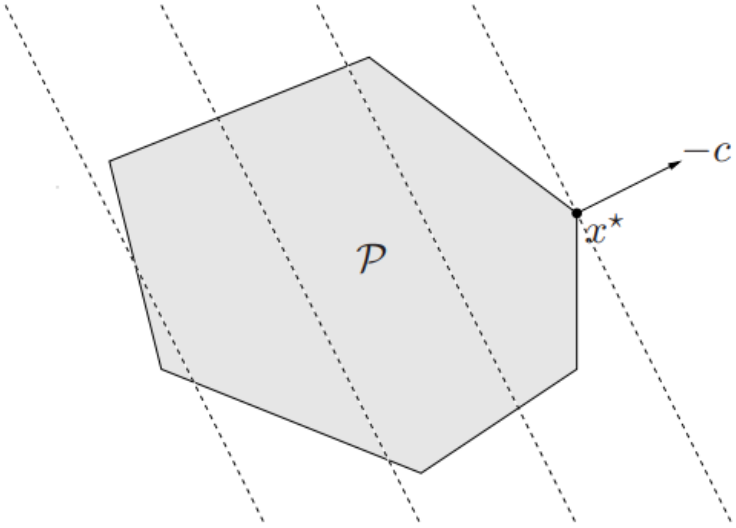

Figure 3.1: ([4]) Geometric view on LP problem where the feasible region $\mathcal{P}$ defined by constraints is shaded, the level curves are shown in dash line and the optimal solution x^* is reached at a vertex.

3.2 Barrier Method

The barrier method is a well-known interior point method algorithm which solves convex optimization problems with inequality constraints [4]. The barrier function approximates the inequality constraints by increasing the cost of the barrier function as the argument approaches the boundary of the feasible set. The barrier methods implicitly impose the constraints of the original problem by subtracting the barrier function from the original objective. In the interior point algorithm, a series of barrier methods with increasing parameter t is performed

to approach the optimal gradually (illustrated in fig. 3.2). In this research, a barrier method with a fixed parameter is applied to approximate an LP problem.

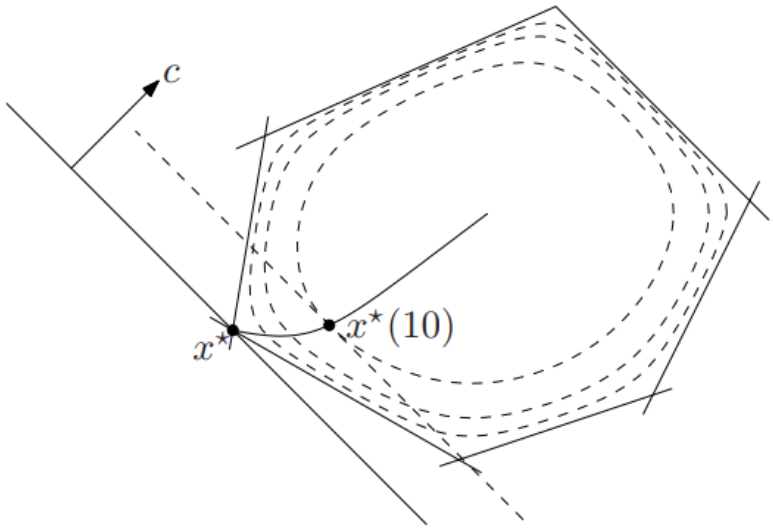

Figure 3.2: ([4]) Central path founded by barrier method approximating the optimal solution

3.2.1 Logarithm(log) Barrier Method

The most common barrier function is the logarithm function. The new objective defined by the barrier method is:

$$\underset{x}{\text{minimize}}\ tc^T x - \sum_{i=1}^{m} \log(b_i - a_i x) \tag{3.3}$$

We can also define the log barrier approximation. Let $f(x,c) = tc^T x - \sum_{i=1}^{m} \log(b_i - a_i x)$. The log barrier approximation of the LP problem in form eq. (3.2) is:

$$x^* = l(c) = \underset{x}{\text{argmin}}\{tc^T x - \sum_{i=1}^{m} \log(b_i - a_i x)\} = \underset{x}{\text{argmin}}\, f(x,c) \tag{3.4}$$

3.2.2 Negative Entropy Barrier Method

The negative entropy function is used as a measurement of the difference between a distribution and the Gaussian distribution in entropy. The negative Shannon entropy function is defined as: $S = \sum_i x_i ln x_i$. The domain of the

negative function is usually defined as $0 \preceq x \preceq 1$, so we only apply this "barrier" when the feasible region of our LP problem is a simplex. It is introduced as the conjugate regularizer of the perturbed maximizer with Gumbel noise in [3] where it was used as:

$$\underset{x}{\text{maximize}} \ -tc^T x - \sum_{i=1}^{m} x_i ln x_i \tag{3.5}$$

Transform it into our LP problem in form eq. (3.2), we have our new objective:

$$\underset{x}{\text{minimize}} \ tc^T x + \sum_{i=1}^{m} x_i ln x_i \tag{3.6}$$

We can define the negative entropy approximation accordingly. Let $f(x, c) = tc^T x + \sum_{i=1}^{m} x_i ln x_i$. The negative entropy barrier approximation of the LP problem in form eq. (3.2) is:

$$x^* = e(c) = \underset{x}{\text{argmin}}\{tc^T x + \sum_{i=1}^{m} x_i ln x_i\} = \underset{x}{\text{argmin}}\, f(x, c) \tag{3.7}$$

3.3 Differentiating Barrier Method Approximation

The method of differentiating parameterized argmin problem is provided in [5]. The gradient is obtained by implicitly differentiating the optimal condition(i.e for $g(x) = \text{argmin}_y f(x, y)$ the optimal condition is $\frac{df(x,y)}{dy}|_{y=g(x)} = 0$). This method can be used in backward gradient calculation for declarative nodes.

Lemma 3.3.1. *([5]) Let $f : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ be a function and $g(x) = \text{argmin}_y f(x, y)$. Suppose f is continuous and twice differentiable. Then*

$$\frac{dg(x)}{dx} = -\frac{f_{XY}(x, g(x))}{f_{YY}(x, g(x))}$$

where $f_{XY} = \frac{\partial^2 f}{\partial x \partial y}$ and $f_{YY} = \frac{\partial^2 f}{\partial y^2}$.

We can find the derivative of log barrier approximation eq. (3.4) using

$$\frac{dl(c)}{dc} = \frac{f_{CX}(l(c), c)}{f_{XX}(l(c), c)}$$

where $f_{CX} = \frac{\partial^2 f}{\partial x \partial c}$ and $f_{XX} = \frac{\partial^2 f}{\partial x^2}$.

$$\frac{\partial f}{\partial x} = tc^T + \sum_{i=1}^{m} \frac{a_i}{(b_i - a_i x)}$$
$$f_{CX} = \frac{\partial^2 f}{\partial x \partial c} = tI_n$$
$$f_{XX} = \frac{\partial^2 f}{\partial x^2} = \sum_{i=1}^{m} \frac{a_i^T a_i}{(b_i - a_i x)^2}$$

Similarly, we can find the derivative of negative entropy barrier approximation eq. (3.7) using

$$\frac{de(c)}{dc} = \frac{f_{CX}(e(c), c)}{f_{XX}(e(c), c)}$$

where $f_{CX} = \frac{\partial^2 f}{\partial x \partial c}$ and $f_{XX} = \frac{\partial^2 f}{\partial x^2}$.

$$\frac{\partial f}{\partial x} = tC^T + \sum_{i=1}^{m} (1 + lnx_i)$$
$$f_{CX} = \frac{\partial^2 f}{\partial x \partial c} = tI_n$$
$$f_{XX} = \frac{\partial^2 f}{\partial x^2} = \begin{bmatrix} \frac{1}{x_1} & & \\ & \ddots & \\ & & \frac{1}{x_n} \end{bmatrix}$$

3.4 Stochastic Method

Given an LP problem in form eq. (3.2) with feasible region $\Omega = \{x | Ax \preceq b\}$, we denote the optimal objective value and optimal solution F and y^*:

$$F(c) = \min_{x \in \Omega} \langle x, c \rangle, \;\; y^*(c) = \operatorname*{argmin}_{x \in \Omega} \langle x, c \rangle \tag{3.8}$$

and $\nabla_c F(c) = y^*(c)$ because when the argmin is unique, the optimal value is obtained at the optimal solution.

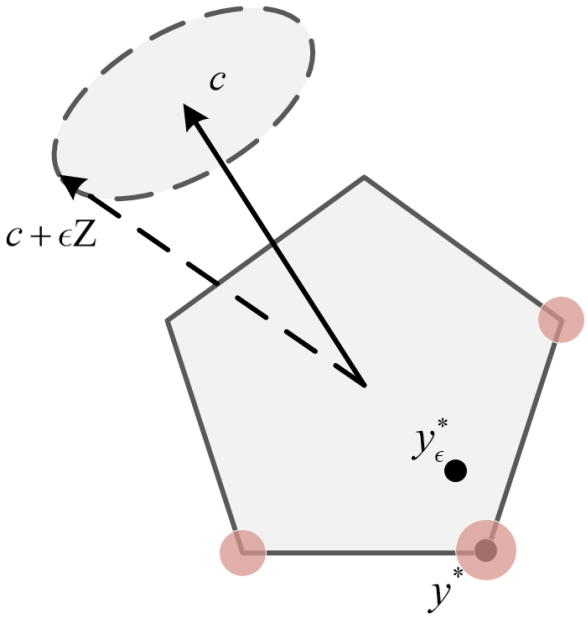

Figure 3.3: Perturbed optimizer y^*_ϵ obtained from stochastic smoothing

In order to deal with vanishing gradient in LP declarative node, a noise ϵZ is added to objective parameter c where $Z \in \mathbb{R}^d$ be a noise vector which has a density distribution of form:

$$d\mu(z) \propto exp(-v(z)dz) \tag{3.9}$$

and $\epsilon > 0 \in \mathbb{R}$ is the temperature parameter which controls the magnitude of the perturbation [3]. Such uncertainties added to the model perturb the output of the LP problem. The output of this stochastic model then has a almost surely uniquely defined distribution $p_c(y) = P(y^*(c + \epsilon Z))$.

We generate the smoothed optimum F and optimizer y^* by calculating the expectation over the noise distribution and denote them as perturbed optimum and perturbed optimizer which is used as an approximation solution of the LP problem [3].

Definition 3.4.1. *perturbed optimum and perturbed optimizer*

$$F_\epsilon(c) = E[F(c + \epsilon Z)] = E[\min_{x \in \Omega} \langle x, c + \epsilon Z \rangle]$$

$$y^*_\epsilon(c) = E_{p_c(x)}[X] = E[\operatorname{argmin}_{x \in \Omega} \langle x, c + \epsilon Z \rangle] = E[\nabla_c \min_{x \in \Omega} \langle x, c + \epsilon Z \rangle] = \nabla_c F_\epsilon(c)$$

3.5 Typical Noise Distribution

3.5.1 Gumbel Distribution

Gumbel distribution (Generalized Extreme Value distribution Type-I) was invented by Gumbel [7] when analysing the distribution of the maximum and minimum of samples of arbitrary distributions. The standard Gumbel distribution ($Z \sim Gumbel(0, 1)$) follows the form eq. (3.9), where

$$\mu(z) = e^{-(z+e^{-z})}$$

$$v(z) = z + e^{-z}$$

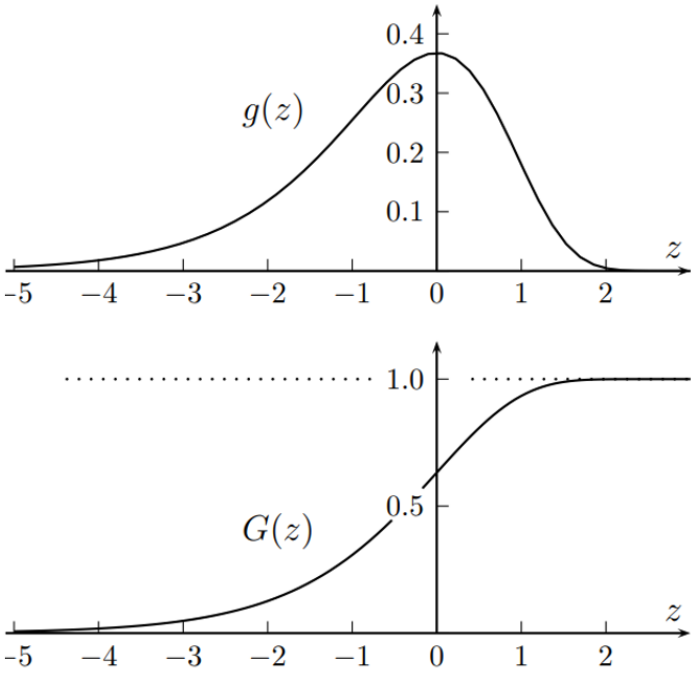

Figure 3.4: Probability density function and cumulative distribution function of standard Gumbel distribution

3.5.2 Gaussian Distribution

Gaussian distribution is often used as it describes many nature events. The standard Gaussian distribution ($Z \sim \mathcal{N}(0,1)$) also follows the form eq. (3.9), where

$$\mu(z) = \frac{1}{\sqrt{2}} e^{\frac{-1}{2}z^2}$$

$$v(z) = \frac{1}{2}z^2$$

Learning the $\mu(z)$ and $v(z)$ is essential when computing the expectations of perturbed optimizer and Jacobian which is discussed in proposition 3.6.1.

3.6 Differentiating Stochastic Method Approximation

Following properties of the perturbed optimal and optimizer are derived using stochastic smoothing in [2]. These properties can used in the forward and backward pass calculation of LP node.

Proposition 3.6.1. *Let Z be a random variable with distribution $d\mu(z) \propto exp(-v(z)dz)$ and a twice differentiable v. we can derive that*

- $y^*_\epsilon(c) = \nabla_c F_\epsilon(c) = E[y^*(c+\epsilon Z)] = E[F(c+\epsilon Z)\nabla_z v(Z)/\epsilon]$
- $J_c y^*_\epsilon(c) = E[y^*(c+\epsilon Z)\nabla_z v(Z)^T/\epsilon]$,
  where $J_c y^*_\epsilon(c)$ is the Jacobian Matrix of y^*_ϵ at c

The practical way of computing the expectation can be the *Monte-Carlo estimate* . The *Monte-Carlo estimate* of the perturbed optimal solution is given in [3]. We define the *Monte-Carlo estimate* of the Jacobian according to proposition 3.6.1. These can be used as

Definition 3.6.2. *Given $c \in \mathbb{R}$, let $(Z^{(1)}, ..., Z^{(N)})$ be N identical independent distributed(i.i.d.) copies of from random variable Z.*

$$y^{(n)} = y^*(c+\epsilon Z)) = \underset{x\in\Omega}{\operatorname{argmin}} \langle x, c\rangle \ \forall n = 1, ..., N$$

- *The Monte-Carlo estimate $\bar{y}_\epsilon(c)$ of the perturbed optimal solution $y^*_\epsilon(c)$ is defined as:*

$$\bar{y}_\epsilon(c) = \frac{1}{N}\sum_{i=1}^{N} y^{(i)}$$

- *The Monte-Carlo estimate $\bar{J}_c y_\epsilon(c)$ of the Jacobian Matrix $J_c y^*_\epsilon(c)$ is defined as:*

$$\bar{J}_c y_\epsilon(c) = \frac{1}{N}\sum_{i=1}^{N} y^{(i)}\nabla_z v(Z)^T/\epsilon$$

Chapter 4

Theory

In this chapter, a detailed poof of a known theorem, the differentiability of the Fenchel conjugate function [17, 3, 1], will be given. Based on the theorem, we build the connection between stochastic and barrier methods for LPs.

4.1 Preliminaries

In this section, mathematical backgrounds used in the proof are introduced.

Function with empty domain is trivial for analysis, so we would like to focus on nonempty domain.

Definition 4.1.1. *Proper. A function f is called proper if the domain of f is nonempty. Denoted as $dom(f) \neq \emptyset$.*

Convexity measures the curvature of the function which has some properties when analyzing bounds and global optimal. Therefore, convex analysis is essential in optimization and for optimization problems there is always some assumption on the convexity of the problem. For our LP problem, objective and constraints functions are affine and hence convex.

Definition 4.1.2. *([4]) Convexity. A function f is convex if:*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y),\ \forall x, y \in dom(f), x \neq y, \lambda \in [0,1]$$

A function f is strictly convex if:

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda)f(y),\ \forall x, y \in dom(f), x \neq y, \lambda \in (0,1)$$

Definition 4.1.3. *([12]) Strong Convexity. A function f is strongly convex with parameter δ if $g(x)$ defined by*

$$g(x) = f(x) - \frac{\delta}{2}\|x\|^2, x \in \mathbb{R}^n$$

is convex.

Strongly convex is a stronger assumption which infers strictly convex.

Theorem 4.1.4. *Strong convexity infers strict convexity, and strictly convexity infers convexity.*

The Fenchel conjugate function is an important operation in convex analysis. It plays an important role in duality. The conjugate function is convex regardless of the convexity of the original function.

Definition 4.1.5. *([4]) Fenchel Conjugate. The conjugate function of $f(x)$ is denoted as $f^*(v)$*

$$f^*(v) = \sup_{x \in dom(f)} = \{\langle v, x\rangle - f(x)\}$$

Subgradient is a generalized derivative for convex function where it not necessarily differentiable. It is helpful in the proof when discussing the differentiability.

Definition 4.1.6. *([12]) Let f be an convex function, An element v is the subgradient of f at $\bar{x}$ if*

$$\langle v, x - \bar{x}\rangle \leq f(x) - f(\bar{x}),\ \forall x \in dom(f)$$

The subdifferential of f at $\bar{x}$ is the set of all subgradient of x at $\bar{x}$ denoted as $\partial f(\bar{x})$.

Fréchet differentiable generalizes differentiability on normed space and is often used for function with multiple real variables or have a vector value.

Definition 4.1.7. *([12]) Fréchet differentiable. f is called Fréchet differentiable at $\bar{x} \in int\, dom(f)$ if there exists an element v in the domain of f, such that*

$$\lim_{x \to \bar{x}} \frac{f(x) - f(\bar{x}) - \langle v, x - \bar{x}\rangle}{\|x - \bar{x}\|} = 0$$

If v is unique, then we can denote $\nabla f(\bar{x}) = v$.

The following theorem describes the relationship between Fréchet differentiability and subgradient. which can be utilized to discuss the differentiability by looking at the subgradient sets.

Theorem 4.1.8. *([12]) Let f be a convex function and $\bar{x} \in int\ dom(f)$. The following propositions are equivalent.*

- *f is Fréchet differentiable at $\bar{x}$.*
- *The subdifferential at $\bar{x}$ $\partial f(\bar{x})$ is a singleton.*

Lower semicontinuity is a weaker condition of continuity which only takes the lower half of the continuous continuity. If we take a continuous function and decrease it at certain point then it becomes lower semicontinuous.

Definition 4.1.9. *([12]) Lower semicontinuity. f is called lower semicontinuous at $\bar{x}$ if $\forall \lambda < f(\bar{x})$ $\exists \delta > 0$ such that,*

$$\lambda < f(x) \ \forall \|x - \bar{x}\| < \delta$$

If f is lower semicontinuous at all points in the domain then f is lower semicontinuous.

Lower semicontinuity has some nice properties for analyzing the existence of the global optimal. Therefore, we only assume lower semicontinuity in theorem 4.2.1.

Theorem 4.1.10. *([12]) Let f be a proper, convex and lower semicontinuous function. There exists $v \in \mathbb{R}^n, b \in \mathbb{R}$ such that,*

$$\langle v, x \rangle + b \leq f(x), \forall x \in \mathbb{R}^n$$

Like continuity, lower semicontinuity has other forms describing the condition, which are all part of that form of continuity.

Theorem 4.1.11. *([12]) f(x) is lower semicontinuous at $\bar{x}$ is equivalent to following propositions:*

- *For all λ, the sublevel set L_λ is closed.*
- *The epigraph of f epi(f) is closed.*
- *For all sequences $\{x_k\}$ that converges to $\bar{x}$, $\lim_{k \to \inf} f(x_k) \geq f(\bar{x})$*

Lower semicontinuity is often used in convex analysis.

Definition 4.1.12. *([12]) Closed. A proper convex function is closed iff it is lower semicontinuous.*

One of the properties of the conjugate function is the symmetric relationship between its subgradient and the subgradient of its conjugate. We can use this theorem to transform the subdifferential with regard to its conjugate.

Theorem 4.1.13. *([12]) If f is closed and convex, then*

$$\bar{x} \in \partial f(v) \ if \ and \ only \ if \ v \in \partial f^*(\bar{x})$$

Proof. This can be proved using the definitions of conjugate (definition 4.1.5) and subgradient(definition 4.1.6). □

Finding the subdifferential can be viewed as an optimization problem. The background on existence of optimal solution can help us discover the cardinality of the subgradient set.

Theorem 4.1.14. *([12]) Let f be a proper, lower semicontinuous function. If for any $\lambda \in \mathbb{R}$ the sublevel set $L_\lambda = \{x \in \mathbb{R}^n,\ f(x) \leq \lambda\}$ is bounded. Then f has an absolute minimizer.*

Using the property of lower semicontinuity, we can derive the following theorem.

Theorem 4.1.15. *([12]) Consider the optimization problem P in eq. (3.1). If Ω is nonempty, closed and bounded and f is lower semicontinuous. Then P has a global optimal solution.*

Definition 4.1.16. *Coercive. A function is called coercive if*

$$\lim_{\|x\| \to \infty} f(x) = \infty$$

Corollary 4.1.17. *([12]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a lower semicontinuous function and coercive function, then there exists an absolute minimizer for f.*

Using the property of convexity, we can derive the following theorem.

Lemma 4.1.18. *If f is proper and strictly convex, there exists at most one global minimizer for f. In another word,* $\operatorname{argmin}_{x \in dom(f)} f(x)$ *is a singleton if it is nonempty*

4.2 Connection between Stochastic and Barrier Methods

In this section, the mathematics backgrounds introduced above will be used to prove the differentiability of the Fenchel conjugate function theorem [17] which is a result of Fenchel duality. The proof is inspired by and built on a lecture [13]. Then the connection between the stochastic and the barrier method will be built based on the theorem.

Theorem 4.2.1. *([17, 3, 1]) Let f be a proper, strongly convex and lowers semicontinuous function. Then the Fenchel conjugate of f, f^* is Fréchet differentiable at any $v \in \mathbb{R}^n$ and $\nabla f^*(v) = \operatorname{argmax}_{x\in\mathbb{R}^n}\{\langle v, x\rangle - f(x)\}$*

Proof. By theorem 4.1.8, f is Fréchet differentiable if and only if the subgradient of f at v is a singleton for all v in the domain of f. Show that the subgradient of f at v is a singleton for all v in the domain of f.

For an arbitrary $v \in \mathbb{R}^n$. Consider the subgradient of at v. Suppose $\bar{x} \in \partial f^*(v)$. We have $v \in \partial f(\bar{x})$ (theorem 4.1.13).

Applying definition 4.1.6, it is equivalent to

$$\langle v, x - \bar{x}\rangle \leq f(x) - f(\bar{x}) \ \forall x \in dom(f)$$

$$\langle v, x\rangle - f(x) \leq \langle v, \bar{x}\rangle - f(\bar{x}) \ \forall x \in dom(f)$$

$$\bar{x} \in \operatorname*{argmax}_{x\in dom(f)} \{\langle v, x\rangle - f(x)\}$$

$$\bar{x} \in \operatorname*{argmin}_{x\in dom(f)} \{f(x) - \langle v, x\rangle\}$$

Since f is strongly convex(definition 4.1.3), f can be written as $f = g + \frac{\delta}{2}\|x\|^2$, where g is also proper, convex and lower semicontinuous, then there exists a support function(theorem 4.1.10)

$$\langle w, x\rangle + b \leq g(x), \ \forall x \in \mathbb{R}^n$$

$$\frac{\delta}{2}\|x\|^2 + \langle w, x\rangle + b \leq f(x), \ \forall x \in \mathbb{R}^n$$

Let $\phi(x) = f(x) - \langle v, x\rangle$,

$$\frac{\delta}{2}\|x\|^2 + \langle w - v, x\rangle + b \leq \phi(x), \ \forall x \in \mathbb{R}^n$$

Therefore $\lim_{\|x\|\to\infty} \phi(x) = \infty$. $\phi(x)$ is proper, coercive and lower semicontinuous. Then we have $\operatorname{argmin}_{x\in\mathbb{R}^n} \phi(x)$ is nonempty, since it is proper and lower semicontinuous from corollary 4.1.17.

From theorem 4.1.4. $\phi(x)$ is also strictly convex. Therefore

$$\bar{x} \in \operatorname*{argmin}_{x\in\mathbb{R}^n}\{f(x) - \langle v, x\rangle\}$$

is a singleton by lemma 4.1.18.

We have shown that the subgradient of f at v is a singleton for all v in the domain of f. Therefore, f is Fréchet differentiable and

$$\nabla f^*(v) = \operatorname*{argmax}_{x\in\mathbb{R}^n}\{\langle v, x\rangle - f(x)\}$$

□

Then we apply this theorem in the LP problem setting and discuss the stochastic and barrier methods.

Theorem 4.2.2. *The perturbation in the stochastic method gives a barrier function in the barrier method, so that two methods are equivalent as they give the same approximation of the LP.*

$$y_\epsilon^*(c) = \underset{y\in\Omega}{\text{argmax}}\{\langle c, y\rangle - F_\epsilon^*(y)\}$$

Proof. In the stochastic method, we assume the probability density function of the noise follows the form $d\mu(z) \propto exp(-v(z)dz)$.

$$F_\epsilon(c) = E[\min_{x\in\Omega}\langle x, c + \epsilon Z\rangle] \propto E[Z]$$

$E[z] = \mu(z) \propto exp(-v(z)dz)$ is continuous and strongly convex. Therefore we can apply theorem 4.2.1,

$$\nabla_c F_\epsilon(c) = \underset{y\in\Omega}{\text{argmax}}\{\langle c, y\rangle - F_\epsilon^*(y)\}$$

$$y_\epsilon^*(c) = \nabla_c F_\epsilon(c) = \underset{y\in\Omega}{\text{argmax}}\{\langle c, y\rangle - F_\epsilon^*(y)\}$$

$$y_\epsilon^*(c) = \underset{y\in\Omega}{\text{argmax}}\{\langle c, y\rangle - \epsilon F^*(y)\} = \underset{y\in\Omega}{\text{argmax}}\{\frac{1}{\epsilon}\langle c, y\rangle - F^*(y)\}$$

This theorem shows the equivalency between stochastic and barrier methods for linear programs. □

4.3 Stochastic Method with Gumbel Noise and Negative Entropy Barrier

In this section, we will study on one particular noise, Gumbel noise, which is widely used in the stochastic method. And we will derive the equivalent barrier function using previous theorems. The perturbed optimizer (Perturb-and-MAP model) resembles the Gibbs distribution (Gibbs Markov Random Field) [15]

Lemma 4.3.1. *([14]) Let $c = (c_1, ..., c_m) \in \mathbb{R}^n$ and perturbation of IID Gumbel samples $z_1, ...z_n$ is added to c. Then the distribution of the minimum of $\tilde{c} = (c + z)$ is*

$$Pr\{\text{argmin}(\tilde{c}_1, ..., \tilde{c}_n) = n\} = \frac{e^{-\tilde{c}_n}}{\sum_{i=1}^{m} e^{-\tilde{c}_i}}$$

From lemma 4.3.1, the minimizer has a distribution of $\frac{e^{-\tilde{c}_n}}{\sum_{i=1}^{m} e^{-\tilde{c}_i}}$, which is the derivative of the log-sum-exp.Since, the Gibbs distribution follows the form of log-sum-exp function, we can claim that Gumbel noise added to the vector c generates a minimum of Gibbs distribution.

Proposition 4.3.2. *The Fenchel conjugate of log-sum-exp is the negative entropy under simplex setting.*

Proof. Let $f(x) = \sum_{i=1}^{M} x_i lnx_i$, for $x \succeq 0$ be the negative entropy function. The conjugate of f is

$$f^*(v) = \sup_{x \succeq 0} \{\langle v, x\rangle - \sum_{i=1}^{M} x_i lnx_i\}$$

Let $g(x) = \sum_{i=1}^{M} v_i x_i - x_i lnx_i$

$$\nabla_x g = \sum_{i=1}^{M} v_i - lnx_i - 1$$

Solve for $\nabla_x g = 0$, $x_i^* = e^{(v_i - 1)}$.

$$g(x) \leq g(x^*), \ \forall x \succeq 0$$

Therefore,

$$f^*(v) = g(x^*) = \sum_{i=1}^{M} v_i e^{(v_i - 1)} - (v_i - 1) v_i e^{(v_i - 1)}$$

$$f^*(v) = \sum_{i=1}^{M} e^{(v_i - 1)}$$

Assume $\sum_{i=1}^{M} v_i = 1$,

$$f^*(v) = \sum_{i=1}^{M} \frac{e^{(v_i)}}{\sum_{i=j}^{M} e^{(v_j)}}$$

which is the log-sum-exp function. □

So far, we learn that the the perturbed minimum obtained from stochastic with Gumbel noise resembles log-sum-exp function. And we found that negative entropy function is the conjugate of log-sum-exp under the simplex setting. A concrete equivalent pair of stochastic method and barrier method has been found.

Chapter 5

Experiment

This chapter consists of two experiments on application of the barrier and the stochastic methods. The first experiment investigate the solution approximated by barrier and stochastic methods. The second one compares the learning processes of singleton deep declarative network implemented by barrier and stochastic methods. Both experiments are designed to verify the equivalency between theses methods.

The barrier and stochastic methods are implemented in Python with the use of CVXPY, a Python-embedded modeling language for convex optimization problems. Firstly, we define a LP solver which uses the CVXPY to solve a LP problem in the form of eq. (3.2).

LP Solver(A, b, c)

Input:

$A \leftarrow$ constant parameter of the LP problem

$b \leftarrow$ constant parameter of the LP problem

$c \leftarrow$ constant parameter of the LP problem

objective $= c^T x$

constraints $= A^T x \preceq b$

problem $=$ convex_optimization_problem(objective, constraints)

$x_{sol} =$ solve(problem)

Return: x_{sol}

Algorithms for the experiments are given in pseudo code. Here is an example algorithm for the LP problem solver.

5.1 Approximate Solutions of LP problems

This experiments compares the approximate solutions of LP problems obtained from barrier and stochastic methods in terms of Euclidean distance. Based on the description of two methods in (ref), the pseudo code for two methods are defined below.

Barrier Method Approximation(A, B, c, t, f)

Input:

$A \leftarrow$ constant parameter of the LP problem
$B \leftarrow$ constant parameter of the LP problem
$c \leftarrow$ Learning parameter of the LP problem
$t \leftarrow$ barrier coefficient
$f \leftarrow$ barrier function

objective $f_0 = tc^T x - f(A, B, x)$
problem = convex_optimization_problem(objective)
x_{sol} = solve(problem)
Return: x_{sol}

The barrier method takes in the parameter of an LP problem, a barrier strength parameter and and barrier function. Adding the barrier function to the objective implicitly impose the constraints of the original problem and the barrier method solves for the convex problem with the new objective and no constraints.

Stochastic Method Approximation$(A, B, c, \epsilon, noise_samples)$

Input:

$A \leftarrow$ constant parameter of the LP problem
$B \leftarrow$ constant parameter of the LP problem
$c \leftarrow$ Learning parameter of the LP problem
$\epsilon \leftarrow$ temperature parameter
$D \leftarrow$ Noise distribution
$N \leftarrow$ sample size

$x_{sols} \leftarrow$ A list of solutions for LP problem perturbed by noise samples
for all Z in *noise_samples* **do**
  $\tilde{c} = \epsilon Z + c$
  objective $f_0 = \tilde{c}^T x$
  constraints = $A^T x \preceq b$
  problem = convex_optimization_problem(objective, constraints)
  append solve(problem) *to* x_{sols}
end for
x_{sol} = mean(x_{sols})
Return: x_{sol}

The stochastic method needs the parameters for the LP problem and the temperature parameter for the perturbation. In addition, the noise samples gives information on the noise distribution and sample size. The solution is obtained from averaging the solution of LP problems with perturbed objectives, which is the *Monte-Carlo estimate* of the perturbed solution.

In this experiment, a test set of randomized LP problems are firstly generated and then solved using both methods under different parameter settings. Then, we compute the averaged solution of all set of problems for both methods. Lastly, the Euclidean distance between the average solutions are compared to study on the equivalency between these two methods. The result is showed heatmaps below.

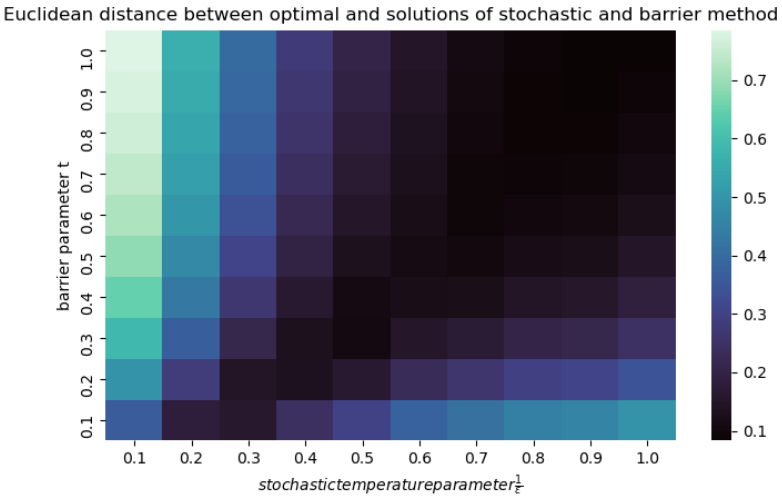

Figure 5.1: Euclidean distance between log barrier method and stochastic method with Gumbel noise experimented on set of general LP problems

As the heatmap shows, the Euclidean distances between averaged solutions from barrier method and stochastic method are smaller on the diagonal. This indicates that barrier method produces similar output as the the stochastic method when the barrier parameter t is proportional to the reciprocal of the temperature parameter ϵ of the stochastic method which conforms with the theorytheorem 4.2.2.

In the theory, the negative entropy barrier is an exact counterpart for the Gumbel noise in the stochastic method under the simplex setting. So, I generate a test set of LP problem where the constraints form a simplex and tests both logarithm barrier and negative entropy barrier. The results are shown below.

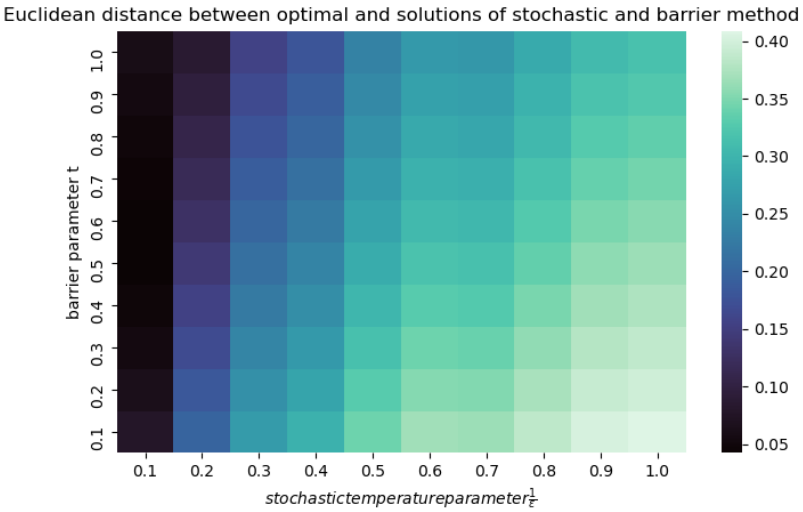

Figure 5.2: Euclidean distance between log barrier method and stochastic method with Gumbel noise experimented on set of simplex LP problems

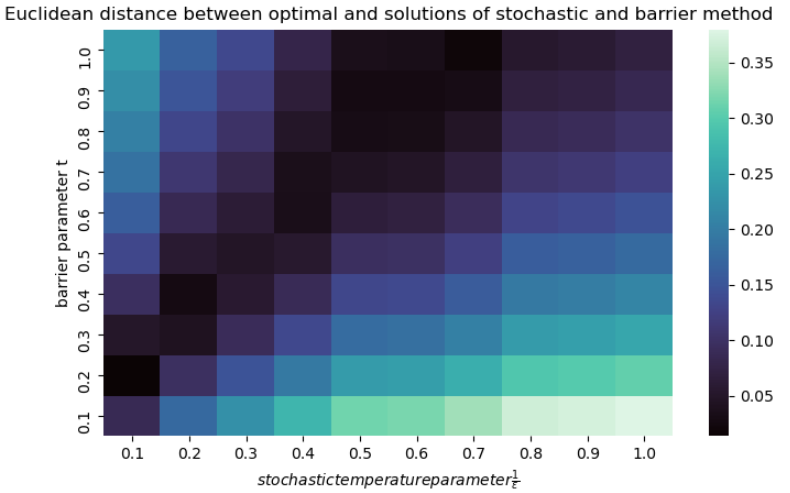

Figure 5.3: Euclidean distance between negative entropy barrier method and stochastic method with Gumbel noise experimented on set of simplex LP problems

Compared to the result of log barrier in fig. 5.2, in fig. 5.3 we can see a positive correlation of negative entropy barrier parameter t and reciprocal of the temperature parameter ϵ. This pattern in fig. 5.3 is also clearer than the pattern in fig. 5.1 as the deep diagonal area is narrower, which evidence that the correlation between the parameters is stronger under simplex setting than general LP problems. Therefore, this result further verifies the equivalency between negative entropy barrier method and stochastic method with Gumbel noise.

5.2 Parameter Learning

In the second experiment, the learning processes of declarative nodes (singleton DDN) defined by LP problems using barrier and stochastic methods are studied. Firstly an simplex LP problem is generated and the ground truth x^* is obtained by solving the LP problem. Then, the parameter c in LP problem is randomized and the new LP problem is used as input to the singleton DDN. The objective function is set to be half of the norm of the ground truth and the output of the singleton DDN. Out goal is to learn the parameter c that produces the ground truth as output. A simple gradient descent method [4] is used for the learning process.

---

gradient descent

---

given a starting point $x \in \mathbf{dom} f$, a starting step size $t > 0$, and decay rate $0 < \gamma \leq 1$

repeat

1. $\Delta x_{nsd} := -\nabla f(x)/\|\nabla f(x)\|$.
2. if $x + t\Delta x_{nsd}$ feasible, then $x := x + t\Delta x_{nsd}$.
3. $t := \gamma t$.

until stopping criterion is met

The pseudo codes of these two methods are shown below which include a forward pass and a backward pass like a DNN node. In the backward pass, we can derive the gradient using chain rule of differentiation. The gradient of the loss function times the gradient of the declarative node which can be computed using differentiating algorithms given in section 3.3 and section 3.6 with practical implementation definition 3.6.2.

Barrier Method Node(A, B, c, t, f, x^*)

Input:

$A \leftarrow$ constant parameter of the LP problem
$B \leftarrow$ constant parameter of the LP problem
$c \leftarrow$ Learning parameter of the LP problem
$t \leftarrow$ barrier coefficient
$f \leftarrow$ barrier function
$x^* \leftarrow$ true optimal solution

Forward Pass:

x_{sol} = Barrier Method Approximation(A, B, c, t, f)
Return: $\frac{1}{2}\|x_{sol} - x^*\|^2$

Backward Pass:

Return: $(x_{sol} - x^*) \cdot -(\nabla_{xx} f_0)^{-1} \nabla xc f_0$

Stochastic Method Node($A, B, c, \epsilon, D, N, x^*$)

Input:

$A \leftarrow$ constant parameter of the LP problem
$B \leftarrow$ constant parameter of the LP problem
$c \leftarrow$ Learning parameter of the LP problem
$\epsilon \leftarrow$ temperature parameter
$D \leftarrow$ Noise distribution which follows $d\mu(z) \propto exp(-v(z)dz)$
$N \leftarrow$ sample size

noise_samples $\leftarrow$ *draw N samples from distribution D*

Forward Pass:

x_{sol} =Stochastic Method Approximation(A, B, c, ϵ, D, N)
Return: $\frac{1}{2}\|x_{sol} - x^*\|^2$

Backward Pass:

grads $\leftarrow$ A list of gradients for LP problem perturbed by noise samples
for all Z in *noise_samples* **do**
  x_{sol_0} =Stochastic Method Approximation(A, B, c, ϵ, D, N)
  $grad_0 = x_{sol_0} \cdot \nabla_z v(Z)^T / \epsilon$
  append $grad_0$ to grads
end for
Return: $(x_{sol} - x^*) \cdot$ mean($grads$)

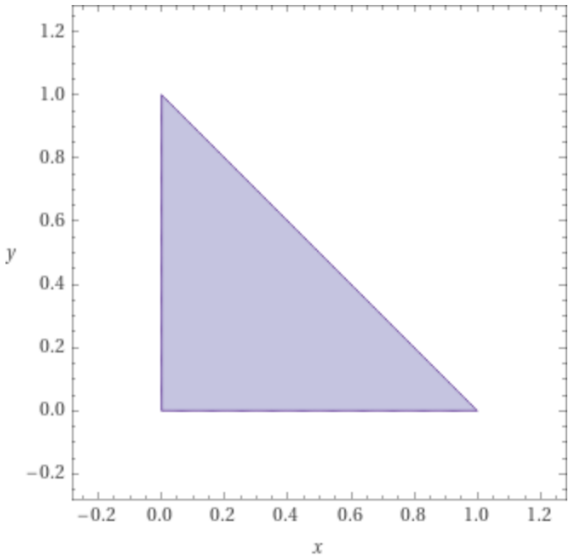

Figure 5.4: Feasible region of the LP problem in experiment 2

The fig. 5.4 shows the feasible region of the LP problem, which is a simplex in $\mathbb{R}^2$. The learning curve obtained from different implementation of the declarative node is shown below.

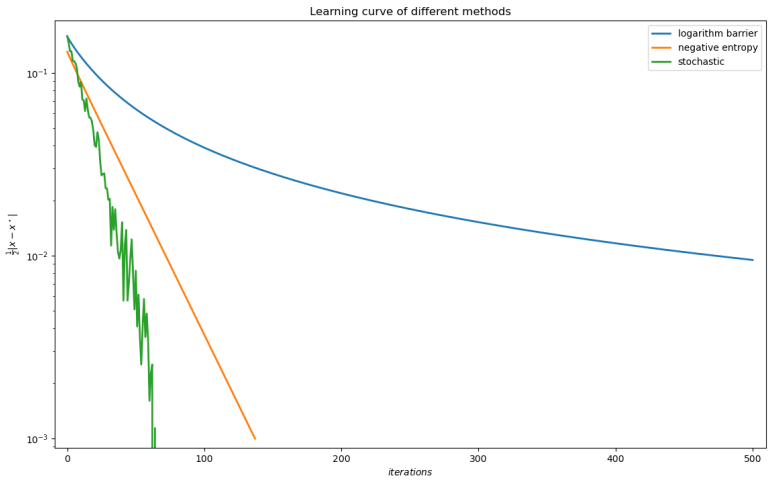

Figure 5.5: Learning of curves of declarative node implemented by different methods.Ground Truth: $c = [3, 1]^T$ Initial: $c_0 = [1, 2]^T$

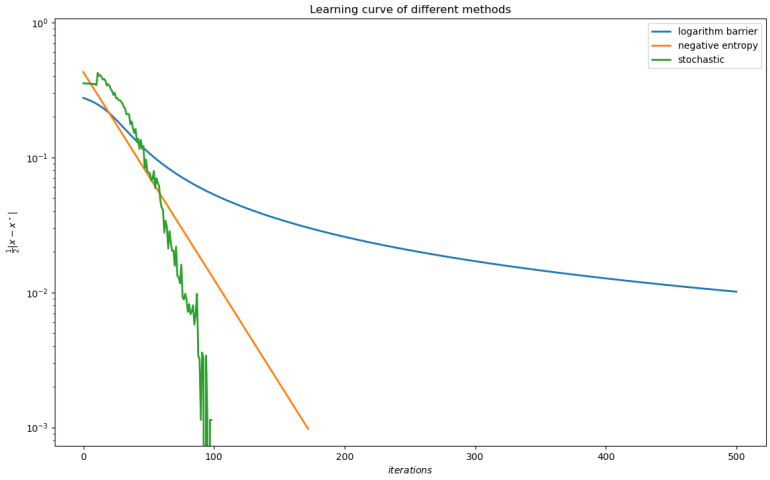

Figure 5.6: Learning of curves of declarative node implemented by different methods. Ground Truth: $c = [3, 1]^T$ Initial: $c_0 = [-1, -1]^T$

As the fig. 5.5 and fig. 5.6 show, the learning curve of the stochastic method converges fastest but fluctuates a lot as we use the *Monte-Carlo estimate* as the gradient. Moreover, the declarative node implemented by the negative entropy barrier method and stochastic method with Gumbel noise converge well and has similar converge rates, while the node implemented by the log barrier method converges much slower and the training loss (around 0.01) of log barrier is much higher. We can conclude that the negative entropy barrier method and stochastic method with Gumbel noise have close effects on the parameter learning of declarative nodes composed of LP problems.

There is a special case where the stochastic method could fail to converge. When the objective coefficient is set to be perpendicular to the boundary of the feasible region $c = [-1, -1]$, it generates multiple optimal solutions and we randomly choose one vertex as the ground truth optimal solution x^*. We can see from fig. 5.7, that the log barrier and negative entropy barrier method converges faster than in the previous settings (fig. 5.5, fig. 5.6), while the stochastic method oscillates greatly and get stuck at a loss higher than the original loss after about 350 iterations.

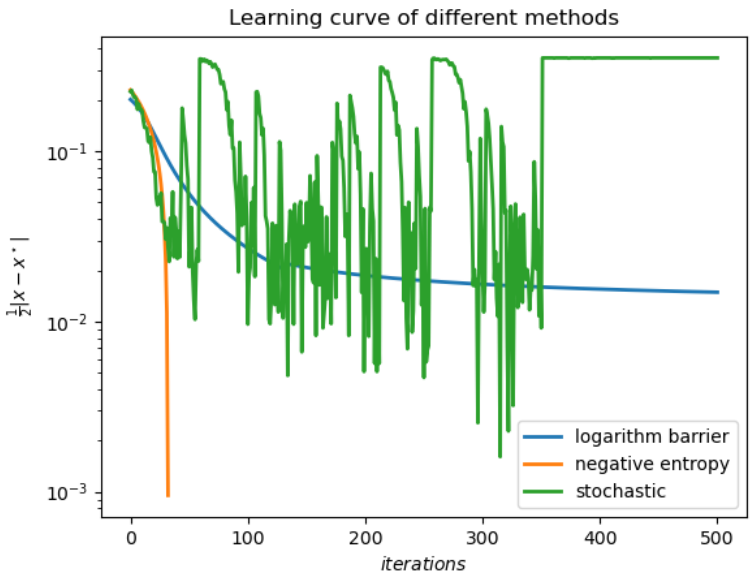

Figure 5.7: Learning of curves of declarative node implemented by different methods. Ground Truth: $c = [-1, -1]^T$ Initial: $c_0 = [1, 2]^T$

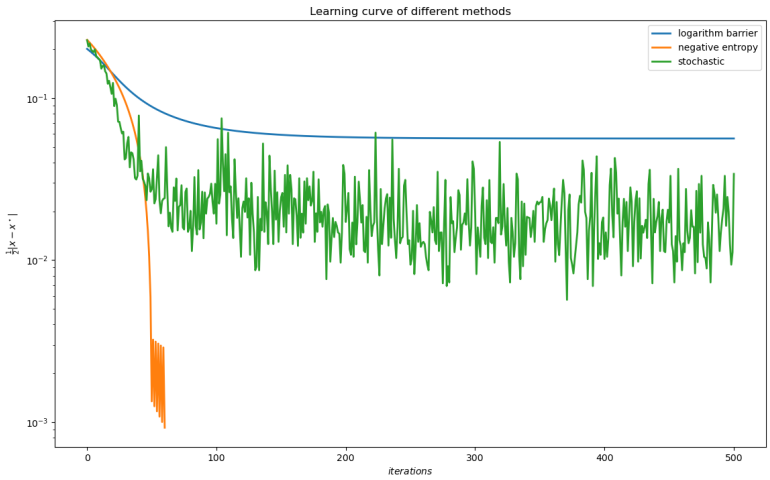

Figure 5.8: Learning of curves of declarative node implemented by different methods. Ground Truth:: $c = [-1, -1]^T$ Initial $c_0 = [1, 2]^T$ Decay Parameter: $\gamma = 0.98$

This can be mitigated by setting the decay parameter γ in the gradient descent algorithm less than 1 gradually decreasing the step size$t$. As shown in fig. 5.8, it converges better than in fig. 5.7 but still does not converge as well as in the previous settings (fig. 5.5, fig. 5.6).

In real-life DDN applications, it is unlikely for the objective coefficient to be perpendicular to the boundary of the feasible region in a higher dimension. Yet, we still need to find a method to cope with this special case.

Chapter 6

Discussion

The connection between the Stochastic method and the barrier method is built based on theorem 4.2.1 which is a classic outcome in convex analysis [17]. Like previous works [1, 9, 8] such a connection can be generalized into the equivalency of perturbation and regularization. This idea has brought stochastic algorithms into deep learning models seeking higher efficiency in the computation of the model. Though this topic has been analyzed and applied by many people, our study provides a detailed proof of such an equivalency and how the connection between the stochastic method and the barrier method can be built when solving LP problems. Moreover, a concrete example of the equivalent stochastic method and barrier method has been derived which is built on the relationship between Gumbel perturbation and the Gibbs distribution[15]. Experiments that verify the connection have also been shown in our study. In the experiment, declarative nodes implemented by both methods are given, which could be helpful when building a DDN. This report can serve as a study guide on related perturbation and convex analysis topics.

Although the theoretical connection between the two methods has been proved, it is hard to apply the theorem to find the counterpart given the barrier function or the noise distribution. The difficulties are encountered when finding the conjugate function of the perturbed optimal which is an expectation over the noise and has no closed-form solution.

For future works, it would be helpful to find a closed-form approximation of the perturbed optimal so that we can compute the conjugate function of the approximated form and obtain an equivalent barrier function. On the other hand, we could also develop an algorithm the generate noise given a perturbed optimal so that we can compute the conjugate of the barrier function and generate noise from it to obtain the equivalent stochastic method. Moreover, the effects of different barrier functions and perturbations on the approximated solution of the LP problem and the learning process of the declarative node composed of an LP problem are worth studying.

Bibliography

[1] J. Abernethy, C. Lee, A. Sinha, and A. Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory*, pages 807–823. PMLR, 2014.

[2] J. Abernethy, C. Lee, and A. Tewari. Perturbation techniques in online learning and optimization. *Perturbations, Optimization, and Statistics*, page 223, 2016.

[3] Q. Berthet, M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, and F. Bach. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems*, 33:9508–9519, 2020.

[4] S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[5] S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

[6] S. Gould, R. Hartley, and D. J. Campbell. Deep declarative networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[7] E. J. Gumbel. Statistics of extremes. In *Statistics of Extremes*. Columbia university press, 1958.

[8] T. Hazan, G. Papandreou, and D. Tarlow. *Perturbations, Optimization, and Statistics*. MIT Press, 2016.

[9] J. Hofbauer and W. H. Sandholm. On the global convergence of stochastic fictitious play. *Econometrica*, 70(6):2265–2294, 2002.

[10] A. G. Ivakhnenko, A. G. Ivakhnenko, V. G. Lapa, and V. G. Lapa. *Cybernetics and forecasting techniques*, volume 8. American Elsevier Publishing Company, 1967.

[11] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[12] B. S. Mordukhovich and N. M. Nam. An easy path to convex analysis and applications. *Synthesis Lectures on Mathematics and Statistics*, 6(2):1–218, 2013.

[13] N. M. Nam. Differentiability of the fenchel conjugate, 2019. URL https://www.youtube.com/watch?v=jUB0JktXqHE.

[14] G. Papandreou and A. Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models–iccv 2011 paper supplementary material–. 2011.

[15] G. Papandreou and A. L. Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *2011 International Conference on Computer Vision*, pages 193–200. IEEE, 2011.

[16] J. V. ROBERT. *Linear programming: Foundations and extensions*. Springer, 2021.

[17] R. T. Rockafellar. Convex analysis. In *Convex analysis*. Princeton university press, 2015.

[18] J. Schmidhuber. Habilitation thesis: System modeling and optimization. *Page 150 ff demonstrates credit assignment across the equivalent of 1,200 layers in an unfolded RNN*, 1993.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.