Bayesian Methods for Tuning Hyperparameters of Loss Functions in Machine Learning

by

Erika Lingo Cole

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Arts

Approved April 2022 by the
Graduate Supervisory Committee:

Lalitha Sankar, Co-Chair
Shiwei Lan, Co-Chair
Giulia Pedrielli
Paul Hahn

ARIZONA STATE UNIVERSITY

May 2022

ABSTRACT

The introduction of parameterized loss functions for robustness in machine learning has led to questions as to how hyperparameter(s) of the loss functions can be tuned. This thesis explores how Bayesian methods can be leveraged to tune such hyperparameters. Specifically, a modified Gibbs sampling scheme is used to generate a distribution of loss parameters of tunable loss functions. The modified Gibbs sampler is a two-block sampler that alternates between sampling the loss parameter and optimizing the other model parameters. The sampling step is performed using slice sampling, while the optimization step is performed using gradient descent. This thesis explores the application of the modified Gibbs sampler to alpha-loss, a tunable loss function with a single parameter $\alpha \in (0, \infty]$, that is designed for the classification setting. Theoretically, it is shown that the Markov chain generated by a modified Gibbs sampling scheme is ergodic; that is, the chain has, and converges to, a unique stationary (posterior) distribution. Further, the modified Gibbs sampler is implemented in two experiments: a synthetic dataset and a canonical image dataset. The results show that the modified Gibbs sampler performs well under label noise, generating a distribution indicating preference for larger values of alpha, matching the outcomes of previous experiments.

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisors, Dr. Lalitha Sankar, Dr. Shiwei Lan, and Dr. Giulia Pedrielli, for the opportunity to work on my research project and for their guidance and support throughout my graduate journey.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Loss functions are an important measure that can drastically change the performance of a machine learning algorithm (Weerts et al., 2020). For example, loss functions such as mean squared error and log-loss are commonly used. These commonly used loss functions are convex. However, convex loss functions can sometimes be problematic in that they do not generalize as well in the face of noisy data (Mei et al., 2018). In view of a growing desire for robustness in machine learning models, there has been increased interest in parameterized loss functions. Namely, this thesis focuses on the application of alpha-loss, a tunable loss function with single parameter $\alpha \in (0, \infty]$ that acts on probability distributions (Sypherd et al., 2019). The benefit of alpha-loss is that it interpolates between common loss functions: exponential-loss, log-loss, and 0-1 loss. This allows for a varying range of common loss functions.

With the introduction of alpha-loss comes the addition of another hyperparameter; thus, a natural question arises as to how the loss parameter should be tuned. Many methods of hyperparameter tuning exist, from the most basic methods, such as grid or random search, to more sophisticated methods, such as Bayesian Optimization (Bischl et al., 2021). In general, there are two classes of hyperparameter tuning: methods that generate a point estimate and methods that generate distributions of the hyperparameter. Examples of the former include grid or random search and Bayesian optimization, while an example of the latter is Markov Chain Monte Carlo (MCMC).

In this thesis, we explore the latter methods and specifically explore the use of MCMC methods in tuning the hyperparameter α of α-loss. In particular, we introduce a modified Gibbs sampling method that alternates between tuning α and all other model parameters. The modified Gibbs sampler is ultimately applied to learn the distribution of α of α-loss. We then compare the obtained distributions to work done in (Sypherd et al., 2021), which finds that larger values of α perform better for data compromised with label noise.

The subsequent chapters are as follows. Section 2 provides background on relevant concepts, including alpha-loss, MCMC, and several common MCMC methods. Section 3 introduces the modified Gibbs sampler in detail, as well as important theoretial properties. Section 4 details experimental results of the modified Gibbs sampler applied to a synthetic dataset and a canonical machine learning image classification dataset. Finally, Section 5 presents conclusions and directions of future work.

Chapter 2

BACKGROUND: LOSS FUNCTIONS & SAMPLING METHODS

2.1 Alpha-Loss

α-loss is a family of tunable loss functions that acts on probabilities. Formally, let $\mathcal{P}(\mathcal{Y})$ be the set of probabilities over $\mathcal{Y}$. For $\alpha \in (0, \infty]$, α-loss is defined for $\alpha \in (0,1) \cup (1,\infty)$, $l^{\alpha} : \mathcal{Y} \times \mathcal{P}(\mathcal{Y}) \to \mathbb{R}_{+}$ as

$$l^{\alpha}(y, P_Y) \coloneqq \frac{\alpha}{\alpha - 1}\left[1 - P_Y(y)^{1-1/\alpha}\right] \tag{2.1}$$

and by continuous extension, we have $l^{1}(y, P_Y) = -logP_Y(y)$ and $l^{\infty}(y, P_Y) = 1 - P_Y(y)$, i.e., log-loss and soft 0-1 loss, respectively (Sypherd et al., 2019). We observe that α-loss is continuous in α and can be interpreted as a class of loss functions that value the probabilistic estimate of the label differently as a function of α (Sypherd et al., 2019). Previous experiments have shown that α-loss is robust to label noise for $\alpha > 1$ and can be sensitive to class imbalance for $\alpha < 1$ (Sypherd et al., 2021).

More specifically, this thesis explores two settings of binary classification. For the classification setting, it is common to work with margin-based loss functions. That is, for all $x \in \mathcal{X}$ and labels $y \in \{-1, +1\}$, the loss is dependent only on the product $z = y \cdot f(x)$, where $f : \mathcal{X} \to \bar{\mathbb{R}}$ is the classification function. Note that the margin z is positive for a correct classification, as the sign of the prediction and true class are the same; while the margin is negative for an incorrect classification, as the sign of the prediction and true class differ. The margin-based α-loss for $\alpha \in (0,1) \cup (1,\infty)$, $\tilde{l}^{\alpha} : \bar{\mathbb{R}} \to \mathbb{R}$ is

$$\tilde{l}^{\alpha}(z) \coloneqq \frac{\alpha}{\alpha - 1}\left(1 - (1 + e^{-z})^{1/\alpha - 1}\right), \tag{2.2}$$

and by continuous extension, $\tilde{l}^1(z) = log(1 + e^{-z})$ (logistic loss) and $\tilde{l}^\infty = (1 + e^z)^{-1}$ (sigmoid loss) (Sypherd et al., 2019). Further, work done in (Sypherd et al., 2021) has shown that there considerable performance gains to using $\alpha > 1$ in the case of label noise. In our work, we aim to prove this result by showing that larger values of α are preferred in the setting of label noise.

2.2 Markov Chain Monte Carlo

Markov Chain Monte Carlo methods define a sequence of samples for which the distributions eventually settle down to the posterior distribution. As the number of samples increases, the densities approach the target distribution, or as is often the case in Bayesian inference, the posterior distribution, $\pi(\theta \mid \mathbf{y})$, which we henceforth abbreviate as $\pi(\theta)$. This construction of samples is helpful because it allows the practitioner to sample from complex posterior distributions that may otherwise be difficult to sample from directly (Christensen et al., 2010).

The Markov chain generates a sequence of identically distributed samples with density $\pi(\theta)$. Further, under certain conditions, by the Ergodic Theorem, the samples, $\theta^1, \ldots, \theta^k$ and function h with finite expectation under the posterior distribution, satisfy

$$\lim_{k\to\infty} \sum_{j=1}^{k} h(\theta^j)/k = \int h(\theta)\pi(\theta)\, d\theta. \tag{2.3}$$

Thus, the Markov chain generates samples that can be used to approximate probabilities and expected values by computing functions of the θ^ks. MCMC is often used in fields such statistics, economics, physics, and computer science (Andrieu et al., 2003). MCMC methods constitute a family of sampling algorithms. We describe several relevant MCMC algorithms in the subsequent sections.

2.2.1 Gibbs Sampler

Gibbs sampling is an MCMC method that generates a Markov chain using the full conditional distributions of each parameter given the other parameters. Each parameter is then updated sequentially. After each update, one iteration of the Gibbs sampler is complete. Suppose we start with a set of n-parameters, $\theta^1 = [\theta_1^1, \ldots, \theta_j^1, \ldots, \theta_n^1]$, then the next sample of parameters is obtained (for $k = 2$) by

$$\theta_1^k \mid \theta_2^{k-1}, \ldots, \theta_n^{k-1} \sim p(\theta_1 \mid \theta_2^{k-1}, \ldots, \theta_n^{k-1}) \tag{2.4}$$

$$\theta_j^k \mid \theta_1^k, \ldots, \theta_{j-1}^k, \theta_{j+1}^{k-1}, \ldots, \theta_n^{k-1} \sim p(\theta_j \mid \theta_1^k, \ldots, \theta_{j-1}^k, \theta_{j+1}^{k-1}, \ldots, \theta_n^{k-1}) \tag{2.5}$$

$$\theta_n^k \mid \theta_1^k, \ldots, \theta_{n-1}^k \sim p(\theta_n \mid \theta_1^k, \ldots, \theta_{n-1}^k). \tag{2.6}$$

The Markov chain obtained through this process defines a valid transition distribution that does not depend on k. The stationary transition distribution is

$$q(\theta^k \mid \theta^{k-1}) \equiv p(\theta_1 \mid \theta_2^{k-1}, \ldots, \theta_n^{k-1}) \cdot \ldots \cdot p(\theta_n \mid \theta_1^k, \ldots, \theta_{n-1}^k) \tag{2.7}$$

i.e. the product of the full conditional distributions. Given this transition distribution, q, the posterior is the stationary distribution (Christensen et al., 2010). That is, running the Gibbs sampler for a sufficiently long time produces a sample of values $\theta^1, \ldots, \theta^k$ from the target (joint posterior) distribution.

2.2.2 Slice Sampler

Motivated by the idea that one can sample from a univariate distribution by sampling points uniformly from the region under the curve of its density function and then analyze the horizontal coordinates of the sampled points, slice sampling is a specific version of a Gibbs sampler (Neal, 2003). Slice sampling alternates between sampling uniformly from the vertical interval defined at the current density and the horizontal interval or “slice” of the current vertical position. Formally, for a target density f and auxiliary variable u, the slice sampling algorithm is

1. Starting with state (x^0, u^0)

2. sample $u^{i+1} \mid x^i \sim Unif_{(0,f(x^i))}(u)$

3. sample $x^{i+1} \mid u^{i+1} \sim Unif_A(x)$ where $A = \{x; f(x) \geq u^{i+1}\}$

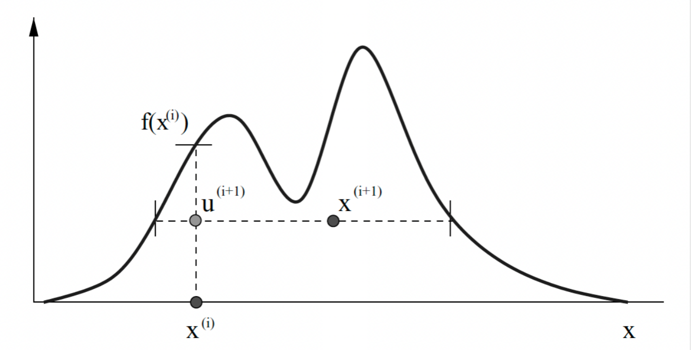

Figure 2.1: A Visual Representation of Slice Sampling. Given a Previous Sample, $x^{(I)}$, We Sample a Uniform Variable $u^{(I+1)}$ Between 0 and $f(X^{(I)})$. One Then Samples $x^{(I+1)}$ Uniformly in the Interval Where $f(X) \geq U^{(I+1)}$ (Andrieu et al., 2003).

A visual representation of the slice sampling algorithm is shown in Figure 2.1. Slice sampling can be advantageous in that it does not have a step size to tune as some methods, such as Metropolis Hastings, require. Further, slice sampling is particularly effective for lower-dimensional parameter spaces (Neal, 2003).

2.2.3 Hybrid Gibbs Sampler

Use of the classical Gibbs sampler necessitates that one can sample from the full conditional distributions. However, this is not always the case. For situations in which the full conditional distributions cannot be sampled from, we can make use of Metropolis-Hastings or slice sampling to sample from a full conditional distribution. These methods are known as Hybrid Gibbs sampling, or Metropolis-within-Gibbs/slice sampling-within-Gibbs . This adds to the flexibility of the Gibbs sampler, as it can be applied to non-standard cases where the full conditionals do not take a standard form or are not easy to sample from (Andrieu et al., 2003). In this work, we discuss such a Hybrid Gibbs sampler, which makes use of the slice sampling within Gibbs method. We discuss the Hybrid Gibbs algorithm and its implementation in the upcoming section.

Chapter 3

HYPERPARAMETER TUNING METHODS

3.1 Modified Gibbs Sampler

The modified Gibbs sampler is a two-block Gibbs sampler that is specifically designed to tune the hyperparameter α of α-loss. For any given model, suppose we have the parameter set $\theta = \{\alpha, w_1, \ldots, w_n\}$. The parameters are broken into two blocks: $\theta_1 = \{\alpha\}$ and $\theta_2 = \{w_1, \ldots, w_n\} = \{\bar{w}\}$. From this, the modified Gibbs sampler alternates between updating α and the remaining parameters, $\bar{w}$, through their full conditional distributions, as represented in the following scheme.
Starting with $\theta^1 = \{\alpha^1, \bar{w}^1\}$, we obtain the k-th iteration by

$$\alpha^k \mid \bar{w}^{k-1} \sim p_{\alpha|\bar{w}}(\alpha \mid \bar{w}^{k-1}) \tag{3.1}$$

$$\bar{w}^k \mid \alpha^k \sim p_{\bar{w}|\alpha}(\bar{w} \mid \alpha^k) \tag{3.2}$$

which then yields the k-th sample, $\theta^k = \{\alpha^k, \bar{w}^k\}$. The modifications to the Gibbs sampler appear in how the full conditionals are defined and sampled from. The sampling scheme employed by the modified Gibbs sampler is an adaption of the Hybrid Gibbs sampler. It first uses Slice sampling to sample from $p(\alpha \mid \bar{w})$ and then performs an optimization step to optimize over the weights $\bar{w}$. The latter step uses gradient descent to optimize $\bar{w}$ which results in $p(\bar{w} \mid \alpha)$ being a point-mass density, $\delta_{w*(\alpha)}(\bar{w})$, where $w^*(\alpha) = \arg\max_{\bar{w}} p(\bar{w} \mid \alpha)$. The choice to use gradient descent within Gibbs rather than sampling the full conditional, $p(\bar{w} \mid \alpha)$, was made in order to save resources, as MCMC methods are known to be computationally intensive. By using the existing, standard method of gradient descent, we aim to save resources.

Next, let us define the first conditional density of α with respect to $\bar{w}$. We now make the dependence on the data, $y_1, \ldots, y_n$, explicit. Assuming that $\bar{w}$ is known, the full conditional of α is

$$\begin{aligned} p(\alpha \mid y_1, \ldots, y_n, \bar{w}) &= p(y_1, \ldots, y_n \mid \alpha, \bar{w}) \times p(\alpha \mid w)/p(y_1, \ldots, y_n \mid \bar{w}) && (3.3) \\ &\propto p(y_1, \ldots, y_n \mid \alpha, \bar{w}) \times p(\alpha \mid w) && (3.4) \\ &\propto exp(-l^{\alpha}) \times \mathcal{N}(\bar{\alpha}, \sigma_{\alpha}^2)\mathbb{I}[\alpha_{min}, \alpha_{max}] && (3.5) \end{aligned}$$

where α-loss is incorporated into the full conditional by defining the joint density to be $exp(-l^{\alpha})$ and the conditional prior is taken to be a truncated normal with range of α depending on the specific dataset. Note that $\bar{\alpha}$ denotes the mean of $\alpha \in [\alpha_{min}, \alpha_{max}]$ and σ_{α}^2 denotes the variance of the truncated Gaussian prior. This conditional density, $p(\alpha \mid \bar{w})$, is sampled via slice sampling, as the method is known to be particularly advantageous to lower dimensional parameter spaces. Slice sampling also has the added advantage that it does not have a step size to tune as methods such as Metropolis-Hastings (Neal, 2003).

The full modified Gibbs sampling scheme then becomes:
Given a starting value, $(\alpha^1, \bar{w}^1)$

$$\begin{aligned} \alpha^k \mid \bar{w}^{k-1} &\sim Slice(p_{\alpha|\bar{w}}(\alpha \mid \bar{w}^{k-1})) && (3.6) \\ \bar{w}^k \mid \alpha^k &= \arg\max_{\bar{w}} p_{\bar{w}|\alpha}(\bar{w} \mid \alpha^k). && (3.7) \end{aligned}$$

To show that this sampling procedure generates a proper Markov chain that can be used for MCMC analysis, we verify several theoretical properties of the modified Gibbs sampler in the following section.

3.1.1 Theoretical Properties

A motivating theorem of MCMC methods is the Ergodic theorem, as this guarantees that the samples generated through our Markov chain can be treated as samples from the posterior, despite the samples not being completely independent (Christensen et al., 2010). The Ergodic theorem also ensures that the transition kernel has a unique stationary distribution, which is important, as it ensures that our samples are drawn from the correct target distribution. The Ergodic theorem is stated below.

Theorem 1 ((Christensen et al., 2010), Ergodic Theorem) *If $\theta^1, \dots, \theta^k$ are sampled from the Markov hain and h is a function with finite expectation under the stationary distribution, then with probability one*

$$\lim_{k\to\infty} \sum_{j=1}^{k} h(\theta^j)/k = \int h(\theta)p(\theta)\, d\theta. \tag{3.8}$$

Thus we can approximate probabilities and expected values relative to the stationary distribution.

Additionally, the definition of an ergodic Markov chain is:

Definition 1 ((Tierney, 1994), Definition of ergodicity) *A Markov chain is called ergodic if it is Harris recurrent and aperiodic.*

We emphasize that ergodicity is a dual faceted condition. That is, ergodicity of a Markov chain implies that there is a unique stationary distribution that is equal to the posterior distribution and, secondly, that the Markov chain is guaranteed to converge to the stationary distribution. Additional conditions for convergence are stated in Theorem 2 below.

Theorem 2 ((Tierney, 1994), Theorem 1) *Suppose P is π-irreducible and $\pi P = \pi$. Then P is positive recurrent and π is the unique invariant distribution of P. If P*

is also aperiodic, then for π-almost all x,

$$||P^n(x, \cdot) - \pi|| \to 0 \tag{3.9}$$

with $|| \cdot ||$ denoting the total variation distance. If P is Harris recurrent, then the convergence occurs for all x.

By the definition of ergodicity, a Markov chain that is aperiodic and Harris recurrent is ergodic. We use these properties to prove that the modified Gibbs sampler generates an ergodic Markov chain. Additionally, we can verify that the posterior is the stationary distribution of the Markov chain generated by the modified Gibbs sampler. We call this second property stationarity. Thus, to prove ergodicity and stationarity, we use the following two sufficient conditions conditions from Christensen et al. (2010).

Condition 1: *If the following holds*

$$\int_A \pi(\theta^k)\, d\theta^k = 0 \quad \text{if and only if} \int_A P(\theta^k \mid \theta^1)\, d\theta^k = 0, \quad \forall \theta^1, \tag{3.10}$$

then the Markov chain is aperiodic, pi-irreducible, and Harris recurrent (and also ergodic, by definition).

Condition 2: *If the following holds*

$$\int_A \pi(\theta^{k-1}) P(\theta^k \mid \theta^{k-1})\, d\theta^{k-1} = \pi(\theta^k), \tag{3.11}$$

then π is the stationary distribution of the Markov chain with transition kernel P.

Once conditions 1 and 2 above are shown to be true, then the Markov chain generated by the modified Gibbs sampler is ergodic, the Ergodic theorem (Theorem 1 above) holds, and the posterior π is the unique stationary distribution. To prove Conditions 1 and 2, we first recall the posterior and transition kernel for the modified

Gibbs sampler.

$$
\begin{aligned}
\pi(\theta^k) &= \pi(\alpha^k, \bar{w}^k) && (3.12)\\
&= p(\bar{w}^k \mid \alpha^k) \cdot p(\alpha^k) && (3.13)\\
&= \delta_{w*(\alpha)}(\bar{w}^k) \cdot p(\alpha^k) && (3.14)
\end{aligned}
$$

Similarly, the transition kernel is

$$
\begin{aligned}
P(\theta^k \mid \theta^{k-1}) &= P(\alpha^k, \bar{w}^k \mid \alpha^{k-1}, \bar{w}^{k-1}) && (3.15)\\
&= p_{\alpha|\bar{w}}(\alpha^k \mid \bar{w}^{k-1}) \cdot p_{\bar{w}|\alpha}(\bar{w}^k \mid \alpha^k) && (3.16)\\
&= p_{\alpha|\bar{w}}(\alpha^k \mid \bar{w}^{k-1}) \cdot \delta_{w*(\alpha)}(\bar{w}^k) && (3.17)
\end{aligned}
$$

Proof of Condition 1

We begin by proving that Condition 1 holds for the modified Gibbs sampler. Using the posterior and transition kernel from Equations (3.14) and (3.17), Condition 1 becomes

$$
\int\int \delta_{w*(\alpha)}(\bar{w}^k) \cdot p(\alpha^k)\, d\alpha^k\, d\bar{w}^k = 0 \quad \text{if and only if}
$$
$$
\int\int \delta_{w*(\alpha)}(\bar{w}^k) \cdot p(\alpha^k \mid \bar{w}^1)\, d\alpha^k\, d\bar{w}^k = 0 \quad \forall\theta^1. \tag{3.18}
$$

Then integrating with respect to $\bar{w}^k$ yields

$$
\int p(\alpha^k)\, d\alpha^k = 0 \quad \text{if and only if} \int p(\alpha^k \mid \bar{w}^1)\, d\alpha^k = 0, \quad \forall\theta^1. \tag{3.19}
$$

We then prove that (3.19) holds for the modified Gibbs sampler. Start by proving the right-hand side. That is, assume $\int p(\alpha^k)\, d\alpha^k = 0$. Then we need to show that

$\int p(\alpha^k \mid \bar{w}^1)\, d\alpha^k = 0, \quad \forall\theta^1$. For any θ^1 we have

$$\int p(\alpha^k \mid \bar{w}^1)\, d\alpha^k = \int \frac{p(\bar{w}^1 \mid \alpha^k) \cdot p(\alpha^k)}{p(\bar{w}^1)}\, d\alpha^k \tag{3.20}$$

$$= \frac{1}{p(\bar{w}^1)} \int \delta_{w*(\alpha)}(\bar{w}^1) \cdot p(\alpha^k)\, d\alpha^k. \tag{3.21}$$

If $\bar{w}^1 = w^*(\alpha^k)$, then $\delta_{w*(\alpha)}(\bar{w}^1) = 1$ and (3.21) becomes

$$\frac{1}{p(\bar{w}^1)} \underbrace{\int 1 \cdot p(\alpha^k)\, d\alpha^k}_{=0 \text{ by assumption}} = 0 \tag{3.22}$$

Otherwise, $\delta_{w*(\alpha)}(\bar{w}^1) = 0$ and (3.21) becomes

$$\frac{1}{p(\bar{w}^1)} \int 0 \cdot p(\alpha^k)\, d\alpha^k = 0. \tag{3.23}$$

Therefore, we have that $\int p(\alpha^k \mid \bar{w}^1)\, d\alpha^k = 0, \quad \forall\theta^1$.

Next, we show that the other direction holds. That is, assume that $\int p(\alpha^k \mid \bar{w}^1)\, d\alpha^k = 0, \quad \forall\theta^1$. Then we need to show that $\int p(\alpha^k)\, d\alpha^k = 0$. Starting with the left-hand side and using the total law of probability, we have

$$\int p(\alpha^k)\, d\alpha^k = \int \int p(\alpha^k \mid \bar{w}^1) p(\bar{w}^1) d\bar{w}^1\, d\alpha^k \tag{3.24}$$

$$= \int \int p(\alpha^k \mid \bar{w}^1)\, d\alpha^k p(\bar{w}^1) d\bar{w}^1 \tag{3.25}$$

$$= \int \underbrace{\int p(\alpha^k \mid \bar{w}^1)\, d\alpha^k}_{=0 \text{ by assumption}} p(\bar{w}^1) d\bar{w}^1 \tag{3.26}$$

$$= 0 \tag{3.27}$$

Therefore, we have that $\int p(\alpha^k)\, d\alpha^k = 0$. This further proves that Condition 1 holds for the modified Gibbs sampler. Thus, the modified Gibbs sampler generates an ergodic Markov chain that converges to its stationary distribution. The next section provides a proof that the posterior distribution is the unique stationary distribution of the modified Gibbs sampler.

Proof of Condition 2

Next, we prove that Condition 2 holds for the modified Gibbs sampler. We start with the left hand side of Condition 2.

$$\int \pi(\theta^{k-1})P(\theta^k \mid \theta^{k-1})\, d\theta^{k-1} \tag{3.28}$$

substituting P with the stationary distribution as in (3.17)

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \pi(\alpha^{k-1}, \bar{w}^{k-1}) \cdot p_{\alpha|\bar{w}}(\alpha^k \mid \bar{w}^{k-1}) p_{\bar{w}|\alpha}(\bar{w}^k \mid \alpha^k)\, d\alpha^{k-1}\, d\bar{w}^{k-1} \tag{3.29}$$

expanding the joint density using conditional probability

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{\bar{w}|\alpha}(\bar{w}^{k-1} \mid \alpha^{k-1}) \cdot p(\alpha^{k-1}) \cdot p_{\alpha|\bar{w}}(\alpha^k \mid \bar{w}^{k-1}) p_{\bar{w}|\alpha}(\bar{w}^k \mid \alpha^k)\, d\alpha^{k-1}\, d\bar{w}^{k-1} \tag{3.30}$$

Substituting $p_{\bar{w}|\alpha}(\bar{w}^{k-1} \mid \alpha^{k-1}) = \delta_{w^*(\alpha)}(\bar{w}^{k-1})$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta_{w^*(\alpha)}(\bar{w}^{k-1}) \cdot p(\alpha^{k-1}) p_{\alpha|\bar{w}}(\alpha^k \mid \bar{w}^{k-1}) \cdot \delta_{w^*(\alpha)}(\bar{w}^k)\, d\alpha^{k-1}\, d\bar{w}^{k-1} \tag{3.31}$$

Removing constants

$$= \delta_{w^*(\alpha)}(\bar{w}^k) \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta_{w^*(\alpha)}(\bar{w}^{k-1}) \cdot p(\alpha^{k-1}) p_{\alpha|\bar{w}}(\alpha^k \mid \bar{w}^{k-1})\, d\alpha^{k-1}\, d\bar{w}^{k-1} \tag{3.32}$$

Integrating with respect to $\bar{w}^{k-1}$

$$= \delta_{w^*(\alpha)}(\bar{w}^k) \int_{-\infty}^{\infty} p(\alpha^{k-1}) p_{\alpha|\bar{w}}(\alpha^k \mid w^*(\alpha^{k-1}))\, d\alpha^{k-1} \tag{3.33}$$

replace $p_{\alpha|\bar{w}}(\alpha^k \mid w^*(\alpha^{k-1}))$ with $p_\alpha(\alpha^k)$ because w^* is a point mass and $\bar{w} = w^*(\alpha)$ is guaranteed after integrating with respect to $\bar{w}^{k-1}$

$$= \delta_{w^*(\alpha)}(\bar{w}^k) \int_{-\infty}^{\infty} p(\alpha^{k-1}) p_\alpha(\alpha^k)\, d\alpha^{k-1} \tag{3.34}$$

$$= \delta_{w^*(\alpha)}(\bar{w}^k) \cdot p_\alpha(\alpha^k) \underbrace{\int_{-\infty}^{\infty} p(\alpha^{k-1})\, d\alpha^{k-1}}_{=1} \tag{3.35}$$

$$= \delta_{w^*(\alpha)}(\bar{w}^k) \cdot p(\alpha^k) \tag{3.36}$$

$$= \pi(\theta^k) \tag{3.37}$$

Therefore, we have that $\pi(\alpha^k, \bar{w}^k) = \delta_{w^*(\alpha)}(\bar{w}^k) \cdot p(\alpha^k)$ is a stationary distribution of the modified Gibbs sampler with transition kernel $P(\alpha^k, \bar{w}^k \mid \alpha^{k-1}, \bar{w}^{k-1}) = p_{\alpha|\bar{w}}(\alpha^k \mid \bar{w}^{k-1}) \cdot \delta_{w^*(\alpha)}(\bar{w}^k)$. Further, by the ergodic theorem, we know that π is the unique stationary distribution of the modified Gibbs sampler.

Thus, with ergodicity in hand, we know that the modified Gibbs sampler allows us to sample from the posterior distribution π. We use this concept in several experiments to learn the posterior distribution over the hyperparameter α of α-loss. The experiments performed are discussed in the following section.

Chapter 4

EXPERIMENTAL RESULTS

In applying the modified Gibbs sampler to datasets, we first examine several baseline experiments. We begin by performing baseline or “offline” experiments which entail splitting the range of α into a grid of evenly spaced α values. For each alpha, we then run gradient descent to obtain a set of “optimal” weights. From this we can visualize the expected posterior of α by directly calculating and plotting the full conditional distribution of α, $p(\alpha \mid \bar{w})$, for each value of α that was optimized in the previous step. It is common practice in Bayesian analysis to use the full conditional to approximate the marginal posterior (Casella and George, 1992). As we know that alpha-loss is robust to label noise for $\alpha > 1$ from Sypherd et al. (2021), we expect this trend to be reflected in the baseline figures as well.

Next, using the optimal weights obtained from the baseline experiments, we check the slice sampling over alpha, with $p(\alpha \mid \bar{w})$ as the target distribution, to ensure correctness of the implementation. This slice sampling is done over alpha and uses a “rounding” process in the posterior evaluations. That is, for a given posterior evaluation in slice sampling, we use the weights of the nearest value of alpha in the range of the offline experiments.

We next apply the alternating sampling scheme proposed as the modified Gibbs sampler in Chapter 3 to learn the distribution of α. All three aforementioned experiments are performed on a synthetic Gaussian Mixture Model dataset and the MNIST dataset of hand-written digits designed for classification (LeCun et al., 1998).

Finally, we examine the effect of varying the prior distribution of α on the Gaussian Mixture Model dataset by selecting a non uniform prior. This is to compare the effect of the previous experiments, which make use of a flat, uninformative prior. We begin describing the experimental results by focusing on the Gaussian Mixture Model dataset.

4.1 Gaussian Mixture Model

For this first dataset, we study the effects of the modified Gibbs sampler in classifying a two-dimensional Gaussian Mixture Model (GMM) with equal mixing probability $\mathbb{P}[Y = -1] = \mathbb{P}[Y = 1]$. The class -1 and $+1$ have means of $\mu_{X|Y=-1} = (-1, -1)^\top$ and $\mu_{X|Y=+1} = (1, 1)^\top$, respectively, and covariance matrix $\Sigma = \frac{1}{2}\mathbb{I}_2$. We explore the case in which the data suffers from label noise; that is, the labels of class $Y = -1$ are flipped with varying probability (10%, 20%, 30%, and 40%).

4.1.1 Baseline Experiment

For the baseline experiment, we train α-loss on the logistic model, which is a generalization of logistic regression that uses α-loss. Specifically, we use a linear classification function $f : \mathcal{X} \to \mathbb{R}$, where $f = \bar{w}^\top x$ and fit this output through a sigmoid, $g : \mathbb{R} \to [0, 1]$, to map the output of the classification function to a probability. That is,

$$g_{\bar{w}}(x) = \sigma(\bar{w}^\top x) = \frac{1}{1 + e^{-\bar{w}^\top x}}. \tag{4.1}$$

We note that $\bar{w} \in \mathbb{R}^3$ to account for the two dimensions of x and a bias or offset term. For the baseline experiment, we consider $\alpha \in [0.5, 4.0]$ in increments of 0.1. For each value of α in this range, we minimize α-loss using gradient descent with a learning rate scheduled by cosine annealing. We run the baseline experiments for five noise levels: 0%, 10%, 20%, 30%, and 40%. From the baseline experiments, we obtain a set of "optimal" weights corresponding to a value of α and a specific noise level. These weights allow us to visualize the full conditional distribution, $p(\alpha \mid \bar{w})$, from which we eventually sample α in the modified Gibbs sampler. For the Gaussian mixture model dataset, the baseline plot is presented in Figure 4.1.

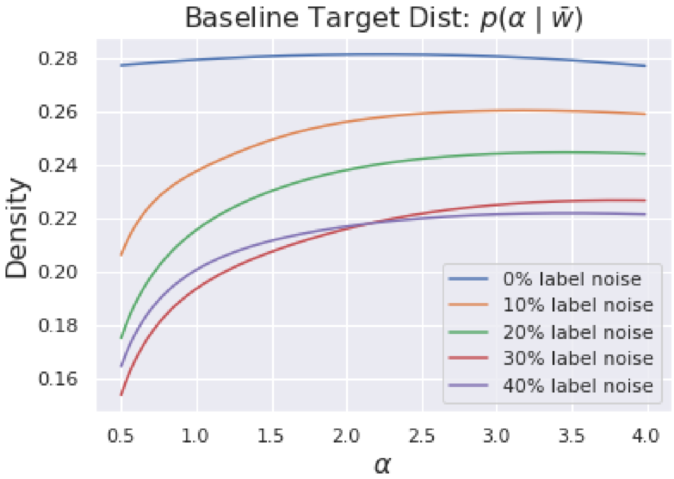

Figure 4.1: Baseline Plots of $p(\alpha \mid \bar{W})$ for the Gaussian Mixture Model Dataset. The Range of $\alpha \in [.5, 4]$ Is Split up into Intervals in Increments Of .1 and We Perform Gradient Descent over the Model Weights until Their Difference in Subsequent Iterations Is Less Than .001. This Provides a Visualization of $p(\alpha \mid \bar{W})$, Which We Later Use to Compare with the Marginal Posterior of α Obtained by the Modified Gibbs Sampler.

The baseline plot for the GMM dataset matches what we expect from previous experiments in (Sypherd et al., 2019). In the case of no label noise (0% label flip), the curve is flat, indicating that all values of α perform equally well. On the other hand, for data with label noise, we see that the density values are smaller for smaller values of α and increase with α, eventually flattening out toward $\alpha = 4$. Again, based on the work in (Sypherd et al., 2019), this matches our expectations as we see that once noise is introduced, higher values of α ($\alpha > 1$) are preferred.

4.1.2 Slice Sampling

Next, we perform slice sampling over α with $p(\alpha \mid \bar{w})$ as the target distribution, using the corresponding weights from the previous baseline experiments to evaluate $p(\alpha \mid \bar{w})$. That is, for each noise level, we perform slice sampling of the full conditional of alpha, $p(\alpha \mid \bar{w})$. Note that each step of slice sampling requires that the target distribution be evaluated. For this slice sampling implementation, we use the weights of the closest alpha present in the baseline calculations to evaluate the target distribution. For example, if the slice sampler required that the target distribution be evaluated for $\alpha = 3.21$, we would use the optimal weights from the baseline experiment for $\alpha = 3.2$ to evaluate the target distribution.

The results from the slice sampling are displayed in Figure 4.2. Again, we find that the slice sampling over α generates a density similar to the baseline plots. We use this as confirmation of correct implementation of the slice sampling algorithm.

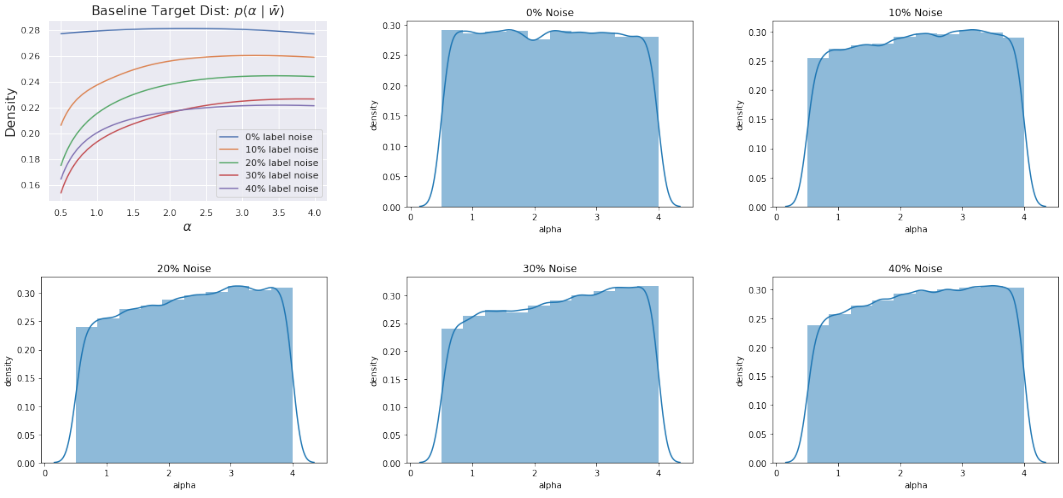

Figure 4.2: Slice Sampling over the Baseline Curves for the GMM Dataset. We Perform Slice Sampling over α, Using $p(\alpha \mid \bar{W})$ as the Target Distribution to Confirm That the Implementation of the Slice Sampling Algorithm Is Correct.

4.1.3 Modified Gibbs Sampler

Finally, we apply the modified Gibbs sampler to the Gaussian Mixture Model dataset. Specifically, we alternate in sampling α using slice sampling and optimizing the model weights $\bar{w}$ using one epoch of gradient descent. The alternating sampling scheme is run for 10,000 iterations, after which we are left with a chain of 10,000 samples of α. As is common in MCMC methods, we discard the first group of samples and treat it as the *burn-in* period, as the beginning of the chain is thought not to have reached stationarity (Roberts and Rosenthal, 2003). Therefore, for the GMM experiment, we take the burn-in period to be the first 1,000 samples. Similarly, practitioners can apply the practice of *thinning* in order to decrease the correlation between subsequent samples that is inherent in the Markov chain. Thinning takes every k samples from the chain and discards the remaining samples. The idea behind thinning is that it makes observations more independent and thus more like a random sample from the posterior distribution (Christensen et al., 2010). For the GMM experiment, we thin the α chain by 2, meaning we discard every other sample to decrease autocorrelation.

Finally, with our post-processed chain, we evaluate the marginal posterior density of α by generating a kernel density estimate of the remaining samples. Further, twenty chains are run (in blue) to analyze the average chain behavior (in red).

The obtained posterior plots of α are shown in Figure 4.3. For 0% label noise, we see that the density is approximately uniform; that is, no preference is shown for a particular value of alpha. However, once label noise is introduced, the density values are higher for larger values of α. This result agrees with previous experiments in using the logistic model and alpha-loss to classify binary Gaussian Mixture models (Sypherd et al., 2019).

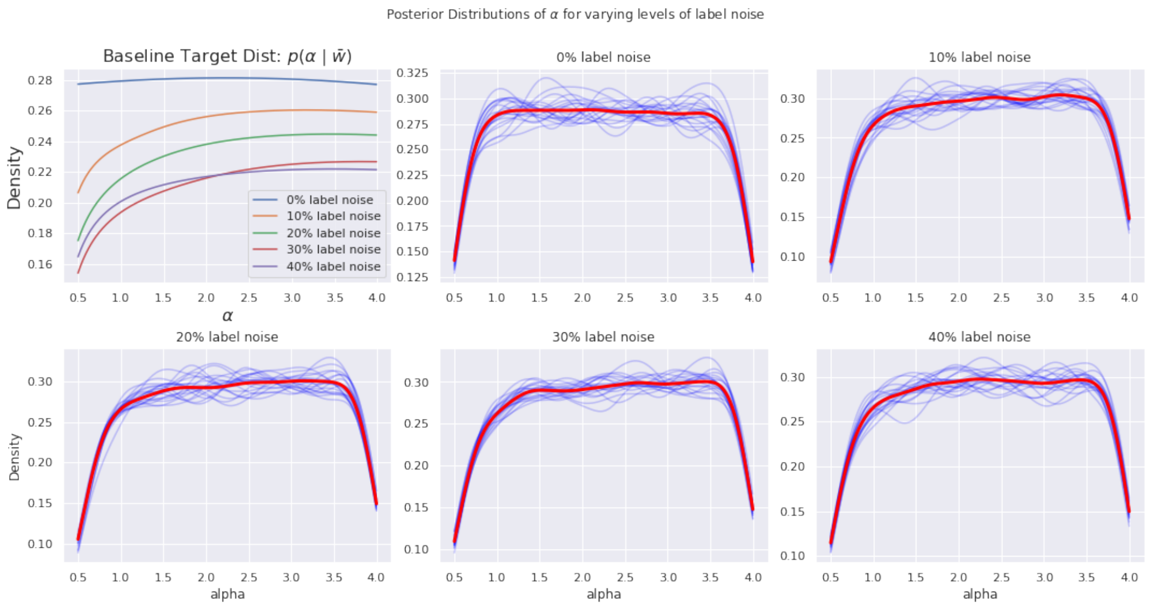

Figure 4.3: The Marginal Posterior Distributions of α for Varying Levels of Label Noise Obtained by the Modified Gibbs Sampler for the GMM Dataset. The Blue Curves Are Individual Chains Obtained over Twenty Different Runs, While the Red Curve Shows the Average Density Values of the Twenty Runs.

4.2 MNIST Dataset

The second set of experiments is performed using the MNIST dataset, which is a collection of images of hand-written digits ranging from 0 through 9 (LeCun et al., 1998). For our experiment, we extract the numbers 1 and 7 only to perform a binary classification. Additionally, the classification algorithm is trained on a binary labeled dataset that suffers from symmetric noisy labels as in (Sypherd et al., 2019). We explore data with four levels of label noise: 0%, 10%, 20%, and 30%.

All code is written in Python and relies on the machine learning framework, Pytorch (Paszke et al., 2019). We consider a convolutional-neural-network (CNN) with two fully connected layers preceded by two convolutional layers. The complete architecture is displayed in Figure 4.4. Again, as in (Sypherd et al., 2019), we use softmax activation to generate probabilities of the labels, after which the model's belief is evaluated using α-loss.

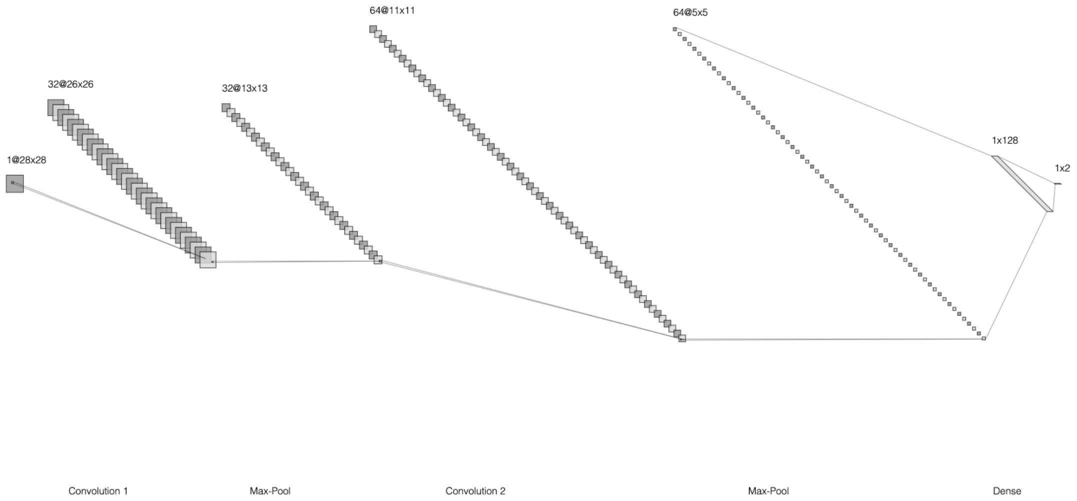

Figure 4.4: Architecture of the 2-layer CNN Used on the MNIST Dataset. The First CNN Layer Has 32 Layers with No Padding and Max-pooling. The Second CNN Layer Has 64 Layers with No Padding and Max-pooling. These Layers Are Followed by Two Fully Connected Layers with 128 and 2 Outputs, Respectively.

4.2.1 Baseline Experiments

For the baseline experiments, we split the range of $\alpha \in [0.5, 10]$ into a grid of values in increments of 0.5 (i.e. 20 values total). Then for each value of α, a model is trained using gradient descent with a learning rate scheduled by cosine annealing to prevent getting stuck in a sub-optimal location. The gradient descent is run for 50 epochs to determine the "optimal" weights for that value of α. We choose 50 epochs as that is the number of epochs used for training in (Sypherd et al., 2019). This experiment is run for all four noise levels. Similar to the GMM baseline experiments, the optimal weights allow us to calculate and visualize the full conditional density of

α, $p(\alpha \mid \bar{w})$. The baseline figure is displayed in Figure 4.5.

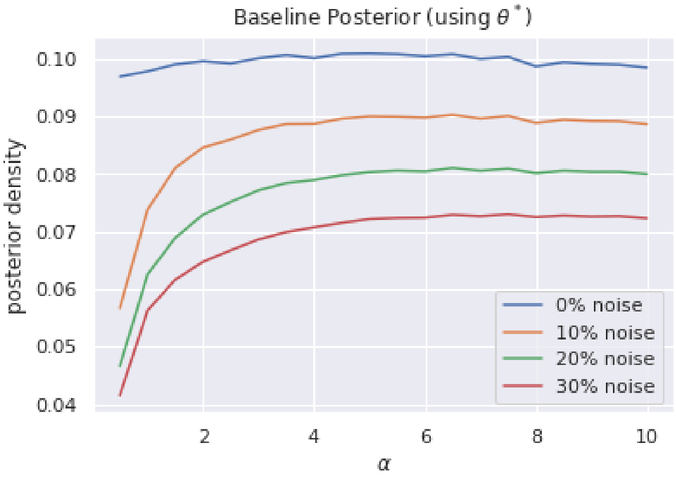

Figure 4.5: Baseline Plots of $p(\alpha \mid \bar{W})$ for the MNIST Dataset. The Range of $\alpha \in [.5, 10]$ Is Split up into Intervals in Increments Of .5 and We Perform 50 Epochs of Gradient Descent over the Model Weights. This Provides a Visualization of $p(\alpha \mid \bar{W})$, Which We Later Use to Compare with the Marginal Posterior of α Obtained by the Modified Gibbs Sampler.

Again we notice from Figure 4.5 that for 0% label noise, the curve is relatively uniform and indicates equal performance over the range of $\alpha \in [0.5, 10]$. However, once label noise is introduced, there is a preference for larger values of α, which eventually levels out indicating a "saturation" effect. That is, sufficiently large α tend to have similar performance as presented in (Sypherd et al., 2019).

4.2.2 Slice Sampling

Next, as with the GMM dataset, we perform slice sampling over α using the corresponding weights from the previous baseline experiments. That is, for each noise level, we perform slice sampling with the full conditional of α, $p(\alpha \mid \bar{w})$, as the target distribution. Again, as with the GMM dataset, we use a "rounding" process to obtain the model weights for the closest α value in each target density evaluation. The results of the slice sampling are shown in Figure 4.6.

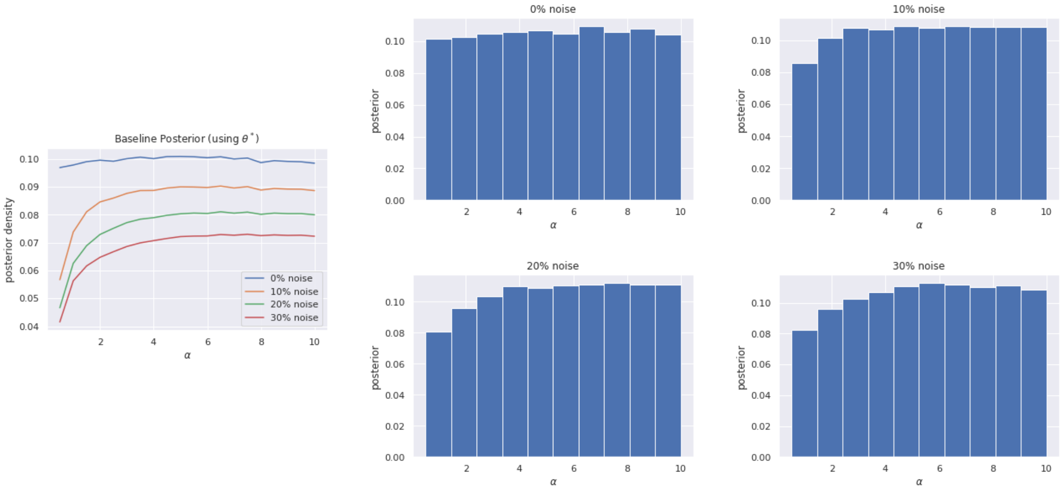

Figure 4.6: Slice Sampling over the Baseline Curves for the MNIST Dataset. We Perform Slice Sampling over α, Using $p(\alpha \mid \bar{W})$ as the Target Distribution to Confirm That the Implementation of the Slice Sampling Algorithm Is Correct.

The results in Figure 4.6 show agreement with the baseline plots. That is, the slice sampling plots for 0% noise are uniform, while the cases with label noise have a density that increases with α. We use this as confirmation of correct implementation of the slice sampling algorithm.

4.2.3 Modified Gibbs Sampler

Finally, the modified Gibbs sampler is applied to the MNIST dataset for varying levels of label noise. To expedite the burn-in process, we perform additional gradient descent steps in the beginning of the sampling process. The exact sampling procedure is

- Iteration 0-200 → sample α every 10 epochs of gradient descent
- Iteration 200-400 → sample α every 5 epochs of gradient descent
- Iteration 400-600 → sample α every 4 epochs of gradient descent
- Iteration 600-800 → sample α every 3 epochs of gradient descent

- Iteration 800-1000 → sample α every 2 epochs of gradient descent
- Iteration 1000+ → sample α every epoch of gradient descent

This sampling procedure allows for added stability in the loss landscape. That is, by running more epochs of gradient descent in the early iterations, we are closer to the optimum, instead of simply switching landscapes with each iteration. Essentially, this allows us to reach the optimum faster and adds stability to the algorithm.

This alternating sampling algorithm is run until 10,000 samples of α are attained. Using these samples, a similar post-processing procedure is performed. We use a burn-in period of 1,000 samples and thin the chain using $k = 2$. That is, we keep every other sample and discarding the remaining samples. The samples that remain after discarding the burn-in period and thinning are then used to generate a kernel density estimate plot to approximate the marginal posterior of α in Figure 4.7.

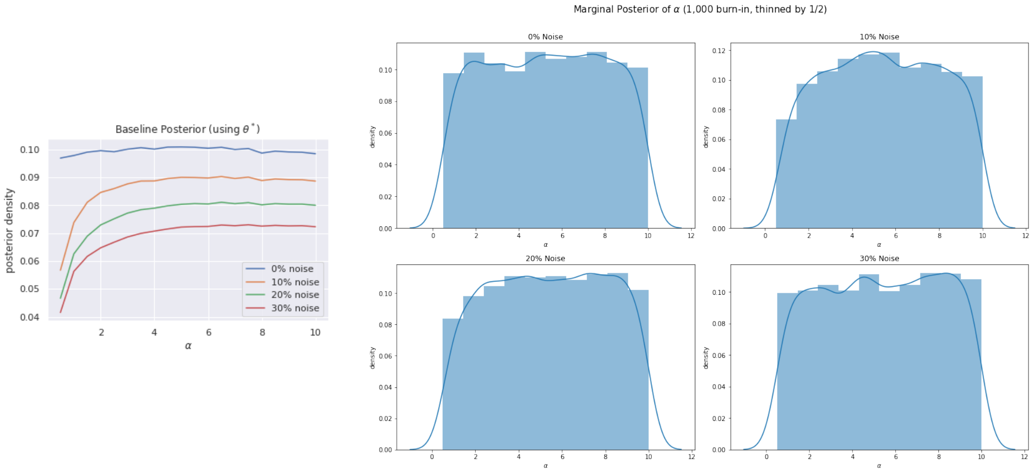

Figure 4.7: RHS: The Marginal Posterior Distributions of α for Varying Levels of Label Noise Obtained by the Modified Gibbs Sampler for the MNIST Dataset. LHS: We Provide the Baseline Figure for Comparison.

The results in Figure 4.7 are encouraging because they tend to agree with our expectations. In particular, the posterior for 0% noise tends to be flat or uniform across the range of alpha. As for the data with label noise, we see a preference for larger values of α, as seen in (Sypherd et al., 2019).

Additionally, we examine several features to ensure the sampling chain has reached convergence. First, we examine the trace plot of α in Figure 4.8, which shows the values of α that were sampled with each iteration of the modified Gibbs sampler. The trace plots of α show good mixing; that is, the chain thoroughly explores the parameter space without any trends or cycles.

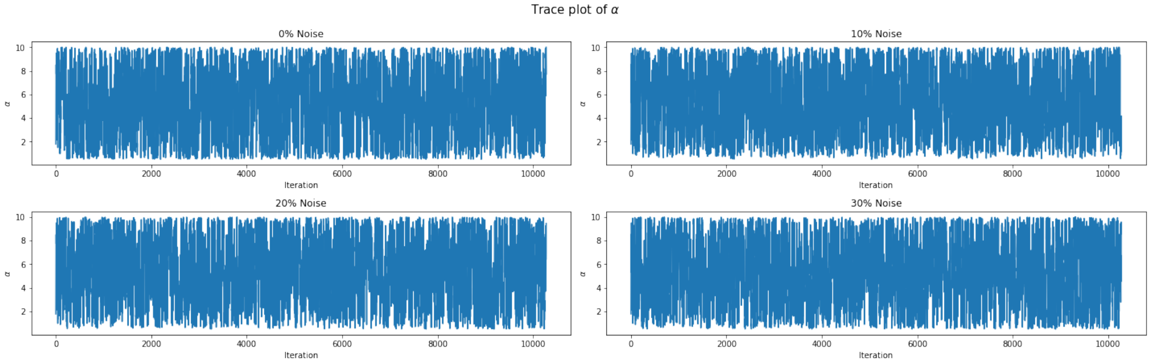

Figure 4.8: Trace Plot of α Chain Obtained by the Modified Gibbs Sampler on the MNIST Dataset.

A metric that can be used to assess convergence of the chain to the stationary distribution is the Geweke diagnostic. This statistic compares the first portion of the chain (usually the first 10%) to the last 50% of the chain with the idea that, if the chain has converged to the target distribution, then the mean of the early portion should not be significantly different from the latter half of the chain (Geweke, 1992). Practically speaking, the Geweke diagnostic takes the first 10% of the chain as the initial portion, θ_s, and splits the final 50% of the chain into twenty segments $\theta_{e1}, \ldots, \theta_{e20}$, and performs a z-test for each segment. The hypothesis is

$$H_0 : \theta_s = \theta_{e_i} \tag{4.2}$$

$$H_1 : \theta_s \neq \theta_{e_i}. \tag{4.3}$$

Thus, if we fail to reject the null hypothesis, we have strong evidence that the means are equivalent, further implying that the chain has reached convergence to the target distribution. We run the Geweke diagnostic on the α chains for each noise level, comparing the mean of the first 10% of each chain to twenty segments of the latter half of the chain. The results of the test are shown in Figure 4.9. We see that the test statistics all lie within two standard deviations, indicating that the first 10% (first 1,000 samples) can be treated as the burn-in period. This result also implies that our chain has reached the target stationary distribution.

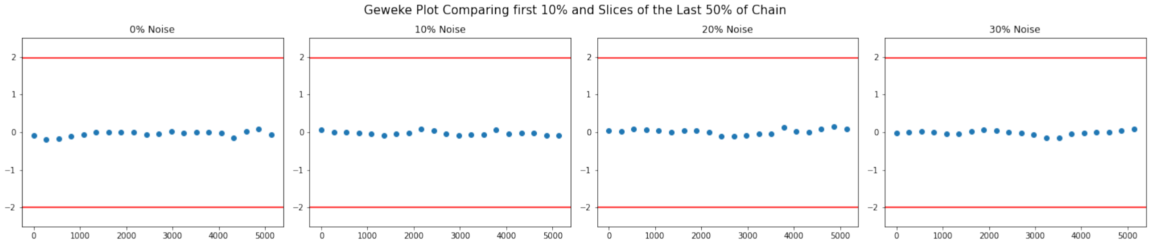

Figure 4.9: Plot of the Geweke Diagnostic Used to Determine Convergence of the α Chain to the Stationary Distribution for the MNIST Dataset.

Thus, according to the Geweke diagnostic, our Markov chain agrees with the theoretical properties of Chapter 3 in that the modified Gibbs sampler converges to the target distribution. Additionally, the marginal posterior distributions learned in the MNIST experiment also agrees with the theoretical results presented in (Sypherd et al., 2021). Figure 4.7 shows that the larger values of α (i.e. $\alpha > 1$) are preferred over smaller values of α for data with label noise. We note that, as the amount of noise increases, the preference for the larger values of alpha also increases. For example, in Figure 4.7, we see that $\alpha > 6$ and larger have a higher density for the datasets with 20% and 30% noise, as compared to the density for 10% noise. This result suggests that the modified Gibbs sampler is effective in learning the distribution of α in tuning α-loss.

4.3 Non-Uniform Priors

Lastly, we examine the use of non-uniform prior over α in the modified Gibbs sampler for the Gaussian Mixture Model dataset. Similar to the experiment in Section 4.1.3, we run the modified Gibbs sampler for 20,000 iterations, but use three non-uniform priors: $\mathcal{N}(\mu_\alpha = 1.0, \sigma_\alpha^2 = 0.01)$, $\mathcal{N}(\mu_\alpha = 1.0, \sigma_\alpha^2 = 0.1)$, and $\mathcal{N}(\mu_\alpha = 1.0, \sigma_\alpha^2 = 1.0)$. Figure 4.10 then shows a kernel density estimate of the obtained samples of α for varying noise levels, as well as the original prior that was used.

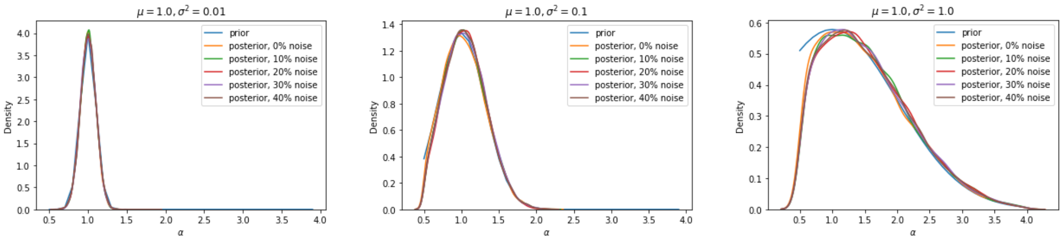

Figure 4.10: Posterior of α Learned Using Three Different Prior Distributions in the Modified Gibbs Sampler for the GMM Dataset. The Priors Used, from Left to Right: $\mathcal{N}(\mu_\alpha = 1.0, \sigma_\alpha^2 = 0.01)$, $\mathcal{N}(\mu_\alpha = 1.0, \sigma_\alpha^2 = 0.1)$, and $\mathcal{N}(\mu_\alpha = 1.0, \sigma_\alpha^2 = 1.0)$

From Figure 4.10, we see that the learned posterior of α tends to coincide with the chosen prior distribution (for all three choices of the prior). In other words, the posterior is dominated by the prior. This indicates that the previously chosen likelihood of $exp(-l^\alpha)$ should be normalized in a way that allows the loss values, and consequently the likelihood, to be compared directly. This particular normalization is established in Jiang and Tanner (2008) and Zhang (2006), which introduce methods of using loss functions as a quasi-likelihood. This stands as a direction for future work.

Chapter 5

CONCLUSION & FUTURE WORK

With the increased need for robustness in machine learning, the ability to properly tune parameterized loss functions has presented itself as a unique challenge. While traditional methods generate point estimates, MCMC algorithms can be leveraged to learn distributions of the hyperparameters in question. This thesis shows the effectiveness of a modified Gibbs sampling method to learn the distribution of the parameter α of α-loss. The experimental results are also fortified by the theoretical proof of ergodicity and stationarity of the modified Gibbs sampler. Given these positive results, one clear area of improvement is the computational resources required for the modified Gibbs sampler. While the baseline experiments need not be completed to implement the modified Gibbs sampler, the algorithm itself is still quite time intensive; however, if resources are not scarce, the practitioner may determine that obtaining a distribution of the hyperparameter is worth the resources.

That said, there are several areas of future work. Preliminary experiments on changing the prior over α indicate that the posterior is heavily dependent on the chosen prior. As we do not want this to be the case, there remains the need to normalize the likelihood. That is, more work is needed on the information theoretic framework to understand how the $exp(-l^{\alpha})$ should be normalized to be a proper likelihood as in Jiang and Tanner (2008) and Zhang (2006).

Specific to α-loss, the experiments performed in this thesis mostly used a relatively flat, uninformative prior. Sometimes it may be the case that the practitioner has prior information about their dataset. This could allow for the use of more informative priors, which could improve the results of the modified Gibbs sampler. Further related to α-loss, it is well documented that $\alpha < 1$ can be leveraged in the case of class imbalance (Sypherd et al., 2019). The application of the modified Gibbs sampler to datasets with class imbalance is an interesting opportunity to explore whether the distribution agrees with previous results.

There are also other Bayesian methods that do not require the use of MCMC algorithms. For example, Bayesian optimization is a two-step optimization procedure that first uses Gaussian process regression to build a surrogate function of the objective and second, uses an evaluation metric, or acquisition function, to determine future sampling areas that hold the most potential for improvement (Frazier, 2018). Bayesian optimization is also leveraged in the hybrid procedure called Bayesian optimization Hyperband method (BOHB). Hyperband uses the process of successive halving - starting with n parameter configurations and discarding the worst performing half after a certain amount of time - for a number of randomly sampled configurations (Li et al., 2017). Hyperband works to balance exploration and exploitation by trying out many parameter configurations while giving more resources to the promising configurations. Thus, the hybrid method, BOHB, combines the speed of evaluations due to Hyperband and refinement of selections due to the guided search capability of Bayesian optimization. Consequently, Bayesian optimization and BOHB are two additional methods with promise to solve the problem of tuning hyperparameters of loss functions.

REFERENCES

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to mcmc for machine learning. *Machine learning*, 50(1):5–43.

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., et al. (2021). Hyperparameter optimization: Foundations, algorithms, best practices and open challenges. *arXiv preprint arXiv:2107.05847*.

Casella, G. and George, E. I. (1992). Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174.

Christensen, R., Johnson, W., Branscum, A., and Hanson, T. E. (2010). *Bayesian ideas and data analysis: an introduction for scientists and statisticians*. CRC press.

Frazier, P. I. (2018). A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to the calculations of posterior moments. *Bayesian statistics*, 4:641–649.

Jiang, W. and Tanner, M. A. (2008). Gibbs posterior for variable selection in high-dimensional classification and data mining. *The Annals of Statistics*, 36(5):2207–2231.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86(11)*, pages 2278–2324.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816.

Mei, S., Bai, Y., and Montanari, A. (2018). The landscape of empirical risk for nonconvex losses. *The Annals of Statistics*, 46(6A):2747–2774.

Neal, R. M. (2003). Slice sampling. *The annals of statistics*, 31(3):705–767.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Roberts, G. O. and Rosenthal, J. S. (2003). Markov chain monte carlo. *Encyclopedia of the Actuarial Sciences*.

Sypherd, T., Diaz, M., Cava, J. K., Dasarathy, G., Kairouz, P., and Sankar, L. (2019). A tunable loss function for robust classification: Calibration, landscape, and generalization. *arXiv preprint arXiv:1906.02314*.

Sypherd, T., Nock, R., and Sankar, L. (2021). Being properly improper. *arXiv e-prints*, pages arXiv–2106.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728.

Weerts, H. J., Mueller, A. C., and Vanschoren, J. (2020). Importance of tuning hyperparameters of machine learning algorithms. *arXiv preprint arXiv:2007.07588*.

Zhang, T. (2006). Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321.