Abstract

Computation and Estimation for Neural Networks via Log-Concave Coupling

Curtis James McDonald

2025

In this work, we consider a Bayesian method to train single-hidden-layer neural networks with ℓ_1 controlled weights by defining posterior distributions using different subsets of the training data, and combining posterior means to form our estimators. We consider both a joint Bayesian model for all parameters of the neural network at once, and a greedy Bayes model training the neurons one at a time based on the residuals of previous fits.

The log-likelihoods of the posterior distributions we define are multimodal and non-concave, so sampling algorithms such as Markov Chain Monte Carlo (MCMC) will not be rapidly mixing to directly sample the posteriors. Using an auxiliary random variable, we produce a mixture distribution which we call a log-concave coupling. Using a continuous uniform prior over the ℓ_1 ball, the conditional distributions of this mixture are log-concave, and the mixing distribution itself is log-concave when the number of parameters in our neural network exceeds the squared number of data points. Thus the mixture distribution can be sampled efficiently to produce samples for our original target density.

For a discrete uniform prior over the ℓ_1 ball intersected with a grid of small spacing, we study the performance of our posterior mean estimator in an arbitrary regret sense and a statistical risk sense. Say we have a target function g, with $\tilde{g}$ being its projection into the closure of the convex hull of signed neurons scaled by a constant. With neuron weight vectors of dimension d and N data points, we show an estimator defined by a combination of our posterior means in the joint sampling problem has arbitrary sequence regret and statistical risk within $O([(\log d)/N]^{1/4})$ of the regret and risk of $\tilde{g}$. For the greedy construction, the additional regret and risk is an improved third root power.

Computation and Estimation for Neural Networks via Log-Concave Coupling

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Curtis James McDonald

Dissertation Director: Andrew R Barron

May 2025

To my parents.

Contents

List of Figures

List of Tables

Acknowledgments

The road to complete my Ph.D. was a journey full of ups and downs, trials and victories. Over my 6 years at Yale I grew very much as a scholar and as a person, and I would like to thank those who have supported me both personally and academically.

To my Ph.D. adviser Andrew Barron, who has been a wealth of knowledge and experience I have always been grateful to benefit from. Andrew never shied away from investing time and effort into our research together. We met for many hours over the course of my studies and they were always full with unique and creative ideas. I consider our work together a true collaboration with significant contribution from both Andrew and myself, as we continue to pursue knowledge and understanding for its own sake.

To my undergraduate and master's adviser Serdar Yüksel, who first helped me find my interest in academia and taught me to be a scholar. His guidance placed me on my path today, and I consider Serdar a mentor and a friend.

To all my classmates in the Statistics and Data Science department, especially Alex, Vijay, and Colleen. SDS was a place to learn, grow, and a community which supported each other. From board game nights to study sessions, it was never a path to walk alone.

To the St. Thomas More community, which was my spiritual home in New Haven.

To my sister Alexandra McDonald, the first Dr. McDonald but not the last. From a young age we encouraged and challenged each other to go deeper and succeed in academics. Though we chose different careers later in life, we owe a lot to each other for becoming who we are today, and share a bond to last a lifetime.

To my parents Doug and Nancy McDonald, whose love and support first gave me a base to grow. My parents allowed me to find my passion in life, and encouraged me to pursue it wherever it may take me. As I have traveled to pursue my dreams, I could rest safe knowing their help was never far away.

Thank you all.

Chapter 1

Introduction

In this work, we study a Bayesian method for training neural networks. We study conditions under which Markov Chain Monte Carlo (MCMC) will be rapidly mixing for our neural network posterior distributions. We show that MCMC will mix rapidly by a method we call a **Log-Concave Coupling**. That is, given a target probability density p for a random variable w, a log-concave coupling is a joint density with an auxiliary random variable ξ with support set B, denoted $q(w, \xi)$, such that

1. The marginal density of w is maintained, $p(w) = \int_B q(w, \xi) d\xi$.
2. The marginal density of ξ is log-concave, $q(\xi) = \int q(w, \xi) dw$.
3. The reverse conditional density $q(w|\xi)$ is log-concave for any conditional ξ in the support set B.

MCMC algorithms are often shown to be rapidly mixing for log-concave target densities, therefore a sample of w from the target posterior density p can be generated by first sampling ξ from its marginal density, and then sampling w from its conditional density given a ξ value.

Our Bayesian methods are considered in two parts: a joint sampling algorithm producing a distribution on all parameters of the neural network at once, and a greedy Bayes

procedure which samples one neuron at a time based on the residuals of the previously trained network. We demonstrate a log-concave coupling both in the joint sampling case, and in the greedy case.

We also study the statistical risk and regret of estimators based on linear combinations of posterior means. Our risk control and regret results are based on the method of the **Index of Resolvability**. This requires that:

1. There exists at least one neural network that could fit our target function well.

2. Our prior places sufficiently large probability on a set of good estimation weights.

Given N data points of dimension d, our statistical risk results show the joint sampling problem has a risk and regret control of $O([(\log d)/N]^{\frac{1}{4}})$, while the greedy Bayes procedure has a risk and regret control of $O([(\log d)/N]^{\frac{1}{3}})$. We now review the motivation for the problem, and define the specifics of our Bayesian model.

1.1 Approximation, Estimation, and Computation

One of the core problems in Statistical Learning Theory is to produce function estimators by combining elements of a library of basis functions. Define a library $\mathcal{H}$ as a (possibly uncountable) collection of functions from an input space $\mathcal{X}$ to an output space $\mathcal{Y}$. For us, it will suffice to consider $\mathcal{X} = \mathbb{R}^d$ for some d and $\mathcal{Y} = \mathbb{R}$. We then consider the problem of constructing linear combinations of elements of $\mathcal{H}$. For some number of elements K, make a selection of elements of the library $(h_k)_{k=1}^K, h_k \in \mathcal{H}$, and a selection of external weights $(c_k)_{k=1}^K, c_k \in \mathbb{R}$. We then construct a function f as a linear combination of elements of the

library

$$f : \mathbb{R}^d \to \mathbb{R} \tag{1.1}$$

$$x \mapsto \sum_{k=1}^{K} c_k h_k(x). \tag{1.2}$$

The purpose of creating the function f is to fit some observed data sequence $(x_i, y_i)_{i=1}^N$ with $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$ for each i. For a loss function $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and some competitor function g, we may want to find a function f that has small regret, which we define as

$$\sum_{n=1}^{N} L(y_n, f(x_n)) - L(y_n, g(x_n)). \tag{1.3}$$

We may allow ourselves to pick a different function $\hat{f}_{n-1}(\cdot|x^{n-1}, y^{n-1})$ at each index n which is a function of the data $(x_i, y_i)_{i=1}^{n-1}$ up to that point. Then regret can be considered for the online learning problem, where data up to index $n-1$ is used to fit a function for index n,

$$\sum_{n=1}^{N} L(y_n, \hat{f}_{n-1}(x_n|x^{n-1}, y^{n-1})) - L(y_n, g(x_n)). \tag{1.4}$$

Additionally, consider if the $(X_i, Y_i)_{i=1}^N$ are random variables that come independently and identically distributed (iid) from a data distribution $P_{X,Y}$ with $E_{P_{Y|X}}[Y|X] = g(X)$ for some function g. Then our goal may be to find an f function with low statistical risk,

$$E_{P_X}[L(g(X), f(X))]. \tag{1.5}$$

If we consider $X^N, Y^N = (X_i, Y_i)_{i=1}^N$ as our training data, and X_{N+1} as some new data point, consider if $\hat{f}(\cdot|X^N, Y^N)$ is some predictor function based on the training data. Then

we can consider the risk as an expectation over both training data, and a new data point

$$E_{P_{X^N,Y^N}}[E_{P_X}[L(g(X), \hat{f}(X|X^N, Y^N))]] \tag{1.6}$$

The question then is what method do we use to map training instances X^N, Y^N to functional approximators $\hat{f}$ for future data. This selection method is referred to as an algorithm which intakes training data instances and after some computation, outputs a functional estimator $\hat{f}$ which is a K linear combination of elements of our library $\mathcal{H}$.

In studying algorithms for data fitting, there are three primary considerations: **approximation**, **estimation**, and **computation**.

Approximation asks the question, given a library $\mathcal{H}$ and a target function g, what bounds can be given for the ideal performance using the best K elements of the library

$$\inf_{h_k \in \mathcal{H}, c_k \in \mathbb{R}} E_{P_X}[L(g(X), \sum_{k=1}^{K} c_k h_k(X))]. \tag{1.7}$$

If a library has poor approximation ability, then any algorithm for selecting elements of that library will have poor risk. For example, linear models are poorly fit to non-linear functions, so the application of linear models as approximators is limited. However, approximation tells us if a good estimator of g exists in my class, but does not indicate an algorithm to select a good estimator based on data.

Estimation asks the question, given an algorithm that maps training instances $(X_i, Y_i)_{i=1}^N$ to function estimators $\hat{f}(\cdot|X^N, Y^N)$, what bounds can be given for the risk of these estimators

$$E_{P_{X^N,Y^N}}[E_{P_X}[L(g(X), \hat{f}(X|X^N, Y^N))]]. \tag{1.8}$$

Simply because a good approximation for g exists, does not guarantee an algorithm will

be able to produce a realized function $\hat{f}$ with similar risk.

Computation asks the question, how many tasks does my algorithm have to complete as a function of the number of data points N, and dimension d. We ideally would like an algorithm that has (low order) polynomial complexity in N and d, as algorithms with exponential complexity in N and d are not practically useful as the values grow large.

Thus, approximation, estimation, and computation are three legs of the statistical learning stool, and if any one of them fails an algorithm is severely limited in its application. However, it is quite difficult to find algorithms that satisfy all three considerations at once. Algorithms with good estimation and computation but poor approximation can find a K term combination of library elements with risk near the best possible combination, but this risk is still poor. Algorithms with good approximation and computation but lacking estimation quality have some function in the class that could fit the data and the algorithm runs quickly, but produces fits $\hat{f}$ that are nothing like the optimal combination and have poor risk. Algorithms with good approximation and estimation but poor computation could produce fits with near optimal risk, but the time required to compute this function would be too long to be practical.

1.2 Algorithms for Neural Networks

Approximation is perhaps the first consideration among the three, as without good approximation even the best algorithm for selection can only perform so well. Single-hidden-layer artificial neural networks provide a flexible class of parameterized functions for data fitting applications, and this will be the function library $\mathcal{H}$ we study in this work.

Specifically, denote a single-hidden-layer neural network as the parameterized function

$$f_w(x) = f(w, x) = \sum_{k=1}^{K} c_k \psi(w_k \cdot x), \tag{1.9}$$

with K neurons, activation function ψ, and interior weights $w_k \in \mathbb{R}^d$. Fix a positive scaling V and let the exterior weights c_k be positive or negative values $c_k \in \{-\frac{V}{K}, \frac{V}{K}\}$. Thus, $f_w(x)$ is a convex combination of K signed neurons scaled by V. Constant and linear terms $c_0 + w_0 \cdot x$ may be added in the definition of $f_w(x)$ to achieve additional flexibility, though we will not address that matter explicitly.

We are interested in potentially wide networks where K may be large. The study of deep nets (i.e. multi-layered) nets is a separate topic not addressed in this work, as we focus on the single-hidden-layer class.

The approximation ability of these networks has been studied for many years, showing which functions can be well approximated by some linear combination of neurons. It is shown in [19] that any continuous function on a compact subset of $\mathbb{R}^d$ and in [29] that any measurable function of finite $L_2(P_X)$ norm can be well approximated by linear combinations of sigmoid functions. These results however do not give explicit statement of how the error relates to the specific width K of the network. This direction is taken up in [7], tying rates of decay in the risk to the width of the network K and Fourier conditions on the target function. These results initially allow for arbitrary sized internal neurons weights, and arbitrary outer weights. In our work here, we control the interior weights to have $\|w_k\|_1 \leq 1$ and $|c_k| \leq \frac{V}{K}$. Nonetheless, in the original approximation results for arbitrary sized neurons weights, consider if a target function f has complex valued Fourier components $\tilde{f}(\omega)$,

$$f(x) = \int_{R^d} e^{i\omega \cdot x} \tilde{f}(\omega) d\omega. \tag{1.10}$$

Define C_f as the integral of the ℓ_1 norm of ω times the magnitudes of the Fourier components,

$$C_f = \int_{\mathbb{R}^d} \|\omega\|_1 |\tilde{f}(\omega)| d\omega. \tag{1.11}$$

Denote Γ_C as the class of functions with $C_f \leq C$. For any density μ over an ℓ_∞ ball of radius $B_r = \{x \in \mathbb{R}^d : \|x\|_\infty \leq r\}$, any function $f \in \Gamma_C$ can be approximated by a single-hidden-layer network f_K with K neurons such that

$$\int_{B_r} (f(x) - f_K(x))^2 \mu(dx) \leq \frac{(2rC_f)^2}{K}. \tag{1.12}$$

So functions with a finite C_f can be approximated by some linear combinations of neurons to arbitrary small error with a sufficiently wide network.

If one specifically works with the ReLU activation function, these results also apply when one forces bounded neuron weight vectors, and not just unbounded. However, when working with general bounded increasing activation functions as in [7], the neuron weights w_k must be allowed to have unbounded components.

These original results put no restrictions on how large the components of the internal weight vectors w_k can be. To facilitate computation and estimation quality, we wish to work only with weight vectors w_k with bounded ℓ_1 norm, $\|w_k\|_1 \leq 1$ [4, 36]. Denote the set of signed neurons with ℓ_1 controlled interior weights as the collection of functions $h : [-1, 1]^d \to \mathbb{R}$

$$\Psi = \{h : h(x) = \pm\psi(w \cdot x), \|w\|_1 \leq 1\}. \tag{1.13}$$

The closed convex hull of Ψ includes functions f which can be written as a possibly infinite mixture of signed neurons, and functions which are the limit of a sequence of such mixtures. Specializing the results of [6],[7],[36], networks of the form (1.9) provide accurate approximation for functions f with $\frac{f}{V}$ in the closure of the convex hull of Ψ. The infimum of such V is called the variation V_f of the function f with respect to the dictionary Ψ. In [36] a variant of the condition on the Fourier components of f is also given that would allow f to have finite variation V_f and hence to be accurately approximated using

convex combinations of elements of Ψ scaled by the variation, with bounded ℓ_1 norm on the weights. For target functions f of this form and any probability distribution P_X on $[-1,1]^d$, using a squared ReLU activation function, there exists a network f_{w^*} of the form (1.9) with added constant and linear terms, with K neurons with ℓ_1 controlled internal weights, such that [36]

$$\|f_{w^*} - f\|^2 \leq \frac{V_f^2}{K}, \tag{1.14}$$

where $\|\cdot\|^2$ is the $L_2(P_X)$ norm.

The approximation with bound (1.14) is an existence result, a useful ingredient in neural net analysis. Yet, by itself, it does not imply anything about the estimation ability of training algorithms based on a finite set of N data points $(x_i, y_i)_{i=1}^N$ independently and identically distributed (iid) from a data distribution $P_{X,Y}$. Currently, the best known results show that for a bounded target function $|f| \leq b$, finding the set of neuron parameters that minimize the empirical squared error,

$$\hat{w} = \text{argmin}_{\|w_k\|_1 \leq 1, k \in \{1,\ldots,K\}} \sum_{i=1}^{N} (y_i - f_w(x_i))^2, \tag{1.15}$$

with a network width $K = O([N/(\log d)]^{1/2})$ yields a statistical risk control of the order [11]

$$E[\|f_{\hat{w}} - f\|^2] = O(\Big(\frac{\log(d)}{N}\Big)^{\frac{1}{2}}), \tag{1.16}$$

provided there is sub-Gaussian control of the distribution of the response Y. The expectation here is with respect to the training data, while the norm square provides the expectation for the loss at an independent new input vector. Analogous deep net conclusions are also in [10], [11].

There has been much research to understand theoretically the optimization of neural networks via gradient based methods [15, 58, 24, 33, 45]. These approaches work by comparing the network to a certain infinite width limit under initialization and scaling assumptions (called the neural tangent kernel, NTK), where the network trained under gradient descent approaches a kernel ridge regression solution. They also utilize a scaling of $1/\sqrt{K}$ on their outer weights rather than the $1/K$ scaling we use.

When choosing network size for favorable statistical risk, we prefer to work with $K < N$. Indeed, our later results will show $K = O([N/(\log d)]^{1/4})$ is an appropriate size for statistical risk control. Then, even in the single-hidden-layer case, no known optimization algorithm is able to solve this optimization problem in a polynomial number of iterations in N and d. Thus, approximation and estimation has been well studied for optimization algorithms on neural networks, but pairing this analysis with computational complexity bounds has remained elusive.

1.3 Bayesian Model

Instead of optimization procedures, we use a Bayesian method of estimation placing a posterior distribution on neuron parameters. We will build upon the proven approximation results for the class of neural networks, establish estimation bounds via risk control that is comparable to the optimization procedures, and establish computational complexity control.

Bayesian neural networks have been studied for many years [52, 25, 16], although specific mixing time bounds for MCMC to guarantee polynomial time complexity have been a barrier to their implementation. Recent approaches have studied the simplification of the posterior in the NTK regime, resulting in the posterior being near the posterior associated with a Gaussian process prior [30, 28]. These approaches require $K/N \to \infty$ to achieve that simplification of the posterior density. The bounded K/N setting is shown in

[30, 28] to be distinct with potentially more flexible non-Gaussian process behavior. In our work we consider $K < N$. Our optimal risk control results will have $K = [N/(\log d)]^{1/4}$ in the joint sampling case, and $K = [N/(\log d)]^{1/3} \log([N/(\log d)]^{1/3})$ in the greedy Bayes procedure. Therefore, our posterior densities are in the regime where flexibility arises in our model and the internal weights are adapted by the posterior.

Say we have data consisting of N input and response pairs $(x_i, y_i)_{i=1}^N$. Define a prior distribution P_0 on $\mathbb{R}^{Kd}$, with density p_0 with respect to a reference measure η (e.g. Lebesgue or counting measure). For each index $i \in \{1, \ldots, N\}$ define the residual of a neural network as

$$\text{res}_i(w) = y_i - \sum_{k=1}^{K} c_k \psi(w_k \cdot x_i). \tag{1.17}$$

For any subset of the data $n \leq N$, define the n-fold loss function as half the sum of squares of the first n residuals

$$\ell_n(w) = \frac{1}{2} \sum_{i=1}^{n} (\text{res}_i(w))^2. \tag{1.18}$$

Define a gain or inverse-temperature parameter $\beta > 0$. Then for every $n \leq N$, define a sequence of posterior densities trained on subsets of the data $x^n, y^n \equiv (x_i, y_i)_{i=1}^n$, by

$$p_n(w|x^n, y^n) = \frac{p_0(w) e^{-\beta \ell_n(w)}}{\int e^{-\beta \ell_n(w)} p_0(w) \eta(dw)}. \tag{1.19}$$

Denote the mean with respect to this density, the n-th posterior mean, as

$$\mu_n(x|x^n, y^n) = E_{P_n}[f(w, x)|x^n, y^n], \tag{1.20}$$

where $E_{P_n}[\cdot]$ denotes expectation with respect to the indicated distribution. For a given weight vector w, define the predictive density $p(y|x, w)$ to be Normal$(f(x, w), \frac{1}{\beta})$. Define

the n-th posterior predictive density as

$$p_n(y|x, x^n, y^n) = E_{P_n}[p(y|x, w)|x^n, y^n]. \tag{1.21}$$

For convenience, we may drop the notation conditioning on x^n, y^n and simply refer to the density as $p_n(w)$, and similarly for the mean and predictive density.

Define the Cesàro average posterior as,

$$q^{\text{avg}}(w|x^N, y^N) = \frac{1}{N+1} \sum_{n=0}^{N} p_n(w|x^n, y^n). \tag{1.22}$$

Also define the Cesàro average of the posterior means and the Cesàro average predictive density

$$\hat{g}(x) = \frac{1}{N+1} \sum_{n=0}^{N} \mu_n(x|x^n, y^n), \tag{1.23}$$

and

$$q^{\text{avg}}(y|x, x^N, y^N) = \frac{1}{N+1} \sum_{n=0}^{N} p_n(y|x, x^n, y^n). \tag{1.24}$$

A typical Bayesian may be confused why we work with the Cesàro average and why we condition on different subsets of the data. Indeed, the posterior distribution using all the data $p_N(w|x^N, y^N)$ and its posterior mean $\mu_N(x|x^N, y^N)$ are the only objects studied in most Bayesian investigations, as these produce optimal performance in a Bayes risk sense, averaging functions over choices of w with respect to the prior, as explained further below (equation (1.49) and the surrounding discussion). However, our choice of prior is not a matter of subjective belief, but rather a computational device to produce estimators of provable quality. Thus we are interested not just in such average risk, but also in what can be said for risk bounds for arbitrary g in the closure of the convex hull of $V\Psi$. This we will do with estimators constructed from the sequence of posteriors $p_n(w|x^n, y^n)$ with $n \leq N$,

rather than just the posterior based on all the training data.

There are additional reasons we may be interested in the different subset defined posteriors. One reason is in an online learning problem, we may predict y_n using the posterior mean trained on data up to index $n-1$ and consider the regret

$$\frac{1}{N}\sum_{n=1}^{N}(y_n - \mu_{n-1}(x_n|x^{n-1}, y^{n-1}))^2 - (y_n - g(x_n))^2. \tag{1.25}$$

Furthermore, the predictive densities have a nice interpretation using Bayes factors. If we define

$$Z_n = \int (\frac{\beta}{2\pi})^{\frac{n}{2}} e^{-\frac{\beta}{2}\sum_{i=1}^{n}(y_i - f_w(x_i))^2} P_0(dw), \tag{1.26}$$

then the posterior predictive densities are equal to the ratio of successive Bayes factors

$$p_n(y_{n+1}|x_{n+1}, x^n, y^n) = \frac{Z_{n+1}}{Z_n}. \tag{1.27}$$

When one considers logarithmic notions of regret, this results in a nice chain rule for information theory which we take advantage of in our risk analysis.

We also construct a greedy Bayes model, where each neuron is trained one at a time based on the residuals of previous fits. The construction of this model is more involved, using a sequence of recursively defined posterior densities. Details of its construction are presented in Chapter 4 of the thesis.

1.3.1 Choice of Prior

We consider two priors in the course of the thesis. Define the set S_1^d as the ℓ_1 ball of unit norm,

$$S_1^d = \{w \in \mathbb{R}^d : \|w\|_1 \leq 1\}. \tag{1.28}$$

For some positive integer $M \leq d$, consider the discrete set which is the intersection of S_1^d with the lattice of points of equal spacing $\frac{1}{M}$. Define this set as $S_{1,M}^d$,

$$S_{1,M}^d = \{w : Mw \in \{-M, \ldots, M\}^d,\ \|w\|_1 \leq 1\}. \tag{1.29}$$

That is, each coordinate $w_{k,j}$ can only be integer multiples of the grid size $\frac{1}{M}$ and we force the ℓ_1 norm to be less than or equal to 1. Define the K fold product of these sets as $(S_1^d)^K$ and $(S_{1,M}^d)^K$.

The first prior we will consider is uniform on the set $(S_1^d)^K$. That is, independently each weight vector w_k is uniform on S_1^d. This can be constructed by the vector of absolute values $|w_k|$ being Dirichlet$(1, 1, \ldots, 1)$ and the signs of each coordinate are independent Rademacher random variables. This has the density function

$$p_0(w) = \prod_{k=1}^{K} \Big(1\{\|w_k\|_1 \leq 1\} \frac{1}{\text{Vol}(S_1^d)} \Big). \tag{1.30}$$

with respect to Lebesgue measure.

We will also consider a discrete version of this density. We consider the prior under which w_k is independent uniform on the discrete set $S_{1,M}^d$. This has probability mass

function

$$p_0(w) = \prod_{k=1}^{K} \Big(1\{w_k \in S_{1,M}^d\} \frac{1}{|S_{1,M}^d|} \Big). \tag{1.31}$$

with respect to counting measure in $(S_{1,M}^d)^K$. When d is large one may choose a smaller order M to arrange sparsity in the weight vector, as at most M of the d coordinates can be non-zero. Furthermore, we have a bound on the cardinality of the support set $|S_M^d| \leq (2d+1)^M$ which will prove useful in future statistical risk analysis. Most notably, $\log |S_M^d|$ only grows logarithmically in the dimension d of the weight vectors.

We can also consider both of these priors as specific marginals of a joint coupled Dirichlet and Multinomial distribution. We arrange a continuous vector $w^{\text{cont}} \in (S_1^d)^K$ and a discrete vector $w^{\text{disc}} \in (S_{1,M}^d)^K$. Say the signs of each coordinate $w_{k,j}^{\text{cont}}$ are distributed as independent Rademacher. Then, for each index k, the vector of absolute values $(|w_k^{\text{cont}}|, |w_k^{\text{disc}}|)$ are independent and distributed as follows. $|w_k^{\text{cont}}|$ is uniform on the $d+1$ dimensional simplex, which is symmetric Dirichlet using the all 1's parameter vector. Then $|w_k^{\text{disc}}|$ conditioned on $|w_k^{\text{cont}}|$ is distributed as $1/M$ times a Multinomial$(M, |w^{\text{cont}}|)$ distribution. This results in w^{cont} and w^{disc} being marginally uniform on $(S_1^d)^K$ and $(S_{1,M}^d)^K$ respectively, but being coupled via this joint distribution.

The continuous prior will be used to prove the log-concave coupling form of our target density, but the finite size of the support of the discrete prior will prove useful for statistical risk control. We are ultimately unable to extend the risk control of the discrete prior to the continuous prior as well, but discuss in Chapter 6 a method that may allow the results to be connected.

1.4 Log-Concave Coupling

Consider the log-likelihood of the posterior densities $p_n(w)$ as defined in equation (1.19), with the continuous uniform prior on $(S_1^d)^K$. The log-likelihood and score of the posterior within the constrained set are equal to (with some constant B that is just the log normalizing constant)

$$\log p_n(w) = -\beta \ell_n(w) + B \tag{1.32}$$

$$\nabla_{w_k} \log p_n(w) = \beta \sum_{i=1}^{n} \text{res}_i(w)(c_k \psi'(w_k \cdot x_i) x_i). \tag{1.33}$$

Denote the Hessian as $H_n(w) \equiv \nabla^2 \log p_n(w)$. The density $p_n(w)$ is log-concave if $H_n(w)$ is negative definite for all choices of w. For any vector $u \in \mathbb{R}^{Kd}$, with blocks $u_k \in \mathbb{R}^d$, the quadratic form $u^{\mathrm{T}} H_n(w) u$ can be expressed as

$$-\beta \sum_{i=1}^{n} \Big(\sum_{k=1}^{K} c_k \psi'(w_k \cdot x_i) u_k \cdot x_i \Big)^2 \tag{1.34}$$

$$+\beta \sum_{i=1}^{n} \text{res}_i(w) \sum_{k=1}^{K} c_k \psi''(w_k \cdot x_i)(u_k \cdot x_i)^2. \tag{1.35}$$

It is clear that for any vector u the first line (1.34) is a negative term, but term (1.35) may be positive. The scalar values $c_k \psi''(w_k \cdot x_i)$ could be either a positive or negative value for each k and i, while the residuals $\text{res}_i(w)$ can also be positive or negative signed. Thus, the Hessian is not a negative definite matrix in general and $p_n(w)$ may not be a log-concave density.

The term (1.35) is capturing how the non-linearity of ψ, which provides the benefit of neural networks over linear regression, is complicating matters. If ψ were linear, $\psi''(z) = 0$ for all z and we would have a simple linear regression problem. However, since ψ has second derivative contributions, this term must be addressed.

For each data index $i \in \{1, \ldots, n\}$ and each neuron index $k \in \{1, \ldots, K\}$ we introduce a coupling with an auxiliary random variable $\xi_{i,k}$. The goal of this auxiliary random variable is to force the corresponding individual i, k terms in (1.35) to be negative.

For a value $\rho > 0$, conditioning on a weight vector w, define the forward coupling as conditionally independent random variables $\xi_{i,k}$ which are normal with mean $w_k \cdot x_i$ and variance $\frac{1}{\rho}$,

$$\xi_{i,k} \sim \text{Normal}(w_k \cdot x_i, \frac{1}{\rho}). \tag{1.36}$$

This then defines a forward conditional density (or coupling)

$$p_n(\xi|w) \propto e^{-\frac{\rho}{2}\sum_{i,k}(\xi_{i,k} - x_i \cdot w_k)^2}, \tag{1.37}$$

and a joint density for w, ξ,

$$p_n(w, \xi) = p_n(w)p_n(\xi|w). \tag{1.38}$$

Via Bayes' rule, this joint density also has expression using the induced marginal on the auxiliary ξ random vector and the reverse conditional density on $w|\xi$,

$$p_n(w, \xi) = p_n(\xi)p_n(w|\xi). \tag{1.39}$$

We will show, with slight modification of ξ to restrict its domain to a highly likely set $\xi \in B$ and with properly chosen ρ, this choice of normal forward coupling provides a negative definite correction to the Hessian of the log-likelihood of $p_n(w|\xi)$ compared to what we had with $p_n(w)$, resulting in a log-concave reverse conditional density.

Furthermore, when the dimension d and number of neurons K exceed

$$Kd \geq C(\beta N)^2 \tag{1.40}$$

for a given constant C, then the induced marginal density $p(\xi)$ is log-concave as well. Further discussion of the specifics of the log-concave coupling for the joint distribution is provided in Chapter 2, and construction in the greedy case is provided in Chapter 4.

1.5 Risk and Regret

When analyzing the performance of our posterior estimators, we will consider two main measures of performance: arbitrary sequence regret, and statistical risk.

For arbitrary sequence regret, let $(x_i, y_i)_{i=1}^N$ be an arbitrary sequence of inputs and response values with no assumption on the underlying data relationship between x_i and y_i. Consider g as an arbitrary competitor function we wish to measure our Bayesian posteriors against.

Then we define notions of square regret, randomized regret, and log regret as follows

$$R^{\text{square}} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\Big[(y_n - \mu_{n-1}(x_n|x^{n-1}, y^{n-1}))^2 - (y_n - g(x_n))^2\Big] \tag{1.41}$$

$$R^{\text{rand}} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\Big[E_{P_{n-1}}[(y_n - f(x_n, w))^2|x^{n-1}, y^{n-1}] - (y_n - g(x_n))^2\Big] \tag{1.42}$$

$$R^{\text{log}} = \frac{1}{N}\sum_{n=1}^{N}\Big[\frac{\log \frac{1}{p_{n-1}(y_n|x_n, x^{n-1}, y^{n-1})}}{\beta} - \frac{1}{2}(y_n - g(x_n))^2\Big]. \tag{1.43}$$

Our regret analysis will primarily focus on log regret, as this has connections to information theory via a chain rule and can be upper bounded by the index of resolvability method. Random regret and squared regret can then be related to upper bounds on log re-

gret. Regret analysis for the joint sampling problem is presented in Chapter 3, with greedy analysis presented in Chapter 5. As a summary of our results, the joint sampling problem has a bound on square regret of the order $O([(\log d)/N]^{\frac{1}{4}})$, while the greedy Bayes method provides an improved bound of the order $O([(\log d)/N]^{\frac{1}{3}})$, relative to the regret of g with respect to its projection into the closure of the convex hull of signed neurons $\tilde{g}$.

For statistical risk, we use the loss function which is half the squared difference

$$L(g(X), f(X)) = \frac{1}{2}(g(X) - f(X))^2. \tag{1.44}$$

We use half the squared difference, as it has better connection with the log-likelihood of a normal random variable. We see that comparing the log probabilities of normal distributions,

$$\frac{\log \frac{1}{p(y|x,w)}}{\beta} - \frac{\log \frac{1}{q(y|g(x))}}{\beta} \tag{1.45}$$

$$= \frac{1}{\beta}\Big[- \log[(\frac{\beta}{2\pi})^{\frac{1}{2}} e^{-\frac{\beta}{2}(y - f_w(x))^2}] + \log[(\frac{\beta}{2\pi})^{\frac{1}{2}} e^{-\frac{\beta}{2}(y - g(x))^2}]\Big] \tag{1.46}$$

$$= \frac{1}{2}(y - f_w(x))^2 - \frac{1}{2}(y - g(x))^2 \tag{1.47}$$

the $1/2$ appears as a natural term. Thus incorporating the half into our notion of loss allows for simpler relationship with logarithmic regret, which is a key tool of our method to bound statistical risk.

Assume the data $(X_i, Y_i)_{i=1}^N$ are iid from a data distribution with $E[Y|X] = g(X)$. Then squared risk of the Cesàro average of the posterior means is expressed as

$$E_{P_{X^N, Y^N}}[E_{P_X}[\frac{1}{2}(g(X) - \hat{g}(X))^2]]. \tag{1.48}$$

Note that even though the estimator $\hat{g}$ is defined by Bayesian posterior means, this is a frequentist notion of risk.

In a fully Bayesian analysis of risk, we would assume a set of target weights w^* is drawn from our prior P_0, and then $g = f_{w^*}$ defines the data distribution that generates X and Y. Then the Bayesian notion of risk would be the expectation over P_0 of the different risks at different w^* values,

$$E_{P_0}[E_{P_{X^N,Y^N}}[E_{P_X}[\frac{1}{2}(f_{w^*}(X) - \hat{g}(X))^2]]]. \tag{1.49}$$

Then it is clear the Bayesian posterior mean using all the data $\mu_N(X|X^N, Y^N)$ would optimize this notion of risk [42, Chapter 4, Thm 1.1 and Cor 1.2].

Yet this is not the perspective we take. This method of Bayesian risk analysis assumes the prior P_0 which defines our Bayesian model actually defines how the g function is realized. Furthermore, two different Bayesians with different priors would have to fight about who has the "optimal" posterior mean, and both could claim their posterior mean is the optimal estimator with respect to the prior they define.

We do not view our Bayesian model as actually defining the distribution of anything about $P_{X,Y}$ and g. Our prior P_0 and posterior based on squared loss is a computational tool for the purposes of producing a posterior mean. We do not assume that g is itself some neuronal network f_{w^*}, nor that $Y|X$ is normal (which would correspond to our choice of square loss, if we interpreted our posteriors as coming from a choice of forward likelihood paired with a prior). We allow g to be whatever function it may be, and still prove that our Bayesian method, as a computational tool, produces the Cesàro average of the posterior means as a good estimator for g. If g is not a neural network, and $\tilde{g}$ is its $L_2(P_X)$ projection into the closure of the convex hull of signed neurons scaled by V, we show

$$E_{P_{X^N,Y^N}}[E_{P_X}[\frac{1}{2}(g(X) - \hat{g}(X))^2]] = O((\frac{\log(d)}{N})^{\frac{1}{4}}) + E_{P_X}[\frac{1}{2}(g(X) - \tilde{g}(X))^2]. \tag{1.50}$$

The greedy Bayes estimator can achieve $O([(\log d)/N]^{\frac{1}{3}})$. For the Cesàro average of the

posterior means, the square risk can be interpreted as an expected square regret, so much of our risk analysis follows first from analyzing regret and adapting these results from a worst case to an average case analysis.

1.6 Notation

Here we present the mathematical notation used in the thesis.

- Capital P refers to a probability distribution, while lowercase p is its probability mass or density function.
- $f'(\cdot)$ refers to the derivative of a scalar function f.
- ∇ is the gradient operator and ∇^2 is the Hessian operator, producing a matrix of second derivatives.
- $\{1, \ldots, N\}$ is the set of whole numbers between 1 and N.
- $[a, b]$ is the interval of real values between a and b.
- $u \cdot v$ is the Euclidean inner product between two vectors.
- $u^{\mathrm{T}}, \mathbf{X}^{\mathrm{T}}$ refers to the transpose of a vector or matrix, so quadratic forms of a vector u with the matrix $\mathbf{X}$ will be written as $u^{\mathrm{T}}\mathbf{X}u$.
- $\|w\|_p$ refers to the ℓ_p norm, $\|w\|_p = (\sum_j (w_j)^p)^{\frac{1}{p}}$.
- The ℓ_1 ball is denoted as $S_1^d = \{w \in \mathbb{R}^d : \|w\|_1 \leq 1\}$.
- The K fold Cartesian product of this set is $(S_1^d)^K$.
- For variables in a sequence, superscripts indicate the set of variables $X^n = (X_i)_{i=1}^n$.

- For a data sequence $(x_i, y_i)_{i=1}^N$, given a function f associate it with the vector with coordinates equal to the function outputs $f_i = f(x_i)$. For any two vectors of length N define the empirical squared norm and inner product

$$\|h_1 - h_2\|_N^2 = \sum_{i=1}^N (h_{1,i} - h_{2,i})^2 \qquad \langle h_1, h_2 \rangle_N = \sum_{i=1}^N h_{1,i} h_{2,i}$$

- Logarithms in the thesis are natural logarithms.

Chapter 2

Log-Concave Coupling for Joint Sampling

2.1 Introduction

Consider the log-likelihood of the posterior densities $p_n(w)$ as defined in equation (1.19), with the continuous uniform prior on $(S_1^d)^K$. Let η be a reference measure, which in this case is Lebesgue measure. As discussed in the introduction, the Hessian of the log-likelihood for $p_n(w)$ is not negative definite, so $p_n(w)$ is not a log-concave density. Our definition of a **log-concave coupling** is a joint density with an auxiliary random variable ξ under which w maintains the same marginal density, the induced marginal density for ξ is log-concave, and the reverse conditional density for $w|\xi$ is log-concave for all ξ in the support of the joint density.

Note that to maintain the the marginal density of w, we must define our ξ via a choice of forward coupling. That is, with $p_n(w)$ as our target density we define a forward conditional density $p_n(\xi|w)$ conditioned on w and then define our joint density as

$$p_n(w, \xi) = p_n(w)p_n(\xi|w). \tag{2.1}$$

Then this defines an induced marginal on ξ by integrating out the w variable,

$$p(\xi) = \int p_n(w) p_n(\xi|w) \eta(dw). \tag{2.2}$$

Then via Bayes' rule, this also defines a reverse conditional density on $w|\xi$ as

$$p_n(w|\xi) = \frac{p_n(w) p_n(\xi|w)}{p_n(\xi)}. \tag{2.3}$$

Consider the log-likelihood of the reverse conditional density,

$$\log p_n(w|\xi) = \log p_n(w) + \log p_n(\xi|w) - \log p_n(\xi). \tag{2.4}$$

$\log p_n(\xi|w)$ will be seen to be a concave function in w (it is essentially a negative cumulant generating function). Thus, this term will add negative definite correction to the reverse conditional density's log-likelihood Hessian. With enough correction, we can overpower any positive definite terms in the Hessian of $\log p_n(w)$ and produce an overall log-concave reverse conditional density. We then must study if the marginal density for ξ is also log-concave under this construction.

For each data index $i \in \{1, \ldots, n\}$ and each neuron index $k \in \{1, \ldots, K\}$ we introduce a coupling with an auxiliary random variable $\xi_{i,k}$. Define the values

$$C_n = \max_{i \leq n} |y_i| + a_0 V \tag{2.5}$$

$$\rho_n = \rho_{n,K} = a_2 \frac{\beta C_n V}{K}. \tag{2.6}$$

We will consider our posterior densities with one fixed value of n at a time. Likewise think of K as fixed, so we will refer to these values as constants in our discussion. We will work with $\rho = \rho_{n,K}$ when it is clear we are talking about a fixed n and K value.

Ultimately we will use bounded auxiliary random variables to yield the desired log-

concave coupling. But to motivate the construction first consider tentatively a simpler unbounded construction.

Conditioning on a weight vector w, define the forward coupling as conditionally independent random variables $\xi_{i,k}$ which are normal with mean $w_k \cdot x_i$ and variance $\frac{1}{\rho}$,

$$\xi_{i,k} \sim \text{Normal}(w_k \cdot x_i, \frac{1}{\rho}). \tag{2.7}$$

This then defines a forward conditional density (or coupling)

$$p_n(\xi|w) \propto e^{-\frac{\rho}{2}\sum_{i,k}(\xi_{i,k} - x_i \cdot w_k)^2}, \tag{2.8}$$

and a joint density for w, ξ,

$$p_n(w, \xi) = p_n(w)p_n(\xi|w). \tag{2.9}$$

2.2 Reverse Conditional Density

First, we allow for $\xi_{i,k}$ to take arbitrary real values arising from the conditional normal distribution.

Theorem 2.1. *Under the continuous uniform prior and $\xi_{i,k} \sim$ Normal$(x_i \cdot w_k, 1/\rho)$ for the given choice of ρ, the reverse conditional density $p_n(w|\xi)$ is log-concave for the given ξ coupling.*

Proof. The log-likelihood for $p_n(w|\xi)$ is given by

$$\log p_n(w|\xi) = -\beta\ell_n(w) + B_n(\xi) \tag{2.10}$$

$$- \sum_{i=1}^{n}\sum_{k=1}^{K} \frac{\rho}{2}(\xi_{i,k} - w_k \cdot x_i)^2, \tag{2.11}$$

for some function $B_n(\xi)$ which does not depend on w and is only required to make the density integrate to 1. The term (2.11) is a negative quadratic in w which treats each w_k as an independent normal random variable. Thus, the additional Hessian contribution will be a $(Kd) \times (Kd)$ negative definite block diagonal matrix with $d \times d$ blocks of the form $\rho \sum_{i=1}^n x_i x_i^{\mathrm{T}}$. Denote the Hessian as $H_n(w|\xi) \equiv \nabla^2 \log p_n(w|\xi)$. For any vector $u \in \mathbb{R}^{Kd}$, with blocks $u_k \in \mathbb{R}^d$, the quadratic form $u^{\mathrm{T}} H_n(w|\xi) u$ can be expressed as

$$-\beta \sum_{i=1}^n \Big(\sum_{k=1}^K c_k \psi'(w_k \cdot x_i) u_k \cdot x_i \Big)^2 \tag{2.12}$$

$$+ \sum_{k=1}^K \sum_{i=1}^n (u_k \cdot x_i)^2 [\beta \mathrm{res}_i(w) c_k \psi''(w_k \cdot x_i) - \rho]. \tag{2.13}$$

By the assumptions on the second derivative of ψ and the definition of ρ we have

$$\max_{i,k} (\beta \mathrm{res}_i(w) c_k \psi''(w_k \cdot x_i) - \rho) \leq 0. \tag{2.14}$$

So all the terms in the sum in (2.13) are negative. Thus, the Hessian of the log-likelihood of $p_n(w|\xi)$ is negative definite and $p_n(w|\xi)$ is a log-concave density. □

While this proof offers a simple way to make a conditional density $p_n(w|\xi)$ which is log-concave, we also wish to study if there is log-concavity of the induced marginal of $p_n(\xi)$. The joint log-likelihood for $p_n(w, \xi)$ contains a bilinear term in ξ, w from expanding the quadratic,

$$\sum_{k=1}^K \sum_{j=1}^d w_{k,j} \sum_{i=1}^n \xi_{i,k} x_{i,j}. \tag{2.15}$$

We want some control on how large this term can become, so we restrict the allowed

support of ξ. We define a slightly larger $\rho = \rho_{n,K}$ value than before,

$$\rho_{n,K} = \sqrt{\frac{3}{2}} a_2 \frac{\beta C_n V}{K}. \tag{2.16}$$

For a positive $\delta \leq 1/16$, we also define a constrained set,

$$B = \Big\{ \xi_{i,k} : \max_{j,k} |\sum_{i=1}^{n} x_{i,j} \xi_{i,k}| \leq n + \sqrt{2 \log(\frac{2Kd}{\delta})} \sqrt{\frac{n}{\rho}} \Big\}. \tag{2.17}$$

We then define our forward conditional distribution for $p_n^*(\xi|w) = p_n(\xi|w, B)$ as the normal distribution restricted to the set B,

$$p_n^*(\xi|w) = p_n(\xi|w, B) = \frac{1_B(\xi) p_n(\xi|w)}{P_n(\xi \in B|w)} \tag{2.18}$$

$$= 1_B(\xi) \frac{\prod_{i=1}^{n} \prod_{k=1}^{K} \left(\frac{\rho}{2\pi}\right)^{\frac{1}{2}} e^{-\frac{\rho}{2}(\xi_{i,k} - x_i \cdot w_k)^2}}{\int_B \prod_{i=1}^{n} \prod_{k=1}^{K} \left(\frac{\rho}{2\pi}\right)^{\frac{1}{2}} e^{-\frac{\rho}{2}(\xi_{i,k} - x_i \cdot w_k)^2} d\xi}. \tag{2.19}$$

Under this constrained density, the term (2.15) will be bounded for any choice of $\xi \in B$ and $w_k \in S_1^d$, which will be a useful property in later proofs.

The denominator of this fraction is the normalizing constant of the density as a result of the restricting set B. Denote the log normalizing constant as $Z(w) = \log[P_n(\xi \in B|w)]$,

$$Z(w) = \log \int_B \prod_{i=1}^{n} \prod_{k=1}^{K} \left(\frac{\rho}{2\pi}\right)^{\frac{1}{2}} e^{-\frac{\rho}{2}(\xi_{i,k} - x_i \cdot w_k)^2} d\xi. \tag{2.20}$$

An equivalent expression for the forward coupling is then

$$p_n^*(\xi|w) = 1_B(\xi) \left(\frac{\rho}{2\pi}\right)^{\frac{NK}{2}} e^{-\frac{\rho}{2} \sum_{i,k} (\xi_{i,k} - x_i \cdot w_k)^2 - Z(w)}. \tag{2.21}$$

This construction also yields for $\xi \in B$ the induced marginal density $p_n^*(\xi)$ with respect to

Lebesgue measure,

$$p_n^*(\xi) = \int p_n(w) p_n^*(\xi|w)\eta(dw) = \frac{1_B(\xi)\int p_n(w) e^{-\frac{\rho}{2}\sum_{i,k}(\xi_{i,k}-x_i\cdot w_k)^2 - Z(w)}\eta(dw)}{\int_B \int p_n(w) e^{-\frac{\rho}{2}\sum_{i,k}(\xi_{i,k}-x_i\cdot w_k)^2 - Z(w)}\eta(dw)d\xi}, \tag{2.22}$$

and the reverse conditional density $p_n^*(w|\xi)$ with respect to reference measure η,

$$p_n^*(w|\xi) = \frac{p_n(w)p_n^*(w|\xi)}{p_n^*(\xi))} = \frac{p_n(w) e^{-\frac{\rho}{2}\sum_{i,k}(\xi_{i,k}-x_i\cdot w_k)^2 - Z(w)}}{\int p_n(w) e^{-\frac{\rho}{2}\sum_{i,k}(\xi_{i,k}-x_i\cdot w_k)^2 - Z(w)}\eta(dw)}. \tag{2.23}$$

Note these densities differ from the $p_n(w|\xi)$ and $p_n(\xi)$ defined before without restricting to the set B due to the presence of the $Z(w)$ function. We then show that for $\xi \in B$ the density $p_n^*(w|\xi)$ is a very similar density to $p_n(w|\xi)$ and also log-concave.

The restriction of ξ to the set B is the restriction to a very likely set under the unconstrained coupling, in particular we have the following:

Lemma 2.1. *For any weight vector w with $\|w_k\|_1 \leq 1$ the set B in* (2.17) *has probability under $p(\xi|w)$ at least*

$$P(\xi \in B|w) \geq 1 - \frac{\delta}{\sqrt{2\log(2Kd/\delta)}}. \tag{2.24}$$

Proof. See Appendix, Section 2.6.1. □

Furthermore, the function $Z(w)$ is nearly constant, having small first and second derivative. Therefore, the function has little impact on the log-likelihood.

Lemma 2.2. *For any specified vector $u \in R^{Kd}$, define the value*

$$\tilde{\sigma}^2 = \frac{\sum_{i=1}^n \sum_{k=1}^K (u_k \cdot x_i)^2}{\rho}. \tag{2.25}$$

For positive values $\delta \leq 1/16$ with $Kd \geq 4$, we then have upper bounds,

$$|u \cdot \nabla Z(w)| \leq \frac{\rho\tilde{\sigma}}{1-\delta}\frac{\delta}{\sqrt{2\pi}} \tag{2.26}$$

and

$$|u^T(\nabla^2 Z(w))u| \leq \frac{\rho^2\tilde{\sigma}^2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\Big(2\sqrt{2\log(1/\delta)} + \frac{\rho^2\tilde{\sigma}^2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\Big). \tag{2.27}$$

Note both bounds go to 0 as $\delta \to 0$, and thus can be made arbitrarily small for a certain choice of δ.

Proof. See Appendix, Section 2.6.1. □

Thus, with restriction to the set B, whose size is determined by δ, and a slightly larger ρ, we can give a similar result to Theorem 2.1. Note this result is for $p_n^*(w|\xi)$ which is distinct from $p_n(w|\xi)$ due to the presence of the $Z(w)$ function in the log-likelihood and the restriction to $\xi \in B$.

Theorem 2.2. *Define the notation*

$$H_1(\delta) = \frac{2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\sqrt{2\log\frac{2}{\delta}} \tag{2.28}$$

$$H_2(\delta) = \Big(a_2\frac{\beta C_n V}{K}\Big)^2\frac{1}{2\pi}\frac{\delta^2}{(1-\delta)^2}. \tag{2.29}$$

Assume a sufficiently small $\delta \leq \frac{1}{16}$ that satisfies

$$H_1(\delta) \leq \frac{1}{100} \tag{2.30}$$

$$H_2(\delta) \leq \frac{1}{10}. \tag{2.31}$$

For the continuous uniform prior, with ξ restricted to the set B defined by δ, and ρ as in equation (2.16)*, the reverse conditional density $p_n^*(w|\xi)$ is a log-concave density in w, for*

any ξ in B.

Proof. See Appendix, Section 2.6.2. □

Corollary 2.1. *A positive δ which satisfies,*

$$\delta \leq \min\Big(\frac{1}{300}, \sqrt{\frac{2\pi}{11}}\frac{K}{a_2\beta C_N V}\Big), \tag{2.32}$$

will satisfy conditions (2.30)*,* (2.31)*.*

The pairing of a normal forward coupling to $p_n^*(\xi|w)$ with a target density $p_n(w)$ to produce a reverse conditional $p_n^*(w|\xi)$ which is log-concave is not a new idea. As we will later discuss, the same concept is used in proximal sampling methods and diffusion models. However, in this work we go further in stating that the induced marginal on $p_n^*(\xi)$ is itself log-concave, which we call a log-concave coupling.

2.3 Marginal Density

Lemma 2.3. *The score and Hessian of the induced marginal density for $p_n^*(\xi)$ for $\xi \in B$ are expressed as*

$$\partial_{\xi_{i,k}} \log p_n^*(\xi) = -\rho\,\xi_{i,k} + \rho\, x_i \cdot E_{P_n^*}[w_k|\xi] \tag{2.33}$$

$$\partial_{\xi_{i_1,k_1},\xi_{i_2,k_2}} \log p_n^*(\xi) = -\rho 1\{(i_1,k_1)=(i_2,k_2)\} \tag{2.34}$$

$$+\rho^2 \boldsymbol{Cov}_{P_n^*}[w_{k_1}\cdot x_{i_1}, w_{k_2}\cdot x_{i_2}|\xi]. \tag{2.35}$$

Equivalently in vector form using the n by d data matrix $\boldsymbol{X}$,

$$\nabla \log p_n^*(\xi) = \rho\Big(-\xi + E_{P_n^*}\begin{bmatrix}\boldsymbol{X}w_1\\ \cdots \\ \boldsymbol{X}w_K\end{bmatrix}|\xi\Big) \tag{2.36}$$

$$\nabla^2 \log p_n^*(\xi) = \rho\Big(-I + \rho\,\boldsymbol{Cov}_{P_n^*}\begin{bmatrix}\boldsymbol{X}w_1\\ \cdots \\ \boldsymbol{X}w_K\end{bmatrix}|\xi\Big). \tag{2.37}$$

Proof. The stated results are a consequence of simple calculus, but we will derive them using a statistical interpretation that avoids tedious calculations.

The log-likelihood of the induced marginal for $p_n^*(\xi)$ is equal to the log of the joint density with w integrated out,

$$\log p_n^*(\xi) = \log\Big(\int p_n(w) p_n^*(\xi|w)\eta(dw)\Big). \tag{2.38}$$

Rearranging the log-likelihood of the Gaussian forward conditional, this can be expressed as a quadratic term in ξ and a term which represents a cumulant generating function plus a constant. Recall $Z(w)$ as defined in equation (2.20). Denote the function

$$h(w) = -\beta\ell_n(w) - \frac{\rho}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}(w_k \cdot x_i)^2 - Z(w), \tag{2.39}$$

which is the part of the log-likelihood of the joint density which does not depend on ξ. The marginal pdf can then be expressed as

$$\log p_n^*(\xi) = -\frac{\rho}{2}\|\xi\|_2^2 \tag{2.40}$$

$$+\log\Big(\int p_0(w) e^{h(w)} e^{\rho\sum_{i=1}^{n}\sum_{k=1}^{K}\xi_{i,k}w_k\cdot x_i}\eta(dw)\Big) + C, \tag{2.41}$$

for some constant C which makes the density integrate to 1. Note that ξ is restricted to have support only on the set B, so there is an indicator of the set B we do not write in the expression for simplicity.

It is clear the term (2.41) is the cumulant generating function of the random variable $u(w)$ defined by

$$u(w) = \xi \cdot \begin{pmatrix}\mathbf{X}w_1\\ \cdots \\ \mathbf{X}w_K\end{pmatrix}, \tag{2.42}$$

when w is distributed according to the density proportional to $p_0(w)e^{h(w)}$. Thus, the gra-

dient in ξ is the mean of the vector and the second derivative is the covariance, as are standard properties of derivatives of cumulant generating functions. The density being integrated is a tilting of the log-likelihood defined by $h(w)$, and this tilted density is the reverse conditional $p_n^*(w|\xi)$. □

We highlight two important consequences of this result.

Corollary 2.2. *The score $\nabla \log p_n^*(\xi)$ is expressed implicitly as a linear transformation of the expected value of the log-concave reverse conditional $p_n^*(w|\xi)$.*

Proof. This is a simple consequence of (2.33) or (2.36). □

Remark 2.1. Therefore, while we do not have an explicit closed form expression for the score of the marginal density, it can be estimated using an MCMC method and thus is readily available for use. In particular, to run an MCMC algorithm such as ULA or MALA on the marginal density $p_n^*(\xi)$, the score is needed. Any time the score needs to be evaluated, it can be computed via its own MCMC algorithm for $p_n^*(w|\xi)$ as needed and then utilized in the sampling algorithm for ξ itself. Possible sampling algorithms for $p_n^*(\xi)$ are discussed in Section 2.5.4.

Corollary 2.3. *The density $p_n^*(\xi)$ is log-concave if for any unit vector $u \in \mathbb{R}^{nK}$, with blocks $u_k \in \mathbb{R}^n$, the variance of a particular linear combination of w, namely*

$$v(w) = \sum_{k=1}^{K} u_k^T \boldsymbol{X} w_k, \tag{2.43}$$

with respect to the reverse conditional $p_n^*(w|\xi)$ is less than $1/\rho$,

$$Var_{P_n^*}[v(w)|\xi] \leq 1/\rho, \tag{2.44}$$

for ξ in the convex support set B.

Proof. This is a simple consequence of (2.37). □

Therefore, to show that $p_n^*(\xi)$ is log-concave we must provide an upper bound on the covariance of w using the reverse conditional density $p_n^*(w|\xi)$. Note that such conditional expectation and conditional covariance representations would also hold using $p_n(\xi)$, which is defined without conditioning on the set B and thus does not include the $Z(w)$ in the joint likelihood. However, the restrictions imposed on maximum inner products by the definition of B will prove useful in upper bounding the reverse conditional covariance.

2.4 Conditional Covariance Control

The log-likelihood for $p_n^*(w|\xi)$ is the log-likelihood of the prior density plus an additional concave term. Under a log-concave prior, one would expect that adding a concave term to the exponent of an already log-concave density should result in less variance in every direction. Thus one can conjecture the prior covariance would be more than the conditional covariance for any conditioning value,

$$\text{Cov}_{P_0}[w] \succ \text{Cov}_{P_n^*}[w|\xi] \quad \forall \xi \in B. \tag{2.45}$$

Under a Gaussian prior, such a statement would follow easily from the Brascamp-Lieb inequality [14, Proposition 2.1]. However, for the uniform prior on a convex set, this method does not directly apply.

The covariance matrix of the uniform prior on $(S_1^d)^K$ is diagonal (note the different coordinates are uncorrelated but not independent due to symmetry) with entries $\text{Var}_{P_0}(w_{k,j}) = \frac{d}{(d+1)^2(d+2)} \leq \frac{1}{d^2}$ which follows from properties of the Dirichlet distribution. Thus, under

conjecture (2.45) we would expect a bound of the form

$$\rho \text{Var}_{P_n^*}[v(w)|\xi] \leq \sqrt{\frac{3}{2}} a_2 \frac{\beta C_n V}{Kd^2} \sum_{j=1}^{d} \sum_{k=1}^{K} (\sum_{i=1}^{n} u_{i,k} x_{i,j})^2 \tag{2.46}$$

$$\leq \sqrt{\frac{3}{2}} a_2 \frac{\beta C_n V}{Kd} \sum_{k=1}^{K} \|u_k\|_1^2 \tag{2.47}$$

$$\leq \sqrt{\frac{3}{2}} a_2 \frac{\beta n C_n V}{Kd} \sum_{k=1}^{K} \|u_k\|_2^2 \tag{2.48}$$

$$= \sqrt{\frac{3}{2}} a_2 C_n V \frac{\beta n}{Kd} \tag{2.49}$$

$$\leq \sqrt{\frac{3}{2}} a_2 \frac{C_N V \beta N}{Kd}. \tag{2.50}$$

Thus for $Kd > C(\beta N)$ for some value C we would have log-concavity of the marginal. However, we are unable to prove this conjecture is true. Instead, using a different approach we will conclude for a specified C,

$$Kd \geq C(\beta N)^2 \tag{2.51}$$

results in log-concavity of the marginal density.

Instead of recreating an inequality like (2.45), we must take a different approach to upper bound the variance in any direction. Denote the function,

$$h_\xi^n(w) = -\beta \ell_n(w) - \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{\rho}{2} (\xi_{i,k} - w_k \cdot x_i)^2 - Z(w). \tag{2.52}$$

Denote the function shifted by its mean under the prior as

$$\tilde{h}_\xi^n(w) = h_\xi^n(w) - E_{P_0}[h_\xi^n(w)]. \tag{2.53}$$

Define its cumulant generating function with respect to the prior as

$$\Gamma_{\xi}^{n}(\tau) = \log E_{P_0}[e^{\tau \tilde{h}_{\xi}^{n}(w)}]. \tag{2.54}$$

Lemma 2.4. *For any integer $\ell \geq 1$ and for any vector $u \in \mathbb{R}^{Kd}$ we have the upper bound*

$$\mathit{Var}_{P_n^*}(u \cdot w|\xi) \leq \left(E_{P_0}[(u \cdot w)^{2\ell}]\right)^{\frac{1}{\ell}} e^{\frac{\ell-1}{\ell}\Gamma_{\xi}^{n}(\frac{\ell}{\ell-1}) - \Gamma_{\xi}^{n}(1)}. \tag{2.55}$$

Proof. The variance of the inner product $u \cdot w$ is less than its expected square. The reverse conditional density $p_n^*(w|\xi)$ can be expressed as

$$p_n^*(w|\xi) = e^{\tilde{h}_{\xi}^{n}(w) - \Gamma_{\xi}^{n}(1)} p_0(w). \tag{2.56}$$

We then apply a Hölder's inequality to the integral expression with parameters p and q such that $\frac{1}{p} + \frac{1}{q} = 1$

$$\begin{aligned} \mathrm{Var}_{P_n^*}(u \cdot w|\xi) &\leq E_{P_0}[(u \cdot w)^2 e^{\tilde{h}_{\xi}^{n}(w) - \Gamma_{\xi}^{n}(1)}] && (2.57) \\ &\leq \left(E_{P_0}[(u \cdot w)^{2p}]\right)^{\frac{1}{p}} \left(E_{P_0}[e^{q\tilde{h}_{\xi}^{n}(w) - q\Gamma_{\xi}^{n}(1)}]\right)^{\frac{1}{q}}. && (2.58) \end{aligned}$$

Let $p = \ell$ and $q = \frac{\ell}{\ell-1}$. The second factor can be written as

$$e^{\frac{\ell-1}{\ell}\Gamma_{\xi}^{n}(\frac{\ell}{\ell-1}) - \Gamma_{\xi}^{n}(1)}. \tag{2.59}$$

□

We then study the moments of the prior density and the behavior of the $\Gamma_{\xi}^{n}(\tau)$ function separately.

Lemma 2.5. *For any unit vector $u \in \mathbb{R}^{nK}$, with blocks $u_k \in \mathbb{R}^n$,*

$$E_{P_0}[(\sum_{k=1}^{K} u_k^T \boldsymbol{X} w_k)^{2\ell}]^{\frac{1}{\ell}} \leq \frac{4\ell n}{\sqrt{e}\, d}. \tag{2.60}$$

Proof. See Appendix, Section 2.6.3. □

Lemma 2.6. *Denote the constants*

$$A_1 = 2a_1 + 4\sqrt{\frac{3}{2}}a_2 \tag{2.61}$$

$$A_2 = \Big(2 + \frac{1}{\sqrt{\pi}}\Big)\sqrt{2a_2\sqrt{\frac{3}{2}}}. \tag{2.62}$$

Assume positive $\delta \leq \frac{1}{16}, d \geq 2, K \geq 2$. For any positive integer $\ell \geq 1$ and any ξ from the constrained set B, we have

$$\frac{\ell - 1}{\ell}\Gamma_\xi^n(\frac{\ell}{\ell - 1}) - \Gamma_\xi^n(1) \leq A_1\frac{C_n V \beta n}{\ell} + A_2\frac{\sqrt{C_n V \beta n}}{\ell}\Big(\sqrt{\log(\frac{2Kd}{\delta})}\sqrt{K}\Big). \tag{2.63}$$

Proof. See Appendix, Section 2.6.3. □

We summarize the conclusions of Lemmas 4,5,6 as follows. Ignoring certain constant factors, we have an upper bound on the variance in (2.44) for any choice of ℓ,

$$\frac{n\ell}{d}\exp\Big(\frac{\beta n + \sqrt{\beta n K \log(\frac{2Kd}{\delta})}}{\ell}\Big). \tag{2.64}$$

Ignoring for now the integer constraint, the optimal continuous choice of ℓ to minimize the expression is the numerator in the exponent. With this choice of ℓ, we would have bound

$$\frac{\beta n^2 + n^{\frac{3}{2}}\sqrt{\beta K \log(\frac{2Kd}{\delta})}}{d}. \tag{2.65}$$

Multiplying this by $\rho \propto \frac{\beta}{K}$ and upper bounding with $n \leq N$, we would have the bound

$$\frac{(\beta N)^2}{Kd}\Big(1 + \Big[\frac{K \log\left(\frac{2Kd}{\delta}\right)}{\beta N}\Big]^{\frac{1}{2}}\Big). \tag{2.66}$$

If $K \log(2Kd/\delta) \leq \beta N$, then we have a $O(\frac{(\beta N)^2}{Kd})$ bound. With a choice of d and K large enough, we can make this expression be less than 1. We make this statement more precise in the following theorem.

Theorem 2.3. *Assume $\delta \leq \frac{1}{16}, d \geq 2, K \geq 2, \beta N \geq 2$. Further assume that*

$$K \log\left(\frac{2Kd}{\delta}\right) \leq \beta N, \tag{2.67}$$

which is essentially a condition than K not be too large (that is, K is less than some multiple of βN).

Define A_1, A_2 as in (2.61)*,* (2.62) *and define the constant*

$$A_3 = 4\sqrt{\frac{3}{2e}} a_2 (C_N V)^{\frac{3}{2}} [A_1 + A_2 (C_N V)^{\frac{1}{2}}]. \tag{2.68}$$

Let d and K satisfy

$$Kd \geq A_3 (\beta N)^2. \tag{2.69}$$

Then for all $n \leq N$, the marginal density for $p_n^*(\xi)$ is log-concave under the continuous uniform prior. If equation (2.69) *is a strict inequality, the density is strictly log-concave.*

A relevant δ may be $1/Kd$ or a power thereof, though a small constant value such as say $1/300$ is also acceptable (to satisfy Corollary 2.1 for example).

Proof. Fix any $n \leq N$. By Corollary 2.3, the Hessian of $\log p_n^*(\xi)$ is log-concave when

for any unit vector u, we have

$$\rho \text{Var}_{P_n^*}[\sum_{k=1}^{K} u_k^{\mathrm{T}} \mathbf{X} w_k | \xi] \leq 1. \tag{2.70}$$

By Lemma 2.4, 2.5, 2.6 we have an upper bound for this variance for any scalar $\ell > 1$ and $\xi \in B$. Recall A_1, A_2 as defined in expressions (2.61), (2.62). Fix the choice,

$$\ell^* = A_1 C_n V \beta n + A_2 \sqrt{C_n V K \beta n \log(\frac{2Kd}{\delta})}. \tag{2.71}$$

This gives upper bound on ρ times the variance,

$$\sqrt{\frac{3}{2}} a_2 \frac{\beta C_n V}{K} \frac{4n}{\sqrt{e}d} \ell^* \tag{2.72}$$

$$= 4\sqrt{\frac{3}{2e}} A_1 a_2 \frac{(C_n V \beta n)^2}{Kd} \tag{2.73}$$

$$+ 4\sqrt{\frac{3}{2e}} A_2 a_2 \frac{(C_n V \beta n)^{\frac{3}{2}} \sqrt{K}}{Kd} \sqrt{\log(\frac{2Kd}{\delta})} \tag{2.74}$$

$$\leq 4\sqrt{\frac{3}{2e}} a_2 \frac{(\beta N)^2}{Kd} \Big[A_2 (C_N V)^2 + A_1 (C_N V)^{\frac{3}{2}} \Big(\frac{K(\log(\frac{2Kd}{\delta})}{\beta N} \Big)^{\frac{1}{2}} \Big]. \tag{2.75}$$

By assumption,

$$\frac{K \log(\frac{2Kd}{\delta})}{\beta N} \leq 1, \tag{2.76}$$

so we have upper bound on (2.70),

$$4\sqrt{\frac{3}{2e}} a_2 (C_N V)^{\frac{3}{2}} [A_1 + A_2 (C_N V)^{\frac{1}{2}}] \frac{(\beta N)^2}{Kd}. \tag{2.77}$$

If Kd satisfies condition (2.69), then ρ times the variance is less than 1 in expression (2.70). By Corollary 2.3, this implies log-concavity of the induced marginal density on ξ. □

Remark 2.2. Note under our conditions on K and d in this theorem, K must be less than some fractional power of N, N^p for some power $0 < p < 1$. Then d must be more than N to a power more than 1, $d > N^q$ for some power $q > 1$. For example, $K = N^{1/4}, \beta = 1/N^{1/4}, d > A_3 N^{5/4}$ would suffice. We need certain control on β, K in our later results to control the risk.

Remark 2.3. Note that ℓ as used in the proof via the Hölder Inequality must be an integer. This is since Lemma 2.5 wants to work with whole number moments of the prior. Whereas the ℓ^* in equation (2.71) is the optimal continuous value. We would have to round up or down to the nearest integer. This would result in $\ell^* \pm \epsilon$ for a number $|\epsilon| < 1$ in equation (2.72) instead of ℓ^*. This would give an additional term $\beta N/(Kd)$ in the expression (2.77), yet this is a lower order dependence that $(\beta N)^2/(Kd)$, so it would still be controlled.

Remark 2.4. Note the interior weight dimension d can be made artificially larger by repeating the input vectors. Say the original input vectors x_i have a default dimension of $\tilde{d}$. Define new input vectors by repeating the data L times

$$\tilde{x}_i = (x_i, \ldots, x_i) \in \mathbb{R}^{\tilde{d}L}. \tag{2.78}$$

We can then consider $\tilde{\mathbf{X}}$ as our data matrix with row dimension $d = L\tilde{d}$.

The span of the new data matrix under ℓ_1 controlled input vectors, $\{z = \tilde{\mathbf{X}}w, \|w\|_1 \leq 1\}$, is the same as the original matrix. So we have the same approximation ability of the network. This can also equivalently be considered as inducing some different prior on the original w_k weight vectors of dimension $\tilde{d}$ that is more concentrated than uniform. However, it is more convenient to consider a uniform prior in a higher $d = L\tilde{d}$ dimensional space. This is introducing even more multi-modality into the original density $p_n(w)$ as multiple longer weight vectors yield the same output in the neural network. Yet by our proceeding theorems we have shown the density can be decomposed into a log-concave mixture.

2.5 Practical Sampling Considerations

The focus of this chapter has been the establishment of the log-concave coupling. That is, showing that our target density $p(w)$ can be written as a mixture distribution

$$p(w) = \int p(w|\xi)p(\xi)d\xi \tag{2.79}$$

where $p(\xi)$ is a log-concave density, and $p(w|\xi)$ is a log-concave density. We have then claimed under this structure, it is possible to sample $p(\xi)$ and $p(w|\xi)$ in polynomial time.

There many technical considerations to back this claim of polynomial sampling. While it is generally true there are many results showing MCMC algorithms mix in polynomial time for log-concave targets [2, 3, 22, 38, 39, 47, 48], the exact details of which algorithm to use, the hyper-parameters of that algorithm such as step size or number of iterations, and the actual coding implementation of these algorithms are beyond the scope of this work. Nonetheless, we point to several references here that begin in the direction of practically implementing a sampling algorithm for the log-concave coupling.

2.5.1 Log-Concave Coupling and Existing Methods

The use of an auxiliary random variable to create log-concavity is not a new idea, and has connections to existing methods. The critical structure of our sampling problem is that our target distribution of interest can be expressed as a mixture distribution with easy to sample components. The structure of a mixture distribution has been recognized in a number of recent papers. For spin glass systems (Sherrington–Kirkpatrick models) of high temperature, [13] expanded the range of known temperatures under which a Log Sobolev constant can be established by using such a mixture structure. For a Bayesian regression problem with a spike and slab (i.e. multi-modal) prior, [51] used the mixture structure to perform easy MCMC sampling. Thus, it is clear this approach of a mixture distribution can

be applied to a number of sampling problems of interest. However, the posterior densities in these problems were much simpler than ours, making explicit use of the quadratic terms of their log-likelihoods which simplifies the analysis. Our view of a log-concave coupling as a mixture distribution applicable to more complex target distributions via a forward coupling is more general.

Our method of creating the mixture is via forward coupling with a Gaussian auxiliary random variable ξ whose mean is determined by the target variable w. This has connections to proximal sampling algorithms and score based diffusion models. A proximal sampling algorithm would sample from the same joint distribution for $p(w, \xi)$ as we define here. However, the sampling method would be the Gibb's sampler alternating between sampling $p(w|\xi)$ and $p(\xi|w)$ which are both log-concave distributions [17, 32, 41, 57]. The mixing time of this sampling procedure must then be determined. If the original density of interest satisfies conditions such as being Lipschitz and having a specified Log Sobolev constant, mixing time bounds can be established for the Gibb's sampler. It remains unclear what the mixing times bounds would be for a more difficult target density such as the one we study here. We instead explicitly examine the log-concavity of the induced marginal density $p(\xi)$ and propose to sample ξ from its marginal, followed by a sample of $w|\xi$ from its conditional. We also note our use of a "cumulant generating function" (see equation 2.41) to recognize the log concavity of $p(\xi)$ has also been called the "Log-Laplace Transform" (LLT) in other work [26], which may have connections to our method of determining the log-concavity of the induced marginal density.

Score based diffusions propose starting with a random variable w' from the target density $p(w')$, and then defining the forward SDE $dw_t = -w_t dt + \sqrt{2} dB_t$. At every time t, this induces a joint distribution on $p(w', w_t)$ under which the forward conditional distribution $p(w_t|w')$ is a Gaussian distribution with mean being a linear function of w'. Paired with this forward SDE is the definition of a reverse SDE that would transport samples from a standard normal distribution to the target distribution of interest. The drift of the re-

verse diffusion is defined by the scores of the marginal distribution of the forward process $\nabla \log p(w_t)$. If these scores can be computed, the target density can be sampled from.

As is the case in our mixture model, the scores of the marginal are defined by expectations with respect to the reverse conditional $p(w'|w_t)$. For some thresholds τ_1, τ_2, for small times $t \leq \tau_1$ the reverse conditionals $p(w'|w_t)$ are log-concave and easily sampled. For large times $t \geq \tau_2$, the marginal density $p(w_t)$ is approaching a standard normal distribution and thus will become log-concave. If $\tau_2 < \tau_1$, these two regions overlap and the original density $p(w')$ can be written as a log-concave mixture of log-concave components $p(w') = \int p(w'|w_t)p(w_t)dw_t$. Thus, the entire procedure of reverse diffusion can be avoided and a one shot sample of w_t from its marginal $p(w_t)$ and a sample from the reverse conditional $p(w'|w_t)$ can computed. A variation of this idea is the core procedure we use in this work, simplifying the processes of a reverse diffusion into one specific and useful choice of joint measure with an auxiliary random variable.

2.5.2 Bayesian Neural Networks

Bayesian neural networks have been studied for many years [16, 25, 52], although specific mixing time bounds for MCMC algorithms to guarantee polynomial time complexity have been a barrier to their implementation. Recent approaches have studied the simplification of the posterior in the Neural Tangent Kernel (NTK) regime, resulting in the posterior being near the posterior associated with a Gaussian process prior [28, 30]. These approaches require $K/N \to \infty$ to achieve that simplification of the posterior density. We work with $K < N$ which is a different regime. The bounded K/N setting is shown in [28, 30] to be distinct with potentially more flexible non-Gaussian process behavior. Indeed, such flexibility arises in our model where the internal weights are adapted by the posterior.

2.5.3 Sampling the Reverse Conditional Density

The density $p(w|\xi)$ represents a weakly log-concave density over a constrained set. Note that our target density $p(w|\xi)$ only depends on the random variables w through their interaction with the data matrix $\mathbf{X}$. Thus, directions w that are orthogonal to the data matrix, that is w where $w_k \cdot x_i = 0$ for all i and k, have no interaction with the Hessian matrix of the log-likelihood. As such, the density is flat along these directions (the log-likelihood is constant for all w, w' with $x_i \cdot w_k = x_i \cdot w'_k$ for all i, k) and thus the Hessian is not negative definite but only negative semi-definite. This means $p(w|\xi)$ is not strongly log-concave, but only weakly log-concave.

There are various methods adapting unconstrained sampling algorithms to constrained spaces such as using a barrier function [56], Dikin Walks [38] and Hamiltonian Monte Carlo in a constrained space [37]. However, these may not work as well with a weakly log-concave density as we have here.

There are existing algorithms which show success for sampling weakly log-concave density on constrained spaces. It is shown in [48] that Ball Walk and Hit and Run algorithms mix in polynomial time for weakly log-concave densities on a convex set. Recent results in [39] improve upon these mixing time bounds, using a method of sampling from uniform densities over convex level sets of the log-likelihood. We also note, with different construction of the auxiliary random variable ξ then the one we have proposed in this work, it may be possible to force strict concavity in every direction of $\log p(w|\xi)$ using a normal with a different mean and covariance matrix for the forward coupling. This may have benefits for the speed of sampling.

2.5.4 Sampling the Induced Marginal Density

With d larger than the bound given in equation 2.69, we can show $p(\xi)$ to be a strongly log-concave density, which makes studying its mixing time bounds easier. However, due

to our set B restriction the ξ random variable is also forced to live in a constrained convex support set. Thus we cannot directly apply unconstrained sampling algorithms over the full state space without considering the effects of the boundary. However, we constructed our set B to be very likely under an un-constrained density for $p(\xi)$, so it is very unlikely for $p(\xi)$ to be drawn near to its boundary.

Furthermore, we do not have access directly to the log-likelihood of $p(\xi)$, but can only express it as implicitly as a convolution. However, we can write the score $\nabla \log p(\xi)$ explicitly as a expectation over the reverse conditional density $p(w|\xi)$. Thus, with the ability to sample $p(w|\xi)$, the score of $p(\xi)$ can be estimated empirically by its own MCMC algorithm as needed. Denote the function

$$h(w) = -\beta \ell_n(w) - \frac{\rho}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} (w_k \cdot x_i)^2 - Z(w), \tag{2.80}$$

which is the part of the log-likelihood of the joint density which does not depend on ξ. The marginal pdf can then be expressed as

$$\log p_n(\xi) = -\frac{\rho}{2} \|\xi\|_2^2 + \log \Big(\int p_0(w) e^{h(w)} e^{\rho \sum_{i=1}^{n} \sum_{k=1}^{K} \xi_{i,k} w_k \cdot x_i} \eta(dw) \Big) + C, \tag{2.81}$$

for some constant C which makes the density integrate to 1. Note that ξ is restricted to have support only on the set B, so there is an indicator of the set B we do not write in the expression for simplicity.

Ignoring the restriction to the boundary set B, there are a few possible ways we could got about sampling $p(\xi)$.

Firstly, we could simply use un-adjusted Langevin diffusion (ULA), which is a basic discretization of the continuous time Langevin diffusion and only needs access to the score of our target density. However, ULA is known to be biased for its target density dependent on the step size used, and does not mix as fast as other better Langevin based algorithms

[22].

We could instead use Metropolis adjusted Langevin diffusion (MALA) which includes an accept/ reject step in the iterations. This is unbiased for the target density in question, and has improved speed up over ULA [22]. However, the Metropolis accept-reject step requires computing ratios of the probabilities $p(\xi')/p(\xi)$ for various proposed ξ' points. Yet we do not have access to the $p(\xi)$ density per-say. However, assuming ξ' is near ξ we can estimate $\log p(\xi') - \log p(\xi)$ in one of two possible ways

$$\log p(\xi') - \log p(\xi) = -\frac{\rho}{2}\|\xi'\|_2^2 + \log\Big(\int p_0(w)e^{h(w)}e^{\rho\sum_{i=1}^n\sum_{k=1}^K \xi'_{i,k}w_k\cdot x_i}\eta(dw)\Big) \tag{2.82}$$

$$+\frac{\rho}{2}\|\xi\|_2^2 - \log\Big(\int p_0(w)e^{h(w)}e^{\rho\sum_{i=1}^n\sum_{k=1}^K \xi_{i,k}w_k\cdot x_i}\eta(dw)\Big) \tag{2.83}$$

$$= -\frac{\rho}{2}(\|\xi'\|_2^2 - \|\xi\|_2^2) + \log E[e^{\rho\sum_{i=1}^n\sum_{k=1}^K(\xi'_{i,k}-\xi_{i,k})w_k\cdot x_i}|\xi]. \tag{2.84}$$

With empirical samples from $p(w|\xi)$ we can estimate this expectation and approximate the acceptance probability of a metropolis correction step. One should be wary of sample averages of exponential of sums of nK terms, as they could have large variance. Fortunately if the size Δ of the proposed differences $\xi' - \xi$ are arranged to be sufficiently small that $\rho nK\Delta$ is bounded, then such sampling will be accurate. We may also approximate

$$\log p(\xi') - \log p(\xi) \approx (\nabla \log p(\xi)) \cdot (\xi' - \xi) \tag{2.85}$$

and the score may be computed. Thus, with access to the ability to sample from the conditional density $p(w|\xi)$ it may be possible to approximate acceptance probabilities and make use of Metropolis adjusted sampling algorithms that are more accurate for the target density and can show speed up in the number of iterations required.

2.6 Appendix: Proofs of Additional Lemmas

Here we present proofs of lemmas that were too long or tedious to present in the main body of the chapter.

2.6.1 Proofs for Near Constancy of $Z(w)$

In this section, we show the restriction of ξ to the set B is a highly likely event under the base Gaussian distribution, and $Z(w)$ has small magnitude first and second derivatives.

Proof of Lemma 2.1:

Proof. We show that the set B is likely for conditionally independent Gaussian distributions for each variable. This proof follows from standard Gaussian complexity arguments.

The object we must bound is $P(\xi \in B|w)$. If the $\xi_{i,k}$ given w are independent Normal$(x_i \cdot w_k, 1/\rho)$ we may arrange a representation using independent standard normals Z_k of dimension n,

$$\xi_k = \mathbf{X}w_k + \frac{1}{\sqrt{\rho}}Z_k. \tag{2.86}$$

Each mean $x_i \cdot w_k$ is in $[-1, 1]$ due to the weight vector having bounded ℓ_1 norm and the data entries having bounded value. Consider the complement of the event we want to study, we wish for this event to have probability less than δ.

$$P(\max_{j,k} |\sum_{i=1}^{n} x_{i,j}\xi_{i,k}| \geq n + \sqrt{2\log\frac{2Kd}{\delta}}\sqrt{\frac{n}{\rho}}), \tag{2.87}$$

where P is the probability using the normal distribution of ξ given w. The max is upper

bound by

$$\max_{j,k} |\sum_{i=1}^{n} x_{i,j}\xi_{i,k}| \leq n + \max_{j,k} |\frac{1}{\sqrt{\rho}} \sum_{i=1}^{n} x_{i,j} Z_{i,k}|. \tag{2.88}$$

Thus we can bound the larger probability event uniformly for $w \in (S_1^d)^K$,

$$P(\max_{j,k} \frac{|\sum_{i=1}^{n} x_{i,j} Z_{i,k}|}{\sqrt{n}} \geq \sqrt{2 \log \frac{2Kd}{\delta}}) \leq \frac{\delta}{\sqrt{2 \log(2Kd/\delta)}}. \tag{2.89}$$

Where the conclusion follows from a union bound and Gaussian tail bound.

□

Proof of Lemma 2.2:

Proof. We provide upper bounds on the magnitude of the first and second derivatives of the function $Z(w)$ as defined in equation (2.20). Denote Φ as the normal CDF and φ as the normal pdf. Throughout the proof recall that $p(w|\xi)$ treats each $\xi_{i,k}$ as independent normal with $\xi_{i,k} \sim \text{Normal}(x_i \cdot w_k, \frac{1}{\rho})$ conditionally independent given w. The gradient of $Z(w)$ inner product with a vector u with blocks u_k is

$$\left|u \cdot \nabla_w Z(w)\right| = \left|\rho E[\sum_{i=1}^{n} \sum_{k=1}^{K} (u_k \cdot x_i)(\xi_{i,k} - x_i \cdot w_k) \frac{1_B(\xi)}{P(\xi \in B|w)} |w]\right|. \tag{2.90}$$

By Lemma 2.1, the set B has probability at least $1 - \delta/\sqrt{2 \log(2Kd/\delta)}$. We note the following upper and lower bounds on the Gaussian CDF provided by the classical results of Gordon [27], we have bounds on the Gaussian CDF

$$\frac{\varphi(x)}{x + \frac{1}{x}} \leq 1 - \Phi(x) \leq \frac{\varphi(x)}{x}. \tag{2.91}$$

Consider then the value

$$\delta^* = \Phi(-\sqrt{2\log(1/\delta)}). \tag{2.92}$$

For our problem, $Kd \geq 2$ by construction. Then for all positive $\delta \leq 1/e$, it can be shown that δ^* is larger than the term which defines the probability of our set B,

$$\frac{\delta}{\sqrt{2\log(2Kd/\delta)}} \leq \delta^*. \tag{2.93}$$

Then consider the collections of all measurable sets $D \subset \mathbb{R}^{NK}$ such that $P(\xi \in D) \geq 1 - \delta^*$. This collection contains our original set B as an object in the class. Then, the absolute value of the expected inner product in (2.90) is less than the maximum for any set D in this class,

$$\max_{\substack{D: \\ P(\xi \in D|w) \geq 1-\delta}} \rho \frac{|E[\sum_{i=1}^{n} \sum_{k=1}^{K} (u_k \cdot x_i)(\xi_{i,k} - x_i \cdot w_k) 1_D(\xi)|w]|}{1-\delta}. \tag{2.94}$$

Define the value

$$\tilde{\sigma} = \sqrt{\frac{\sum_{i=1}^{n} \sum_{k=1}^{K} (u_k \cdot x_i)^2}{\rho}}. \tag{2.95}$$

Under the normal distribution for ξ, the integrand in question is a scalar mean 0 normal random variable with this variance,

$$\sum_{i=1}^{n} \sum_{k=1}^{K} (u_k \cdot x_i)(\xi_{i,k} - x_i \cdot w_k) \sim \text{Normal}(0, \tilde{\sigma}^2). \tag{2.96}$$

The set D which maximizes expression (2.94) is then the set which controls the size of this integrand,

$$D^* = \{\xi : \frac{\sum_{i=1}^{n}\sum_{k=1}^{K}(u_k \cdot x_i)(\xi_{i,k} - x_i \cdot w_k)}{\tilde{\sigma}} \leq \tau\}, \tag{2.97}$$

for some choice of τ. We can also equally consider the set D^* where the object in the expression being more than some negative τ, due to symmetry. The proper choice of τ is $\sqrt{2\log(1/\delta)}$.We then have upper bound

$$|u \cdot \nabla_w Z(w)| \leq \frac{\rho\tilde{\sigma}}{1-\delta}\Big|\int_{-\infty}^{\sqrt{2\log(1/\delta)}} z\varphi(z)dz\Big| = \frac{\rho\tilde{\sigma}\delta}{\sqrt{2\pi}1-\delta}, \tag{2.98}$$

using the fact that $-z\varphi(z) = \varphi'(z)$ and fundamental theorem of calculus. This yields an upper bound on our expression of interest,

$$|u \cdot \nabla_w Z(w)| \leq \frac{\rho\tilde{\sigma}}{1-\delta}\frac{\delta}{\sqrt{2\pi}}. \tag{2.99}$$

Which notably goes to 0 as $\delta \to 0$.

The Hessian is then a difference in variances,

$$u^{\mathrm{T}}[\nabla^2 Z(w)]u = -\rho\sum_{i=1}^{n}\sum_{k=1}^{K}(x_i \cdot u_k)^2 \tag{2.100}$$

$$+\rho^2\mathrm{Var}[\sum_{i=1}^{n}\sum_{k=1}^{K}(x_i \cdot u_k)\xi_{i,k}|w, B]. \tag{2.101}$$

Note that $\xi_{i,k}$ is independent normal with variance $1/\rho$, so if we did not constrain the set B, expressions (2.100) and (2.101) would cancel to 0. That is, (2.100) is the variance of the linear function of ξ given w if we did not condition on the set B, and (2.101) is the variance conditioned on the set B.

Note that the object whose variance we are taking in (2.101) is a linear function of

ξ, and ξ is a normal random variable given w with diagonal covariance matrix $\frac{1}{\rho}$. By an application of a Brascamp-Lieb inequality, see for example [14, Proposition 2.1], we would have an upper bound on this variance by the norm of this linear vector divided by ρ, which times ρ^2 is exactly expression (2.100). Thus, the term (2.101) is less than or equal to the absolute value of term (2.100) so an upper bound on the quadratic form is 0, that is $u^{\mathrm{T}}[\nabla^2 Z(w)]u \leq 0$.

We then compute a lower bound on the variance term in (2.101). Note a Cramer-Rao lower bound is not applicable here since restriction to a compact set makes integration by parts inapplicable due to boundary conditions. In particular, the expectation of the score of a constrained distribution is not always 0.

Using a bias-variance decomposition, we can write the variance as a non-centered expected squared difference minus a bias correction,

$$\mathrm{Var}[\sum_{i=1}^{n}\sum_{k=1}^{K}(x_i \cdot u_k)\xi_{i,k}|w, B] \tag{2.102}$$

$$= E[\Big(\sum_{i=1}^{n}\sum_{k=1}^{K}(x_i \cdot u_k)(\xi_{i,k} - x_i \cdot w_k)\Big)^2|w, \xi \in B] \tag{2.103}$$

$$- \Big(\sum_{i=1}^{n}\sum_{k=1}^{K}(u_k \cdot x_i)\Big(E[\xi_{i,k}|w, \xi \in B] - x_i \cdot w_k\Big)\Big)^2 \tag{2.104}$$

$$\geq E[\Big(\sum_{i=1}^{n}\sum_{k=1}^{K}(x_i \cdot u_k)(\xi_{i,k} - x_i \cdot w_k)\Big)^2 \frac{1_B(\xi)}{P(\xi \in B|w)}|w] \tag{2.105}$$

$$- \frac{\rho^2\tilde{\sigma}^2}{(1-\delta)^2}\frac{\delta^2}{2\pi}, \tag{2.106}$$

where we have applied the previously derived bound on the score to expression (2.104) to deduce expression (2.106), which is the square of the previous bond.

If we did not condition on the set B, the expression (2.105) would be the variance of a simple normal variable with variance $\tilde{\sigma}^2$. We will show restricting to B still results in a value very close to $\tilde{\sigma}^2$.

The set B has probability at least $1 - \delta/\sqrt{2\log(2Kd/\delta)}$. Define the value

$$\delta^{**} = 2\Phi(-\sqrt{2\log(1/\delta)}). \tag{2.107}$$

If $Kd \geq 4$, for all positive $\delta \leq 1/16$ we have that δ^{**} is larger than the term which defines the set B probability,

$$\frac{\delta}{\sqrt{2\log(2Kd/\delta)}} \leq \delta^{**}. \tag{2.108}$$

Then, the expected value of the variable in question restricted to B is lower bound by the minimum for any set D with $P(\xi \in D) \geq 1 - \delta^{**}$,

$$E[\Big(\sum_{i=1}^{n}\sum_{k=1}^{K}(x_i \cdot u_k)(\xi_{i,k} - x_i \cdot w_k)\Big)^2 \frac{1_B(\xi)}{P(\xi \in B|w)}|w] \tag{2.109}$$

$$\geq \min_{\substack{D: \\ P(\xi \in D|w) \geq 1-\delta^{**}}} \frac{E[\Big(\sum_{i=1}^{n}\sum_{k=1}^{K}(x_i \cdot u_k)(\xi_{i,k} - x_i \cdot w_k)\Big)^2 1_D(\xi)|w]}{1-\delta}. \tag{2.110}$$

The integrand in question, as before, is the same normal variable now squared. The minimizing set D^* is then the set placing an upper bound on that expression,

$$D^* = \{\xi : -\tau \leq \frac{\sum_{i=1}^{n}\sum_{k=1}^{K}(x_i \cdot u_k)(\xi_{i,k} - x_i \cdot w_k)}{\tilde{\sigma}} \leq \tau\}, \tag{2.111}$$

for some value τ, the proper choice being $\tau = \sqrt{2\log(1/\delta)}$.

Note this set D^* can be deduced from the Neyman-Pearson Lemma [43, Theorem 3.2.1], comparing the distribution where each $\xi_{i,k}$ is independent normal with mean $x_i \cdot w_k$ and variance $\frac{1}{\rho}$, to the distribution which has this normal density times $(\sum_{i=1}^{n}\sum_{k=1}^{K}(x_i \cdot u_k)(\xi_{i,k} - x_i \cdot w_k))^2$. (Likewise, the previous D^* in (2.97) can be deduced by a generalization of the Neyman-Pearson lemma in which the alternative is a signed measure measure with the normal density times the factor $\sum_{i=1}^{n}\sum_{k=1}^{K}(x_i \cdot u_k)(\xi_{i,k} - x_i \cdot w_k)$).

We are then integrating a squared normal on a truncated range and have lower bound,

$$\min_{\substack{D:\\ P(\xi\in D|w)\geq 1-\delta}} \frac{E[\Big(\sum_{i=1}^{n}\sum_{k=1}^{K}(x_i\cdot u_k)(\xi_{i,k}-x_i\cdot w_k)\Big)^2 1_D(\xi)|w]}{1-\delta} \tag{2.112}$$

$$=\frac{\tilde{\sigma}^2}{1-\delta}\int_{-\sqrt{2\log(1/\delta)}}^{\sqrt{2\log(1/\delta)}} z^2\varphi(z)dz. \tag{2.113}$$

To evaluate this integral use its complement set and symmetry of the normal pdf,

$$\int_{-\sqrt{2\log(1/\delta)}}^{\sqrt{2\log(1/\delta)}} z^2\varphi(z)dz = 1-2\int_{-\infty}^{-\sqrt{2\log(1/\delta)}} z^2\varphi(z)dz. \tag{2.114}$$

Then apply integration by parts,

$$-\int_{-\infty}^{-\sqrt{2\log(1/\delta)}} z^2\varphi(z)dz = z\varphi(z)|_{-\infty}^{-\sqrt{2\log(1/\delta)}} - \Phi(-\sqrt{2\log(1/\delta)}). \tag{2.115}$$

This gives a lower bound for the expression in (2.113)

$$\frac{\tilde{\sigma}^2}{1-\delta}\Big(1-\frac{2\delta}{\sqrt{2\pi}}(\sqrt{2\log(1/\delta)}+\frac{1}{\sqrt{2\log(1/\delta)}})\Big), \tag{2.116}$$

which converges to $\tilde{\sigma}^2$ as $\delta\to 0$. We then combine expressions (2.101), (2.106), and (2.116) to give a lower bound on Hessian quadratic form,

$$u^{\mathrm{T}}[\nabla^2 Z(w)]u \geq -\rho^2\tilde{\sigma}^2+\rho^2\tilde{\sigma}^2(\frac{1}{1-\delta}-\frac{2\delta}{(1-\delta)\sqrt{2\pi}}(\sqrt{2\log(1/\delta)}+\frac{1}{\sqrt{2\log(1/\delta)}})) \tag{2.117}$$

$$-\frac{\rho^4\tilde{\sigma}^2}{(1-\delta)^2}\frac{\delta^2}{2\pi} \tag{2.118}$$

$$=-\frac{\rho^2\tilde{\sigma}^2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\Big(-\sqrt{2\pi}+2\sqrt{2\log(1/\delta)}(1+\frac{1}{2\log(1/\delta)})+\frac{\rho^2\tilde{\sigma}^2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\Big) \tag{2.119}$$

$$\geq -\frac{\rho^2\tilde{\sigma}^2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\Big(2\sqrt{2\log(1/\delta)}+\frac{\rho^2\tilde{\sigma}^2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\Big) \tag{2.120}$$

which converges to 0 as $\delta \to 0$. $\square$

2.6.2 Log-Concavity of $p_n^*(w|\xi)$ with Conditioning on the Set B

In this section, we show the conditioning of ξ given w to the set B does not affect the log-concavity of the reverse conditional much.

Proof of Theorem 2.2:

Proof. We prove the reverse conditional is log-concave when restricting ξ to live in the set B. This proof follows much the same way as Theorem 2.1. The log-likelihood for $p_n^*(w|\xi)$ is given by

$$\log p_n^*(w|\xi) = -\beta \ell_n(w) + H(\xi) \tag{2.121}$$

$$- \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{\rho}{2} (\xi_{i,k} - w_k \cdot x_i)^2 \tag{2.122}$$

$$- Z(w), \tag{2.123}$$

for some function $H(\xi)$ which does not depend on w and is only required to make the density integrate to 1. The term (2.122) is a negative quadratic in w which treats each w_k as if it were an independent normal random variable. Thus, the additional Hessian contribution will be a $(Kd) \times (Kd)$ negative definite block diagonal matrix with $d \times d$ blocks of the form $\rho \sum_{i=1}^{n} x_i x_i^{\mathrm{T}}$. Denote the Hessian as $H_n(w|\xi) \equiv \nabla^2 \log p_n^*(w|\xi)$. For any vector $u \in \mathbb{R}^{Kd}$, with blocks $u_k \in \mathbb{R}^d$, the quadratic form $u^{\mathrm{T}} H_n(w|\xi) u$ can be

expressed as

$$-\beta\sum_{i=1}^{n}\Big(\sum_{k=1}^{K}\psi'(w_k\cdot x_i)u_k\cdot x_i\Big)^2 \tag{2.124}$$

$$+\sum_{k=1}^{K}\sum_{i=1}^{n}(u_k\cdot x_i)^2\Big[\beta\text{res}_i(w)c_k\psi''(w_k\cdot x_i)-\rho)\Big] \tag{2.125}$$

$$+u^{\mathrm{T}}(\nabla^2 Z(w))u. \tag{2.126}$$

By the assumptions on the second derivative of ψ and the definition of ρ in equation (2.16) we have

$$\max_{i,k}(\beta\text{res}_i(w)c_k\psi''(w_k\cdot x_i)-\rho)\leq-(\sqrt{\frac{3}{2}}-1)a_2\frac{\beta C_n V}{K}, \tag{2.127}$$

so all the terms in the sum in (2.125) are negative. Recall the definition of $\tilde{\sigma}^2$,

$$\tilde{\sigma}^2=\frac{\sum_{k=1}^{K}\sum_{i=1}^{n}(u_k\cdot x_i)^2}{\rho}. \tag{2.128}$$

Therefore, expression (2.125) is less than

$$-(\sqrt{\frac{3}{2}}-1)\sqrt{\frac{3}{2}}\Big(a_2\frac{\beta C_n V}{K}\Big)^2\tilde{\sigma}^2. \tag{2.129}$$

By Lemma 2.2, the largest the Hessian term from the correction function Z can be is

$$u^{\mathrm{T}}(\nabla^2 Z(w))u\leq\frac{\rho^2\tilde{\sigma}^2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\Big(2\sqrt{2\log(1/\delta)}+\frac{\rho^2\tilde{\sigma}^2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\Big). \tag{2.130}$$

Thus term (2.125) plus (2.126) is less than

$$-\tilde{\sigma}^2\Big(a_2\frac{\beta C_n V}{K}\Big)^2\Big(\sqrt{\frac{3}{2}}-1\Big)\Big(\sqrt{\frac{3}{2}}\Big) \tag{2.131}$$

$$+\tilde{\sigma}^2\Big(a_2\frac{\beta C_n V}{K}\Big)^2\Big(\sqrt{\frac{3}{2}}\Big)^2\frac{2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\sqrt{2\log\frac{\delta}{2}} \tag{2.132}$$

$$+\tilde{\sigma}^2\Big(a_2\frac{\beta C_n V}{K}\Big)^4\Big(\sqrt{\frac{3}{2}}\Big)^4\frac{1}{2\pi}\frac{\delta^2}{(1-\delta)^2}. \tag{2.133}$$

Recall the definitions of H_1 and H_2 in the theorem statement,

$$H_1(\delta) = \frac{2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\sqrt{2\log\frac{\delta}{2}} \tag{2.134}$$

$$H_2(\delta) = \Big(a_2\frac{\beta C_n V}{K}\Big)^2\frac{1}{2\pi}\frac{\delta^2}{(1-\delta)^2}. \tag{2.135}$$

Simplifying expressions (2.131) to (2.133) by dividing out common terms, to have a negative expression for the Hessian we require yields,

$$\sqrt{\frac{3}{2}}(-1+H_1(\delta)) + \Big(\sqrt{\frac{3}{2}}\Big)^3 H_2(\delta) \leq -1. \tag{2.136}$$

By the assumptions $H_1(\delta) \leq \frac{1}{100}$, and $H_2(\delta) \leq \frac{1}{10}$. Under these conditions, the inequality is satisfied

$$\sqrt{\frac{3}{2}}(-1+H_1(\delta)) + \Big(\sqrt{\frac{3}{2}}\Big)^3 H_2(\delta) \leq \sqrt{\frac{3}{2}}(-\frac{99}{100}) + \Big(\sqrt{\frac{3}{2}}\Big)^3\frac{1}{10} \tag{2.137}$$

$$= -\frac{21}{25}\sqrt{\frac{3}{2}} < -1. \tag{2.138}$$

□

2.6.3 Hölder Inequality Proofs

In this section, we bound the two terms in the Hölder inequality. First, we need a supporting lemma.

Lemma 2.7. *For any vector $x \in [-1,1]^d$ and any integer $\ell > 0$, the expected inner product with random vector w from the continuous uniform distribution on S_1^d raised to the power 2ℓ is upper bound by,*

$$E_{P_0}[(\sum_{j=1}^{d} x_j w_j)^{2\ell}] \leq \frac{1}{(d)^\ell} \frac{(2\ell)!}{\ell!}. \tag{2.139}$$

Proof. The sum $\sum_{j=1}^{d} x_j w_j$ raised to the power 2ℓ can be expressed as sum using a multi-index $J = (j_1, \ldots, j_{2\ell})$ where each $j_i \in \{1, \ldots, d\}$ and there are $d^{2\ell}$ terms,

$$E[\Big(\sum_{j=1}^{d} x_j w_j\Big)^{2\ell}] = \sum_{j_1,\ldots,j_{2\ell}} \prod_{i=1}^{2\ell} (x_{j_i}) E[\prod_{i=1}^{2\ell} w_{j_i}]. \tag{2.140}$$

For a given multi-index vector J, let $r(j, J)$ count the number of occurrences of the value j in the vector, $r(j, J) = \sum_{i=1}^{2\ell} 1\{j_i = j\}$. Then for any multi-index we would have,

$$\prod_{i=1}^{2\ell} w^{j_i} = \prod_{j=1}^{d} w_j^{r(j,J)}. \tag{2.141}$$

Abbreviate $r_j = r(j, J)$ for a fixed vector J also note $\sum_{j=1}^{d} r_j = 2\ell$. Consider the expectation $E[\prod_{i=1}^{d} w_j^{r_j}]$. Due to the symmetry of the prior, if any of the r_j are odd then the whole expectation is 0. Thus, we only consider vectors $\vec{r} = (r_1, \ldots, r_d)$ where all entries are even. If we fix the signs of the w_j points to live in a given orthant, then the distribution is uniform on the $d+1$ dimensional simplex. Define $w_{d+1} = 1 - \sum_{j=1}^{d} |w_j|$ then $(|w_1|, \ldots, |w_d|, w_{d+1})$ has a symmetric Dirichlet $(1, \ldots, 1)$ distribution in $d+1$ dimensions. Note a general Dirichlet distribution in $d+1$ dimensions with parameter vector

$\vec{\alpha} = (\alpha_1, \ldots, \alpha_{d+1})$ has a properly normalized density as

$$p_{\vec{\alpha}}(w_1, \ldots, w_d) = \frac{\Gamma(\sum_{j=1}^{d} \alpha_j)}{\prod_{j=1}^{d+1} \Gamma(\alpha_j)} \prod_{j=1}^{d} (w_j)^{\alpha_j - 1} (1 - \sum_{j=1}^{d} w_j)^{\alpha_{d+1} - 1}. \tag{2.142}$$

Thus the expectation of $\prod_{j=1}^{d} w_j^{r_j}$ with respect to a symmetric Dirichlet has the form of an un-normalized Dir$(r_1 + 1, \ldots, r_d + 1, 1)$ distribution. Thus, the expectation is a ratio of their normalizing constants,

$$E[\prod_{j=1}^{d} w_j^{r_j}] = \frac{\Gamma(d+1) \prod_{j=1}^{d} \Gamma(r_j + 1)}{\Gamma(d + 1 + \sum_{j=1}^{d} r_j)} \tag{2.143}$$

$$= \frac{d! \prod_{j=1}^{d} r_j!}{(d + 2\ell)!}. \tag{2.144}$$

The number of times a specific vector $\vec{r}$ appears from the multi-index J is $\frac{(2\ell)!}{\prod_{j=1}^{d} r_j!}$ thus we have,

$$E[(\sum_{j=1}^{d} x_j w_j)^{2\ell}] = \sum_{\substack{\vec{r} \text{ even} \\ \sum_j r_j = 2\ell}} \prod_{j=1}^{d} (x_j)^{r_j} \frac{(2\ell)!}{\prod_{j=1}^{d} r_j!} E[\prod_{j=1}^{d} w_j^{r_j}] \tag{2.145}$$

$$= \frac{(2\ell!)(d!)}{(d + 2\ell)!} \sum_{\substack{\vec{r} \text{ even} \\ \sum_j r_j = 2\ell}} \prod_{j=1}^{d} (x_j)^{r_j} \tag{2.146}$$

$$= \frac{(2\ell!)(d!)}{(d + 2\ell)!} \sum_{\substack{\vec{r} \text{ even} \\ \sum_j r_j = 2\ell}} \prod_{j=1}^{d} (x_j^2)^{\frac{r_j}{2}} \tag{2.147}$$

$$\leq \frac{(2\ell!)(d!)}{(d + 2\ell)!} \frac{(d + \ell - 1)!}{\ell!(d-1)!} \tag{2.148}$$

$$= \frac{(d + \ell - 1) \cdots (d)}{(d + 2\ell) \cdots (d + 1)} \frac{(2\ell)!}{(\ell)!} \tag{2.149}$$

$$\leq \frac{1}{d^\ell} \frac{2\ell!}{\ell!}, \tag{2.150}$$

where inequality (2.148) follows from each $x_j^2 \leq 1$ thus each term in the sum is less than

1 and there being $\binom{d+\ell-1}{\ell}$ terms in the sum.

□

Proof of Lemma 2.5:

Proof. We bound the first term in the Hölder inequality depending on the higher order moments of the prior. We have unit vector $u \in \mathbb{R}^{nK}$ with n dimensional blocks u_k. Define vectors in $\mathbb{R}^d$ as $v_k = \mathbf{X}^{\mathrm{T}} u_k$ and the object we study is

$$E[(\sum_{k=1}^{K} v_k \cdot w_k)^{2\ell}]. \tag{2.151}$$

Use a multinomial expansion of this power of a sum and we have expression,

$$E[\sum_{\substack{j_1,\ldots,j_K \\ \sum j_k = 2\ell}} \binom{2\ell}{j_1,\ldots,j_K} \prod_{k=1}^{K} (v_k \cdot w_k)^{j_k}] = \sum_{\substack{j_1,\ldots,j_K \\ \sum j_k = 2\ell}} \binom{2\ell}{j_1,\ldots,j_K} \prod_{k=1}^{K} E[(v_k \cdot w_k)^{j_k}], \tag{2.152}$$

since the prior treats each neuron weigh vector w_k as independent and uniform on S_1^d. By the symmetry of the prior, if any j_k are odd the whole expression is 0 thus we only sum using even j_k values,

$$\sum_{\substack{j_1,\ldots,j_K \\ \sum j_k = \ell}} \binom{2\ell}{2j_1,\ldots,2j_K} \prod_{k=1}^{K} E[(v_k \cdot w_k)^{2j_k}]. \tag{2.153}$$

Each vector v_k is a linear combination of the rows of the data matrix,

$$v_k = \sum_{i=1}^{n} u_{k,i} x_i. \tag{2.154}$$

Define $s_{k,i} = \text{sign}(u_{k,i})$ and $\alpha_{k,i} = \frac{|u_{k,i}|}{\|u_k\|_1}$. We can then interpret the above inner product as a scaled expectation on the data indexes,

$$v_k \cdot w_k = (\|u_k\|_1) \sum_{i=1}^{n} \alpha_{k,i} s_{k,i}\, x_i \cdot w_k. \tag{2.155}$$

The average is then less than the maximum term in index i,

$$E[(v_k \cdot w_k)^{2j_k}] = (\|u_k\|_1)^{2j_k} E[\Big(\sum_{i=1}^{n} \alpha_{k,i} s_{k,i}\, x_i \cdot w_k \Big)^{2j_k}] \tag{2.156}$$

$$\leq (\|u_k\|_1)^{2j_k} \sum_{i=1}^{n} \alpha_{k,i} E[\Big(x_i \cdot w_k \Big)^{2j_k}] \tag{2.157}$$

$$\leq (\|u_k\|_1)^{2j_k} \max_i E[(x_i \cdot w_k)^{2j_k}] \tag{2.158}$$

$$\leq (\|u_k\|_1)^{2j_k} \frac{1}{(d)^{j_k}} \frac{(2j_k)!}{j_k!}, \tag{2.159}$$

where we have applied Lemma 2.7. We then plug this result into equation (2.153),

$$\frac{1}{d^\ell} \frac{(2\ell!)}{\ell!} \Big(\sum_{\substack{j_1,\ldots,j_K \\ \sum j_k = \ell}} \binom{\ell}{j_1, \ldots, j_K} \prod_{k=1}^{K} (\|u_k\|_1)^{2j_k} \Big) = \frac{1}{d^\ell} \frac{(2\ell)!}{\ell!} \Big(\sum_{k=1}^{K} \|u_k\|_1^2 \Big)^\ell. \tag{2.160}$$

For each sub block u_k of dimension n we have $\|u_k\|_1^2 \leq n\|u_k\|_2^2$ and $\|u\|^2 = \sum_{k=1}^{K} \|u_k\|^2 = 1$ is a unit vector which gives upper bound

$$\frac{n^\ell (2\ell)!}{d^\ell \ell!}. \tag{2.161}$$

Via Stirling's bound [55],

$$\sqrt{2\pi\ell}(\frac{\ell}{e})^\ell e^{\frac{1}{12\ell+1}} \leq \ell! \leq \sqrt{2\pi\ell}(\frac{\ell}{e})^\ell e^{\frac{1}{12\ell}}. \tag{2.162}$$

Taking the ℓ root we have

$$\left(\frac{n^\ell}{d^\ell}\frac{(2\ell)!}{\ell!}\right)^{\frac{1}{\ell}} \leq \frac{n}{d}\left(2^{2\ell+\frac{1}{2}}(\frac{\ell}{e})^\ell e^{\frac{1}{24\ell}-\frac{1}{12\ell+1}}\right)^{\frac{1}{\ell}} \tag{2.163}$$

$$= \frac{2^{2+\frac{1}{2\ell}}n\ell}{d}e^{\frac{1}{24\ell^2}-\frac{1}{12\ell^2+\ell}-1} \tag{2.164}$$

$$\leq \frac{4n\ell}{d}\sqrt{2}e^{\frac{1}{24}+\frac{1}{13}-1} \tag{2.165}$$

$$\leq \frac{4n\ell}{\sqrt{e}d}. \tag{2.166}$$

□

of Lemma 2.6:

Proof. We bound the second term in the Hölder inequality determined by the growth rate of the cumulant generating function. By the mean value theorem, there exists some value $\tilde{\tau} \in [1, \frac{\ell}{\ell-1}]$ such that

$$\Gamma_\xi^n(\frac{\ell}{\ell-1}) = \Gamma_\xi^n(1) + (\Gamma_\xi^n)'(\tilde{\tau})[\frac{\ell}{\ell-1} - 1]. \tag{2.167}$$

Rearranging, we can express the difference

$$\frac{\ell-1}{\ell}\Gamma_\xi^n(\frac{\ell}{\ell-1}) - \Gamma_\xi^n(1) = (\Gamma_\xi^n)'(\tilde{\tau})\frac{1}{\ell} - \frac{1}{\ell}\Gamma_\xi^n(1). \tag{2.168}$$

By construction, $\Gamma_\xi^n(\tau)$ is an increasing convex function with $\Gamma_\xi^n(0) = 0$. Thus $\Gamma_\xi^n(1) > 0$ and we can study the upper bound

$$\frac{\ell-1}{\ell}\Gamma_\xi^n(\frac{\ell}{\ell-1}) - \Gamma_\xi^n(1) \leq (\Gamma_\xi^n)'(\tilde{\tau})\frac{1}{\ell}. \tag{2.169}$$

Recall $\Gamma_\xi^n(\tau)$ defined in equation (2.54) is a cumulant generating function of $\tilde{h}_\xi^n(w)$. Thus, its derivative at $\tilde{\tau}$ is the mean of $\tilde{h}_\xi^n(w)$ under the tilted distribution. The mean is then less

than the maximum difference of any two points on the constrained support set,

$$(\Gamma_\xi^n)'(\tilde{\tau}) = E_{\tilde{\tau}}[\tilde{h}_\xi^n(w)|\xi] \leq \max_{w,w_0 \in (S_1^d)^K} (\tilde{h}_\xi^n(w) - \tilde{h}_\xi^n(w_0)). \tag{2.170}$$

By the mean value theorem, for any choice of $w, w_0 \in (S_1^d)^K$ there exists a $\tilde{w} \in (S_1^d)^K$ along the line between w and w_0 such that

$$\tilde{h}_\xi^n(w) - \tilde{h}_\xi^n(w_0) = \nabla_w \tilde{h}_\xi^n(\tilde{w}) \cdot (w - w_0). \tag{2.171}$$

For each k, the gradient in w_k is

$$\nabla_{w_k} \tilde{h}_\xi^n(\tilde{w}) = \beta \sum_{i=1}^n (\text{res}_i(\tilde{w}) c_k \psi'(w_k \cdot x_i) - a_2 \sqrt{\frac{3}{2}} \frac{C_n V}{K} [w_k \cdot x_i]) x_i \tag{2.172}$$

$$+ a_2 \sqrt{\frac{3}{2}} \frac{\beta C_n V}{K} \sum_{i=1}^n \xi_{i,k} x_i + \nabla_{w_k} Z(w). \tag{2.173}$$

The terms in the sum in (2.172) satisfy

$$|\text{res}_i(\tilde{w}) c_k \psi'(w_k \cdot x_i) - a_2 \sqrt{\frac{3}{2}} \frac{C_n V}{K} [w_k \cdot x_i]| \leq (a_1 + a_2 \sqrt{\frac{3}{2}}) \frac{C_n V}{K}, \tag{2.174}$$

for each i. The vector $w_k - w_{0,k}$ satisfies $\|w_k - w_{0,k}\|_1 \leq 2$. Since each x_i vector has bounded entries between -1 and 1, the inner product with the first term is bounded as

$$\Big[\beta \sum_{i=1}^n (\text{res}_i(\tilde{w}) c_k \psi'(w_k \cdot x_i) - \frac{C_n V}{K}) x_i\Big] \cdot (w_k - w_{0,k}) \leq 2\Big(a_1 + a_2 \sqrt{\frac{3}{2}}\Big) \frac{C_n V \beta n}{K}. \tag{2.175}$$

As for the second term,

$$\Big[\sum_{i=1}^n \xi_{i,k} x_i\Big] \cdot (w_k - w_{0,_k}) \leq 2 \max_j |\sum_{i=1}^n \xi_{i,k} x_{i,j}|. \tag{2.176}$$

Our original restriction of ξ to the set B is specifically designed to control this term. By definition of the set B, for all k,

$$\max_j |\sum_{i=1}^n \xi_{i,k} x_{i,j}| \leq n + \sqrt{2\log(\frac{2Kd}{\delta})}\sqrt{\frac{n}{\rho}} \tag{2.177}$$

$$= n + \sqrt{2\log\frac{2Kd}{\delta}}\sqrt{\sqrt{\frac{2}{3}}\frac{nK}{a_2\beta C_n V}}. \tag{2.178}$$

For the final term, $Z(w)$ is shown to have small derivative. By Lemma 2.2,

$$\sum_k \nabla_{w_k} Z(w)\cdot(w_k - w_{0,k}) \leq \sqrt{\rho}\sqrt{\sum_{i=1}^n\sum_{k=1}^K((w_k - w_{0,k})\cdot x_i)^2}\frac{1}{(1-\delta)}\frac{\delta}{\sqrt{2\pi}} \tag{2.179}$$

$$\leq \sqrt{4a_2\sqrt{\frac{3}{2}}C_n V\beta n}\frac{\delta}{\sqrt{2\pi}}\frac{1}{1-\delta}. \tag{2.180}$$

Summing using index k for terms (2.175), (2.178) and combining with term (2.180), we can upper bound the difference in the CGF as,

$$2\Big(a_1 + a_2\sqrt{\frac{3}{2}}\Big)\frac{C_n V\beta n}{\ell} + 2a_2\sqrt{\frac{3}{2}}\frac{\beta C_n V}{\ell}\Big(n + \sqrt{2\log\frac{2Kd}{\delta}}\sqrt{\sqrt{\frac{2}{3}}\frac{nK}{a_2\beta C_n V}}\Big) \tag{2.181}$$

$$+2\sqrt{a_2\sqrt{\frac{3}{2}}C_n V\beta n}\frac{\delta}{\sqrt{2\pi}}\frac{1}{1-\delta} \tag{2.182}$$

$$=\frac{C_n V\beta n}{\ell}(2a_1 + 4a_2\sqrt{\frac{3}{2}}) + \frac{\sqrt{C_n V\beta n}}{\ell}\sqrt{a_2\sqrt{\frac{3}{2}}}\Big(2\sqrt{2\log\frac{2Kd}{\delta}}\sqrt{K} + \sqrt{2}\frac{\delta}{\sqrt{\pi}(1-\delta)}\Big). \tag{2.183}$$

By assumption $d \geq 2, K \geq 2, \delta \leq \frac{1}{16}$. For all values $0 < z \leq \frac{1}{2}$ we have the inequality

$$\frac{z}{(1-z)} \leq \sqrt{\log\frac{2}{z}} \leq \sqrt{\log\frac{2Kd}{z}}\sqrt{K}. \tag{2.184}$$

This gives the final upper bound

$$\frac{C_n V \beta n}{\ell}(2a_1 + 4a_2\sqrt{\frac{3}{2}}) + \frac{\sqrt{C_n V \beta n}}{\ell}(2 + \frac{1}{\sqrt{\pi}})\sqrt{2a_2\sqrt{\frac{3}{2}}}\Big(\sqrt{\log\frac{2Kd}{\delta}}\sqrt{K}\Big). \quad (2.185)$$

□

Chapter 3

Statistical Risk for Joint Sampling

3.1 Introductory Concepts in Risk Control

For risk control, we want to compare the performance of our Bayesian posterior to the best possible approximation in the model class. Note our previous sampling results are for the continuous uniform prior on $(S_1^d)^K$. When bounding posterior risk, we will work with the discrete uniform prior on $(S_{1,M}^d)^K$. In Chapter 6, we discuss possible ways to extend the risk results we prove here for the discrete prior to the continuous prior, but this remains future work.

Consider $(x_i, y_i)_{i=1}^N$ as an arbitrary sequence of inputs and response values. Let $p_n(w|x^n, y^n)$ be the posterior density trained on data up to index n with gain β. Recall the definitions of posterior mean and predictive density

$$\mu_n(x) = E_{P_n}[f(x,w)|x^n, y^n] \tag{3.1}$$

$$p_n(y|x, x^n, y^n) = E_{P_n}[\frac{\sqrt{\beta}}{\sqrt{2\pi}} e^{-\frac{\beta}{2}(y-f(x,w))^2}|x^n, y^n]. \tag{3.2}$$

Let g be a competitor function we want to compare our performance to. Define its predictive density $q(y|x)$ as Normal$(g(x), \frac{1}{\beta})$. The individual squared error regret is defined

as

$$r_n^{\text{square}} = \frac{1}{2}\Big[(y_n - \mu_{n-1}(x_n))^2 - (y_n - g(x_n))^2\Big]. \tag{3.3}$$

We also define the randomized regret and log regret as

$$r_n^{\text{rand}} = \frac{1}{2}\Big[E_{P_{n-1}}[(y_n - f(x_n, w))^2] - (y_n - g(x_n))^2\Big] \tag{3.4}$$

$$r_n^{\text{log}} = \frac{1}{\beta}\Big[\log \frac{1}{p_{n-1}(y_n|x_n, x^{n-1}, y^{n-1})} - \log \frac{1}{q(y_n|x_n)}\Big]. \tag{3.5}$$

We then have the following ordering of the regrets [12].

Lemma 3.1. *Assume f_w, g are bounded in absolute value by b_f, b_g. Define*

$$\epsilon_n = y_n - g(x_n) \quad b = \frac{b_f + b_g}{2} \quad \lambda_n = b|\epsilon_n| + b^2. \tag{3.6}$$

Then we have

$$r_n^{log} \leq r_n^{rand} \tag{3.7}$$

$$r_n^{square} \leq r_n^{rand} \leq r_n^{log} + 2\beta\lambda_n^2. \tag{3.8}$$

Proof. $r_n^{\text{square}} \leq r_n^{\text{rand}}$ and $r_n^{\text{log}} \leq r_n^{\text{rand}}$ by Jensen's inequality. Consider

$$\frac{1}{2}[(y_n - f(x_n, w))^2 - (y_n - g(x_n))^2], \tag{3.9}$$

as a random variable in w. Then r_n^{rand} is its expected value and r_n^{log} is $-\frac{1}{\beta}$ times its cumulant

generating function at $-\beta$. Note that by a difference in squares identity,

$$\frac{1}{2}[(y_n - f(x_n, w))^2 - (y_n - g(x_n))^2] = (g(x_n) - f(x_n, w))(\epsilon_n + \frac{g(x_n) - f(x_n, w)}{2}) \tag{3.10}$$

$$\leq 2b(|\epsilon_n| + b) \tag{3.11}$$

$$= 2\lambda_n. \tag{3.12}$$

By second order Taylor expansion, the cumulant generating function of a bounded random variable matches the mean to within half the range squared. Thus, we have

$$r_n^{\text{rand}} \leq r_n^{\log} + 2\beta\lambda_n^2. \tag{3.13}$$

□

Define the averaged quantities as

$$R_N^{\text{square}} = \frac{1}{N}\sum_{n=1}^{N} r_n^{\text{square}} \qquad R_N^{\text{rand}} = \frac{1}{N}\sum_{n=1}^{N} r_n^{\text{rand}} \tag{3.14}$$

$$R_N^{\log} = \frac{1}{N}\sum_{n=1}^{N} r_n^{\log} \qquad \Lambda_N^2 = \frac{1}{N}\sum_{n=1}^{N} \lambda_n^2. \tag{3.15}$$

The average regrets follow the same ordering as the pointwise components,

$$R_N^{\text{square}} \leq R_N^{\text{rand}} \leq R_N^{\log} + 2\beta\Lambda_N^2. \tag{3.16}$$

The easiest of the regrets to bound is the log regret as it has a telescoping cancellation of log terms.

Lemma 3.2. *The average log regret is upper bound as*

$$R_N^{log} \leq -\frac{1}{\beta N} \log E_{P_0}[e^{-\frac{\beta}{2}\sum_{n=1}^{N}(y_n - f(x_n,w))^2}] - \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}(y_n - g(x_n))^2. \tag{3.17}$$

Proof. Denote the Bayes factor as

$$Z_n = E_{P_0}[\frac{e^{-\frac{\beta}{2}\sum_{i=1}^{n}(y_i - f(x_i,w))^2}}{(2\pi/\beta)^{\frac{n}{2}}}]. \tag{3.18}$$

The predictive density for p_{n-1} is then the ratio of Z_n to Z_{n-1},

$$p_{n-1}(y_n|x_n, x^{n-1}, y^{n-1}) = \frac{Z_n}{Z_{n-1}}. \tag{3.19}$$

Note this result requires reciprocal variance in our predictive density to match the β gain used in the definition of our Bayesian model. The sum of logs then becomes a telescoping product of canceling terms.

$$-\frac{1}{N}\sum_{n=1}^{N}\log p_{n-1}(y_n|x_n, x^{n-1}, y^{n-1}) \tag{3.20}$$

$$= -\frac{1}{N}\log\prod_{n=1}^{N}\frac{Z_n}{Z_{n-1}} \tag{3.21}$$

$$= -\frac{1}{N}\log\frac{Z_N}{Z_0} \tag{3.22}$$

$$= -\frac{1}{2}\log\left(\frac{\beta}{2\pi}\right) - \frac{1}{N}\log E_{P_0}[e^{-\frac{\beta}{2}\sum_{n=1}^{N}(y_n - f(x_n,w))^2}]. \tag{3.23}$$

The $\beta/2\pi$ terms appear in both p and q, and cancel. □

The key term for bounding risk performance will ultimately depend on a cumulant generating function of loss using the prior,

$$-\frac{1}{\beta N}\log E_{P_0}[e^{-\frac{\beta}{2}\sum_{n=1}^{N}(y_n - f(x_n,w))^2}]. \tag{3.24}$$

Providing upper bounds on this term is the main driving force of risk control. With this key expression controlled by a choice of prior, various notions of risk such as expected Kullback divergence, mean squared risk, and arbitrary sequence regret can be deduced.

One way to upper bound this cumulant generating function is through the index of resolvability [8] approach, which relies on the prior probability of a set of good approximators.

Lemma 3.3 (Index of Resolvability)**.** *Let the prior distribution P_0 have support S and let A be any measurable subset of S. Then we have upper bound*

$$-\frac{1}{\beta N}\log E_{P_0}[e^{-\frac{\beta}{2}\sum_{n=1}^{N}(y_n-f(x_n,w))^2}] \leq \frac{-\log P_0(A)}{\beta N} + \max_{w\in A}\frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}(y_n-f(x_n,w))^2. \tag{3.25}$$

Proof. The proof of this approach is quite simple. The integral on the full space is more than the integral on a subset, thus restricting to a set A upper bounds the negative log integral,

$$-\frac{1}{\beta N}\log E_{P_0}[e^{-\frac{\beta}{2}\sum_{n=1}^{N}(y_n-f(x_n,w))^2}] \leq -\frac{1}{\beta N}\log\int_A e^{-\frac{\beta}{2}\sum_{n=1}^{N}(y_n-f(x_n,w))^2}P_0(dw). \tag{3.26}$$

Multiply and divide by the prior probability of the set $P_0(A)$.

$$\frac{-\log P_0(A)}{\beta N} - \frac{1}{\beta N}\log E_{P_0}[e^{-\frac{\beta}{2}\sum_{n=1}^{N}(y_n-f(x_n,w))^2}|w\in A]. \tag{3.27}$$

Then upper bound the conditional mean by the largest value of the object in the exponent for w in A. □

This philosophy makes risk control quite clear. First, there must exist at least one point in the support of the prior which produces a good fit for the data. Second, the prior must

place enough probability around this point (or rather, at this point in the case of discrete priors) so that the prior probability of the set A is not exponentially small in N. Then both terms of the index of resolvability are controlled.

Note that our finite width neural networks can approximate functions well when the target function lives in V times the convex hull of signed neurons. For a given input data $x^N = (x_i)_{i=1}^N$ and for each weight vector $w \in S_1^d$, consider the vector in $\mathbb{R}^N$ of a single neuron evaluated at the $w \cdot x_i$ points for $i \in 1, \ldots, N$. Let the subset of $\mathbb{R}^N$ denoted $\text{Hull}_N(V\Psi)$ be the closure of the set of convex combinations of V times signed neurons in Ψ evaluated at x^N. This is (the closure of) the set of single-hidden-layer neural networks with variation at most V, evaluated at the given data. For a vector of target function values $(g(x_i))_{i=1}^N$, or more generally any vector of values $g = (g_1, \ldots, g_N)$, we denote its projection as

$$\tilde{g} = \text{argmin}_{f \in \text{Hull}_N(V\Psi)} \|g - f\|_N. \tag{3.28}$$

Note $\tilde{g}$ is the vector of numerical values $\tilde{g} = (\tilde{g}_1, \ldots, \tilde{g}_N) \in \mathbb{R}^N$, which may be interpreted as the vector of outputs of some network evaluated at the x_i points (or a limit thereof), not the network itself that would give rise to these outputs.

We will also have consideration of $\text{Hull}(V\Psi)$ defined as the $L_2(P_X)$ closure of the set of convex combinations of V times signed neurons in Ψ as functions on $[-1, 1]^d$. The $L_2(P_X)$ projection of a function g defined as $\tilde{g}$, the corresponding minimizer of $\|g - f\|^2$ within $\text{Hull}(V\Psi)$, is then a function itself not a vector of specific output values.

For the arbitrary sequence regret bounds the best competitor $\tilde{g}$ is the Euclidean projection into $\text{Hull}_N(V\Psi)$, and for the statistical mean square risk bounds it is the $L_2(P_X)$ projection into $\text{Hull}(V\Psi)$.

We now review results for functions g in V times the convex hull of Ψ, concerning how well a finite width network can approximate them.

3.2 Approximation Ability of Single-Hidden-Layer Neural Networks

First, we recall some known results about the approximation ability of neural networks. We have the following established approximation result from previous work [36].

Lemma 3.4. *Let $x_1, \ldots, x_N$ be an input sequence with each $x_i \in [-1,1]^d$. Assume h is a target function with variation V, that is $\frac{h}{V}$ lives in the closure of the convex hull of neurons with ℓ_1 controlled weight vectors evaluated at the x_i. Then there exists a finite width network with K neurons and some choice of continuous neurons weights $w_1^*, \ldots, w_K^* \in (S_1^d)^K$ and outer weights $c_1, \ldots c_K \in \{-\frac{V}{K}, \frac{V}{K}\}^K$ such that*

$$\sum_{i=1}^{N} (f(x_i, w^*) - h(x_i))^2 \leq N \frac{a_0^2 V^2}{K}. \tag{3.29}$$

We can slightly modify this result to focus on discrete neuron weight vectors in $S_{1,M}^d$ as opposed to the full continuous space.

Lemma 3.5. *Let $x_1, \ldots, x_N$ be a sequence of input values with each $x_i \in [-1,1]^d$. Assume h lives in Hull$_N(V\Psi)$, the closure of the convex hull of signed neurons scaled by V. Then there exists a choice of K discrete-valued interior weights $(w_1^*, \ldots, w_K^*) \in (S_{1,M}^d)^K$ and signed outer weights $c_k \in \{-\frac{V}{K}, \frac{V}{K}\}$ such that for any sequence $(y_i)_{i=1}^N$, the regret compared to h is bound by*

$$\sum_{i=1}^{N} \left(y_i - \sum_{k=1}^{K} c_k \psi(x_i \cdot w_k^*)\right)^2 - (y_i - h(x_i))^2 \leq N \frac{a_0^2 V^2}{K} + N \frac{(V C_N a_2 + V^2 a_1^2)}{M}, \tag{3.30}$$

where a_0, a_1, a_2 are the bounds on ψ and its derivatives, and $C_N = \max_{n \leq N} |y_n| + a_0 V$.

Proof. Fix $x_1, \ldots, x_n$ and $h(x_1), \ldots, h(x_N)$ (or more generally $h_1, \ldots, h_N$). Since h lives

in the closure of the convex hull of signed neurons scaled by V, for every $\epsilon > 0$ there exists some finite width neural network with continuous-valued weight vectors $w_\ell \in S_1^d$ and outer weights c_ℓ with $\sum_\ell |c_\ell| = 1$ such that

$$\tilde{h}(x) = V \sum_\ell c_\ell \psi(x \cdot w_\ell), \quad \sum_{i=1}^N (h(x_i) - \tilde{h}(x_i))^2 \leq \epsilon. \tag{3.31}$$

Let L be a random draw of neuron index where $L = \ell$ with probability $|c_\ell|$. Define $w^{\text{cont}} = w_L$ as the continuous neuron vector at the selected random index L, and $s^{\text{cont}} = \text{sign}(c_L)$ as the sign of the outer weight.

Given a continuous vector w^{cont} of dimension d, we then make a random discrete vector as follows. Define a $d+1$ coordinate, $w_{d+1}^{\text{cont}} = 1 - \|w_{1:d}^{\text{cont}}\|_1$, to have a $d+1$ length vector which sums to 1. Consider a random index $J \in \{1, \ldots d{+}1\}$ where $J = j$ with probability $|w_j^{\text{cont}}|$. Given w^{cont}, this defines a distribution on $\{1, \ldots, d+1\}$. Draw M iid random indices $J_1, \ldots, J_M$ from this distribution and define the counts of each index

$$m_j = \sum_{i=1}^M 1\{J_i = j\}. \tag{3.32}$$

We then define the discrete vector $w^{\text{disc}} \in S_{1,M}^d$ with coordinate values

$$w_j^{\text{disc}} = \text{sign}(w_j^{\text{cont}}) \frac{m_j}{M}. \tag{3.33}$$

Consider then K iid draws of random indexes $L_1, \ldots L_K$, as well as corresponding signs $s_k = \text{sign}(c_{L_k})$. For each L_k consider M iid drawn indexes $J_1^k, \ldots, J_M^k$. This also defines continuous vectors w_k^{cont} and discrete vectors w_k^{disc}. Denote the neural network using a random set of weights and signs,

$$f(x, w, s) = \sum_{k=1}^K \frac{V}{K} s_k \psi(x \cdot w_k). \tag{3.34}$$

Recall the empirical norm and inner product definitions $\|\cdot\|_N^2, \langle\cdot,\cdot\rangle_N$ from the notation section. Consider the expected regret using random discrete neuron weights.

$$E\Big[\|y - f(\cdot, w^{\text{disc}}, s)\|_N^2 - \|y - h\|_N^2\Big]. \tag{3.35}$$

Note this expectation is with respect to the previously defined distribution for w^{disc}, w^{cont}, and s. The data $(x_i, y_i)_{i=1}^N$ are fixed.

Add and subtract the norm using continuous weight vectors, noting that the discrete and continuous vectors of the same index are dependent via the construction,

$$E\Big[\|y - f(\cdot, w^{\text{cont}}, s)\|_N^2 - \|y - h\|_N^2\Big] \tag{3.36}$$

$$+E\Big[\|y - f(\cdot, w^{\text{disc}}, s)\|_N^2 - \|y - f(\cdot, w^{\text{cont}}, s)\|_N^2\Big]. \tag{3.37}$$

Note that using continuous weight vectors the expected value of the random neural network is exactly $\tilde{h}$,

$$E[\frac{V}{K}\sum_{k=1}^{K} s_k \psi(x_i \cdot w_k^{\text{cont}})] = \sum_{i=1}^{N} \tilde{h}(x_i). \tag{3.38}$$

Thus using a bias variance decomposition we have the bound on expression (3.36),

$$E\Big[\|y - f(\cdot, w^{\text{cont}}, s)\|_N^2 - \|y - h\|_N^2\Big] \tag{3.39}$$

$$=\sum_{n=1}^{N} \text{Var}(f(x_i, w^{\text{cont}}, s))^2 + \|y - \tilde{h}\|_N^2 - \|y - h\|_N^2 \tag{3.40}$$

$$\leq N\frac{a_0^2 V^2}{K} + 2\|y - h\|_N \|h - \tilde{h}\|_N + \|\tilde{h} - h\|_N^2 \tag{3.41}$$

$$=N\frac{a_0^2 V^2}{K} + 2\sqrt{N} C_N \sqrt{\epsilon} + \epsilon. \tag{3.42}$$

Where we have used that $f(x, w^{\text{cont}}, s)$ is an average of K iid terms each bounded by $a_0 V$,

so its variance is less than $a_0^2 V^2/K$.

For expression (3.37), perform a second order Taylor expansion of $\|y - f(\cdot, w^{\text{disc}}, s)\|_N^2$ as a function of w^{disc} centered at w^{cont}. For any other vector $\tilde{w}$, denote the expressions

$$\text{res}_i(w, s) = y_i - \sum_{k=1}^{K} s_k \frac{V}{K} \psi(x_i \cdot w_k) \tag{3.43}$$

$$a_{i,k} = - s_k \frac{2V}{K} \text{res}_i(w^{\text{cont}}, s) \psi'(x_i \cdot w_k^{\text{cont}}) \tag{3.44}$$

$$\begin{aligned} b_{i,k,k'}(\tilde{w}, s) = &- s_k \frac{2V}{K} \text{res}_i(\tilde{w}, s) \psi''(x_i \cdot \tilde{w}_k) \delta_{k=k'} \\ &+ 2 s_k s_{k'} \frac{V^2}{K^2} \psi'(x_i \cdot \tilde{w}_k) \psi'(x_i \cdot \tilde{w}_{k'}). \end{aligned} \tag{3.45}$$

Then for any continuous-valued vector w^{cont} and discrete-valued vector w^{disc}, there exists some vector $\tilde{w}$ (in fact along the line between w^{disc} and w^{cont}) such that the second order expansion is exact using that $\tilde{w}$ in the second derivative terms,

$$\|y - f(\cdot, w^{\text{disc}}, s)\|_N^2 \tag{3.46}$$

$$= \|y - f(\cdot, w^{\text{cont}}, s)\|_N^2 + \sum_{i=1}^{N} \sum_{k=1}^{K} a_{i,k} (x_i \cdot (w_k^{\text{disc}} - w_k^{\text{cont}})) \tag{3.47}$$

$$+ \frac{1}{2} \sum_{i=1}^{n} \sum_{k,k'=1}^{K} b_{i,k,k'}(\tilde{w}, s)(x_i \cdot (w_k^{\text{disc}} - w_k^{\text{cont}}))(x_i \cdot (w_{k'}^{\text{disc}} - w_{k'}^{\text{cont}})). \tag{3.48}$$

Expanding the terms we have the expression,

$$E\Big[\|y - f(\cdot, w^{\text{disc}}, s)\|_N^2 - \|y - f(\cdot, w^{\text{cont}}, s)\|_N^2\Big] \tag{3.49}$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} a_{i,k} E[x_i \cdot (w_k^{\text{disc}} - w_k^{\text{cont}})] \tag{3.50}$$

$$- \frac{V}{K} \sum_{i=1}^{N} \sum_{k=1}^{K} E\Big[\text{res}_i(\tilde{w}, s) \psi''(x_i \cdot \tilde{w}_k)(x_i \cdot (w_k^{\text{disc}} - w_k^{\text{cont}}))^2\Big] \tag{3.51}$$

$$+ \sum_{i=1}^{N} E\Big[\Big(\sum_{k=1}^{K} s_k \frac{V}{K} \psi'(\tilde{w}_k)(x_i \cdot (w_k^{\text{disc}} - w_k^{\text{cont}}))\Big)^2\Big]. \tag{3.52}$$

By construction of the distribution, $E[w_k^{\text{disc}}|w_k^{\text{cont}}] = w_k^{\text{cont}}$ so the first order term (3.50) is mean 0. Note that for each i, $|\text{res}_i(\tilde{w}, s)| \leq C_N, \psi'(\cdot) \leq a_1, \psi''(\cdot) \leq a_2$ so we have upper bound

$$= (VC_N a_2 + V^2 a_1^2) \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{1}{K} E[(x_i \cdot (w_k^{\text{disc}} - w_k^{\text{cont}}))^2] \tag{3.53}$$

$$= (VC_N a_2 + V^2 a_1^2) \sum_{i=1}^{N} E[\text{Var}[x_i \cdot w_1^{\text{disc}}|w_1^{\text{cont}}]], \tag{3.54}$$

since the distribution of $(w_k^{\text{disc}}, w_k^{\text{cont}})$ is the same for $k = 1, \ldots, K$.

For a fixed choice of continuous w_1^{cont}, let $x_{i,d+1} = 0$ and consider x_i as a $d+1$ dimension vector. Then $x_i \cdot w_1^{\text{disc}}$ is the inner product of x_i with a vector defined by counts of the independent random indexes $J_1^1, \ldots, J_M^1$. Therefore, the inner product can equivalently be written as an average of M iid random variables using these indexes,

$$\text{Var}[x_i \cdot w_1^{\text{disc}}|w_1^{\text{cont}}] = \text{Var}[\frac{1}{M} \sum_{t=1}^{M} x_{i,J_t^1}|w_1^{\text{cont}}] \tag{3.55}$$

$$= \frac{1}{M} \text{Var}[x_{i,J_1^1}|w_1^{\text{cont}}] \tag{3.56}$$

$$\leq \frac{1}{M}, \tag{3.57}$$

since the $|x_{i,j}|$ are all bounded by 1.

The support of the product measure on discrete weights and outer signs is $(S_{1,M}^d)^K \times \{-1, 1\}^K$. There must be at least one element of the support that has a regret equal to or lower than the average regret. Then taking $\epsilon \to 0$ completes the proof. □

This result allows for analysis of regret with arbitrary $y's$ and competitor h. If our y_i values are specifically the outputs of a neural network in the closure of the convex hull, we can give an improved $1/M^2$ control instead of $1/M$.

Lemma 3.6. *Let $x_1, \cdots, x_N$ be a sequence of input values with each $x_i \in [-1, 1]^d$. As-*

sume h lives in Hull$_N(V\Psi)$, the closure of the convex hull of signed neurons scaled by V. Then there exists a choice of K discrete-valued interior weights $(w_1^*, \cdots, w_K^*) \in (S_{1,M}^d)^K$ and signed outer weights $c_k \in \{-\frac{V}{K}, \frac{V}{K}\}$ such that

$$\sum_{i=1}^{N} \left(h(x_i) - \sum_{k=1}^{K} c_k \psi(x_i \cdot w_k^*) \right)^2 \leq N \frac{a_0^2 V^2}{K} + N \frac{a_2^2 V^2}{4M^2}, \tag{3.58}$$

where a_0, a_1, a_2 are the bounds on ψ and its derivatives.

Proof. See Appendix Section 3.6.1. □

Remark 3.1. We make a note here about odd symmetric activation functions, such as as the $tanh$ function, and non-odd symmetric functions, such as the ReLU squared. For our established approximators in the convex hull, the signs of the outer weights c_r are not known to us in defining our model. Yet in our Bayesian model we fix the signs of our outer neuron scalings c_k as specific signed values, and they are not modeled as flexible in the posterior distribution.

For odd symmetric activation functions, we can consider all signed outer weights to be positive, and any negative outer scalings could be equivalently generated by using negative inner weight vectors. Thus, we can consider all $c_k = \frac{V}{K}$ in our model and the signed discussion in the previous proof becomes irrelevant.

For non-odd symmetric activation functions, if we use double the variation $\tilde{V} = 2V$ and double the number of neurons $\tilde{K} = 2K$, fix the first K outer weights to be positive and the second K to be negative. Then by setting half of inner the weights to be the zero vector, any selection of K inner weights and K signed outer weights can be generated by the model twice as wide. In essence, a non-odd symmetric activation function uses twice the variation and twice the number of neurons to ensure any signed network of size K and variation V can be generated by a certain choice of interior weights alone and fixed outer weights.

3.3 Arbitrary Sequence Regret

We now apply these results to a specific choice of prior. The discrete uniform prior on $(S^d_{1,M})^K$ is a uniform distribution with less than $(2d+1)^{MK}$ possible values. As such, the negative prior log probability of a single point only grows logarithmically in the dimension. By Lemma 3.5, for any target function of the given variation, the set $(S^d_{1,M})^K$ contains at least one choice of parameters that is a good approximation to the function. This yields the following result.

Theorem 3.1 (Odd-Symmetric Neurons)**.** *Let $(x_i)_i^N$ be a sequence of input values with all $x_i \in [-1,1]^d$. Let g be a target function and let h be any element of Hull$_N(V\Psi)$, the closure of the convex hull of signed neurons scaled by V. Let P_0 be the uniform prior on $(S^d_{1,M})^K$. Assume the neuron activation function is odd symmetric and set all outer weights as $c_k = \frac{V}{K}$. For any sequence of values $(y_i)_{i=1}^N$, define the terms*

$$\epsilon_n = y_n - g(x_n) \qquad \tilde{\epsilon}_n = y_n - h(x_n). \tag{3.59}$$

Then the average log regret of the sequence of posterior predictive distributions is upper bounded by

$$R_N^{\log} \leq \frac{MK\log(2d+1)}{\beta N} + \frac{a_0^2 V^2}{2K} + \frac{(VC_N a_2 + V^2 a_1^2)}{2M} + \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}(\tilde{\epsilon}_n^2 - \epsilon_n^2). \tag{3.60}$$

In particular, h may be the Hull$_N(V\Psi)$ projection of g, which is denoted $\tilde{g}$.

Proof. Recall the definition of $\|\cdot\|_N^2$ and $\langle\cdot,\cdot\rangle_N$ given in the notation section. By Lemmas 3.2 and 3.3, for any set A of discrete neuron values, we can upper bound the average log

regret as

$$-\frac{\log P_0(A)}{\beta N}+\frac{1}{2N}\max_{w\in A}((\|y-f_w\|_N^2-\|y-g\|_N^2) \tag{3.61}$$

$$=-\frac{\log P_0(A)}{\beta N}+\frac{1}{2N}\max_{w\in A}((\|y-f_w\|^2-\|y-h\|_N^2)+\frac{1}{2N}(\|y-h\|_N^2-\|y-g\|_N^2). \tag{3.62}$$

By Lemma 3.5, there exists a single discrete point with bounded regret from h. Select A as the singleton set at this point. We then consider the number of points in the support of the prior.

Let w be a vector of length d with ℓ_1 norm less than or equal to 1. To make a vector with only positive entries, use double the coordinates and set $\tilde{w}_j = w_j$ if $w_j > 0$ and $\tilde{w}_{d+j} = -w_j$ else. Then add one more coordinate to count how far the ℓ_1 norm is from 1, $\tilde{w}_{2d+1} = 1 - \|w\|_1$. Thus, each w vector can be uniquely expressed as a $2d+1$ size vector of positive entries that sums to exactly 1.

Consider the entries of $\tilde{w}$ as having to be multiples of $\frac{1}{M}$. Each $\tilde{w}$ vector is then a histogram on $2d+1$ locations where the heights at each location can be $\{0, 1, \ldots, M\}/M$. An over-counting of the number of possible histograms is then $(2d+1)^M$. The product prior on K independent weight vectors gives an additional K power. Since the discrete uniform prior support set has less than or equal to $(2d+1)^{MK}$ points,

$$-\log P_0(A) \leq (MK)\log(2d+1). \tag{3.63}$$

Combined with the bound from Lemma 3.5 this completes the proof.

□

In general, for a non-odd symmetric activation function (e.g. squared ReLU) we use twice the number of neurons with fixed outer weights to ensure any choice of signed neurons of half the width can be generated. Thus, we can prove the same order bounds

but with slightly different constants. Here, we give the explicit changes, but all future theorems will be given for the odd-symmetric case and the non-odd symmetric version can be similarly derived.

Corollary 3.1 (Non-Odd Symmetric Neurons)**.** *For a neural network with non-odd symmetric neurons, use twice the number of neurons $\tilde{K} = 2K$ neurons and twice the variation $\tilde{V} = 2V$. Set the first K outer weights as positive $c_k = \frac{V}{K}$ and the second K outer weights as negative $c_k = -\frac{V}{K}$. Then we have the bound of*

$$R_N^{\log} \leq \frac{M\tilde{K}\log(2d+1)}{\beta N} + \frac{a_0^2\tilde{V}^2}{\tilde{K}} + \frac{(\tilde{V}C_N a_2 + \tilde{V}^2 a_1^2)}{2M} + \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}(\tilde{\epsilon}_n^2 - \epsilon_n^2). \tag{3.64}$$

Proof. By Lemma 3.5, there exists some signed neural network of width K that achieves the given regret bound with target function g. Our chosen network of width $\tilde{K}$ of fixed signed neurons has the flexibility to generate arbitrary signed (i.e. any number proportion of positive or negative signs) networks of width $K = \frac{\tilde{K}}{2}$. The proof then follows. □

Theorem 3.2. *Let $(x_i)_{i=1}^N$ be a sequence of input values with all $x_i \in [-1,1]^d$. Let g be a target function bounded by a value b and let h be any element of Hull$_N(V\Psi)$, the closure of the convex hull of signed neurons scaled by V. Let P_0 be the uniform prior on $(S_{1,M}^d)^K$. Assume the neuron activation function is odd symmetric and set all outer weights as $c_k = \frac{V}{K}$. For any sequence of values $(y_i)_{i=1}^N$, define the terms*

$$\epsilon_n = y_n - g(x_n) \qquad \tilde{\epsilon}_n = y_n - h(x_n). \tag{3.65}$$

Then the average squared regret of the posterior mean predictions is upper bounded by

$$R_N^{square} \leq \frac{MK\log(2d+1)}{\beta N} + \frac{a_0^2V^2}{2K} + \frac{(VC_Na_2 + V^2a_1^2)}{2M} \tag{3.66}$$

$$+ 2\beta\frac{1}{N}\sum_{n=1}^{N}\Big(\frac{a_0V+b}{2}|\tilde{\epsilon}_n| + \Big(\frac{a_0V+b}{2}\Big)^2\Big)^2 + \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}(\tilde{\epsilon}_n^2 - \epsilon_n^2). \tag{3.67}$$

Proof. Apply Lemma 3.1 and Theorem 3.1 to upper bound squared regret by log regret and an additional β term. Note that f_w is bounded by a_0V and g is bounded by b. □

Note that $|\tilde{\epsilon}_n| \leq C_N$ and can provide an upper bound, since $\tilde{\epsilon}$ is known to the user. We next derive the choices of β, M, K which optimize the bounds.

Corollary 3.2. *Replace the residuals $\tilde{\epsilon}_n$ with C_N in expression* (3.67)*. Denote the value*

$$B_1 = (C_N + \frac{a_0V+b}{2})^2 \tag{3.68}$$

Let

$$\beta^* = \gamma_1\Big(\frac{\log(2d+1)}{N}\Big)^{\frac{1}{4}} \tag{3.69}$$

$$K^* = \gamma_2\Big(\frac{N}{\log(2d+1)}\Big)^{\frac{1}{4}} \tag{3.70}$$

$$M^* = \gamma_3\Big(\frac{N}{\log(2d+1)}\Big)^{\frac{1}{4}}, \tag{3.71}$$

where

$$\gamma_1 = \frac{(a_0V)^{\frac{1}{2}}(\frac{a_2VC_N+a_1^2V^2}{2})^{\frac{1}{4}}}{2(\frac{a_0V+b}{2})^{\frac{3}{2}}(B_1)^{\frac{3}{4}}} \tag{3.72}$$

$$\gamma_2 = \frac{(a_0V)^{\frac{3}{2}}}{2(\frac{a_0V+b}{2})^{\frac{1}{2}}(B_1)^{\frac{1}{4}}(\frac{a_2VC_N+a_1^2V^2}{2})^{\frac{1}{4}}} \tag{3.73}$$

$$\gamma_3 = \frac{(\frac{a_2VC_N+a_1^2V^2}{2})^{\frac{3}{4}}}{(a_0V)^{\frac{1}{2}}(\frac{a_0V+b}{2})^{\frac{1}{2}}(B_1)^{\frac{1}{4}}}. \tag{3.74}$$

Then we have a bound on the squared regret of the form

$$4\Big(a_0 V(\frac{a_0 V + b}{2})\Big)^{\frac{1}{2}}\Big(B_1(\frac{a_2 V C_N + a_1^2 V^2}{2})\Big)^{\frac{1}{4}}\Big(\frac{\log(2d+1)}{N}\Big)^{\frac{1}{4}} + \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}(\tilde{\epsilon}_n^2 - \epsilon_n^2). \tag{3.75}$$

In particular, if the function g lives in the convex hull scaled by V and h is chosen to be g, then $\epsilon_n = \tilde{\epsilon}_n$ and we have an upper bound of

$$R_N^{square} = O((C_N)^{\frac{3}{4}}\Big(\frac{\log(2d+1)}{N}\Big)^{\frac{1}{4}}). \tag{3.76}$$

In the algorithm M, K must be integers. The closest integer values to the stated continuous values achieve a similar bound.

Remark 3.2. Equations (3.69), (3.70), (3.71) represent the choice of modeling parameters that optimize our derived bound. However, we do not advocate plugging in these specific parameter choices directly into the model and training only one model based on these values. If, for example, it happens that the target is such that it can be approximated with an improved $1/K^2$ instead of $1/K$ in (3.58) (and hence in (3.66)), then the given bounds would not provide the best choices of K, M and β. We instead advocate adaptive modeling by putting a prior on a number of possible M, K, β values, say 100-1000 possible values each.

Corollary 3.2 shows one choice of β^*, K^*, M^* that can achieve bounded regret. If we include these values in our prior set, by a further index of resolvability argument we can show using a uniform prior on a finite number of M, K, β possible values, we would pay a log number of possible values divided by βN price in the bound, which can be easily controlled. We note that computationally, all different M, K, β combinations result in different models that can be sampled in parallel and independently on different cores at the same time and the results combined at the end. Thus, such an approach is amenable to

GPU usage and distributed computing from a practical perspective.

3.4 IID Sequence Predictive Risk Control

In the previous section, we studied risk control for arbitrary data sequences with no assumptions on the data. We compared performance in terms of regret to a competitor fit. Here, we assume training data iid from a data distribution and prove bounds on predictive risk for future data pairs.

Suppose $(x_i, y_i)_{i=1}^N$ are independent with y having conditional mean $E[Y|X = x] = g(x)$ and conditional variance $\text{Var}[Y|X = x] = \sigma_x^2$, with bound on the variance $\max_x \sigma_x^2 \leq \sigma^2$. Recall that our neural network is trained with a gain β. In a typical setting with assumed independent Gaussian errors, $\sigma_x^2 = \sigma^2$ for each x value and β would be set as a constant matching the inverse variance $\beta = \frac{1}{\sigma^2}$. However, we would also like to consider gains decaying in N, such as $\beta = [(\log d)/N]^{\frac{1}{4}}$. Using such a β, we can reproduce the arbitrary regret results above and show for the Cesàro mean estimator $\hat{g}$,

$$E[\|g - \hat{g}\|^2] = O(\Big(\frac{\log(d)}{N+1}\Big)^{\frac{1}{4}}). \tag{3.77}$$

Note that this statistical risk bound makes no assumptions about the distribution of Y given X aside from its mean and variance. In particular, the distribution of the data need not be Gaussian even though we use quadratic loss to define our posterior densities. Additionally, our sampling gain β does not have to match any data specific value exactly (that is β does not depend on σ^2 which may not be known).

If we further assume the conditional distribution is independent normal with constant variance, $Y|X \sim \text{Normal}(g(X), \sigma^2)$, and the gain β accurately represents the inverse variance, $\beta = \frac{1}{\sigma^2}$, then we can give a similar bound for Kullback risk which has an improved

$1/3$ power

$$E[D(P_{Y|X}\|Q^{\text{avg}}_{Y|X,X^N,Y^N})] = O(\Big(\frac{\log(d)}{N+1}\Big)^{\frac{1}{3}}). \tag{3.78}$$

We first bound the mean squared risk without any assumptions on β and no normality assumptions.

Theorem 3.3. *Let g be a target function with absolute value bounded by b and let $\tilde{g}$ be its $L_2(P_X)$ projection into the closure of the convex hull of signed neurons scaled by V. Let P_0 be the uniform prior on $(S^d_{1,M})^K$. Assume the neuron activation function is odd symmetric and set all outer weights as $c_k = \frac{V}{K}$. Let $(X_i, Y_i)_{i=1}^N$ be training data iid with conditional mean $g(X_i)$ and conditional variance $\sigma^2_{X_i}$ with variance bound $\sigma_x^2 \leq \sigma^2$. Assume the data distribution P_X has support in $[-1,1]^d$. Then the mean squared statistical risk of the averaged posterior mean estimator $\hat{g}$ is upper bounded by*

$$E[\|g - \hat{g}\|^2] \leq \frac{MK\log(2d+1)}{\beta(N+1)} + \frac{a_0^2V^2}{2K} + \frac{(V(a_0V+b)a_2 + V^2a_1^2)}{2M} \tag{3.79}$$

$$+ 2\beta(\frac{a_0V+b}{2})^2(\sigma + \frac{a_0V+b}{2})^2 + \|g-\tilde{g}\|^2. \tag{3.80}$$

Let

$$\beta^* = \gamma_1\Big(\frac{\log(2d+1)}{N+1}\Big)^{\frac{1}{4}} \tag{3.81}$$

$$K^* = \gamma_2\Big(\frac{N+1}{\log(2d+1)}\Big)^{\frac{1}{4}} \tag{3.82}$$

$$M^* = \gamma_3\Big(\frac{N+1}{\log(2d+1)}\Big)^{\frac{1}{4}}, \tag{3.83}$$

where

$$\gamma_1 = \frac{(a_0V)^{\frac{1}{2}}\big(\frac{V(a_0V+b)a_2+V^2a_1^2}{2}\big)^{\frac{1}{4}}}{2\big(\frac{a_0V+b}{2}\big)^{\frac{3}{2}}\big(\sigma+\frac{a_0V+b}{2}\big)^{\frac{3}{2}}} \tag{3.84}$$

$$\gamma_2 = \frac{(a_0V)^{\frac{3}{2}}}{2\big(\frac{a_0V+b}{2}\big)^{\frac{1}{2}}\big(\sigma+\frac{a_0V+b}{2}\big)^{\frac{1}{2}}\big(\frac{V(a_0V+b)a_2+V^2a_1^2}{2}\big)^{\frac{1}{4}}} \tag{3.85}$$

$$\gamma_3 = \frac{\big(\frac{V(a_0V+b)a_2+V^2a_1^2}{2}\big)^{\frac{3}{4}}}{(a_0V)^{\frac{1}{2}}\big(\frac{a_0V+b}{2}\big)^{\frac{1}{2}}\big(\sigma+\frac{a_0V+b}{2}\big)^{\frac{1}{2}}}. \tag{3.86}$$

Then we have a bound on the mean squared risk of the form

$$4\Big(a_0V\big(\frac{a_0V+b}{2}\big)\big(\sigma+\frac{a_0V+b}{2}\big)\Big)^{\frac{1}{2}}\Big(\frac{V(a_0V+b)a_2+V^2a_1^2}{2}\Big)^{\frac{1}{4}}\Big(\frac{\log(2d+1)}{N}\Big)^{\frac{1}{4}} \tag{3.87}$$

$$+\|g-\tilde{g}\|^2. \tag{3.88}$$

Proof. Note the following expectations are with respect to training data $(X_i, Y_i)_{i=1}^N$ and a new input and response pair $(X, Y) = (X_{N+1}, Y_{N+1})$ all iid from the data distribution $P_{X,Y}$. Note that since there are many expectations with respect to different random variables in the proof, we will make explicit use of subscripts to indicate which random variable each expectation is with respect to. The initial expectation is for the data distribution $P_{X,Y}$ for the training data as well as the new X point which we are evaluating at. Bring the average of the Cesàro mean outside the square to upper bound

$$\frac{1}{2}E_{P_{X^{N+1},Y^{N+1}}}[(g(X)-\hat{g}(X))^2] \leq \frac{1}{2}\sum_{n=0}^{N}\frac{1}{N+1}E_{P_{X^{N+1},Y^{N+1}}}[(g(X)-\mu_n(X))^2] \tag{3.89}$$

$$=\frac{1}{2}E_{P_{X^{N+1},Y^{N+1}}}\Big[\sum_{n=0}^{N}\frac{(g(X_{n+1})-\mu_n(X_{n+1}))^2}{N+1}\Big] \tag{3.90}$$

$$=\frac{1}{2}E_{P_{X^{N+1},Y^{N+1}}}\Big[\sum_{n=0}^{N}\frac{(Y_{n+1}-\mu_n(X_{n+1}))^2-(Y_{n+1}-g(X_{n+1}))^2}{N+1}\Big], \tag{3.91}$$

where we have added the Y in using the fact that $Y_{n+1} - g(X_{n+1})$ is mean 0 under $P_{X,Y}$. This is then exactly the expectation of a squared regret. Define notation $R^{\log}_{N+1}(X^{N+1}, Y^{N+1})$, $R^{\text{square}}_{N+1}(X^{N+1}, Y^{N+1})$ as the log and squared regret relative to g at the random $(X_i, Y_i)_{i=1}^{N+1}$ values. Then by Lemma 3.1 we have,

$$E_{P_{X^{N+1},Y^{N+1}}}\left[R^{\text{square}}_{N+1}(X^{N+1}, Y^{N+1})\right] \leq E_{P_{X^{N+1},Y^{N+1}}}\left[R^{\log}_{N+1}(X^{N+1}, Y^{N+1})\right] \quad (3.92)$$

$$+2E_{P_{X^{N+1},Y^{N+1}}}\left[\beta\frac{1}{N+1}\sum_{n=0}^{N}(\frac{a_0V+b}{2}|Y_{n+1} - g(X_{n+1})| + (\frac{a_0V}{2})^2)^2\right] \quad (3.93)$$

$$\leq E_{P_{X^{N+1},Y^{N+1}}}\left[R^{\log}_{N+1}(X^{N+1}, Y^{N+1})\right] + 2\beta(\frac{a_0V+b}{2})^2(\sigma + \frac{a_0V+b}{2})^2. \quad (3.94)$$

Then by Lemma 3.2,

$$E_{P_{X^{N+1},Y^{N+1}}}[R^{\log}_{N+1}(X^{N+1}, Y^{N+1})] \leq -\frac{1}{2}\frac{1}{N+1}\sum_{n=0}^{N} E_{P_{X^{N+1},Y^{N+1}}}[(Y_{n+1} - g(X_{n+1}))^2] \quad (3.95)$$

$$+ \frac{1}{\beta(N+1)}E_{P_{X^{N+1},Y^{N+1}}}[-\log\int e^{-\frac{\beta}{2}\sum_{n=0}^{N}(Y_{n+1}-f(X_{n+1},w))^2}P_0(dw)]. \quad (3.96)$$

Use the $\|\cdot\|^2_{N+1}$ and $\langle\cdot,\cdot\rangle_{N+1}$ notation defined earlier. Note the outer expectation in (3.96) is with respect to X^{N+1}, Y^{N+1} from the data distribution and the inner integral is for w using the prior, as a consequence of our index of resolvability bound. Recall that our prior P_0 is absolutely continuous with respect to a reference η with density $p_0(w)$. In this proof, η can be considered as counting measure on $(S^d_{1,M})^K$ for the discrete uniform prior, but in other instances it could be considered as Lebesgue measure.

Add and subtract $g(X_{n+1})$ inside each of the terms in the exponent of (3.96), expand

the terms and note the cancellation of the first quadratic term,

$$-\frac{1}{2}\frac{1}{N+1}E_{P_{X^{N+1},Y^{N+1}}}[\|Y-g\|_{N+1}^2] \tag{3.97}$$

$$+\frac{1}{\beta(N+1)}E_{P_{X^{N+1},Y^{N+1}}}[-\log\int p_0(w)e^{-\frac{\beta}{2}\|Y-g+g-f_w\|_{N+1}^2}\eta(dw)] \tag{3.98}$$

$$=\frac{1}{\beta(N+1)}E_{P_{X^{N+1},Y^{N+1}}}[-\log\int p_0(w)e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2-\beta\langle Y-g,g-f_w\rangle_{N+1}}\eta(dw)]. \tag{3.99}$$

Inside the log, multiply and divide by $\int p_0(w)e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2}\eta(dw)$, which acts as the normalizing constant of a density with respect to η,

$$\frac{E_{P_{X^{N+1},Y^{N+1}}}[-\log\int\left(\frac{p_0(w)e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2}}{\int p_0(w)e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2}\eta(dw)}\right)e^{-\beta\langle Y-g,g-f_w\rangle_{N+1}}\eta(dw)]}{\beta(N+1)} \tag{3.100}$$

$$+\frac{E_{P_{X^{N+1},Y^{N+1}}}[-\log\int p_0(w)e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2}\eta(dw)]}{\beta(N+1)}. \tag{3.101}$$

Interestingly, the density in equation (3.100) can be viewed as a pseudo posterior $p_n(w|g)$ using the $g(x_i)$ data points in place of the y_i to define the likelihood. This cannot be used for actual training since the function g is not known to us, but is a tool for risk analysis.

We can then bring the $-\log$, which is a convex function, inside the integral to get an upper bound in (3.100). This brings the inner product in the exponent down. Then switch the order of the inner w integral and outer $Y^{N+1}|X^{N+1}$ expectation. Note in this analysis, the distribution of w is the prior distribution P_0 and is independent of the X^{N+1}, Y^{N+1} values. Under the data distribution, Y^{N+1} conditioned on X^{N+1} is independent of w and mean $g(X^{N+1})$, thus the expected value of the inner product is 0 for any choice of w. Thus

expression (3.100) is less than 0.

$$\frac{E_{P_{X^{N+1},Y^{N+1}}}[-\log \int \left(\frac{p_0(w)e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2}}{\int p_0(w)e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2\eta(dw)}}\right)e^{-\beta\langle Y-g,g-f_w\rangle_{N+1}}\eta(dw)]}{\beta(N+1)} \tag{3.102}$$

$$\leq \frac{E_{P_{X^{N+1}}}[\int \left(\frac{p_0(w)e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2}}{\int p_0(w)e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2\eta(dw)}}\right)E_{P_{Y^{N+1}|X^{N+1}}}[\langle Y-g,g-f_w\rangle_{N+1}|X^{N+1}]\eta(dw)]}{N+1} \tag{3.103}$$

$$=0. \tag{3.104}$$

Then consider expression (3.101). This term can be bounded by the logic in Lemma 3.5.

Now, we must make a distinction between the $L_2(P_X)$ projection of g into the convex Hull of signed neurons, and a specific choice of finite linear combination of neurons (i.e. a finite width neural network). Let $\tilde{g}$ be the projection of g into the $L_2(P_X)$ closure of the convex hull of signed neurons. $\tilde{g}$ is not itself a finite width neural network, but a limit thereof. Therefore, let $\tilde{g}_\epsilon$ be a specific finite linear combination of signed neurons scaled by V that is ϵ close to $\tilde{g}$ in $L_2(P_X)$ distance. Then, $\tilde{g}_\epsilon$ evaluated at any sequence of values $x_1, \ldots, x_{N+1}$ is an element of the Euclidean closure of the convex Hull of signed neurons which we have called $\text{Hull}_{N+1}(V\Psi)$.

Add and subtract $\|g-\tilde{g}_\epsilon\|_{N+1}^2$ in the exponent of expression (3.101) and we have the

result,

$$\frac{E_{P_{X^{N+1}}}[-\log\int e^{-\frac{\beta}{2}(\|g-f_w\|^2_{N+1}-\|g-\tilde{g}_\epsilon\|^2_{N+1})}P_0(dw)]}{\beta(N+1)}+\frac{1}{2}\frac{E_{P_{X^{N+1}}}[\|g-\tilde{g}_\epsilon\|^2_{N+1}]}{N+1} \tag{3.105}$$

$$=\frac{E_{P_{X^{N+1}}}[-\log\int e^{-\frac{\beta}{2}(\|g-f_w\|^2_{N+1}-\|g-\tilde{g}_\epsilon\|^2_{N+1})}P_0(dw)]}{\beta(N+1)}+\frac{E_{P_X}[(g(X)-\tilde{g}(X))^2]}{2} \tag{3.106}$$

$$+\frac{1}{2}E_{P_X}[2(g(X)-\tilde{g}(X))(\tilde{g}(X)-\tilde{g}_\epsilon(X))]+\frac{1}{2}E_{P_X}[(\tilde{g}(X)-\tilde{g}_\epsilon)^2]. \tag{3.107}$$

$$\leq\frac{E_{P_{X^{N+1}}}[-\log\int e^{-\frac{\beta}{2}(\|g-f_w\|^2_{N+1}-\|g-\tilde{g}_\epsilon\|^2_{N+1})}P_0(dw)]}{\beta(N+1)}+\frac{E_{P_X}[(g(X)-\tilde{g}(X))^2]}{2} \tag{3.108}$$

$$+2b\epsilon+\frac{1}{2}\epsilon^2. \tag{3.109}$$

Focus now on expression (3.108). To bound this further, think of $g(x_i)$ as the "y_i" observations in Lemma 3.5, and $\tilde{g}_\epsilon$ here is playing the role of the h competitor. The result of Lemma 3.5 would then apply. However, our $g(x_i)$ are now bounded which offers an improvement. Each instance of $C_{N+1}=\max_{1\leq n\leq N+1}|y_n|+a_0V$ in the result of Lemma 3.5 can be replaced with

$$\max_{1\leq n\leq N+1}|g(x_n)|+a_0V\leq b+a_0V, \tag{3.110}$$

which is not y dependent. Thus, the random variable y can have unbounded range, yet its mean function is bounded and the range of the mean function is the relevant term for the bound. An expression like Theorem 3.1 then follows replacing C_N with a_0V+b. Returning to expression (3.94) and applying this bound, we have our final expression,

$$\frac{MK\log(2d+1)}{\beta(N+1)}+\frac{a_0^2V^2}{2K}+\frac{(V(a_0V+b)a_2+V^2a_1^2)}{2M} \tag{3.111}$$

$$+2\beta(\frac{a_0V+b}{2})^2(\sigma+\frac{a_0V+b}{2})^2+\frac{1}{2}E[(g(X)-\tilde{g}(X))^2]+2b\epsilon+\frac{1}{2}\epsilon^2. \tag{3.112}$$

Plugging in the stated β^*, M^*, K^* gives the more specific bound. Then take $\epsilon \to 0$. $\square$

When the target g does not have variation less than or equal to V, there is the unavoidable squared loss never smaller than $\|g - \tilde{g}\|^2$, the squared loss of the projection. Nevertheless, the Theorem shows that the mean square risk $E[\|g - \hat{g}\|^2]$ is close to that minimal squared loss of $\tilde{g}$. A corollary is that by a Pythagorean inequality $\hat{g}$ is close to $\tilde{g}$ itself in mean squared distance.

Corollary 3.3. *Let g be the target function and $\tilde{g}$ its $L_2(P_X)$ projection into the closure of the convex hull of signed neurons scaled by V. Assume the risk of the Cesàro mean estimator is bounded by*

$$E[\|g - \hat{g}\|^2] \leq \|g - \tilde{g}\|^2 + O(\Big(\frac{\log(d)}{N}\Big)^{\frac{1}{4}}) \tag{3.113}$$

Then the distance from $\hat{g}$ to the projection $\tilde{g}$ is bounded by this error term decaying N,

$$E[\|\tilde{g} - \hat{g}\|^2] = O(\Big(\frac{\log(d)}{N}\Big)^{\frac{1}{4}}) \tag{3.114}$$

Proof. The closure of the convex hull of signed neurons is a convex set. $\tilde{g}$ being the projection of g onto the set provides a half-space of functions h with the inner product $\langle h - \tilde{g}, g - \tilde{g} \rangle$ less than or equal to 0 which includes that convex set, where here $\langle \cdot, \cdot \rangle$ is the $L_2(P_X)$ inner product. This means for all points inside the closure of the convex hull, of which $\hat{g}$ is a member, we have a Pythagorean inequality,

$$\|g - \tilde{g}\|^2 + \|\tilde{g} - \hat{g}\|^2 \leq \|g - \hat{g}\|^2, \tag{3.115}$$

and thus

$$\|\tilde{g} - \hat{g}\|^2 \leq \|g - \hat{g}\|^2 - \|g - \tilde{g}\|^2. \tag{3.116}$$

The conclusion follows by taking the expectation. □

In the result of Theorem 3.3, the $1/M$ dependence in equation (3.80) comes from applying the approximation result in Lemma 3.5. However, if the target function g itself is assumed to live in the closure of the convex Hull of signed neurons, we can use the improved Lemma 3.6 which has a $1/M^2$ dependence instead. We can then get a risk control of the order $O([(\log d)/N]^{2/7})$. The $2/7$ power is slightly better than the $2/8 = 1/4$ power of the previous theorem.

Corollary 3.4. *Let g be a target function and assume it lives in the $L_2(P_X)$ closure of the convex hull of signed neurons scaled by V. Let P_0 be the uniform prior on $(S^d_{1,M})^K$. Assume the neuron activation function is odd symmetric and set all outer weights as $c_k = \frac{V}{K}$. Let $(X_i, Y_i)_{i=1}^N$ be training data iid with conditional mean $g(X_i)$ and conditional variance $\sigma^2_{X_i}$ with variance bound $\sigma_x^2 \leq \sigma^2$. Assume the data distribution P_X has support in $[-1, 1]^d$. Then the mean squared statistical risk of the averaged posterior mean estimator $\hat{g}$ is upper bounded by*

$$E[\|g - \hat{g}\|^2] \leq \frac{MK \log(2d+1)}{\beta(N+1)} + \frac{a_0^2 V^2}{2K} + \frac{a_2^2 V^2}{8M^2} + 2\beta(a_0 V)^2(\sigma + a_0 V)^2. \quad (3.117)$$

If we set

$$M = \Big(\frac{N+1}{\log(2d+1)}\Big)^{\frac{1}{7}} \qquad K = \Big(\frac{N+1}{\log(2d+1)}\Big)^{\frac{2}{7}} \quad (3.118)$$

$$\beta = \Big(\frac{\log(2d+1)}{N+1}\Big)^{\frac{1}{7}} \quad (3.119)$$

we have an error bound of the form

$$E[\|g - \hat{g}\|^2] \leq \Big(\frac{\log(2d+1)}{N+1}\Big)^{\frac{2}{7}} [1 + \frac{a_0^2 V^2}{2} + \frac{a_2^2 V^2}{8} + 2a_0^2 V^2(\sigma + a_0 V)^2]. \quad (3.120)$$

Proof. See Appendix Section 3.6.1. □

For a target function g, consider the distribution for $Y|X$ as Normal$(g(X), \frac{1}{\beta})$. Consider X^N, Y^N as training data used to train our Bayesian model independent according to $P_{X,Y}$ and a new (X, Y) (also denoted X_{N+1}, Y_{N+1}) as a input and response arising independently from the same distribution. We then bound the expected Kullback divergence between $P_{Y|X}$ and $Q^{\text{avg}}_{Y|X,X^N,Y^N}$.

Theorem 3.4. *Assuming the data distribution is $Y|X \sim$ Normal$(g(X), \frac{1}{\beta})$ we bound the we bound the Kullback risk of the posterior predictive distribution as*

$$E[D(P_{Y|X} \| Q^{avg}_{Y|X,X^N,Y^N})] \leq E\Big[\frac{-\log E_{P_0}[e^{-\frac{\beta}{2}\sum_{i=1}^{N+1}(f(X_i,w)-g(X_i))^2}]}{N+1}\Big]. \tag{3.121}$$

Proof. The proof of this theorem follows much the same as the arbitrary log regret proof, with a few changes using the iid nature of the data.

The Cesàro average predictive density is a mixture of $N+1$ predictive densities $p_n(y|x, x^n, y^n)$. Since Kullback divergence is a convex function, this is less than the average of individual divergences

$$\frac{1}{N+1}\sum_{n=0}^{N} E[D(P_{Y|X} \| P_{Y|X,X^n,Y^n})]. \tag{3.122}$$

We assume the training data and new data come iid from the same distribution. Therefore, the predictive distribution for any $P_{Y_{i^*}|X_{i^*},X^n,Y^n}$ is the same distribution for all $i^* > n$. That is, if a Bayesian model is only trained on data up to index n, all data of higher index has the same predictive distribution. We can consider our new pair X, Y we are predicting on as a future index X_{N+1}, Y_{N+1}. Thus, we have

$$\frac{1}{N+1}\sum_{n=0}^{N} E[D(P_{Y|X} \| P_{Y|X,X^n,Y^n})] \tag{3.123}$$

$$=\frac{1}{N+1}\sum_{n=0}^{N} E[D(P_{Y_{n+1}|X_{n+1}} \| P_{Y_{n+1}|X_{n+1},X^n,Y^n})]. \tag{3.124}$$

Note that in this form we can recognize via the chain rule of information theory as in [8] that expression (3.124) is equal to the total Kullback divergence of the product measure $P_{Y^{N+1}|X^{N+1}}$ from the Bayes joint distribution

$$Q_{Y^{N+1}|X^{N+1}}(\cdot) = \int (\prod_{n=0}^{N} Q_{Y_{n+1}|w,X_{n+1}}(\cdot)) P_0(dw) \tag{3.125}$$

where $Q_{Y_{n+1}|w,X_{n+1}}$ is Normal($f_w(X_{n+1}), 1/\beta$). That is,

$$\frac{1}{N+1} \sum_{n=0}^{N} E[D(P_{Y_{n+1}|X_{n+1}} \| P_{Y_{n+1}|X_{n+1},X^n,Y^n})] \tag{3.126}$$

$$= \frac{1}{N+1} E_{P_{X^{N+1}}}[D(P_{Y^{N+1}|X^{N+1}} \| Q_{Y^{N+1}|X^{N+1}})]. \tag{3.127}$$

However we will derive this expression directly as well, and show that it has the bound indicated at the right side of (3.121) (this bound on total divergence is akin to one derived in [5]). Consider each individual term in (3.124), we will see a similar telescoping cancellation as in the log regret proof. Denote the Bayes factor,

$$Z_n = E_{P_0}[\frac{e^{-\frac{\beta}{2}\sum_{i=1}^{n}(y_i - f(x_i,w))^2}}{(2\pi/\beta)^{\frac{n}{2}}}]. \tag{3.128}$$

Then the predictive density $p_n(y_{n+1}|x_{n+1}, x^n, y^n)$ is the ratio of Z_{n+1} to Z_n,

$$p_n(y_{n+1}|x_{n+1}, x^n, y^n) = \frac{Z_{n+1}}{Z_n}. \tag{3.129}$$

For each individual Kullback term we have

$$E[D(P_{Y_{n+1}|X_{n+1}} \| P_{Y_{n+1}|X_{n+1},X^n,Y^n})] = E[-\frac{\beta}{2}(Y_{n+1} - g(X_{n+1}))^2 - \log \frac{Z_{n+1}}{Z_n}] \tag{3.130}$$

$$- \frac{1}{2}\log(\frac{2\pi}{\beta}). \tag{3.131}$$

Use notation $\|\cdot\|_{N+1}, \langle\cdot,\cdot\rangle_{N+1}$ as before. The sum of Kullback risks divided by $N+1$ is

$$-\frac{\beta}{2}E[\frac{\|Y-g\|_{N+1}^2}{N+1}] - \frac{1}{2}\log(\frac{2\pi}{\beta}) - \frac{1}{N+1}E[\log\prod_{n=0}^{N}\frac{Z_{n+1}}{Z_n}] \tag{3.132}$$

$$= -\frac{\beta}{2}E[\frac{\|Y-g\|_{N+1}^2}{N+1}] - \frac{1}{2}\log(\frac{2\pi}{\beta}) - \frac{1}{N+1}E[\log\frac{Z_{N+1}}{Z_0}]. \tag{3.133}$$

We now proceed with an argument similar to bounding equation (3.96). Consider the negative log of Z_{N+1}. Recall the prior is absolutely continuous with respect to reference measure η. Add and subtract g inside the exponent and simplify

$$E[-\log Z_{n+1}] = E[-\log E_{P_0}[e^{-\frac{\beta}{2}\|Y-f_w\|_{N+1}^2}] + \frac{N+1}{2}\log(\frac{2\pi}{\beta}) \tag{3.134}$$

$$= E[-\log E_{P_0}[e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2}] + \frac{\beta}{2}\|Y-g\|_{N+1}^2] + \frac{N+1}{2}\log(\frac{2\pi}{\beta}) \tag{3.135}$$

$$+ E[-\log\int\frac{p_0(w)e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2}}{E_{P_0}[e^{-\frac{\beta}{2}\|g-f_w\|_{N+1}^2}]}e^{-\beta\langle Y-g,g-f_w\rangle_{N+1}}\eta(dw)]. \tag{3.136}$$

The second and third terms in (3.135) will cancel with the first and second terms in the Kullback risk (3.133). Term (3.136) is the same expression as (3.100), and was shown to be less than 0. □

Theorem 3.5. *Let $g(x)$ be a target function with absolute value bounded by b and let $\tilde{g}$ be its $L_2(P_X)$ projection into the closure of the convex hull of signed neurons scaled by V. Let P_0 be the uniform prior on $(S_{1,M}^d)^K$. Assume the neuron activation function is odd symmetric and set all outer weights as $c_k = \frac{V}{K}$. Assuming the data distribution has $Y|X \sim$ Normal$(g(X), \frac{1}{\beta})$, with P_X having support in $[-1,1]^d$. We bound the expected*

Kullback divergence as

$$E[D(P_{Y|X}\|Q^{avg}_{Y|X,X^N,Y^N})] \leq \frac{MK\log(2d+1)}{N+1} + \beta\frac{a_0^2V^2}{2K} + \beta\frac{V(a_0V+b)a_2+V^2a_1^2}{2M} \tag{3.137}$$

$$+ \beta\|g-\tilde{g}\|^2. \tag{3.138}$$

In particular, with the choice

$$K^* = \frac{(\frac{\beta}{2}V^4)^{\frac{1}{3}}(a_0^2)^{\frac{2}{3}}}{(V(a_0V+b)a_2+V^2a_1^2)^{\frac{1}{3}}}\Big(\frac{(N+1)}{\log(2(d+1))}\Big)^{\frac{1}{3}} \tag{3.139}$$

$$M^* = \frac{(((a_0V+b)a_2+V^2a_1^2)^{\frac{2}{3}}(\frac{\beta}{2})^{\frac{1}{3}}}{(a_0V)^{\frac{2}{3}}}\Big(\frac{(N+1)}{\log(2(d+1))}\Big)^{\frac{1}{3}}, \tag{3.140}$$

we would have a bound of

$$3(\frac{\beta}{2})^{\frac{2}{3}}(a_0V)^{\frac{2}{3}}(V(a_0V+b)a_2+V^2a_1^2)^{\frac{1}{3}}\Big(\frac{\log(2d+1)}{N+1}\Big)^{\frac{1}{3}} + \beta\|g-\tilde{g}\|^2. \tag{3.141}$$

Proof. This proof follows much that same as the proof of Theorem 3.3. Let $\tilde{g}$ be the $L_2(P_X)$ projection of g into the closure of the convex hull of signed neurons scaled by V. Let $\tilde{g}_\epsilon$ be a specific finite width neural network that is within ϵ $L_2(P_X)$ distance of $\tilde{g}$. Add and subtract $\|g-\tilde{g}_\epsilon\|^2_{N+1}$ in the exponent of equation (3.121) to get the expression

$$\frac{E[-\log E_{P_0}[e^{-\frac{\beta}{2}(\|g-f_w\|^2_{N+1}-\|g-\tilde{g}_\epsilon\|^2_{N+1})}]]}{(N+1)} + \beta\frac{1}{2}\frac{E[\|\tilde{g}-g\|^2_{N+1}]}{N+1} + \beta 2b\epsilon^2 + \frac{1}{2}\beta\epsilon^2. \tag{3.142}$$

This is the same expression as (3.108), scaled by a β. Doing the same analysis gives the bound

$$\frac{MK\log(2d+1)}{(N+1)} + \beta\frac{a_0^2V^2}{2K} + \beta\frac{(V(a_0V+b)a_2+V^2a_1^2)}{2M} + \beta\|g-\tilde{g}\|^2. \tag{3.143}$$

Note now that β, being the inverse variance of the data distribution, is not a design parameter we can choose. However, M and K are modeling choices. Setting M^* and K^* as given yields the final expression. $\square$

3.5 Other Discrete Priors With Risk Control

We have worked with the discrete uniform prior on $(S^d_{1,M})^K$ and shown it has risk control via the index of resolvability argument 3.3. This approach decomposes the cumulant generating function into two terms, based on a subset A of weight values,

$$-\frac{\log P_0(A)}{\beta N} + \frac{1}{N}\max_{w\in A}\sum_{i=1}^{N}(\tilde{g}(x_i) - f_w(x_i))^2. \tag{3.144}$$

The first term is the minus log probability (or log of 1 over the probability) of the set A under the prior. The second term is determined by the worst regret of any point in the set A relative to $\tilde{g}$.

As our approximation results show 3.5, for any element of the closure of the convex hull $\tilde{g}$, there exists some element of $(S^d_{1,M})^K$ which has regret bounded by $O(N(\frac{1}{M}+\frac{1}{K}))$. Thus we can make our set A be this point and bound the worst regret over the set.

We must then consider the probability of this single point. The set $S^d_{1,M}$ is a set of sparse vectors. That is, with d locations that are integer multiples of $1/M$ that sum to 1 or less in absolute value, only $M << d$ of the coordinates can be non-zero at any time. Thus for any element of $S^d_{1,M}$, most of the coordinates are 0.

As we have shown, $|S^d_{1,M}| \leq (2d+1)^M$, so using a uniform prior we can have control over the minus log probability of any one point in $S^d_{1,M}$. Since our prior is product over $w_1, \cdots, w_K$, we pay a further factor of K in our log bound to have the bound

$$\max_{w\in(S^d_{1,M})^K} -\log P_0(w) \leq (MK)\log(2d+1). \tag{3.145}$$

Importantly, this grows only logarithmically in d and linearly in M and K. With M, K being a power of N such as $N^{1/4}$ we can get regret and risk bounds that decay in N.

Here we consider some other discrete priors that similarly have the bound

$$\max_{w \in S_{1,M}^d} -\log P_0(w) = O(M \log(d)). \tag{3.146}$$

These priors are also product over index k so pay a linear factor of K when considering the full vector of weights.

We define our distributions by first sampling positive integer vectors $m_k \in Z_+^d$ from a distribution Q, and then setting absolute values

$$|w_{k,j}| = \frac{m_{k,j}}{M}, j \leq d. \tag{3.147}$$

Then for those indexes j where $m_{k,j} \neq 0$, we assign a positive or negative sign with probability $1/2$. Note for the distributions Q we will no longer make it a strict condition that $\|w_k\|_1 \leq 1$, but under the Q we define $Q(\|w_k\|_1 \leq 1)$ is a very likely event, and we control the minimum probability of any element of $S_{1,M}^d$.

Thus with a distribution Q over the integer vectors m, the probability of any vector $w_k \in S_{1,M}^d$ is defined as

$$P_0(w) = \left(\frac{1}{2}\right)^{\sum_{j=1}^d 1\{w_j \neq 0\}} Q(M|w|). \tag{3.148}$$

Define the number of non-zero coordinates as

$$Z_+(w) = \sum_{j=1}^d 1\{w_j \neq 0\}. \tag{3.149}$$

Since the elements of $S_{1,M}^d$ are sparse, $Z_+(w) \leq M$ for $w \in S_{1,M}^d$. Then for any element

$w \in S^d_{1,M}$, we have bound on the prior probability

$$\max_{w \in S^d_{1,M}} -\log P_0(w) = \max_{w \in S^d_{1,M}} \left[Z_+(w) \log 2 + -\log Q(M|w|) \right]$$
$$\leq M \log 2 + \max_{w \in S^d_{1,M}} -\log Q(M|w|).$$

We now consider a few choices for the Q distribution on absolute values of the integer components. We require these distributions to satisfy the property,

$$\max_{w \in S^d_{1,M}} -\log Q(M|w|) = O(M \log(d)). \tag{3.150}$$

Consider that most of the coordinates of the $m = M|w|$ vector are 0, and the sum of the coordinate values is less than or equal to M. Thus, as a proxy for what our densities must look like, we would need marginally that the log-likelihood of each m_j coordinate is near linear with slope $-\log(d)$ and intercept $-M/d$,

$$\log Q(m_j) \approx -\frac{M}{d} - m_j \log(d). \tag{3.151}$$

This linear log-likelihood is describing a Geometric distribution, so we seek prior densities that are similar to independent Geometric random variable in each coordinate. If the joint probability of an m vector is approximately equal to the product the marginals (two of the priors we consider are iid in each coordinate so this is exact, one is a multinomial which for large d is near independent in each coordinate), then this linear marginal log-likelihood form (3.151) is the correct scaling to give us the prior probability we require,

$$-\log Q(m) \approx -\sum_{j=1}^{d} \log Q(m_j) \approx M + (\sum_{j=1}^{d} m_j) \log(d) = M(\log(d) + 1). \tag{3.152}$$

Thus, we will consider several distributions Q where the marginal distribution of each

coordinate $m_j = M|w_j|$ is near a geometric distribution. These include a Multinomial, Geometric, and Poisson distribution that approximately follow this log probability and have $-\log Q(M|w|) = O(M\log(d))$ points for $w \in S^d_{1,M}$.

3.5.1 Multinomial, Geometric, and Poisson Distributions

We want a prior which makes any sparse vector likely. That is, any vector of d positive integer values where less than M locations are non zero and $\sum_{j=1}^d m_j \leq M$. We consider different discrete distributions and scale them so that their expected sum of coordinates is approximately M, $E_Q[\sum_{j=1}^d m_j] \approx M$. Under this scaling, we show these distributions satisfy the desired log probability bound we desire.

A natural choice is the Symmetric Multinomial distribution in $d+1$ dimensions that sums to M. By symmetric we mean all probabilities in each $d+1$ coordinate are $1/(d+1)$. Define $m_{d+1} = M - \sum_{j=1}^d m_j$ to define a vector which exactly sums to 1 in $d+1$ coordinates. Then the symmetric Multinomial in $d+1$ dimensions has the pmf

$$\text{Sym Multinomial}_{M,d+1}(m) = (\frac{1}{d+1})^M \binom{M}{m_1, \cdots, m_{d+1}} 1\{\sum_{j=1}^{d+1} m_j = M\} 1\{m \in (Z_{\geq 0})^{d+1}\}. \tag{3.153}$$

This can be thought of as sampling $j \in \{1, \cdots, d+1\}$ locations M times with replacement and assigning 1 unit to the location j selected. There are $(d+1)^M$ ordered ways to select the locations, and different permutations of the same selection of indexes gives rise to the same m vector. Thus every possible vector m which sums to M in $d+1$ dimensions has at least one ordered way to select indexes, so the probability of any one point is lower bound as

$$\min_{m \in Z^{d+1}_{\geq 0}, \sum_{j=1}^{d+1} m_j = M} \left[\text{Sym Multinomial}_{M,d+1}(m)\right] \geq (d+1)^{-M}, \tag{3.154}$$

which gives upper bound on the minus log probability

$$\max_{m \in Z_{\geq 0}^{d+1}, \sum_{j=1}^{d+1} m_j = M} -\log \left[\text{Sym Multinomial}_{M,d+1}(m) \right] \leq M \log(d+1). \tag{3.155}$$

Note that the coordinates m_j are then marginally Binomial$(M, \frac{1}{d+1})$. This choice of Q is natural and gives sufficiently large probability to any M sparse vector, however the support of Q is restricted the constrained set $\{m \in \mathbb{Z}_{\geq 0}^{d+1} : \sum_{j=1}^{d+1} m_j = M\}$. Also, the different coordinates m_j are not quite independent. We would much prefer a product prior where each coordinate m_j is sampled iid from a product distribution, and is allowed to have unconstrained support. This means that the resulting w vector will not be forced to live in $S_{1,M}^d$, but rather each coordinate w_j can be any integer multiple of $1/M$ and the coordinates do not have to sum to 1. However, we want our priors on Q to make any element of $S_{1,M}^d$ likely, so instead of a hard constraint that the m vector coordinates sum to 1, we scale the distribution so their expected sum is 1, $E_Q[\sum_{j=1}^d m_j] = M$, which since they are iid means $E_Q[m_j] = M/d$.

One choice is to have the coordinates m_j distributed independently from a Geometric distribution with parameter $(1 - \frac{M}{d})$. Then $E[\sum_{j=1}^d m_j] = M$ so generating a vector which sums to 1 is not a hard constraint as in the previous priors, but a highly likely event. Indeed, checking the joint density,

$$Q(m) = \prod_{i=1}^{d} (1 - \frac{M}{d}) \frac{M}{d}^{m_j} = (1 - \frac{M}{d})^d \frac{M}{d}^{\sum_j m_j}. \tag{3.156}$$

We see that the density is a function of the sum of the coordinates. Since each $w \in S_{1,M}^d$ has $\sum_{j=1}^d M|w_j| \leq M$, we would have $\sum_{j=1}^d m_j \leq M$. This gives lower bound on the

probability,

$$\min_{m \in Z_{\geq 0}^{d+1}, \sum_{j=1}^{d+1} m_j \leq M} \big[\text{Geometric}_{1-M/d}(m)\big] \geq (1 - \frac{M}{d})^d \Big(\frac{M}{d}\Big)^M \tag{3.157}$$

$$= \Big(\frac{M}{d}\Big)^M \Big(\Big(1 - \frac{1}{\frac{d}{M}}\Big)^{\frac{d}{M}}\Big)^M \tag{3.158}$$

$$\approx \Big(\frac{M}{d}\Big)^M e^{-M}. \tag{3.159}$$

This upper bounds the minus log probability,

$$\max_{m \in Z_{\geq 0}^{d+1}, \sum_{j=1}^{d+1} m_j \leq M} \big[-\log \text{Geometric}_{1-M/d}(w)\big] \leq M \log \frac{d}{M} - d \log(1 - \frac{M}{d}). \tag{3.160}$$

By standard log inequalities,

$$-\log(1 - \frac{M}{d}) \leq \frac{\frac{M}{d}}{1 - \frac{M}{d}} \quad \text{for } 0 \leq \frac{M}{d} \leq 1, \tag{3.161}$$

which gives rise to the upper bound

$$\max_{m \in Z_{\geq 0}^{d+1}, \sum_{j=1}^{d+1} m_j \leq M} \big[-\log \text{Geometric}_{1-M/d}(m)\big] \leq \log M \frac{d}{M} + M \frac{1}{1 - \frac{M}{d}} \tag{3.162}$$

$$\approx M[\log(d) - \log(M) + 1]. \tag{3.163}$$

A Poisson($\frac{M}{d}$) distribution for each m_j coordinate will also have the properties we desire. Again $E_Q[\sum_{j=1}^d m_j] = M$ and the product density has the structure

$$\text{Poisson}_{M/d}(m) = \prod_{j=1}^d \frac{\frac{M}{d}^{m_j} e^{-\frac{M}{d}}}{m_j!} = \Big(\frac{\frac{M}{d}^{\sum_{j=1}^d m_j}}{(\sum_{j=1}^d m_j)!} e^{-M}\Big)\Big(\frac{(\sum_{j=1}^d m_j)!}{m_1! \cdots m_d!}\Big). \tag{3.164}$$

From our previous discussion on the symmetric multinomial, we have lower bound

$$\min_{m \in Z^d_{\geq 0}: \sum_{j=1}^d m_j = M} \frac{M!}{m_1! \cdots m_d!} \geq (d/2)^{-M}. \tag{3.165}$$

This gives lower bound on the joint Poisson probability

$$\min_{m \in Z^d_{\geq 0}: \sum_{j=1}^d m_j = M} [\text{Poisson}_{M/d}(m)] \geq \frac{\left(\frac{d}{M}\right)^{-M} e^{-M}}{M!} (\frac{d}{2})^{-M}, \tag{3.166}$$

and minus log probability upper bound

$$\max_{m \in Z^d_{\geq 0}: \sum_{j=1}^d m_j = M} [-\log \text{Poisson}_{M/d}(m)] \leq M \log \frac{d}{M} + \log(M!) + M \log(\frac{d}{2}) + M.$$

Apply a Stirling's bound, this is equal to

$$= M[2\log(d) - \log(M) - \log(2)] + M\log(M) - M + O(\log(M)) + M \tag{3.167}$$

$$= M[2\log(d) - \log(2)] + O(\log(M)). \tag{3.168}$$

We summarize the results here. Recall that we are looking for discrete prior distributions with the property

$$\max_{w \in S^d_{1,M}} -\log Q(M|w|) = O(M \log(d)). \tag{3.169}$$

Our proxy for what these distributions should look like is iid in each coordinate, with marginal log-likelihoods that are near linear (i.e. near a Geometric distribution in each coordinate). For each coordinate, the marginal should be approximately approximately

linear with slope $-\log(d)$, and intercept $-\frac{M}{d}$,

$$\log Q(m_j) \approx \log Q(m_j = 0) - m_j \log(d) \tag{3.170}$$

$$\log Q(m_j = 0) \approx -\frac{M}{d}. \tag{3.171}$$

We see in this table, the distributions we have proposed have marginal log-likelihoods that are approximately of this linear form. The Geometric is the most similar to our proxy being iid in each coordinate with exact linear log-likelihood. All have a bound on (3.169) of the desired order. We also see in Figure 3.1 for $M = 20$ and $d = 10000$ the log-likelihoods of the proposed densities are near the linear proxy (black line) we conjectured they should be similar to.

Distribution	Proxy	Symmetric Multinomial$(M, d+1)$
m_j Marginal	-	Binomial$(M, 1/(d+1))$
$\log Q(m_j)$	$-\frac{M\log(d)}{d} - m_j \log(d)$	$M\log(1-1/(d+1)) + \log\binom{M}{m_j} - m_j\log(d)$
$\log Q(m_j = 0)$	$-\frac{M\log d}{d}$	$M\log(1-1/(d+1))$
Max Value	$M[\log(d)+1]$	$M\log(d+1)$

Table 3.1: Summary of Discrete Prior Likelihoods

Distribution	Geometric$(1-M/d)$	Poisson(M/d)
m_j Marginal	-	-
$\log Q(m_j)$	$\log(1-\frac{M}{d}) - m_j\log(\frac{d}{M})$	$-\frac{M}{d} - \log(m_j!) - m_j\log(\frac{d}{M})$
$\log Q(m_j = 0)$	$\log(1-\frac{M}{d})$	$-\frac{M}{d}$
Max Value	$M[\log(d) - \log(M) + 1]$	$M[2\log(d) - \log(2)] + O(\log(M))$

Table 3.2: Summary of Discrete Prior Likelihoods

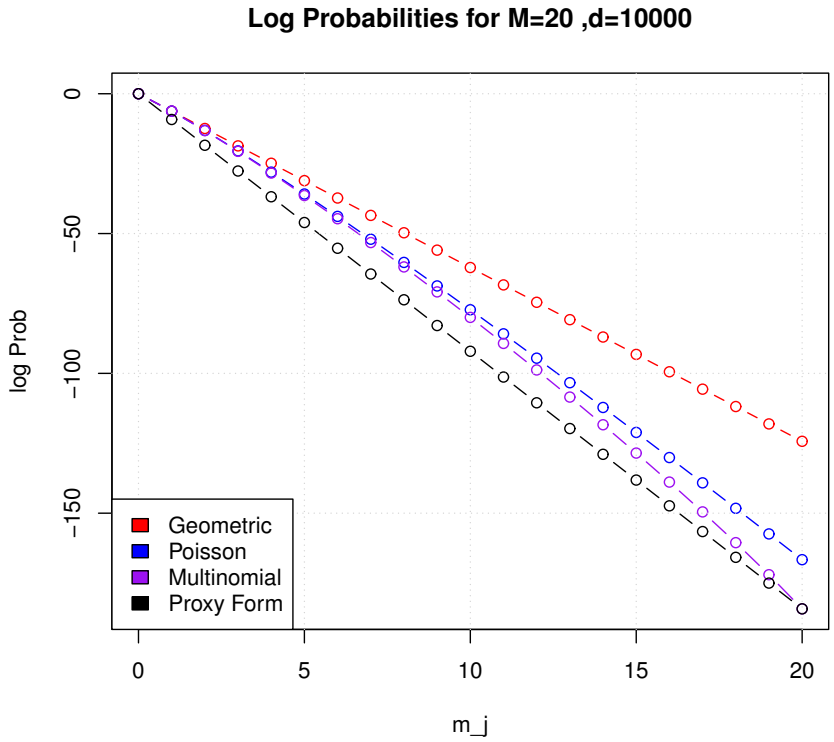

Figure 3.1: Plot of Log Prior Probabilities for Different Discrete Priors

3.6 Appendix: Proofs of Additional Lemmas

Here, we present the proofs of results that were too long or tedious to include in the main body of the chapter.

3.6.1 Improved $1/M^2$ Regret Proofs

Proof of Lemma 3.6:

Proof. Here we show that when the y_i observations are direct outputs of a neural network, we can give an improved $1/M^2$ regret control.

Fix $x_1, \cdots, x_n$ and $h(x_1), \cdots, h(x_N)$ (or more generally $h_1, \cdots, h_N$). Since h lives

in the closure of the convex hull of signed neurons scaled by V, for every $\epsilon > 0$ there exists some finite width neural network with continuous-valued weight vectors $w_\ell \in S_1^d$ and outer weights c_ℓ with $\sum_\ell |c_\ell| = 1$ such that

$$\tilde{h}(x) = V \sum_\ell c_\ell \psi(x \cdot w_\ell), \quad \sum_{i=1}^N (h(x_i) - \tilde{h}(x_i))^2 \leq \epsilon. \tag{3.172}$$

Let L be a random draw of neuron index where $L = \ell$ with probability $|c_\ell|$. Define $w^{\text{cont}} = w_L$ as the continuous neuron vector at the selected random index L, and $s^{\text{cont}} = \text{sign}(c_L)$ as the sign of the outer weight.

Given a continuous vector w^{cont} of dimension d, we then make a random discrete vector as follows. Define a $d+1$ coordinate, $w_{d+1}^{\text{cont}} = 1 - \|w_{1:d}^{\text{cont}}\|_1$, to have a $d+1$ length vector which sums to 1. Consider a random index $J \in \{1, \cdots d+1\}$ where $J = j$ with probability $|w_j^{\text{cont}}|$. Given w^{cont}, this defines a distribution on $\{1, \cdots, d+1\}$. Draw M iid random indices $J_1, \cdots, J_M$ from this distribution and define the counts of each index

$$m_j = \sum_{i=1}^M 1\{J_i = j\}. \tag{3.173}$$

We then define the discrete vector $w^{\text{disc}} \in S_{1,M}^d$ with coordinate values

$$w_j^{\text{disc}} = \text{sign}(w_j^{\text{cont}}) \frac{m_j}{M}. \tag{3.174}$$

Consider then K iid draws of random indexes $L_1, \cdots L_K$, as well as corresponding signs $s_k = \text{sign}(c_{L_k})$. For each L_k consider M iid drawn indexes $J_1^k, \cdots, J_M^k$. This also defines continuous vectors w_k^{cont} and discrete vectors w_k^{disc}. Denote the neural network using a random set of weights and signs,

$$f(x, w, s) = \sum_{k=1}^K \frac{V}{K} s_k \psi(x \cdot w_k). \tag{3.175}$$

Recall the empirical norm and inner product definitions $\|\cdot\|_N^2, \langle\cdot,\cdot\rangle_N$ from the notation section. Consider the expected regret using random discrete neuron weights.

$$E\Big[\|h - f(\cdot, w^{\text{disc}}, s)\|_N^2\Big]. \tag{3.176}$$

Note this expectation is with respect to the previously defined distribution for w^{disc}, w^{cont}, and s. The data $(x_i)_{i=1}^N$ are fixed. Using a bias variance decomposition, this is equal to

$$E\Big[\|f(\cdot, w^{\text{disc}}, s) - E[f(\cdot, w^{\text{disc}}, s)]\|_N^2\Big] + \|h - E[f(\cdot, w^{\text{disc}}, s)]\|_N^2 \tag{3.177}$$

The first term is the variance of an average of K iid random variables bounded by $a_0 V$, and thus will have a $1/K$ order

$$E\Big[\|f(\cdot, w^{\text{disc}}, s) - E[f(\cdot, w^{\text{disc}}, s)]\|_N^2\Big] \tag{3.178}$$

$$= \sum_{i=1}^{N} E[\big(\sum_{k=1}^{K} \frac{V}{K}(s_k \psi(x_i \cdot w_k) - E[s_k \psi(x_i \cdot w_k)])\big)^2] \tag{3.179}$$

$$\leq N \frac{a_0^2 V^2}{K} \tag{3.180}$$

Then for the bias term. Add and subtract $\tilde{h}$, the specific finite neural neural net that is ϵ close to h, inside the square. We have,

$$\|h - E[f(\cdot, w^{\text{disc}}, s)]\|_N^2 = \|\tilde{h} - E[f(\cdot, w^{\text{disc}}, s)]\|_N^2 \tag{3.181}$$

$$+ 2\langle h - \tilde{h}, \tilde{h} - E[f(\cdot, w^{\text{disc}}, s)]\rangle_N + \|h - \tilde{h}\|_N^2 \tag{3.182}$$

$$\leq \|\tilde{h} - E[f(\cdot, w^{\text{disc}}, s)]\|_N^2 + 4\sqrt{\epsilon}\sqrt{N}(a_0 V) + \epsilon \tag{3.183}$$

Then recall $\tilde{h}$ is defined by a specific set of weights c_ℓ whose absolute values sum to 1. The weights c_ℓ also define the probabilities of the w_k^{cont} begin equal to w_ℓ. Thus thus this

difference of expectations can be made a common sum over $|c_\ell|$.

$$\|\tilde{h} - E[f(\cdot, w^{\text{disc}}, s)]\|_N^2 \tag{3.184}$$

$$= V^2 \sum_{i=1}^{N} \Big(\sum_{\ell} |c_\ell| \text{sign}(c_\ell) [\psi(x_i \cdot w_\ell) - E[\psi(x_i \cdot w_1^{\text{disc}}) | w_1^{\text{cont}} = w_\ell]] \Big)^2 \tag{3.185}$$

Then, noting that $w_1^{\text{disc}} - w_1^{\text{cont}}$ is mean 0 under the conditional distribution, we may add in $(x_i \cdot w_1^{\text{disc}} - x_i \cdot w_1^{\text{cont}})\psi'(x_i \cdot w_1^{\text{cont}})$, which is the first order Taylor expansion of $\psi(x_i \cdot w_1^{\text{disc}}) - \psi(x_i \cdot w_1^{\text{cont}})$. We then have

$$V^2 \sum_{i=1}^{N} \Big(\sum_{\ell} |c_\ell| \text{sign}(c_\ell) E[(x_i \cdot w_1^{\text{disc}} - x_i \cdot w_1^{\text{cont}})\psi'(x_i \cdot w_1^{\text{cont}}) \tag{3.186}$$

$$+ \psi(x_i \cdot w_1^{\text{disc}}) - \psi(x_i \cdot w_\ell) | w_1^{\text{cont}} = w_\ell] \Big)^2. \tag{3.187}$$

Then take an absolute value inside the expectation to upper bound. By a second order Taylor expansion and $|\psi''(z)| \leq a_2 \forall\, z \in [-1, 1]$ we have the following bound

$$\begin{aligned} &\left| (x_i \cdot w_1^{\text{disc}} - x_i \cdot w_1^{\text{cont}})\psi'(x_i \cdot w_1^{\text{cont}}) + \psi(x_i \cdot w_1^{\text{disc}}) - \psi(x_i \cdot w_\ell) \right| \\ &\leq \frac{1}{2} a_2 (x_i \cdot w_1^{\text{disc}} - x_i \cdot w_1^{\text{cont}})^2 \end{aligned}$$

Noting that $E[w_1^{\text{disc}} | w_1^{\text{cont}}] = w_1^{\text{cont}}$, we have a squared sum of variances,

$$\frac{1}{4} a_2^2 V^2 \sum_{i=1}^{N} (\sum_{\ell} |c_\ell| \text{Var}[x_i \cdot w_1^{\text{disc}} | w_1^{\text{cont}} = w_\ell])^2$$

For a fixed choice of continuous w_1^{cont}, let $x_{i,d+1} = 0$ and consider x_i as a $d+1$ dimension vector. Then $x_i \cdot w_1^{\text{disc}}$ is the inner product of x_i with a vector defined by counts of the independent random indexes $J_1^1, \cdots, J_M^1$. Therefore, the inner product can equivalently

be written as an average of M iid random variables using these indexes,

$$\text{Var}[x_i \cdot w_1^{\text{disc}} | w_1^{\text{cont}}] = \text{Var}[\frac{1}{M} \sum_{t=1}^{M} x_{i,J_t^1} | w_1^{\text{cont}}] \tag{3.188}$$

$$= \frac{1}{M} \text{Var}[x_{i,J_1^1} | w_1^{\text{cont}}] \tag{3.189}$$

$$\leq \frac{1}{M}, \tag{3.190}$$

since the $|x_{i,j}|$ are all bounded by 1. Taking $\epsilon \to 0$, we conclude, under the distribution we have defined on $(S_{1,M}^d)^K$,

$$E\Big[\|h - f(\cdot, w^{\text{disc}}, s)\|_N^2\Big] \leq N\frac{a_0^2 V^2}{K} + N\frac{a_2^2 V^2}{4M^2} \tag{3.191}$$

□

Proof of Corollary 3.4:

Proof. This proof follows much the same as the proof of Theorem 3.3. Follow the same steps of the proof up to equation (3.101) where we have the expression

$$\frac{E_{P_{X^{N+1},Y^{N+1}}}[-\log \int p_0(w) e^{-\frac{\beta}{2}\|g - f_w\|_{N+1}^2} \eta(dw)]}{\beta(N+1)}. \tag{3.192}$$

At this point, note that g itself is assumed th live in the $L_2(P_X)$ closure of the convex Hull of signed neurons scaled by V. Thus, let $\tilde{g}_\epsilon$ be some finite convex combination of neurons scaled by V which is ϵ close to g itself in $L_2(P_X)$ distance. Add and subtract $\tilde{g}_\epsilon$ inside the norm in the exponent, then by a Cauchy-Schwarz inequality we have upper bound

$$\frac{E_{P_{X^{N+1},Y^{N+1}}}[-\log \int p_0(w) e^{-\frac{\beta}{2}\|\tilde{g}_\epsilon - f_w\|_{N+1}^2} \eta(dw)]}{\beta(N+1)} + 4a_0 V\sqrt{\epsilon} + \epsilon. \tag{3.193}$$

Apply Lemma 3.6 to the approximation in the exponent, rather than Lemma 3.5 which gives rise the the $1/M^2$ term in place of the $1/M$. The rest of the proof follows as in Theorem 3.3 noting that now since g is in the closure of the convex Hull of signed neurons scaled by V we have bound on g of $b = a_0 V$. □

Chapter 4

Log-Concave Coupling for Greedy Bayes

4.1 Introduction

Our model for a neural network with K neurons and internal weight vectors of dimension d results in a parameterized function with Kd parameters overall (note we fix the outer weights of our network at the start as either $c_k = \pm\frac{V}{K}$ and these are not a parameter to train). In the previous sections, we constructed a posterior distribution, or rather a sequence of posteriors using different subsets of the data $n \leq N$, that was a joint distribution on all Kd parameters at once. Our prior treated each d dimensional neuron weight vector w_k as independent, and the different K neurons are then coupled in the log-likelihood via their joint ability to fit the observed data y,

$$p_n(w_1, \ldots, w_k | x^n, y^n) \propto \Big[\prod_{k=1}^{K} p_0(w_k)\Big] \exp\Big(-\frac{\beta}{2} \sum_{i=1}^{n} (y_i - \sum_{k=1}^{K} c_k \psi(x_i \cdot w_k))^2 \Big). \quad (4.1)$$

This is a natural way to set up a Bayesian model by defining a likelihood that involves all the parameters at once. However, sampling algorithms have polynomial mixing time

bounds dependent on the number of dimensions in the distribution. Thus, by sampling all Kd parameters at once, we pay a polynomial price in the number of neurons K and the internal dimension d. Furthermore, our risk control results for the joint Bayesian model showed square risk is bounded by $O([(\log d)/N]^{1/4})$.

The original direction planned for this research was to build a *greedy Bayes* procedure, training the neurons one at a time in order based on the residuals of previous fits. This was inspired by past work on greedy optimization.

The best known theoretical guarantees for neural network risk control are not for greedy optimization, but for jointly optimizing all neuron weights jointly. In [11], it is shown if one can optimize all weights of the network at once to minimize the training loss, then a network with $K = O([N/(\log d)]^{1/2})$ neurons achieves statistical risk control of the order $O([(\log d)/N]^{1/2})$.

However, there does not currently exist an algorithm that can optimize all neurons at once in polynomial dependency on N and d as they grow large. Thus, in the pursuit of a feasible computation algorithm in high dimensions that achieve risk near $O([(\log d)/N]^{1/2})$, greedy optimization procedures that train one neuron at a time have been investigated as an alternative to joint optimization.

In [9, 31], greedy optimization is shown to achieve a risk of order $O([(d \log(N))/N]^{1/2})$. This is only useful for low dimensional problems with $d < N$, that is not over parameterized. For high-dimensional problems, [35] shows greedy optimization achieves risk control of the order $O([(\log d)/N]^{1/3})$, close to the $1/2$ power achievable by joint optimization (if one was actually able to implement either algorithm).

However, similar to the joint optimization problem, even in the greedy optimization problem there does not exist a known polynomial time algorithm, despite bound on the theoretical risk performance. Thus, we endeavor to replace greedy optimization with greedy sampling (one neuron at a time), to produce a method with risk control as well as computational ability via sampling. We achieve a risk control of $O([(\log d)/N]^{1/3})$ for

our greedy Bayes procedure, which matches the greedy optimization order of risk control.

In the joint sampling problem, we have N posteriors to sample from, each with a random variable of dimension Kd. In the greedy construction, we have NK sampling problems, yet each is only for a random variable with dimension d, (each weight vector w_k is sampled one at a time, and once for each subset $n \leq N$ of the data). This has potential benefits in the number of iterations required to compute the posterior averages, since the MCMC sampling complexity will only be polynomial in d and not K.

Additionally, our risk control for the greedy Bayes problem proves a bound of order $O([(\log d)/N]^{1/3})$, which is better than the $1/4$ power we achieved for the joint sampling problem. We now introduce the specifics of how the greedy Bayes estimator is constructed via a series of recursively defined densities.

4.2 Construction of the Greedy Bayes Estimator

Consider a set of training data $(x_i, y_i)_{i=1}^N$, initialize a fit for each data point and a residual value

$$\hat{f}_{n,0}(x) = 0 \quad \forall n \in \{1, \ldots, N\} \tag{4.2}$$

$$r_{n,0} = y_n. \tag{4.3}$$

Note the initial fit does not have to be set to be 0 for each data point, we set it as this for simplicity in the explanation. We wish to improve this fit by creating a linear combination with 1 new additional neuron.

There are a few important parameters to define the problem. As before, $\beta > 0$ will represent a gain we use in our posterior density. However, we now introduce an $\alpha \in (0, 1)$ as a "mixture weight" to combine the old fit and the new neuron, as well as a sign choice $s \in \{-1, 1\}$. One should think of α as small, something like $1/N^{1/3}$. As we will see, β

does not seem to be as critical a parameter in the greedy problem as it does in the joint sampling problem, since we now have a small α value to help control the size of the log-likelihood. Thus in many cases, it is fine to think of $\beta = 1$ and α as a fractional power of $1/N$, perhaps with log factors (we will see $\alpha = \frac{\log(K)}{K}$ is a good choice for risk control later on, with $K \approx N^{1/3}$).

Define a prior P_0 on neuron weights w and signs s, this prior has support on $(S_1^d) \times \{-1, 1\}$. Let η be a reference measure on $S_1^d \times \{-1, 1\}$. This may be either Lebesgue measure on S_1^d cross counting measure on $\{-1, 1\}$, or when we consider discrete support sets may be counting measure on $S_{1,M}^d \times \{-1, 1\}$. In either case, assume the prior P_0 has density p_0 with respect to this reference measure. Then for each $n \leq N$, define a posterior for new neuron weight vector w and sign s using the first n residuals

$$\ell_{n,1}(w, s) = \frac{1}{2} \sum_{i=1}^{n} (r_{i,0} - \alpha s V \psi(x_i \cdot w))^2 \tag{4.4}$$

$$p_{n,1}(w, s | x^n, r_0^n) = \frac{p_0(w, s) e^{-\beta \ell_{n,1}(w,s)}}{E_{P_0}[e^{-\beta \ell_{n,1}(w,s)}]}. \tag{4.5}$$

Note that if our neural network uses odd symmetric neurons such as tanh, then the use of the sign is not needed and we can consider $P_0(s = 1) = 1$ and the sign is fixed to be positive for all neurons.

The updated fit is then a linear combination of the old fit and the posterior mean using the mixture weight α,

$$\hat{f}_{n,1}(x) = (1 - \alpha) \hat{f}_{n,0}(x) + \alpha V E_{P_{n,1}}[s \psi(x \cdot w) | x^n, r_0^n]. \tag{4.6}$$

For each index n, a new residual is defined as the point y_n minus $(1-\alpha)$ $\hat{f}_{n-1,1}(\cdot)$ evaluated

at x_n,

$$r_{n,1} = y_n - (1-\alpha)\hat{f}_{n-1,1}(x_n). \tag{4.7}$$

Recursively, given a vector of residuals at level k, $r_k^n = (r_{1,k}, \ldots, r_{n,k})$, define a new posterior density at level $k+1$ as follows,

$$\ell_{n,k+1}(w,s) = \frac{1}{2}\sum_{i=1}^{n}(r_{i,k} - \alpha s V \psi(x_i \cdot w))^2 \tag{4.8}$$

$$p_{n,k+1}(w,s|x^n, r_k^n) = \frac{p_0(w)e^{-\beta \ell_{n,k+1}(w,s)}}{E_{P_0}[e^{-\beta \ell_{n,k+1}(w,s)}]}. \tag{4.9}$$

These posteriors are defined for every $0 \leq n \leq N$. The posterior at $n = 0$ is just the prior. Note that the posterior at level $k+1$ is defined by residuals at level k. For example, the posterior at level 2 is defined by the residuals from level 1. Based on this posterior, define an updated fit at level $k+1$ and residuals at level $k+1$ as follows,

$$\hat{f}_{n,k+1}(x) = (1-\alpha)\hat{f}_{n,k}(x) + \alpha V E_{P_{n,k+1}}[s\psi(x \cdot w)|x^n, r_k^n] \tag{4.10}$$

$$r_{n,k+1} = y_n - (1-\alpha)\hat{f}_{n-1,k+1}(x_n). \tag{4.11}$$

Using this procedure, we define a set of posterior mean estimators $\hat{f}_{n,k}$ one for each index $n \in \{0, \ldots, N\}$ and each neuron $k \in \{1, \ldots, K\}$. The posterior densities $p_{n,k+1}(w, s|x^n, r_k^n)$ are function of x^n and r_k^n, and the r_k^n are a ultimately a function of the (x^n, y^n). So each $p_{n,k+1}(w, s|x^n, r_k^n)$ is not a function of (x_i, y_i) values for index $i > n$.

At level k, each residual $r_{n,k}$ is a function of data up to index n, x^n, y^n. Each new fit function $\hat{f}_{n,k+1}$ is a function of the residuals of the previous level residuals up to index n, and then the new residual is $\hat{f}_{n-1,k+1}$ evaluated at x_n. Thus the new set of residuals at level $k+1$ still maintains that $r_{n,k+1}$ is a function of x^n, y^n. A dependence diagram is seen in Figure 4.1.

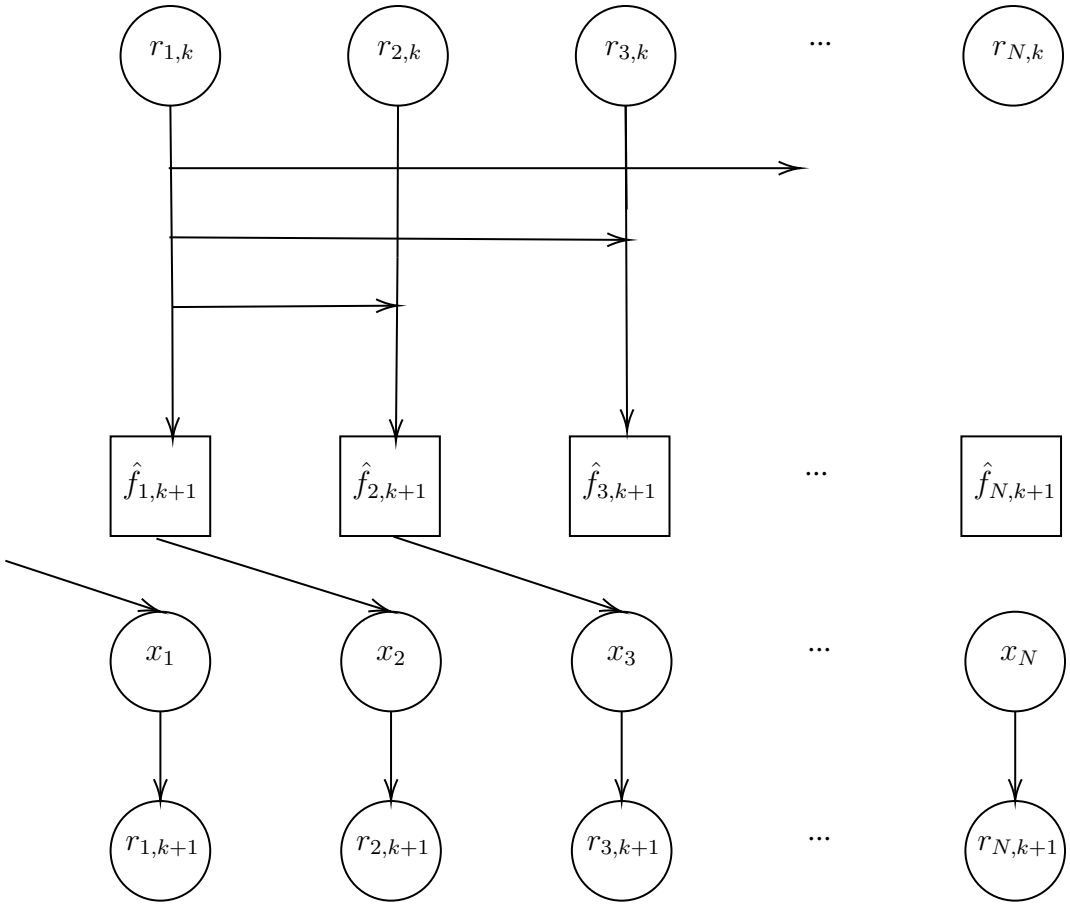

Figure 4.1: Flow Diagram for Recursive Greedy Fits

We then define an overall estimator as the Cesàro average of the level K estimators,

$$\hat{g}(x) = \frac{1}{N+1} \sum_{n=0}^{N} \hat{f}_{n,K}(x). \tag{4.12}$$

The recursive structure can also be decomposed into a specific linear combination of the individual posterior means

$$\hat{g}(x) = \frac{1}{N+1} \sum_{n=0}^{N} \Big[\alpha V \sum_{k=1}^{K} (1-\alpha)^{K-k} E_{P_{n,k}}[s\psi(x \cdot w) | x^n, r_{k-1}^n] + (1-\alpha)^K \hat{f}_{n,0}(x) \Big]. \tag{4.13}$$

Note we have set our initial fit to be $\hat{f}_{n,0}(x) = 0$, but this does not have to be the case.

We may also be interested in the individual $\hat{f}_{n,k}$ estimators themselves rather than just the Cesàro average of the level K estimators. Given an arbitrary sequence of data $(x_i, y_i)_{i=1}^N$, $\hat{f}_{n-1,k}(\cdot)$ evaluated at x_n can be used in online learning problem. That is, data up to index $n-1$ being used as a predictor for data at index n, and we can study the overall regret. For a competitor function g, we will be interested in regrets of the form

$$\bar{R}_{N,k}^{\text{square}} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{2} \big[(y_n - \hat{f}_{n-1,k}(x_n))^2 - (y_n - g(x_n))^2 \big]. \tag{4.14}$$

In fact, it is the study of these online regret objects that is the core of our analysis. Taking expectations, it provides for Cesàro averages of the statistical risks of $\hat{f}_{n,K}$ and thence for the risk of the Cesàro average $\hat{g}$.

However, theoretical bounds on greedy risk and regret are discussed in the next chapter, here we focus on computational concerns and forming a log-concave coupling which allows for sampling of the various posterior means.

From a computational perspective, we initialize with a set of initial fits $\hat{f}_{n,0}$ and initial residuals $r_{n,0}$. Then, a set of L iid samples can be made from each $P_{n,1}$ distribution, denote these draws $(w_{n,1}^{\ell}, s_{n,1}^{\ell})_{\ell=1}^{L}$. From these, new estimated residuals at level 1 can be computed,

$$\hat{r}_{n,1} = y_n - (1-\alpha)\hat{f}_{n,0}(x_n) - \alpha V \sum_{\ell=1}^{L} \frac{1}{L} s_{n-1,1}^{\ell} \psi(x_n \cdot w_{n-1,1}^{\ell}). \tag{4.15}$$

With this new set of $\hat{r}_{n,1}$ estimated residuals, draws of $(s_{n,2}^{\ell}, w_{n,2}^{\ell})_{\ell=1}^{L}$ can be made from $P_{n,2}(\cdot | x^n, \hat{r}_1^n)$. This recursion can be continued to produce samples $(s_{n,k}^{\ell}, w_{n,k}^{\ell})$ for $n \in \{0, \ldots, N\}, k \in \{1, \ldots, K\}, \ell \in \{1, \ldots, L\}$. Then the overall estimator can be written as

a combination of the empirical averages,

$$\hat{g}_{\text{empirical}}(x) = \frac{1}{N+1}\sum_{n=0}^{N}\Big[\alpha V\sum_{k=1}^{K}(1-\alpha)^{K-k}\big(\sum_{\ell=1}^{L}\frac{1}{L}s_{n,k}^{\ell}\psi(x\cdot w_{n,k}^{\ell})\big)+(1-\alpha)^{K}\hat{f}_{n,0}(x)\Big]. \tag{4.16}$$

Of course, empirical averages are only estimates of the true theoretical average they are approximating. Thus, each estimated residual $\hat{r}_{n,k}$ is only an approximation of $r_{n,k}$, so there are cascading errors in this method of recursively defining our posterior densities empirically. Thus the $\hat{g}_{\text{empirical}}$ which we would compute in practice is not equal to the theoretical $\hat{g}$, which may lead to the joint sampling problem of the previous chapters being more accurate in practice since it does not have the same cascading sequence of approximation errors that the empirical greedy estimator does. We leave the consideration of $\hat{g}_{\text{empirical}}$ vs $\hat{g}$ as a study for future work, and instead consider how to sample from the $P_{n,k}(\cdot|x^n, r_{k-1}^n)$ distributions, as well as to provide risk control for $\hat{g}$ itself.

4.3 Posterior Sign Probability

In this section, we consider the structure of the posterior densities we describe. Given a vector of residuals $r_{k-1}^n = (r_{i,k-1})_{i=1}^n$ and a set of input data points $x^n = (x_i)_{i=1}^n$, our densities (with respect to reference measure η which may be considered the product measure of Lebesgue with counting measure on $\{-1,1\}$) are of the form

$$p_{n,k}(w,s|x^n,r_{k-1}^n) = \frac{p_0(w,s)e^{-\frac{\beta}{2}\sum_{i=1}^{n}(r_{i,k-1}-s\alpha V\psi(x_i\cdot w))^2}}{\int p_0(w,s)e^{-\frac{\beta}{2}\sum_{i=1}^{n}(r_{i,k-1}-s\alpha V\psi(x_i\cdot w))^2}\eta(dw,ds)}. \tag{4.17}$$

Our prior is uniform over weight choices and sign values, and treats each as independent. Thus we may consider the integral conditioned on $s=1$ and $s=-1$ separately, and the

posterior probability of $s = -1, 1$,

$$p_{n,k}(w, s|x^n, r_{k-1}^n) = p_{n,k}(w|s, x^n, r_{k-1}^n)p_{n,k}(s|x^n, r_{k-1}^n). \tag{4.18}$$

where

$$p_{n,k}(w|s, x^n, r_{k-1}^n) = \frac{p_0(w)e^{-\frac{\beta}{2}\sum_{i=1}^n (r_{i,k-1} - s\alpha V\psi(x_i \cdot w))^2}}{\int p_0(w)e^{-\frac{\beta}{2}\sum_{i=1}^n (r_{i,k-1} - s\alpha V\psi(x_i \cdot w))^2}\eta(dw)} \tag{4.19}$$

$$p_{n,k}(s|x^n, r_{k-1}^n) = \frac{\int p_0(w)e^{-\frac{\beta}{2}\sum_{i=1}^n (r_{i,k-1} - s\alpha V\psi(x_i \cdot w))^2}\eta(dw)}{\int p_0(w)[e^{-\frac{\beta}{2}\sum_{i=1}^n (r_{i,k-1} - \alpha V\psi(x_i \cdot w))^2} + e^{-\frac{\beta}{2}\sum_{i=1}^n (r_{i,k-1} + \alpha V\psi(x_i \cdot w))^2}]\eta(dw)}. \tag{4.20}$$

$p_{n,k}(s = 1|x^n, r_{k-1}^n)$ is then a value between 0 and 1 which indicates the posterior probability that the positive sign is chosen. $p_{n,k}(w|s, x^n, r_{k-1}^n)$ is then only a distribution over w once a sign is fixed in the conditioning. Note the index k is not relevant to the structure of this density, the sampling problem is the same for all k just with different input residuals. Thus we shall drop the k indexing and simply consider r^n as some vector of input residuals.

In order to produce samples from this posterior density, we must be able to accomplish two tasks:

1. Sample from both $p_n(w|s = 1, x^n, r^n)$ and $p_n(w|s = -1, x^n, r^n)$ as needed.

2. Compute the posterior probability of a positive sign $p_n(s = 1|x^n, r^n)$.

Assume we are able to accomplish Task 1 using our method of log-concave coupling (we will discuss this shortly in the next section). Thus, assume we have access to L empirical samples $(w_\ell^+)_{\ell=1}^L$ from $p_n(w|s = 1, x^n, r^n)$ and $(w_\ell^-)_{\ell=1}^L$ from $p_n(w|s = -1, x^n, r^n)$. Then we can compute an empirical estimate of $\hat{p}_n(s = 1|x^n, r^n)$. To produce a pool of samples from the joint distribution on signs and weights, for every index ℓ we set (w_ℓ, s_ℓ) equal to the ℓ positive sampled point $(w_\ell^+, 1)$ with probability $\hat{p}(s = 1|x^n, r^n)$ or instead

equal to the ℓ negative sampled point $(w_\ell^-, -1)$ with probability $1 - \hat{p}_n(s = 1|x^n, r^n)$. Thus, we consider methods to approximate $\hat{p}_n(s = 1|x^n, r^n)$ given a pool of samples from the positive and negative sign conditional densities on w.

Note if we are using an odd symmetric neuron activation function such as a tanh, we can instead consider our prior as $P_0(s = 1) = 1$ and this discussion on posterior sign probabilities is not necessary, we can sample all our neurons from the positive sign conditional.

4.3.1 Methods to Compute Posterior Sign Probability Given Samples

Given empirical samples from the positive and negative sign conditional, we consider four possible ways to estimate $\hat{p}(s = 1|x^n, r^n)$ of increasing complexity.

a) Basic 0-1 Probability Estimate

The first option is to choose $\hat{p}_n(s = 1|x^n, r^n)$ as either 0 or 1 depending on if the neuron values from the positive sampling have better loss than the negative sampling. It is most likely that one sign choice is far superior to the other, so it is likely that the true posterior sign probability is very near either 1 or 0. Define

$$\ell^+ = \frac{1}{Ln} \sum_{\ell=1}^{L} \sum_{i=1}^{n} (r_i - \alpha V \psi(x_i \cdot w_\ell^+))^2 \tag{4.21}$$

$$\ell^- = \frac{1}{Ln} \sum_{\ell=1}^{L} \sum_{i=1}^{n} (r_i + \alpha V \psi(x_i \cdot w_\ell^-))^2 \tag{4.22}$$

and set

$$\hat{p}_n(s = 1|x^n, r^n) = 1\{\ell^+ < \ell^-\}. \tag{4.23}$$

b) Importance Sampler

The $0-1$ estimator is a crude estimator. Instead, consider decomposing the posterior sign probability in two ways as expectations over $w|s=1$ and $w|s=-1$ respectively,

$$p_n(s=1|x^n,r^n) = \frac{1}{1+E_{P_n}\big[\frac{e^{-\frac{\beta}{2}\sum_{i=1}^n (r_i+\alpha V\psi(x_i\cdot w))^2}}{e^{-\frac{\beta}{2}\sum_{i=1}^n (r_i-\alpha V\psi(x_i\cdot w))^2}}\big|s=1,x^n,r^n\big]} \tag{4.24}$$

$$= \frac{1}{1+E_{P_n}\big[e^{-2\alpha\beta V\sum_{i=1}^n r_i\psi(x_i\cdot w)}\big|s=1,x^n,r^n\big]}. \tag{4.25}$$

$$p_n(s=-1|x^n,r^n) = \frac{1}{1+E_{P_n}\big[\frac{e^{-\frac{\beta}{2}\sum_{i=1}^n (r_i-\alpha V\psi(x_i\cdot w))^2}}{e^{-\frac{\beta}{2}\sum_{i=1}^n (r_i+\alpha V\psi(x_i\cdot w))^2}}\big|s=-1,x^n,r^n\big]} \tag{4.26}$$

$$= \frac{1}{1+E_{P_n}\big[e^{2\alpha\beta V\sum_{i=1}^n r_i\psi(x_i\cdot w)}\big|s=-1,x^n,r^n\big]}. \tag{4.27}$$

Given that we have empirical samples from each of these two densities, we can empirically estimate these two fractions. Then we can estimate $\hat{p}_n(s=1|x^n,r^n)$ as the average of these two fractions, which makes use of all the samples we have available

$$\hat{p}_n(s=1|x^n,r^n) = \frac{1}{2}\Big(\frac{1}{1+\sum_{\ell=1}^L \frac{1}{L}e^{-2\alpha\beta V\sum_{i=1}^n r_i\psi(x_i\cdot w_\ell^+)}} + \frac{1}{1+\sum_{\ell=1}^L \frac{1}{L}e^{2\alpha\beta V\sum_{i=1}^n r_i\psi(x_i\cdot w_\ell^-)}}\Big). \tag{4.28}$$

Note that these empirical averages could be highly variable, so this estimator may not have very good performance in estimating the probability.

c) Discriminant Estimator

The fraction $\frac{p_n(s=1|x^n,r^n)}{p_n(s=-1|x^n,r^n)}$ is the ratio of the partition function (or normalizing constant) of $p_n(w|s=1,x^n,r^n)$ compared to $p_n(w|s=-1,x^n,r^n)$. With empirical samples from each density, we can mix the samples together and then remove the labels. Then if we try to

reclassify the samples as which set they came from, the ratio of these partition functions is a parameter of the classifier. By solving for the optimal classifier, we can get an estimate of the ratio of the partition functions. This method is based on the technique of discriminant sampling to estimate partition functions [46]. Define

$$Z_+ = \int p_0(w) e^{-\frac{\beta}{2}\sum_{i=1}^n (r_i - \alpha V \psi(x_i \cdot w))^2} \eta(dw) \tag{4.29}$$

$$Z_- = \int p_0(w) e^{-\frac{\beta}{2}\sum_{i=1}^n (r_i + s\alpha V \psi(x_i \cdot w))^2} \eta(dw). \tag{4.30}$$

Then take our $2L$ samples from the positive and negative signed density and put them together into one set of sampled points. Remove the class labels, either positive or negative, from each sample. Then the conditional probability of the sign label of any single element in the set is equal to

$$p_n(s=1|w) = \frac{\frac{1}{Z+} e^{-\frac{\beta}{2}\sum_{i=1}^n (r_i - \alpha V \psi(x_i \cdot w))^2}}{\frac{1}{Z+} e^{-\frac{\beta}{2}\sum_{i=1}^n (r_i - \alpha V \psi(x_i \cdot w))^2} + \frac{1}{Z_-} e^{-\frac{\beta}{2}\sum_{i=1}^n (r_i + \alpha V \psi(x_i \cdot w))^2}} \tag{4.31}$$

$$= \frac{1}{1 + \frac{Z_+}{Z_-} e^{-2\beta\alpha V \sum_{i=1}^n r_i \psi(x_i \cdot w)}} \tag{4.32}$$

$$p_n(s=-1|w) = \frac{\frac{Z_+}{Z_-} e^{-2\beta\alpha V \sum_{i=1}^n r_i \psi(x_i \cdot w)}}{1 + \frac{Z_+}{Z_-} e^{-2\beta\alpha V \sum_{i=1}^n r_i \psi(x_i \cdot w)}}. \tag{4.33}$$

Define $c = \frac{Z_+}{Z_-}$ as a value which is not know to us. Then we can ask, given the true labels s_ℓ, what choice of c has the highest log probability for correctly classifying the points in our set,

$$\text{argmax}_{c \geq 0} \sum_{\ell=1}^{L} \log\left(\frac{1}{1 + c e^{-2\beta\alpha V \sum_{i=1}^n r_i \psi(x_i \cdot w_\ell^+)}}\right) + \sum_{\ell=1}^{L} \log\left(\frac{c e^{-2\beta\alpha V \sum_{i=1}^n r_i \psi(x_i \cdot w_\ell^-)}}{1 + c e^{-2\beta\alpha V \sum_{i=1}^n r_i \psi(x_i \cdot w_\ell^-)}}\right). \tag{4.34}$$

Taking the derivative in c and setting it equal to 0, the condition for the maximizing c is

$$1 = \frac{1}{L}\sum_{\ell=1}^{L} \frac{ce^{-2\beta\alpha V \sum_{i=1}^{n} r_i \psi(x_i \cdot w_\ell^+)}}{1 + ce^{-2\beta\alpha V \sum_{i=1}^{n} r_i \psi(x_i \cdot w_\ell^+)}} + \frac{1}{L}\sum_{\ell=1}^{L} \frac{ce^{-2\beta\alpha V \sum_{i=1}^{n} r_i \psi(x_i \cdot w_\ell^-)}}{1 + ce^{-2\beta\alpha V \sum_{i=1}^{n} r_i \psi(x_i \cdot w_\ell^-)}}. \tag{4.35}$$

This right hand side is an increasing function in c with value 0 at $c = 0$ and value 2 as $c \to \infty$. Define the right hand side of the expression as

$$R(c) = \frac{1}{L}\sum_{\ell=1}^{L} \frac{ce^{-2\beta\alpha V \sum_{i=1}^{n} r_i \psi(x_i \cdot w_\ell^+)}}{1 + ce^{-2\beta\alpha V \sum_{i=1}^{n} r_i \psi(x_i \cdot w_\ell^+)}} + \frac{1}{L}\sum_{\ell=1}^{L} \frac{ce^{-2\beta\alpha V \sum_{i=1}^{n} r_i \psi(x_i \cdot w_\ell^-)}}{1 + ce^{-2\beta\alpha V \sum_{i=1}^{n} r_i \psi(x_i \cdot w_\ell^-)}}. \tag{4.36}$$

We can then estimate c using a binary search algorithm, terminating at a c' with $|1 - R(c')| \leq \epsilon$. Initialize $c_0 = 0$ and $c_1 = 1$. If $R(c_1) > 1$, perform a binary search between the endpoints c_0, c_1 using the midpoint each time until we get a value c' where $|1 - R(c')| < \epsilon$ for some pre-chosen ϵ. If $R(c_1) < 1$, set $c_0 = c_1, c_1 = 2 * c_1$. Check now if $R(c_1) < 1$. If so, repeat this doubling process until eventually we have a $R(c_1) > 1$. Then at that point, perform a binary search as before. Then our estimate of $\hat{p}_n(s = 1|x^n, r^n)$ is

$$\hat{p}_n(s = 1|x^n, r^n) = \frac{c'}{1 + c'}. \tag{4.37}$$

d) Using Recursive Bayes Factors

Our object of interest is to compute the Z_+ and Z_- given in equation (4.29) (4.30). Consider a slight rescaling and allowing us to index by n, we define the Bayes factors,

$$Z_n^+ = \int p_0(w)\left(\frac{\beta}{2\pi}\right)^{\frac{n}{2}} e^{-\frac{\beta}{2}\sum_{i=1}^{n}(r_i - \alpha V \psi(x_i \cdot w))^2} \eta(dw) \tag{4.38}$$

$$Z_n^- = \int p_0(w)\left(\frac{\beta}{2\pi}\right)^{\frac{n}{2}} e^{-\frac{\beta}{2}\sum_{i=1}^{n}(r_i + \alpha V \psi(x_i \cdot w))^2} \eta(dw). \tag{4.39}$$

The log ratio of posterior sign probability is then equal to the log difference of the Bayes factors

$$\log \frac{p_n(s=1|x^n, r^n)}{p_n(s=-1|x^n, r^n)} = \log Z_n^+ - \log Z_n^-. \tag{4.40}$$

Then, consider the ratio of two Bayes factors of successive indexes

$$\frac{Z_n^+}{Z_{n-1}^+} = (\frac{\beta}{2\pi})^{\frac{1}{2}} \int e^{-\frac{\beta}{2}(r_n - \alpha V \psi(x_n \cdot w))^2} p_{n-1}(w|s=1, x^{n-1}, r^{n-1}) \eta(dw). \tag{4.41}$$

This can be computed as an expectation over $p_{n-1}(w|x^{n-1}, r^{n-1})$. The object in the integrand has bounded range between $[e^{-\frac{\beta}{2}(|r_n| + \alpha a_0 V)^2}, 1]$, so the variance of an empirical estimate of this integral would be small. Thus we may write

$$\log Z_n^+ - \log Z_n^- \tag{4.42}$$

$$= \sum_{i=0}^{n-1} \log \frac{Z_{i+1}^+}{Z_i^+} - \log \frac{Z_{i+1}^-}{Z_i^-} \tag{4.43}$$

$$= \sum_{i=0}^{n-1} \log E[e^{-\frac{\beta}{2}(r_{i+1} - \alpha V \psi(x_{i+1} \cdot w))^2} | s=1, x^i, r^i] \tag{4.44}$$

$$- \sum_{i=0}^{n-1} \log E[e^{-\frac{\beta}{2}(r_{i+1} + \alpha V \psi(x_{i+1} \cdot w))^2} | s=-1, x^i, r^i]. \tag{4.45}$$

This would require $2n$ sampling problems, each with their own polynomial dependence on d for the MCMC sampling, to estimate the posterior sign probability. This then is the most computationally intensive method to approximate the posterior sign probability of the methods we have discussed, but has the potential to be the most accurate.

4.4 Log-Concave Coupling

4.4.1 Reverse Conditional Density

Therefore, under the assumption we are able to sample $p_n(w|s=1,x^n,r^n)$ and $p_n(w|s=-1,x^n,r^n)$ we are able to estimate $\hat{p}_n(s=1|x^n,r^n)$. We now focus on constructing a log-concave coupling for sampling w with a fixed sign choice.

Assume a fixed sign value s and consider the density

$$p_n(w) = p_n(w|s,x^n,r^n) \propto e^{-\frac{\beta}{2}\sum_{i=1}^n (r_i - \alpha s V \psi(x_i \cdot w))^2}, \tag{4.46}$$

Going forward, we will refer to this density only as $p_n(w)$ and the conditioning on a choice of x^n, r^n, s is implied.

Then this is exactly the form of densities we studied in the joint sampling problem if we consider $K=1$ and $\tilde{V} = \alpha V$. Thus, the log-concave coupling results follow from the previous proofs in the case $K=1$ and taking note of the α appearing alongside the V. We restate the relevant quantities here in terms of α and V, but note all proofs follow from the joint sampling case.

Here we side-by-side compare the joint sampling problem with the individual sampling problem to note the similarities and the differences. In the joint sampling problem, we define

$$\text{res}_i(w) = y_i - \sum_{k=1}^K s_k \frac{V}{K} \psi(x_i \cdot w_k) \tag{4.47}$$

$$p_n(w) \propto e^{-\frac{\beta}{2}\sum_{i=1}^n (\text{res}_i(w))^2}. \tag{4.48}$$

An important quantity we define is

$$C_n = \max_n |y_n| + a_0 V, \tag{4.49}$$

such that

$$\max_{i \leq n, w \in (S_1^d)^K} |\text{res}_i(w)| \leq C_n. \tag{4.50}$$

We then use this to define our ρ for the forward coupling. If we expand $\beta(\text{res}_i(w))^2$ we see the leading linear term on ψ has a β scaling,

$$\beta(\text{res}_i(w))^2 = \beta y_i^2 - 2\beta \sum_{k=1}^{K} \frac{V}{K} s_k y_i \psi(x_i \cdot w_k) + \beta(\sum_{k=1}^{K} \frac{V}{K} s_k \psi(x_i \cdot w_k))^2. \tag{4.51}$$

We ultimately determine that $Kd = O((\beta N)^2)$ is needed to achieve a log-concave coupling.

For greedy sampling, define the value,

$$\tilde{\text{res}}_i(w) = r_i - \alpha V s \psi(x_i \cdot w) \tag{4.52}$$

$$p_n(w) \propto e^{-\frac{\beta}{2} \sum_{i=1}^{n} (\tilde{\text{res}}_i(w))^2}. \tag{4.53}$$

Now, we have already referred to r_n as the "residuals" that we are defining our density with. So we will call $\tilde{\text{res}}_i(w)$ the "sampling residuals" in this context, as they still represent a useful quantity to directly compare results in the joint sampling case to the greedy sampling case. We now define

$$\tilde{C}_n = \max_{i \leq n} |r_i| + \alpha a_0 V, \tag{4.54}$$

such that

$$\max_{i \leq n, w \in (S_1^d)^K} |\tilde{\text{res}}_i(w)| \leq \tilde{C}_n. \tag{4.55}$$

If we expand $\beta(\tilde{\text{res}}_i(w))^2$, we see the leading linear term has an $\alpha\beta$ scaling. If α is less than 1 (indeed, think of it as $\frac{\log(N)}{N^{1/3}}$), then the α^2 is much smaller than the α term, and the linear term is the dominant term in defining the likelihood.

$$\beta(\tilde{\text{res}}_i(w))^2 = \beta r_i^2 - 2\alpha\beta s V r_i \psi(x_i \cdot w) + \beta\alpha^2 V^2 (\psi(x_i \cdot w))^2. \tag{4.56}$$

We will ultimately conclude that $d = O((\alpha\beta N)^2)$ is needed to achieve a log-concave coupling for the greedy sampling problem. In our later risk analysis for the greedy case, there does not seem to be any improvement using a β which is not constant order. Thus, while we state our results for general β here, one may consider $\beta = 1$ and α as 1 over some fractional power of N with log factors.

For our greedy sampling problem, denote the Hessian as $H_n(w) \equiv \nabla^2 \log p_n(w)$. The density $p_n(w)$ is log-concave if $H_n(w)$ is negative definite for all choices of w. For any vector $u \in \mathbb{R}^d$, the quadratic form $u^{\text{T}} H_n(w) u$ can be expressed as

$$-\beta \sum_{i=1}^n \left(s\alpha V \psi'(w \cdot x_i) u \cdot x_i \right)^2 + \beta\alpha V s \sum_{i=1}^n \tilde{\text{res}}_i(w) \psi''(w \cdot x_i)(u \cdot x_i)^2. \tag{4.57}$$

The first term is negative, but the second term may be positive or negative since the signs are not known. We then go about defining n auxiliary random variables ξ_i as follows. Define the value

$$\rho = \sqrt{\frac{3}{2}} a_2 \alpha\beta V \tilde{C}_n. \tag{4.58}$$

For a positive $\delta \leq 1/16$, we also define a constrained set,

$$B = \Big\{\xi \in \mathbb{R}^n : \max_j |\sum_{i=1}^n x_{i,j}\xi_i| \leq n + \sqrt{2\log(\frac{2d}{\delta})}\sqrt{\frac{n}{\rho}}\Big\}. \tag{4.59}$$

Define the function,

$$Z(w) = \log \int_B \prod_{i=1}^n \big(\frac{\rho}{2\pi}\big)^{\frac{1}{2}} e^{-\frac{\rho}{2}(\xi_i - x_i \cdot w)^2} d\xi. \tag{4.60}$$

We then define the unrestricted conditional density for $\xi_1, \ldots, \xi_n$ with respect to Lebesgue measure as

$$p_n(\xi|w) = \big(\frac{\rho}{2\pi}\big)^{\frac{n}{2}} e^{-\frac{\rho}{2}\sum_{i=1}^n (\xi_i - x_i \cdot w)^2}. \tag{4.61}$$

The restricted conditional density forcing $\xi \in B$ is defined as

$$p_n^*(\xi|w) = 1_B(\xi)\big(\frac{\rho}{2\pi}\big)^{\frac{n}{2}} e^{-\frac{\rho}{2}\sum_{i=1}^n (\xi_i - x_i \cdot w)^2} e^{-Z(w)}. \tag{4.62}$$

Using the restricted conditional density $p_n^*(\xi|w)$ paired with the target density $p_n(w)$, this defines a joint density $p_n^*(w, \xi)$ as well as induced marginal $p_n^*(\xi)$ and reverse conditional $p_n^*(w|\xi)$.

The following results are restatements of the joint sampling results but in a greedy setting. The proofs of these results follow from the joint sampling case by setting $K = 1$, using our new definition or ρ, and using $\tilde{C}_n$ in place of C_n.

Lemma 4.1. *For any weight vector w with $\|w\|_1 \leq 1$ the set B has probability under $p(\xi|w)$ at least*

$$P(\xi \in B|w) \geq 1 - \frac{\delta}{\sqrt{2\log(2d/\delta)}}. \tag{4.63}$$

Lemma 4.2. *For any specified vector $u \in R^d$, define the value*

$$\tilde{\sigma}^2 = \frac{\sum_{i=1}^{n}(u \cdot x_i)^2}{\rho}. \tag{4.64}$$

For positive values $\delta \leq 1/16$ with $d \geq 4$, we then have upper bounds,

$$|u \cdot \nabla Z(w)| \leq \frac{\rho\tilde{\sigma}}{1-\delta}\frac{\delta}{\sqrt{2\pi}} \tag{4.65}$$

and

$$|u^T(\nabla^2 Z(w))u| \leq \frac{\rho^2\tilde{\sigma}^2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\Big(2\sqrt{2\log(1/\delta)} + \frac{\rho^2\tilde{\sigma}^2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\Big). \tag{4.66}$$

Note both bounds go to 0 as $\delta \to 0$, and thus can be made arbitrarily small for a certain choice of δ.

Theorem 4.1. *Define the notation*

$$H_1(\delta) = \frac{2}{\sqrt{2\pi}}\frac{\delta}{1-\delta}\sqrt{2\log\frac{2}{\delta}} \tag{4.67}$$

$$H_2(\delta) = \Big(a_2\alpha\beta V\tilde{C}_n\Big)^2\frac{1}{2\pi}\frac{\delta^2}{(1-\delta)^2}. \tag{4.68}$$

Assume a sufficiently small $\delta \leq \frac{1}{16}$ that satisfies

$$H_1(\delta) \leq \frac{1}{100} \tag{4.69}$$

$$H_2(\delta) \leq \frac{1}{10}. \tag{4.70}$$

For the continuous uniform prior, with ξ restricted to the set B defined by δ, and ρ as in equation (4.58)*, the reverse conditional density $p_n^*(w|\xi)$ is a log-concave density in w, for any ξ in B.*

Corollary 4.1. *A positive δ which satisfies,*

$$\delta \leq \min\Big(\frac{1}{300}, \sqrt{\frac{2\pi}{11}}\frac{1}{a_2\alpha\beta V\tilde{C_n}}\Big), \tag{4.71}$$

will satisfy conditions (4.69)*,* (4.70)*.*

Note α should be considered quite small, like $\frac{\log(N)}{N^{1/3}}$, so the $1/300$ condition should be considered the dominant term to satisfy.

4.4.2 Marginal Density

As before, the score and Hessian of the marginal density $p_n^*(\xi)$ are determined by the mean and variance of the reverse conditional density $p_n^*(w|\xi)$. Thus the object to study to show the marginal $p_n^*(\xi)$ is log-concave is the maximum variance of $u^T\mathbf{X}w$ for any unit vector u (this is the previous joint sampling result now with $K = 1$ neuron).

Lemma 4.3. *The density $p_n^*(\xi)$ is log-concave if for any unit vector $u \in \mathbb{R}^n$ the variance of a particular linear combination of w, namely $u^T\boldsymbol{X}w$, with respect to the reverse conditional $p_n^*(w|\xi)$ is less than $1/\rho$,*

$$\mathit{Var}_{P_n^*}[u^T X w|\xi] \leq 1/\rho, \tag{4.72}$$

for ξ in the convex support set B.

We then study this variance using a Hölder inequality as before. However, with a β and an α in the definition of $\log(p_n(w))$ we must make a slightly modified definition of the CGF function Γ that we work with. Define the function

$$h_\xi^n(w) = -\frac{\beta}{2}\sum_{i=1}^{n}(r_i - \alpha s V\psi(x_i \cdot w))^2 - \sum_{i=1}^{n}\frac{\rho}{2}(\xi_i - x_i \cdot w)^2 - Z(w). \tag{4.73}$$

Define the function shifted by its mean under the prior

$$\tilde{h}_\xi^n(w) = h_\xi^n(w) - E_{P_0}[h_\xi^n(w)]. \tag{4.74}$$

Define its cumulant generating function with respect to the prior as

$$\Gamma_\xi^n(\tau) = \log E_{P_0}[e^{\tau \tilde{h}_\xi^n(w)}]. \tag{4.75}$$

Then the following results carry over from the joint sampling problem.

Lemma 4.4. *For any integer $\ell \geq 1$ and for any vector $u \in \mathbb{R}^{Kd}$ we have the upper bound*

$$\mathit{Var}_{P_n^*}(u \cdot w|\xi) \leq \Big(E_{P_0}[(u \cdot w)^{2\ell}]\Big)^{\frac{1}{\ell}} e^{\frac{\ell-1}{\ell}\Gamma_\xi^n(\frac{\ell}{\ell-1}) - \Gamma_\xi^n(1)}. \tag{4.76}$$

Lemma 4.5. *For any unit vector $u \in \mathbb{R}^n$,*

$$E_{P_0}[(u^T \boldsymbol{X} w)^{2\ell}]^{\frac{1}{\ell}} \leq \frac{4\ell n}{\sqrt{e}\, d}. \tag{4.77}$$

Lemma 4.6. *Denote the constants*

$$A_1 = 2a_1 + 4\sqrt{\frac{3}{2}}a_2 \tag{4.78}$$

$$A_2 = \Big(2 + \frac{1}{\sqrt{\pi}}\Big)\sqrt{2a_2\sqrt{\frac{3}{2}}}. \tag{4.79}$$

Assume positive $\delta \leq \frac{1}{16}, d \geq 4$. For any positive integer $\ell \geq 1$ and any ξ from the constrained set B, we have

$$\frac{\ell-1}{\ell}\Gamma_\xi^n(\frac{\ell}{\ell-1}) - \Gamma_\xi^n(1) \leq A_1\frac{\alpha\beta V\tilde{C}_n n}{\ell} + A_2\frac{\sqrt{\alpha\beta V\tilde{C}_n n}}{\ell}\Big(\sqrt{\log(\frac{2d}{\delta})}\Big). \tag{4.80}$$

Theorem 4.2. *Assume* $\delta \leq \frac{1}{16}, d \geq 4$. *Further assume that*

$$\log\left(\frac{2d}{\delta}\right) \leq \alpha\beta N, \tag{4.81}$$

which is a mild condition that d *not be exponentially large in* N.

Define A_1, A_2 *as in* (4.78)*,* (4.79) *and define the constant*

$$A_3 = 4\sqrt{\frac{3}{2e}}a_2(\tilde{C}_N V)^{\frac{3}{2}}[A_1 + A_2(\tilde{C}_N V)^{\frac{1}{2}}]. \tag{4.82}$$

Let d *satisfy*

$$d \geq A_3(\alpha\beta N)^2. \tag{4.83}$$

Then for all $n \leq N$*, the marginal density for* $p_n^*(\xi)$ *is log-concave under the continuous uniform prior. If equation* (4.83) *is a strict inequality, the density is strictly log-concave.*

Chapter 5

Statistical Risk for Greedy Bayes

5.1 Introduction

Our method of risk control for the greedy Bayes estimator is different than our method to control the risk of the joint sampling algorithm, but does share some similarities. We start by first analyzing arbitrary sequence regret, and then treat risk as a form of expected regret. We first analyze log regret, then relate this to square regret, square risk can be analyzed as an expected square regret.

When sampling all neurons at once, there is only one log regret object to consider, and that is the cumulative log regret using the sequence of predictive densities. We then bound the log regret of this object from any particular target function, noting there is at least one set of parameters that is a good fit for the target in the prior support, and using the index of resolvability argument.

However, in the greedy case we have a different regret at every level k of our iterative sampling procedure. We will show a recursive relationship between regret at level $k+1$ and regret at level k

$$A_{k+1} \leq (1-\alpha)A_k + \tau C \tag{5.1}$$

for some $0 \leq \alpha, \tau \leq 1$ and some constant C. With α and τ small, this recursive relationship can be shown to imply decay in the successive regret bounds at each level k of the iteration. For example, it is a simple task to show the following result [49, Lemma 4.5.4]

Lemma 5.1. *Let $A_0, A_1, \ldots, A_K$ be a sequence of values following an iterative formula*

$$A_k \leq (1 - \alpha)A_{k-1} + \tau C, \tag{5.2}$$

for $1 \leq k \leq K$, $\tau C \geq 0$. Then the final term A_K in the recursion can be bounded explicitly as

$$A_K \leq (1 - \alpha)^K A_0 + \tau C \frac{1 - (1 - \alpha)^K}{\alpha}. \tag{5.3}$$

In particular, the choice $\alpha = \frac{\log(K)}{K}$ and $\tau = \alpha^2$ results in the bound

$$A_K \leq \frac{1}{K} \max(A_0, 0) + \frac{\log K}{K} C. \tag{5.4}$$

With careful choice of the α and τ (which are functions of the α, β, M, N, K of our problem), we will show such a recursion arises in the regret terms of our greedy Bayes method.

First, we review some greedy optimization results to show how such a recursion is typically established in an optimization problem.

5.2 An Overview of Greedy Optimization Procedures for Neural Networks

There is a long history of greedy methods to continually improve a linear combination of functions by adding in one simple function at a time [7, 34, 40, 31, 9]. Consider if we have

a (possibly uncountable) library of base functions $\mathcal{H}$ and we want to construct a function which is a linear combination of elements of that library. If one has an existing function f which is a linear combination of elements of $\mathcal{H}$, we can define a new function by taking a linear combination with an element of the library, $f'(x) = \alpha_1 f(x) + \alpha_2 h(x)$ for some $h \in \mathcal{H}$ and $\alpha_1, \alpha_2 \in \mathbb{R}$. This new function f' can be taken as our new base function, and we can repeat the process to add in another element of the library.

Say this algorithm is repeated K times so we have recursively defined functions

$$f_k = \alpha_{k,1} f_{k-1} + \alpha_{k,2} h_k. \tag{5.5}$$

This algorithm requires a sequence of update weights $(\alpha_{k,1}, \alpha_{k,2})_{k=1}^K$ as well as a selection rule to select h_k at each iteration. The method of recursively updating a function via a selection rule can also be applied to other greedy algorithms such as the Frank Wolfe algorithm [23].

There are a variety of ways to define the update weights $\alpha_{k,1}, \alpha_{k,2}$ and selection rule for h_k. Say we have a sequence of values $(x_i, y_i)_{i=1}^n$ and a loss function L and our goal is to minimize the empirical loss using f_K at the termination of our greedy algorithm

$$\min \sum_{i=1}^n L(y_i, f_K(x_i)). \tag{5.6}$$

Then the selection rule is usually to pick h_k to minimize the loss at iteration k, or some penalized loss to prefer specific elements of $\mathcal{H}$. The updates weights $(\alpha_{k,1}, \alpha_{k,2})_{k=1}^K$ can themselves also be part of the optimization at each iteration k of the algorithm. That is, our selection rule could be

$$\alpha_{k,1}, \alpha_{k,2}, h_k = \min_{a,b \in \mathbb{R}, h \in \mathcal{H}} \sum_{i=1}^n L(y_i, a f_{k-1}(x_i) + b h(x_i)) + \text{pen}(a, b, h), \tag{5.7}$$

for some potential penalty function. However, this often results in a difficult optimiza-

tion problem. Instead, the update weights themselves do not have to be considered as an optimization problem, and a pre-selected sequence of weights can be chosen. It is often chosen so that $\alpha_{k,1} + \alpha_{k,2} = 1, \alpha_{k,1}, \alpha_{k,2} > 0$ so that our fits f_k are convex combinations of elements of the library whose weights of combination add to 1. We can also then only have one value $\alpha_k \in (0, 1)$ at each iteration k and have

$$f_k = (1 - \alpha_k) f_{k-1} + \alpha_k h_k. \tag{5.8}$$

The α_k can vary at each iteration k, for example $\alpha_k = 1/k$ starting off with large weights of combination and cooling down to lower weights. However, it may also be desirable to have α_k constant and small at each iteration of the algorithm, for example we will work with $\alpha_k = \alpha = \frac{\log(K)}{K}$ at each iteration of our algorithm.

Then, with a fixed choice of α we may define our selection rule for h_k to minimize the loss at iteration k based on the previous fit,

$$h_k = \min_{h \in \mathcal{H}} \sum_{i=1}^{n} L(y_i, (1 - \alpha) f_{k-1}(x_i) + \alpha h(x_i)). \tag{5.9}$$

Specializing to our case of neural networks, given an activation function ψ our function library is all signed neurons with interior weights with ℓ_1 norm less than 1 scaled by V,

$$\mathcal{H} = \{h : h(x) = sV\psi(x \cdot w), s \in \{-1, 1\}, w \in \mathbb{R}^d, \|w\|_1 \leq 1\}. \tag{5.10}$$

Our loss function will be chosen as square loss, and our update rule for the neurons is

$$f_k = (1 - \alpha) f_{k-1} + \alpha V s_k \psi(w_k, \cdot) \tag{5.11}$$

$$s_k, w_k = \min_{s \in \{-1,1\}, \|w\|_1 \leq 1} \sum_{i=1}^{n} (y_i - (1 - \alpha) f_{k-1}(x_i) - \alpha s V \psi(x_i \cdot w))^2. \tag{5.12}$$

The key outcome of this selection rule is that we can establish a recursive relationship

(5.2) between the loss at each iteration k of the algorithm. We can then apply Lemma 5.1 to bound the loss of the terminal linear combination.

Lemma 5.2 (Greedy Optimization Lemma)**.** *Let $(x_i)_{i=1}^N$ be a sequence of input values and $(h_i)_{i=1}^N$ be a pre-existing vector of fit values. For some value $0 < \alpha < 1$, given a neuron weight vector $w \in S_1^d$ and a sign $s \in \{-1, 1\}$, define the new vector*

$$f_{w,s}(x_i) = (1 - \alpha)h_i + \alpha V s\psi(x_i \cdot w) \tag{5.13}$$

Let $(y_i)_{i=1}^N$ be a vector of values, and let $g = (g_i)_{i=1}^N$ be any element of the closure of Hull$_N(V\psi)$. Define $w^* \in S_1^d$ and sign $s^* \in \{-1, 1\}$ as the minimizing values of the regret,

$$w^*, s^* = \textit{argmin}_{w \in S_1^d, s \in \{-1,1\}} \|y - (1 - \alpha)h - \alpha V s\psi_w\|_N^2 \tag{5.14}$$

Then using this w^*, s^*, the regret for f_{w^*,s^*} has the following relationship with the regret using h

$$\|y - (1 - \alpha)h - \alpha V s^*\psi_{w^*}\|_N^2 - \|y - g\|_N^2 \leq (1 - \alpha)(\|y - h\|_N^2 - \|y - g\|_N^2) + \alpha^2 N a_0^2 V^2. \tag{5.15}$$

Proof. Since g lives in the closure of the convex hull of signed neurons scaled by V, for every $\epsilon > 0$ there exists some finite width neural network with continuous-valued weight vectors $w_\ell \in S_1^d$ and outer weights c_ℓ with $\sum_\ell |c_\ell| = 1$ such that

$$\tilde{g}(x) = V \sum_\ell |c_\ell| \text{sign}(c_\ell)\psi(x \cdot w_\ell), \quad \sum_{i=1}^N (g(x_i) - \tilde{g}(x_i))^2 \leq \epsilon. \tag{5.16}$$

Decompose our regret in the following way,

$$\|y-(1-\alpha)h-\alpha Vs^*\psi_{w^*}\|_N^2-\|y-g\|_N^2 \tag{5.17}$$

$$=\|(1-\alpha)y-(1-\alpha)h+\alpha y-\alpha Vs^*\psi_{w^*}\|_N^2-\|y-g\|_N^2 \tag{5.18}$$

$$=(1-\alpha)^2\|y-h\|_N^2+2\alpha(1-\alpha)\langle y-h,y-s^*V\psi_{w^*}\rangle_N+\alpha^2\|y-s^*V\psi_{w^*}\|_N^2-\|y-g\|_N^2. \tag{5.19}$$

Then, w^*, s^* is the minimizer of this expression. In particular, the minimum is less than any average using any distribution for w, s. Thus use the distribution defined by c_ℓ, and this is an upper bound on the expression,

$$(1-\alpha)^2\|y-h\|_N^2 \tag{5.20}$$

$$+\sum_\ell |c_\ell|2\alpha(1-\alpha)\langle y-h,y-\text{sign}(c_\ell)V\psi_{w_\ell}\rangle_N \tag{5.21}$$

$$+\alpha^2\sum_\ell |c_\ell|\|y-\text{sign}(c_\ell)V\psi_{w_\ell}\|_N^2-\|y-g\|_N^2. \tag{5.22}$$

Note by definition, the average of the $V\text{sign}(c_\ell)\psi_{w_{ell}}$ using $|c_\ell|$ as the distribution is exactly the definition of $\tilde{g}$. Thus consider expression (5.21),

$$\sum_\ell |c_\ell|2\alpha(1-\alpha)\langle y-h,y-\text{sign}(c_\ell)V\psi_{w_\ell}\rangle_N=2\alpha(1-\alpha)\langle y-h,y-\tilde{g}_\epsilon\rangle_N \tag{5.23}$$

$$\leq 2(\alpha)(1-\alpha)\big(\frac{1}{2}\|y-h\|_N^2+\frac{1}{2}\|y-\tilde{g}_\epsilon\|^2\big). \tag{5.24}$$

Now consider expression (5.22). Using a bias variance decomposition, this term is like a

variance plus a bias term,

$$\alpha^2 \sum_\ell |c_\ell| \|y - \text{sign}(c_\ell) V \psi_{w_\ell}\|_N^2 = \alpha^2 \sum_\ell |c_\ell| \|\text{sign}(c_\ell) V \psi_{w_\ell} - \tilde{g}_\epsilon\|_N^2 + \alpha^2 \|y - \tilde{g}_\epsilon\|_N^2 \tag{5.25}$$

$$\leq \alpha^2 N a_0^2 V^2 + \alpha^2 \|y - \tilde{g}_\epsilon\|_N^2. \tag{5.26}$$

Thus we have upper bound

$$[(1-\alpha)^2 + \alpha(1-\alpha)] \|y - h\|_N^2 + [\alpha^2 + \alpha(1-\alpha)] \|y - \tilde{g}_\epsilon\|^2 - \|y - g\|_N^2 + \alpha^2 N a_0^2 V^2 \tag{5.27}$$

$$= (1-\alpha) \|y - h\|_N^2 + \alpha \|y - g_\epsilon\|_N^2 - \|y - g\|_N^2 + \alpha^2 N a_0^2 V^2 \tag{5.28}$$

$$= (1-\alpha) \|y - h\|_N^2 + (\alpha - 1) \|y - g\|_N^2 + \alpha (\|g - g_\epsilon\|_N^2 + 2\langle y - g, g - g \rangle_N) + \alpha^2 N V^2 \tag{5.29}$$

$$\leq (1-\alpha) [\|y - h\|_N^2 - \|y - g\|_N^2] + \alpha^2 N a_0^2 V^2 + \alpha\epsilon + 4\sqrt{\epsilon} (\max_{i \leq N} |y_i| + a_0 V). \tag{5.30}$$

Taking $\epsilon \to 0$ yields the final result.

□

5.3 Arbitrary Sequence Regret

Consider some integer M, and let P_0 be the uniform prior on $S_{1,M}^d \times \{-1, 1\}$. That is, each possible weight vector $w \in S_{1,M}^d$ is considered equally likely under the prior, and then the outer sign of the neuron is considered equally likely to be ± 1. Note if we are working with odd symmetric neurons such as a tanh, we can instead consider $P_0(s = 1) = 1$ and consider all outer signs as positive.

Recall the definition of the residuals at level k of our greedy method,

$$r_{n,k} = y_n - (1-\alpha)\hat{f}_{n-1,k}(x_n). \tag{5.31}$$

Recall also the definition of the posterior and fit at level k,

$$\ell_{n,k}(w,s) = \frac{1}{2}\sum_{i=1}^{n}(r_{i,k-1} - \alpha s V \psi(x_i \cdot w))^2 \tag{5.32}$$

$$p_{n,k}(w,s|x^n, r_{k-1}^n) \propto p_0(w,s)e^{-\beta \ell_{n,k}(w,s)} \tag{5.33}$$

$$\hat{f}_{n,k}(x) = (1-\alpha)\hat{f}_{n,k-1}(x) + \alpha V E_{P_{n,k}}[s\psi(x \cdot w)|x^n, r_{k-1}^n]. \tag{5.34}$$

Define the conditional density $p(y|\hat{f}_{n,k}, x, w, s)$ as Normal with mean $(1-\alpha)\hat{f}_{n,k}(x) + \alpha s V \psi(x \cdot w)$ and variance $\frac{1}{\beta}$. Then define the posterior predictive density at level k and index n,

$$p_{n,k}(y|x, x^n, r_{k-1}^n) = E_{P_{n,k}}[p(y|\hat{f}_{n,k-1}, x, w, s)|x^n, r_{k-1}^n]. \tag{5.35}$$

We then define three notions of regret: square, random, and log. Given a set of $(x_i)_{i=1}^N$ input values, a competitor function g, and a sequence of outputs $(y_i)_{i=1}^N$, define the individual terms in the regret for each n and k,

$$R_{n,k}^{\text{square}} = \frac{1}{2}\big[(y_n - \hat{f}_{n-1,k}(x_n))^2 - (y_n - g(x_n))^2\big] \tag{5.36}$$

$$R_{n,k}^{\text{rand}} = \frac{1}{2}\big[E_{P_{n-1,k}}[(y_n - (1-\alpha)\hat{f}_{n-1,k-1}(x_n) - \alpha\psi(x_n \cdot w))^2|x^{n-1}, r_{k-1}^{n-1}] - (y_n - g(x_n))^2\big] \tag{5.37}$$

$$R_{n,k}^{\text{log}} = \frac{1}{\beta}\Big[\log\frac{1}{p_{n-1,k}(y_n|x_n, x^{n-1}, r_{k-1}^{n-1})} - \log\frac{1}{q(y_n|x_n)}\Big] \tag{5.38}$$

where

$$q(y|x) = (\frac{\beta}{2\pi})^{\frac{1}{2}} e^{-\frac{\beta}{2}(y-g(x))^2}. \tag{5.39}$$

We note the following relationship between squared norms with y's and our definition of recursive residuals,

$$y_n - \hat{f}_{n-1,k}(x_n) = y_n - (1-\alpha)\hat{f}_{n-1,k-1}(x_n) - \alpha V E_{P_{n-1,k}}[s\psi(x_n \cdot w)|x^{n-1}, r_{k-1}^{n-1}] \tag{5.40}$$

$$= r_{n,k-1} - \alpha V E_{P_{n-1,k}}[s\psi(x_n \cdot w)|x^{n-1}, r_{k-1}^{n-1}] \tag{5.41}$$

$$= E_{P_{n-1,k}}[r_{n,k-1} - \alpha s V \psi(x_n \cdot w)|x^{n-1}, r_{k-1}^{n-1}]. \tag{5.42}$$

We then have the following lemma relating the ordering of these various notions of regret.

Lemma 5.3. *Assume $\hat{f}_{n,k}$ is bounded in absolute value by $a_0 V$ for all indexes $0 \leq n \leq N, 1 \leq k \leq K$. Define*

$$\lambda_{n,k} = \alpha |r_{n,k-1}| a_0 V + \frac{1}{2}\alpha^2 a_0^2 V^2 \tag{5.43}$$

Then we have

$$R_{n,k}^{log} \leq R_{n,k}^{rand} \tag{5.44}$$

$$R_{n,k}^{square} \leq R_{n,k}^{rand} \leq R_{n,k}^{log} + 2\beta\lambda_{n,k}^2. \tag{5.45}$$

Proof. $R_{n,k}^{\text{square}} \leq R_{n,k}^{\text{rand}}$ and $R_{n,k}^{\text{log}} \leq R_{n,k}^{\text{rand}}$ by Jensen's inequality. Consider

$$\frac{1}{2}[y_n - (1-\alpha)\hat{f}_{n-1,k-1}(x_n) - \alpha s V \psi(x_n \cdot w)]^2 = \frac{1}{2}[r_{n,k-1} - \alpha s V \psi(x_n \cdot w)]^2 \tag{5.46}$$

as a random variable in w, s where w, s follow the distribution of $P_{n-1,k}(\cdot|x^{n-1}, r_{k-1}^{n-1})$.

Then $R_{n,k}^{\text{rand}}$ is the expected value of this random variable, and $R_{n,k}^{\log}$ is $-1/\beta$ times its cumulant generating function at $-\beta$. Isolate the part of the random variable that depends on w, s,

$$-\alpha V r_{n,k-1} s\psi(x_n \cdot w) + \frac{1}{2}\alpha^2 V^2 \psi(x_i \cdot w)^2. \tag{5.47}$$

Note this object is bounded by,

$$\lambda_{n,k} = \alpha |r_{n,k-1}| a_0 V + \frac{1}{2}\alpha^2 a_0^2 V^2. \tag{5.48}$$

The CGF of a bounded random variable matches its mean to within half the square of the bound, by second order Taylor expansion. Thus we have

$$R_{n,k}^{\text{rand}} \leq R_{n,k}^{\log} + \frac{1}{2}\beta\lambda_{n,k}^2. \tag{5.49}$$

□

Define the averaged quantities,

$$\bar{R}_{N,k}^{\text{square}} = \frac{1}{N}\sum_{n=1}^{N} R_{n,k}^{\text{square}} \qquad \bar{R}_{N,k}^{\text{rand}} = \frac{1}{N}\sum_{n=1}^{N} R_{n,k}^{\text{rand}} \tag{5.50}$$

$$\bar{R}_{N,k}^{\log} = \frac{1}{N}\sum_{n=1}^{N} R_{n,k}^{\log} \qquad \bar{\Lambda}_{N,k}^2 = \frac{1}{N}\sum_{n=1}^{N} \lambda_{n,k}^2. \tag{5.51}$$

Then the averaged quantities follow the same ordering as their individual terms,

$$\bar{R}_{N,k}^{\text{square}} \leq \bar{R}_{N,k}^{\text{rand}} \leq \bar{R}_{N,k}^{\log} + 2\beta\bar{\Lambda}_{N,k}^2. \tag{5.52}$$

First, we give a lemma showing that for any continuous valued neuron weight in S_1^d, there exists a discrete valued neuron weight in $S_{1,M}^d$ that achieves regret of order $O(1/M)$

relative to the continuous vector. This is the analog of our joint approximation lemma, which showed any finite width neural network with continuous weights has a neural network of width K and discrete weights that can approximate it well. This result is only for one neuron at a time.

Lemma 5.4 (Greedy Approximation Lemma)**.** *Let $(x_i)_{i=1}^N$ be a sequence of values with $x_i \in [-1,1]^N$. Let w^{cont} be any particular continuous weight vector in S_1^d, and s a sign value in $\{-1,1\}$. Then there exists a choice of discrete weight $w^* \in S_{1,M}^d$ such that for any sequence $(y_i)_{i=1}^N$, the regret compared to $\psi(\cdot, w^{cont})$ is bounded by*

$$\sum_{i=1}^{N}(y_i - \alpha V s\psi(x_i \cdot w^*))^2 - (y_i - \alpha V \psi(x_i \cdot w^{cont}))^2 \leq a_2 \alpha V \frac{\|y\|_1}{M} + \alpha^2 V^2 (a_1^2 + a_0 a_2)\frac{N}{M}. \tag{5.53}$$

Proof. As we have done before, given the continuous vector $w^{\text{cont}} \in S_1^d$ we will define a distribution over discrete random variables $w^{\text{disc}} \in S_{1,M}^d$ using M iid random index selections. Given a continuous vector w^{cont} of dimension d, we then make a random discrete vector as follows. Define a $d+1$ coordinate, $w_{d+1} = 1 - \|w_{1:d}^{\text{cont}}\|_1$, to have a $d+1$ length vector which sums to 1. Consider a random index $J \in \{1, \dots d+1\}$ where $J = j$ with probability $|w_j^{\text{cont}}|$. Given w^{cont}, this defines a distribution on $\{1, \dots, d+1\}$. Draw M iid random indices $J_1, \dots, J_M$ from this distribution and define the counts of each index

$$m_j = \sum_{i=1}^{M} 1\{J_i = j\}. \tag{5.54}$$

We then define the discrete vector $w^{\text{disc}} \in S_{1,M}^d$ with coordinate values

$$w_j^{\text{disc}} = \text{sign}(w_j^{\text{cont}})\frac{m_j}{M}. \tag{5.55}$$

Then consider the expected regret using w^{disc} drawn from this distribution,

$$E[\|y - \alpha V s \psi_{w^{\text{disc}}}\|_N^2] - \|y - \alpha V s \psi_{w^{\text{cont}}}\|_N^2 \tag{5.56}$$

$$= \sum_{i=1}^{N} E[y_i^2 - 2\alpha V s y_i \psi(x_i \cdot w^{\text{disc}}) + \alpha^2 V^2 (\psi(x_i \cdot w^{\text{disc}}))^2] - \|y - \alpha V s \psi_{w^{\text{cont}}}\|_N^2 \tag{5.57}$$

Perform a second order Taylor expansion of function in the expectation centered at w^{cont}. Noting that $|\psi(z)| \leq a_0, |\psi(z)'| \leq a_1, |\psi''(z)| \leq a_2$ for all $z \in [-1, 1]$, we have the following upper bound,

$$y_i^2 - 2\alpha V s y_i \psi(x_i \cdot w^{\text{disc}}) + \alpha^2 V^2 (\psi(x_i \cdot w^{\text{disc}}))^2 \tag{5.58}$$

$$\leq (y_i - \alpha V \psi(x_i \cdot w^{\text{cont}}))^2 \tag{5.59}$$

$$- 2\alpha s V y_i \psi'(x_i \cdot w^{\text{cont}})(x_i \cdot w^{\text{disc}} - x_i \cdot w^{\text{cont}}) + a_2 \alpha V |y_i| (x_i \cdot w^{\text{cont}})(x_i \cdot w^{\text{disc}} - x_i \cdot w^{\text{cont}})^2 \tag{5.60}$$

$$+ 2\alpha^2 V^2 \psi(x_i \cdot w^{\text{cont}}) \psi'(x_i \cdot w^{\text{cont}})(x_i \cdot w^{\text{disc}} - x_i \cdot w^{\text{cont}}) \tag{5.61}$$

$$+ \alpha^2 V^2 (a_1^2 + a_0 a_2)(x_i \cdot w^{\text{disc}} - x_i \cdot w^{\text{cont}})^2. \tag{5.62}$$

Then note that $E[w^{\text{disc}}|w^{\text{cont}}] = w^{\text{cont}}$ so all first order terms are mean 0, and all second order terms will be variances which will be of the order $1/M$ since they are the variance of a sum of M iid random variables divided by M. This gives us the upper bound,

$$E[\|y - \alpha V s \psi_{w^{\text{disc}}}\|_N^2] - \|y - \alpha V s \psi_{w^{\text{cont}}}\|_N^2 \tag{5.63}$$

$$\leq a_2 \alpha V \sum_{i=1}^{N} |y_i| \text{Var}(x_i \cdot w^{\text{disc}}|w^{\text{cont}}) + \alpha^2 V^2 (a_1^2 + a_0 a_2) \sum_{i=1}^{N} \text{Var}(x_i \cdot w^{\text{disc}}|w^{\text{cont}}) \tag{5.64}$$

$$\leq a_2 \alpha V \|y\|_1 \frac{1}{M} + \alpha^2 V^2 (a_1^2 + a_0 a_2) \frac{N}{M}. \tag{5.65}$$

□

Thus, we have established that: greedy optimization creates a certain recursion for regret (Lemma 5.2), square regret can be closely related to log regret (Lemma 5.3), and any continuous neuron has a discrete neuron that has small regret relative to that neuron (Lemma 5.4). Combining these results, we show that square regret using the greedy Bayes estimators instead of optimization establishes a similar recursive relationship.

Theorem 5.1. *Let $(x_i)_{i=1}^N$ be a sequence of input values with all $x_i \in [-1,1]^d$. Let g be a competitor function which is an element of Hull$_N(V\Psi)$, the closure of the convex hull of signed neurons scaled by V. Let P_0 be the uniform prior on $(S_{1,M}^d) \times \{-1,1\}$. For any sequence of values $(y_i)_{i=1}^N$, define the value*

$$C_N = \max_{n=1}^{N} |y_n| + a_0 V. \tag{5.66}$$

Define the term,

$$\tau = \frac{1}{2}\alpha^2 a_0^2 V^2 + \frac{\beta}{2}\alpha^2 a_0^2 V^2 (C_N + \frac{1}{2}\alpha a_0 V)^2 \tag{5.67}$$

$$+ \frac{M\log(2d+1)}{\beta N} + \frac{1}{2}\frac{1}{M} a_2 \alpha V C_N + \frac{1}{2}\alpha^2 V^2 (a_1^2 + a_0 a_2)\frac{1}{M}. \tag{5.68}$$

Then consider the average square regret of our sequence of estimators as defined in equations (5.36)*,* (5.50)*. The square regrets at level k satisfy the recursive relationship,*

$$\bar{R}_{N,k}^{\textit{square}} \leq (1-\alpha)\bar{R}_{N,k-1}^{\textit{square}} + \tau. \tag{5.69}$$

Proof. Consider first the average square regret at level k,

$$\bar{R}_{N,k}^{\text{square}} = \frac{1}{N}\sum_{n=1}^{N}[\frac{1}{2}(y_n - (1-\alpha)\hat{f}_{n-1,k-1}(x_n) - \alpha V E_{P_{n-1,k}}[s\psi(x_n \cdot w)])^2 - \frac{1}{2}(y_n - g(x_n))^2]. \tag{5.70}$$

Recall the definition of the $r_{n,k-1}$ residuals,

$$r_{n,k-1} = y_n - \hat{f}_{n-1,k-1}(x_n). \tag{5.71}$$

For notational convenience we can write

$$\bar{R}_{N,k}^{\text{square}} = \frac{1}{N}\sum_{n=1}^{N}[\frac{1}{2}(r_{n,k-1} - \alpha V E_{P_{n-1,k}}[s\psi(x_n \cdot w)])^2 - \frac{1}{2}(y_n - g(x_n))^2]. \tag{5.72}$$

By Lemma 5.3, square regret can be upper bound by log regret plus an additional term

$$\bar{R}_{N,k}^{\text{square}} \leq \frac{1}{\beta N}\sum_{n=1}^{N}\Big[-\log\Big(\int \frac{\beta}{\sqrt{2\pi}} e^{-\frac{\beta}{2}(r_{n,k-1} - \alpha s V \psi(x_n \cdot w))^2} p_{n-1,k}(w, s | x^{n-1}, r_{k-1}^{n-1})\eta(dw, ds)\Big)\Big] \tag{5.73}$$

$$+ \frac{\beta}{2N}\sum_{n=1}^{N}(\alpha |r_{n,k-1}| a_0 V + \frac{1}{2}\alpha^2 a_0^2 V^2)^2 \tag{5.74}$$

$$- \frac{1}{2}\frac{1}{N}\|y - g\|_N^2 + \frac{1}{2\beta}\log(\frac{\beta}{2\pi}). \tag{5.75}$$

Define the Bayes factors

$$Z_{n,k} = \Big(\frac{\beta}{2\pi}\Big)^{\frac{n}{2}} \int e^{-\frac{\beta}{2}\sum_{i=1}^{n}(r_{i,k-1} - \alpha s V \psi(x_i \cdot w))^2} P_0(dw, ds). \tag{5.76}$$

Then the predictive densities $p_{n-1,k}(y_n | x_n, x^{n-1}, r_{k-1}^{n-1})$ are equal to the ratio of Bayes factors

$$p_{n-1,k}(y_n | x_n, x^{n-1}, r_{k-1}^{n-1}) = \frac{Z_{n,k}}{Z_{n-1,k}}. \tag{5.77}$$

Via a telescoping sum of log factors, we have the following upper bound on the square

regret using only the final Bayes factor $Z_{N,K}$ inside the log,

$$\bar{R}_{N,k}^{\text{square}} \leq \frac{1}{\beta N}\Big[-\log\Big(\int e^{-\frac{\beta}{2}\sum_{n=1}^{N}(r_{n,k-1}-\alpha s V\psi(x_n\cdot w))^2} P_0(dw,ds)\Big)\Big] \tag{5.78}$$

$$+\frac{\beta}{2N}\sum_{n=1}^{N}(\alpha|r_{n,k-1}|a_0V+\frac{1}{2}\alpha^2a_0^2V^2)^2-\frac{1}{2}\frac{1}{N}\|y-g\|_N^2. \tag{5.79}$$

Let $s^*, w^{*,\text{cont}}$ be the sign continuous weight vector that would minimize the square loss with the $r_{n,k-1}$ residuals,

$$s^*, w^{*,\text{cont}} = \text{argmin}_{w\in S_1^d, s\in\{-1,1\}}\sum_{n=1}^{N}(r_{n,k-1}-\alpha s V\psi(x_n\cdot w))^2. \tag{5.80}$$

Add and subtract the square distance from this residual inside the exponent of the integral, so we have

$$\frac{1}{\beta N}\Big[-\log\Big(\int e^{-\frac{\beta}{2}\sum_{n=1}^{N}(r_{n,k-1}-\alpha s V\psi(x_n\cdot w))^2+\frac{\beta}{2}\sum_{n=1}^{N}(r_{n,k-1}-\alpha s V\psi(x_n\cdot w^{*,\text{cont}}))^2} P_0(dw,ds)\Big)\Big] \tag{5.81}$$

$$+\frac{\beta}{2N}\sum_{n=1}^{N}(\alpha|r_{n,k-1}|a_0V+\frac{1}{2}\alpha^2a_0^2V^2)^2 \tag{5.82}$$

$$+\frac{1}{2}\frac{1}{N}\|y-(1-\alpha)\hat{f}_{n-1,k-1}-\alpha s V\psi_{w^{*,\text{cont}}}\|_N^2-\frac{1}{2}\frac{1}{N}\|y-g\|_N^2. \tag{5.83}$$

By Lemma 5.4, for any continuous weight vector, there exists at least one discrete weight vector that has a certain bounded regret compared to the continuous weight vector. There are less than $(2d+1)^M$ points in the discrete state space, thus we have the bound on the

CGF term of the form

$$\frac{1}{\beta N}\Big[-\log\Big(\int e^{-\frac{\beta}{2}\sum_{n=1}^{N}(r_{n,k-1}-\alpha sV\psi(x_n\cdot w))^2+\frac{\beta}{2}\sum_{n=1}^{N}(r_{n,k-1}-\alpha sV\psi(x_n\cdot w^{*,\text{cont}}))^2}P_0(dw,ds)\Big)\Big] \tag{5.84}$$

$$\leq \frac{M\log(2d+1)}{\beta N}+\frac{1}{2}a_2\alpha V\frac{1}{N}\sum_{n=1}^{N}|r_{n,k-1}|\frac{1}{M}+\frac{1}{2}\alpha^2V^2(a_1^2+a_0a_2)\frac{1}{M}. \tag{5.85}$$

By Lemma 5.2, the regret using the the optimal neuron as a fixed relationship to he regret using the previous fit,

$$\frac{1}{2}\frac{1}{N}\|y-(1-\alpha)\hat{f}_{\cdot,k-1}-\alpha sV\psi_{w^{*,\text{cont}}}\|_N^2-\frac{1}{2}\frac{1}{N}\|y-g\|_N^2 \tag{5.86}$$

$$\leq(1-\alpha)\frac{1}{2}(\frac{1}{N}\|(\|y-\hat{f}_{\cdot,k-1}\|_N^2-\frac{1}{N}\|y-g\|_N^2)+\frac{1}{2}\alpha^2a_0^2V^2. \tag{5.87}$$

The conclusion of these two results is that log regret using our Bayesian posterior density is nearly the same as if we had been able to do greedy optimization. We pay the price of some additional terms which will appear in the object we call τ. Putting these results together gives the recursive relationship for square regret,

$$\bar{R}_{N,k}^{\text{square}}\leq(1-\alpha)(\bar{R}_{N,k-1}^{\text{square}})+\frac{1}{2}\alpha^2a_0^2V^2 \tag{5.88}$$

$$+\frac{\beta}{2N}\sum_{n=1}^{N}(\alpha|r_{n,k-1}|a_0V+\frac{1}{2}\alpha^2a_0^2V^2)^2 \tag{5.89}$$

$$+\frac{M\log(2d+1)}{\beta N}+\frac{1}{2}\frac{1}{M}a_2\alpha V\frac{1}{N}\sum_{n=1}^{N}|r_{n,k-1}|+\frac{1}{2}\alpha^2V^2(a_1^2+a_0a_2)\frac{1}{M}. \tag{5.90}$$

The maximum the residual can be is controlled by its bounded inputs,

$$|r_{n,k-1}|\leq C_N. \tag{5.91}$$

This yields upper bound

$$\bar{R}^{\text{square}}_{N,k} \leq (1-\alpha)(\bar{R}^{\text{square}}_{N,k-1}) + \frac{1}{2}\alpha^2 a_0^2 V^2 \tag{5.92}$$

$$+ \frac{\beta}{2}(\alpha C_N a_0 V + \frac{1}{2}\alpha^2 a_0^2 V^2)^2 \tag{5.93}$$

$$+ \frac{M\log(2d+1)}{\beta N} + \frac{1}{2}\frac{1}{M}a_2 \alpha V C_N + \frac{1}{2}\alpha^2 V^2 (a_1^2 + a_0 a_2)\frac{1}{M}. \tag{5.94}$$

The terms aside from $(1-\alpha)\bar{R}^{\text{sqaure}}_{N,k-1}$ are denoted as τ in the theorem statement. □

Thus, we have established the kind of recursive relationship we would like to arise for our greedy Bayes estimator. Note this relationship is easy to establish for greedy optimization, in fact Lemma 5.2 essentially does this for greedy optimization. By keeping careful track of how cumulant generating functions are close to their means (with certain controlled difference terms), we can relate the posterior means of our estimator to this recursion we would get using greedy optimization, with additional terms absorbed by the τ we have defined.

Thus, we must consider the τ that appears in our recursive result, and determine the correct choices of α, β, M, K, N that can give good risk control. We will see the following choices:

$$K = \big(\frac{N+1}{\log(2d+1)}\big)^{\frac{1}{3}} \log\Big(\big(\frac{N+1}{\log(2d+1)}\big)^{\frac{1}{3}}\Big) \qquad M = \big(\frac{N+1}{\log(2d+1)}\big)^{\frac{1}{3}} \tag{5.95}$$

$$\alpha = \frac{\log(K)}{K} \qquad \beta = 1 \tag{5.96}$$

gives a bound on average square regret of the order $O([(\log d)/N]^{1/3})$.

Theorem 5.2. *Let $(x_i)_{i=1}^N$ be a sequence of input values with all $x_i \in [-1,1]^d$. Let g be a target function and let h be any element of Hull$_N(V\Psi)$, the closure of the convex hull of signed neurons scaled by V. Let P_0 be the uniform prior on $(S^d_{1,M}) \times \{-1,1\}$. For any*

sequence of values $(y_i)_{i=1}^N$, define the terms

$$\epsilon_n = y_n - g(x_n) \qquad \tilde{\epsilon}_n = y_n - h(x_n). \tag{5.97}$$

Then the average square regret for the y sequence using g as the competitor at level K can be bound as

$$\bar{R}_{N,K}^{square} \leq (1-\alpha)^K \bar{R}_{N,0}^{square} + \frac{1}{2}\alpha a_0^2 V^2 + \frac{\beta}{2}\alpha a_0^2 V^2 (C_N + \frac{1}{2}\alpha a_0 V)^2 \tag{5.98}$$

$$+ \frac{M\log(2d+1)}{\alpha\beta N} + \frac{1}{2}\frac{1}{M} a_2 V C_N + \frac{1}{2}\alpha V^2 (a_1^2 + a_0 a_2)\frac{1}{M} \tag{5.99}$$

$$+ \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}(\tilde{\epsilon}_n^2 - \epsilon_n^2). \tag{5.100}$$

In particular, assume

$$\log\big(\big(\frac{N+1}{\log(2d+1)}\big)^{\frac{1}{3}}\big) \geq 1, \tag{5.101}$$

which is a mild assumption about having a certain amount of data. With the choice of parameters

$$K = \big(\frac{N+1}{\log(2d+1)}\big)^{\frac{1}{3}} \log\Big(\big(\frac{N+1}{\log(2d+1)}\big)^{\frac{1}{3}}\Big) \qquad M = \big(\frac{N+1}{\log(2d+1)}\big)^{\frac{1}{3}} \tag{5.102}$$

$$\alpha = \frac{\log(K)}{K} \qquad \beta = 1 \tag{5.103}$$

We achieve a bound of the form

$$\bar{R}^{square}_{N,K} \leq \big(\frac{\log(2d+1)}{N}\big)^{\frac{1}{3}}(\bar{R}^{square}_{N,0} + 1.4\frac{1}{2}a_0^2V^2 + 1 + \frac{1}{2}a_2VC_N) \tag{5.104}$$

$$+ 1.4\big(\frac{\log(2d+1)}{N}\big)^{\frac{1}{3}}\frac{1}{2}a_0^2V^2\big(C_N + 1.4\big(\frac{\log(2d+1)}{N}\big)^{\frac{1}{3}}\frac{1}{2}\alpha a_0V\big)^2 \tag{5.105}$$

$$+ 1.4\big(\frac{\log(2d+1)}{N}\big)^{\frac{2}{3}}\frac{1}{2}V^2(a_1^2 + a_0a_2) \tag{5.106}$$

$$+ \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}(\tilde{\epsilon}_n^2 - \epsilon_n^2). \tag{5.107}$$

If we consider a_0, a_1, a_2, V as fixed constants this is of the order

$$\bar{R}^{square}_{N,K} = O\Big(\big(\frac{\log(d)}{N}\big)^{\frac{1}{3}}(\max_{n=1}^{N}|y_n|)^2\Big) + \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}(\tilde{\epsilon}_n^2 - \epsilon_n^2). \tag{5.108}$$

Proof. Theorem 1 requires the competitor function to be an element of the closure of the convex Hull, which we have not assumed g is. However, for regret relative to g it is simple to add and subtract $\frac{1}{2}(y_n - h(x_n))^2$ to instead consider the regret of our fits relative to h and then the regret of h relative to g (which is the average of the $1/2[(\epsilon'_n)^2 - \epsilon_n^2]$).

$$\bar{R}^{\text{square}}_{N,k} = \frac{1}{N}\sum_{n=1}^{N}[\frac{1}{2}(y_n - \hat{f}_{n-1,k}(x_n))^2 - \frac{1}{2}(y_n - g(x_n))^2] \tag{5.109}$$

$$= \frac{1}{N}\sum_{n=1}^{N}[\frac{1}{2}(y_n - \hat{f}_{n-1,k}(x_n))^2 - \frac{1}{2}(y_n - h(x_n))^2] \tag{5.110}$$

$$+ \frac{1}{N}\sum_{n=1}^{N}[\frac{1}{2}(y_n - h(x_n))^2 - \frac{1}{2}(y_n - g(x_n))^2] \tag{5.111}$$

$$= \frac{1}{N}\sum_{n=1}^{N}[\frac{1}{2}(y_n - \hat{f}_{n-1,k}(x_n))^2 - \frac{1}{2}(y_n - h(x_n))^2] + \frac{1}{N}\sum_{n=1}^{N}[\frac{1}{2}\tilde{\epsilon}_n^2 - \frac{1}{2}\epsilon_n^2]. \tag{5.112}$$

So we can can consider the regret of our fits relative to h instead and simply let the difference between $\tilde{\epsilon}_n$ and ϵ_n be as it is. In particular, h can be chosen as the projection of g

into $\text{Hull}_N(V\psi)$. Nonetheless, with h now acting as our competitor function we can apply the result of Theorem 5.1 and get the recursive relationship. Then with this recursion established, we can apply Lemma 5.1 to have a bound on the final regret,

$$\bar{R}_{N,K}^{\text{square}} \leq (1-\alpha)^K \bar{R}_{N,0}^{\text{square}} + \frac{\tau}{\alpha}. \tag{5.113}$$

With τ as defined in equation (5.68). Divide each term in (5.68) by out α. Our first consideration is the term $(1-\alpha)^K$. If we set $\alpha = \frac{1}{K}$, this would converge to e^{-1} for large K, which is a constant limit not one decaying in K. However, if we set $\alpha = \frac{\log K}{K}$, we have the relationship

$$(1 - \frac{\log(K)}{K})^K \leq \frac{1}{K}. \tag{5.114}$$

This then gives a $1/K$ control on the term with $\bar{R}_{N,0}^{\text{square}}$.

Now let

$$K = \big(\frac{(N+1)}{\log(2d+1)}\big)^{\frac{1}{3}} \Big(\frac{\log\big(\frac{(N+1)}{\log(2d+1)}\big)}{3}\Big) \qquad M = \big(\frac{(N+1)}{\log(2d+1)}\big)^{\frac{1}{3}}. \tag{5.115}$$

Assume that

$$\frac{\log\big(\frac{(N+1)}{\log(2d+1)}\big)}{3} \geq 1. \tag{5.116}$$

Note that for $x > 1$ we have

$$0 \leq \frac{\log(x)}{x} \leq 0.4. \tag{5.117}$$

So we have the bound

$$\frac{\log(K)}{K} = \big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}}\big(1 + \frac{\log\Big(\frac{\log\left(\frac{(N+1)}{\log(2d+1)}\right)}{3}\Big)}{\frac{\log\left(\frac{(N+1)}{\log(2d+1)}\right)}{3}}\big) \tag{5.118}$$

Which implies

$$\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}} \leq \frac{\log(K)}{K} \leq 1.4\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}}. \tag{5.119}$$

Inputting these values yields the bound.

□

5.4 IID Risk Control for Greedy Bayes

We now consider risk control for iid sequences. Risk can be expressed as an expected regret, thus the proof of this result follows by redoing the proof of the arbitrary regret theorem inside of an expectation using the data distribution. We can then take advantage of bounding $E_{P_{X^{N+1},Y^{N+1}}}[|r_{n,k-1}|]$ in an average sense, instead of bounding $|r_{n,k-1}|$ in a worst case sense. If $E[Y_i|X_i] = g(X_i)$ with absolute value of g bounded by b, and $\text{Var}(Y_i|X_i) \leq \sigma^2$, we can replace each instance of $C_N = (\max_{n=1}^{N} |y_n| + a_0 V)$ in equation 5.98 with $\sqrt{\sigma^2 + (b + a_0 V)^2}$ instead.

Theorem 5.3. *Let g be a target function with absolute value bounded by b and let $\tilde{g}$ be its $L_2(P_X)$ projection into the closure of the convex hull of signed neurons scaled by V. Let P_0 be the uniform prior on $(S_{1,M}^d) \times \{-1, 1\}$. Let $(X_i, Y_i)_{i=1}^N$ be training data iid with conditional mean $g(X_i)$ and conditional variance $\sigma_{X_i}^2$ with variance bound $\sigma_x^2 \leq \sigma^2$.*

Assume the data distribution P_X has support in $[-1,1]^d$. Define the value

$$\tau' = \frac{1}{2}\alpha^2 a_0^2 V^2 + \frac{\beta}{2}(\alpha\sqrt{\sigma^2 + (b+a_0V)^2}a_0V + \frac{1}{2}\alpha^2 a_0^2 V^2)^2] + \frac{M\log(2d+1)}{\beta(N+1)} \quad (5.120)$$

$$+ \frac{1}{2}\frac{1}{M}a_2\alpha V\sqrt{\sigma^2 + (b+a_0V)^2} + \frac{1}{2}\alpha^2 V^2(a_1^2 + a_0a_2)\frac{1}{M}. \quad (5.121)$$

Then the mean squared statistical risk of the Cesàro average of the level K estimators $\hat{g}$ is upper bounded by

$$E[\frac{1}{2}(g(X) - \hat{g}(X))^2] \leq E[\frac{1}{2}(\tilde{g}(X) - g(X))^2] + (1-\alpha)^K(\sigma^2 + (b+a_0V)^2) + \frac{\tau'}{\alpha}. \quad (5.122)$$

In particular, assume

$$\log\big((\frac{N+1}{\log(2d+1)})^{\frac{1}{3}}\big) \geq 1, \quad (5.123)$$

which is a mild assumption about having a certain amount of data. With the choice of parameters

$$K = (\frac{N+1}{\log(2d+1)})^{\frac{1}{3}}\log\Big((\frac{N+1}{\log(2d+1)})^{\frac{1}{3}}\Big) \qquad M = (\frac{N+1}{\log(2d+1)})^{\frac{1}{3}} \quad (5.124)$$

$$\alpha = \frac{\log(K)}{K} \qquad \beta = 1 \quad (5.125)$$

we have the bound

$$E[\frac{1}{2}(g-\hat{g})^2] \leq E[\frac{1}{2}(\tilde{g}(X)-g(X))^2] \tag{5.126}$$

$$+\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}}\Big[1.4\frac{1}{2}a_0^2V^2+1+\sigma^2+(b+a_0V)^2+\frac{1}{2}a_2V\sqrt{\sigma^2+(b+a_0V)^2}\Big] \tag{5.127}$$

$$+1.4\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}}\frac{1}{2}a_0^2V^2\big(\sqrt{\sigma^2+(b+a_0V)^2}+1.4\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}}\frac{1}{2}a_0V\big)^2 \tag{5.128}$$

$$+1.4\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{2}{3}}\frac{1}{2}V^2(a_1^2+a_0a_2). \tag{5.129}$$

If one considers $\sigma, b, V, a_0, a_1, a_2$ as fixed values not growing in N or d, this bound is of the order,

$$E[\frac{1}{2}(g-\hat{g})^2] = E[\frac{1}{2}(\tilde{g}(X)-g(X))^2] + O(\big(\frac{\log(d)}{N}\big)^{\frac{1}{3}}). \tag{5.130}$$

Proof. This proof follows much of the same steps as the square regret proof for arbitrary sequences Theorem 5.2. However, in the arbitrary sequence proof we had to settle for an upper bound on the residuals determined by the maximum absolute value of y_n in our training set. We do not want to have such a value appear in this proof, as for unbounded distributions this will be growing with N and affect our rates. Instead, we take advantage of our objects now existing inside an expectation, thus we can instead take the expectation of $|r_{n,k-1}|$ with respect to the data distribution, which will be controlled by the bound the absolute value of the mean function g and the standard deviation σ. Thus we can allow the the y_n to have very large values, and it is only the bound on the mean function and the variance which will determine the constants of our risk.

We will now set up a recursion for the expected log regret at a level k and show

$$E[\bar{R}_{N,k}^{\text{square}}] \leq (1-\alpha)E[\bar{R}_{N,k-1}^{\text{square}}] + \tau'. \tag{5.131}$$

First, we show that the square risk with the Cesàro average of the level K estimators is equal to an expected regret at level K. Let $(X_i, Y_i)_{i=1}^N$ be the training data iid from the data distribution $P_{X,Y}$. Let $(X, Y) = (X_{N+1}, Y_{N+1})$ be a new input and response pair independent from the data distribution. Then our square risk is an expectation over both the training data and new data pair,

$$\frac{1}{2}E_{P_{X^{N+1},Y^{N+1}}}[(g(X) - \hat{g}(X))^2] \leq \frac{1}{2}\frac{1}{N+1}\sum_{n=0}^{N} E_{P_{X^{N+1},Y^{N+1}}}[(g(X) - \hat{f}_{n,K}(X))^2] \tag{5.132}$$

$$= \frac{1}{2}E_{P_{X^{N+1},Y^{N+1}}}[\sum_{n=0}^{N}\frac{(g(X_{n+1}) - \hat{f}_{n,K}(X_{n+1}))^2}{N+1}] \tag{5.133}$$

$$= \frac{1}{2}E_{P_{X^{N+1},Y^{N+1}}}[\sum_{n=0}^{N}\frac{(Y_{n+1} - \hat{f}_{n,K}(X_{n+1}))^2 - (Y_{n+1} - g(X_{n+1}))^2}{N+1}], \tag{5.134}$$

where we have added in the Y_{n+1} using the fact that $Y_{n+1} - g(X_{n+1})$ is mean 0 under the data distribution, and conditionally independent of $\hat{f}_{n,K}(X_{n+1})$ since $\hat{f}_{n,k}$ is only trained on data up to index n in the training data. This object is now exactly an expected square regret at level K. Thus, we consider the recursion of lower k levels to provide a bound on the expected regret at level K.

Consider the expected regret using the level k estimators,

$$\frac{1}{2}E_{P_{X^{N+1},Y^{N+1}}}[\sum_{n=0}^{N}\frac{(Y_{n+1} - \hat{f}_{n,k}(X_{n+1}))^2 - (Y_{n+1} - g(X_{n+1}))^2}{N+1}]. \tag{5.135}$$

g is our target function, which is not assumed to live in the $L_2(P_X)$ closure of the convex hull of signed neurons scaled by V. Thus, let $\tilde{g}$ be its $L_2(P_X)$ projection into Hull$(V\psi)$. $\tilde{g}$ is not per say a specific finite width neural network, but a limit thereof. Let $\tilde{g}_\epsilon$ be a specific finite width neural network that is ϵ close to $\tilde{g}$ in $L_2(P_X)$ distance. Note that for any sequence $(x_i)_{i=1}^{N+1}$, that $(\tilde{g}_\epsilon(x_i))_{i=1}^{N+1}$ is then an element of the Euclidean closure $\text{Hull}_{N+1}(V\psi)$, which is a condition needed in our previous approximation lemmas. This

is all to say, add and subtract $\frac{1}{2}\|y-\tilde{g}_\epsilon\|_N^2$, and the recursion we really want to study is the regret with respect to $\tilde{g}_\epsilon$ at different levels k, plus the regret of $\tilde{g}_\epsilon$ relative to g,

$$\frac{1}{2}E_{P_{X^{N+1},Y^{N+1}}}[\sum_{n=0}^{N}\frac{(Y_{n+1}-\hat{f}_{n,k}(X_{n+1}))^2-(Y_{n+1}-\tilde{g}_\epsilon(X_{n+1}))^2}{N+1}] \tag{5.136}$$

$$+\frac{1}{2}E_{P_{X^{N+1},Y^{N+1}}}[\sum_{n=0}^{N}\frac{(Y_{n+1}-\tilde{g}_\epsilon(X_{n+1}))^2-(Y_{n+1}-g(X_{n+1}))^2}{N+1}] \tag{5.137}$$

Consider then the object inside the expectation of equation (5.136). This is exactly a square regret with respect to an element of $\text{Hull}_N(V\psi)$. Thus the proof technique of Theorem 5.1 to establish a recursion will apply. Follow that proof the same way up to equation (5.90). Picking up at equation (5.90) we have the result (note these regrets are with respect to $\tilde{g}_\epsilon$ as the competitor)

$$E_{P_{X^{N+1},Y^{N+1}}}[\bar{R}_{N,k}^{\text{square}}] \leq (1-\alpha)E_{P_{X^{N+1},Y^{N+1}}}[\bar{R}_{N,k-1}^{\text{square}}]+\frac{1}{2}\alpha^2a_0^2V^2 \tag{5.138}$$

$$+\frac{\beta}{2(N+1)}\sum_{n=1}^{N+1}E_{P_{X^{N+1},Y^{N+1}}}[(\alpha|r_{n,k-1}|a_0V+\frac{1}{2}\alpha^2a_0^2V^2)^2] \tag{5.139}$$

$$+\frac{M\log(2d+1)}{\beta(N+1)}+\frac{1}{2}\frac{1}{M}a_2\alpha V\frac{1}{N+1}\sum_{n=1}^{N+1}E_{P_{X^{N+1},Y^{N+1}}}[|r_{n,k-1}|] \tag{5.140}$$

$$+\frac{1}{2}\alpha^2V^2(a_1^2+a_0a_2)\frac{1}{M}. \tag{5.141}$$

In the arbitrary sequence regret proof, we had upper bounded the residuals terms using equation (5.91). Now we can instead use their expectation and bound this instead. Note y_n

is mean $g(x_n)$ and $y_n|x_n$ has variance less than σ^2. Thus via a bias variance decomposition,

$$E_{P_{X^{N+1},Y^{N+1}}}[(y_n - \hat{f}_{n-1,k-1}(x_n))^2] = \sigma^2 + (g(x_n) - E_{P_{X^{N+1},Y^{N+1}}}[\hat{f}_{n-1,k-1}(x_n)])^2 \tag{5.142}$$

$$\leq \sigma^2 + (b + a_0 V)^2. \tag{5.143}$$

In a similar way the expected absolute value is less than $\sqrt{\sigma^2 + (b + a_0 V)^2}$ via a Jensen's inequality. Thus, we can apply this bound on the expectations and we have overall bound

$$E_{P_{X^{N+1},Y^{N+1}}}[\bar{R}^{\text{square}}_{N,k}] \leq (1-\alpha) E_{P_{X^{N+1},Y^{N+1}}}[\bar{R}^{\text{square}}_{N,k-1})] + \frac{1}{2}\alpha^2 a_0^2 V^2 \tag{5.144}$$

$$+ \frac{\beta}{2}(\alpha\sqrt{\sigma^2 + (b + a_0 V)^2} a_0 V + \frac{1}{2}\alpha^2 a_0^2 V^2)^2 \tag{5.145}$$

$$+ \frac{M \log(2d+1)}{\beta(N+1)} + \frac{1}{2}\frac{1}{M} a_2 \alpha V \sqrt{\sigma^2 + (b + a_0 V)^2} \tag{5.146}$$

$$+ \frac{1}{2}\alpha^2 V^2 (a_1^2 + a_0 a_2)\frac{1}{M} \tag{5.147}$$

which notably, does not depend on the maximum value of the $y's$ but only on σ and b. We have now established a recursive relationship of the form

$$E_{P_{X^{N+1},Y^{N+1}}}[\bar{R}^{\text{square}}_{N,k}] \leq (1-\alpha) E_{P_{X^{N+1},Y^{N+1}}}[\bar{R}^{\text{square}}_{N,k-1})] + \tau'. \tag{5.148}$$

Applying Lemma 5.1, this gives a bound on the level K regret

$$E_{P_{X^{N+1},Y^{N+1}}}[\bar{R}^{\text{square}}_{N,k}] \leq (1-\alpha)^K E_{P_{X^{N+1},Y^{N+1}}}[\bar{R}^{\text{square}}_{N,0}] + \frac{\tau'}{\alpha} \tag{5.149}$$

$$\leq (1-\alpha)^K (\sigma^2 + (b + a_0 V)^2) + \frac{\tau'}{\alpha}. \tag{5.150}$$

As before, set $\alpha = \frac{\log K}{K}$ so we have

$$(1-\alpha)^K \leq \frac{1}{K}. \tag{5.151}$$

This gives control on the final expected regret

$$E_{P_{X^{N+1},Y^{N+1}}}[\bar{R}_{N,k}^{\text{square}}] \leq \frac{1}{K}(\sigma^2 + (b + a_0 V)^2) + \frac{1}{2}\alpha a_0^2 V^2 \tag{5.152}$$

$$+ \frac{\beta}{2}\alpha a_0^2 V^2 (\sqrt{\sigma^2 + (b + a_0 V)^2} + \frac{1}{2}\alpha a_0 V)^2 \tag{5.153}$$

$$+ \frac{M \log(2d+1)}{\alpha\beta(N+1)} + \frac{1}{2}\frac{1}{M} a_2 V \sqrt{\sigma^2 + (b + a_0 V)^2} \tag{5.154}$$

$$+ \frac{1}{2}\alpha V^2 (a_1^2 + a_0 a_2)\frac{1}{M}. \tag{5.155}$$

Now let

$$K = \big(\frac{(N+1)}{\log(2d+1)}\big)^{\frac{1}{3}} \Big(\frac{\log\big(\frac{(N+1)}{\log(2d+1)}\big)}{3}\Big) \qquad M = \big(\frac{(N+1)}{\log(2d+1)}\big)^{\frac{1}{3}}. \tag{5.156}$$

Assume that

$$\frac{\log\big(\frac{(N+1)}{\log(2d+1)}\big)}{3} \geq 1. \tag{5.157}$$

Note that for $x > 1$ we have

$$0 \leq \frac{\log(x)}{x} \leq 0.4. \tag{5.158}$$

So we have the bound

$$\frac{\log(K)}{K} = \big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}} (1 + \frac{\log\Big(\frac{\log\big(\frac{(N+1)}{\log(2d+1)}\big)}{3}\Big)}{\frac{\log\big(\frac{(N+1)}{\log(2d+1)}\big)}{3}}) \tag{5.159}$$

Which implies

$$\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}} \leq \frac{\log(K)}{K} \leq 1.4\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}}. \tag{5.160}$$

Inputting these values yields the bound

$$E_{P_{X^{N+1},Y^{N+1}}}[\bar{R}^{\text{square}}_{N,k}] \leq \big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}}(\sigma^2 + (b+a_0V)^2) + 1.4\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}}\frac{1}{2}a_0^2V^2 \tag{5.161}$$

$$+ 1.4\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}}\frac{\beta}{2}a_0^2V^2(\sqrt{\sigma^2 + (b+a_0V)^2} + 1.4\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}}\frac{1}{2}a_0V)^2 \tag{5.162}$$

$$+ \frac{(\log(2d+1))^{\frac{1}{3}}}{\beta(N+1)^{\frac{1}{3}}} + \big(\frac{\log(2d+1)}{N+1}\big)^{\frac{1}{3}}\frac{1}{2}a_2V\sqrt{\sigma^2 + (b+a_0V)^2} \tag{5.163}$$

$$+ 1.4\big(\frac{\log(2d+1)}{N+1}\big)^{\frac{2}{3}}\frac{1}{2}V^2(a_1^2 + a_0a_2). \tag{5.164}$$

β only appears in two of these terms. We can choose $\beta = \sqrt{\frac{1}{\sigma^2+(b+a_0V)^2}}$ to change the constants of these two terms, but this will not effect other terms or the order of dependence in N. So for simplicity, let $\beta = 1$.

This provides the bound on term (5.136), which is the regret with respect to $\tilde{g}_\epsilon$. For equation (5.137), it can be shown to be close to $\frac{1}{2}E_{P_X}[(\tilde{g}(X) - g(X))^2] + O(\sqrt{\epsilon})$. Taking epsilon to 0 removes the extra terms.

If we consider $\sigma, b, V, a_0, a_1, a_2$ as fixed and not growing in N or d, then this bound is $O([(\log d)/N]^{1/3})$.

□

Chapter 6

Additional Content and Discussion

6.1 Combining Continuous and Discrete Results

We note that unfortunately, our results showing when a log-concave coupling occurs and our results providing risk control are currently not compatible. For our log-concave coupling results, we use a continuous prior which is uniform over the continuous ℓ_1 ball S_1^d. For our risk control results, we use a discrete prior with grid size $1/M$ for some integer M, and let it be uniform over the intersection of the continuous ball with the grid of intervals $\frac{1}{M}$, which we call $S_{1,M}^d$.

It seems promising that one can combine these results by using a large M and showing that the behavior of the discrete grid prior is very similar to the continuous prior. From our log-concave coupling results, we consider K as some fractional power of $N^p, 0 < p < 1$, and dimensions $d > N^q$, for a power $q > 1$. From our risk control results, we have bound on the risk of the order (ignoring constants),

$$\frac{MK\log(d)}{\beta N} + \frac{1}{M} + \frac{1}{K} + \beta. \tag{6.1}$$

Thus, it is clear $MK \leq \beta N$ to have any kind of risk control that is decaying in N. The

optimal choice is to have $M = K = \frac{1}{\beta} = (N/(\log d))^{\frac{1}{4}}$, but this is not the only choice. We may want to work with larger M in order to make our discrete grid prior behave more like the continuous prior. Thus, we can scale M up to $(N/(\log d))^p$ for $p < 1$ and still get a (notable worse than $1/4$) decay rate of $[(\log d)/N]^{\frac{(1-p)}{3}}$ for our overall error rate. $p = 1/4$ is the solution to $p = (1-p)/3$ when all the terms in the risk bound are of the same order. Using a larger p will give worse risk bounds.

Notably this forces $M < N$ as a hard upper bound to get any form of risk control, and this makes it difficult to show the discrete prior and continuous prior have comparable performance.

To connect the discrete and continuous prior, note that both can be considered as marginal distributions of the same joint distribution. This allows us to think of $w^{\text{disc}} = (w_k^{\text{disc}})_{k=1}^K$ and $w^{\text{cont}} = (w_k^{\text{cont}})_{k=1}^K$ as jointly distributed variables. Then any expectation using either the continuous or discrete prior, is really an expectation using the joint distribution with the other variable integrated out.

Consider P_0 as a joint distribution on $(S_1^d)^K \times (S_{1,M}^d)^K$, with the continuous random vector $w^{\text{cont}} \in (S_1^d)^K$ and the discrete random vector $w^{\text{disc}} \in (S_{1,M}^d)^K$. Consider the marginal distribution on w^{cont} as treating each w_k^{cont} vector as independent uniform on S_1^d. Consider an additional coordinate for each w_k^{cont} vector to track its ℓ_1 distance from 1, $w_{k,d+1}^{\text{cont}} = 1 - \sum_{j=1}^d |w_{k,j}^{\text{cont}}|$.

Then define the conditional distribution on $w_k^{\text{disc}} | w_k^{\text{cont}}$ as follows. Force the signs of the coordinates to stay the same, $\text{sign}(w_{k,j}^{\text{disc}}) = \text{sign}(w_{k,j}^{\text{cont}})$, and have the absolute values be distributed as $1/M$ times a Multinomial$(M, |w_{k,1}^{\text{cont}}|, \ldots, |w_{k,d+1}^{\text{cont}}|)$ distribution. That is, the conditional probability mass function of the absolute values of the discrete vector can be written as

$$p_0(|w_k^{\text{disc}}| \,\big|\, |w_k^{\text{cont}}|) = \frac{M!}{\prod_{j=1}^{d+1}(M|w_{k,j}^{\text{disc}}|)!} \prod_{j=1}^{d+1} |w_{k,j}^{\text{cont}}|^{M|w_{k,j}^{\text{disc}}|}. \tag{6.2}$$

Note the discrete vector's coordinates themselves are whole number multiples of $1/M$, thus M times the discrete vector coordinates are whole numbers between 0 and M. There is also a $w_{k,d+1}^{\text{disc}}$ coordinate in this construction which is 1 minus the sum of the other coordinates. Then the overall joint distribution P_0 has a density (with respect to the product of Lebesgue measure on $(S_1^d)^K$ and counting measure on $(S_{1,M}^d)^K$) of the form

$$p_0(w^{\text{cont}}, w^{\text{disc}}) = \prod_{k=1}^{K} p_0(w_k^{\text{cont}}) p_0(w_k^{\text{disc}} | w_k^{\text{cont}}) \tag{6.3}$$

$$= \prod_{k=1}^{K} \text{Uniform}_{S_1^d}(w_k^{\text{cont}}) \text{Multinomial}_{M,|w_k^{\text{cont}}|}(M|w_k^{\text{disc}}|) \prod_{j=1}^{d+1} 1\{\text{sign}(w_{k,j}^{\text{cont}}) = \text{sign}(w_{k,j}^{\text{disc}})\}. \tag{6.4}$$

This results in the marginal distribution for w^{disc} to treat each w_k^{disc} as uniform on $S_{1,M}^d$. This is a special case of the Dirichlet-Multinomial distribution using the all 1's vector in the parameter vector of the Dirichlet distribution [54, Chapter 6].

Note than than $E[w_k^{\text{disc}} | w_k^{\text{cont}}] = w_k^{\text{cont}}$. Furthermore, $w_k^{\text{disc}} | w_k^{\text{cont}}$ can be realized as an average of M iid random variables, so its variance will be like $\frac{1}{M}$.

Let $w_{k,j}^{\text{cont}}$ have sign $s_{k,j}$. Then let e_j be the vector in $\mathbb{R}^{d+1}$ with a 1 in the j coordinate and 0 else. Let $Z_t, t \in \{1, \cdots, M\}$ be a random variable where $Z_t = s_{k,j} e_j$ with probability $|w_{k,j}^{\text{cont}}|$. Then we have

$$w_k^{\text{disc}} = \sum_{t=1}^{M} \frac{1}{M} Z_t. \tag{6.5}$$

The Z_t are then iid random index selections, similar to how a Binomial is constructed by a sum of Bernoulli random variables. This is the Multinomial analog. Then for any vector

x with $\|x\|\|_\infty \leq 1$, we have

$$\text{Var}(x \cdot w_k^{\text{disc}} | w_k^{\text{cont}}) = \sum_{t=1}^{M} \frac{1}{M^2} \text{Var}(x \cdot Z_t | w_k^{\text{cont}}) \leq \frac{1}{M}. \tag{6.6}$$

So the w_k^{disc} vectors are centered at the continuous ones, and have low variance around their mean. One would hope to use these results to relate the discrete and continuous variables.

Our primary object for risk control is the cumulant generating function under the prior of the square loss used in the index of resolvability. That is, in the study of log regret we have the object

$$\frac{-\log E_{P_0}[e^{-\frac{\beta}{2}\sum_{i=1}^{N+1}(y_i - f_{w^{\text{disc}}}(x_i))^2}]}{\beta(N+1)}. \tag{6.7}$$

When the variable in the exponent is w^{disc}, we are able to upper bound this object using the index of resolvability. Really, with the $E[Y_i|X_i] = g(X_i)$ and X_i being iid from a data distribution P_X, it is actually sufficient to have risk control if we can upper bound the object

$$E_{P_{X^{N+1}}}[\frac{-\log E_{P_0}[e^{-\frac{\beta}{2}\sum_{i=1}^{N+1}(g(X_i) - f_{w^{\text{disc}}}(x_i))^2}]}{\beta(N+1)}]. \tag{6.8}$$

For the discrete prior with $|g| \leq b$, we can upper bound this object as

$$\frac{MK\log(2d+1)}{\beta(N+1)} + \frac{a_0^2 V^2}{2K} + \frac{(V(a_0 V + b)a_2 + V^2 a_1^2)}{2M}. \tag{6.9}$$

To get risk control for the continuous prior, the object we must understand is the same expression if we use w^{cont} in place of w^{disc},

$$E_{P_{X^{N+1}}}[\frac{-\log E_{P_0}[e^{-\frac{\beta}{2}\sum_{i=1}^{N+1}(g(X_i) - f_{w^{\text{cont}}}(x_i))^2}]}{\beta(N+1)}]. \tag{6.10}$$

We do note the following result relating the continuous object to the discrete object,

Lemma 6.1. *Using the joint distribution defined above, the cumulant generating function using the continuous vector is less than twice the cumulant generating function using the discrete vector plus an additional term,*

$$-\log E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{cont}}\|_{N+1}^2}] \tag{6.11}$$

$$\leq\ 2\left(-\log E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{disc}}\|_{N+1}^2}]\right)+\log E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{disc}}\|_{N+1}^2+\frac{\beta}{2}\|g-f_{w^{cont}}\|_{N+1}^2}]. \tag{6.12}$$

Proof. We show that (6.11) minus (6.12) is less than 0. Collecting all log terms under one expression, (6.11) minus (6.12) is written as

$$-\log\Big(\frac{E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|_{N+1}^2}]E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2+\frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|_{N+1}^2}]}{\Big(E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}]\Big)^2}\Big). \tag{6.13}$$

Note the square in the denominator is due to the factor of 2 in (6.12). Distribute one of these factors in the denominator to each expectation in the numerator and separate into two log expressions,

$$-\log E_{P_0}[\frac{e^{-\frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|_{N+1}^2}}{E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}]}]-\log E_{P_0}[\frac{e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2+\frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|_{N+1}^2}}{E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}]}]. \tag{6.14}$$

We wish to consider the expectation in the denominators as the normalizing constant of a density. In the first expression, add and subtract $\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2$ in the exponent. Then treat each term as an expectation using a properly normalized density,

$$-\log\int\frac{e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}}{E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}]}E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|_{N+1}^2+\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}|w^{\text{disc}}]P_0(dw^{\text{disc}}) \tag{6.15}$$

$$-\log\int\frac{e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}}{E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}]}E_{P_0}[e^{\frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|_{N+1}^2}|w^{\text{disc}}]P_0(dw^{\text{disc}}). \tag{6.16}$$

Apply Jensen's inequality on each term twice to bring the negative log into the inner most expectation. This will bring the terms in the exponent down with a negative sign, so we have upper bound

$$\int \frac{e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}}{E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}]} E_{P_0}[\frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|_{N+1}^2 - \frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2 | w^{\text{disc}}] P_0(dw^{\text{disc}}) \tag{6.17}$$

$$+\int \frac{e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}}{E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}]} E_{P_0}[-\frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|_{N+1}^2 | w^{\text{disc}}] P_0(dw^{\text{disc}}). \tag{6.18}$$

These expectations are then with respect to the same distribution, so we can collect into a common integral. The norms with $f_{w^{\text{cont}}}$ are of opposite sign and cancel, while the norm with $f_{w^{\text{disc}}}$ remains with a negative sign. Thus we have,

$$-\frac{\beta}{2}\int \frac{e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}}{E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2}]} \|g-f_{w^{\text{disc}}}\|_{N+1}^2 P_0(dw^{\text{disc}}) \leq 0. \tag{6.19}$$

Since the loss function is always non-negative, this expectation is always positive, and the negative in front makes it less than or equal to 0. □

Thus the object to understand is this discrete-continuous error term, and its expectation taken over the data distribution

$$E_{P_{X^{N+1}}}[\frac{\log E_{P_0}[E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2 + \frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|_{N+1}^2} | w^{\text{cont}}]]}{\beta(N+1)}]. \tag{6.20}$$

This is the "additional" square risk our continuous prior will pay on top of the risk control we already proved for the discrete prior. Now, one can show for the objects in the exponent,

$$E_{P_0}[-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2 + \frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|_{N+1}^2 | w^{\text{cont}}] = O(\frac{\beta(N+1)}{M}) \tag{6.21}$$

$$\text{Var}[-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|_{N+1}^2 + \frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|_{N+1}^2 | w^{\text{cont}}] = O((\frac{\beta(N+1)}{M})^2) \tag{6.22}$$

Thus one could conjecture under a Bernstein inequality,

$$E_{P_{X^{N+1}}}[\frac{\log E_{P_0}[E_{P_0}[e^{-\frac{\beta}{2}\|g-f_{w^{\text{disc}}}\|^2_{N+1}+\frac{\beta}{2}\|g-f_{w^{\text{cont}}}\|^2_{N+1}}|w^{\text{cont}}]]}{\beta(N+1)}] = O(\frac{1}{M}+\frac{\beta(N+1)}{M^2}). \tag{6.23}$$

Then our risk control for the continuous prior would be of the order (ignoring constants)

$$\frac{MK\log(d)}{\beta(N+1)}+\frac{1}{M}+\frac{1}{K}+\beta+\frac{\beta(N+1)}{M^2}. \tag{6.24}$$

Setting

$$M = (\frac{N+1}{\log(d)})^{\frac{1}{2}} \tag{6.25}$$

$$\beta = (\frac{\log(d)}{(N+1)})^{\frac{1}{6}} \tag{6.26}$$

$$K = (\frac{N+1}{\log(d)})^{\frac{1}{6}} \tag{6.27}$$

Gives risk control of the order $O(\frac{[\log(d)]^{\frac{7}{6}}}{N^{\frac{1}{6}}})$. Thus we could recover risk control for the continuous prior, of a lower order than $1/4$.

However, Bernstein's inequality is about sub-exponential random variables, and with $\beta N > M$ as we need for discrete risk control, the scaling of our random variable puts us outside the range where these results apply. The best bounds were are able to prove are $O(1)$, but we need a bound of the form $O(\frac{1}{M^p}+\frac{\beta(N+1)}{M^q})$ for any $p>0, q>0$ to get some form of risk control for the continuous prior.

It remains an open problem to try and connect the continuous risk control to the discrete risk control, and future work hopes to bridge this gap.

6.2 Optimization and Infinite Width Limits

While in this work we focus on a Bayesian method to train neural networks with MCMC as our primary algorithm, there has been much research to understand theoretically the optimization of neural networks via gradient based methods [15, 58, 24, 33, 45]. Optimization has proven empirically successful for highly over-parameterized networks with small initial scaling of parameters. Such networks quickly converge under gradient descent to interpolating (0 training loss) solutions, so it is important to understand if such networks are overfit and generalize poorly, and exactly what limiting object are these training methods converging to.

For classification problems with well separated classes and with rather large (potentially overfit) single-hidden-layer networks, [15] shows that gradient descent with large step size converges quickly to an interpolating solution on the training data. [58] demonstrates this solution still has good generalization risk via a form of "benign overfitting", however this comes at a cost of being susceptible to adversarial perturbations in specific directions that flip model outputs [24].

For very wide networks $K > N$, [45] shows neural networks satisfy a Polyak-Łojasiewicz (PL) condition proving convergence of stochastic gradient descent to a global minimizer of the loss function. This is an interesting phenomenon, however without suitable parameter controls (such as ℓ_1 controls), it is not clear if generalization properties will be favorable in this setting for general function learning.

Another approach to understanding optimization in very large neural networks is to compare them to certain infinite width limits. Two main objects arise in the analysis depending on the scaling of the problem: the Neural Tangent Kernel (NTK) [33] and mean field limit [50].

Consider a single-hidden-layer neural network where we do not specify the scaling of

the exterior weights

$$f(x) = \alpha_K \sum_{k=1}^{K} s_k \psi(x \cdot w_k). \tag{6.28}$$

For a constant V, we can consider the choice of scaling $\alpha_K = \frac{V}{\sqrt{K}}$ or $\alpha_K = \frac{V}{K}$ and consider the limiting behavior of the network as we take $K \to \infty$.

For $\alpha_K = \frac{V}{\sqrt{K}}$, in over parameterized problems it has been shown that for small norm initial weights, the network behaves very similarly to a linear first order Taylor expansion around its initialization point. With small random initialization and control on the number of gradient steps, gradient methods quickly converge to a near interpolating solution [1, 21, 59]. Networks trained in this regime approach regression with a fixed kernel determined by the covariance of the gradient of the network at the random initialization. This is tantamount to a linear regression onto a prefixed set of basis functions (the large eigenvalue eigenvectors of the kernel).

A related perspective is in [18] which identifies this regime of an approximate linear model of small weights as "lazy" training, that does not allow the internal weights to have much freedom to adjust the basis. Models trained in this regime can have poor generalization, compared to models trained in the more difficult non-lazy regime. In contrast, we seek a procedure that performs as well as if the set of directions w_k are adapted to the observed data.

With a scaling of $\alpha_K = \frac{V}{K}$, which is the scaling of networks we use in our analysis, the infinite width behavior if the network is quite different than the NTK regime. When a network with scaling $\frac{V}{K}$ is trained with stochastic gradient descent, as K grows large the network approaches a different limit called the mean field limit [50]. That is, there is a density $\rho(w, a)$ on interior and exterior weights such that the network approaches the

expected of a single neuron under this density,

$$\bar{f}(x) = \int a\psi(x \cdot w)\rho(dw, da). \tag{6.29}$$

Thus, the NTK regime converges to a linearly parameterized model fit by ridge regression, while the mean field regime converges to a fully non-linear model.

6.3 Implications for Proven Hard Training Problems

Our original conception of the problem did not consider the a_0, a_1, a_2, V values as values growing in N or d, but rather as fixed constants. We can consider, if these values are N or d dependent, what scaling we can allow in these values and still maintain our results. Additionally, we force our neurons to be ℓ_1 controlled with norm 1. Can we allow them to have larger ℓ_1 control, for example $\|w_k\|_1 \leq d$, so that $\{-1, 1\}^d \subset S_d^d$?

We note that we were not able to connect the log-concave coupling result we proved using a continuous uniform prior to risk control we proved for the discrete uniform prior. Therefore, we do not technically have guaranteed risk control and a log-concave coupling at the same time. However, in this section we assume we are able to connect the two results and consider what the implications would be for sampling and risk control when the parameters of the problems grow in N and d.

6.3.1 Using a Larger Network than the Target

Assume we allow our neuron weights $\|w\|_k \leq d$, and consider our target function as a linear combination of K signed neurons,

$$g(x) = \sum_{k=1}^{K} \frac{1}{K} s_k \psi(w_k \cdot x_i). \tag{6.30}$$

Define $\tilde{w}_k = \frac{w_k}{d}$ as a neuron weight with $\|\tilde{w}_k\|_1 \leq 1$ as we consider in our problem. Then we equivalently consider $g(x)$ as a linear combination of K signed neurons using ℓ_1 controlled neurons, with a factor of d inside the neuron activation function

$$g(x) = \sum_{k=1}^{K} \frac{1}{K} s_k \psi(d\tilde{w}_k \cdot x). \tag{6.31}$$

We can then consider a new activation function which incorporates the d into its definition, $\tilde{\psi}(z) = \psi(dz)$, and our target can be expressed as

$$g(x) = \sum_{k=1}^{K} \frac{1}{K} s_k \tilde{\psi}(\tilde{w}_k \cdot x). \tag{6.32}$$

Now, it may seem that for this neural network, $a_1 = \max_z |\tilde{\psi}'(z)| = d, a_2 = \max_z |\tilde{\psi}''(z)| = d^2$ scale with d which would be problematic for our construction. However, this is a matter of perspective. The only "intrinsic" property of this network is its scaling V (set here to be 1) and activation function $\tilde{\psi}$ (and its corresponding derivative bounds). Nothing is stopping us from considering a much wider network with $\tilde{K} > K$, with larger internal dimension $\tilde{d} > d$ which uses the same scaling V and the same activation function $\tilde{\psi}$. For instance, we can repeat neurons as many times as we like, and maintaining a convex combination this is an equivalent representation of g. Furthermore, we may increase the internal dimension $\tilde{d}$ while maintaining our neuron weights as having ℓ_1 norm less than or equal to 1 by simply repeating the coordinate values and normalizing to have the same ℓ_1 norm, eg.

$$x \cdot w_k = (1, 0.75) \cdot (\frac{1}{4}, -\frac{3}{4}) \tag{6.33}$$

is the same as the repeated version

$$x \cdot w_k = (1, 0.75, 1, 0.75) \cdot \frac{(\frac{1}{4}, -\frac{3}{4}, \frac{1}{4}, -\frac{3}{4})}{2}. \tag{6.34}$$

Thus, let K and d be properties of our "base" target network. Define $a_1 = d, a_2 = d^2$. Let $a_0 = 1$. Then we are free to train a neural network of arbitrary width $\tilde{K} > K$ and arbitrary internal dimension $\tilde{d} > d$ with scaling $V = 1$ using neuron activation function $\tilde{\psi}$.

How large then should $\tilde{d}, \tilde{K}$ be? Also, how many data points N do we need in our training? Consider if for some $\kappa > 1$ and some power p we have a large amount of training data such that

$$N = \kappa d^p, \qquad p \geq 4, \tag{6.35}$$

We will see $p \geq 4$ is the threshold we would need to achieve both a log-concave coupling and risk control. Note d here is the dimension of our "base" target, $\tilde{d}$ is the dimension of the larger network we will actually train. Define our internal weight dimension to be for some power q,

$$\tilde{d} = \kappa^2 d^q, \qquad q > \frac{3}{2}p + 5. \tag{6.36}$$

Let the width of our network $\tilde{K}$ be

$$\tilde{K} = \frac{1}{\sqrt{d}}\Big(\frac{N}{\log(\tilde{d})}\Big)^{\frac{1}{4}} = \frac{1}{\sqrt{d}}\Big(\frac{\kappa d^p}{q\log(d) + 2\log(\kappa)}\Big)^{\frac{1}{4}} = d^{\frac{p}{4}-\frac{1}{2}}\Big(\frac{\kappa}{q\log(d) + 2\log(\kappa)}\Big)^{\frac{1}{4}}. \tag{6.37}$$

Then define the gain and grid size

$$M = d^{\frac{3}{2}}\tilde{K} \qquad \beta = \frac{1}{\tilde{K}}. \tag{6.38}$$

This fully defines the relevant parameters of our training algorithms $\beta, N, M, \tilde{K}, \tilde{d}$. We can then see that,

$$\beta N = \kappa d^p \frac{1}{d^{\frac{p}{4}-\frac{1}{2}}} \Big(\frac{q\log(d) + 2\log(\kappa)}{\kappa} \Big)^{\frac{1}{4}} = (q\log(d) + 2\log(\kappa))^{\frac{1}{4}} \kappa^{\frac{3}{4}} d^{\frac{3}{4}p+\frac{1}{2}}. \tag{6.39}$$

Consider then the implications of this choice of parameters for our log-concave coupling results and our risk control results. First, we check if these parameters satisfy a log-concave coupling. Via Theorem 2.3, we must satisfy two conditions, the first being

$$\tilde{K}[\log(2\tilde{K}\tilde{d}/300)] \leq \beta N. \tag{6.40}$$

Noting that $\tilde{K} = d^{\frac{p}{4}-\frac{1}{2}} O(\kappa^{\frac{1}{4}})$ and $\beta N = d^{\frac{3}{4}p+\frac{1}{2}} O(\kappa^{\frac{3}{4}})$ this holds for κ not too large. The second condition for a log-concave coupling is

$$\tilde{d} \geq 4\sqrt{\frac{3}{2e}} d^2 (C_N)^{\frac{3}{2}} [2d + 4\sqrt{3} 2d^2 + \Big(1 + \frac{1}{\sqrt{\pi}}\Big) d \sqrt{2\sqrt{\frac{3}{2}}} (C_N)^{\frac{1}{2}}] (\beta N)^2 \tag{6.41}$$

$$\tilde{d} \geq \Big(4\sqrt{\frac{3}{2e}} (C_N)^{\frac{3}{2}} [2\frac{1}{d} + 4\sqrt{\frac{3}{2}} + \frac{1}{d}\Big(1 + \frac{1}{\sqrt{\pi}}\Big) \sqrt{2\sqrt{\frac{3}{2}}} (C_N)^{\frac{1}{2}}]\Big) (q\log(d) + 2\log(\kappa))^{\frac{1}{2}} \kappa^{\frac{3}{2}} d^{\frac{3}{2}p+5}. \tag{6.42}$$

Then our $\tilde{d} \geq \kappa^2 d^q$ this holds for moderately small κ, when $q > \frac{3}{2}p + 5$.

Our overall risk control with these parameters (using Theorem 3.3 and noting we have set $M = d^{\frac{3}{2}}\tilde{K}, \beta = 1/\tilde{K}, a_0 = V = b = 1$) is of the form.

$$d^{\frac{3}{2}} \frac{\tilde{K}^3 \log(2\tilde{d}+1)}{N} + \frac{1}{2\tilde{K}} + \frac{3}{2}\frac{d^{\frac{1}{2}}}{\tilde{K}} + 2\frac{1}{\tilde{K}}(\sigma+1)^2 \tag{6.43}$$

$$= \frac{\log(2\tilde{d}+1)}{\log(\tilde{d})^{\frac{3}{4}}} \frac{1}{\kappa^{\frac{1}{4}} d^{\frac{p}{4}}} + \Big(\frac{(q\log(d) + 2\log(\kappa))}{\kappa} \Big)^{\frac{1}{4}} \frac{(\frac{1}{2} + 2(\sigma+1)^2 + \frac{3}{2}d^{\frac{1}{2}})}{d^{\frac{p}{4}-\frac{1}{2}}}. \tag{6.44}$$

If $p > 4$, then we have bound

$$\Big(\frac{\log(2\tilde{d}+1)}{\log(\tilde{d})}\Big)^{\frac{3}{4}}\Big(\frac{\log(2\kappa^2 d^q+1)}{\kappa}\Big)^{\frac{1}{4}} + \Big(\frac{(q\log(d)+2\log(\kappa))}{\kappa}\Big)^{\frac{1}{4}}(\frac{1}{2}+2(\sigma+1)^2+\frac{3}{2}). \tag{6.45}$$

Then for $\kappa > \log(d)$, the fractions involving κ will be less than 1, and increasing κ beyond this point can make the risk arbitrarily small.

Note this analysis is for the derivative parameters a_1, a_2 being d dependent, similar analysis can be done if V is d dependent. The conclusion being that with enough data N, a larger network with larger internal weight dimension with ℓ_1 norm bounded by 1 can be fit to the original target network using our method.

6.3.2 Application To Intersection of Half Spaces

Thus, if we are able handle weights with ℓ_1 norm equal to their dimension d and claim we can fit them arbitrarily well in polynomial time, this would seem on the surface to contradict known hard problems in cryptography that are strongly conjectured to not be solvable in polynomial time [20, 53, 44]. However, on closer analysis, it can be seen that our method and results do not conflict with these hard problems and would not produce an arbitrarily good solution in poly time for those problems. The key difference is we cannot use ReLU functions in our method since we need second derivatives for our activation function, and our notion of risk is in an L_2 sense, not a pointwise risk as we will now discuss.

We now briefly define the intersection of half spaces problem. Say for some dimension d, we have $d/2$ weight vectors $w_1, \cdots, w_{d/2}$ each in $\{-1, 1\}^d$. We then define a function f on $\{-1, 1\}^d :\to 0, 1$ where for a vector x of plus/minus one values, $f(x) = 1$ if $\langle x, w_k\rangle > 0$ for all $k \in \{1, \cdots, d/2\}$. Else, $f(x) = 0$. Thus, f only outputs 0 or 1 values.

With a distribution P_X for the x values, given a function $h : \{-1, 1\}^d \to \{0, 1\}$, its

error with f is the probability of misclassification

$$\text{Error}_{P_X}(h) = P_X(f(X) \neq h(X)) = E_{P_X}[1\{f(X) \neq h(X)\}]. \tag{6.46}$$

Note this is a quite different notion of error than the L_2 error we work with

$$E_{P_X}[(f(X) - h(X))^2]. \tag{6.47}$$

Given access to as many samples $(x_i, f(x_i))_{i=1}^N$ as we like, and given any $\epsilon > 0$, and for any d, can we train an algorithm in poly$(n, \frac{1}{\epsilon})$ that produces a function h with $\text{Error}_{P_X}(h) \leq \epsilon$? It is strongly conjectured the answer is no, under the so called Strong Random Constraint Satisfaction Problem assumption, or SRCSP. If SRCSP was shown to be not true, then the entire field of cryptography would be thrown into disarray, so it may be taken as a very strong conjecture.

Note that $f(x)$ can be realized as a thresholded ReLU network in the following way. Let $\tilde{f}$ be the neural network

$$\tilde{f} = \sum_{k=1}^{\frac{d}{2}} \frac{2}{d}[(x \cdot w_k + 1)_+ - (x \cdot w_k)_+]. \tag{6.48}$$

Then $f(x) = 1\{\tilde{f}(x) = 1\}$. $\tilde{f}$ seems like the kind of network we should be able to train under our model, however, we cannot allow ReLU activation functions. Instead, consider that this difference in ReLU's can be approximated by a tanh activation function, see Figure 6.1.

$$(z+1)_+ - (z)_+ \approx \frac{\tanh(4(z+0.5)) + 1}{2}. \tag{6.49}$$

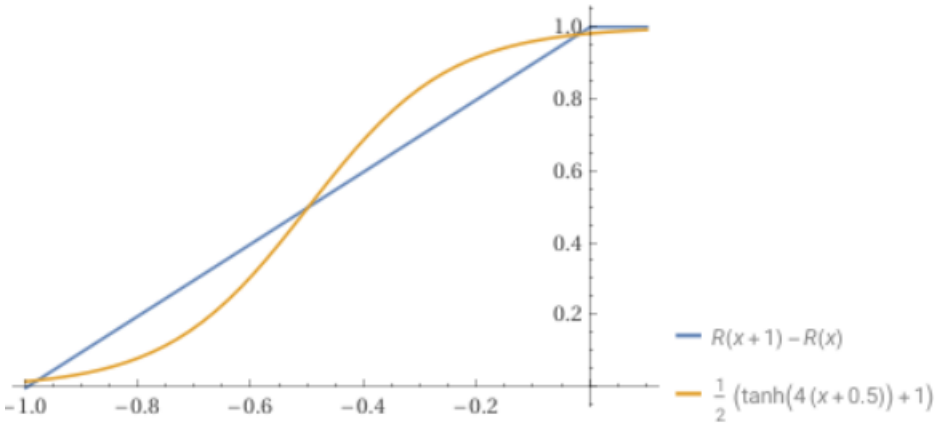

Figure 6.1: Comparison of difference of ReLU's and tanh approximation

Thus, we can train a network of $\tanh$ activation functions with our methodology. By the previous discussion on dealing with internal weights with $\|w_k\|_1 = d$, we can handle this by training a much larger network with larger internal weight dimension.

Thus, consider the set of functions

$$\mathcal{H} = \{h : [-1,1]^d \to [0,1], h(x) = \frac{\tanh(4(x \cdot w + 0.5))}{2}, \|w\|_1 \leq d\}. \tag{6.50}$$

Let $\bar{\mathcal{H}}$ be its closure under $L_2(P_X)$. Then, define $\tilde{g}$ to be the $L_2(P(X))$ projection of f into $\bar{\mathcal{H}}$,

$$\tilde{g} = \text{argmin}_{g \in \bar{\mathcal{H}}} E_{P_X}[(f(X) - \tilde{g}(X))^2]. \tag{6.51}$$

We can train a network of tanh's with $L_2(P_X)$ approximation error arbitrarily close to $\tilde{g}$'s approximation error. However, $\tilde{g}$ has some intrinsic L_2 error that is non-zero,

$$E_{P_X}[(f(x) - \tilde{g}(X))^2] = \epsilon^* > 0, \tag{6.52}$$

and we can never train a network with error less than this. Furthermore, $\tilde{g}$ is not a 0-1

output function, but outputs real values. We must then threshold $\tilde{g}$ to produce

$$g(x) = 1\{\tilde{g}(x) > \tau\}. \tag{6.53}$$

For some threshold τ. Our probability of misclassification is then

$$\text{Error}_{P_X}(g) = E_{P_X}[1\{f(X) \neq g(X)\}]. \tag{6.54}$$

We can show

$$E[(1 - g(X))^2 | f(X) = 1)] \leq \frac{\epsilon^*}{P(f(X) = 1)} \tag{6.55}$$

$$E[(g(X))^2 | f(X) = 0)] \leq \frac{\epsilon^*}{P(f(X) = 0)}. \tag{6.56}$$

Then by Markov inequality for any $\kappa > 0$, we have

$$P((1 - g(X))^2 \geq \kappa | f(X) = 1) \leq \frac{\epsilon^*}{\kappa P(f(X) = 1)} \tag{6.57}$$

$$P((g(X))^2 \geq \kappa | f(X) = 0) \leq \frac{\epsilon^*}{\kappa P(f(X) = 0)}. \tag{6.58}$$

Set our threshold as $\tau = 1 - \kappa$, then we have probability of error

$$\text{Error}_{P_X}(g) = P(f(X) = 1)P(g(X) < 1 - \kappa | f(X) = 1) \tag{6.59}$$

$$+ P(f(X) = 0)P(g(X) > 1 - \kappa | f(X) = 0) \tag{6.60}$$

$$= P(f(X) = 1)P((1 - g(X))^2 > \kappa^2 | f(X) = 1) \tag{6.61}$$

$$+ P(f(X) = 0)P((g(X))^2 > (1 - \kappa)^2 | f(X) = 0) \tag{6.62}$$

$$\leq \frac{\epsilon^*}{\kappa^2} + \frac{\epsilon^*}{(1 - \kappa)^2}. \tag{6.63}$$

Set $\kappa = \frac{1}{2}$ and we have

$$\text{Error}_{P_X}(g) \leq 8\epsilon^*. \tag{6.64}$$

So up to an error probability $\epsilon = 8\epsilon^*$, our method should be able to train a neural network in polynomial time. However, for errors below this threshold we cannot train a better network, so we cannot provide a solution in poly$(d, \frac{1}{\epsilon})$ for arbitrarily small ϵ.

Bibliography

[1] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019. 165

[2] Dominique Bakry and Michel Emery. Diffusions hypercontractives. *Seminaire de probabilites de Strasbourg*, 19:177–206, 1985. 39

[3] Dominique Bakry, Ivan Gentil, and Michel Ledoux. *Analysis and Geometry of Markov Diffusion Operators*, volume 103. Springer, 2014. 39

[4] Andrew Barron, Lucien Birgé, and Pascal Massart. Risk bounds for model selection via penalization. *Probability theory and related fields*, 113:301–413, 1999. 7

[5] Andrew R Barron. *The exponential convergence of posterior probabilities with implications for Bayes estimators of density functions*. Department of Statistics, University of Illinois Champaign, IL, 1988. 90

[6] Andrew R Barron. Neural net approximation. In *Proc. 7th Yale Workshop on Adaptive and Learning Systems*, volume 1, pages 69–72, 1992. 7

[7] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993. 6, 7, 130

[8] Andrew R Barron. Information-theoretic characterization of Bayes performance and the choice of priors in parametric and nonparametric problems. *In Proc. Valencia Conference, Bayesian Statistics*, 6:22–52, 1998. 67, 90

[9] Andrew R. Barron, Albert Cohen, Wolfgang Dahmen, and Ronald A. DeVore. Approximation and learning by greedy algorithms. *The Annals of Statistics*, 36(1):64 – 94, 2008. 108, 130

[10] Andrew R Barron and Jason M Klusowski. Approximation and estimation for high-dimensional deep learning networks. *arXiv:1809.03090*, 2018. 8

[11] Andrew R Barron and Jason M Klusowski. Complexity, statistical risk, and metric entropy of deep nets using total path variation. *arXiv:1902.00800*, 2019. 8, 108

[12] Andrew R Barron and Curtis McDonald. Log concave coupling for sampling from neural net posterior distributions. In *Proc. IMS-NUS Singapore Workshop on Statistical Machine Learning for High Dimensional Data*, 2024. 64

[13] Roland Bauerschmidt and Thierry Bodineau. A very simple proof of the LSI for high temperature spin systems. *Journal of Functional Analysis*, 276(8):2582–2588, 2019. 39

[14] Sergey G Bobkov and Michel Ledoux. From Brunn-Minkowski to Brascamp-Lieb and to logarithmic Sobolev inequalities. *Geometric and Functional Analysis*, 10:1028–1052, 2000. 32, 49

[15] Yuhang Cai, Jingfeng Wu, Song Mei, Michael Lindsey, and Peter Bartlett. Large stepsize gradient descent for non-homogeneous two-layer networks: Margin improvement and fast optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 9, 164

[16] Tom Charnock, Laurence Perreault-Levasseur, and François Lanusse. Bayesian neural networks. In *Artificial Intelligence for High Energy Physics*, pages 663–713. WORLD SCIENTIFIC, 2020. 9, 41

[17] Yongxin Chen, Sinho Chewi, Adil Salim, and Andre Wibisono. Improved analysis for a proximal algorithm for sampling. In *Conference on Learning Theory*, pages 2984–3014. PMLR, 2022. 40

[18] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019. 165

[19] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989. 6

[20] Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 441–448, 2014. 170

[21] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019. 165

[22] Raaz Dwivedi, Yuansi Chen, Martin J Wainwright, and Bin Yu. Log-concave sampling: Metropolis-Hastings algorithms are fast. *Journal of Machine Learning Research*, 20(183):1–42, 2019. 39, 44

[23] Marguerite Frank, Philip Wolfe, et al. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956. 131

[24] Spencer Frei, Gal Vardi, Peter Bartlett, and Nati Srebro. The double-edged sword of implicit bias: Generalization vs. robustness in relu networks. In *The Thirty-seventh Annual Conference on Neural Information Processing Systems*, 2023. 9, 164

[25] Víctor Gallego and David Ríos Insua. Current advances in neural networks. *Annual Review of Statistics and Its Application*, 9(1):197–222, 2022. 9, 41

[26] Sivakanth Gopi, Yin Tat Lee, Daogao Liu, Ruoqi Shen, and Kevin Tian. Algorithmic aspects of the log-Laplace transform and a non-Euclidean proximal sampler. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 2399–2439. PMLR, 2023. 40

[27] Robert D Gordon. Values of Mills' ratio of area to bounding ordinate and of the normal probability integral for large values of the argument. *The Annals of Mathematical Statistics*, 12(3):364–366, 1941. 46

[28] Boris Hanin and Alexander Zlokapa. Bayesian inference with deep weakly nonlinear networks, May 2024. arXiv:2405.16630. 9, 10, 41

[29] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. 6

[30] Jiri Hron, Roman Novak, Jeffrey Pennington, and Jascha Sohl-Dickstein. Wide Bayesian neural networks have a simple weight posterior: theory and accelerated sampling. In *International Conference on Machine Learning*, pages 8926–8945. PMLR, 2022. 9, 10, 41

[31] C Huang, AR Barron, and GHL Cheang. Risk of penalized least squares, greedy selection and l1 penalization for flexible function libraries. 108, 130

[32] Xunpeng Huang, Difan Zou, Yi-An Ma, Hanze Dong, and Tong Zhang. Faster sampling via stochastic gradient proximal sampler. *arXiv:2405.16734*, 2024. 40

[33] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018. 9, 164

[34] Lee K Jones. A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *The Annals of Statistics*, pages 608–613, 1992. 130

[35] Jason M Klusowski and Andrew R Barron. Risk bounds for high-dimensional ridge function combinations including neural networks. *arXiv preprint arXiv:1607.01434*, 2016. 108

[36] Jason M Klusowski and Andrew R Barron. Approximation by combinations of ReLU and squared ReLU ridge functions with ℓ_1 and ℓ_0 controls. *IEEE Transactions on Information Theory*, 64(12):7649–7656, 2018. 7, 8, 69

[37] Yunbum Kook, Yin-Tat Lee, Ruoqi Shen, and Santosh Vempala. Sampling with Riemannian Hamiltonian Monte Carlo in a constrained space. *Advances in Neural Information Processing Systems*, 35:31684–31696, 2022. 42

[38] Yunbum Kook and Santosh S Vempala. Gaussian cooling and Dikin walks: The interior-point method for logconcave sampling. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 3137–3240. PMLR, 2024. 39, 42

[39] Yunbum Kook and Santosh S Vempala. Sampling and integration of logconcave functions by algorithmic diffusion. *arXiv preprint arXiv:2411.13462*, 2024. 39, 42

[40] Wee Sun Lee, Peter L Bartlett, and Robert C Williamson. On efficient agnostic learning of linear combinations of basis functions. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, pages 369–376, 1995. 130

[41] Yin Tat Lee, Ruoqi Shen, and Kevin Tian. Structured logconcave sampling with a restricted Gaussian oracle. In *Conference on Learning Theory*, pages 2993–3050. PMLR, 2021. 40

[42] Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006. 19

[43] Erich Leo Lehmann, Joseph P Romano, and George Casella. *Testing Statistical Hypotheses*, volume 3. Springer, 1986. 50

[44] Shuchen Li, Ilias Zadik, and Manolis Zampetakis. On the hardness of learning one hidden layer neural networks. In *Proceedings of The 36th International Conference on Algorithmic Learning Theory*, volume 272 of *Proceedings of Machine Learning Research*, pages 700–701. PMLR, 24–27 Feb 2025. 170

[45] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022. 9, 164

[46] Qiang Liu, Jian Peng, Alexander Ihler, and John Fisher III. Estimating the partition function by discriminance sampling. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 514–522, 2015. 118

[47] Samuel Livingstone, Michael Betancourt, Simon Byrne, and Mark Girolami. On the geometric ergodicity of Hamiltonian Monte Carlo. *Bernoulli*, 25(4A):3109 – 3138, 2019. 39

[48] László Lovász and Santosh Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms*, 30(3):307–358, 2007. 39, 42

[49] Xi Luo. *Penalized Likelihoods: Fast Algorithms and Risk Bounds*. PhD thesis, Yale University, 2009. Accessed: 2025-03-06. 130

[50] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018. 164, 165

[51] Andrea Montanari and Yuchen Wu. Provably efficient posterior sampling for sparse linear regression via measure decomposition. *arXiv:2406.19550*, 2024. 39

[52] Radford M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, New York, NY, 1996. 9, 41

[53] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on learning theory*, pages 1376–1401. PMLR, 2015. 170

[54] Kai Wang Ng, Guo-Liang Tian, and Man-Lai Tang. *Dirichlet and Related Distributions: Theory, Methods and Applications*. John Wiley & Sons, 2011. 159

[55] Herbert Robbins. A remark on Stirling's formula. *The American Mathematical Monthly*, 62(1):26–29, 1955. 58

[56] Vishwak Srinivasan, Andre Wibisono, and Ashia Wilson. Fast sampling from constrained spaces using the Metropolis-adjusted mirror Langevin algorithm. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 4593–4635. PMLR, 2024. 42

[57] Michalis K Titsias and Omiros Papaspiliopoulos. Auxiliary gradient-based sampling algorithms. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(4):749–767, 2018. 40

[58] Alexander Tsigler and Peter L Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023. 9, 164

[59] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109:467–492, 2020. 165