CIS-LDRD Project 222359, Final Technical Report. Discretized Posterior Approximation in High Dimensions

Jed A. Duersch JADUERS@SANDIA.GOV
Thomas A. Catanach TACATAN@SANDIA.GOV
Sandia National Laboratories
Livermore, CA 94550, United States

SAND Report Number: SAND2021-11478

Abstract

Rigorous uncertainty quantification requires integration of predictions over many models according to their respective degrees of plausibility using Bayesian inference or a close approximation of it. When we seek to obtain predictions from abstract algorithms using small or otherwise limited datasets, it is particularly important for inference to begin with well-justified prior belief. Algorithmic Probability and Minimum Description Length are closely related theories that may be understood within the Bayesian paradigm to derive prior belief in algorithms from the amount of information contained their encodings. A primary challenge to apply this perspective to high-parameter architectures is how to formulate information-efficient model encodings and then, given encodings, how to find models with high posterior probability from an information-suppressing prior.

We examine an algorithmic framework to merge subspace-constrained inference over continuous parameters with a discretized prior that is only nonzero at a few discrete values per parameter. This allows us to constrain and control the amount of information that may be present in an individual model. We capture the approximate shape of the corresponding posterior using transformations on approximate first moments, which we then cast as a Laplace approximation, despite the fact that the actual posterior is neither continuous nor differentiable.

Keywords: machine learning, Bayesian inference, uncertainty quantification, complexity, Algorithmic Probability, Minimum Description Length, moment-transformation approximation

This work was funded by the U.S. Department of Energy and the Laboratory Directed Research and Development program at Sandia National Laboratories.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

1. Introduction

Our primary aim in this work is to understand how to efficiently obtain reliable uncertainty quantification in automatic learning algorithms with limited training datasets. Standard approaches rely on cross-validation to tune hyper parameters. Unfortunately, when our datasets are too small, holdout datasets become unreliable—albeit unbiased—measures of prediction quality due to the lack of adequate sample size. We should not place confidence in holdout estimators under conditions wherein the sample variance is both large and unknown. More poigniantly, our training experiments on limited data (Duersch and Catanach, 2021) show that even if we could improve estimator quality under these conditions, the typical training trajectory may never even encounter generalizable models.

Bayesian inference provides a rigorous theoretical framework to obtain well-justified uncertainty in predictions. In the context of automatic learning algorithms, where we seek to obtain predictions from otherwise arbitrary programs, suppressing unnecessary complexity in abstract predictive architectures requires constructing prior belief from a concrete notion of complexity suppression. The work of both Solomonoff (1960, 1964a,b, 2009) on Algorithmic Probability (AP) and Rissanen (1983, 1984) on Minimum Description Length (MDL) associate complexity with information, originally cast in terms of Shannon's entropy (Shannon, 1948) or as a change in belief with the Kullback–Leibler divergence (Kullback and Leibler, 1951; Kullback, 1997; MacKay, 2003; Ebrahimi et al., 2010; Duersch and Catanach, 2020). The key distinction between AP and MDL is that AP only restricts belief in plausible programs to those that are consistent with training data whereas MDL typically optimizes a single model, minimizing the information required to store the description combined with the information requires to store residual information in training data. Solomonoff's theory effectively computes the posterior-predictive distribution over programs that reproduce a binary sequence, whereas Rissanen's picture optimizes the Maximum À Posteriori (MAP) model. Expectation formulations of MDL McAllester (1999), however, can be understood as Variational Inference (VI) approximations of the posterior distribution for which the prior is taken from AP. Grünwald and Roos (2019) provide a recent overview of MDL.

Since it is not tractible to propose and evaluate arbitrary predictive programs, we can leverage the expressive potential of neural networks to describe diverse phenomena while incorporating complexity-suppressing prior belief by regarding a neural network as an abstract interpreter that is capable of accepting encoded parameter states to generate predictions. We typically regard a neural network as a continuous function with real parameters ,$\boldsymbol{\theta} \in \mathbb{R}^d$, floating-point implementations not withstanding. If, however, we can efficiently encode a diverse set of discrete parameter states, then we can apply an analogous form of Solomonoff's prior during training.

We consider doing this by first constructing a diverse set of representable values, $\mathcal{R} = \{\boldsymbol{r}_j \mid j \in [m]\}$, that each parameter may take. We can then assign each value an efficient encoding so that short codes are spread over a wide domain of parameter states. Let the encoding length of the representable value $\boldsymbol{r}_j$ be given by $\ell(\boldsymbol{r}_j)$, measured in natural information units (nats). Indexing network parameters by $i \in [d]$, our prior belief becomes

$$\mathbf{p}(\boldsymbol{\theta}) = \prod_{i=1}^{d} \sum_{j=1}^{m} e^{-\ell(\boldsymbol{r}_j)} \delta(\boldsymbol{\theta}_i - \boldsymbol{r}_j).$$

This produces an inference problem with a continuous likelihood function and discretized prior belief.

We require a training scheme to discover efficient parameter representations that simultaneously limits complexity while promoting agreement with our data. In principle, robust posterior integration would optimize learning efficiency and eliminate the need for cross-validation while also providing strong uncertainty quantification from small datasets.

1.1 Our Contributions

First, we propose a parameter discretization that allows representable parameter states to gradually increase in encoding length to obtain increases in specificity only as needed while still approximating a wide variety of outcomes. Building on Marzouk's Likelihood Informed Subspaces (LIS) (Marzouk and Najm, 2009; Cui et al., 2014), we then discuss a quadrature integration method to approximate the local structure of likelihood within a critical subspace near an expansion point. This method only requires computing loss gradients in the typical fashion with backward propagation.

We can then merge the continuous likelihood formulation with discretized prior belief to obtain a discretized posterior. To capture the shape of this posterior, we propose an approximation method similar to that of Laplace, but derived from mean transformations that can be efficiently computed over discrete distributions. We can formulate a minimal information perturbation to the posterior belief for which an approximate first moment becomes a fixed point of a Rao-Blackwellized moment update. We can then interpret the distribution perturbation as a weak gradient of a Laplace posterior approximation. This allows us to capture the local shape of the posterior distribution using a strategy similar to that of the likelihood approximation. Our posterior approximation method produces enough information to formulate a precision matrix as diagonal plus low rank. In contrast to L_2 regularization, which always compels the posterior toward the origin, we show how this framework compels parameters towards simple representations when the likelihood is broad, but allows increased specificity as the likelihood becomes sharp.

Finally, we propose a training algorithm to incorporate these techniques into a quasi-Newton posterior optimization scheme. This report includes early test results and comparisons with standard training using regularized stochastic gradient descent (SGD) and cross-validation. When we construct an ensemble of models with our training algorithm, the resulting averaged predictions out-perform any individual model in the set.

Section 2 provides background discussion, notation, and analysis of the key methods we develop. Section 3 combines these approximation techniques into key training algorithms. Section 4 provides a review of our numerical results, discusses key considerations our investigation uncovered, and summarizes our main results.

2. Localized Distribution Approximation and Integration

In the Bayesian paradigm, we regard loss as the negative log likelihood (NLL), which we will approximate as $J(\boldsymbol{\theta}) \approx -\log \mathbf{p}(\mathcal{D} \mid \boldsymbol{\theta})$. Likewise, we may regard a regularization term as the negative log prior (NLPr) so that regularized loss becomes the negative log posterior

(NLPo), up to a constant offset,

$$K(\boldsymbol{\theta}) \approx -\log \mathbf{p}(\mathcal{D} \mid \boldsymbol{\theta}) - \log \mathbf{p}(\boldsymbol{\theta}) = -\log \mathbf{p}(\boldsymbol{\theta} \mid \mathcal{D}) - \log \mathbf{p}(\mathcal{D}).$$

Table 1 provides a quick reference for additional notation.

The difficulty we encounter when performing Bayesian inference on a high-dimensional abstract machine learning model stems from unjustified prior belief. For example, L_2 regularization may be understood as deriving from a standard normal prior, $\mathbf{p}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid 0, \boldsymbol{I})$, but the resulting Maximum À Posterior (MAP) model may still exhibit memorization of training data. This is due to excessive creation of algorithmic information that results from inference with a loose prior in many dimensions. Although such priors are simple to describe, the model evidence $\mathbf{p}(\mathcal{D})$ becomes very small since normalization in high dimensions generates a shallow distribution spread over a large domain, with most models poorly fitting our data.

2.1 Bisected Gaussian Codes

By constraining parameter representations to values of increasing specificity, it may be possible to efficiently represent a wide variety of outcomes. We hypothesize that performing inference within the space of representations, thereby efficiently constraining our belief as needed, the model evidence for more simple restrictions may be much higher and dominate over the increase in information required to articulate them.

Early work on representation simplification goes back to Hinton and van Camp (1993). Hinton construct discrete encodings from equispaced bins on a standard normal distribution. By applying Shannons Source-Coding Theorem, we know that for a long enough sequence of realizations, an encoding may be constructed to become arbitrary close to the probabilities specified within each bin. This construction is well-approximated by L2 regularization.

Other discretization approachs include binary quantization (Courbariaux et al., 2015, 2016; Rastegari et al., 2016), ternary quantization (Mellempudi et al., 2017; Zhu et al., 2017), and fixed-point quantization (Lin et al., 2016). Fixed point, i.e. fixed precision as opposed to floating point, representations (Courbariaux et al., 2014; Lin et al., 2016) can be more generalizable if the network is trained to account for the representation, rather than rounding after the fact (Rastegari et al., 2016; Lin and Talathi, 2016). A similar approach employs stochastic rounding, (Gupta et al., 2015; Gysel et al., 2016; Lin and Talathi, 2016), which associates an affine probability distribution with intermediate values between representations. Given a value in the interval $x \in [a, b]$, round to a with probability $p(a) = \frac{x-a}{b-a}$. Otherwise, round to b.

Ternary Neural Networks with Fine-Grain Quantization (Mellempudi et al., 2017). A network is pretrained according to standard practices and then a threshold Δ is set such that $\hat{W}_i = sign(W_i)$ and we minimize the error of the ternary quantization $\{-\alpha, 0, \alpha\}$ so that $(\alpha, \Delta) = \text{argmin} \|W - \alpha\hat{W}\|_F$. This doesn't allow training to account for parsimonious snapping.

Blier and Ollivier (2018) investigates description length of deep learning models using prequential parameter encodings. Baldassarre et al. (2012); Han et al. (2015, 2016); Louizos et al. (2017) use structured sparsity penalty functions, sensitivity-based pruning, and other forms of sparsity enforcement to reduce trained model complexity.

Table 1: Notation

$\mathcal{D}$	Training dataset
d	Number of model parameters
$\boldsymbol{\theta}_i$	Model parameters, $i \in [d]$
m	Number of representable values
$\boldsymbol{r}_j$	Representable values, $\mathcal{R} = \{\boldsymbol{r}_j \mid j \in [m]\}$
$\ell(\boldsymbol{r}_j)$	Representation lengths in nats
$\mathbf{p}(\boldsymbol{\theta})$	Prior belief, $\mathbf{p}(\boldsymbol{\theta}) = \prod_i \sum_j e^{-\ell(\boldsymbol{r}_j)} \delta(\boldsymbol{\theta}_i - \boldsymbol{r}_j)$
$\mathbf{p}(\mathcal{D} \mid \boldsymbol{\theta})$	Likelihood
$\mathbf{p}(\boldsymbol{\theta} \mid \mathcal{D})$	Posterior
$\boldsymbol{\tau}$	Reparameterization length scales
$\boldsymbol{z}$	Rescaled parameters, $\boldsymbol{\theta} = \boldsymbol{z} \circledast \boldsymbol{\tau}$
$J(\boldsymbol{z})$	Local quadratic approximation of negative log likelihood
$\boldsymbol{z_0}$	Quadratic expansion point
J_0	Constant term, $J(\boldsymbol{z_0})$
$\boldsymbol{g_0}$	Gradiant term, $\nabla_{\boldsymbol{z}} J(\boldsymbol{z_0})$
$\boldsymbol{H}$	Hessian (precision matrix), $\boldsymbol{H}_{ij} = \frac{\partial^2}{\partial \boldsymbol{z}_i \partial \boldsymbol{z}_j} J(\boldsymbol{z})$
k	Tracked subspace rank
$\boldsymbol{U}$	Orthonormal column basis, $\boldsymbol{U} \in \mathbb{R}^{d \times k}$ so that $\boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}$
$\boldsymbol{C}$	Core matrix of low-rank precision, $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T$
$K(\boldsymbol{z})$	Local quadratic approximation of negative log posterior (unnormalized)
K_0	Constant term, $K(\boldsymbol{z_0})$
$\boldsymbol{c_0}$	Gradiant term, $\nabla_{\boldsymbol{z}} K(\boldsymbol{z_0})$
$\boldsymbol{\Lambda}$	Posterior precision matrix
$\boldsymbol{\Delta}$	Posterior precision core in low-rank approximation, $\boldsymbol{\Lambda} = \boldsymbol{I} + \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{U}^T$
$\boldsymbol{\delta}$	Diagonal of increase in posterior precision, $\boldsymbol{\Delta} = \mathrm{diag}(\boldsymbol{\delta})$
$\mathbf{m}(\boldsymbol{z} \mid \hat{\boldsymbol{z}})$	Moment-mapping distribution for posterior approximation
$\mathbf{q}(\mathcal{D} \mid \boldsymbol{z})$	Local likelihood approximation, $\mathbf{q}(\mathcal{D} \mid \boldsymbol{z}) = \exp(J(\boldsymbol{z}))$
$\mathbf{q}(\boldsymbol{z} \mid \mathcal{D})$	Local posterior approximation, $\mathbf{q}(\boldsymbol{z} \mid \mathcal{D}) \propto \exp(K(\boldsymbol{z}))$
$\mathbf{r}(\boldsymbol{z})$	Current view from which expectations and approximations are computed

The discretization we propose is derived by articulating intervals with a sequence of binary digits. We equate each 1-bit extension to the description with a corresponding elimination of an interval of belief containing half of the remaining probability that starts from a standard normal distribution. Thus, the KL divergence from each intermediate distribution to the update given by reading the next bit is exactly 1-bit also. We then associate each subinterval with a single value at the median. Figure 1 shows selected intervals and plots the corresponding representations for two parameter dimensions. change in the distribution

In order to efficiently approximate posterior-predictive integrals needed to obtain robust predictions, we require a means to both discover and articulate the shape of local domains

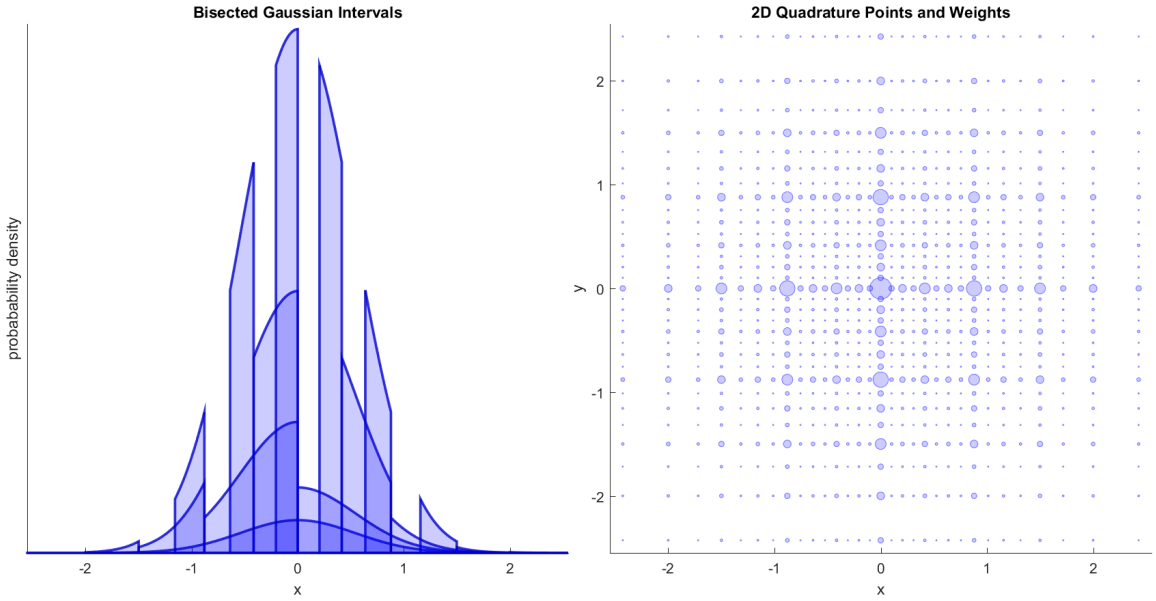

Figure 1: *Left:* Illustration of several probability distributions over subintervals that are obtained by sequentially bisecting a Gaussian distribution into equal-probability halves. The resulting distribution for each subinterval is then approximated as a single point located at the median, what would be the next bisection location. *Right:* The corresponding discrete prior distribution in two dimensions. The prior probability of each point is both the probability contained in the respective subinterval and, equivalently, the prior weight from representation length as a sequence of binary decisions, $\exp(-\ell(\boldsymbol{r}_j))$. This fractal-like decomposition of space distributes representable values throughout the domain with a coherent association between specificity and probability.

with high likelihood. LIS constrains exploration to the dimensions for which changes in parameters are most sensitive. Then, our next task will be to incorporate discretized prior belief into an approximate posterior over such domains.

Clearly, the actual posterior takes nonzero probability mass at every combination of representations and, thus, an exact description is intractible, having m^d components. Instead, we only need to capture the approximate shape of the posterior in domains that substantially contribute to the posterior-predictive integral. Writting our local likelihood approximation as $\mathbf{q}(\mathcal{D} \mid \boldsymbol{z})$, we will compute a corresponding posterior approximation

$$\mathbf{q}(\boldsymbol{z} \mid \mathcal{D}) \approx \frac{\mathbf{q}(\mathcal{D} \mid \boldsymbol{z})\mathbf{p}(\boldsymbol{z})}{\int d\boldsymbol{z}\, \mathbf{q}(\mathcal{D} \mid \boldsymbol{z})\mathbf{p}(\boldsymbol{z})}.$$

2.2 Bayesian inference and integration

Although we seek to obtain an efficient approximation of plausible models through inference, our primary objective remains to propagate justifiable model uncertainty through to

predictions on new data by approximating the integral of the posterious distribution over the corresponding predictions obtained from individual models. Thus, one of our primary concerns is how we understand efficient integration in approximate inference. We will also see that efficient integration directly feeds into variational inference optimization for local posterior approximations. Therefore, we spend some time examining the properties we desire in a belief approximation and corresponding integral.

In high-dimensional parameter spaces, some form of restricted exploration is unavoidable. Suppose we have d parameter dimensions, but we can only afford to evaluate the loss function $\mathcal{O}(k)$ times where k is the smallest hyperplane dimension that contains all of the evaluation locations. Although we may be able to obtain high-order integral approximations with this hyperplane, the fact remains that after integration, all orthogonal dimensions have only been probed at a single coordinate. Since we are considering high-dimensional architectures, $d \gg k$, we are forced to approximate posterior integrals in most dimensions with what may be regarded as a 1-point quadrature in those dimensions.

The best we can do in this setting is ensure that the evaluation point is a Gaussian quadrature. It is trivial to show that for a 1-point quadrature approximation to exactly integrate 1^{st}-degree polynomials, the evaluation point must be located at the mean of the distribution, and the corresponding weight must be 1.

$$\int dx\, \mathbf{p}(x) f(x) \approx f(\mu) \quad \text{where} \quad \mu = \int dx\, \mathbf{p}(x) x.$$

Within the statistics literature, this simplifying approximation and a variance reduction technique is known as Rao-Blackwellization (Rao, 1945; Blackwell, 1947; Casella and Robert, 1996). Fortunately, we are still free to choose the best k dimensions within which we can afford a more efficient integral approximation with a higher-order quadrature formula.

2.3 Likelihood Approximation

Since we are primarily concerned with small data regimes, where it is computationally feasible to iterate over our full training dataset many times, we can afford to compute likelihood approximations that extracts curvature from a single training example through several evaluations. Not only does this allow us to integrate critical dimensions more efficiently, it also provides a means to refine the basis for the dimensions that benefit most from high-order integration.

The key properties we require to approximate the local likelihood structure are the ability to articulate a center, critical dimensions, and corresponding length scales. The simplest analytic NLL construction with these properties is quadratic. In anticipation of reparameterization, which will be useful for our posterior approximation, we cast the likelihood in terms of a reparameterization, $\boldsymbol{\theta} = \boldsymbol{\tau} \circledast \boldsymbol{z}$, where the vector $\boldsymbol{\tau}$ specifies fixed length scales in each coordinate and $\boldsymbol{z}$ indicates the parameters to our quadratic approximations. Thus, our NLL is approximated as $-\log \mathbf{p}(\mathcal{D} \mid \boldsymbol{\theta} = \boldsymbol{\tau} \circledast \boldsymbol{z}) \approx J(\boldsymbol{z})$ where

$$J(\boldsymbol{z}) = J_0 + (\boldsymbol{z} - \boldsymbol{z_0})^T \boldsymbol{g_0} + \frac{1}{2}(\boldsymbol{z} - \boldsymbol{z_0})^T \boldsymbol{H} (\boldsymbol{z} - \boldsymbol{z_0}).$$

The corresponding likelihood approximation, $\mathbf{q}(\mathcal{D} \mid \boldsymbol{z}) = \exp(-J(\boldsymbol{z}))$, is a form of the Laplace approximation. Since the most sensitive dimensions have a small covariance, or

high precision, we only retain a low-rank precision approximation, $\boldsymbol{H} \approx \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T$, where $\boldsymbol{U} \in \mathbb{R}^{d\times k}$ has orthonormal columns, $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, and $\boldsymbol{C} \in \mathbb{R}^{k\times k}$. Daxberger et al. (2021) also discusses Laplace approximations for Bayesian deep learning.

Note that we track low-rank precision, rather than low-rank covariance, because the improper distribution that follows is simple to describe and evaluate. Low-rank covariance is not meaningful without restricting the distribution to the subspace containing positive eigenvalues.

It is also important for us to track curvature in drift dimensions that exhibit shallow local curvature, but at distances that are long enough that they should not be dropped. Otherwise, disregarding cumulative likelihood improvements in such dimensions would severly inhibit our ability to keep high likelihood domains within our quadratic approximation. To do this, the algorithm we propose will record important parameter states from the most recent epochs, say $\mathcal{Z} = \left\{\boldsymbol{z}^{(e)}, \boldsymbol{z}^{(e-1)}, \ldots, \boldsymbol{z}^{(e-t)}\right\}$. This will allow us to extend the basis $\boldsymbol{U}$ as needed with additional orthonormal columns $\boldsymbol{U}^+$ to ensure that each tracked state remains in the extended span, $\mathcal{Z} \subset \text{span}([\boldsymbol{U}\ \boldsymbol{U}^+])$.

We can approximate the precision using only likelihood gradients, $-\nabla_{\boldsymbol{z}} \log \mathbf{p}(\mathcal{D} \mid \boldsymbol{z}) \approx \nabla_{\boldsymbol{z}} J(\boldsymbol{z})$. Since we have $\nabla_{\boldsymbol{z}} J(\boldsymbol{z}) = \boldsymbol{g_0} + \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T(\boldsymbol{z} - \boldsymbol{z_0})$, integrating a local distribution centered at $\boldsymbol{z_0}$ easily recovers the corresponding gradient term at the expansion point $\boldsymbol{z_0}$. Further, if we compute the gradient at symmetric evaluation points within a hyperplane, so that $\boldsymbol{z}^{\pm} = \boldsymbol{z_0} \pm \boldsymbol{U}\boldsymbol{\varphi}$ with the vector $\boldsymbol{\varphi} \in \mathbb{R}^k$ specifying the displacement in the basis $\boldsymbol{U}$, we have

$$\boldsymbol{g}^{\pm} = \nabla_{\boldsymbol{z}} J(\boldsymbol{z}^{\pm}) = \boldsymbol{g_0} \pm \boldsymbol{U}\boldsymbol{C}\boldsymbol{\varphi}.$$

Averaging over pairs gives $\boldsymbol{g_0}$ and projecting the difference onto $\boldsymbol{U}$ allows us to evaluate matrix-vector products with the precision core matrix

$$\frac{1}{2}\boldsymbol{U}^T(\boldsymbol{g}^+ - \boldsymbol{g}^-) = \boldsymbol{C}\boldsymbol{\varphi}.$$

2.4 Sigma Points

By evaluating $\boldsymbol{\varphi}$ at each standard basis vector, we can easily reconstruct each column of $\boldsymbol{C}$ in our NLL approximation. This is equivalent to evaluating a function at the sigma points proposed in the Unscented Transform (Uhlmann, 1995), except we restrict our evaluations to a critical subspace. We only need to keep the symmetric component of $\boldsymbol{C}$ since it provides the only contribution to the symmetric innerproduct in the quadratic approximation.

This construction can be framed as an equal-weight quadrature to simplify our numerical approximation of the integral of a unit-normal distribution against another function

$$N[f] = \int d\boldsymbol{\varphi}\, \mathcal{N}(\boldsymbol{\varphi} \mid 0, \boldsymbol{I}) f(\boldsymbol{\varphi}) \quad \text{and}$$

$$Q[f] = \frac{1}{2k+1}\left(f(0) + \sum_{i=1}^{k} f(\boldsymbol{e}_i\rho) + f(-\boldsymbol{e}_i\rho)\right)$$

for some $\rho > 0$ so that $N[f] \approx Q[f]$. This construction is invariant under all signed permutations of the k coordinates, the actions of the Coxeter group B_k.

Because the normal distribution is even in each coordinate, the quadrature exactly integrates any basis function that separates into an odd function of at least one coordinate,

$$Q\big[f(\boldsymbol{\varphi}_j)g(\boldsymbol{\varphi}_1,\ldots,\boldsymbol{\varphi}_{j-1},\boldsymbol{\varphi}_{j+1},\ldots,\boldsymbol{\varphi}_k)\big] = N\big[f(\boldsymbol{\varphi}_j)g(\boldsymbol{\varphi}_1,\ldots,\boldsymbol{\varphi}_{j-1},\boldsymbol{\varphi}_{j+1},\ldots,\boldsymbol{\varphi}_k)\big] = 0,$$

for any $j \in [k]$ when $f(\cdot)$ is odd, thus including all first-order and third-order monomials. Clearly, the quadrature weights $(2k+1)^{-1}$ also correctly integrate the unit function, $Q[1] = N[1] = 1$. It only remains to solve the distance ρ to correctly integrate quadratic monomials. Since we have

$$N\big[\boldsymbol{\varphi}_j^2\big] = 1 \quad \text{and} \quad Q\big[\boldsymbol{\varphi}_j^2\big] = \frac{2}{2k+1}\rho^2 \quad \text{, it follows} \quad \rho = \sqrt{k+\frac{1}{2}}.$$

Because both $N[\cdot]$ and $Q[\cdot]$ are linear functionals, the fact that $Q[\cdot]$ exactly integrates a basis for third-degree polynomials in $\mathbb{R}^k$ implies that this equal-weight quadrature integrates all third-order polynomials exactly.

We can find an affine coordinate transformation from $\mathcal{N}(\boldsymbol{\varphi} \mid 0, \boldsymbol{I})$ to any Gaussian $\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{z_0}, \boldsymbol{\Gamma})$ where the eigenvalue decomposition of the covariance is $\boldsymbol{\Gamma} = \boldsymbol{U}\,\mathrm{diag}(\boldsymbol{\sigma})^2\boldsymbol{U}^T$ so that $\boldsymbol{z} = \boldsymbol{z_0} + \boldsymbol{U}\,[\boldsymbol{\sigma} \circledast \boldsymbol{\varphi}]$. If the basis vectors are $\boldsymbol{U} = [\boldsymbol{u}_1\,\boldsymbol{u}_2\,\cdots\,\boldsymbol{u}_k]$ then the modified quadrature for the critical subspace is

$$Q[f] = \frac{1}{2k+1}\left(f(\boldsymbol{z_0}) + \sum_{j=1}^{k} f(\boldsymbol{z_0} + \boldsymbol{u}_j\boldsymbol{\sigma}_j\rho) + f(\boldsymbol{z_0} - \boldsymbol{u}_j\boldsymbol{\sigma}_j\rho)\right).$$

Menegaz et al. (2015) provides a review of this and other variants of sigma point construction.

2.5 Moment-Mapping Posterior Approximation

The primary challenge remains to combine our local likelihood approximation with discretized prior belief in a way that allows us to efficiently describe and integrate a local posterior approximation. Since the actual posterior is discrete, and not differentiable, we can formulate the NLPo in terms of first moments as approximate Rao-Blackwellizations of the posterior. We find this approach attractive because, given $J(\boldsymbol{z})$, we can easily afford to compute the corresponding posterior within a single coordinate, provided all other coordinates are held fixed—or approximately Rao-Blackwellized. This technique draws on the mechanism employed by Gibbs sampling wherein, for a given parameter state, we form the posterior distribution in a single dimension conditioned on holding all other coordinates fixed. Gibbs sampling then proceeds by drawing updates in each dimension and iterating over all coordinates many times until it converges to a sample of the posterior.

The problem with Gibbs sampling is that high posterior perturbations become increasingly rare when they require coordination in high dimensions. Consequently, this sampling approach generates highly correlated Markov chains that take extremely long time scales to converge to the posterior. Instead, our approach allows us to identify correlations within a critical subspace by seeing how an approximate posterior mean, say $\hat{\boldsymbol{z}}$, would induce Rao-Blackwellizations, constructed independently in each coordinate, that then yield a new

posterior mean approximation, $\hat{\boldsymbol{z}}'$. This will allow us to form a Laplace approximation of the local posterior.

This analysis proceeds by evaluating our likelihood approximation, $J(\boldsymbol{z})$, for perturbations in a single coordiante, $\boldsymbol{z}_i$, conditioned on evaluating all other coordinates at $\hat{\boldsymbol{z}}$. Taking $\hat{\boldsymbol{z}}$ as the quadratic expansion point gives the gradient $\hat{\boldsymbol{g}} = \boldsymbol{g_0} + \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T(\hat{\boldsymbol{z}} - \boldsymbol{z_0})$, so that

$$-\log \mathbf{q}(\mathcal{D} \mid \boldsymbol{z}) = \text{const} + (\boldsymbol{z} - \hat{\boldsymbol{z}})^T \hat{\boldsymbol{g}} + \frac{1}{2}(\boldsymbol{z} - \hat{\boldsymbol{z}})^T \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T(\boldsymbol{z} - \hat{\boldsymbol{z}}).$$

This allows us to easily express perturbations in $\boldsymbol{z}_i$. Let $\hat{\boldsymbol{z}}_i^c$ represents the fixed complement to $\boldsymbol{z}_i$, i.e. evaluating all coordinates except i at $\hat{\boldsymbol{z}}$, so that

$$\begin{aligned} -\log \mathbf{q}(\mathcal{D} \mid \boldsymbol{z}_i, \hat{\boldsymbol{z}}_i^c) &= -\log \mathbf{q}(\mathcal{D} \mid \boldsymbol{z} = \hat{\boldsymbol{z}} + (\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i)\boldsymbol{e}_i) \\ &= \text{const} + (\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i)^T \boldsymbol{e}_i^T \hat{\boldsymbol{g}} + \frac{1}{2}(\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i)^2 \boldsymbol{e}_i^T \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T \boldsymbol{e}_i. \end{aligned}$$

To incorporate our prior belief in an approximate NLPo, we must evaluate this distribution at all possible representations and add the NLPr, or corresponding representation lengths. Recalling that we are using a reparameterization, $\boldsymbol{\theta} = \boldsymbol{\tau} \circledast \boldsymbol{z}$, the representable locations are $\boldsymbol{z}_i = \boldsymbol{r}_j / \boldsymbol{\tau}_i$ for $j \in m$. We can construct a representation perturbation matrix for all coordinates as $\boldsymbol{R}_{ij} = \boldsymbol{r}_j / \boldsymbol{\tau}_i - \hat{\boldsymbol{z}}_i$. This allows us to simultaneoulsy compute the conditional NLPo approximations in each coordinate as rows of the matrix

$$\boldsymbol{K}_{ij} = \ell(\boldsymbol{r}_j) + \boldsymbol{R}_{ij}\left(\hat{\boldsymbol{g}}_i + \frac{1}{2}\boldsymbol{R}_{ij}\sum_{\ell=1}^{k}(\boldsymbol{U}\boldsymbol{C})_{i\ell}\boldsymbol{U}_{i\ell}\right) = \text{const}_i - \log \mathbf{q}(\boldsymbol{z}_i = \frac{\boldsymbol{r}_j}{\boldsymbol{\tau}_i} \mid \mathcal{D}, \hat{\boldsymbol{z}}_i^c).$$

Each posterior slice is formed by taking the elementwise negative exponential and normalizing over rows, i.e. $\mathbf{q}(\boldsymbol{z}_i \mid \mathcal{D}, \hat{\boldsymbol{z}}_i^c) = \text{softmax}(-\boldsymbol{e}_i^T \boldsymbol{K})$. Taking the joint distribution defined by combining the conditionals independently allow us to obtain an updated posterior mean

$$\mathbf{m}(\boldsymbol{z} \mid \mathcal{D}, \hat{\boldsymbol{z}}) = \prod_{i=1}^{d} \mathbf{q}(\boldsymbol{z}_i \mid \mathcal{D}, \hat{\boldsymbol{z}}_i^c) \quad \text{so that} \quad \hat{\boldsymbol{z}}' = \mathbb{E}_{\mathbf{m}(\boldsymbol{z} \mid \mathcal{D}, \hat{\boldsymbol{z}})}\left[\boldsymbol{z}\right].$$

Rather than updating the mean, however, it will be more useful to solve a perturbed distribution for which $\hat{\boldsymbol{z}}$ becomes a true Rao-Blackwellization, so that $\hat{\boldsymbol{z}}' = \hat{\boldsymbol{z}}$ is a fixed point.

2.6 Weak Gradients from Minimum Information Perturbations

By constructing a minimum information perturbation to $\mathbf{m}(\boldsymbol{z} \mid \mathcal{D}, \hat{\boldsymbol{z}})$ for which a parameter state maps back to itself, $\hat{\boldsymbol{z}}' = \hat{\boldsymbol{z}}$, we obtain a weak gradient over posterior moments, despite the fact that true posterior gradients do not exist. This analysis is similar to the entropy-maximizing analysis due to Jaynes (1957). We seek a minimum information perturbation $\mathbf{m}^*(\boldsymbol{z})$ from $\mathbf{m}(\boldsymbol{z} \mid \mathcal{D}, \hat{\boldsymbol{z}})$, as measured using the the Kullback-Leibler divergence (Kullback and Leibler, 1951), that yields our desired first moment

$$\mathbf{m}^*(\boldsymbol{z}) = \underset{\mathbf{q}(\boldsymbol{z})}{\text{argmin}} \quad D[\,\mathbf{q}(\boldsymbol{z}) \,\|\, \mathbf{m}(\boldsymbol{z} \mid \mathcal{D}, \hat{\boldsymbol{z}})\,] \quad \text{such that} \quad \mathbb{E}_{\mathbf{q}(\boldsymbol{z})}[\boldsymbol{z}] = \hat{\boldsymbol{z}}.$$

Algorithm 1 Posterior Rao-Blackwellized Distributions

Require: The vector of allowable parameter representations is $\boldsymbol{r}$ and each value has encoding length $\ell(\boldsymbol{r}_j)$ natural units. The NLL is approximated as $J(\boldsymbol{z}) = J_0 + (\boldsymbol{z} - \boldsymbol{z_0})^T \boldsymbol{g_0} + 1/2(\boldsymbol{z} - \boldsymbol{z_0})^T \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T(\boldsymbol{z} - \boldsymbol{z_0})$. Reparameterization scales are $\boldsymbol{\tau}$ so that $\boldsymbol{\theta} = \boldsymbol{z} \circledast \boldsymbol{\tau}$. An optional perturbation gradient $\boldsymbol{c}$ is added to the gradient at the evaluation point $\hat{\boldsymbol{z}}$.

Ensure: $\mathbf{q}(\boldsymbol{\theta}_i = \boldsymbol{r}_j \mid \mathcal{D}, \hat{\boldsymbol{z}}_i^c)$ is the posterior distribution over allowable representations of $\boldsymbol{\theta}_i$ conditioned on evaluating the likelihood function in all other coordinates at $\hat{\boldsymbol{z}}_i^c$.

1: **function** PosteriorRB($\boldsymbol{r}, \ell(\boldsymbol{r}), \boldsymbol{\tau}, \boldsymbol{z_0}, J_0, \boldsymbol{g_0}, \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{c}, \hat{\boldsymbol{z}}$)

2: Construct the perturbation matrix, $\boldsymbol{R}_{ij} = {}^{\boldsymbol{r}_j}/_{\boldsymbol{\tau}_i} - \hat{\boldsymbol{z}}_i$.

3: Construct the NLL gradient at $\hat{\boldsymbol{z}}$ as $\hat{\boldsymbol{g}} = \boldsymbol{g_0} + \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T(\hat{\boldsymbol{z}} - \boldsymbol{z_0}) - \boldsymbol{c}$.

4: Construct the coordinatewise log posterior matrix,

$$\boldsymbol{K}_{ij} = \ell(\boldsymbol{r}_j) + \boldsymbol{R}_{ij}\left(\hat{\boldsymbol{g}}_i + \frac{1}{2}\boldsymbol{R}_{ij}\sum_{\ell=1}^{k}(\boldsymbol{U}\boldsymbol{C})_{i\ell}\boldsymbol{U}_{i\ell}\right)$$

5: Compute coordinatewise posterior distributions $\mathbf{q}(\boldsymbol{\theta}_i = \boldsymbol{r}_j \mid \mathcal{D}, \hat{\boldsymbol{z}}_i^c) = \text{softmax}(\boldsymbol{e}_i^T K)_j$.

6: **end function**

We can derive the form of $\mathbf{m}^*(\boldsymbol{z})$ using variational analysis. Let $\mathbf{q}(\boldsymbol{z}) = \mathbf{m}^*(\boldsymbol{z}) + \varepsilon\boldsymbol{\eta}(\boldsymbol{z})$, where $\mathbf{m}^*(\boldsymbol{z})$ is the normalized optimizer, ε is a differential element, and $\boldsymbol{\eta}(\boldsymbol{z})$ is normalization-preserving, $\int d\boldsymbol{z}\,\boldsymbol{\eta}(\boldsymbol{z}) = 0$, but an otherwise arbitrary perturbation. We construct the Lagrangian as

$$\omega[\mathbf{q}(\boldsymbol{z}), \boldsymbol{c}] = \int d\boldsymbol{z}\,\mathbf{q}(\boldsymbol{z})\left[\log\left(\frac{\mathbf{q}(\boldsymbol{z})}{\mathbf{m}(\boldsymbol{z} \mid \mathcal{D}, \hat{\boldsymbol{z}})}\right) - \boldsymbol{c}^T(\boldsymbol{z} - \hat{\boldsymbol{z}})\right]$$

and note that, at the optimizer, the variational principle must be satisfied for arbitrary directional derivatives with respect to $\boldsymbol{\eta}(\boldsymbol{z})$ and with respect to the vector of Lagrange coefficients $\boldsymbol{c}$.

$$\frac{\partial}{\partial\varepsilon}\omega[\mathbf{q}(\boldsymbol{z}), \boldsymbol{c}]\bigg|_{\varepsilon=0} = \int d\boldsymbol{z}\,\boldsymbol{\eta}(\boldsymbol{z})\left[\log\left(\frac{\mathbf{m}^*(\boldsymbol{z})}{\mathbf{m}(\boldsymbol{z} \mid \mathcal{D}, \hat{\boldsymbol{z}})}\right) - \boldsymbol{c}^T(\boldsymbol{z} - \hat{\boldsymbol{z}}) + 1\right] = 0.$$

As this must hold for arbitrary $\boldsymbol{\eta}(\boldsymbol{z})$, in domains for which $\mathbf{m}^*(\boldsymbol{z}) > 0$, the term in brackets must be constant. Thus,

$$\mathbf{m}^*(\boldsymbol{z}) \propto \mathbf{m}(\boldsymbol{z} \mid \mathcal{D}, \hat{\boldsymbol{z}})e^{\boldsymbol{c}^T(\boldsymbol{z} - \hat{\boldsymbol{z}})}.$$

This analysis shows that the perturbation factor simply contributes an additional gradient to the NLPo approximation, $-\nabla_{\boldsymbol{z}}\log\mathbf{m}^*(\boldsymbol{z}) = -\nabla_{\boldsymbol{z}}\log\mathbf{m}(\boldsymbol{z} \mid \mathcal{D}, \hat{\boldsymbol{z}}) - \boldsymbol{c}$. Since we are interested in constructing a Laplace approximation, $\mathbf{p}(\boldsymbol{z} \mid \mathcal{D}) \approx \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\nu}, \boldsymbol{\Lambda}^{-1})$, we observe that the gradient of the perturbed Gaussian would be

$$\begin{aligned} -\nabla_{\boldsymbol{z}}\log\mathbf{m}^*(\boldsymbol{z}) &= -\nabla_{\boldsymbol{z}}\log\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\nu}, \boldsymbol{\Lambda}^{-1}) - \boldsymbol{c} \\ &= \boldsymbol{\Lambda}(\boldsymbol{z} - \boldsymbol{\nu}) - \boldsymbol{c}. \end{aligned}$$

For $\hat{\boldsymbol{z}}$ to be the mean of the perturbed distribution, the gradient would have to vanish so that $\boldsymbol{c} = \boldsymbol{\Lambda}(\hat{\boldsymbol{z}} - \boldsymbol{\nu})$. In other words, $\boldsymbol{c}$ serves the same role as evaluating the gradient of a log Laplace approximation at $\hat{\boldsymbol{z}}$, thus allowing us to reconstruct the Laplace approximation. In this sense, it is a weak gradient of the discrete posterior corresponding to an approximate first moment. Just as we were able to probe the structure of the precision matrix at sigma points to construct a likelihood approximation, we can use the same technique to probe and reconstruct a local posterior approximation.

We must solve the Lagrange coefficients using a sequence of Newton iterations, $\boldsymbol{c}^{(t)}$, that converges in a few steps $t = 1, 2, \cdots$. Initializing $\boldsymbol{c}^{(0)} = 0$, so that the perturbed likelihood gradients at $\hat{\boldsymbol{z}}$ become $\hat{\boldsymbol{g}}^{(t)} = \boldsymbol{g_0} + \boldsymbol{UCU}^T(\hat{\boldsymbol{z}} - \boldsymbol{z_0}) - \boldsymbol{c}^{(t)}$, we compute mean updates as

$$\hat{\boldsymbol{z}}^{'(t)} = \mathbb{E}_{\mathbf{m}(\boldsymbol{z}|\mathcal{D},\hat{\boldsymbol{z}},\boldsymbol{c}^{(t)})}\left[\boldsymbol{z}\right]$$

Note that the expectation in each coordinate, $\hat{\boldsymbol{z}}'_i$, only depends on $\boldsymbol{c}_i$. Taking the gradient of the expectation error with respect to $\boldsymbol{c}_i$ gives

$$\frac{\partial}{\partial \boldsymbol{c}_i}(\hat{\boldsymbol{z}}'_i - \hat{\boldsymbol{z}}_i) = \mathbb{E}_{\mathbf{m}(\boldsymbol{z}|\mathcal{D},\hat{\boldsymbol{z}},\boldsymbol{c}^{(t)})}\left[(\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i)^2\right] = \boldsymbol{\lambda}_i^{-1},$$

or the second moment centered at the target expectation $\hat{\boldsymbol{z}}$. We use this notation because, at convergence, $\boldsymbol{\lambda}$ becomes the diagonal of the posterior precision matrix. Thus we solve the update as

$$\hat{\boldsymbol{z}}_i = \hat{\boldsymbol{z}}_i^{'(t)} + \boldsymbol{\lambda}_i^{-1}(\boldsymbol{c}_i^{(t+1)} - \boldsymbol{c}_i^{(t)}) \quad \text{or} \quad \boldsymbol{c}^{(t+1)} = \boldsymbol{c}^{(t)} + (\hat{\boldsymbol{z}} - \hat{\boldsymbol{z}}^{'(t)}) \circledast \boldsymbol{\lambda}.$$

Note that when we compute $\boldsymbol{\lambda}$, if the likelihood function becomes tightly constrained around a single representation, the variance may vanish to numerical precision and cause this computation to become unstable. We can easily prevent this problem, however, by recognizing that each prior discretization is an approximation that represents a distribution. Suppose our belief in a single coordinate is described by a convex combination of distributions

$$\mathbf{p}(x) = \sum_{j=1}^{m} \alpha_j \mathbf{q}_j(x) \quad \text{where} \quad \alpha_j \geq 0 \quad \text{for all} \quad j \in [m] \quad \text{and} \quad \sum_{j}^{m} \alpha_j = 1.$$

If each distribution has mean $\mathbb{E}_{\mathbf{q}_j(x)}\left[x\right] = \mu_j$ and variance $\mathbb{E}_{\mathbf{q}_j(x)}\left[(x-\mu_j)^2\right] = \sigma_j^2$, then the mean and variance of $\mathbf{p}(x)$ are easily obtained as $\mathbb{E}_{\mathbf{p}(x)}[x] = \sum_{j=1}^{m} \alpha_j \mu_j = \mu$ and

$$\mathbb{E}_{\mathbf{p}(x)}[(x-\mu)^2] = \sum_{j=1}^{m} \alpha_j \, \mathbb{E}_{\mathbf{q}_j(x)}\left[(x - \mu_j + \mu_j - \mu)^2\right] = \sum_{j=1}^{m} \alpha_j \left(\sigma_j^2 + (\mu_j - \mu)^2\right).$$

When we compute the variance from the discrete posterior distribution, it is as though we are only accounting for the second term in the convex sum. By associating each representable location with an intrinsic variance, $\boldsymbol{\sigma}_j^2$, and remembering to correctly scale for reparameterization, we prevent the precision diagonal from diverging. Algorithm 2 provides details.

Algorithm 2 Weak Gradient of Posterior Laplace Approximation

Require: The vector of allowable parameter representations is $\boldsymbol{r}$ and each value has encoding length $\ell(\boldsymbol{r}_j)$ natural units. The NLL is approximated as $J(\boldsymbol{z}) = J_0 + (\boldsymbol{z} - \boldsymbol{z_0})^T \boldsymbol{g_0} + 1/2(\boldsymbol{z} - \boldsymbol{z_0})^T \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T(\boldsymbol{z} - \boldsymbol{z_0})$. Reparameterization scales are $\boldsymbol{\tau}$ so that $\boldsymbol{\theta} = \boldsymbol{z} \circledast \boldsymbol{\tau}$. The posterior gradient will be approximated at the evaluation point $\hat{\boldsymbol{z}}$. Optional intrinsic variance for each representation $\boldsymbol{r}_j$ is $\boldsymbol{\sigma}_j^2$.

Ensure: $\boldsymbol{c}$ serves as the gradient of a Laplace approximation of the posterior, $\boldsymbol{c} = -\nabla_{\boldsymbol{z}} \log \mathbf{q}(\boldsymbol{z} = \hat{\boldsymbol{z}} \mid \mathcal{D})$, and $\boldsymbol{\lambda}$ approximates the precision diagonal.

function WeakPosteriorGradient($\boldsymbol{r}, \ell(\boldsymbol{r}), \boldsymbol{\tau}, \boldsymbol{z_0}, J_0, \boldsymbol{g_0}, \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{c}, \hat{\boldsymbol{z}}, \boldsymbol{\sigma}$)

  Initialize $\boldsymbol{c} = 0$.

  for Newton steps $t = 1, 2, \ldots, 5$ **do**

    Evaluate conditional posteriors at $\hat{\boldsymbol{z}}$ with gradient perturbation $\boldsymbol{c}$,

$$\mathbf{q}(\boldsymbol{\theta}_i = \boldsymbol{r}_j \mid \mathcal{D}, \hat{\boldsymbol{z}}_i^c) = \text{PosteriorRB}(\boldsymbol{r}, \ell(\boldsymbol{r}), \boldsymbol{\tau}, \boldsymbol{z_0}, J_0, \boldsymbol{g_0}, \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{c}, \hat{\boldsymbol{z}})$$

    Evaluate mean update,

$$\hat{\boldsymbol{z}}_i' = \sum_{j=1}^{k} \frac{\boldsymbol{r}_j}{\boldsymbol{\tau}_i} \mathbf{q}(\boldsymbol{\theta}_i = \boldsymbol{r}_j \mid \mathcal{D}, \hat{\boldsymbol{z}}_i^c).$$

    Compute precision diagonals, $\boldsymbol{\lambda}_i^{-1} = \sum_{j=1}^{k} \left(\boldsymbol{r}_j/\boldsymbol{\tau}_i - \hat{\boldsymbol{z}}_i'\right)^2 \mathbf{q}(\boldsymbol{\theta}_i = \boldsymbol{r}_j \mid \mathcal{D}, \hat{\boldsymbol{z}}_i^c)$.

$$\boldsymbol{\lambda}_i^{-1} = \sum_{j=1}^{k} \left[\frac{\boldsymbol{\sigma}_j^2}{\boldsymbol{\tau}_i^2} + (\frac{\boldsymbol{r}_j}{\boldsymbol{\tau}_i} - \hat{\boldsymbol{z}}_i')^2 \right] \mathbf{q}(\boldsymbol{\theta}_i = \boldsymbol{r}_j \mid \mathcal{D}, \hat{\boldsymbol{z}}_i^c).$$

    Update weak gradient, $\boldsymbol{c} \leftarrow \boldsymbol{c} + (\hat{\boldsymbol{z}} - \hat{\boldsymbol{z}}') \circledast \boldsymbol{\lambda}$.

  end for

end function

2.7 Diagonal Plus Low-Rank Posterior Precision

Actually, we can go a bit further and include the diagonal of the precision matrix, in addition to a low-rank approximation of critical dimensions, in our posterior approximation. This requires little extra cost because we already have $\boldsymbol{\lambda}$ at each sigma point. As with the likelihood approximation, when we take differences of weak gradients at sigma point, we are effectively computing $\boldsymbol{X} = \boldsymbol{U}^T \boldsymbol{\Lambda} \boldsymbol{U}$, where $\boldsymbol{\Lambda}$ is the full posterior precision matrix. Since we can approximate $\text{diag}(\boldsymbol{\Lambda}) = Q[\boldsymbol{\lambda}]$, our diagonal plus low-rank approximation, $\boldsymbol{\Lambda} \approx \text{diag}(\boldsymbol{d}) + \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{U}^T$, must simultaneously satisfy

$$Q[\boldsymbol{\lambda}] = \boldsymbol{d} + \text{diag}\left(\boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{U}^T\right) \quad \text{and} \quad \boldsymbol{X} = \boldsymbol{U}^T \text{diag}(\boldsymbol{d})\boldsymbol{U} + \boldsymbol{\Delta}.$$

A simple iterative scheme easily accomplishes this. We initialize $\boldsymbol{d}^{(0)}$ to a minimum precision by setting the NLL to zero to obtain the localized precision of prior belief. Each

approximate diagonal $\boldsymbol{d}^{(t)}$ then induces an approximate core of the low-rank precision components

$$\boldsymbol{\Delta}^{(t)} = \boldsymbol{X} - \boldsymbol{U}^T \operatorname{diag}(\boldsymbol{d}^{(t)})\boldsymbol{U}.$$

A robust implementation requires diagonalization, $\boldsymbol{\Delta}^{(t)} = \boldsymbol{V} \operatorname{diag}(\boldsymbol{\delta})\boldsymbol{V}^T$, to remove negative eigenvalues. This ensures that the low-rank contribution only increases precision. We can then absorb the orthogonal matrices into an updated basis, $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{V}$, and evaluate the residuals, $\boldsymbol{s}$, of the full precision diagonal

$$\begin{aligned}\boldsymbol{s} &= \boldsymbol{d}^{(t)} + \operatorname{diag}(\boldsymbol{W} \operatorname{diag}(\boldsymbol{\delta})\boldsymbol{W}^T) - Q[\boldsymbol{\lambda}] \\ &= \boldsymbol{d}^{(t)} + \operatorname{diag}(\boldsymbol{W}[\boldsymbol{V}^T\boldsymbol{X}\boldsymbol{V} - \boldsymbol{W}^T \operatorname{diag}(\boldsymbol{d}^{(t)})\boldsymbol{W}]\boldsymbol{W}^T) - Q[\boldsymbol{\lambda}]\,.\end{aligned}$$

We can then compute the gradient of each residual element with respect to the corresponding diagonal as

$$\boldsymbol{\gamma}_i = \frac{\partial}{\partial \boldsymbol{d}_i^{(t)}} \boldsymbol{s}_i = 1 - \boldsymbol{e}_i^T \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{e}_i \boldsymbol{e}_i^T \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{e}_i = 1 - \left[\sum_{j=1}^{k} \boldsymbol{W}_{ij}^2\right]^2.$$

Thus, a suitable Newton update solves

$$0 = \boldsymbol{s} + \boldsymbol{\gamma} \circledast (\boldsymbol{d}^{(t+1)} - \boldsymbol{d}^{(t)}) \quad \text{so that} \quad \boldsymbol{d}^{(t+1)} = \boldsymbol{d}^{(t)} - \boldsymbol{s} \oslash \boldsymbol{\gamma}.$$

Blinders and Reparameterization

We found that there is high risk associated with allowing posterior updates to move too far from the hyperplane $\boldsymbol{z_0} + \operatorname{span}(\boldsymbol{U})$, within which we have reasonable knowledge of the likelihood structure. This is because we have, at best, only a rough analytic approximation of how the likelihood might react to such perturbations. Thus, it may be prudent to restrict posterior moment correction in dimensions that are orthogonal to the LIS. This can be accomplished by attaching a quadratic penality to the approximate negative log posterior in all dimensions orthogonal to $\boldsymbol{U}$. If we have iteratively concentrate Hessian eigenvalues through symmetric eigenvalue decompositions that retain the maximal eigenvalues, we can safely assume that all orthogonal dimensions will have an eigenvalue that is less than the minimum eigenvalue within the precision-maximizing subspace.

To improve implementation simplicity, we can always express the posterior precision matrix as identity plus low rank, $\boldsymbol{\Lambda} = \boldsymbol{I} + \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{U}$, provided we reparameterize to absorb the diagonal elements, $\boldsymbol{d}$, from the diagonal plus low-rank approximation above. We may rewrite the NLPo approximation as

$$\begin{aligned}K(\boldsymbol{z}) = K_0 + (\boldsymbol{z}' - \boldsymbol{z_0}')^T \boldsymbol{c_0}' + \frac{1}{2}(\boldsymbol{z}' - \boldsymbol{z_0}')^T \left[\boldsymbol{I} + \boldsymbol{W}' \operatorname{diag}(\boldsymbol{c})\boldsymbol{W}'^T\right](\boldsymbol{z}' - \boldsymbol{z_0}') \\ \text{where} \quad \boldsymbol{z}'_i = \boldsymbol{z}_i \boldsymbol{d}_i^{1/2}, \qquad \boldsymbol{c_0}'_i = \boldsymbol{c_0}'_i \boldsymbol{d}_i^{-1/2}, \quad \text{and} \quad \boldsymbol{W}'_{ij} = \boldsymbol{W}_{ij} \boldsymbol{d}_i^{-1/2}.\end{aligned}$$

To maintain $\boldsymbol{\theta} = \boldsymbol{z} \circledast \boldsymbol{\tau} = \boldsymbol{z}' \circledast \boldsymbol{\tau}'$, we simply update $\boldsymbol{\tau}' = \boldsymbol{\tau} \circledast \boldsymbol{d}^{-1/2}$. We must also reconstruct an orthonormal basis by reorthogonalizing $\boldsymbol{W}' = \boldsymbol{Q}\boldsymbol{R}$ using the QR decomposition

Algorithm 3 Diagonal Plus Low-Rank Posterior Precision

Require: The NLL is approximated as

$$J(\boldsymbol{z}) = J_0 + (\boldsymbol{z} - \boldsymbol{z_0})^T \boldsymbol{g_0} + 1/2 (\boldsymbol{z} - \boldsymbol{z_0})^T \boldsymbol{U C U}^T (\boldsymbol{z} - \boldsymbol{z_0}).$$

Ensure: Approximate the NLPo as

$$K(\boldsymbol{z}) = (\boldsymbol{z_0} - \boldsymbol{z})^T \boldsymbol{c_0} + 1/2 (\boldsymbol{z_0} - \boldsymbol{z})^T \left[\operatorname{diag}(\boldsymbol{d}) + \boldsymbol{W} \operatorname{diag}(\boldsymbol{\delta}) \boldsymbol{W}^T \right] (\boldsymbol{z_0} - \boldsymbol{z}).$$

function ApproximateNLPo($\boldsymbol{r}, \ell(\boldsymbol{r}), \boldsymbol{\tau}, \boldsymbol{z_0}, J_0, \boldsymbol{g_0}, \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{c}, \boldsymbol{\delta}, \boldsymbol{\sigma}$)

Initialize quadrature radius, $\rho = \sqrt{k + 1/2}$ and scaling factors $\boldsymbol{\sigma} = 1 \oslash \sqrt{1 + \boldsymbol{\delta}}$.

$$\boldsymbol{c} = \text{WeakPosteriorGradient}(\cdots, \hat{\boldsymbol{z}} = \boldsymbol{z_0})$$

for each basis vector $j = 1, 2, \ldots, k$ **do**

Weal NLPo gradient at j points,

$$\{\boldsymbol{c}^-, \boldsymbol{\lambda}^-\} = \text{WeakPosteriorGradient}(\cdots, \hat{\boldsymbol{z}} = \boldsymbol{z_0} - \boldsymbol{u}^{(j)} \rho \boldsymbol{\sigma}_j)$$
$$\{\boldsymbol{c}^+, \boldsymbol{\lambda}^+\} = \text{WeakPosteriorGradient}(\cdots, \hat{\boldsymbol{z}} = \boldsymbol{z_0} + \boldsymbol{u}^{(j)} \rho \boldsymbol{\sigma}_j)$$

Update running sums, $\boldsymbol{c} \leftarrow \boldsymbol{c} + \boldsymbol{c}^- + \boldsymbol{c}^+$ and $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \boldsymbol{\lambda}^- + \boldsymbol{\lambda}^+$.

Update column j of precision projection, $\boldsymbol{x}^{(j)} = \frac{1}{2\rho\boldsymbol{\sigma}_j} (\boldsymbol{c}^+ - \boldsymbol{c}^-)$.

end for

Compute means, $\boldsymbol{c_0} \leftarrow \frac{1}{2k+1} \boldsymbol{c}$ and $\boldsymbol{\lambda} \leftarrow \frac{1}{2k+1} \boldsymbol{\lambda}$

Symmetrize core, $\boldsymbol{X} \leftarrow 1/2 (\boldsymbol{X} + \boldsymbol{X}^T)$.

Initialize precision diagonal $\boldsymbol{d}$ term to minimum.

for Newton steps $t = 1, 2, \ldots, 5$ **do**

Approximate low rank core $\boldsymbol{\Delta} = \boldsymbol{X} - \boldsymbol{U}^T \operatorname{diag}(\boldsymbol{d}) \boldsymbol{U}^T$.

Diagonalize core, $\boldsymbol{\Delta} = \boldsymbol{V} \operatorname{diag}(\boldsymbol{\delta}) \boldsymbol{V}^T$.

Updated basis, $\boldsymbol{W} = \boldsymbol{U V}$, and residuals, $\boldsymbol{s} = \boldsymbol{d} + \operatorname{diag}(\boldsymbol{W} \operatorname{diag}(\boldsymbol{\delta}) \boldsymbol{W}^T) - \boldsymbol{\lambda}$.

Compute residual gradients, $\boldsymbol{\gamma}_i = 1 - (\sum_{j=1}^k \boldsymbol{W}_{ij}^2)^2$.

Update precision diagonal, $\boldsymbol{d} \leftarrow \boldsymbol{d} - \boldsymbol{s} \oslash \boldsymbol{\gamma}$.

end for

end function

and rediagonalizing the corresponding core, $\boldsymbol{R \Delta R}^T = \boldsymbol{V} \operatorname{diag}(\boldsymbol{\delta}) \boldsymbol{V}^T$. Then we finish the basis update as $\boldsymbol{U}' = \boldsymbol{Q V}$ and $\boldsymbol{\Delta}' = \operatorname{diag}(\boldsymbol{\delta})$. Since we also maintain the likelihood approximation, we update the gradient, $\boldsymbol{g_0}' = \boldsymbol{g_0} \circledast \boldsymbol{d}^{-1/2}$, and the core, $\boldsymbol{C}' = \boldsymbol{V}^T \boldsymbol{R C R}^T \boldsymbol{V}$, to be compatible with the new scaling and basis.

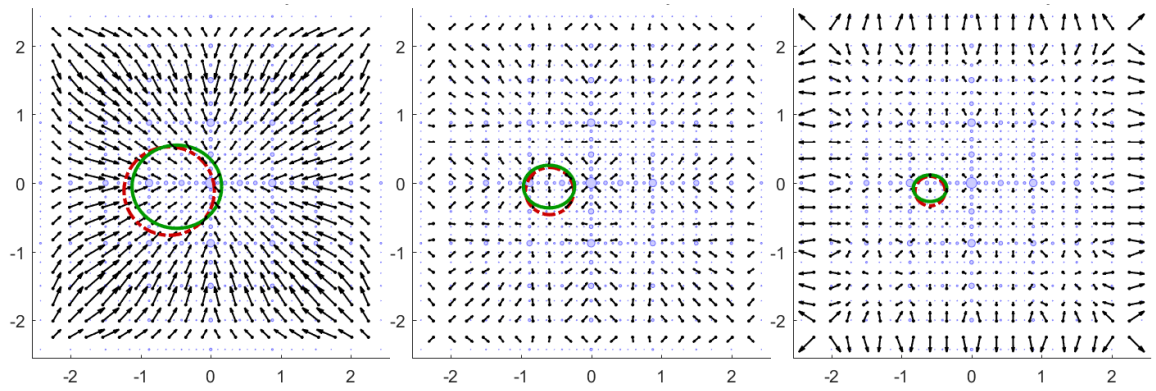

Figure 2: Visualization of likelihood to posterior updates in two dimensions. Each arrow shows how a likelihood mean at the given location would map to a posterior mean with discretized prior belief. The dashed circled represents 90% of the normalized likelihood probability and the solid elipse represents 90% of the updated posterior approximation for a specific example. Notice that as the likelihood domain becomes sharper, the posterior mapping allows for increasing specificity of outcomes, rather than only shifting toward the origin.

3. Main Algorithms

In order for training to begin, we often have to start with randomized parameters. Due to the high symmetry of the origin of a neural network parameterization, it is prone to be a critical saddle point in many dimensions. As a consequence, it is particularly difficult to obtain a reasonable quadratic approximation of the likelihood at the origin.

Further, we need efficient discovery of high likelihood domains to find good posterior approximations, which means it will not be sufficient to merely optimize our belief approximations with Newton steps. Instead, training must be formulated to allow discovery of better likelihood domains that are not captured by a local approximation. Even so, having the local curvature allows us to frame discovery without using tuned learning rates. This mechanism is explained first.

Next we show how after a discovery phase, we correct the local likelihood approximation so that we can construct a robust local posterior approximation. Our numerical experiments showed that aggressively optimizing the local posterior fails to allow the parameters to move to higher likelihood domains. Therefore, during posterior optimization, we must only take tentative steps that, while allowing convergence within a convex basin of attraction, would nevertheless allow the parameter state to drift to higher likelihood domains if discovered.

3.1 Exploration and Discovery

Discovering high likelihood domains requires an iterative strategy that will allow us to correct and optimize our local NLL approximations. Likelihood annealing gradually increases the degree to which our data have been taken into account by factorizing the posterior

distribution as

$$\mathbf{p}(\boldsymbol{z} \mid \mathcal{D}, \alpha) = \frac{\mathbf{p}(\mathcal{D} \mid \boldsymbol{z})^{\alpha}\mathbf{p}(\boldsymbol{z})}{\mathbf{p}(\mathcal{D} \mid \alpha)} \quad \text{where} \quad \mathbf{p}(\mathcal{D} \mid \alpha) = \int d\boldsymbol{z}\, \mathbf{p}(\mathcal{D} \mid \boldsymbol{z})^{\alpha}\mathbf{p}(\boldsymbol{z}).$$

The posterior easily follows after the complementary likelihood is taken into account,

$$\mathbf{p}(\boldsymbol{z} \mid \mathcal{D}) = \frac{\mathbf{p}(\mathcal{D} \mid \boldsymbol{z})^{1-\alpha}\mathbf{p}(\boldsymbol{z} \mid \mathcal{D}, \alpha)\mathbf{p}(\mathcal{D} \mid \alpha)}{\mathbf{p}(\mathcal{D})} \quad \text{with} \quad \mathbf{p}(\mathcal{D}) = \int d\boldsymbol{z}\, \mathbf{p}(\mathcal{D} \mid \boldsymbol{z})\mathbf{p}(\boldsymbol{z}).$$

The likelihood exploration phase begins from the current position $\boldsymbol{z_0}$ and an annealed quadratic NLL approximation. If we have a quadratic NLL approximation, $-\log \mathbf{p}(\mathcal{D} \mid \boldsymbol{z}) \approx J(\boldsymbol{z})$, then the annealed likelihood simply requires us to scale each term

$$J_0{}' = \alpha J_0 \quad , \quad \boldsymbol{g_0}' = \alpha \boldsymbol{g_0} \quad \text{, and} \quad \boldsymbol{C}' = \alpha \boldsymbol{C}$$

so that $-\log \mathbf{p}(\mathcal{D} \mid \boldsymbol{z})^{\alpha} \approx J'(\boldsymbol{z})$. For the first epoch, we have $\alpha = 0$ and all likelihood terms are zero.

Our likelihood exploration framework alternating expansion and extraction iteration similar to that of large sparse eigenvalue solvers (Duersch et al., 2018). After computing the average gradient over the first k dimensions, we can extract and normalize the remaining orthogonal component to finish extending the basis. Note that basis expansion dimensions always use the maximum scaling factor within the tracked subspace, $\sigma_{\max} = (1+\min(\boldsymbol{\delta}))^{-1/2}$. Although our basis expansion is constructed from a single orthogonal gradient component, a block formulation—potentially evaluating mini batches of data and generating orthonormal blocks at once—is an obvious extension of this basic approach.

After each evaluation the posterior precision is concentrated with so that $\boldsymbol{U}$ and $\boldsymbol{\Delta} = \operatorname{diag}(\boldsymbol{\delta})$ has descending eigenvalues. This allows us to extract a concentrated likelihood curvature approximation, the maximal precision eigenvalues, as well as the corresponding basis dimensions with limited memory and computational resources. When the center of our approximation $\boldsymbol{z_0}$ moves, the quadratic approximations are easily shifted to the updated expansion point as

$$\begin{aligned} J(\boldsymbol{z}) &= J_0 + (\boldsymbol{z} - \boldsymbol{z_0})^T \boldsymbol{g_0} + \frac{1}{2}(\boldsymbol{z} - \boldsymbol{z_0})^T \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T(\boldsymbol{z} - \boldsymbol{z_0}) \\ &= \left[J_0 + (\boldsymbol{z_0}' - \boldsymbol{z_0})^T \boldsymbol{g_0} + \frac{1}{2}(\boldsymbol{z_0}' - \boldsymbol{z_0})^T \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T(\boldsymbol{z_0}' - \boldsymbol{z_0})\right] \\ &\quad + (\boldsymbol{z} - \boldsymbol{z_0}')^T \left[\boldsymbol{g_0} + \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T(\boldsymbol{z_0}' - \boldsymbol{z_0})\right] \\ &\quad + \frac{1}{2}(\boldsymbol{z} - \boldsymbol{z_0}')^T \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T(\boldsymbol{z} - \boldsymbol{z_0}'). \end{aligned}$$

The exploration phase continues by iterating over all cases in the training dataset using Algorithm 4 to accumulate new likelihood components, $-\log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z})^{1-\alpha}$ for each $\boldsymbol{d} \in \mathcal{D}$, with the annealed likelihood starting values.

3.2 Correction and Posterior Optimization

The approximate posterior updates in Algorithm 4 are very rough, but the proceedure successfully moves $\boldsymbol{z_0}$ to high likelihood domains. To better account for discretized prior

Algorithm 4 Explore Case

Require: The current view is $\mathbf{r}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{z_0}, \boldsymbol{\Gamma} = [\boldsymbol{I} + \boldsymbol{U}\,\mathrm{diag}(\boldsymbol{\delta})\boldsymbol{U}^T]^{-1})$. Basis matrices, $\boldsymbol{U}$, contain k orthonormal columns corresponding to increased precision $\boldsymbol{\delta} > 0$. The basis should be extended as needed to also include important states in the span, $\mathcal{Z} = \{\boldsymbol{z}^{(s)}\} \in \mathrm{span}(\boldsymbol{U})$.

Ensure: Accumulate likelihood contribution $\mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z})^{1-\alpha}$ within quadratic NLL and NLPo approximations. Take exploration step.

function ExploreCase($\boldsymbol{d}, \boldsymbol{z_0}, J_0, \boldsymbol{g_0}, \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{\delta}$)

Initialize quadrature radius, $\rho = \sqrt{k + 3/2}$, scaling factors $\boldsymbol{\sigma} = 1 \oslash \sqrt{1 + \boldsymbol{\delta}}$.

Use sigma point quadrature to integrate NLL, gradient, and precision core,

$$J = -\log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0}) \quad \text{and} \quad \boldsymbol{g} = -\nabla_{\boldsymbol{z}} \log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0}).$$

for each basis vector $j = 1, 2, \ldots, k, k+1$ **do**

Evaluate NLL and gradient at j points,

$$J^- = -\log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0} - \boldsymbol{u}^{(j)}\rho\boldsymbol{\sigma}_j) \qquad J^+ = -\log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0} + \boldsymbol{u}^{(j)}\rho\boldsymbol{\sigma}_j)$$
$$\boldsymbol{g}^- = -\nabla_{\boldsymbol{z}} \log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0} - \boldsymbol{u}^{(j)}\rho\boldsymbol{\sigma}_j) \qquad \boldsymbol{g}^+ = -\nabla_{\boldsymbol{z}} \log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0} + \boldsymbol{u}^{(j)}\rho\boldsymbol{\sigma}_j).$$

If $j = k + 1$, extract final orthonormal basis extension from $\boldsymbol{g}$ and evaluate as above.

Update running sums, $J \leftarrow J + J^- + J^+$ and $\boldsymbol{g} \leftarrow \boldsymbol{g} + \boldsymbol{g}^- + \boldsymbol{g}^+$.

Update column j of precision projection, $\boldsymbol{c}^{(j)} = \frac{1}{2\rho\boldsymbol{\sigma}_j}(\boldsymbol{g}^+ - \boldsymbol{g}^-)$.

end for

Compute means, $J \leftarrow \frac{J}{2k+3}$ and $\boldsymbol{g} \leftarrow \frac{\boldsymbol{g}}{2k+3}$. Symmetrize core, $\boldsymbol{C} \leftarrow 1/2(\boldsymbol{C} + \boldsymbol{C}^T)$.

Update approximations, $J_0 \leftarrow J_0 + (1-\alpha)(J - \frac{1}{2}\,\mathrm{tr}(\boldsymbol{C}\,\mathrm{diag}(\boldsymbol{\sigma})^2))$,

$$\boldsymbol{g_0} \leftarrow \boldsymbol{g_0} + (1-\alpha)\boldsymbol{g}, \qquad \boldsymbol{C} \leftarrow \boldsymbol{C} + (1-\alpha)\boldsymbol{C},$$
$$\boldsymbol{c_0} \leftarrow \boldsymbol{c_0} + (1-\alpha)\boldsymbol{g}, \qquad \boldsymbol{\Delta} \leftarrow \boldsymbol{\Delta} + (1-\alpha)\boldsymbol{C}.$$

Diagonalize posterior precision $\boldsymbol{\Delta}$ and update basis $\boldsymbol{U}$ accordingly.

Compute $\frac{3}{2}$ Newton step, $\boldsymbol{z_0} \leftarrow \boldsymbol{z_0} - \frac{3}{2}(1-\alpha)(\boldsymbol{I} + \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{U}^T)^{-1}\boldsymbol{g}$.

Update quadratic approximation for new expansion point.

end function

belief, however, we must use the techniques in Sections 2.6 and 2.7. Unfortunately, the domain shifts also reduce the accuracy of $J(\boldsymbol{z})$. Thus, before we update our posterior distribution, we reconstruct a more accurate likelihood.

This is easily accomplished by using an algorithm very similar to Algorithm 4 that simply avoids movement and new basis extensions. Resetting $J_0 = 0$, $\boldsymbol{g_0} = 0$, and $\boldsymbol{C} = 0$, we can construct a better local NLL approximation by iterating Algorithm 5 over the full training dataset.

Algorithm 5 Correct Case

Require: The current view is $\mathbf{r}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{z_0}, \boldsymbol{\Gamma} = [\boldsymbol{I} + \boldsymbol{U}\,\mathrm{diag}(\boldsymbol{\delta})\boldsymbol{U}^T]^{-1})$. Basis matrices, $\boldsymbol{U}$, contain k orthonormal columns corresponding to increased precision $\boldsymbol{\delta} > 0$. The basis should still be extended to include $\mathcal{Z} \subset \mathrm{span}(\boldsymbol{U})$.

Ensure: Reconstruct quadratic NLL approximation from full contributions, $\mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z})$.

function CorrectCase($\boldsymbol{d}, \boldsymbol{z_0}, J_0, \boldsymbol{g_0}, \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{\delta}$)

Initialize quadrature radius, $\rho = \sqrt{k + 1/2}$, scaling factors $\boldsymbol{\sigma} = 1 \oslash \sqrt{1 + \boldsymbol{\delta}}$.

Use sigma point quadrature to integrate NLL, gradient, and precision core,

$$J = -\log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0}) \quad \text{and} \quad \boldsymbol{g} = -\nabla_{\boldsymbol{z}} \log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0}).$$

for each basis vector $j = 1, 2, \ldots, k$ **do**

Evaluate NLL and gradient at j points,

$$J^- = -\log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0} - \boldsymbol{u}^{(j)}\rho\boldsymbol{\sigma}_j) \qquad J^+ = -\log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0} + \boldsymbol{u}^{(j)}\rho\boldsymbol{\sigma}_j)$$
$$\boldsymbol{g}^- = -\nabla_{\boldsymbol{z}} \log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0} - \boldsymbol{u}^{(j)}\rho\boldsymbol{\sigma}_j) \qquad \boldsymbol{g}^+ = -\nabla_{\boldsymbol{z}} \log \mathbf{p}(\boldsymbol{d} \mid \boldsymbol{z_0} + \boldsymbol{u}^{(j)}\rho\boldsymbol{\sigma}_j).$$

Update running sums, $J \leftarrow J + J^- + J^+$ and $\boldsymbol{g} \leftarrow \boldsymbol{g} + \boldsymbol{g}^- + \boldsymbol{g}^+$.

Update column j of precision projection, $\boldsymbol{c}^{(j)} = \frac{1}{2\rho\boldsymbol{\sigma}_j}(\boldsymbol{g}^+ - \boldsymbol{g}^-)$.

end for

Compute means, $J \leftarrow \frac{J}{2k+1}$ and $\boldsymbol{g} \leftarrow \frac{\boldsymbol{g}}{2k+1}$. Symmetrize core, $\boldsymbol{C} \leftarrow 1/2(\boldsymbol{C} + \boldsymbol{C}^T)$.

Update approximations, $J_0 \leftarrow J_0 + (J - \frac{1}{2}\,\mathrm{tr}(\boldsymbol{C}\,\mathrm{diag}(\boldsymbol{\sigma})^2))$,

$$\boldsymbol{g_0} \leftarrow \boldsymbol{g_0} + \boldsymbol{g}, \qquad \boldsymbol{C} \leftarrow \boldsymbol{C} + \boldsymbol{C}.$$

end function

With this, we can construct a better NLPo approximation and take a step to move $\boldsymbol{z_0}$ towards higher posterior domains. We find, however, that aggressively moving toward the local posterior maximizer, a full Newton step, interfers with the ability of subsequent epochs to move towards higher likelihood domains that would eventually dominate the posterior. By taking half steps, we still ensure that an equilibrium will be reached that balances likelihood exploration with posterior optimization, but still allows the center of our belief approximation to drift to high likelihood domains early on.

4. Numerical Experiments, Discussion, and Summary

Our training experiments use a small neural network, with 1114 parameters, to classify MNIST (LeCun et al., 1998) digits from only 200 randomly selected images. The discretized prior contains 31 representations that correspond to all Gaussian subintervals formed by taking from 0 up to 4 bisections, thus having a maximum to minimum prior probability ratio of 16:1 for the shortes (zero) to longest encodings.

The likelihood and posterior approximations concentrate precision in a 30-dimensional subspace. This basis is extended to include the 6 most recent parameters states, which include states that result from either full exploration epoch or the posterior update that follows it. Each training run includes 20 posterior updates, amounting to 40 passes over the data if we include likelihood corrections.

When we compare our training algorithm to SGD, with a learning rate and L_2 regularization tuned with cross-validation, we see in Figure 3 that our algorithm navigates the parameter space much more efficiently to generalizable prediction domains. It is important to note, however, that each epoch of our algorithm is significantly more expensive to evaluate.

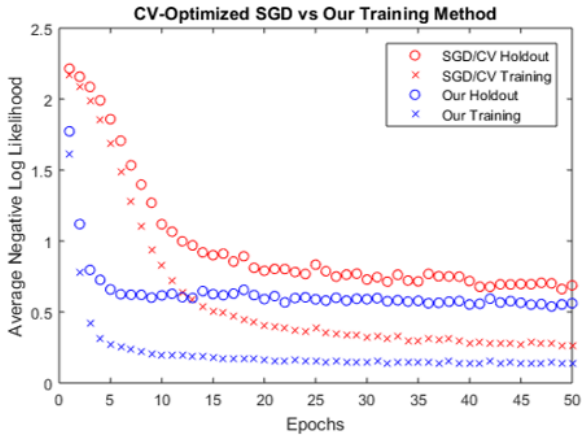

Figure 3: Training comparison of our algorithm to SGD tuned using cross-validation with additional data to optimize both the learning rate and L_2 regularization weight. We see that despite finding an SGD learning rate that optimizes initial loss descent, our algorithm is able to navigate through the parameter space more efficiently. In this particular case, our algorithm also shows better performance on holdout data where were not used during optimization. We note, however, that the prediction quality varies significantly over the typical basins of posterior attraction both algorithms are subject to discover.

The posterior ensemble is formed from 120 training runs. Figure Figure 4 shows that the ensemble predictions, formed by simply averaging the predictions obtained from each individual model, outperforming all of the individual models.

4.1 Sensitivity of Training Trajectories

During the course of our experiments with generating ensembles of models, we find that both the initial position of $\boldsymbol{z_0}$ as well as the random ordering of the training data can affect the quality of the basin of posterior attraction to which the model converges. This difference in quality is often stark and observable within the first few epochs. It may be useful in future work to consider that optimizing an initial ensemble may be more beneficial than using substantial computational resources to optimize a training trajectory that could

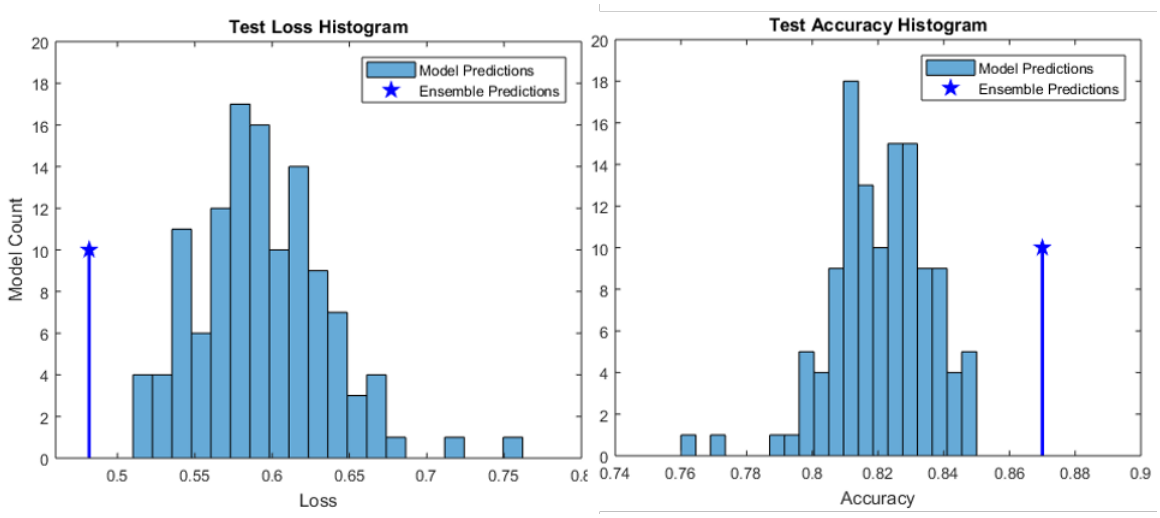

Figure 4: Ensemble prediction quality is measured using both accuracy and holdout loss, the negative log likelihood applied to unseen data. We see that ensemble averaging yields predictions that are better, by both metrics, than any individual model.

have been pruned early on. More generally, we would like to consider methods that reduce the dependence of the final ensemble quality on the stochastic qualities of training.

4.2 Dimensionality Reduction

If the true posterior distribution has nontrivial dependence in several dimensions of a given architecture, we cannot avoid the computationally difficult task of characterizing the posterior dependence in each of those dimensions in order to obtain a reasonable approximation of the posterior-predictive distribution for robust uncertainty quantification on new data. The fact that many standard practices work well without attempting to capture such a comprehensive description of the posterior indicates that it should be possible to efficiently discover manifolds that dominate the posterior, and thus posterior-predictive integrals, that would significantly reduce the amount of computation needed to obtain robust predictions.

4.3 Summary

We developed a training approach that is fundamentally designed to limit the amount of information that may be contained within trained network parameters. This is accomplished by allowing only a limited set of specific parameter representations, which can then be efficiently encoded. Moreover, the discretization we proposed contains codes of increasing length to efficiently represent increasing specificity. Encodings with a given level of specificity are distributed evenly across the space and codes of increasing length are interleaved to gradually increase the density of representations between shorter codes.

Within the theory of algorithmic probability, optimizing the posterior probability of such models allows us to construct an ensemble that approximates the posterior-predictive

integral. As anticipated by the theory of algorithmic probability, we demonstrated that combining our training subroutine with ensemble averaging allows us to obtain predictions that generalize better than any individual model discovered. In order to compute individual models within the ensemble, we investigated mechanisms to navigate the parameter space efficiently. While the original formulation of Algorithmic Probability is computationally challenging, perhaps even intractible, the methods we developed are able to tractably suppress information through simplified parameters.

Training proceeds by approximating and optimizing quadratic approximation of the negative log likelihood using sigma point quadratures that are exact to 3rd-order within a critical hyperplane and 1st-order otherwise. The critical hyperplane is iteratively improved by concentrating likelihood precision with a sequence of truncated eigenvalue decompositions. Futher, we also track additional key dimensions that training encounters over the most recent epochs in order to better accurately capture structure in the dominant directions of improvement. We then explained how to merge our continuous quadratic likelihood with discretized prior belief to obtain a quadratic approximation of the log posterior. This approximation is designed to perserve the structure of first moments that arise from the discrete posterior, rather than directly within the space of representations.

We incorporated these techniques into a training algorithm that iteratively improves posterior approximations, which can then be used to form an ensemble of models. Our experiments show that this training technique moves through the parameter more efficiently that tuned stochastic gradient descent to find basins of high posterior probability, allowing us to construct ensembles that generalize to yield superior predictions. This provides an important step towards limiting our dependence on cross-validation during training, which becomes highly problematic when we only have small datasets.

This work required a substantial investment to understand essential properties of efficient parameter navigation subject to discretized prior belief and design compatible subroutines with computationally efficient approximations. Although our experiments are promising, additional investigation is still required to see if these methods are robust under a variety of contexts. Given the implemenation difficulty of these approaches, we anticipate investigating simplifications to both architectures and information suppression methods to improve the adoption and impact of this work.

Appendix A. Appendix

A.1 First Alternative Posterior Approximation

We also developed an alternative form of the Laplace approximation of the posterior driven by analysis of the unperturbed distribution $\mathbf{m}(\boldsymbol{z} \mid \mathcal{D}, \hat{\boldsymbol{z}})$. Suppose we have a Gaussian distribution

$$\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \propto \exp\left(\frac{-1}{2}\left[(\boldsymbol{z}-\boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{z}-\boldsymbol{\mu})\right]\right)$$

We consider holding all coordiantes fixed at $\hat{\boldsymbol{z}}$ except one. Without loss of generality, let us write the free coordinate as $\boldsymbol{y}_1 = \boldsymbol{z}_1 - \boldsymbol{\mu}_1$ and and evaluate the complementary coordinates

at the fixed location $\hat{\boldsymbol{z}}$, written as $\boldsymbol{x}_2 = \hat{\boldsymbol{z}}_2 - \boldsymbol{\mu}_2$, so that

$$\boldsymbol{z} - \boldsymbol{\mu} \mapsto \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \quad \text{and compatibly partition} \quad \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1 & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{bmatrix}$$

This gives

$$\begin{aligned} -\log \mathbf{q}(\boldsymbol{y}_1 \mid \boldsymbol{x}_2) &= c_1 + \frac{1}{2}\boldsymbol{y}_1^2 \boldsymbol{\lambda}_1 + \boldsymbol{y}_1 \boldsymbol{\Lambda}_{12} \boldsymbol{x}_2 \\ &= c_1 + \frac{1}{2}(\boldsymbol{y}_1 - \boldsymbol{a}_1 + \boldsymbol{a}_1)^2 \boldsymbol{\lambda}_1 + (\boldsymbol{y}_1 - \boldsymbol{a}_1 + \boldsymbol{a}_1)\boldsymbol{\Lambda}_{12}\boldsymbol{x}_2 \\ &= c_2 + \frac{1}{2}(\boldsymbol{y}_1 - \boldsymbol{a}_1)^2 \boldsymbol{\lambda}_1 + (\boldsymbol{y}_1 - \boldsymbol{a}_1)\left[\boldsymbol{\lambda}_1 \boldsymbol{a}_1 + \boldsymbol{\Lambda}_{12}\boldsymbol{x}_2\right] \end{aligned}$$

where we have absorbed constants independent of $\boldsymbol{y}_1$ into the definitions of c. The quadratic term contains all $\boldsymbol{y}_1$ dependence if we set $\boldsymbol{a}_1 = -\boldsymbol{\lambda}_1^{-1}\boldsymbol{\Lambda}_{12}\boldsymbol{x}_2$, which must be the mean of $\boldsymbol{y}_1$ as conditioned by holding all other coordinates at $\boldsymbol{x}_2$. Noting that $\boldsymbol{\Lambda}_{12}\boldsymbol{x}_2 = \boldsymbol{e}_1^T\boldsymbol{\Lambda}\boldsymbol{x} - \boldsymbol{\lambda}_1\boldsymbol{x}_1$, we can write the vector of all such means, each formed independently by holding the complementary coordinates fixed, as a vector

$$\boldsymbol{a} = \boldsymbol{x} - \operatorname{diag}(\boldsymbol{\lambda})^{-1}\boldsymbol{\Lambda}\boldsymbol{x}$$

It easily follows that the updated mean $\hat{\boldsymbol{z}}'$ would be

$$\begin{aligned} \hat{\boldsymbol{z}}_i' &= \mathbb{E}_{\mathbf{q}(\boldsymbol{z}_i \mid \hat{\boldsymbol{z}}_i^c)}[\boldsymbol{z}_i] && = \boldsymbol{e}_i^T\left(\boldsymbol{\mu} + \boldsymbol{a}\right) \\ &= \boldsymbol{e}_i^T\left(\boldsymbol{z} - \operatorname{diag}(\boldsymbol{\lambda})^{-1}\boldsymbol{\Lambda}(\boldsymbol{z} - \boldsymbol{\mu})\right) \end{aligned}$$

or simply $\hat{\boldsymbol{z}}' = \hat{\boldsymbol{z}} - \operatorname{diag}(\boldsymbol{\lambda})^{-1}\boldsymbol{\Lambda}(\hat{\boldsymbol{z}} - \boldsymbol{\mu})$. Thus, we can construct precision matrix-vector products, as we did for the likelihood, if we have an approximation for the precision diagonal $\boldsymbol{\lambda} \approx \operatorname{diag}(\boldsymbol{\Lambda})$. That is, $\operatorname{diag}(\boldsymbol{\lambda})(\hat{\boldsymbol{z}} - \hat{\boldsymbol{z}}') = \boldsymbol{\Lambda}(\hat{\boldsymbol{z}} - \boldsymbol{\mu})$. If we evaluate $\hat{\boldsymbol{z}}$ at a point in the hyperplane containing $\boldsymbol{z_0}$, so that $\hat{\boldsymbol{z}} = \boldsymbol{z_0} + \boldsymbol{U}\boldsymbol{\varphi}$, we have

$$\begin{aligned} \operatorname{diag}(\boldsymbol{\lambda})(\hat{\boldsymbol{z}} - \hat{\boldsymbol{z}}') &= \boldsymbol{\Lambda}(\boldsymbol{z_0} + \boldsymbol{U}\boldsymbol{\varphi} - \boldsymbol{\mu}) \\ &= \boldsymbol{c_0} + \boldsymbol{\Lambda}\boldsymbol{U}\boldsymbol{\varphi} \quad \text{where} \quad \boldsymbol{c_0} = \boldsymbol{\Lambda}(\boldsymbol{z_0} - \boldsymbol{\mu}). \end{aligned}$$

Thus we can reconstruct an approximate gradient and low-rank precision centered at the expansion point $\boldsymbol{z_0}$ as before.

A.2 Second Alternative Posterior Approximation

We also examined a posterior approximation method that operates in two steps. First, we approximate the posterior covariance as a simple composition of Gaussians to obtain a simple approximation of principal components of the posterior precision. Then we compute an update to each component using a quadrature formula that is designed to replicate power iteration against the covariance matrix.

If we had a normal prior, $\mathcal{N}(\boldsymbol{z} \mid 0, \boldsymbol{I})$, and an improper Gaussian likelihood with low-rank precision, $\mathbf{q}(\mathcal{D} \mid \boldsymbol{z}) = \exp\left[\frac{-1}{2}(\boldsymbol{z} - \boldsymbol{\mu})^T\boldsymbol{U}\operatorname{diag}(\boldsymbol{\delta})\boldsymbol{U}^T(\boldsymbol{z} - \boldsymbol{\mu})\right]$, normalization easily yields the composite distribution $\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{U}\boldsymbol{\nu}, \boldsymbol{\Gamma})$ with $\boldsymbol{\nu}$ and $\boldsymbol{\Gamma}$ computed as follows. Arguments to the

diagonal matrix constructions below, $\mathrm{diag}(\cdot)$, are always evaluated elementwise. Further, $\boldsymbol{U}_\perp$ is a complementary orthonormal basis to $\boldsymbol{U}$ so that $\boldsymbol{U}_\perp^T\boldsymbol{U}_\perp = \boldsymbol{I}$ and $\boldsymbol{U}_\perp^T\boldsymbol{U} = 0$. We have,

$$\boldsymbol{\nu} = \mathrm{diag}(\frac{\boldsymbol{\delta}}{1+\boldsymbol{\delta}})\boldsymbol{U}^T\boldsymbol{\mu} \quad \text{and} \quad \boldsymbol{\Gamma} = \begin{bmatrix}\boldsymbol{U} & \boldsymbol{U}_\perp\end{bmatrix}\begin{bmatrix}\mathrm{diag}(\frac{1}{1+\boldsymbol{\delta}}) & 0 \\ 0 & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{U}^T \\ \boldsymbol{U}_\perp^T\end{bmatrix}.$$

Note that the improper precision eigenvectors $\boldsymbol{u}_i$ remain eigenvectors of the composite covariance and we can represent the result as a product of independent Gaussians with a simple basis transformation, $\boldsymbol{c} = \begin{bmatrix}\boldsymbol{U}^T \\ \boldsymbol{U}_\perp^T\end{bmatrix}\boldsymbol{z}$ so that

$$\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\nu}, \boldsymbol{\Gamma}) = \prod_{i=1}^{r}\mathcal{N}(\boldsymbol{c}_i \mid \boldsymbol{\nu}_i, \frac{1}{1+\boldsymbol{\delta}_i})\prod_{i=r+1}^{n}\mathcal{N}(\boldsymbol{c}_i \mid 0, 1)$$

and $\boldsymbol{z} = \begin{bmatrix}\boldsymbol{U} & \boldsymbol{U}_\perp\end{bmatrix}\boldsymbol{c}$. This allows us to separate and evaluate integrals in each $\boldsymbol{c}$ dimension independently.

Our objective is to find a mechanism to approximate the principal components of the composite covariance where $\mathbf{p}(\boldsymbol{z})$ is discrete, rather than the standard normal approximation. Since we already have a method to evaluate expectations coordinate by coordinate from the parsimonious prior composed with an improper Gaussian likelihood, we can compute these components as the eigenvector scaled by the corresponding standard deviation

$$\boldsymbol{u}_i\boldsymbol{\sigma}_i = \int d\boldsymbol{z}\,(\boldsymbol{z} - \boldsymbol{U}\boldsymbol{\nu})\left[(\boldsymbol{z} - \boldsymbol{U}\boldsymbol{\nu})^T\boldsymbol{u}_i\boldsymbol{\sigma}_i^{-1}\mathcal{N}(\boldsymbol{z} \mid \boldsymbol{U}\boldsymbol{\nu}, \boldsymbol{\Gamma})\right].$$

We found that the first-order moment kernel, in brackets, may be approximated as a difference of two Gaussians

$$\begin{aligned}(\boldsymbol{c}_i - \boldsymbol{\nu}_i)^T\boldsymbol{\sigma}_i^{-1}&\mathcal{N}(\boldsymbol{c}_i \mid \boldsymbol{\nu}_i, \frac{1}{1+\boldsymbol{\delta}_i}) \\ &\approx \mathrm{const}\,\mathcal{N}(\boldsymbol{c}_i \mid 0, 1)\left(\exp\left[\frac{-(1+12\boldsymbol{\delta}_i)}{22}\left(\boldsymbol{c}_i - \frac{12\boldsymbol{\delta}_i\boldsymbol{u}_i^T\boldsymbol{\mu} + 6\sqrt{1+\boldsymbol{\delta}_i}}{1+12\boldsymbol{\delta}_i}\right)^2\right]\right. \\ &\qquad \left. - \exp\left[\frac{-(1+12\boldsymbol{\delta}_i)}{22}\left(\boldsymbol{c}_i - \frac{12\boldsymbol{\delta}_i\boldsymbol{u}_i^T\boldsymbol{\mu} - 6\sqrt{1+\boldsymbol{\delta}_i}}{1+12\boldsymbol{\delta}_i}\right)^2\right]\right)\end{aligned}$$

where the constant of proportionality normalizes the composition of the standard normal factor with each improper Gaussian in the difference.

Within this eigenspace, the leading moments of the original kernel are

$$\int d\boldsymbol{c}_i\,(\boldsymbol{c}_i - \boldsymbol{\nu}_i)^m\left[(\boldsymbol{c}_i - \boldsymbol{\nu}_i)^T\boldsymbol{\sigma}_i^{-1}\mathcal{N}(\boldsymbol{c}_i \mid \boldsymbol{\nu}_i, \frac{1}{1+\boldsymbol{\delta}_i})\right] \tag{1}$$

$$\left\{0, \quad \frac{1}{(1+\boldsymbol{\delta}_i)^{1/2}}, \quad 0, \quad \frac{3}{(1+\boldsymbol{\delta}_i)^{3/2}}, \quad 0\right\} \tag{2}$$

for $m = \{0, 1, 2, 3, 4\}$, respectively. In comparison, the corresponding moments, centered at $\boldsymbol{\nu}_i$, of each normalized Gaussian in the difference above are

$$\left\{1, \quad \frac{1}{2(1+\boldsymbol{\delta}_i)^{1/2}}, \quad \frac{7}{6(1+\boldsymbol{\delta}_i)}, \quad \frac{3}{2(1+\boldsymbol{\delta}_i)^{3/2}}, \quad \frac{95}{24(1+\boldsymbol{\delta}_i)^2}\right\} \quad \text{and} \tag{3}$$

$$\left\{1, \quad \frac{-1}{2(1+\boldsymbol{\delta}_i)^{1/2}}, \quad \frac{7}{6(1+\boldsymbol{\delta}_i)}, \quad \frac{-3}{2(1+\boldsymbol{\delta}_i)^{3/2}}, \quad \frac{95}{24(1+\boldsymbol{\delta}_i)^2}\right\}, \quad \text{respectively.} \tag{4}$$

Thus, this difference of Gaussians exactly integrates 4th-order polynomials. When we replace the standard normal Gaussian with the discretized prior, $\mathbf{p}(\boldsymbol{z})$, we can construct each expectation in the difference as before, coordinate by coordinate.

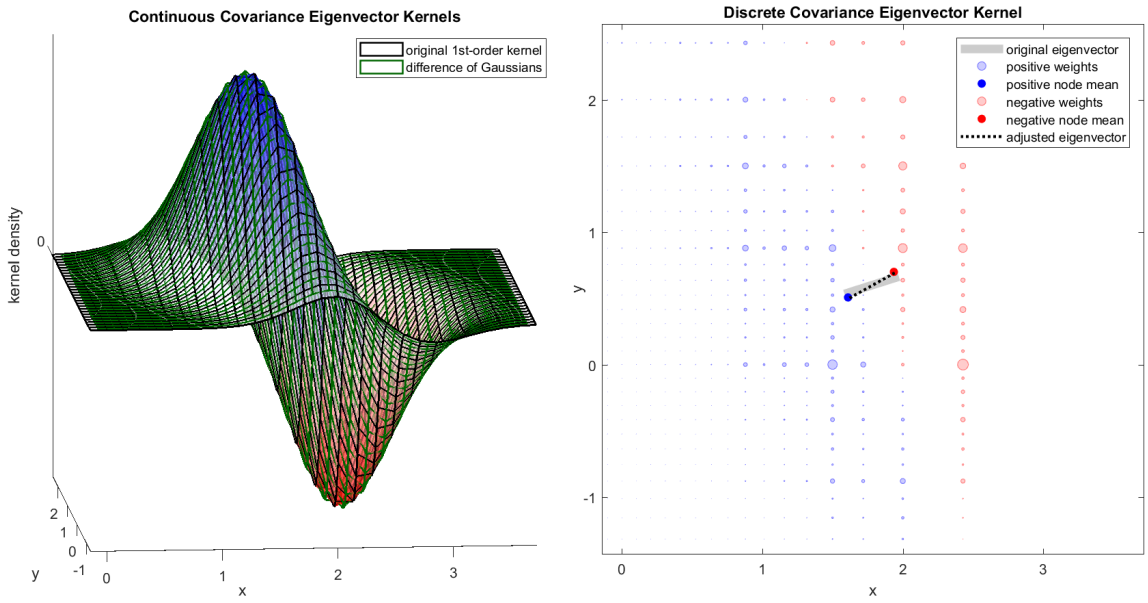

Figure 5: Visualization of continuous first-order eigenvector kernel and our approximation as a difference in Gaussians. The approximation is exact to 4th order and we observe no visually discernable difference. At the right, we see the corresponding discrete kernel and note the slight correction accounting for representational simplicity.

Bibliography

L. Baldassarre, J. Morales, A. Argyriou, and M. Pontil. A general framework for structured sparsity via proximal optimization. In *Artificial Intelligence and Statistics*, pages 82–90. PMLR, 2012.

D. Blackwell. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, pages 105–110, 1947.

L. Blier and Y. Ollivier. The description length of deep learning models. *arXiv preprint arXiv:1802.07044*, 2018.

G. Casella and C. P. Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1): 81–94, 1996.

M. Courbariaux, Y. Bengio, and J.-P. David. Training deep neural networks with low precision multiplications. *arXiv preprint arXiv:1412.7024*, 2014.

M. Courbariaux, Y. Bengio, and J.-P. David. Binaryconnect: Training deep neural networks with binary weights during propagations. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper/2015/file/3e15cc11f979ed25912dff5b0669f2cd-Paper.pdf`.

M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

T. Cui, J. Martin, Y. M. Marzouk, A. Solonen, and A. Spantini. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems*, 30(11):114015, 2014.

E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig. Laplace redux–effortless bayesian deep learning. *arXiv preprint arXiv:2106.14806*, 2021.

J. A. Duersch and T. A. Catanach. Generalizing information to the evolution of rational belief. *Entropy*, 22(1):108, 2020.

J. A. Duersch and T. A. Catanach. Parsimonious inference. *arXiv preprint arXiv:2103.02165*, 2021.

J. A. Duersch, M. Shao, C. Yang, and M. Gu. A robust and efficient implementation of LOBPCG. *SIAM Journal on Scientific Computing*, 40(5):C655–C676, Jan. 2018. doi: 10.1137/17m1129830. URL https://doi.org/10.1137/17m1129830.

N. Ebrahimi, E. S. Soofi, and R. Soyer. Information measures in perspective. *International Statistical Review*, 78(3):383–412, 2010.

P. Grünwald and T. Roos. Minimum description length revisited. *International journal of mathematics for industry*, 11(01):1930001, 2019.

S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan. Deep learning with limited numerical precision. In *International conference on machine learning*, pages 1737–1746. PMLR, 2015.

P. Gysel, M. Motamedi, and S. Ghiasi. Hardware-oriented approximation of convolutional neural networks. *arXiv preprint arXiv:1604.03168*, 2016.

S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*, 2015.

S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2016.

G. E. Hinton and D. van Camp. Keeping neural networks simple. In *International Conference on Artificial Neural Networks*, pages 11–18. Springer, 1993.

E. T. Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

S. Kullback. *Information theory and statistics*. Courier Corporation, 1997.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, Mar. 1951. doi: 10.1214/aoms/1177729694.

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

D. Lin, S. Talathi, and S. Annapureddy. Fixed point quantization of deep convolutional networks. In *International conference on machine learning*, pages 2849–2858. PMLR, 2016.

D. D. Lin and S. S. Talathi. Overcoming challenges in fixed point training of deep convolutional networks. *arXiv preprint arXiv:1607.02241*, 2016.

C. Louizos, K. Ullrich, and M. Welling. Bayesian compression for deep learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/69d1fc78dbda242c43ad6590368912d4-Paper.pdf.

D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. ISBN 0521642981.

Y. M. Marzouk and H. N. Najm. Dimensionality reduction and polynomial chaos acceleration of bayesian inference in inverse problems. *Journal of Computational Physics*, 228 (6):1862–1902, 2009.

D. A. McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999.

N. Mellempudi, A. Kundu, D. Mudigere, D. Das, B. Kaul, and P. Dubey. Ternary neural networks with fine-grained quantization. *arXiv preprint arXiv:1705.01462*, 2017.

H. M. Menegaz, J. Y. Ishihara, G. A. Borges, and A. N. Vargas. A systematization of the unscented kalman filter theory. *IEEE Transactions on automatic control*, 60(10): 2583–2598, 2015.

C. Rao. Information and accuracy attainable in the estimation of statistical parameters. kotz s & johnson nl (eds.), breakthroughs in statistics volume i: Foundations and basic theory, 235–248, 1945.

M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.

J. J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of statistics*, pages 416–431, 1983.

J. J. Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information theory*, 30(4):629–636, 1984.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, July 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

R. J. Solomonoff. A preliminary report on a general theory of inductive inference. United States Air Force, Office of Scientific Research, 1960.

R. J. Solomonoff. A formal theory of inductive inference. part i. *Information and control*, 7(1):1–22, 1964a.

R. J. Solomonoff. A formal theory of inductive inference. part ii. *Information and control*, 7(2):224–254, 1964b.

R. J. Solomonoff. Algorithmic probability: Theory and applications. In *Information theory and statistical learning*, pages 1–23. Springer, 2009.

J. K. Uhlmann. *Dynamic map building and localization: New theoretical foundations*. PhD thesis, University of Oxford Oxford, 1995.

C. Zhu, S. Han, H. Mao, and W. J. Dally. Trained ternary quantization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=S1_pAu9xl`.