Master Presentation Marius Hobbhahn

Start: 14:30

Fast Predictive Uncertainty for Classification with Bayesian Deep Networks

Marius Hobbhahn

30 June 2020

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning

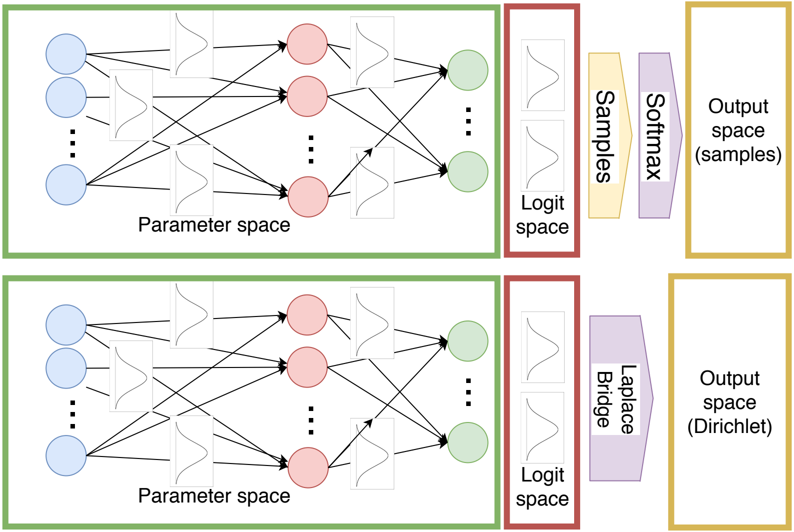
Motivation

Why do we need fast uncertainty in neural networks?

- safety-critical applications e.g. self-driving cars
- trade-off between accuracy and speed
- out-of-distribution detection

Context

What's our new contribution?



Theory

Let $\mathbf{x}$ be an n-dimensional continuous random variable with joint density function $p_{\mathbf{x}}$. If $\mathbf{y} = g(\mathbf{x})$, where g is a differentiable function, then $\mathbf{y}$ has density $p_{\mathbf{y}}$:

$$p_{\mathbf{y}}(\mathbf{y}) = p_{\mathbf{x}}\left(g^{-1}(\mathbf{y})\right) \left| \det \left[\frac{dg^{-1}(\mathbf{y})}{d\mathbf{y}} \right] \right| \tag{1}$$

where the differential is the Jacobian of the inverse of g evaluated at $\mathbf{y}$.

A new basis for the Dirichlet

The math

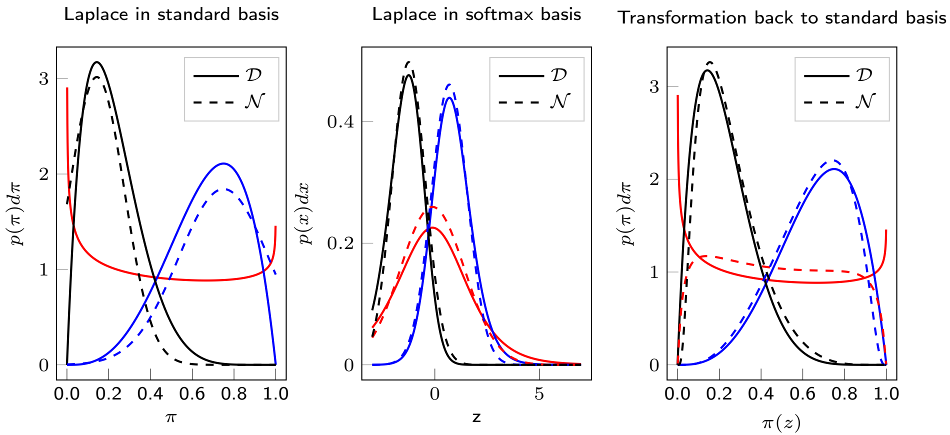
- $$\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) := \frac{1}{B(\alpha)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \tag{2}$$

- $$\pi_k(\mathbf{z}) := \frac{\exp(z_k)}{\sum_{l=1}^{K} \exp(z_l)}\,, \tag{3}$$

- $$\mathrm{Dir}_{\mathbf{z}}(\boldsymbol{\pi}(\mathbf{z})|\boldsymbol{\alpha}) := \frac{1}{B(\alpha)} \prod_{k=1}^{K} \pi_k(\mathbf{z})^{\alpha_k}\,, \tag{4}$$

A new basis for the Dirichlet

In pictures



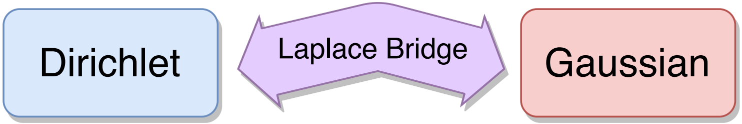
The Laplace Bridge

A bridge between the parameters of the Dirichlet and Gaussian

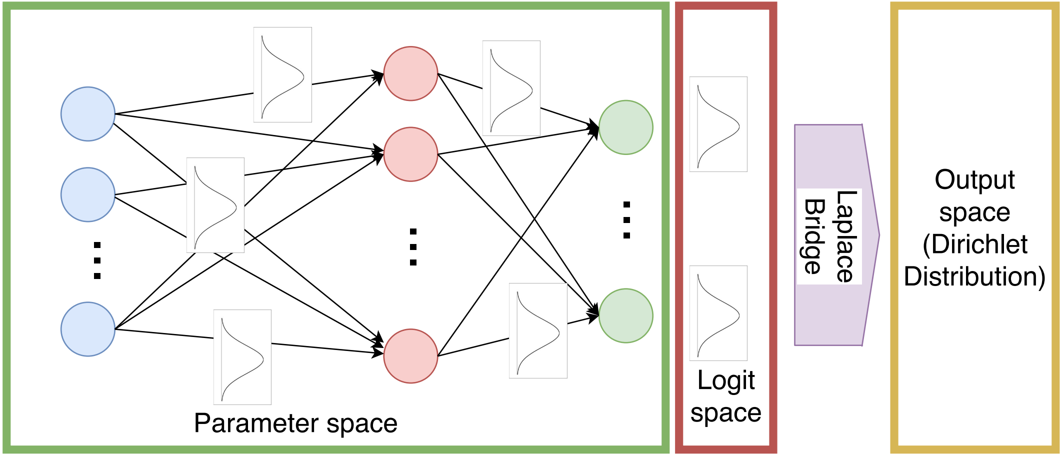
$$\alpha_k = \frac{1}{\Sigma_{kk}} \left(1 - \frac{2}{K} + \frac{e^{\mu_k}}{K^2} \sum_{l}^{K} e^{-\mu_l} \right) \tag{5}$$

$$\mu_k = \log \alpha_k - \frac{1}{K} \sum_{l=1}^{K} \log \alpha_l \tag{6}$$

$$\Sigma_{kl} = \delta_{kl} \frac{1}{\alpha_k} - \frac{1}{K} \left[\frac{1}{\alpha_k} + \frac{1}{\alpha_l} - \frac{1}{K} \sum_{u=1}^{K} \frac{1}{\alpha_u} \right] \tag{7}$$



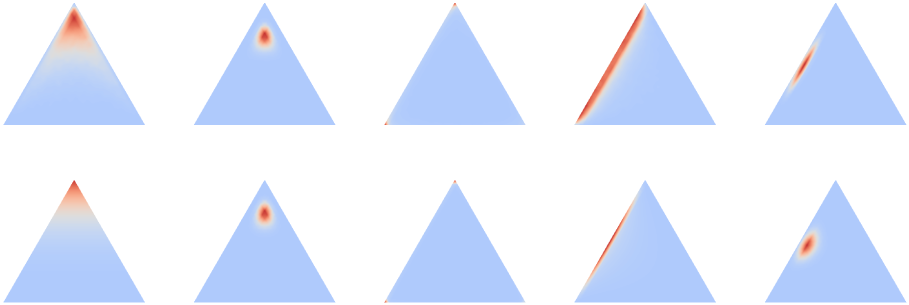
- The Dirichlet in the inverse softmax basis approximates a Gaussian
- Via the Laplace approximation in the transformed basis we can create a closed-form transformation $\alpha \to (\mu, \Sigma)$.
- We can also construct an inverse of this transformation $(\mu, \Sigma) \to \alpha$
- In total, we have a **fast** way to transform between the parameters of a Dirichlet and a Gaussian


Parameter space
Logit space
Laplace Bridge
Output space (Dirichlet Distribution)

Experiments

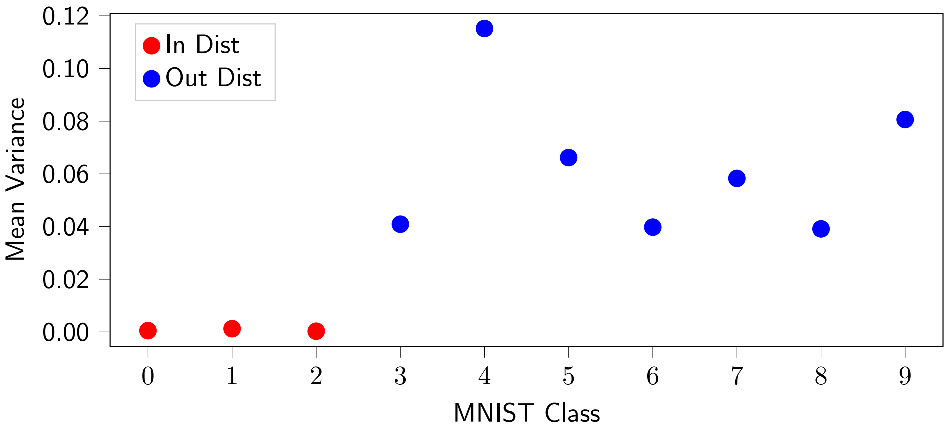
A sanity check

Samples from a 3D Gaussian + Softmax vs. Dirichlet



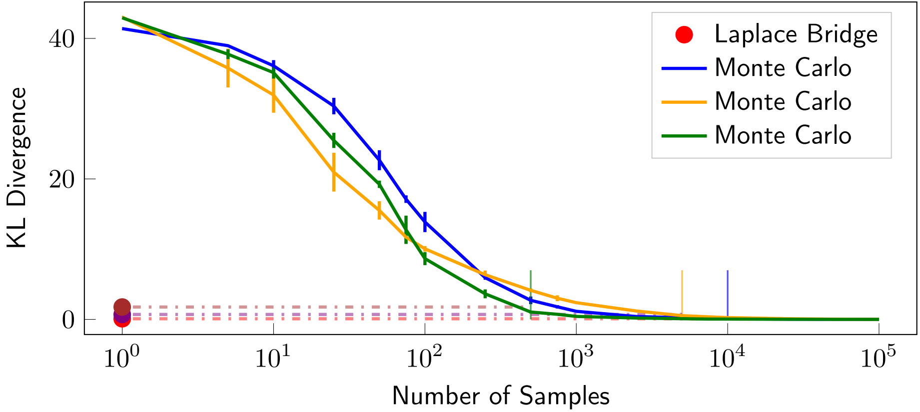
MNIST

Train on 0,1,2; test on 0-9



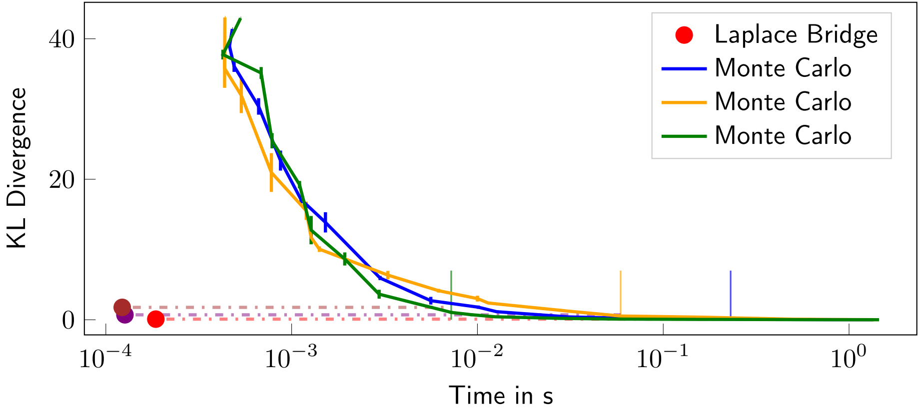
Speedtest - I

KL divergence vs. number of samples



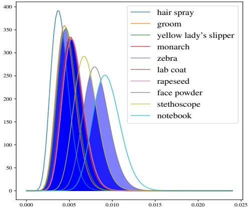
Speedtest - II

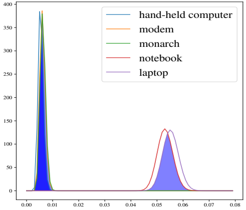
KL divergence vs. wall-clock time

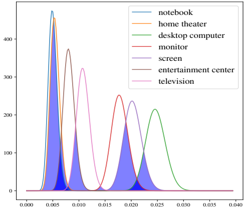


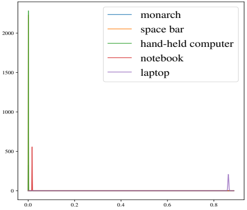
Imagenet

Using the properties of the Dirichlet - The marginal of a Dirichlet is a Dirichlet





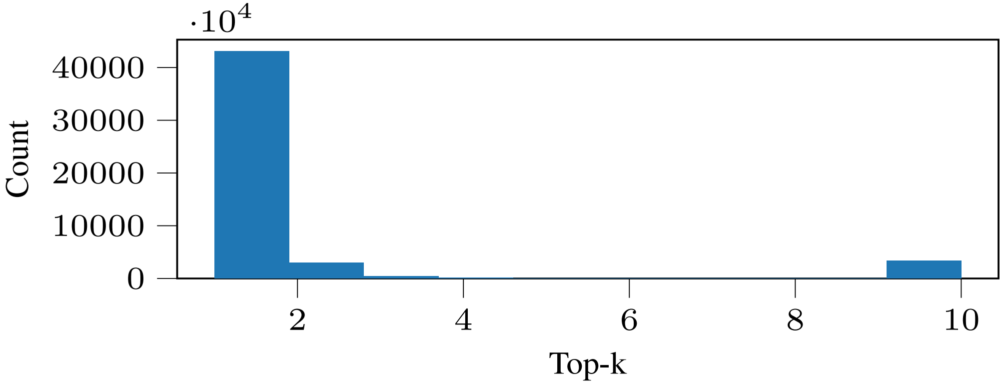




We can use the overlap of the distributions to create an uncertainty-aware top-k ranking.

Imagenet - II

How good is the flexible top-k ranking?



- The original top-1 accuracy of DenseNet on ImageNet is 0.744 and top-5 accuracy is 0.919
- The uncertainty-aware top-k accuracy is 0.797, where k is on average 1.688.

Out-of-distribution Detection

Looking at the numbers

Train	Test	Diag Sampling MMC ↓	Diag Sampling AUROC ↑	Diag LB MMC ↓	Diag LB AUROC ↑	KFAC Sampling MMC ↓	KFAC Sampling AUROC ↑	KFAC LB MMC↓	KFAC LB AUROC ↑	Time in s ↓ Sampling	Time in s ↓ LB
MNIST	MNIST	0.942 ± 0.007	-	**0.987** ± 0.000	-	-	-	-	-	26.8	**0.062**
MNIST	FMNIST	0.397 ± 0.001	0.992 ± 0.000	**0.363** ± 0.000	**0.996** ± 0.000	-	-	-	-	26.8	**0.062**
MNIST	notMNIST	**0.543** ± 0.000	0.960 ± 0.000	0.649 ± 0.000	**0.961** ± 0.000	-	-	-	-	50.3	**0.117**
MNIST	KMNIST	**0.513** ± 0.001	**0.974** ± 0.000	0.637 ± 0.000	0.973 ± 0.000	-	-	-	-	26.9	**0.062**
CIFAR-10	CIFAR-10	0.948 ± 0.000	-	**0.966** ± 0.000	-	0.857 ± 0.003	-	**0.966** ± 0.000	-	6.58	**0.017**
CIFAR-10	CIFAR-100	**0.708** ± 0.000	**0.889** ± 0.000	0.742 ± 0.000	0.866 ± 0.000	**0.562** ± 0.003	**0.880** ± 0.012	0.741 ± 0.000	0.866 ± 0.000	6.59	**0.016**
CIFAR-10	SVHN	**0.643**± 0.000	0.933 ± 0.000	0.647 ± 0.000	**0.934** ± 0.000	**0.484** ± 0.004	**0.939** ± 0.001	0.648 ± 0.003	0.934 ± 0.001	17.0	**0.040**
SVHN	SVHN	0.986 ± 0.000	-	**0.993** ± 0.000	-	0.947 ± 0.002	-	**0.993** ± 0.000	-	17.1	**0.042**
SVHN	CIFAR-100	0.595 ± 0.000	0.984 ± 0.000	**0.526** ± 0.000	**0.985** ± 0.000	**0.460** ± 0.004	**0.986** ± 0.001	0.527 ± 0.002	0.985 ± 0.000	6.62	**0.017**
SVHN	CIFAR-10	0.593 ± 0.000	0.984 ± 0.000	**0.520** ± 0.000	**0.987** ± 0.000	**0.458** ± 0.004	0.986 ± 0.001	0.520 ± 0.002	**0.987** ± 0.000	6.62	**0.017**
CIFAR-100	CIFAR-100	**0.762** ± 0.000	-	0.590 ± 0.000	-	0.404 ± 0.000	-	**0.593** ± 0.000	-	6.76	**0.016**
CIFAR-100	CIFAR-10	0.467 ± 0.000	0.788 ± 0.000	**0.206** ± 0.000	**0.791** ± 0.000	0.213 ± 0.000	0.788 ± 0.000	**0.209** ± 0.000	**0.791** ± 0.000	6.71	**0.017**
CIFAR-100	SVHN	0.461 ± 0.000	0.795 ± 0.000	**0.170** ± 0.000	**0.815** ± 0.000	0.180 ± 0.001	**0.838** ± 0.001	**0.173** ± 0.000	0.815 ± 0.000	17.3	**0.041**

- The Laplace Bridge seems to be have better MMC and AUROC compared to sampling from a diagonal Gaussian approximation
- The Laplace Bridge is as good as a KFAC approximation
- The Laplace Bridge is around 400 times faster on average

Conclusions

What can or can't the Laplace Bridge achieve in the context of BNNs?

- The Laplace Bridge improves an important part of Bayesian Neural Network inference for classification (fast & non-invasive)
- The Dirichlet distribution has some additional interesting use cases (e.g. the top-k ranking)
- It will not revolutionize BNNs; it is just one piece in the larger puzzle

Questions?

Future

The generalized Laplace Bridge

Looking at the larger pattern

- Similar “Bridges” can be found for all exponential families.
- Develop a general theoretically grounded framework for the general Laplace Bridge
- Compute KL-divergences in the different basis

Implications: (with a small error)

- All exponential families can be transformed to Gaussians
- All exponential families can be transformed to each other
- All exponential families are conjugate priors for each other

Backup

Backup

Laplace approximations of a neural network

$$p(c|x) = \mathcal{N}(x; f(x, w_{\text{MAP}}), J(x)^T H^{-1} J(x)) \quad (8)$$

- $f(x; w_{\text{MAP}})$ is the network output induced by the MAP estimate w_{MAP}.
- $J(x) = \frac{\partial f(x, w_{\textbf{MAP}})}{\partial w} \in \mathbb{R}^{K \times P}$ is the Jacobian of the network
- $H_{ij} = \frac{\partial^2 \mathcal{L}(f(x), y)}{\partial w_i \partial w_j} \in \mathbb{R}^{P \times P}$ its Hessian.
- K, P are the number of classes and parameters of the network respectively.

Proposition

Let $\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha})$ *be obtained via the Laplace Bridge from a Gaussian distribution* $\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *over* $\mathbb{R}^K$. *Then, for each* $k = 1, \dots, K$, *letting* $\alpha_{\neq k} := \sum_{l \neq k} \alpha_l$, *if*

$$\alpha_k > \frac{1}{4}\left(\sqrt{9\alpha_{\neq k}^2 + 10\alpha_{\neq k} + 1} - \alpha_{\neq k} - 1\right),$$

then the variance $\mathrm{Var}(\pi_k|\boldsymbol{\alpha})$ *of the* k*-th component of* $\boldsymbol{\pi}$ *is increasing in* $\boldsymbol{\Sigma}_{kk}$.

First, we consider the special case where $\boldsymbol{\pi}$ is confined to a $I-1$ dimensional subspace satisfying $\sum_i \boldsymbol{\pi}_i = c$. In this subspace we can represent $\boldsymbol{\pi}$ by an $I-1$ dimensional vector $\mathbf{a}$ such that

$$\pi_i = a_i \quad i, ..., I-1 \tag{9}$$

$$\pi_I = c - \sum_i^{I-1} a_i \tag{10}$$

and similarly we can represent $\mathbf{z}$ by an $I-1$ dimensional vector $\boldsymbol{\varrho}$:

$$z_i = \varrho_i \quad i, ..., I-1 \tag{11}$$

$$z_I = 1 - \sum_i^{I-1} \varrho_i \tag{12}$$

then we can find the density over $\boldsymbol{\varrho}$ (which is proportional to the required density over z) from the density over $\boldsymbol{\pi}$ (which is proportional to the given density over $\boldsymbol{\pi}$) by finding the determinant of the $(I-1) \times (I-1)$ Jacobian $\mathbf{J}$ given by

$$J_{ik} = \frac{\partial \varrho_i}{\partial a_i} = \sum_{j}^{I} \frac{\partial z_i}{\partial \pi_j} \frac{\partial \pi_j}{\partial a_k} \tag{13}$$

$$= \delta_{ik}\mathbf{z}_i - \mathbf{z}_i\mathbf{z}_k + \mathbf{z}_i\mathbf{z}_I = \mathbf{z}_i(\delta_{ik} - (\mathbf{z}_k - \mathbf{z}_I)) \tag{14}$$

We define two additional $I-1$ dimensional helper vectors $\mathbf{z}_k^+ := \mathbf{z}_k - \mathbf{z}_I$ and $n_k := 1$, and use $\det(I - xy^T) = 1 - x \cdot y$ from linear algebra. It follows that

$$\det J = \prod_{i=1}^{I-1} \mathbf{z}_i \times \det[I - n{\mathbf{z}^+}^T] \tag{15}$$

$$= \prod_{i=1}^{I-1} \mathbf{z}_i \times (1 - n \cdot \mathbf{z}^+) \tag{16}$$

$$= \prod_{i=1}^{I-1} \mathbf{z}_i \times \left(1 - \sum_k \mathbf{z}_k^+\right) = I \prod_{i=1}^{I} \mathbf{z}_i \tag{17}$$