AutoGP: Exploring the Capabilities and Limitations of Gaussian Process Models — Supplementary Material

Karl Krauth **Edwin V. Bonilla** **Kurt Cutajar** **Maurizio Filippone**
The University of New South Wales EURECOM
karl.krauth@gmail.com e.bonilla@unsw.edu.au {kurt.cutajar, maurizio.filippone}@eurecom.fr

1 DERIVATION OF LEAVE-ONE-OUT OBJECTIVE

In this section we derive an expression for the leave-one-out objective and show that this does not require training of N models. A similar derivation can be found in Vehtari et al. (2016). Let $\mathcal{D}_{\neg n} = \{\mathbf{X}_{\neg n}, \mathbf{y}_{\neg n}\}$ be the dataset resulting from removing observation n. Then our leave-one-out objective is given by:

$$\mathcal{L}_{\text{oo}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N} \log p(\mathbf{y}_n|\mathbf{x}_n, \mathcal{D}_{\neg n}, \boldsymbol{\theta}). \tag{1}$$

We now that the marginal posterior can be computed as:

$$p(\mathbf{f}_{n\cdot}|\mathcal{D}) = p(\mathbf{f}_{n\cdot}|\mathbf{X}_{\neg n}, \mathbf{y}_{\neg n}, \mathbf{x}_n, \mathbf{y}_n) = \frac{p(\mathbf{y}_n|\mathbf{f}_{n\cdot})p(\mathbf{f}_{n\cdot}|\mathbf{x}_n, \mathcal{D}_{\neg n})}{p(\mathbf{y}_n|\mathbf{x}_n, \mathcal{D}_{\neg n}, \boldsymbol{\theta})} \tag{2}$$

and re-arranging terms

$$\int p(\mathbf{f}_{n\cdot}|\mathbf{x}_n, \mathcal{D}_{\neg n}, \boldsymbol{\theta}) d\mathbf{f}_{n\cdot} = \int \frac{p(\mathbf{f}_{n\cdot}|\mathcal{D}, \boldsymbol{\theta})p(\mathbf{y}_n|\mathbf{x}_n, \mathcal{D}_{\neg n}, \boldsymbol{\theta})}{p(\mathbf{y}_n|\mathbf{f}_{n\cdot})} d\mathbf{f}_{n\cdot} \tag{3}$$

$$p(\mathbf{y}_n|\mathbf{x}_n, \mathcal{D}_{\neg n}, \boldsymbol{\theta}) = 1/\int \frac{p(\mathbf{f}_{n\cdot}|\mathcal{D}, \boldsymbol{\theta})}{p(\mathbf{y}_n|\mathbf{f}_{n\cdot})} d\mathbf{f}_{n\cdot} \tag{4}$$

$$\log p(\mathbf{y}_n|\mathbf{x}_n, \mathcal{D}_{\neg n}; \boldsymbol{\theta}) = -\log \int \frac{p(\mathbf{f}_{n\cdot}|\mathcal{D}, \boldsymbol{\theta})}{p(\mathbf{y}_n|\mathbf{f}_{n\cdot})} d\mathbf{f}_{n\cdot}, \tag{5}$$

and substituting this expression in Equation (1) we have

$$\mathcal{L}_{\text{oo}}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N} \log \int p(\mathbf{f}_{n\cdot}|\mathcal{D}, \boldsymbol{\theta}) \frac{1}{p(\mathbf{y}_n|\mathbf{f}_{n\cdot})}\, d\mathbf{f}_{n\cdot}. \tag{6}$$

We see that the objective only requires estimation of the marginal posterior $p(\mathbf{f}_{n\cdot}|\mathcal{D}, \boldsymbol{\theta})$, which we can approximate using variational inference, hence:

$$\mathcal{L}_{\text{oo}}(\boldsymbol{\theta}) \approx -\frac{1}{N}\sum_{n=1}^{N} \log \int q(\mathbf{f}_{n\cdot}|\mathcal{D}, \boldsymbol{\theta}) \frac{1}{p(\mathbf{y}_n|\mathbf{f}_{n\cdot})}\, d\mathbf{f}_{n\cdot}, \tag{7}$$

where $q(\mathbf{f}_{n\cdot}|\mathcal{D}, \boldsymbol{\theta})$ is our approximate variational posterior.

Table 1: The datasets used in the experiments and the corresponding models used. N_{train}, N_{test}, D are the number of training points, test points and input dimensions respectively.

Dataset	N_{train}	N_{test}	D	Model
SARCOS	$44,484$	$4,449$	21	GPRN
RECTANGLES-IMAGE	$12,000$	$50,000$	784	Binary classification
MNIST	$60,000$	$10,000$	784	Multi-class classification
CIFAR10	$50,000$	$10,000$	3072	Multi-class classification
MNIST8M	8.1M	$10,000$	784	Multi-class classification

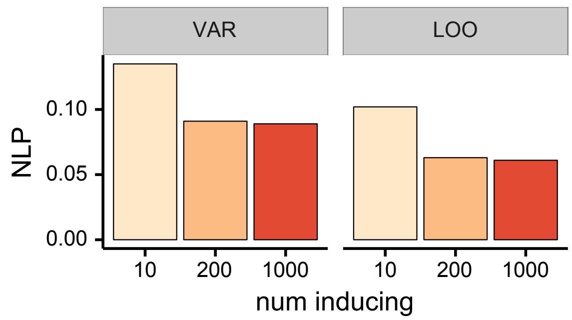

Figure 1: NLP for multiclass classification using a softmax likelihood model on the MNIST dataset. VAR shows the performance of AutoGP where all parameters are learned using only the variational objective $\widehat{\mathcal{L}}_{\text{elbo}}$, while LOO represents the performance of AutoGP when hyperparameters are learned using the leave-one-out objective $\widehat{\mathcal{L}}_{\text{oo}}$.

2 ADDITIONAL DETAILS OF EXPERIMENTS

2.1 EXPERIMENTAL SET-UP

The datasets used are described in Table 1. We trained our model stochastically using the RMSProp optimizer provided by TensorFlow (Abadi et al., 2015) with a learning rate of 0.003 and mini-batches of size 1000. We initialized inducing point locations by using the k-means clustering algorithm, and initialized the posterior mean to a zero vector, and the posterior covariances to identity matrices. When jointly optimizing $\widehat{\mathcal{L}}_{\text{oo}}$ and $\widehat{\mathcal{L}}_{\text{elbo}}$, we alternated between optimizing each objective for 100 epochs. Unless otherwise specified we used 100 Monte-Carlo samples to estimate the expected log likelihood term.

All timed experiments were performed on a machine with an Intel(R) Core(TM) i5-4460 CPU, 24GB of DDR3 RAM, and a GeForce GTX1070 GPU with TensorFlow 0.10rc.

2.2 ADDITIONAL RESULTS

Figure 1 shows the NLP for our evaluation of the LOO-CV-based hyperparameter learning. As with the error rates described in the main text, the NLP obtained with LOO-CV are significantly better than those obtained with a purely variational approach.

References

Martín Abadi, Ashish Agarwal, Paul Barham, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

Aki Vehtari, Tommi Mononen, Ville Tolvanen, Tuomas Sivula, and Ole Winther. Bayesian leave-one-out cross-validation approximations for Gaussian latent variable models. *Journal of Machine Learning Research*, 17(103): 1–38, 2016.