
Towards Improved Learning in Gaussian Processes: The Best of Two Worlds

Rui Li
Aalto University
rui.li@aalto.fi

ST John
Aalto University & FCAI
ti.john@aalto.fi

Arno Solin
Aalto University
arno.solin@aalto.fi

Abstract

Gaussian process training decomposes into inference of the (approximate) posterior and learning of the hyperparameters. For non-Gaussian (non-conjugate) likelihoods, two common choices for approximate inference are Expectation Propagation (EP) and Variational Inference (VI), which have complementary strengths and weaknesses. While VI's lower bound to the marginal likelihood is a suitable objective for inferring the approximate posterior, it does not automatically imply it is a good learning objective for hyperparameter optimization. We design a hybrid training procedure where the inference leverages conjugate-computation VI and the learning uses an EP-like marginal likelihood approximation. We empirically demonstrate on binary classification that this provides a good learning objective and generalizes better.

1 Introduction

Gaussian processes (GPs, [17]) provide a principled way of incorporating prior knowledge over functions and quantifying uncertainty. GPs are widely used in a range of applications such as robotics [4], numerics [8], geostatistics [5], and optimization [7]. We focus on the non-conjugate case where the training of GP decomposes into two parts: *inferring* the approximate posterior and *learning* the hyperparameters of the model. Variational Inference (VI, [16]) and Expectation Propagation (EP, [12]) are two commonly used approximate inference methods for non-conjugate GP models, which have complementary advantages: VI optimizes a lower bound of the marginal likelihood, and is thus easy to implement, straightforward to use, and guaranteed to converge, but is known to underestimate variance [3, 2]. EP on the other hand requires implementation-wise tuning per likelihood, and thus is not guaranteed to converge [19], but provides a good approximation for the marginal likelihood [11, 15].

When it comes to model performance and generalization to unseen test data, the learning of hyperparameters plays a crucial role. However, it is not entirely clear what is a good learning objective for hyperparameter optimization. This paper fuses the complementary advantages of EP and VI in inference and learning in non-conjugate GP models. We build on [10] that provides a link between VI and EP, where the approximate posterior obtained through VI has exactly the same structure as the approximate posterior of EP, and thus provides a plug-in method for obtaining an EP-like marginal likelihood estimate from the VI approximate posterior. Based on this bridge, we propose a hybrid training procedure of GPs where the advantages of VI and EP are combined. The main contributions of this work are: (i) We empirically compare the quality of approximated marginal likelihood in EP and VI, and show EP provides better approximation compared with VI (Fig. 1); (ii) We evaluate the new training procedure in binary classification and demonstrate that it will result in better generalization.

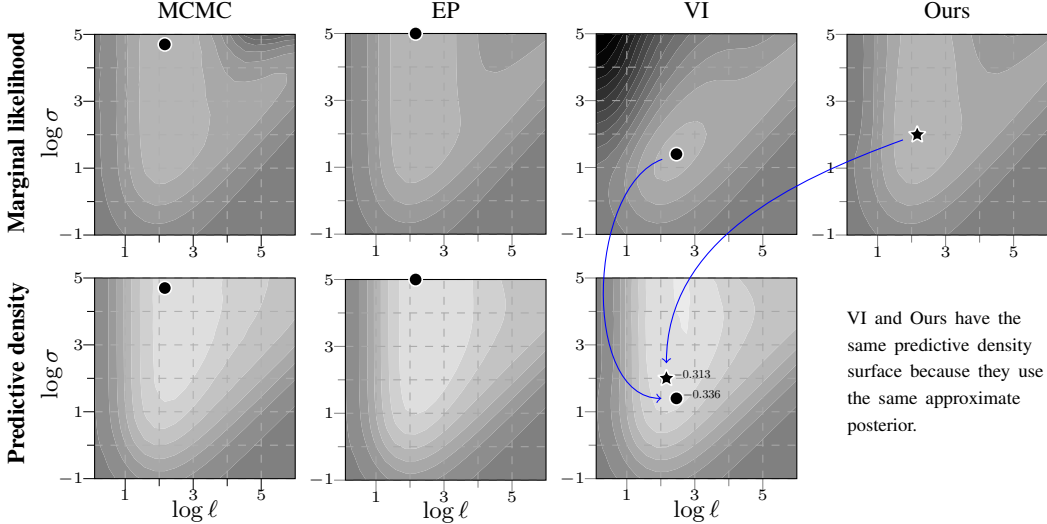


Figure 1: Log marginal likelihood and predictive density surfaces (normalized by n) for the SONAR data set by varying kernel magnitude σ and lengthscale ℓ . The colour scale is the same in all plots: -1.2 -0.2 . Black markers show optimal hyperparameter locations. EP and EP-like marginal likelihood estimation (Ours) match the MCMC baseline better than VI and result in better prediction.

2 Background

Gaussian processes are distributions over functions, $f(\cdot) \sim \mathcal{GP}(\mu(\cdot), \kappa(\cdot, \cdot))$, fully specified by the mean function $\mu(\cdot)$ and the covariance function $\kappa(\cdot, \cdot)$. Given a data set $\mathcal{D} = (\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ of input-output pairs, the posterior is $p(\mathbf{f} | \mathbf{y}; \boldsymbol{\theta}) \propto p(\mathbf{y} | \mathbf{f}; \boldsymbol{\theta}) p(\mathbf{f}; \boldsymbol{\theta})$, where \mathbf{f} is the vector of function values evaluated at the inputs, $p(\mathbf{y} | \mathbf{f}; \boldsymbol{\theta}) = \prod_{i=1}^n p(y_i | f_i; \boldsymbol{\theta})$ is the likelihood, and $\boldsymbol{\theta}$ denotes the hyperparameters of GP prior and likelihood. For a Gaussian likelihood, the posterior is available in closed form; when the likelihood is non-conjugate, we need to resort to approximate inference.

Expectation Propagation (EP, [12]) seeks to approximate each likelihood term $p(y_i | f_i; \boldsymbol{\theta})$ with a *site* function $t_i(f_i; \boldsymbol{\zeta}_i)$. For inference in GPs, the posterior $p(\mathbf{f} | \mathbf{y}; \boldsymbol{\theta})$ is approximated by

$$q(\mathbf{f}; \boldsymbol{\theta}, \boldsymbol{\zeta}) = \frac{1}{Z} p(\mathbf{f}; \boldsymbol{\theta}) \prod_{i=1}^n t_i(f_i; \boldsymbol{\zeta}_i), \quad (1)$$

where $t_i(f_i; \boldsymbol{\zeta}_i)$ is a Gaussian distribution and Z^{-1} is the normalization term. EP approximately minimizes $D_{\text{KL}}[p(\mathbf{f} | \mathbf{y}; \boldsymbol{\theta}) \| q(\mathbf{f}; \boldsymbol{\theta}, \boldsymbol{\zeta})]$. During inference, EP tunes $\boldsymbol{\zeta}_i$ one by one through

$$\arg \min_{\boldsymbol{\zeta}_i} D_{\text{KL}}[p(y_i | f_i; \boldsymbol{\theta}) p(\mathbf{f}; \boldsymbol{\theta}) \prod_{j \neq i} t_j(f_j; \boldsymbol{\zeta}_j) \| t_i(f_i; \boldsymbol{\zeta}_i) p(\mathbf{f}; \boldsymbol{\theta}) \prod_{j \neq i} t_j(f_j; \boldsymbol{\zeta}_j)]. \quad (2)$$

Learning under the GP paradigm in machine learning typically amounts to finding point estimates for the parameters $\boldsymbol{\theta}$ in the likelihood and kernel by optimizing w.r.t. the log marginal likelihood: $\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta}} \log p(\mathbf{y}; \boldsymbol{\theta})$. In EP, the log marginal likelihood is directly approximated as

$$\log p(\mathbf{y}; \boldsymbol{\theta}) \approx \mathcal{L}_{\text{EP}}(\boldsymbol{\zeta}, \boldsymbol{\theta}) = \log \int p(\mathbf{f}; \boldsymbol{\theta}) \prod_{i=1}^n t_i(f_i; \boldsymbol{\zeta}_i) d\mathbf{f}. \quad (3)$$

The direct approximation in Eq. (3) is known to lead to a good objective for learning hyperparameters [9]. However, the iterative solution of Eq. (2) can be numerically unstable, and is not guaranteed to converge in the general case (see [19] for details).

Variational Inference (VI, [16]) approximates the GP posterior $p(\mathbf{f} | \mathbf{y}; \boldsymbol{\theta})$ with a Gaussian distribution $q(\mathbf{f}; \boldsymbol{\xi}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$ with variational parameters $\boldsymbol{\xi} = \{\mathbf{m}, \mathbf{S}\}$. VI minimizes the reverse KL $D_{\text{KL}}[q(\mathbf{f}; \boldsymbol{\xi}) \| p(\mathbf{f} | \mathbf{y}; \boldsymbol{\theta})]$ (cf. the EP section above) by maximizing the following evidence lower bound (ELBO) to the log marginal likelihood:

$$\log p(\mathbf{y}; \boldsymbol{\theta}) \geq \mathcal{L}_{\text{VI}}(\boldsymbol{\xi}, \boldsymbol{\theta}) = -D_{\text{KL}}[q(\mathbf{f}; \boldsymbol{\xi}) \| p(\mathbf{f}; \boldsymbol{\theta})] + \sum_{i=1}^n \mathbb{E}_{q(f_i; \boldsymbol{\xi}_i)}[\log p(y_i | f_i; \boldsymbol{\theta})], \quad (4)$$

with respect to variational parameters $\boldsymbol{\xi}$. During inference, as VI optimizes a lower bound of the marginal likelihood, it is guaranteed to converge. In practice, the same lower bound $\mathcal{L}_{\text{VI}}(\boldsymbol{\xi}, \boldsymbol{\theta})$ is used to jointly optimize the variational parameters and a point estimate of the hyperparameters. However, using the lower bound tends to result in biased hyperparameters [11, 15, 3].

Khan and Lin [10] introduced a dual parameterization which enables fast natural gradient descent (NGD) on the variational distribution using $\mathcal{L}_{\text{VI}}(\boldsymbol{\xi}, \boldsymbol{\theta})$ as the objective. The Gaussian distribution is part of the exponential family, which means $q(\mathbf{f}) = \text{N}(\mathbf{m}, \mathbf{S}) = \exp(\boldsymbol{\eta}^\top \mathbf{T}(\mathbf{f}) - a(\boldsymbol{\eta}))$ where $\boldsymbol{\eta} = [\mathbf{S}^{-1}\mathbf{m}, -\frac{1}{2}\mathbf{S}^{-1}]$, $\mathbf{T}(\mathbf{f}) = [\mathbf{f}, \mathbf{f}\mathbf{f}^\top]$, and $\exp(-a(\boldsymbol{\eta}))$ is the normalization term. Instead of using the mean-covariance parameterization $\boldsymbol{\xi} = \{\mathbf{m}, \mathbf{S}\}$ of the Gaussian distribution, we can also use the natural parameters $\boldsymbol{\eta} = \{\mathbf{S}^{-1}\mathbf{m}, -\frac{1}{2}\mathbf{S}^{-1}\}$ and expectation parameters $\boldsymbol{\mu} = \mathbb{E}_{q(\mathbf{f})}[\mathbf{T}(\mathbf{f})] = \{\mathbf{m}, \mathbf{S} + \mathbf{m}\mathbf{m}^\top\}$. Khan and Lin [10] point out that NGD in the natural parameters space has the same computational cost as gradient descent, and the resulting approximate posterior is

$$q(\mathbf{f}; \boldsymbol{\lambda}, \boldsymbol{\theta}) \propto p(\mathbf{f}; \boldsymbol{\theta}) \prod_{i=1}^n \underbrace{\exp(\boldsymbol{\lambda}_i, \mathbf{T}(f_i))}_{t_i(f_i; \boldsymbol{\lambda}_i)}, \text{ where } \boldsymbol{\lambda}_i = \nabla_{\boldsymbol{\mu}_i} \mathbb{E}_{q(f_i; \boldsymbol{\lambda}_i, \boldsymbol{\theta})} [\log p(y_i | f_i; \boldsymbol{\theta})], \quad (5)$$

whose natural parameters $\boldsymbol{\eta} = \boldsymbol{\lambda}_0 + \boldsymbol{\lambda}$ where $\boldsymbol{\lambda}_0 = (\mathbf{0}, -\frac{1}{2}\kappa(\mathbf{X}, \mathbf{X})^{-1})$ are the natural parameters of the prior $p(\mathbf{f}; \boldsymbol{\theta})$ and $\boldsymbol{\lambda}$ are the parameters of the likelihood approximation term $t(\mathbf{f})$. Then, we could also parameterize the approximate posterior with $\boldsymbol{\lambda}$, to which we refer as $\boldsymbol{\lambda}$ parameterization. Eq. (5) has the same form as the EP approximate posterior in Eq. (1). This links EP with VI (see [1] for details), which is the starting point for our proposed learning objective.

3 Methods

VI is the de facto way for approximate inference in GPs, and the de facto method for learning is jointly optimizing the ELBO (cf. Eq. (4)) with respect to the variational parameters $\boldsymbol{\xi}$ and the hyperparameters $\boldsymbol{\theta}$. We separate learning from inference by employing a Variational Expectation-Maximization (VEM) procedure similar to Adam et al. [1]. They alternate between optimizing the variational distribution in the $\boldsymbol{\lambda}$ parameterization and $\boldsymbol{\theta}$ using the following steps at the t^{th} iteration:

$$\begin{aligned} \text{E-step (inference): } & \boldsymbol{\lambda}^{(t+1)} \leftarrow \arg \max_{\boldsymbol{\lambda}} \mathcal{L}_{\text{VI}}(\boldsymbol{\lambda}, \boldsymbol{\theta}^{(t)}), \\ \text{M-step (learning): } & \boldsymbol{\theta}^{(t+1)} \leftarrow \arg \max_{\boldsymbol{\theta}} \mathcal{L}_{\text{VI}}(\boldsymbol{\lambda}^{(t+1)}, \boldsymbol{\theta}). \end{aligned} \quad (6)$$

The objective for both inference and learning steps is

$$\mathcal{L}_{\text{VI}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = -\text{D}_{\text{KL}}[q(\mathbf{f}; \boldsymbol{\lambda}, \boldsymbol{\theta}) \| p(\mathbf{f} | \mathbf{y}; \boldsymbol{\theta})] + \sum_{i=1}^n \mathbb{E}_{q(f_i; \boldsymbol{\lambda}_i, \boldsymbol{\theta})} [\log p(y_i | f_i; \boldsymbol{\theta})], \quad (7)$$

and the approximate posterior is formed as a product of the prior and Gaussian sites $t_i(f_i; \boldsymbol{\lambda}_i)$ just as in EP (Eq. (1)) [10]. This allows us to calculate an EP-like estimate of the log marginal likelihood by plugging $\boldsymbol{\lambda}_i$ from Eq. (5) into ζ_i in Eq. (3):

$$\mathcal{L}_{\text{EP}}(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \log \int p(\mathbf{f}; \boldsymbol{\theta}) \prod_{i=1}^n t_i(f_i; \boldsymbol{\lambda}_i) \text{d}\mathbf{f}. \quad (8)$$

As EP tends to provide a better estimation of the marginal likelihood [11, 15] instead of using Eq. (7) as the learning objective for hyperparameters, we propose to use Eq. (8).

4 Experiments

We consider binary classification with a Bernoulli likelihood. We use four common small and mid-sized classification data sets such that there is no need for sparse approximation. The exact details of the data sets can be found in App. A.3. In all experiments we use the isotropic Matérn- $5/2$ kernel [17] where the hyperparameters are lengthscale ℓ and kernel magnitude σ .

Quality of Marginal Likelihood Approximations We compare the marginal likelihood approximations of VI, EP and our EP-like VI. As the gold-standard baseline we use Markov Chain Monte Carlo (MCMC). Following Kuss and Rasmussen [11] and Nickisch and Rasmussen [15], we use Annealed Importance Sampling (AIS, [14]) to obtain an MCMC estimate of the marginal likelihood. The implementation details can be found in App. A.1. For each method, we estimate log marginal likelihood on a 21×21 grid of values for the log hyperparameters $\log \boldsymbol{\theta} = (\log \ell, \log \sigma)$ based on one single fold. We compare different methods by plotting the contour shapes on the hyperparameters grid. As shown in Fig. 1 on SONAR and Fig. 2 on IONOSPHERE (see Figs. 3 and 4 in App. A.2 for USPS and DIABETES), the marginal likelihood estimation of EP closely matches the MCMC baseline,

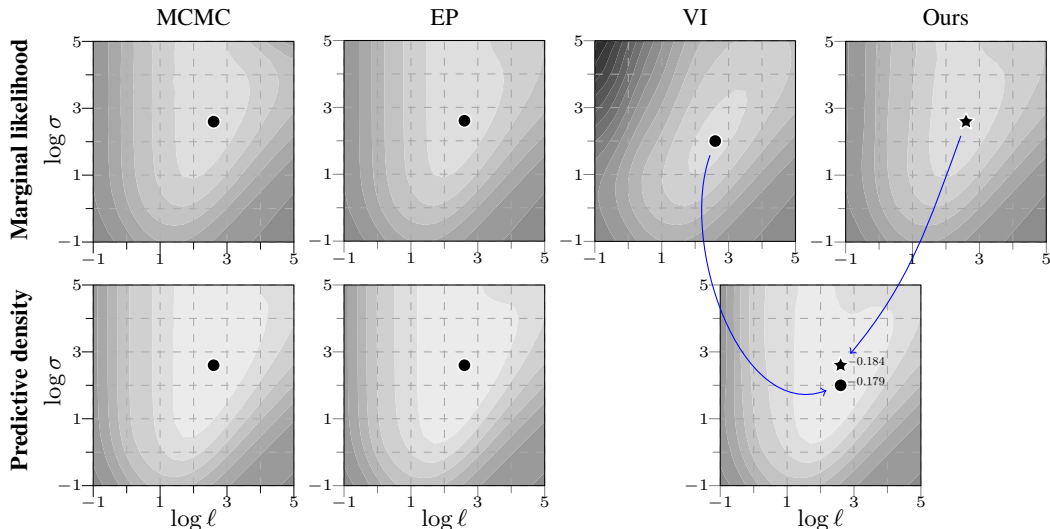



Figure 2: Log marginal likelihood surfaces for the IONOSPHERE data set. The colour scale is the same in all plots: -1.4  -0.1 (normalized by n). Refer to Fig. 1 for details.

whereas that of VI looks clearly different. Notably, when we estimate the marginal likelihood by plugging the site parameters of VI into the marginal likelihood estimation of EP, the contour shapes become much closer to the MCMC result, which means we have an improved marginal likelihood estimation by using the EP-like marginal likelihood estimation from site parameters of VI. We plot the maximal value of the estimated log marginal likelihood and notice that EP and EP-like VI (Ours) are closer to the MCMC result.

Evaluation on Classification Tasks The log marginal likelihood is a surrogate to the generalization ability of the model to unseen data. To explicitly evaluate the generalization ability of our hybrid training procedure, we compare test set accuracy and log predictive density against VI (following the setup of [1]). Our training procedure uses the EP-like marginal likelihood estimate as learning objective while VI uses ELBO. We perform 5-fold cross-validation and the performance on the test set is given in Table 1 (training details can be found in App. A.4). Our training procedure outperforms VI on SONAR and USPS and gives very similar performance on IONOSPHERE and DIABETES. This demonstrates that an improved learning objective can give us models that generalize better.

Table 1: Test set accuracy and log predictive density on different data sets (mean \pm standard deviation). Higher is better. Results that are statistically significantly different under a paired t -test ($p = 0.05$) are **bolded**.

| | Accuracy | | Log Predictive Density | |
|----------------|-------------------|-------------------------------------|------------------------|--------------------------------------|
| | VI | Ours | VI | Ours |
| IONOSPHERE [6] | 0.940 ± 0.016 | 0.946 ± 0.016 | -0.179 ± 0.023 | -0.176 ± 0.023 |
| SONAR [6] | 0.836 ± 0.036 | 0.860 ± 0.034 | -0.353 ± 0.013 | -0.340 ± 0.015 |
| DIABETES [6] | 0.783 ± 0.015 | 0.781 ± 0.013 | -0.473 ± 0.030 | -0.473 ± 0.030 |
| USPS [17] | 0.974 ± 0.010 | 0.974 ± 0.010 | -0.080 ± 0.011 | -0.077 ± 0.011 |

5 Conclusion

The training of GP models decomposes into inference and learning. In the non-conjugate case EP and VI are two widely used approximate inference methods. In this paper, we study the common choice of the learning objective in VI and empirically demonstrate that EP has a better learning objective. Based on this observation and conjugate-computation VI (CVI, [10]) which provides a bridge between VI and EP, we design a hybrid training procedure which has the benefits of VI for inference and the benefits of EP for learning—without any added computational complexity. We evaluate the hybrid training procedure on binary classification tasks and demonstrate that it provides a good learning objective and generalizes better.

References

- [1] Vincent Adam, Paul E. Chang, Mohammad Emtiyaz E. Khan, and Arno Solin. Dual parameterization of sparse variational Gaussian processes. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, volume 34, pages 11474–11486, 2021.
- [2] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859–877, 2017.
- [3] Thang D. Bui, Josiah Yan, and Richard E. Turner. A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation. *Journal of Machine Learning Research*, 18:3649–3720, 2017.
- [4] Marc P. Deisenroth and Carl E. Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, pages 465–472, 2011.
- [5] Peter J. Diggle and Paulo J. Ribeiro. *Model-based Geostatistics*. Springer New York, 2007.
- [6] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- [7] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2022.
- [8] Philipp Hennig, Michael A. Osborne, and Hans P. Kersting. *Probabilistic Numerics: Computation as Machine Learning*. Cambridge University Press, 2022.
- [9] Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. Robust Gaussian process regression with a Student- t likelihood. *Journal of Machine Learning Research*, 12:3227–3257, 2011.
- [10] Mohammad Emtiyaz E. Khan and Wu Lin. Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, pages 878–887, 2017.
- [11] Malte Kuss and Carl E. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.
- [12] Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI)*, Proceedings of Machine Learning Research, pages 362–369, 2001.
- [13] Iain Murray, Ryan Adams, and David MacKay. Elliptical slice sampling. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9 of *Proceedings of Machine Learning Research*, pages 541–548, 2010.
- [14] Radford M. Neal. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.
- [15] Hannes Nickisch and Carl E. Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9:2035–2078, 2008.
- [16] Manfred Opper and Cédric Archambeau. The variational Gaussian approximation revisited. *Neural Computation*, 21(3):786–92, 2009.
- [17] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [18] Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (GPML) toolbox. *Journal of Machine Learning Research (JMLR)*, 11:3011–3015, 2010.
- [19] Aki Vehtari, Andrew Gelman, Tuomas Sivula, Pasi Jylänki, Dustin Tran, Swupnil Sahai, Paul Blomstedt, John P. Cunningham, David Schiminovich, and Christian P. Robert. Expectation propagation as a way of life. *Journal of Machine Learning Research*, 21:1–53, 2020.

A Appendix

A.1 Markov Chain Monte Carlo Baseline

As in previous work [11, 15], we use an MCMC approach as the gold-standard baseline for marginal likelihood estimation. Here we recapitulate their annealed importance sampling approach, which defines a sequence of $t = 0, 1, \dots, T$ steps: $Z_t = \int p(\mathbf{y} | \mathbf{f}; \boldsymbol{\theta})^{\tau(t)} p(\mathbf{f}; \boldsymbol{\theta}) d\mathbf{f}$, where $\tau(t) = (t/T)^4$ (such that $\tau(0) = 0$ and $\tau(T) = 1$). The marginal likelihood can be rewritten as

$$p(\mathbf{y}; \boldsymbol{\theta}) = \frac{Z_T}{Z_0} = \frac{Z_T}{Z_{T-1}} \frac{Z_{T-1}}{Z_{T-2}} \dots \frac{Z_1}{Z_0}, \quad (9)$$

where $\frac{Z_t}{Z_{t-1}}$ is approximated by importance sampling using samples from $q_t(\mathbf{f}) \propto p(\mathbf{y} | \mathbf{f}; \boldsymbol{\theta})^{\tau(t-1)} p(\mathbf{f}; \boldsymbol{\theta})$:

$$\begin{aligned} \frac{Z_t}{Z_{t-1}} &= \frac{\int p(\mathbf{y} | \mathbf{f}; \boldsymbol{\theta})^{\tau(t)} p(\mathbf{f}; \boldsymbol{\theta}) d\mathbf{f}}{Z_{t-1}} = \int \frac{p(\mathbf{y} | \mathbf{f}; \boldsymbol{\theta})^{\tau(t)} p(\mathbf{y} | \mathbf{f}; \boldsymbol{\theta})^{\tau(t-1)} p(\mathbf{f}; \boldsymbol{\theta})}{p(\mathbf{y} | \mathbf{f}; \boldsymbol{\theta})^{\tau(t-1)} Z_{t-1}} d\mathbf{f} \\ &\approx \frac{1}{S} \sum_{s=1}^S p(\mathbf{y} | \mathbf{f}_t^{(s)}; \boldsymbol{\theta})^{\tau(t) - \tau(t-1)}, \quad \text{where } \mathbf{f}_t^{(s)} \sim \frac{p(\mathbf{y} | \mathbf{f}; \boldsymbol{\theta})^{\tau(t-1)} p(\mathbf{f}; \boldsymbol{\theta})}{Z_{t-1}}. \end{aligned} \quad (10)$$

By using a single sample $S = 1$ and a large number of steps T , the estimation of log marginal likelihood can be written as

$$\log p(\mathbf{y}; \boldsymbol{\theta}) = \sum_{t=1}^T \log \frac{Z_t}{Z_{t-1}} \approx \sum_{t=1}^T (\tau(t) - \tau(t-1)) \log p(\mathbf{y} | \mathbf{f}_t; \boldsymbol{\theta}). \quad (11)$$

Following [11], we set $T = 8000$ and combine three estimates of log marginal likelihood by their geometric mean. We use the implementation in GPML toolbox [18] and use Elliptical Slice Sampling [13] to sample $\mathbf{f}_t^{(s)}$.

A.2 Log Marginal Likelihood on USPS and DIABETES Data Sets

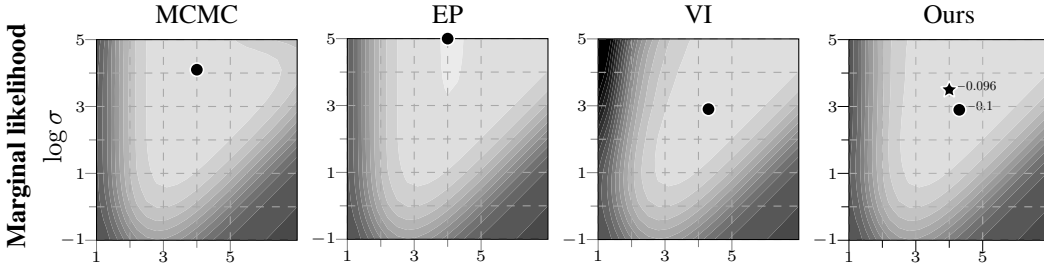


Figure 3: Log marginal likelihood on USPS data set. Text annotations are the log predictive density at the corresponding hyperparameter location.

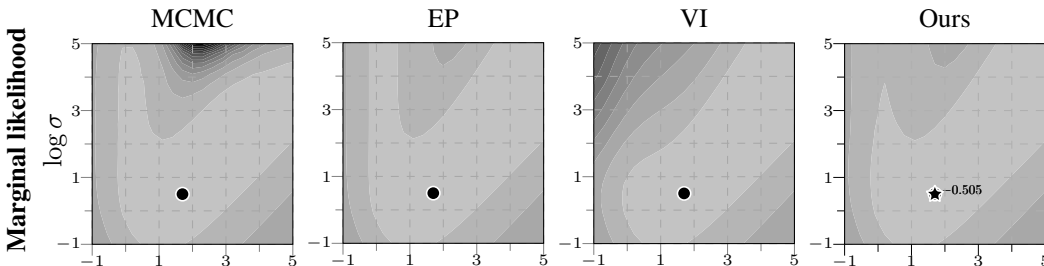


Figure 4: Log marginal likelihood on DIABETES data set. Text annotations are the log predictive density at the corresponding hyperparameter location.

A.3 Data Sets

The details of data sets we used in experiments are given in [Table 2](#), where n is the number of data points and d is the dimension of each data point.

Table 2: Details of data sets.

| | n | d | Brief description of problem domain |
|----------------|------|-----|---|
| IONOSPHERE [6] | 351 | 34 | Classification of radar returns from the ionosphere |
| SONAR [6] | 208 | 60 | Sonar signals returned by a metal or rock cylinder |
| DIABETES [6] | 768 | 8 | Predict the outcome on diabetes experiment |
| USPS [17] | 1540 | 256 | Binary sub-problem of the USPS handwritten digit data set |

A.4 Training Details

We initialize lengthscale $\ell = 1$ and magnitude $\sigma = 1$. For all data sets, both E-step and M-step are composed of 20 iterations. In the E-step we use natural gradient descent and set the learning rate to 0.1. In the M-step we use gradient descent and set the learning rate to 0.001.