Function-space Inference with Sparse Implicit Processes

Simón Rodríguez Santana[1]
Bryan Zaldivar[2]
Daniel Hernández-Lobato[3]

[1] Instituto de Ciencias Matemáticas (ICMAT-CSIC)
[2] Instituto de Física Corpuscular, Universidad de Valencia y CSIC
[3] Escuela Politécnica Superior, Universidad Autónoma de Madrid

Estimating the uncertainty of the predictions

Modern *ML* (*e.g.* NNs) $\rightarrow$ **point-wise predictions**

Info. on the **uncertainty of the predictions** $\rightarrow$ **Bayesian formulation**

$$\text{Posterior dist.} \qquad p(\mathbf{w}|\mathcal{D}) = p(\mathbf{w})p(\mathcal{D}|\mathbf{w})/p(\mathcal{D})$$

$$\text{Predictive dist.} \qquad p(y|\mathcal{D}, x) = \int p(y|\mathbf{w}, x)\, p(\mathbf{w}|\mathcal{D})\, \mathrm{d}\mathbf{w}$$

Estimating the uncertainty of the predictions

Modern *ML* (*e.g.* NNs) $\rightarrow$ **point-wise predictions**

Info. on the **uncertainty of the predictions** $\rightarrow$ **Bayesian formulation**

Posterior dist.	$p(\mathbf{w}\vert\mathcal{D}) = p(\mathbf{w})p(\mathcal{D}\vert\mathbf{w})/p(\mathcal{D})$
Predictive dist.	$p(y\vert\mathcal{D},x) = \int p(y\vert\mathbf{w},x)\, p(\mathbf{w}\vert\mathcal{D})\, \mathrm{d}\mathbf{w}$

$p(\mathcal{D})$ intractable! $\Rightarrow$ *approximate solutions* s.a. *MCMC*-based techinques, *VI*, *EP*, *AVB*, etc.

$\Rightarrow$ Inference with finite set of parameters (*e.g.* neurons in BNNs)

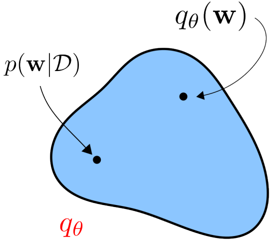
Variational Inference

VI $\rightarrow$ Parametric q to **approximate** target (intractable) posterior p

Evidence Lower Bound (ELBO):

$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_q[\log p(\mathbf{y}_i|\mathbf{W}, \mathbf{x}_i)] - \text{KL}(q|\text{prior})$$

- Monte Carlo and mini-batches!
- Closed-form solution if p and q are Gaussian!



If $p(\mathbf{w}|\mathcal{D}) \in q_\theta$,
good
approximation!

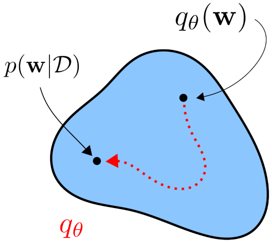
Variational Inference

VI $\rightarrow$ Parametric q to **approximate** target (intractable) posterior p

Evidence Lower Bound (ELBO):

$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_q[\log p(\mathbf{y}_i|\mathbf{W}, \mathbf{x}_i)] - \mathrm{KL}(q|\text{prior})$$

- Monte Carlo and mini-batches!
- Closed-form solution if p and q are Gaussian!



If $p(\mathbf{w}|\mathcal{D}) \in q_\theta$, good approximation!

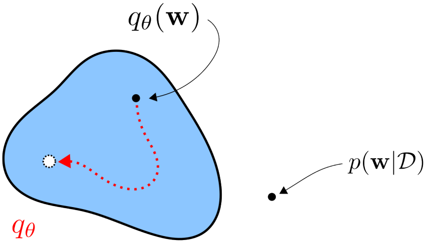
Variational Inference

VI $\rightarrow$ Parametric q to **approximate** target (intractable) posterior p

Evidence Lower Bound (ELBO):

$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_q[\log p(\mathbf{y}_i|\mathbf{W}, \mathbf{x}_i)] - \mathrm{KL}(q|\mathrm{prior})$$

- Monte Carlo and mini-batches!
- Closed-form solution if p and q are Gaussian!



If $p(\mathbf{w}|\mathcal{D}) \notin q_\theta$, we do the best we can (*maybe not enough...*)

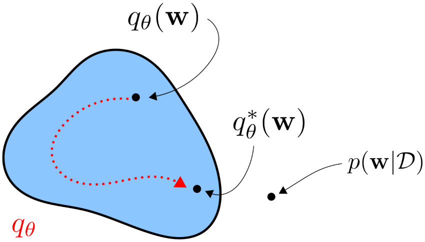
Variational Inference

VI $\rightarrow$ Parametric q to **approximate** target (intractable) posterior p

Evidence Lower Bound (ELBO):

$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_q[\log p(\mathbf{y}_i|\mathbf{W},\mathbf{x}_i)] - \mathrm{KL}(q|\text{prior})$$

- Monte Carlo and mini-batches!
- Closed-form solution if p and q are Gaussian!



If $p(\mathbf{w}|\mathcal{D}) \notin q_\theta$, we do the best we can (*maybe not enough...*)

VI with implicit distributions

More flexible inference model ⇒ Implicit model for weights

Implicit distribution: Samples available, but not the p.d.f.

VI with implicit distributions

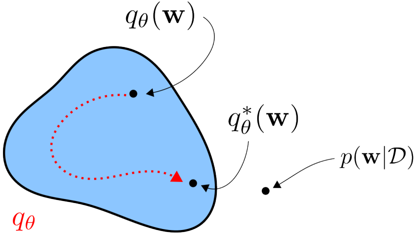
More flexible inference model $\Rightarrow$ Implicit model for weights

Implicit distribution: Samples available, but not the p.d.f.

$$\mathbf{w} = g(\epsilon, \theta)$$

$$\Downarrow$$

$$q_\theta(\mathbf{w})$$



VI with implicit distributions

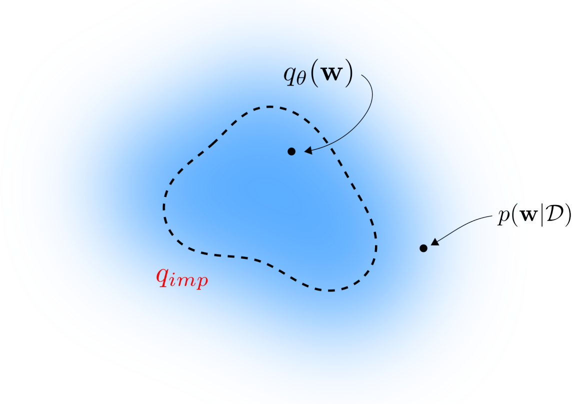
More flexible inference model ⇒ Implicit model for weights

Implicit distribution: Samples available, but not the p.d.f.

$$\mathbf{w} = g(\epsilon, \theta)$$

$$\Downarrow$$

$$q_\theta(\mathbf{w})$$



VI with implicit distributions

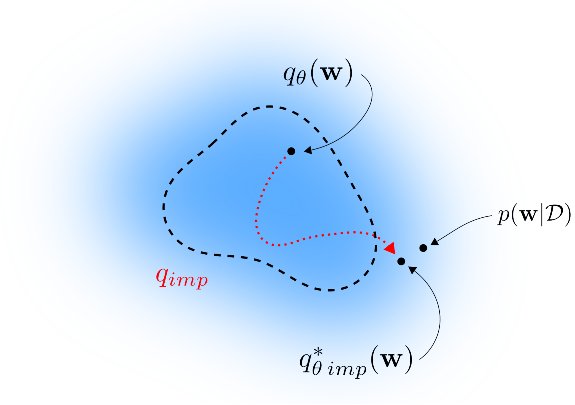
More flexible inference model $\Rightarrow$ Implicit model for weights

Implicit distribution: Samples available, but not the p.d.f.

$$\mathbf{w} = g(\epsilon, \theta)$$

$$\Downarrow$$

$$q_\theta(\mathbf{w})$$



VI with implicit distributions

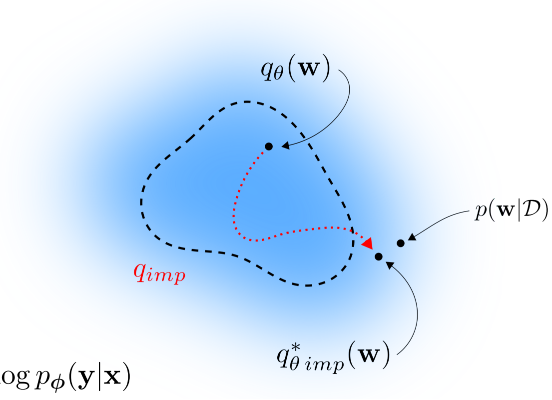
More flexible inference model ⇒ Implicit model for weights

Implicit distribution: Samples available, but not the p.d.f.

$$\mathbf{w} = g(\epsilon, \theta)$$

$$\Downarrow$$

$$q_\theta(\mathbf{w})$$



ML training → $\mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} \log p_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})$

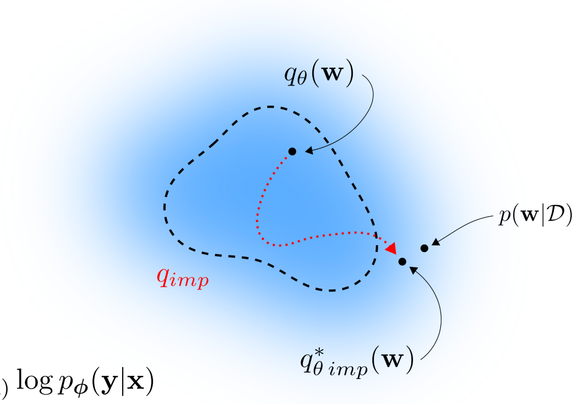
$$\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\Big[\underbrace{-\mathrm{KL}(q_{\boldsymbol{\theta}}(\mathbf{w})||p(\mathbf{w}))}_{\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{w})}[\log p(\mathbf{w}) - \log q_{\boldsymbol{\theta}}(\mathbf{w})]} + \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{w})} \log p_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x},\mathbf{w})\Big]$$

VI with implicit distributions

More flexible inference model $\Rightarrow$ Implicit model for weights

Implicit distribution: Samples available, but not the p.d.f.

$$\mathbf{w} = g(\epsilon, \theta)$$
$$\Downarrow$$
$$q_\theta(\mathbf{w})$$



ML training $\rightarrow \mathbb{E}_{p_\mathcal{D}(\mathbf{x})} \log p_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})$

$$\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{p_\mathcal{D}(\mathbf{x})} \Big[\underbrace{-\mathrm{KL}(q_{\boldsymbol{\theta}}(\mathbf{w}) || p(\mathbf{w}))}_{\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{w})}[T^*_w(\mathbf{w})]} + \mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{w})} \log p_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x}, \mathbf{w}) \Big]$$

[Mescheder et. al., 2017]

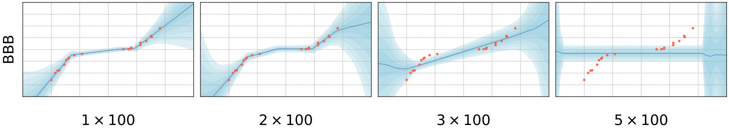
Parameter-space vs. Function-space

Regular approximate Bayesian inference ⇒ **parameter space**

Parameter-space vs. Function-space

Regular approximate Bayesian inference ⇒ **parameter space**

- Curse of dimensionality, correlations and symmetries & local optima

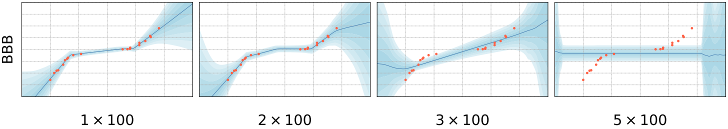


[Sun et al., 2019]

Parameter-space vs. Function-space

Regular approximate Bayesian inference $\Rightarrow$ **parameter space**

- Curse of dimensionality, correlations and symmetries & local optima



Function-space is challenging but with **benefitial**:

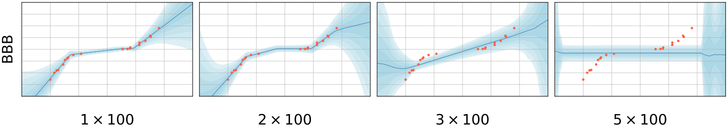
1. Avoids issues related to the original inference problem space
2. Better predictions and uncertainty estimates
3. More flexible priors than GPs

[Sun et al., 2019]

Parameter-space vs. Function-space

Regular approximate Bayesian inference ⇒ **parameter space**

- Curse of dimensionality, correlations and symmetries & local optima



Function-space is challenging but with **benefitial**:

1. Avoids issues related to the original inference problem space
2. Better predictions and uncertainty estimates
3. More flexible priors than GPs

Implicit Processes ⇒ generalization for the prior and posterior formulation in function-space

[Sun et al., 2019]

Implicit Processes

Collection of random variables $f(\cdot)$, such that any finite collection $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$ has joint distribution defined by the generative process:

$$\mathbf{z} \sim p(\mathbf{z}), \quad f(\mathbf{x}_n) = g_\theta(\mathbf{x}_n, \mathbf{z})$$

Implicit Processes

Collection of random variables $f(\cdot)$, such that any finite collection $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$ has joint distribution defined by the generative process:

$$\mathbf{z} \sim p(\mathbf{z}), \quad f(\mathbf{x}_n) = g_\theta(\mathbf{x}_n, \mathbf{z})$$

Bayesian neural networks: $\theta \Rightarrow$ means and variances of $\mathbf{W}$

$$\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad f(\mathbf{x}) = g_\theta(\mathbf{W}, \mathbf{x})$$

Implicit Processes

Collection of random variables $f(\cdot)$, such that any finite collection $(f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))$ has joint distribution defined by the generative process:

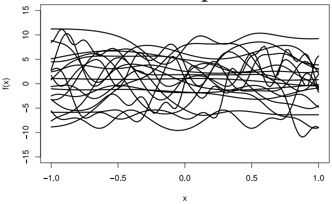
$$\mathbf{z} \sim p(\mathbf{z}), \quad f(\mathbf{x}_n) = g_\theta(\mathbf{x}_n, \mathbf{z})$$

Bayesian neural networks: $\theta \Rightarrow$ means and variances of $\mathbf{W}$

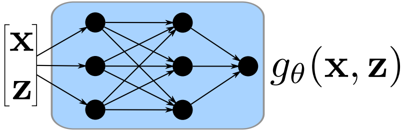
$$\mathbf{W} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad f(\mathbf{x}) = g_\theta(\mathbf{W}, \mathbf{x})$$

Neural sampler: $\theta \Rightarrow$ weights of non-linear function $g_\theta(\cdot, \cdot)$.

Prior samples



Neural sampler



Learning under Implicit Process Priors

Goals:

1. Find flexible approximations to the exact posterior distribution
2. Train all model's parameters

Learning under Implicit Process Priors

Goals:

1. Find flexible approximations to the exact posterior distribution
2. Train all model's parameters

Previous approaches:

1. Variational Implicit Process (**VIP**, Ma et al., 2019)
   - ▶ IP prior and GP approximation for the predictions
   - ⊘ Only provides GP-like predictions (Normally distributed)
2. Functional Bayesian Neural Network (**FBNN**, Sun et al., 2019)
   - ▶ IP prior & posterior, trained using Stein Gradient Estimator
   - ⊘ SGE approach cannot train the prior parameters

Inference with IPs and inducing points

Implicit process $f(\mathbf{x}) = h_\phi(\mathbf{x}, \boldsymbol{\epsilon})$ as approximate implicit posterior of the IP prior ($\sim$*FBNNs*, *full IP-based model*)

Approximate Inference via functional VI (*f-ELBO*):

$$\mathcal{L}(q) = \sum_{i=1}^{N} \mathbb{E}_q[\log p(y_i|f(\mathbf{x}_i))] - \mathrm{KL}(q|\mathrm{prior})\,.$$

Inference with IPs and inducing points

Implicit process $f(\mathbf{x}) = h_\phi(\mathbf{x}, \boldsymbol{\epsilon})$ as approximate implicit posterior of the IP prior ($\sim$*FBNNs*, *full IP-based model*)

Approximate Inference via functional VI (*f-ELBO*):

$$\mathcal{L}(q) = \sum_{i=1}^{N} \mathbb{E}_q[\log p(y_i|f(\mathbf{x}_i))] - \mathrm{KL}(q|\mathrm{prior})\,.$$

Challenges:

1. Scalability with N
   - $M \ll N$ inducing points as in Sparse GPs ($\overline{\mathbf{X}}$, $\mathbf{u}$), with

$$\mathbf{u} = f(\overline{\mathbf{X}})$$

Inference with IPs and inducing points

Implicit process $f(\mathbf{x}) = h_\phi(\mathbf{x}, \boldsymbol{\epsilon})$ as approximate implicit posterior of the IP prior ($\sim$*FBNNs*, *full IP-based model*)

Approximate Inference via functional VI (*f-ELBO*):

$$\mathcal{L}(q) = \sum_{i=1}^{N} \mathbb{E}_q[\log p(y_i|f(\mathbf{x}_i))] - \mathrm{KL}(q|\mathrm{prior})\,.$$

Challenges:

1. Scalability with N
   - $M \ll N$ inducing points as in Sparse GPs ($\overline{\mathbf{X}}$, $\mathbf{u}$), with

     $$\mathbf{u} = f(\overline{\mathbf{X}})$$

2. Intractable conditional posterior
   - Partial Monte Carlo GP approximation for the conditional $p(\mathbf{f}|\mathbf{u})$ in the posterior ($\sim$*VIPs*)

Training the system

Final posterior approximation (with implicit $q_\phi(\mathbf{u})$):

$$q(\mathbf{f}, \mathbf{u}) = p_\theta(\mathbf{f}|\mathbf{u})q_\phi(\mathbf{u})$$

Training the system

Final posterior approximation (with implicit $q_\phi(\mathbf{u})$):

$$q(\mathbf{f}, \mathbf{u}) = p_\theta(\mathbf{f}|\mathbf{u}) q_\phi(\mathbf{u})$$

f-ELBO objective:

$$\begin{aligned}\mathcal{L}(q) &= \mathbb{E}_q \left[\log \frac{p(\mathbf{y}|\mathbf{f}) \cancel{p_\theta(\mathbf{f}|\mathbf{u})} p_\theta(\mathbf{u})}{\cancel{p_\theta(\mathbf{f}|\mathbf{u})} q_\phi(\mathbf{u})} \right] \\ &= \sum_{i=1}^{N} \mathbb{E}_{q_{\phi,\theta}} [\log p(y_i|f_i)] - \mathrm{KL}(q_\phi(\mathbf{u}) | p_\theta(\mathbf{u}))\end{aligned}$$

Training the system

Final posterior approximation (with implicit $q_\phi(\mathbf{u})$):

$$q(\mathbf{f}, \mathbf{u}) = p_\theta(\mathbf{f}|\mathbf{u}) q_\phi(\mathbf{u})$$

f-ELBO objective:

$$\begin{aligned}\mathcal{L}(q) &= \mathbb{E}_q \left[\log \frac{p(\mathbf{y}|\mathbf{f}) \cancel{p_\theta(\mathbf{f}|\mathbf{u})} p_\theta(\mathbf{u})}{\cancel{p_\theta(\mathbf{f}|\mathbf{u})} q_\phi(\mathbf{u})} \right] \\ &= \sum_{i=1}^{N} \mathbb{E}_{q_{\phi,\theta}} [\log p(y_i|f_i)] - \mathrm{KL}(q_\phi(\mathbf{u}) | p_\theta(\mathbf{u}))\end{aligned}$$

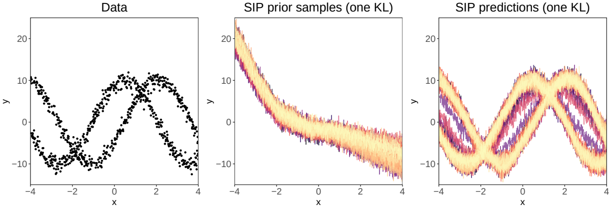
KL-divergence **intractable** (**implicit** q and p) $\Rightarrow$ classifier (DNN)

$$\mathrm{KL}(q_\phi(\mathbf{u}) | p_\theta(\mathbf{u})) = -\mathbb{E}_q \left[\log \frac{p_\theta(\mathbf{u})}{q_\phi(\mathbf{u})} \right] = -\mathbb{E}_q \left[T_{\Omega^\star}(\mathbf{u}) \right]$$

[Mescheder et. al., 2017]

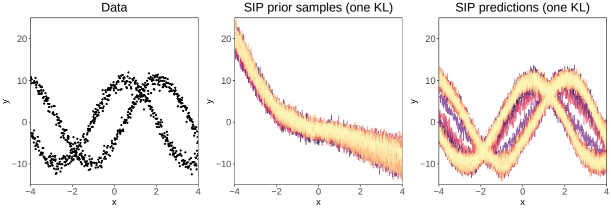
Challenges: KL-evaluation

Poor prior fit, important in complex models [Knoblauch et al. 2019]



Challenges: KL-evaluation

Poor prior fit, important in complex models [Knoblauch et al. 2019]

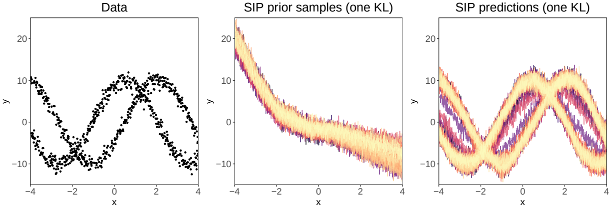


Solution: Exchange KL by the symmetrized KL-divergence

$$\mathrm{KL}(q_\phi|p_\theta) \approx \frac{1}{2}(\mathrm{KL}(q_\phi|p_\theta) + \mathrm{KL}(p_\theta|q_\phi)) \,,$$

Challenges: KL-evaluation

Poor prior fit, important in complex models [Knoblauch et al. 2019]



Solution: Exchange KL by the symmetrized KL-divergence

$$\mathrm{KL}(q_\phi | p_\theta) \approx \frac{1}{2}(\mathrm{KL}(q_\phi | p_\theta) + \mathrm{KL}(p_\theta | q_\phi)) \, ,$$

KL as regularization in the ELBO $\Rightarrow$ changes often improve results

- Easy to compute dependencies w.r.t. θ
- Good empirical results + little added computational cost

[Wenzel et. al., 2020]

Final setup

Final objective function (with α-divergences + symmetrized KL):

$$\boxed{\mathcal{L}_{\alpha}^{\star}(\phi, \theta) = \frac{1}{\alpha} \sum_{i=1}^{N} \log \mathbb{E}_{q_{\phi,\theta}}[p(y_i|f_i)^{\alpha}] - \frac{1}{2} \left[\mathrm{KL}(q_{\phi}||p_{\theta}) + \mathrm{KL}(p_{\theta}||q_{\phi})\right]}$$

Final setup

Final objective function (with α-divergences + symmetrized KL):

$$\mathcal{L}^{\star}_{\alpha}(\phi, \theta) = \frac{1}{\alpha} \sum_{i=1}^{N} \log \mathbb{E}_{q_{\phi,\theta}} [p(y_i | f_i)^{\alpha}] - \frac{1}{2} [\mathrm{KL}(q_{\phi} || p_{\theta}) + \mathrm{KL}(p_{\theta} || q_{\phi})]$$

And $p_{\theta}(\mathbf{f}|\mathbf{u})$?

Remember that $q(\mathbf{f}, \mathbf{u}) = p_{\theta}(\mathbf{f}|\mathbf{u}) q_{\phi}(\mathbf{u})$

Final setup

Final objective function (with α-divergences + symmetrized KL):

$$\mathcal{L}^{\star}_{\alpha}(\phi, \theta) = \frac{1}{\alpha} \sum_{i=1}^{N} \log \mathbb{E}_{q_{\phi,\theta}}[p(y_i|f_i)^{\alpha}] - \frac{1}{2}\left[\mathrm{KL}(q_{\phi}||p_{\theta}) + \mathrm{KL}(p_{\theta}||q_{\phi})\right]$$

And $p_{\theta}(\mathbf{f}|\mathbf{u})$?

Remember that $q(\mathbf{f}, \mathbf{u}) = p_{\theta}(\mathbf{f}|\mathbf{u}) q_{\phi}(\mathbf{u})$

GP approximation ($\sim$VIP)

$$\mathbb{E}[f(\mathbf{x})] = m^{\star}_{MLE}(\mathbf{x}) + \mathbf{K}_{\mathbf{f},\mathbf{u}}(\mathbf{K}_{\mathbf{u},\mathbf{u}} + \mathbf{I}\sigma^2)^{-1}(\mathbf{u} - m^{\star}_{\mathrm{MLE}}(\mathbf{X}))\,,$$
$$\mathrm{Var}(f(\mathbf{x})) = \mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{f}}(\mathbf{K}_{\mathbf{u},\mathbf{u}} + \mathbf{I}\sigma^2)^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}$$

Covariances $\Rightarrow$ Monte Carlo methods sampling from the prior

Final setup

Final objective function (with α-divergences + symmetrized KL):

$$\mathcal{L}^{\star}_{\alpha}(\phi, \theta) = \frac{1}{\alpha} \sum_{i=1}^{N} \log \mathbb{E}_{q_{\phi,\theta}}[p(y_i|f_i)^{\alpha}] - \frac{1}{2} \left[\mathrm{KL}(q_\phi || p_\theta) + \mathrm{KL}(p_\theta || q_\phi) \right]$$

And $p_\theta(\mathbf{f}|\mathbf{u})$?

Remember that $q(\mathbf{f}, \mathbf{u}) = p_\theta(\mathbf{f}|\mathbf{u}) q_\phi(\mathbf{u})$

GP approximation ($\sim$VIP)

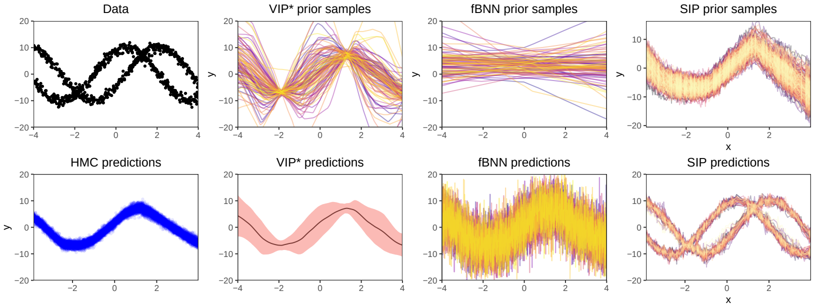
$$\mathbb{E}[f(\mathbf{x})] = m^{\star}_{MLE}(\mathbf{x}) + \mathbf{K}_{\mathbf{f},\mathbf{u}}(\mathbf{K}_{\mathbf{u},\mathbf{u}} + \mathbf{I}\sigma^2)^{-1}(\mathbf{u} - m^{\star}_{\mathrm{MLE}}(\mathbf{X}))\,,$$
$$\mathrm{Var}(f(\mathbf{x})) = \mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{f}}(\mathbf{K}_{\mathbf{u},\mathbf{u}} + \mathbf{I}\sigma^2)^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}$$

Covariances $\Rightarrow$ Monte Carlo methods sampling from the prior

Predictions approximated by Monte Carlo (mixture of Gaussians):

$$p(f(\mathbf{x}_*)|\mathbf{y}, \mathbf{X}) \approx \frac{1}{S} \sum_{s=1}^{S} p_\theta(f(\mathbf{x}_*)|\mathbf{u}_s)\,, \qquad \mathbf{u} \sim q_\phi(\mathbf{u})$$

Synthetic data experiments



VIP regularization term is not used

Same BNN prior for all methods

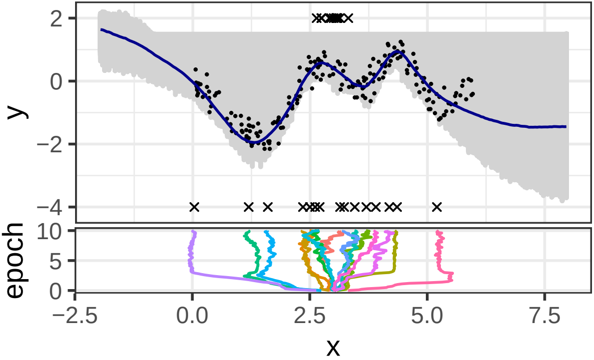
SIP is the only one with **fitted prior samples** and **bimodal predictive distribution**

SIP corrects the model bias that induces the wrong posterior!

- Combination of flexibility of the framework + α-divergences

Evolution of the inducing points

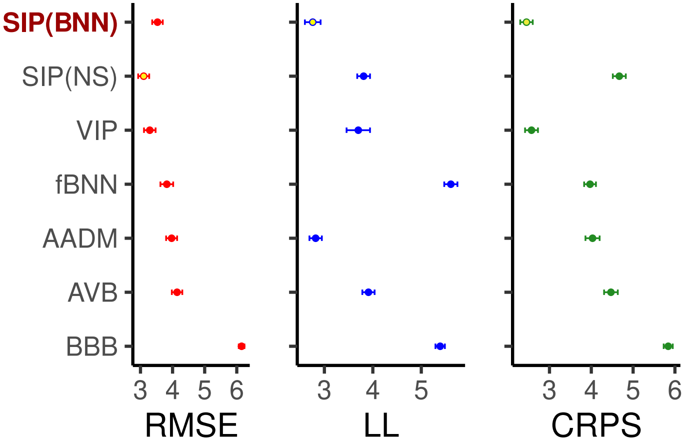
Inducing points spread and cover the whole training data range



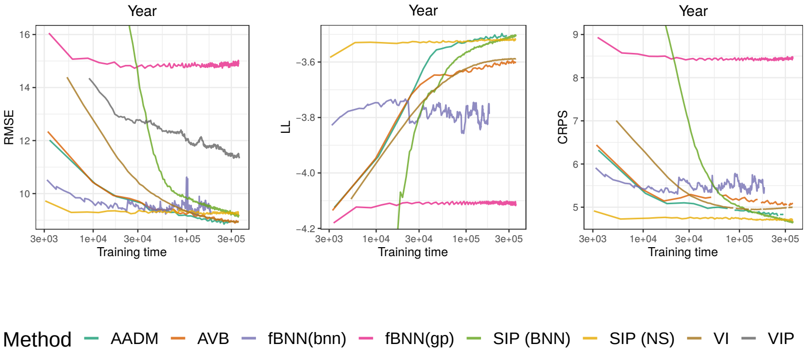
Posterior parameters are not trained for this example:
slight underfitting + *adversarial initialization*

Regression results

Ranking analysis (lower is better, 8 UCI datasets, 20 splits each, 2σ)



Convergence experiments



$\mathbf{SIP_{NS}}$ is clearly faster, $\mathbf{SIP_{BNN}}$ performs the best overall

Conclusions

1. Approximate inference in parameter space presents intrinsic difficulties

2. Approximate inference in function space is advantageous but hard

   ! Allowing the model to train all of its parameters

   ! Provide flexible predictive distributions

3. **SIP** has new important properties
   - ✓ Can learn the prior parameters
   - ✓ Flexible posterior approximation via mixture of Gaussians
   - ✓ Scalable with large amounts of data
   - ✓ SIP can use other flexible priors based on implicit processes
   - ✓ Capable of correcting wrong model bias from the formulation

References

- Ma, C., Li, Y., Hernández-Lobato, J. M. Variational implicit processes. International Conference on Machine Learning, 2019.
- Titsias, M. (2009, April). Variational learning of inducing variables in sparse Gaussian processes. In Artificial Intelligence and Statistics (pp. 567-574).
- Knoblauch, J., Jewson, J. and Damoulas, T. "Generalized variational inference: Three arguments for deriving new posteriors." arXiv preprint arXiv:1904.02063 (2019).
- S. Sun, G. Zhang, J. Shi, R. Grosse. Functional Variational Bayesian Neural Networks. International Conference on Learning Representations, 2019.
- Mescheder, L., Nowozin, S., Geiger, A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. International Conference on Machine Learning, 2017.
- Rodrguez Santana, S. and Hernández-Lobato, D. Adversarial α-divergence minimization for Bayesian approximate inference. Neurocomputing, (2020).

Thanks for your attention!

`https://github.com/simonrsantana/sparse-implicit-processes`

✉ simon.rodriguez@icmat.es

simonrodsan