Deep Learning: Classics and Trends

Imperial College London

# Variational Gaussian Processes without Matrix Inverses

**Mark van der Wilk**

Department of Computing
Imperial College London

@markvanderwilk
m.vdwilk@imperial.ac.uk

June 10, 2020

# Work in progress

Ongoing work with some initial results presented in

2nd Symposium on Advances in Approximate Bayesian Inference, 2019 1–8

**Variational Gaussian Process Models without Matrix Inverses**

**Mark van der Wilk** MARK@PROWLER.IO
**ST John** ST@PROWLER.IO
**Artem Artemev** ARTEM@PROWLER.IO
**James Hensman** JAMES@PROWLER.IO
*PROWLER.io*

# Motivation: Model selection

In deep learning, how to choose the hyperparameters like

- number of layers?
- number of hidden units?
- convolutional or fully connected layer?
- other invariances?
- parameters of data augmentation?
- ...

# Motivation: Model selection

In deep learning, how to choose the hyperparameters like

- number of layers?
- number of hidden units?
- convolutional or fully connected layer?
- other invariances?
- parameters of data augmentation?
- ...

Current solution: **manual tuning** and **cross-validation**.

# Motivation: Model selection

In deep learning, how to choose the hyperparameters like

- number of layers?
- number of hidden units?
- convolutional or fully connected layer?
- other invariances?
- parameters of data augmentation?
- ...

Current solution: **manual tuning** and **cross-validation**.

Wouldn't it be great
if we could just find these by backprop?

# Bayesian model selection

Bayesian inference gives a solution

$$p(f, \theta \mid \mathbf{y}, X) = \frac{p(\mathbf{y}, f, \theta \mid X)}{p(\mathbf{y}|X)} = \frac{p(\mathbf{y} \mid f, X, \theta) p(f|\theta) p(\theta)}{p(\mathbf{y}|X)} \tag{1}$$

$$= \underbrace{\frac{p(\mathbf{y} \mid f, X, \theta) p(f|\theta)}{p(\mathbf{y} \mid X, \theta)}}_{p(f|\mathbf{y},X)} \underbrace{\frac{p(\mathbf{y} \mid X, \theta) p(\theta)}{p(\mathbf{y}|X)}}_{p(\theta \mid \mathbf{y},X)} \tag{2}$$

Posterior over $f$ and $\theta$ consists of two parts

1. The original posterior over $f$,
2. A posterior over $\theta$ using the **marginal likelihood**:

$$p(\mathbf{y}|X, \theta) = \int p(\mathbf{y}|f, X, \theta) p(f|\theta) \mathrm{d}\theta \tag{3}$$

# Bayesian Deep Learning

Bayesian deep learning has

- focussed strongly on getting uncertainty from the posterior $p(f \mid \mathbf{y}, X)$.
- **not** focussed on model selection, because it is **very hard** to find an approximation to the marginal likelihood $p(\mathbf{y}|X, \theta) = \int p(\mathbf{y}|f, X, \theta) p(f|\theta) \mathrm{d}\theta$.

# Bayesian Deep Learning

Bayesian deep learning has

- focussed strongly on getting uncertainty from the posterior $p(f \mid \mathbf{y}, X)$.
- **not** focussed on model selection, because it is **very hard** to find an approximation to the marginal likelihood $p(\mathbf{y}|X, \theta) = \int p(\mathbf{y}|f, X, \theta) p(f|\theta) \mathrm{d}\theta$.

Contrast to Gaussian process models where hyperparameters are **routinely learned using the marginal likelihood!**

# Bayesian Deep Learning

Bayesian deep learning has

- focussed strongly on getting uncertainty from the posterior $p(f \mid \mathbf{y}, X)$.
- **not** focussed on model selection, because it is **very hard** to find an approximation to the marginal likelihood $p(\mathbf{y}|X, \theta) = \int p(\mathbf{y}|f, X, \theta) p(f|\theta) \mathrm{d}\theta$.

Contrast to Gaussian process models where hyperparameters are **routinely learned using the marginal likelihood!**

- *Convolutional Gaussian Processes* (van der Wilk et al., 2017)
  How much convolutional structure to use vs fully connected?

# Bayesian Deep Learning

Bayesian deep learning has

- focussed strongly on getting uncertainty from the posterior $p(f \mid \mathbf{y}, X)$.
- **not** focussed on model selection, because it is **very hard** to find an approximation to the marginal likelihood $p(\mathbf{y}|X, \theta) = \int p(\mathbf{y}|f, X, \theta) p(f|\theta) \mathrm{d}\theta$.

Contrast to Gaussian process models where hyperparameters are **routinely learned using the marginal likelihood!**

- *Convolutional Gaussian Processes* (van der Wilk et al., 2017)
  How much convolutional structure to use vs fully connected?
- *Learning Invariances using the Marginal Likelihood* (van der Wilk et al., 2018)
  Backpropagate the parameters of data augmentation, without a validation set.

# Bayesian Deep Learning

Bayesian deep learning has

- focussed strongly on getting uncertainty from the posterior $p(f \mid \mathbf{y}, X)$.
- **not** focussed on model selection, because it is **very hard** to find an approximation to the marginal likelihood $p(\mathbf{y}|X, \theta) = \int p(\mathbf{y}|f, X, \theta) p(f|\theta) \mathrm{d}\theta$.

Contrast to Gaussian process models where hyperparameters are **routinely learned using the marginal likelihood!**

- *Convolutional Gaussian Processes* (van der Wilk et al., 2017)
  How much convolutional structure to use vs fully connected?
- *Learning Invariances using the Marginal Likelihood* (van der Wilk et al., 2018)
  Backpropagate the parameters of data augmentation, without a validation set.
- *Deep Gaussian Processes* (Damianou and Lawrence, 2013)
  How many hidden units to use? **Deep, but with marg. lik.!**

# Gaussian processes as a building block

It is already as **simple** to perform variational inference in complex GP models as parametric models.

# Gaussian processes as a building block

It is already as **simple** to perform variational inference in complex GP models as parametric models.

So why aren't we using deep Gaussian processes everywhere?

# Gaussian processes as a building block

It is already as **simple** to perform variational inference in complex GP models as parametric models.

So why aren't we using deep Gaussian processes everywhere?

$$\mathbf{K}_{ZZ}^{-1}$$

# Gaussian processes as a building block

It is already as **simple** to perform variational inference in complex GP models as parametric models.

So why aren't we using deep Gaussian processes everywhere?

$$\mathbf{K}_{ZZ}^{-1}$$

- Neural networks only rely on cheap matrix-vector products.
- As long as GPs rely on matrix decompositions in each iteration, they will be slower
- Want **computations** to be similar to deep learning. Doing things **through optimisation** seems key.

# Overview

Variational inference in Gaussian processes

An inverse-free approximate posterior

A general inverse-free variational bound

Recent progress

Conclusions

# A solution to many problems

Variational inference for GPs has been developed over a long period of time

1. Avoid large matrix inverse for regression (Titsias, 2009)
2. Allow big data through minibatching (Hensman et al., 2013)
3. Analytical intractability of non-Gaussian likelihoods (Hensman et al., 2015)
4. General models: Latent variables (Titsias and Lawrence, 2010), deep structure (Damianou and Lawrence, 2013), recurrent structure (Frigola et al., 2014), ...

# A solution to many problems

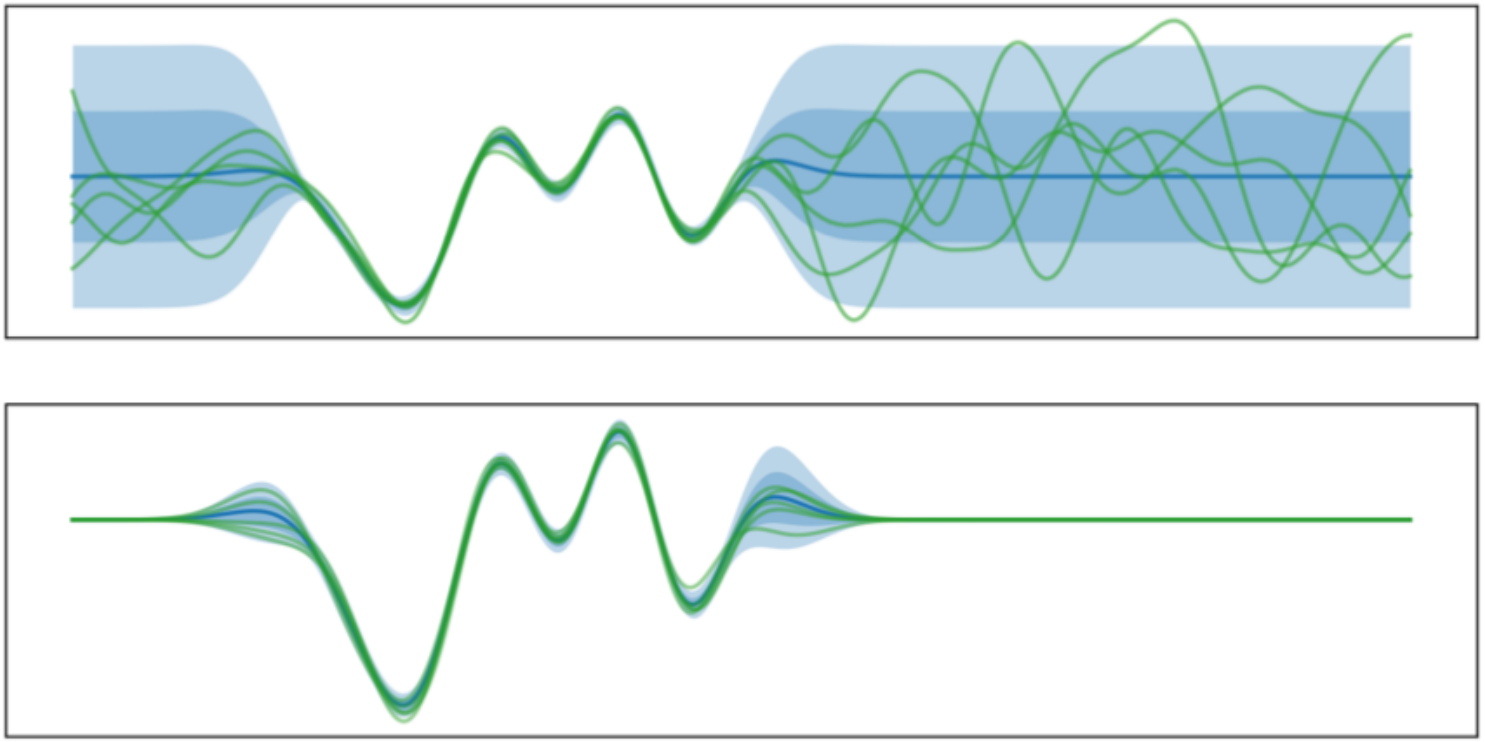
Variational inference for GPs has been developed over a long period of time

1. Avoid large matrix inverse for regression (Titsias, 2009)
2. Allow big data through minibatching (Hensman et al., 2013)
3. Analytical intractability of non-Gaussian likelihoods (Hensman et al., 2015)
4. General models: Latent variables (Titsias and Lawrence, 2010), deep structure (Damianou and Lawrence, 2013), recurrent structure (Frigola et al., 2014), ...

Currently it is **generally applicable** to a wide variety of models.

# Variational Inference in GPs

Crucial property that other approximations lack



Variational approx maintain properties of the non-parametric GP

- Predict with infinite basis functions (better uncertainty)
- Approximate marginal likelihood of non-parametric model

# Recap: Sparse Stochastic Variational Inference

In three simple steps.

1. Introduce tractable variational distribution

$$q(f(\cdot)) = p(f(\cdot)|f(Z))q(f(Z)) \tag{4}$$

# Recap: Sparse Stochastic Variational Inference

In three simple steps.

1. Introduce tractable variational distribution

$$q(f(\cdot)) = p(f(\cdot)|f(Z))q(f(Z)) \tag{4}$$

2. Formulate variational lower bound

$$\mathcal{L} = \sum_n \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(f(Z))||p(f(Z))] \tag{5}$$

# Recap: Sparse Stochastic Variational Inference

In three simple steps.

1. Introduce tractable variational distribution

$$q(f(\cdot)) = p(f(\cdot)|f(Z))q(f(Z)) \tag{4}$$

2. Formulate variational lower bound

$$\mathcal{L} = \sum_n \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(f(Z))||p(f(Z))] \tag{5}$$

3. Maximise $\mathcal{L}$ to minimise $\mathrm{KL}[q(f)||p(f|\mathbf{y})]$

# Approximate posterior

# Approximate posterior

1. Conditioning the prior on observations at **inducing input locations** $Z$. We sometimes denote $\mathbf{u} = f(Z)$ for brevity.

$$p(f(\cdot) \mid \mathbf{u}) = \mathcal{N}\left(f(\cdot); \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{u}, k_{\cdot\cdot} - \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot}\right) \tag{6}$$

# Approximate posterior

1. Conditioning the prior on observations at **inducing input locations** $Z$. We sometimes denote $\mathbf{u} = f(Z)$ for brevity.

$$p(f(\cdot) \mid \mathbf{u}) = \mathcal{N}\left(f(\cdot); \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{u}, k_{\cdot\cdot} - \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot}\right) \tag{6}$$

2. Specify a freely parameterised Gaussian marginal on $\mathbf{u}$:

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}, \mathbf{S}) \tag{7}$$

# Approximate posterior

1. Conditioning the prior on observations at **inducing input locations** $Z$. We sometimes denote $\mathbf{u} = f(Z)$ for brevity.

$$p(f(\cdot) \mid \mathbf{u}) = \mathcal{N}\left(f(\cdot); \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{u}, k_{\cdot\cdot} - \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot}\right) \tag{6}$$

2. Specify a freely parameterised Gaussian marginal on $\mathbf{u}$:

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}, \mathbf{S}) \tag{7}$$

3. Marginalise $\mathbf{u}$ to find approximate posterior:

$$\begin{aligned} q(f(\cdot)) &= \int p(f(\cdot) \mid \mathbf{u}) q(\mathbf{u}) \mathrm{d}\mathbf{u} \\ &= \mathcal{N}\left(f(\cdot); \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{m}, k_{\cdot\cdot} - \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot} + \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{S}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot}\right) \end{aligned} \tag{8}$$

Variational parameters: $\{Z, \mathbf{m}, \mathbf{S}\}$

# Predictions and bound

Remember the ELBO:

$$\mathcal{L} = \sum_n \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(f(Z))||p(f(Z))] \tag{9}$$

Start with focus on expected log likelihood...

# Predictions and bound

Remember the ELBO:

$$\mathcal{L} = \sum_n \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(f(Z))||p(f(Z))] \tag{9}$$

Start with focus on expected log likelihood... Take e.g. Gaussian

$$\mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(y_n - \mu_n)^2 - \frac{\sigma_n^2}{2\sigma^2} \tag{10}$$

$$q(f(\cdot)) = \mathcal{N}\Big(f(\cdot); \underbrace{\mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{m}}_{\mu_n}, \underbrace{k_{\cdot\cdot} - \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot} + \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{S}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot}}_{\sigma_n^2}\Big)$$

# Predictions and bound

Remember the ELBO:

$$\mathcal{L} = \sum_n \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(f(Z))||p(f(Z))] \tag{9}$$

Start with focus on expected log likelihood... Take e.g. Gaussian

$$\mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(y_n - \mu_n)^2 - \frac{\sigma_n^2}{2\sigma^2} \tag{10}$$

$$q(f(\cdot)) = \mathcal{N}\Big(f(\cdot); \underbrace{\mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{m}}_{\mu_n}, \underbrace{k_{\cdot\cdot} - \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot} + \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{S}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot}}_{\sigma_n^2}\Big)$$

We require computation of matrix inverse $\mathbf{K}_{ZZ}^{-1}$

# Removing matrix inverses

Can we reparameterise the approximate posterior to remove the matrix inverses?

$$q(f(\cdot)) = \mathcal{N}\left(f(\cdot); \mu_n, \sigma_n^2\right) \tag{11}$$

$$\mu_n = \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{m} = \mathbf{k}_{\cdot Z}\mathbf{m}' \tag{12}$$

$$\begin{aligned}\sigma_n^2 &= k_{\cdot\cdot} - \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot} + \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{S}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot} \\ &= k_{\cdot\cdot} - \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot} + \mathbf{k}_{\cdot Z}\mathbf{S}'\mathbf{k}_{Z\cdot}\end{aligned} \tag{13}$$

# Removing matrix inverses

Can we reparameterise the approximate posterior to remove the matrix inverses?

$$q(f(\cdot)) = \mathcal{N}\left(f(\cdot); \mu_n, \sigma_n^2\right) \tag{11}$$

$$\mu_n = \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{m} = \mathbf{k}_{\cdot Z}\mathbf{m}' \tag{12}$$

$$\begin{aligned}\sigma_n^2 &= k_{\cdot\cdot} - \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot} + \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{S}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot} \\ &= k_{\cdot\cdot} - \mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot} + \mathbf{k}_{\cdot Z}\mathbf{S}'\mathbf{k}_{Z\cdot}\end{aligned} \tag{13}$$

Some progress, but $\mathbf{k}_{\cdot Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\cdot}$ is the difficult term.

# Upper bounding the predictive variance

Looking back at the expected log likelihood term

$$\mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(y_n - \mu_n)^2 - \frac{\sigma_n^2}{2\sigma^2} \quad (14)$$

$$\sigma_n^2 = k_{..} - \mathbf{k}_{.Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{.Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \quad (15)$$

# Upper bounding the predictive variance

Looking back at the expected log likelihood term

$$\mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(y_n - \mu_n)^2 - \frac{\sigma_n^2}{2\sigma^2} \quad (14)$$

$$\sigma_n^2 = k_{..} - \mathbf{k}_{.Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{.Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \quad (15)$$

Observation: An upper bound on $\sigma_n^2$ gives a lower bound to the ELBO. I.e. for $\bar{\sigma}_n^2 \geqslant \sigma_n^2$,

$$\mathcal{L}' = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL} \leqslant \mathcal{L} \leqslant p(\mathbf{y}) \quad (16)$$

# Upper bounding the predictive variance

Can we find an upper bound to the predictive variance?

$$\sigma_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} - \mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \tag{17}$$

Observation: $-\mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n}$ is the minimum of a quadratic.

$$-\mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} = \min_{\mathbf{v}} \quad \mathbf{v}^\top\mathbf{K}_{ZZ}\mathbf{v} - 2\mathbf{k}_{\mathbf{x}_n Z}^\top\mathbf{v} \tag{18}$$

$$\mathbf{v}_n^* = \mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} = \operatorname*{argmin}_{\mathbf{v}} \quad \mathbf{v}^\top\mathbf{K}_{ZZ}\mathbf{v} - 2\mathbf{k}_{\mathbf{x}_n Z}^\top\mathbf{v} \tag{19}$$

# Upper bounding the predictive variance

Can we find an upper bound to the predictive variance?

$$\sigma_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} - \mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \tag{17}$$

Observation: $-\mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n}$ is the minimum of a quadratic.

$$-\mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} = \min_{\mathbf{v}} \quad \mathbf{v}^\top\mathbf{K}_{ZZ}\mathbf{v} - 2\mathbf{k}_{\mathbf{x}_n Z}^\top\mathbf{v} \tag{18}$$

$$\mathbf{v}_n^* = \mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} = \operatorname*{argmin}_{\mathbf{v}} \quad \mathbf{v}^\top\mathbf{K}_{ZZ}\mathbf{v} - 2\mathbf{k}_{\mathbf{x}_n Z}^\top\mathbf{v} \tag{19}$$

Both follow from

$$(\mathbf{v}_n - \mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n})^\top\mathbf{K}_{ZZ}(\mathbf{v}_n - \mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n}) \geqslant 0 \tag{20}$$

Also noted by Gibbs and MacKay (1997) and discussed in Davies (2015) for Conjugate Gradient implementations of GPs

# Upper bounding the predictive variance

Problem: Need to optimise over $\mathbf{v}_n \in \mathbb{R}^M$ **for all** $N$ data points!

# Upper bounding the predictive variance

Problem: Need to optimise over $\mathbf{v}_n \in \mathbb{R}^M$ **for all** $N$ data points!

Since solution $\mathbf{v}_n^* = \mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n}$, we can alternatively parameterise

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{v}_n - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \tag{21}$$

# Upper bounding the predictive variance

Problem: Need to optimise over $\mathbf{v}_n \in \mathbb{R}^M$ **for all** $N$ data points!

Since solution $\mathbf{v}_n^* = \mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n}$, we can alternatively parameterise

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{v}_n - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \tag{21}$$

- Optimise over $\mathbf{T} \in \mathbb{R}^{M\times M}$ instead of $N$ times $\mathbf{v}_n \in \mathbb{R}^M$.

# Upper bounding the predictive variance

Problem: Need to optimise over $\mathbf{v}_n \in \mathbb{R}^M$ **for all** $N$ data points!

Since solution $\mathbf{v}_n^* = \mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n}$, we can alternatively parameterise

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{v}_n - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \tag{21}$$

- Optimise over $\mathbf{T} \in \mathbb{R}^{M\times M}$ instead of $N$ times $\mathbf{v}_n \in \mathbb{R}^M$.
- Recovers original bound at $\mathbf{T} = \mathbf{K}_{ZZ}^{-1}$

# Log-concave bound

Using the upper bound on the predictive variance

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2\,, \quad (22)$$

we get a lower bound on the ELBO

$$\mathcal{L}_{\mathrm{lc}} = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL} \quad (23)$$

$$\leqslant \mathcal{L} \leqslant p(\mathbf{y})\,. \quad (24)$$

# Log-concave bound

Using the upper bound on the predictive variance

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2\,, \quad (22)$$

we get a lower bound on the ELBO

$$\mathcal{L}_{\text{lc}} = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \text{KL} \quad (23)$$

$$\leqslant \mathcal{L} \leqslant p(\mathbf{y})\,. \quad (24)$$

- Contains no matrix inverses ($O(M^2)$ instead of $O(M^3)$)

# Log-concave bound

Using the upper bound on the predictive variance

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2\,, \quad (22)$$

we get a lower bound on the ELBO

$$\mathcal{L}_{\text{lc}} = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \text{KL} \quad (23)$$

$$\leqslant \mathcal{L} \leqslant p(\mathbf{y})\,. \quad (24)$$

- Contains no matrix inverses ($O(M^2)$ instead of $O(M^3)$)
- Valid for all **log-concave** likelihoods

# Log-concave bound

Using the upper bound on the predictive variance

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2\,, \quad (22)$$

we get a lower bound on the ELBO

$$\mathcal{L}_{\text{lc}} = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \text{KL} \quad (23)$$

$$\leqslant \mathcal{L} \leqslant p(\mathbf{y})\,. \quad (24)$$

- Contains no matrix inverses ($O(M^2)$ instead of $O(M^3)$)
- Valid for all **log-concave** likelihoods
- Recovers the original bound when $\mathbf{T} = \mathbf{K}_{ZZ}^{-1}$

# Log-concave bound

Using the upper bound on the predictive variance

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2\,, \quad (22)$$

we get a lower bound on the ELBO

$$\mathcal{L}_{\text{lc}} = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \text{KL} \quad (23)$$

$$\leqslant \mathcal{L} \leqslant p(\mathbf{y})\,. \quad (24)$$

- Contains no matrix inverses ($O(M^2)$ instead of $O(M^3)$)
- Valid for all **log-concave** likelihoods
- Recovers the original bound when $\mathbf{T} = \mathbf{K}_{ZZ}^{-1}$
- Is it a **variational** bound?

# Log-concave bound

Using the upper bound on the predictive variance

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2\,, \quad (22)$$

we get a lower bound on the ELBO

$$\mathcal{L}_{\text{lc}} = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \text{KL} \quad (23)$$

$$\leqslant \mathcal{L} \leqslant p(\mathbf{y})\,. \quad (24)$$

- Contains no matrix inverses ($O(M^2)$ instead of $O(M^3)$)
- Valid for all **log-concave** likelihoods
- Recovers the original bound when $\mathbf{T} = \mathbf{K}_{ZZ}^{-1}$
- Is it a **variational** bound? No! $\log p(\mathbf{y}) - \mathcal{L}_{\text{lc}} \neq \text{KL}[q||p(f\,|\,\mathbf{y})]$

# KL term

**But**,

# KL term

**But**, we still have the log determinant in the KL term

$$\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] = \frac{1}{2}[\mathrm{Tr}[\mathbf{K}_{ZZ}\mathbf{S}'] + \mathbf{m}^\top \mathbf{K}_{ZZ}\mathbf{m} - M - \log|\mathbf{K}_{ZZ}| - \log|\mathbf{S}'|]\,.$$

# KL term

**But**, we still have the log determinant in the KL term

$$\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] = \frac{1}{2}[\mathrm{Tr}[\mathbf{K}_{ZZ}\mathbf{S}'] + \mathbf{m}^\top \mathbf{K}_{ZZ}\mathbf{m} - M - \log|\mathbf{K}_{ZZ}| - \log|\mathbf{S}'|]\,.$$

- Trace term can be handled by Hutchinson estimator

$$\mathrm{Tr}[\mathbf{K}_{ZZ}\mathbf{S}'] = \mathbb{E}_{\mathbf{r}}[\mathbf{r}^\top \mathbf{K}_{ZZ}\mathbf{S}'\mathbf{r}] \tag{25}$$

# KL term

**But**, we still have the log determinant in the KL term

$$\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] = \frac{1}{2}[\mathrm{Tr}[\mathbf{K}_{ZZ}\mathbf{S}'] + \mathbf{m}^\top \mathbf{K}_{ZZ}\mathbf{m} - M - \log|\mathbf{K}_{ZZ}| - \log|\mathbf{S}'|]\,.$$

- Trace term can be handled by Hutchinson estimator

$$\mathrm{Tr}[\mathbf{K}_{ZZ}\mathbf{S}'] = \mathbb{E}_{\mathbf{r}}[\mathbf{r}^\top \mathbf{K}_{ZZ}\mathbf{S}'\mathbf{r}] \tag{25}$$

- Logdet is a bit harder

# Logdet estimator

We only need **gradient** of $\log|\mathbf{K}_{ZZ}|$ to train.

$$\begin{aligned}\frac{\partial \log|\mathbf{K}_{ZZ}|}{\partial \mathbf{K}_{ZZ}} &= \mathbf{K}_{ZZ}^{-1} = \mathbb{E}_{\mathbf{r}}\left[\mathbf{K}_{ZZ}^{-1}\mathbf{r}\mathbf{r}^{\top}\right] \\ &\approx \mathbf{K}_{ZZ}^{-1}\mathbf{r}\mathbf{r}^{\top} \end{aligned} \tag{26}$$

# Logdet estimator

We only need **gradient** of $\log|\mathbf{K}_{ZZ}|$ to train.

$$\begin{aligned}\frac{\partial \log|\mathbf{K}_{ZZ}|}{\partial \mathbf{K}_{ZZ}} &= \mathbf{K}_{ZZ}^{-1} = \mathbb{E}_{\mathbf{r}}\left[\mathbf{K}_{ZZ}^{-1}\mathbf{r}\mathbf{r}^{\top}\right] \\ &\approx \mathbf{K}_{ZZ}^{-1}\mathbf{r}\mathbf{r}^{\top}\end{aligned} \tag{26}$$

Use **Conjugate Gradient** to estimate $\mathbf{K}_{ZZ}^{-1}\mathbf{r}$:

$$\frac{\partial \log|\mathbf{K}_{ZZ}|}{\partial \mathbf{K}_{ZZ}} = \mathrm{CG}(\mathbf{K}_{ZZ}, \mathbf{r})\mathbf{r}^{\top} \tag{27}$$

# Logdet estimator

We only need **gradient** of $\log|\mathbf{K}_{ZZ}|$ to train.

$$\frac{\partial \log|\mathbf{K}_{ZZ}|}{\partial \mathbf{K}_{ZZ}} = \mathbf{K}_{ZZ}^{-1} = \mathbb{E}_{\mathbf{r}}\left[\mathbf{K}_{ZZ}^{-1}\mathbf{r}\mathbf{r}^{\mathsf{T}}\right] \approx \mathbf{K}_{ZZ}^{-1}\mathbf{r}\mathbf{r}^{\mathsf{T}} \tag{26}$$

Use **Conjugate Gradient** to estimate $\mathbf{K}_{ZZ}^{-1}\mathbf{r}$:

$$\frac{\partial \log|\mathbf{K}_{ZZ}|}{\partial \mathbf{K}_{ZZ}} = \mathrm{CG}(\mathbf{K}_{ZZ}, \mathbf{r})\mathbf{r}^{\mathsf{T}} \tag{27}$$

- CG is iterative, and in worst case costs $O(M^3)$ to find the inverse-vector product.

# Logdet estimator

We only need **gradient** of $\log|\mathbf{K}_{ZZ}|$ to train.

$$\frac{\partial \log|\mathbf{K}_{ZZ}|}{\partial \mathbf{K}_{ZZ}} = \mathbf{K}_{ZZ}^{-1} = \mathbb{E}_{\mathbf{r}}\left[\mathbf{K}_{ZZ}^{-1}\mathbf{r}\mathbf{r}^{\mathsf{T}}\right] \approx \mathbf{K}_{ZZ}^{-1}\mathbf{r}\mathbf{r}^{\mathsf{T}} \tag{26}$$

Use **Conjugate Gradient** to estimate $\mathbf{K}_{ZZ}^{-1}\mathbf{r}$:

$$\frac{\partial \log|\mathbf{K}_{ZZ}|}{\partial \mathbf{K}_{ZZ}} = \mathrm{CG}(\mathbf{K}_{ZZ}, \text{preconditioner} = \mathbf{T}, \mathbf{r})\mathbf{r}^{\mathsf{T}} \tag{27}$$

- CG is iterative, and in worst case costs $O(M^3)$ to find the inverse-vector product.
- If we use $\mathbf{T}$ as a preconditioner: At the optimum it will converge in a **single iteration** since $\mathbf{K}_{ZZ}\mathbf{T}^* = \mathbf{I}$!

# Log-concave bound: overview

$$\mathcal{L}' = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL} \tag{28}$$

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2 \tag{29}$$

Tricks:

# Log-concave bound: overview

$$\mathcal{L}' = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL} \tag{28}$$

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2 \tag{29}$$

Tricks:

- Upper bound to the predictive variance is lower bound to ELBO.

# Log-concave bound: overview

$$\mathcal{L}' = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL} \tag{28}$$

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2 \tag{29}$$

Tricks:

- Upper bound to the predictive variance is lower bound to ELBO.
- Introduce new parameter $\mathbf{T}$, with $\mathrm{argmax}_{\mathbf{T}}\, \mathcal{L}' = \mathbf{K}_{ZZ}^{-1}$.

# Log-concave bound: overview

$$\mathcal{L}' = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL} \tag{28}$$

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2 \tag{29}$$

Tricks:

- Upper bound to the predictive variance is lower bound to ELBO.
- Introduce new parameter $\mathbf{T}$, with $\mathrm{argmax}_{\mathbf{T}}\, \mathcal{L}' = \mathbf{K}_{ZZ}^{-1}$.
- Preconditioned conjugate gradient for gradient of $\log|\mathbf{K}_{ZZ}|$.

# Log-concave bound: overview

$$\mathcal{L}' = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL} \tag{28}$$

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2 \tag{29}$$

Tricks:

- Upper bound to the predictive variance is lower bound to ELBO.
- Introduce new parameter $\mathbf{T}$, with $\mathrm{argmax}_{\mathbf{T}}\, \mathcal{L}' = \mathbf{K}_{ZZ}^{-1}$.
- Preconditioned conjugate gradient for gradient of $\log|\mathbf{K}_{ZZ}|$.

# Log-concave bound: overview

$$\mathcal{L}' = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL} \tag{28}$$

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2 \tag{29}$$

Tricks:

- Upper bound to the predictive variance is lower bound to ELBO.
- Introduce new parameter $\mathbf{T}$, with $\mathrm{argmax}_{\mathbf{T}}\, \mathcal{L}' = \mathbf{K}_{ZZ}^{-1}$.
- Preconditioned conjugate gradient for gradient of $\log|\mathbf{K}_{ZZ}|$.

Properties:

# Log-concave bound: overview

$$\mathcal{L}' = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL} \tag{28}$$

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2 \tag{29}$$

Tricks:

- Upper bound to the predictive variance is lower bound to ELBO.
- Introduce new parameter $\mathbf{T}$, with $\mathrm{argmax}_{\mathbf{T}}\, \mathcal{L}' = \mathbf{K}_{ZZ}^{-1}$.
- Preconditioned conjugate gradient for gradient of $\log|\mathbf{K}_{ZZ}|$.

Properties:

- Recovers Hensman et al. (2013) at optimum $\mathbf{T} = \mathbf{K}_{ZZ}^{-1}$.

# Log-concave bound: overview

$$\mathcal{L}' = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL} \tag{28}$$

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2 \tag{29}$$

Tricks:

- Upper bound to the predictive variance is lower bound to ELBO.
- Introduce new parameter $\mathbf{T}$, with $\mathrm{argmax}_{\mathbf{T}}\, \mathcal{L}' = \mathbf{K}_{ZZ}^{-1}$.
- Preconditioned conjugate gradient for gradient of $\log|\mathbf{K}_{ZZ}|$.

Properties:

- Recovers Hensman et al. (2013) at optimum $\mathbf{T} = \mathbf{K}_{ZZ}^{-1}$.
- Convex in $\mathbf{T}$ — no new local optima

# Log-concave bound: overview

$$\mathcal{L}' = \mathbb{E}_{\mathcal{N}\left(f(\mathbf{x}_n);\mu_n,\bar{\sigma}_n^2\right)}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL} \tag{28}$$

$$\bar{\sigma}_n^2 = k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}^\top \mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n} \geqslant \sigma_n^2 \tag{29}$$

Tricks:

- Upper bound to the predictive variance is lower bound to ELBO.
- Introduce new parameter $\mathbf{T}$, with $\mathrm{argmax}_{\mathbf{T}}\, \mathcal{L}' = \mathbf{K}_{ZZ}^{-1}$.
- Preconditioned conjugate gradient for gradient of $\log|\mathbf{K}_{ZZ}|$.

Properties:

- Recovers Hensman et al. (2013) at optimum $\mathbf{T} = \mathbf{K}_{ZZ}^{-1}$.
- Convex in $\mathbf{T}$ — no new local optima
- Computational cost becomes $O(M^2)$ if $\mathbf{T}$ near its optimum.

# A fully variational bound

Log-concave bound has limitations. Can we find a "proper" variational bound?

# A fully variational bound

Log-concave bound has limitations. Can we find a "proper" variational bound?

Work backwards from predictive distribution:

$$\mathcal{N}\big(f_n; \quad \mathbf{k}_{\mathbf{x}_n Z}\mathbf{m}, \quad k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n}\big) \quad =$$
$$\mathcal{N}\big(f_n; \quad \mathbf{k}_{\mathbf{x}_n Z}\mathbf{m}, \quad k_{\mathbf{x}_n\mathbf{x}_n} - \mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{S}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n}\big)$$

$$\implies \mathbf{S} = \mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} - 2\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{S}'\mathbf{K}_{ZZ} \tag{30}$$

# A fully variational bound

Log-concave bound has limitations. Can we find a "proper" variational bound?

Work backwards from predictive distribution:

$$\mathcal{N}\left(f_n; \quad \mathbf{k}_{\mathbf{x}_n Z}\mathbf{m}, \quad k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n}\right) =$$
$$\mathcal{N}\left(f_n; \quad \mathbf{k}_{\mathbf{x}_n Z}\mathbf{m}, \quad k_{\mathbf{x}_n\mathbf{x}_n} - \mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{S}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n}\right)$$

$$\implies \mathbf{S} = \mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} - 2\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{S}'\mathbf{K}_{ZZ} \tag{30}$$

- Gives a 1-1 mapping between our inverse-free bound and original bound

# A fully variational bound

Log-concave bound has limitations. Can we find a "proper" variational bound?

Work backwards from predictive distribution:

$$\mathcal{N}\big(f_n; \quad \mathbf{k}_{\mathbf{x}_n Z}\mathbf{m}, \quad k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n}\big) =$$
$$\mathcal{N}\big(f_n; \quad \mathbf{k}_{\mathbf{x}_n Z}\mathbf{m}, \quad k_{\mathbf{x}_n\mathbf{x}_n} - \mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{S}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n}\big)$$

$$\implies \mathbf{S} = \mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} - 2\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{S}'\mathbf{K}_{ZZ} \tag{30}$$

- Gives a 1-1 mapping between our inverse-free bound and original bound
- By substituting $\mathbf{S}$ into the original bound, we get a **fully variational inverse-free bound**

# A fully variational bound

Log-concave bound has limitations. Can we find a "proper" variational bound?

Work backwards from predictive distribution:

$$\mathcal{N}\left(f_n; \quad \mathbf{k}_{\mathbf{x}_n Z}\mathbf{m}, \quad k_{\mathbf{x}_n\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} - 2\mathbf{k}_{\mathbf{x}_n Z}\mathbf{T}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{S}'\mathbf{k}_{Z\mathbf{x}_n}\right) =$$
$$\mathcal{N}\left(f_n; \quad \mathbf{k}_{\mathbf{x}_n Z}\mathbf{m}, \quad k_{\mathbf{x}_n\mathbf{x}_n} - \mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n} + \mathbf{k}_{\mathbf{x}_n Z}\mathbf{K}_{ZZ}^{-1}\mathbf{S}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z\mathbf{x}_n}\right)$$

$$\implies \mathbf{S} = \mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} - 2\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{S}'\mathbf{K}_{ZZ} \tag{30}$$

- Gives a 1-1 mapping between our inverse-free bound and original bound
- By substituting $\mathbf{S}$ into the original bound, we get a **fully variational inverse-free bound**
- Only KL term changes, can be dealt with in similar way

# Fully variational bound: overview

$$\mathcal{L} = \sum_n \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(f(Z))||p(f(Z))] \tag{31}$$

- Identical to Hensman et al. (2013) bound with substitution $\mathbf{S} = \mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} - 2\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{S}'\mathbf{K}_{ZZ}$

# Fully variational bound: overview

$$\mathcal{L} = \sum_n \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(f(Z))||p(f(Z))] \tag{31}$$

- Identical to Hensman et al. (2013) bound with substitution $\mathbf{S} = \mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} - 2\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{S}'\mathbf{K}_{ZZ}$
- Requires CG estimator for the logdet term.

# Fully variational bound: overview

$$\mathcal{L} = \sum_n \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(f(Z))||p(f(Z))] \tag{31}$$

- Identical to Hensman et al. (2013) bound with substitution $\mathbf{S} = \mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} - 2\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{S}'\mathbf{K}_{ZZ}$
- Requires CG estimator for the logdet term.
- Drop-in change for **any** variational GP model (e.g. deep GPs)

# Fully variational bound: overview

$$\mathcal{L} = \sum_n \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(f(Z))||p(f(Z))] \tag{31}$$

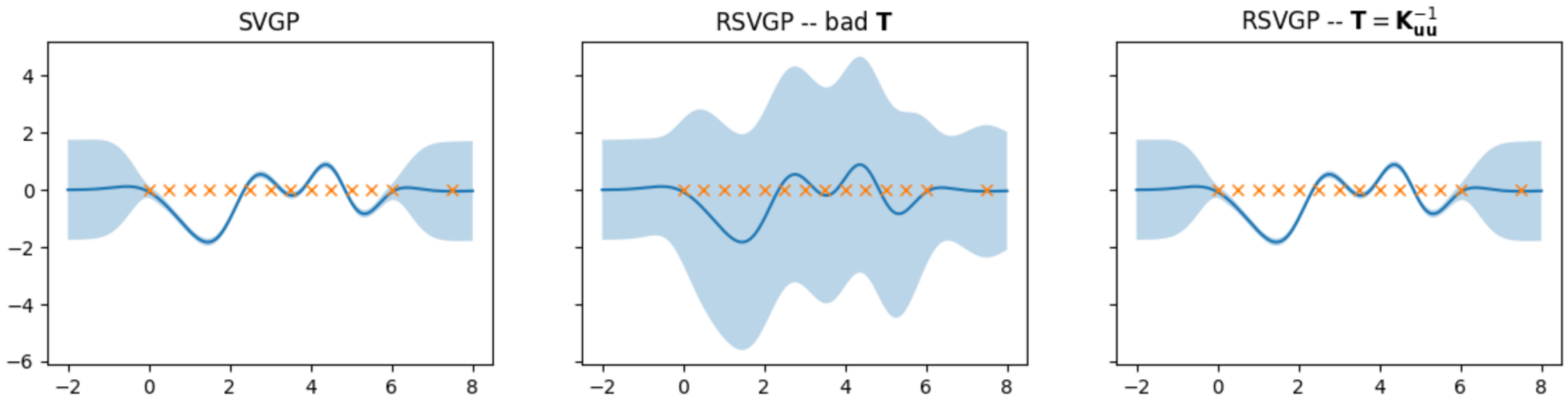
- Identical to Hensman et al. (2013) bound with substitution $\mathbf{S} = \mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} - 2\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{S}'\mathbf{K}_{ZZ}$
- Requires CG estimator for the logdet term.
- Drop-in change for **any** variational GP model (e.g. deep GPs)
- Only requires matrix-vector multiplies, $O(M^2)$ cost when $\mathbf{T} \approx \mathbf{K}_{ZZ}^{-1}$.

# Fully variational bound: overview

$$\mathcal{L} = \sum_n \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(f(Z))||p(f(Z))] \tag{31}$$

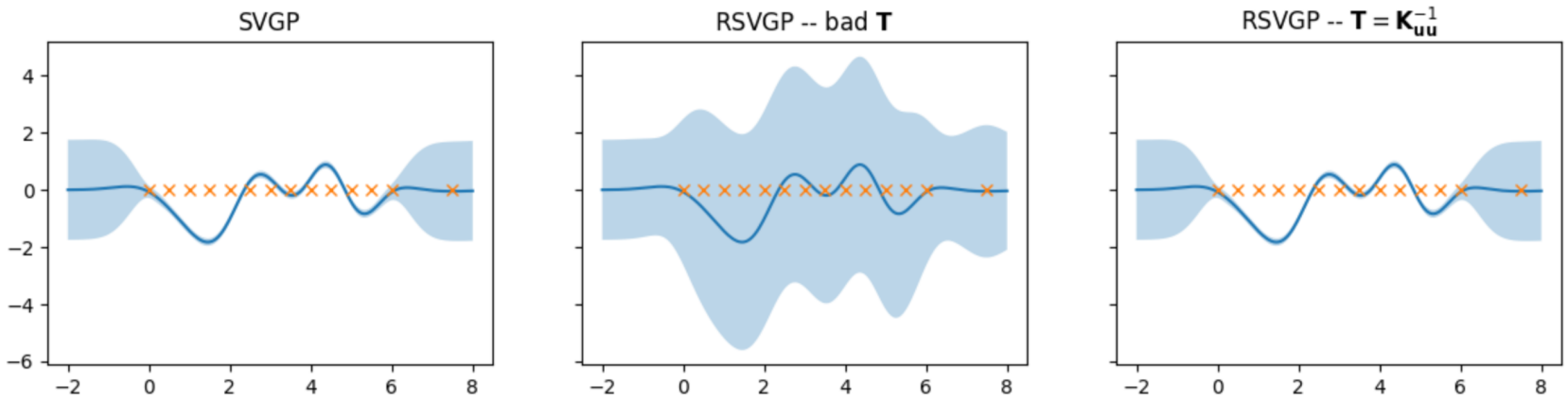
- Identical to Hensman et al. (2013) bound with substitution $\mathbf{S} = \mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} - 2\mathbf{K}_{ZZ}\mathbf{T}\mathbf{K}_{ZZ} + \mathbf{K}_{ZZ}\mathbf{S}'\mathbf{K}_{ZZ}$
- Requires CG estimator for the logdet term.
- Drop-in change for **any** variational GP model (e.g. deep GPs)
- Only requires matrix-vector multiplies, $O(M^2)$ cost when $\mathbf{T} \approx \mathbf{K}_{ZZ}^{-1}$.
- No additional gap when $\mathbf{T} = \mathbf{K}_{ZZ}^{-1}$.

# Toy 1D dataset

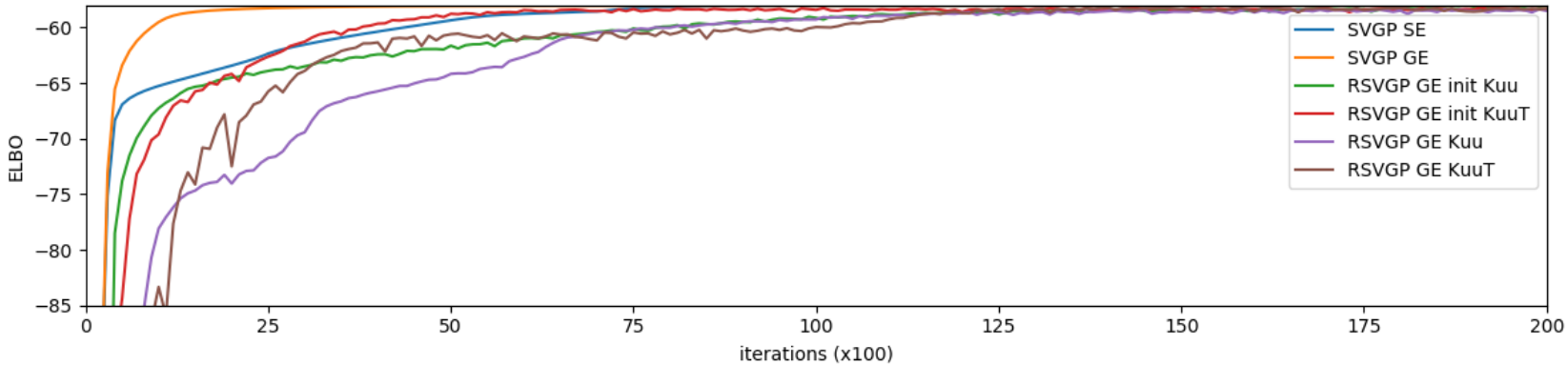



- Left: SVGP fit to data
- Middle: Inverse-free predictions with $\mathbf{T} = 0$
- Right: Optimised $\mathbf{T} = \mathbf{K}_{ZZ}^{-1}$

# Toy 1D dataset

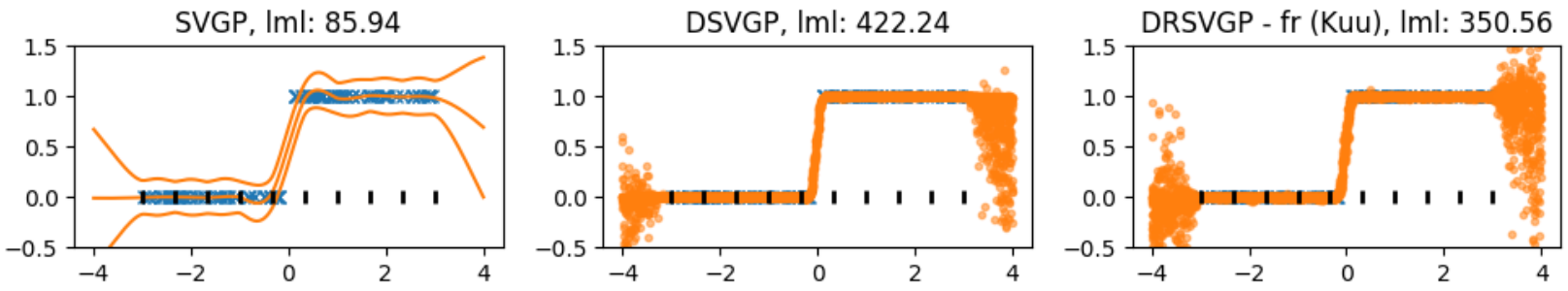


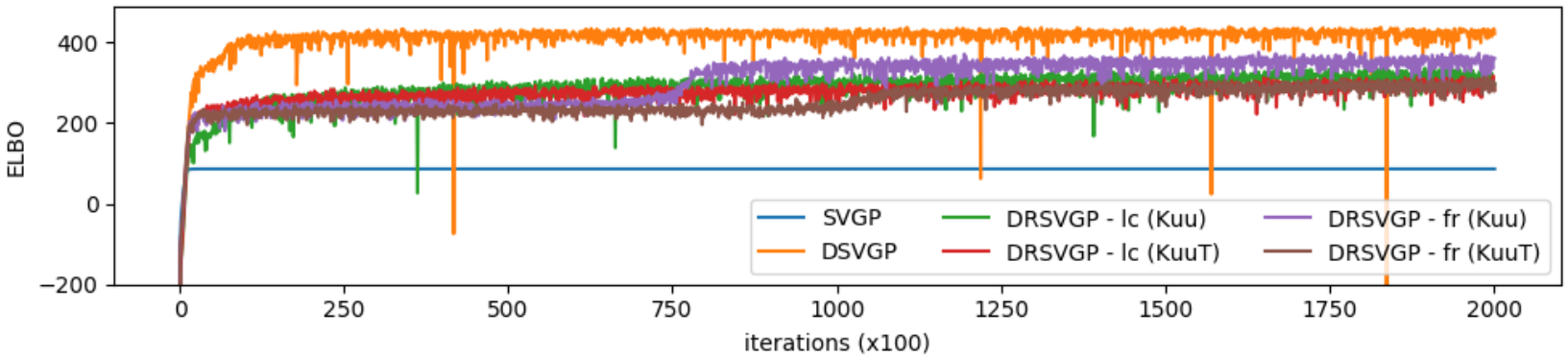

- Left: SVGP fit to data
- Middle: Inverse-free predictions with $\mathbf{T} = 0$
- Right: Optimised $\mathbf{T} = \mathbf{K}_{ZZ}^{-1}$



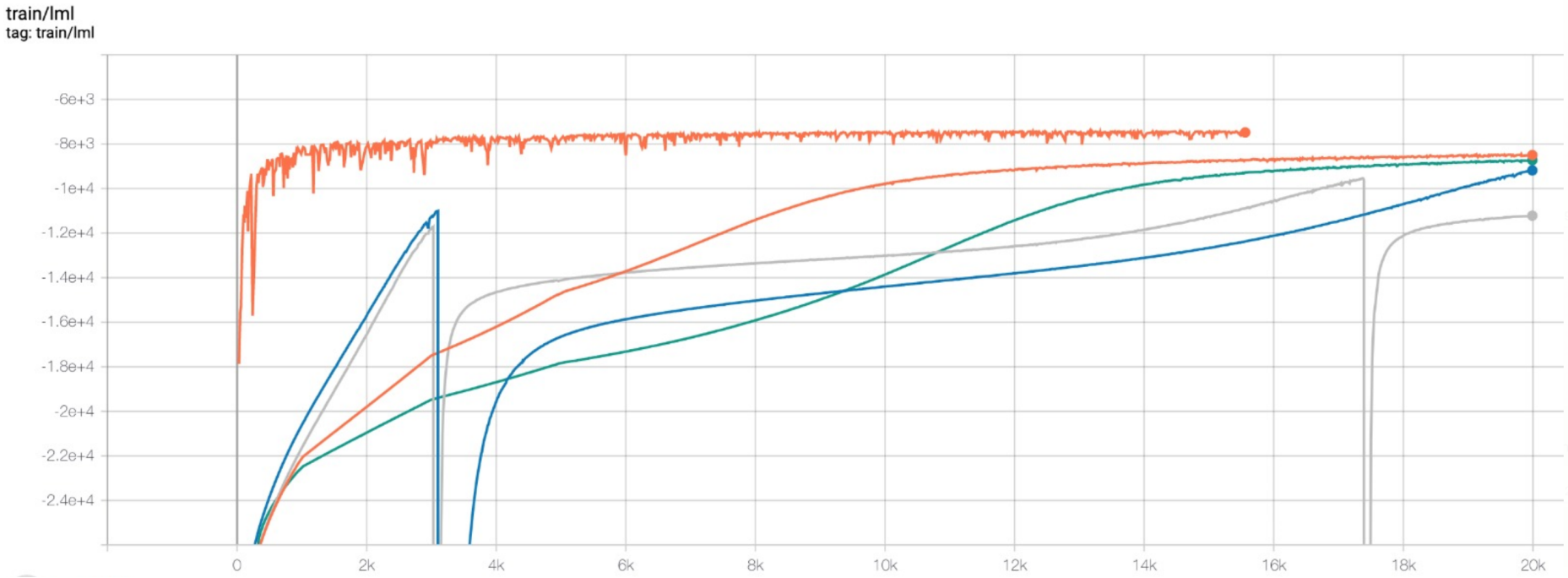
# Deep Gaussian process

## Fully variational bound



# The bad news




Orange: SVGP, Others: Inverse-free

Procedure: Optimise everything with Adam, including $\mathbf{T}$

- Less time per iteration
- Slower convergence
- Strange divergence behaviour

# Discussion

Work in progress because of the difficult optimisation behaviour.

Developments since AABI:

# Discussion

Work in progress because of the difficult optimisation behaviour.

Developments since AABI:

- Improved logdet estimators (no more CG inner loops)

# Discussion

Work in progress because of the difficult optimisation behaviour.

Developments since AABI:

- Improved logdet estimators (no more CG inner loops)
- Analysis of curvature of objective function gives hints into what causes behaviour

# Conclusions

- We have introduced inverse-free variational bounds to GP models
- We prove properties about their optima, and validate those experimentally
- However, a wall-clock speed-up in training is still elusive

# Conclusions

- We have introduced inverse-free variational bounds to GP models
- We prove properties about their optima, and validate those experimentally
- However, a wall-clock speed-up in training is still elusive

**Thanks for your attention!**

I'm curious about your thoughts!

# References I


Damianou, A. C. and Lawrence, N. D. (2013). Deep Gaussian processes. In Proceedings of the 16th International Conference on Artificial Intelligence and Statistics, pages 207–215.

Davies, A. (2015). Effective implementation of Gaussian process regression for machine learning. PhD thesis, University of Cambridge.

Frigola, R., Chen, Y., and Rasmussen, C. E. (2014). Variational gaussian process state-space models. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, Advances in Neural Information Processing Systems 27, pages 3680–3688. Curran Associates, Inc.

Gibbs, M. N. and MacKay, D. J. (1997). Efficient implementation of gaussian processes. Technical report.

# References II


Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI), pages 282–290.

Hensman, J., Matthews, A. G., and Ghahramani, Z. (2015). Scalable variational Gaussian process classification. In Proceedings of the 18th International Conference on Artificial Intelligence and Statistics, pages 351–360.

Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. In Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, pages 567–574.

Titsias, M. K. and Lawrence, N. D. (2010). Bayesian Gaussian process latent variable model. In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, pages 844–851.

# References III

van der Wilk, M., Bauer, M., John, S., and Hensman, J. (2018). Learning invariances using the marginal likelihood. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, Advances in Neural Information Processing Systems 31, pages 9938–9948. Curran Associates, Inc.

van der Wilk, M., Rasmussen, C. E., and Hensman, J. (2017). Convolutional gaussian processes. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, Advances in Neural Information Processing Systems 30, pages 2849–2858. Curran Associates, Inc.