StatML Workshop @ Amazon

Imperial College London

Automating Gaussian Process Approximations

Mark van der Wilk

Department of Computing
Imperial College London

@markvanderwilk
m.vdwilk@imperial.ac.uk

Apr 5, 2022

About our research group

- 2020–: Lecturer (Assistant Prof) at Imperial College London.
- Currently growing a research group.
- Research focus:
  - Gaussian process inference, backed by theory to make it **reliable**.
  - Automatic learning of inductive bias in neural networks.
    Central question: When should neurons be connected?

PhD Candidates

Artem Artemev

Jose Pablo Folch

Ruby Sedgwick

Seth Nabarro

Tycho van der Ouderaa

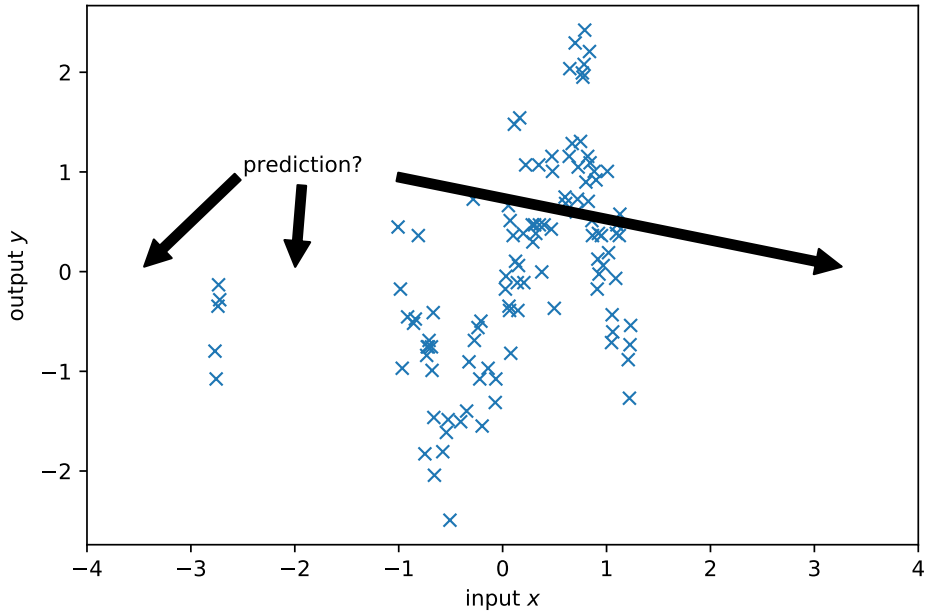
Regression

A lot of Machine Learning is just curve fitting.

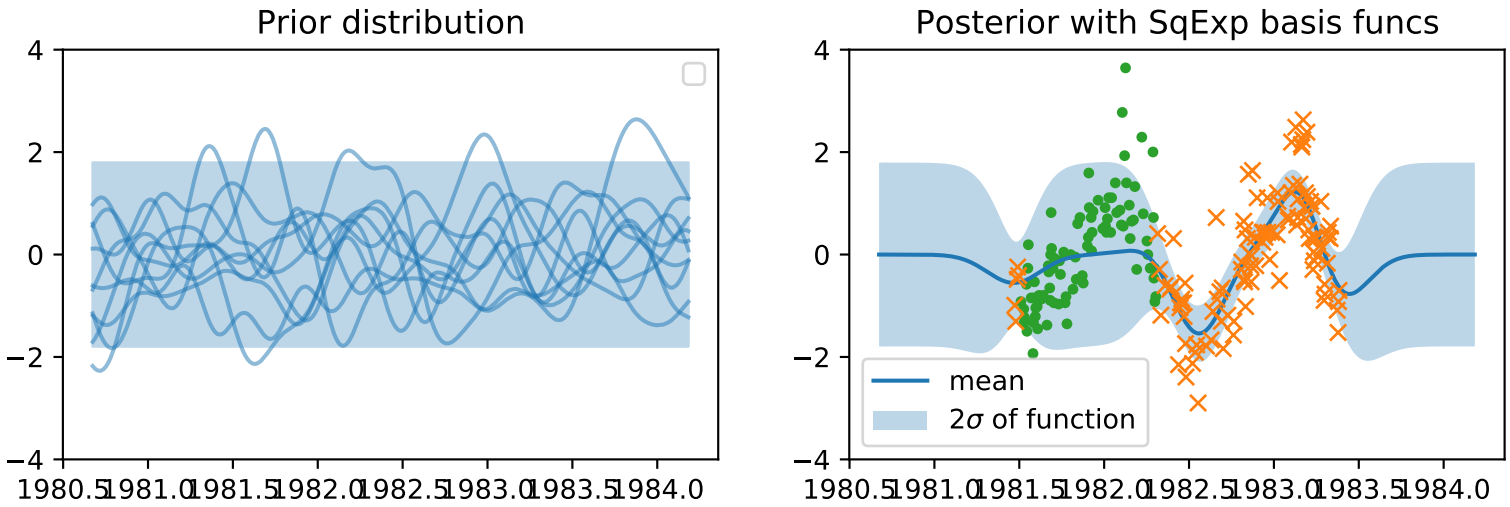
Given dataset $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$ related as $y_n = f(\mathbf{x}_n) + \epsilon_n$,

with $\mathbb{E}[\epsilon_n] = 0$,

find $f(\cdot)$.



Gaussian Process Regression



Gaussian processes are great because:

- they quantify **uncertainty**,
  which is good for decision-making,
- they are **automatic**,
  i.e. there are clear methods for setting parameters
  (e.g. in "hyperparameters" the prior).

Gaussian Process Inference

Performing regression with GPs requires two steps:

1. Finding the posterior given parameters of the prior

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

2. Finding the hyperparameters $\boldsymbol{\theta}$
   by maximising the marginal likelihood (MaxLik type-II):

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log p(\mathbf{y}|\boldsymbol{\theta}) \tag{2}$$

No gridsearch, no cross-validation, no trial-and-error

$\Longrightarrow$ **super convenient**.

Gaussian Process Approximations

Computations are hard because of:

- $O(N^3)$ computational cost for N datapoints,
- Non-conjugate inference (classification, deep, ...)

Approximations have been studied for **decades**...

- Eigenfunction / spectral decompositions

  (Ferrari-Trecate et al., 1998; Rahimi and Recht, 2008; Hensman et al., 2016; Dutordoir et al., 2020)

- Nyström / inducing points

  (Williams and Seeger, 2001; Seeger et al., 2003; Snelson and Ghahramani, 2005; Titsias, 2009; Hensman et al., 2013; Burt, Rasmussen, and van der Wilk, 2020)

- Conjugate Gradient methods

  (Gibbs and Mackay, 1997; Davies, 2015; Gardner et al., 2018; Artemev, Burt, and van der Wilk, 2021)

- Many, many more (structured matrices, sparse precision, ...)

Still no straightforward procedure!

Why aren't we finished yet?

- Consider the simplest case: **GP Regression**.
- Still no straightforward recommendation of what to do!
- Vast, almost incomprehensible literature of approximations!

Why so complicated?

- Approximations have parameters. User needs to set them.
- Papers don't tune properly (difficult and time-consuming).
- Difficult to evaluate properly.

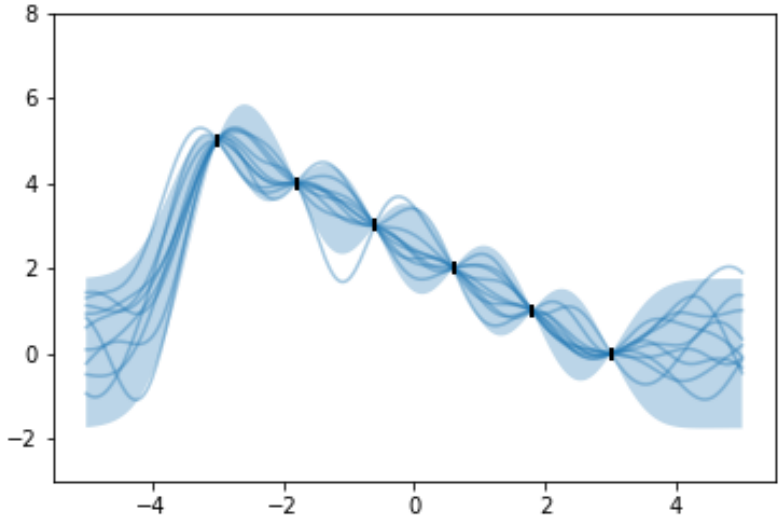
I want to share work on
Automating and Evaluating
Joint work with David Burt.

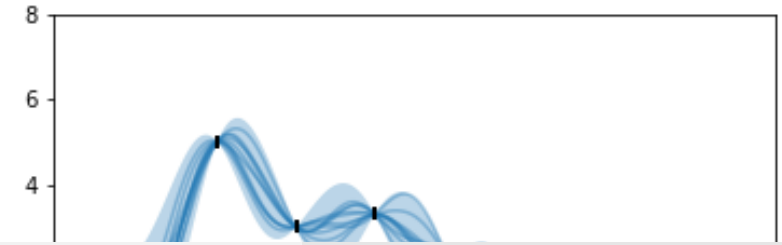
Approximation 1: Variational Inference

Define an approximate posterior:

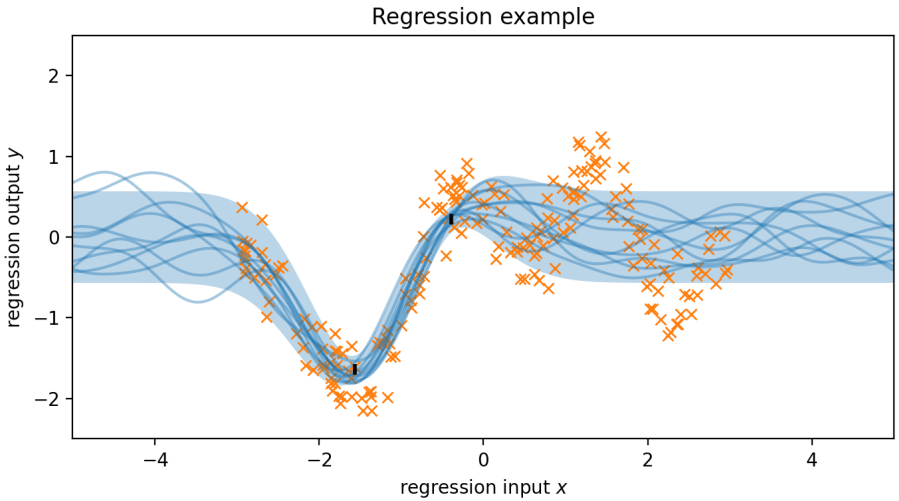
$$q(f_\cdot) = \int p(f_\cdot|f_Z)q(f_Z)\mathrm{d}f_Z$$

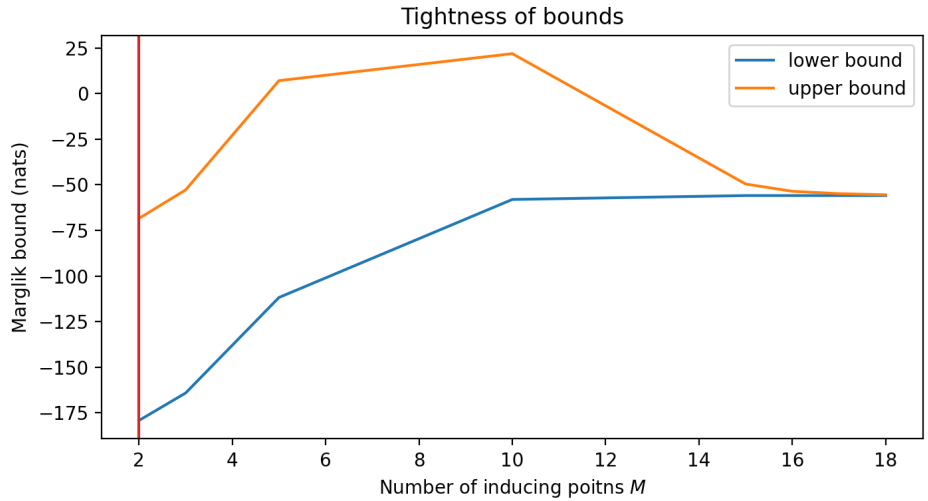
$$= \mathcal{N}\Big(\quad \mathrm{K}_{ZZ}^{-1}\mathrm{K}_{Z\cdot}\Big)$$

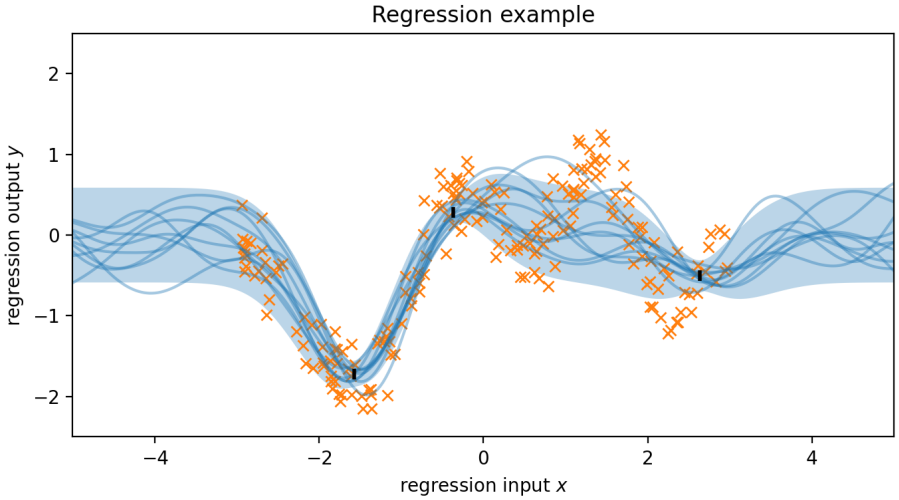


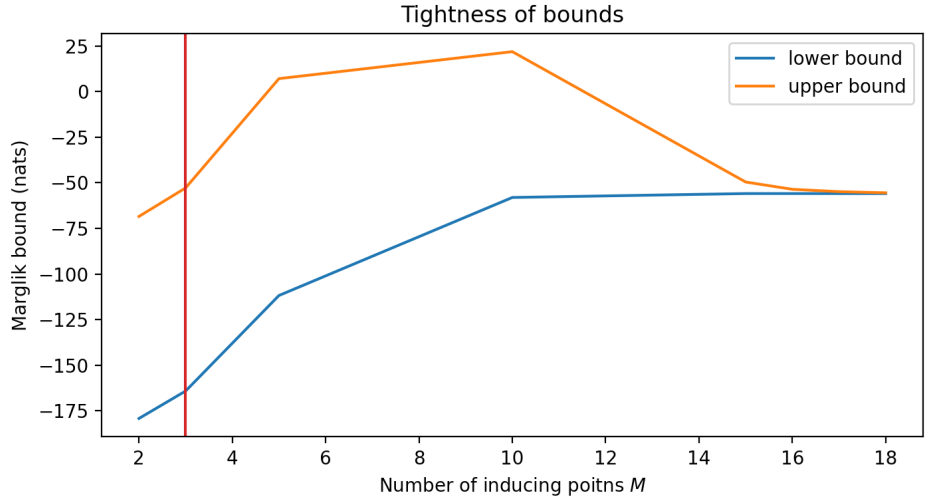


Approximation 1: Variational Inference









Variational Inference: Optimising Inducing Inputs

Finding inducing inputs Z is a major difficulty:

- How many inducing points to use?
  Left to the user. $\implies$ **not automatic!**
- Need to initialise Z. Subsample data? Gaussian? K-means?
  Folk wisdom. $\implies$ **not automatic!**
- Large number of parameters of Z $\implies$ slow convergence.
  May not even get close to optimal solution!
  Also need to decide how long to run for $\implies$ **not automatic!**
- How should number of inducing points grow with data N?

Theory provides solutions.

Variational Inference: Proofs of Accuracy

We (Burt, Rasmussen, and van der Wilk, 2019, 2020) set out to find out

- how quickly the number of inducing points would need to grow,
- with the dataset size N,
- for KL $\to 0$.

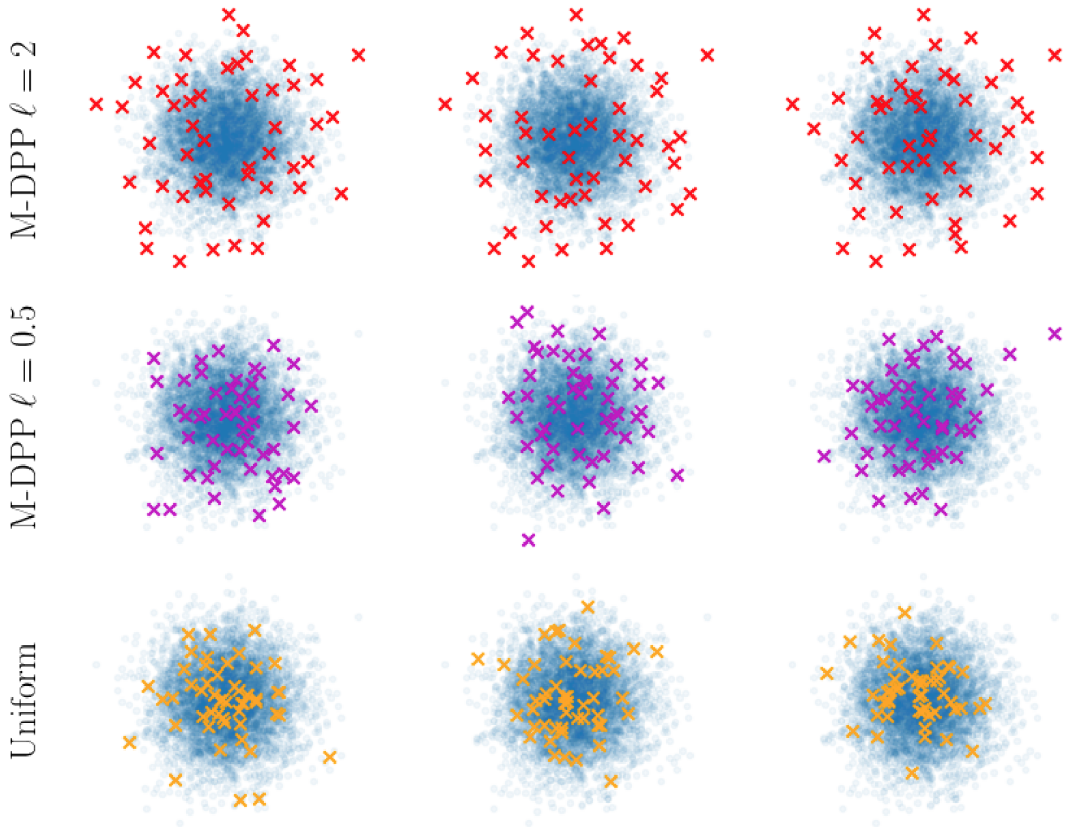
This requires making assumptions about:

- The input distribution: iid from some $p(\mathbf{x})$ (can weaken this).
- The function we're learning (some weak assumptions).
- **The method for selecting inducing inputs** Z.

We show this is the case
if we sample Z from an approximate M-DPP.

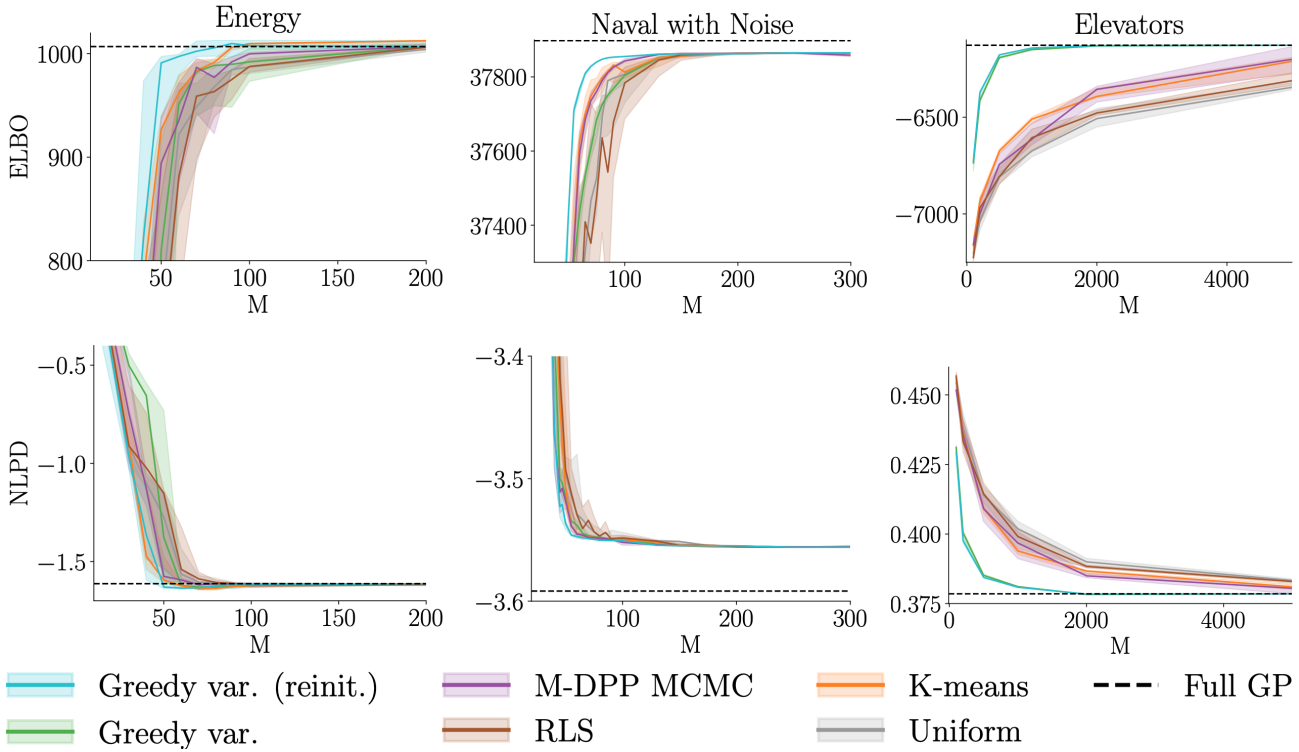
See theorems in Burt et al. (2019, 2020)

Variational Inference: Initialising Z



- *M*-DPP spreads out inducing points better than Uniform.
- Proof shows that gradient-based optimisation is not needed!
- For simplicity we use approximate *M*-DPP (no proof, empirical evidence only).

Variational Inference: Current Status



- "Greedy var" recovers GP performance quickest.
- No need to choose initialisation procedure. $\implies$ **automatic!**
- Optimisation now only over kernel and likelihood hyperparameters. BFGS actually converges. $\implies$ **automatic!**
- Final problem: How to select *number* of inducing points M.

Approximation 2: Conjugate Gradients

Training objective:

$$\mathcal{L} = c - \frac{1}{2}\log|\mathbf{K}_{\boldsymbol{\theta}}| - \frac{1}{2}\mathbf{y}^\top \mathbf{K}_{\boldsymbol{\theta}}^{-1}\mathbf{y}$$

$$\nabla_{\boldsymbol{\theta}}\mathcal{L} = -\frac{1}{2}\operatorname{Tr}\left(\mathbf{K}_{\boldsymbol{\theta}}^{-1}\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}\right) - \frac{1}{2}\mathbf{y}^\top \mathbf{K}_{\boldsymbol{\theta}}^{-1}\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}\mathbf{K}_{\boldsymbol{\theta}}^{-1}\mathbf{y}$$

Idea (Gibbs and Mackay, 1997; Davies, 2015; Gardner et al., 2018): Find $\mathbf{K}_{\boldsymbol{\theta}}^{-1}\mathbf{v}$ by solving:

$$\operatorname*{argmin}_{\mathbf{x}} \frac{1}{2}\mathbf{x}^\top \mathbf{K}_{\boldsymbol{\theta}}\mathbf{x} - \mathbf{x}^\top \mathbf{v}$$

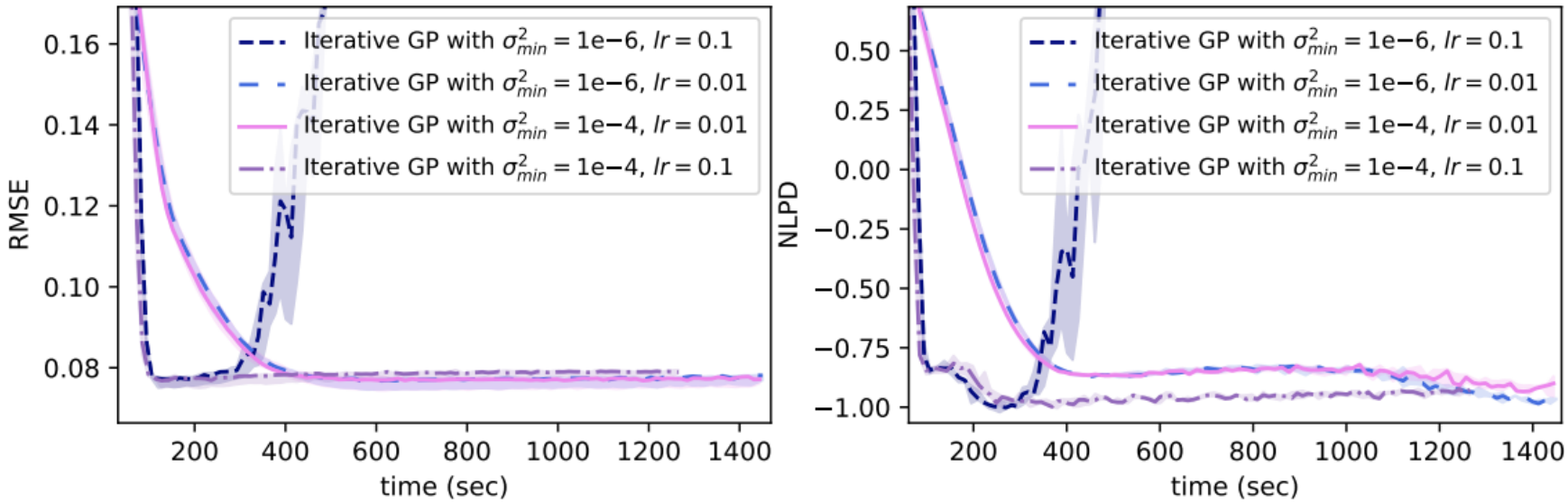
- **Conjugate Gradients** gives iterative solution, exact in the limit.
- May give better speed-accuracy trade-off than inducing points (particularly when $\mathbf{K}_{\boldsymbol{\theta}}$ not low-rank).
- Genuinely impressive results, e.g. *Exact GPs on a Million Data Points* (Wang et al., 2019).[1]

[1] However, I disagree that the method can be called exact.

Conjugate Gradients: Free parameters

- How does CG error influence the hyperparameter gradients?
- How many CG iterations to run for good behaviour?
- How many CG iterations to run for good accuracy-speed trade-off?

This has practical consequences, with behaviour that you would not expect from an exact method:



Conjugate Gradient Lower Bound

We (Artemev, Burt, and van der Wilk, 2021) develop the **Conjugate Gradient Lower Bound** (CGLB).
Idea:

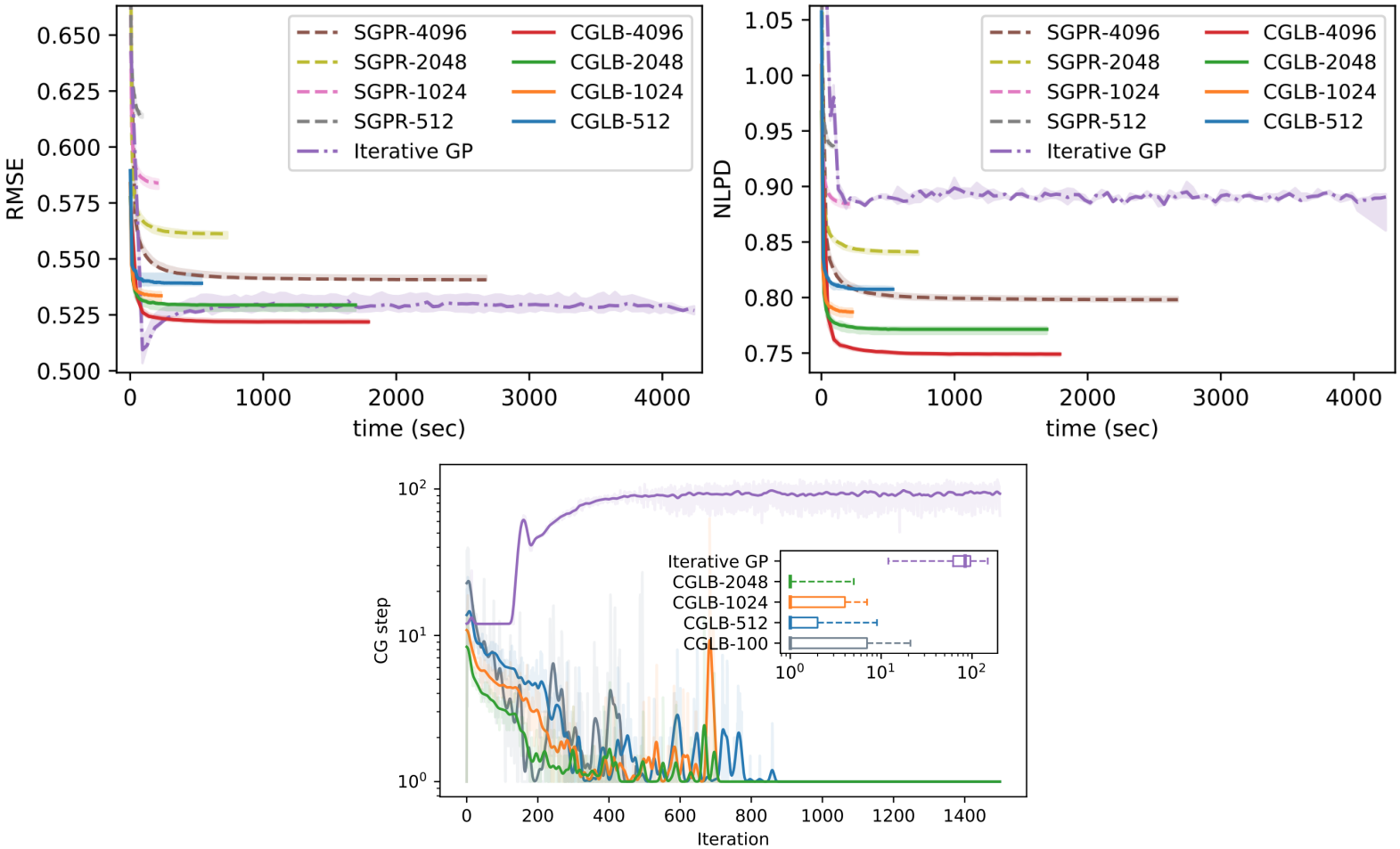
- Partial solution to inverse $\mathbf{x}$ is a *parameter* in the objective.
- This unifies CG optimisation to find inverse with hyperparameter optimisation! $\implies$ **prevents divergence**.
- Objective measures how close $\mathbf{x}$ is to $\mathbf{K}_{\boldsymbol{\theta}}^{-1}\mathbf{y}$ (like variational!)

$$\boldsymbol{\theta}^*, \mathbf{x}^* = \operatorname*{argmax}_{\boldsymbol{\theta}, \mathbf{x}} L(\boldsymbol{\theta}, \mathbf{x})$$

$$\text{with} \qquad \mathbf{x}^* = \operatorname*{argmax}_{\mathbf{x}} L(\boldsymbol{\theta}, \mathbf{x}) = \mathbf{K}^{-1}\mathbf{y}\,, \qquad \forall \boldsymbol{\theta}$$

- Additional upper bound on $L(\theta, \mathbf{x}^*) - L(\theta, \mathbf{x})$ to automatically determine number of CG iterations.
  (This stops CG when it is guaranteed within 1 nat of solution).

Conjugate Gradient Lower Bound



- Fewer iterations of CG $\implies$ **faster**.
- No divergence during optimisation $\implies$ **better performance**.
- No CG tolerance parameters $\implies$ **automatic!**

Conclusions

We want to run GP approximations to work transparently and automatically on a wide range of datasets!

- GP approximation is still open because methods are **not automatic enough**.
- Theoretical guarantees help with automating parameter selection (we saw this in variational and CG methods).
- Conjecture: Differences between similar approximations are down to uninteresting parameter tuning, which we want to automate away.

There is still work to be done.

- Selecting number of inducing points.
- Good software support (underrated but important!)
- Find benchmarks and demos (team up with industry?)
- Understand relationship between approximations and misspecification (ongoing work)

Evaluating GP Approximations

Under model-misspecification...

good approximation and good prediction

are not the same.

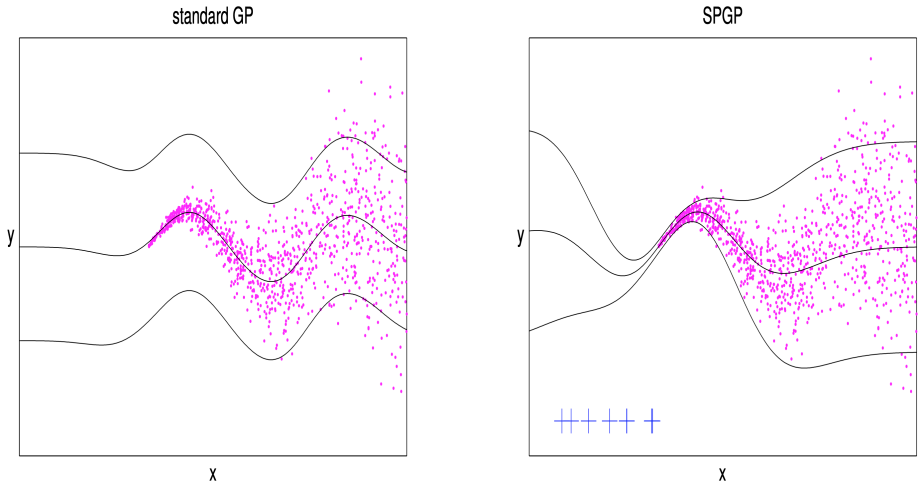


Figure: FITC can predict better than a GP, *because* it can be a bad approximation (From Snelson and Ghahramani, 2005).

- FITC predicts better than a GP *because* it can be a bad approximation (Bauer, van der Wilk, and Rasmussen, 2016).
- **Recommendation**: Researchers should measure approximation quality, not just performance.
- A discussion of any sensible metric is better than nothing!

Collaborate

- We need more benchmarks!
- Wide range of data scales, input dimensions, ...
- Test: To automatically fit all of them without intervention.

We are close to a solution

but *really* making things work is hard.

References I

Artem Artemev, David R. Burt, and Mark van der Wilk. Tighter bounds on the log marginal likelihood of gaussian process regression using conjugate gradients. In **Proceedings of the 38th International Conference on Machine Learning (ICML)**, 2021.

Matthias Stephan Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse gaussian process approximations. In **Advances in neural information processing systems**, 2016.

David Burt, Carl Edward Rasmussen, and Mark van der Wilk. Rates of convergence for sparse variational Gaussian process regression. In **Proceedings of the 36th International Conference on Machine Learning**, 2019.

David R. Burt, Carl Edward Rasmussen, and Mark van der Wilk. Convergence of sparse variational inference in gaussian processes regression. **Journal of Machine Learning Research**, 2020.

References II

Alex Davies. **Effective implementation of Gaussian Processes**. PhD thesis, University of Cambridge, 2015.

Vincent Dutordoir, Nicolas Durrande, and James Hensman. Sparse Gaussian processes with spherical harmonic features. In **Proceedings of the 37th International Conference on Machine Learning**, 2020.

Giancarlo Ferrari-Trecate, Christopher Williams, and Manfred Opper. Finite-dimensional approximation of gaussian processes. In **Advances in Neural Information Processing Systems 11**, 1998.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In **Advances in Neural Information Processing Systems 31**, 2018.

References III

Mark Gibbs and David Mackay. Efficient implementation of Gaussian processes. Technical report, Cavendish Laboratory, University of Cambridge, 1997.

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

James Hensman, Nicolas Durrande, and Arno Solin. Variational fourier features for gaussian processes. **arXiv preprint arXiv:1611.06740**, 2016.

Alexander G. de G. Matthews, James Hensman, Richard E. Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In **Proceedings of the 19th International Conference on Artificial Intelligence and Statistics**, pages 231–238, 2016.

References IV

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In **Advances in Neural Information Processing Systems 20**, 2008.

Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In **Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics**, 2003.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In **Advances in Neural Information Processing Systems 18**, pages 1257–1264, 2005.

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

References V

Ke Wang, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. In **Advances in Neural Information Processing Systems 32**, 2019.

Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In **Advances in Neural Information Processing Systems 13**, pages 682–688, 2001.