Computationally Efficient Gaussian Processes

Vincent Adam, University Pompeu Fabra

GPSS - 12th of September 2023

Outline

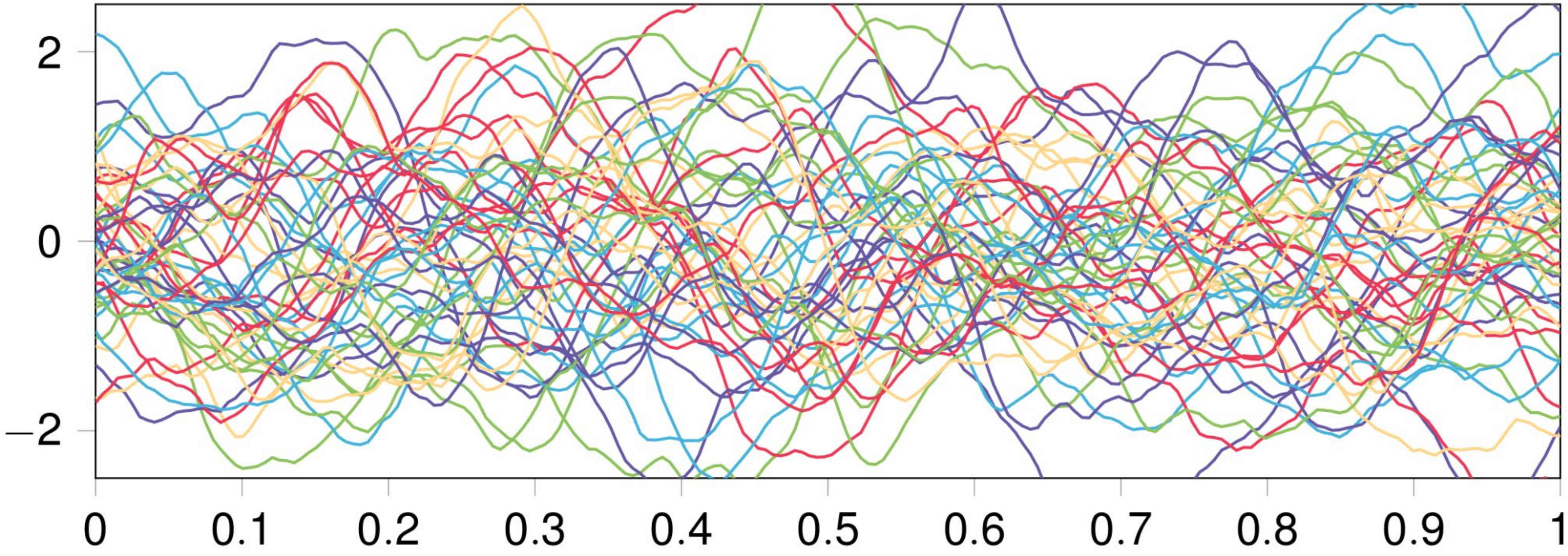
Part 1: Extension to non-Gaussian likelihoods

For non Gaussian observations, the posterior is intractable, we need approximations!

Part 2: Scaling up Gaussian process regression

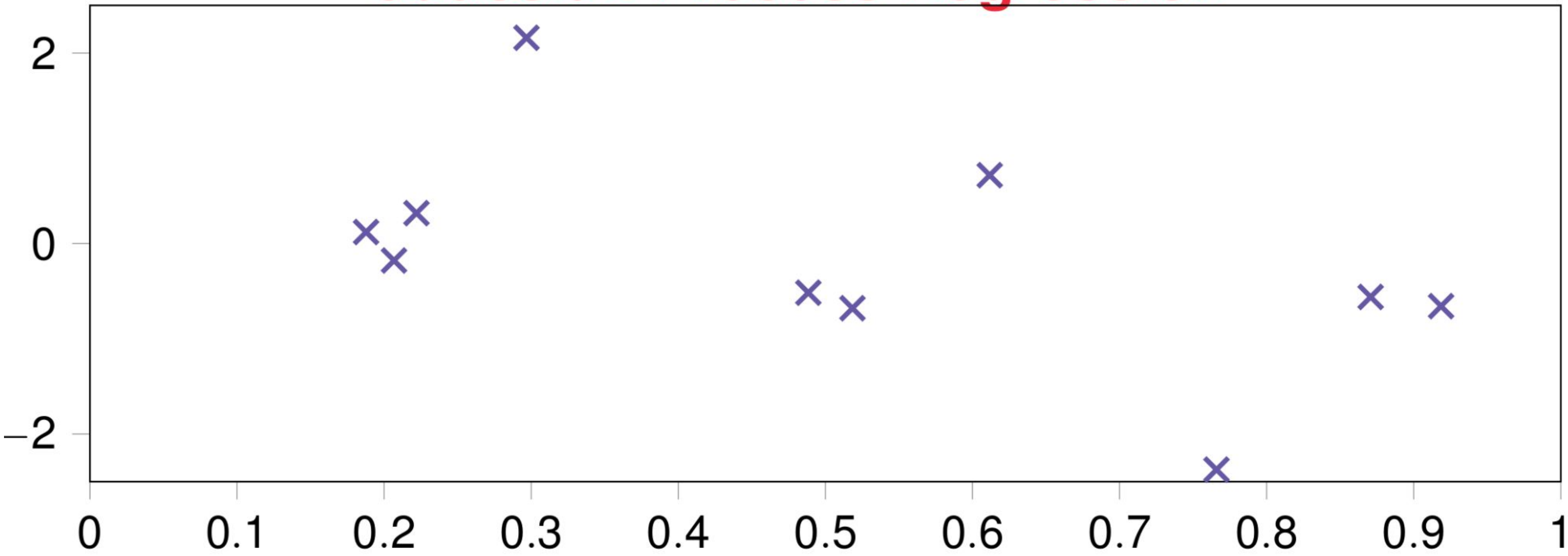
Or how to bypass the $O(N^3)$ computational bottleneck

Gaussian Process Regression



$$p(f(\cdot)) = \mathcal{GP}(0, k(\cdot, \cdot))$$

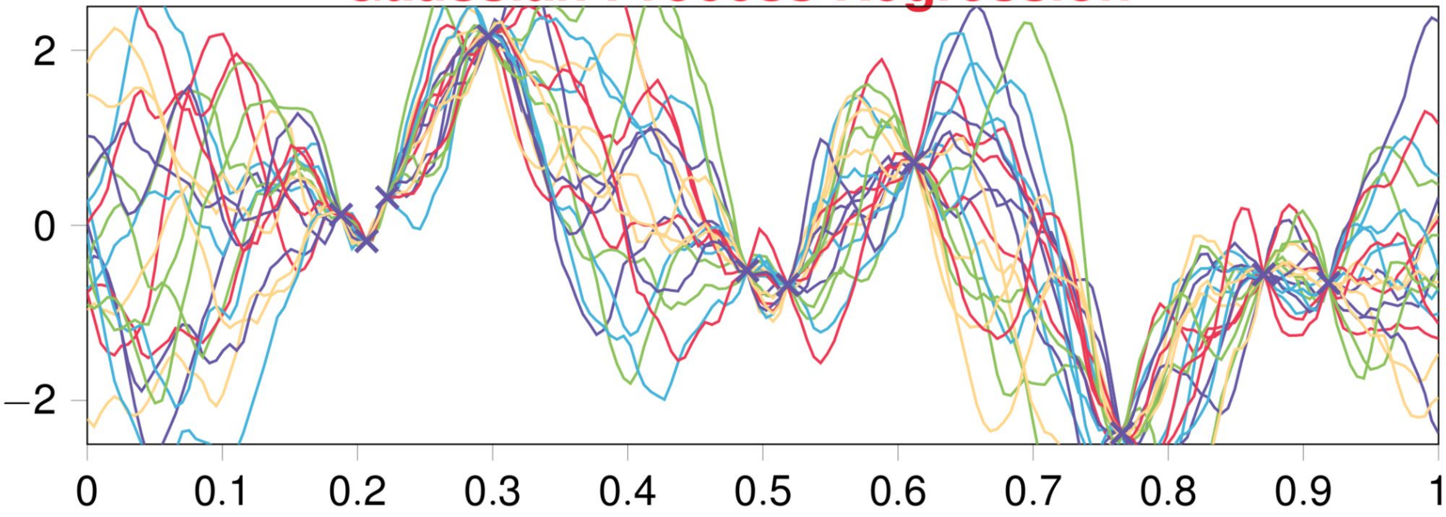
Gaussian Process Regression



$$p(f(\cdot)) = \mathcal{GP}(0, k(\cdot,\cdot))$$

$$p(\mathbf{y}|f(\cdot)) = \prod_{i=1}^{n} p(y_i|f(x_i))$$

Gaussian Process Regression

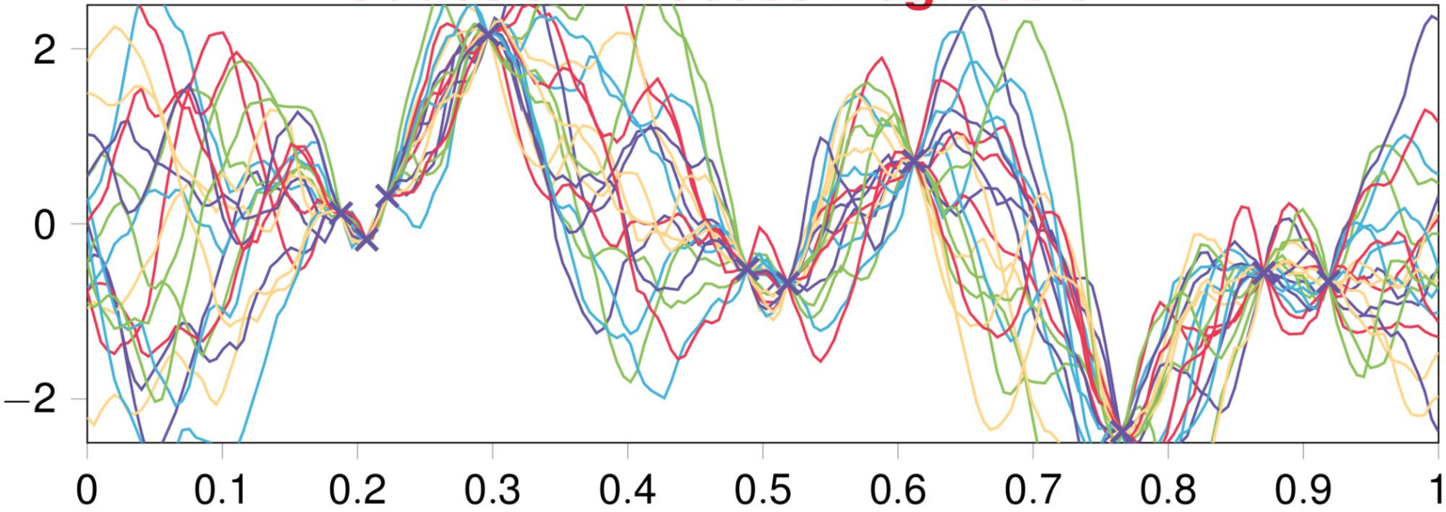


$$p(f(\cdot)) = \mathcal{GP}(\mathbf{0}, k(\cdot, \cdot))$$

$$p(\mathbf{y}|f(\cdot)) = \prod_{i=1}^{n} p(y_i|f(x_i))$$

$$p(f(\cdot)|\mathbf{y}) = \frac{p(\mathbf{y}|f(\cdot))p(f(\cdot))}{p(\mathbf{y})}$$

Gaussian Process Regression



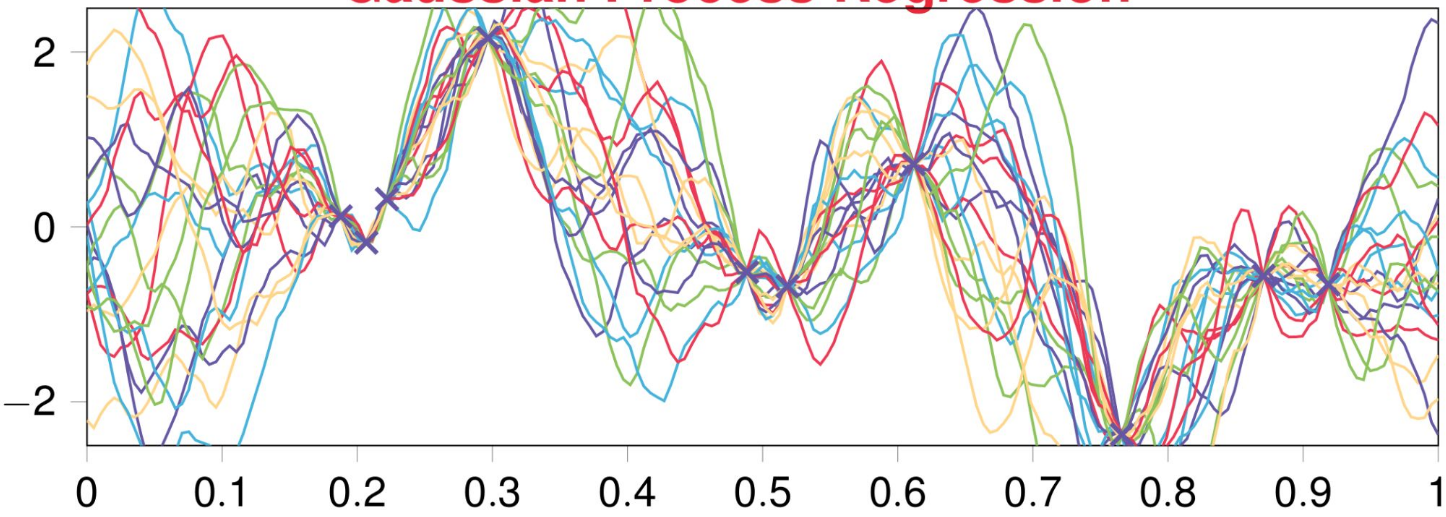
$$\mathbf{y}_i|f_i = f_i + \mathcal{N}(0, \sigma^2)$$

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; 0, \mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})$$

Objective for hyperparameter optimization

$$p(\mathbf{f}_*|\mathbf{y}) = \mathcal{N}(\mathbf{f}_*; \mathbf{K}_{\mathbf{f}_*\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{f}_*\mathbf{f}_*} - \mathbf{K}_{\mathbf{f}_*\mathbf{f}}(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{f}\mathbf{f}_*})$$

Gaussian Process Regression



$$p(\mathbf{y}_i|f_i) \text{ non Gaussian}$$

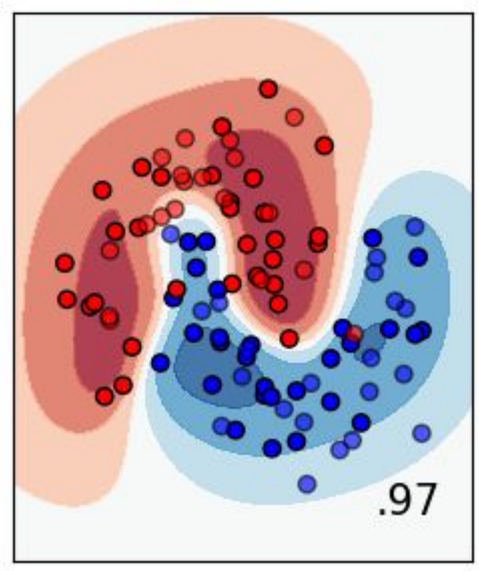
$$p(\mathbf{y}) = ???$$

$$p(\mathbf{f}_*|\mathbf{y}) = ???$$

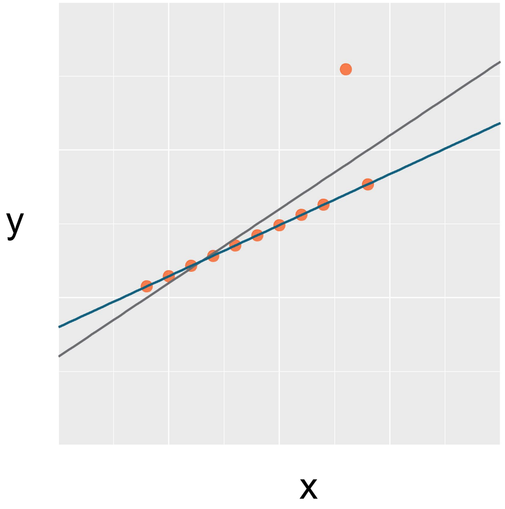
PART 1 - Extension to non-Gaussian likelihoods

Motivation

Beyond Gaussian regression …



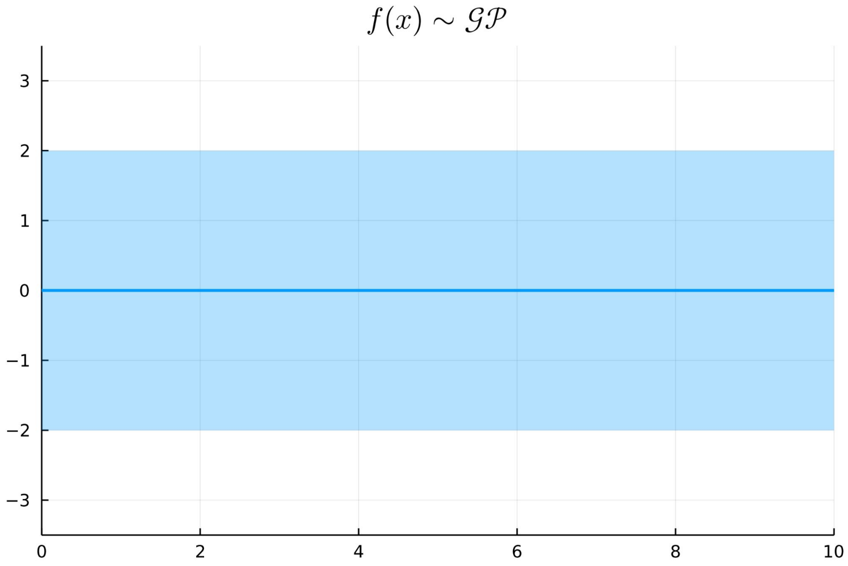
Classification



Robust Regression

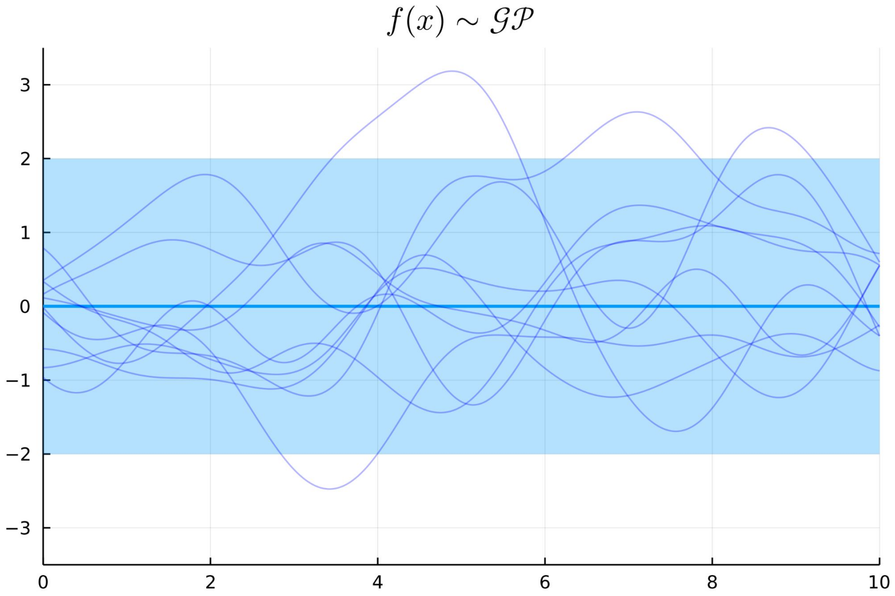
GP classification: the generative model

$$f(\cdot) \sim \mathcal{GP}(0, k)$$
$$y_i | f(x_i) \sim \text{Bernoulli}\big(\sigma(f(x_i))\big)$$



GP classification: the generative model

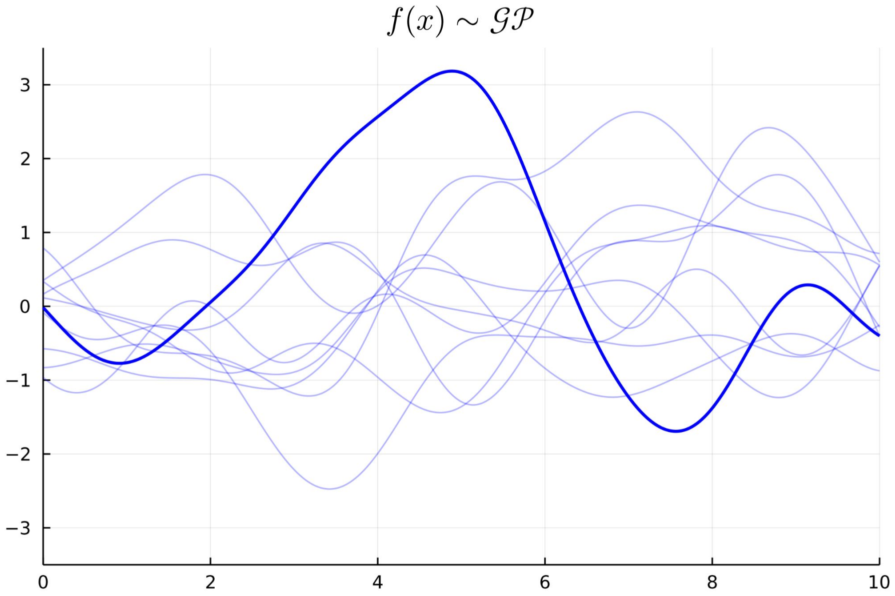
$$f(\cdot) \sim \mathcal{GP}(0, k)$$
$$y_i | f(x_i) \sim \text{Bernoulli}\big(\sigma(f(x_i))\big)$$



GP classification: the generative model

$$f(\cdot) \sim \mathcal{GP}(0, k)$$

$$y_i | f(x_i) \sim \mathrm{Bernoulli}\big(\sigma(f(x_i))\big)$$



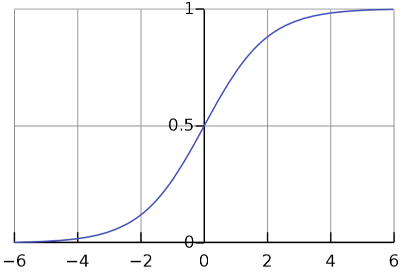
f(x) ∈ ℝ

GP classification: the generative model

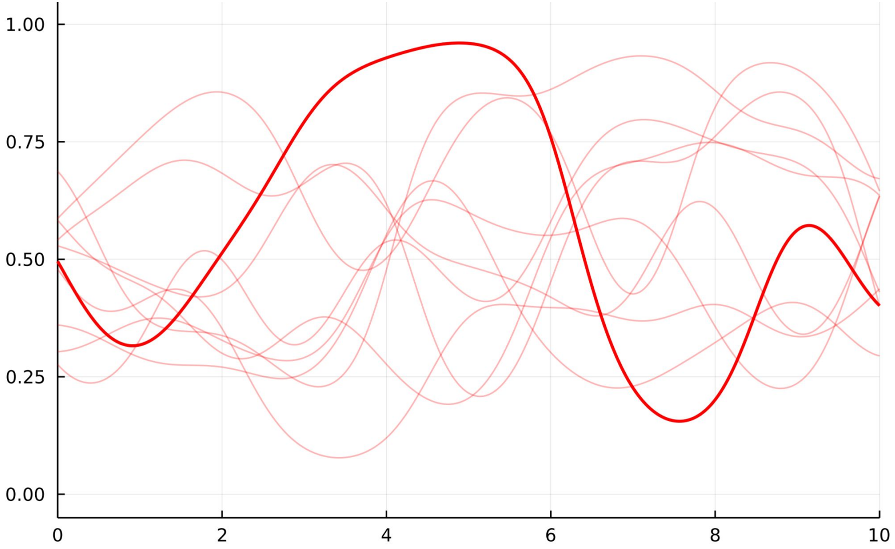
$$f(\cdot) \sim \mathcal{GP}(0, k)$$

$$y_i | f(x_i) \sim \text{Bernoulli}(\sigma(f(x_i)))$$

σ = link function



σ(f(x)) ∈ [0, 1]

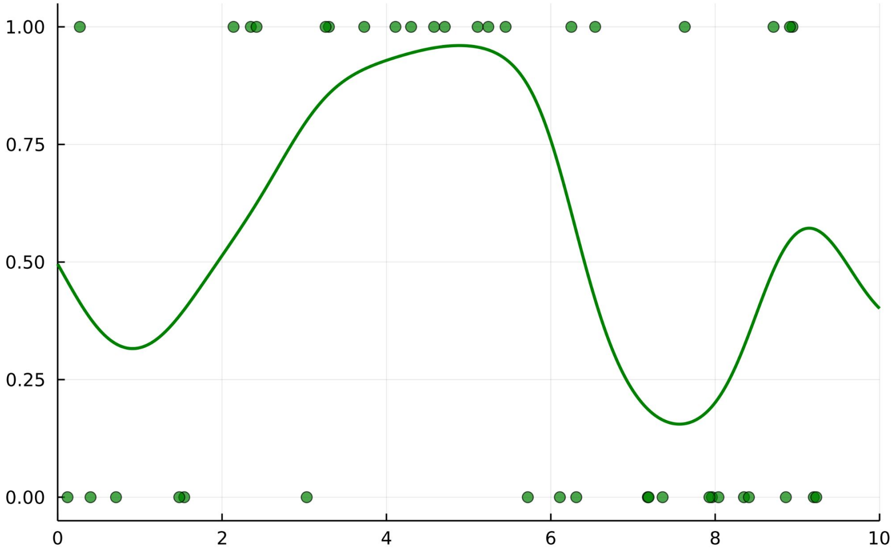


GP classification: the generative model

$$f(\cdot) \sim \mathcal{GP}(0, k)$$

$$y_i | f(x_i) \sim \text{Bernoulli}\big(\sigma(f(x_i))\big)$$

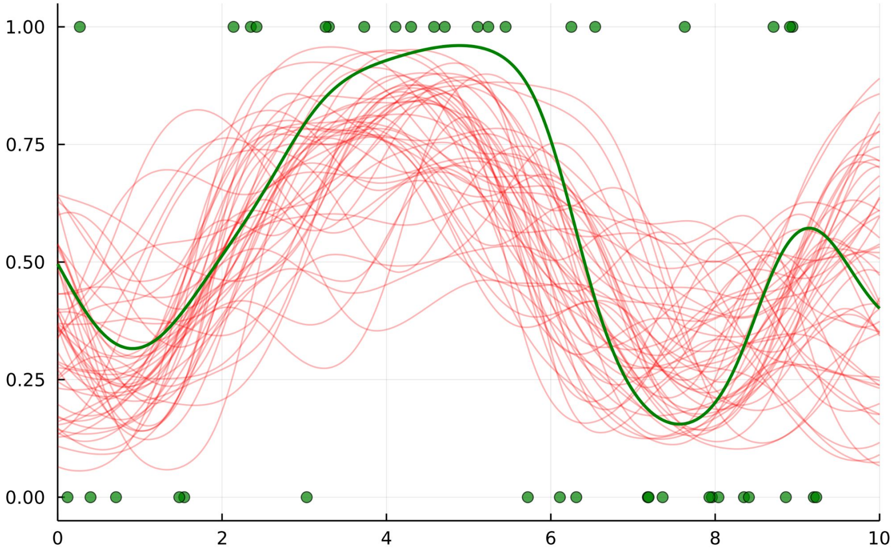
y(x) ∈ {0, 1}



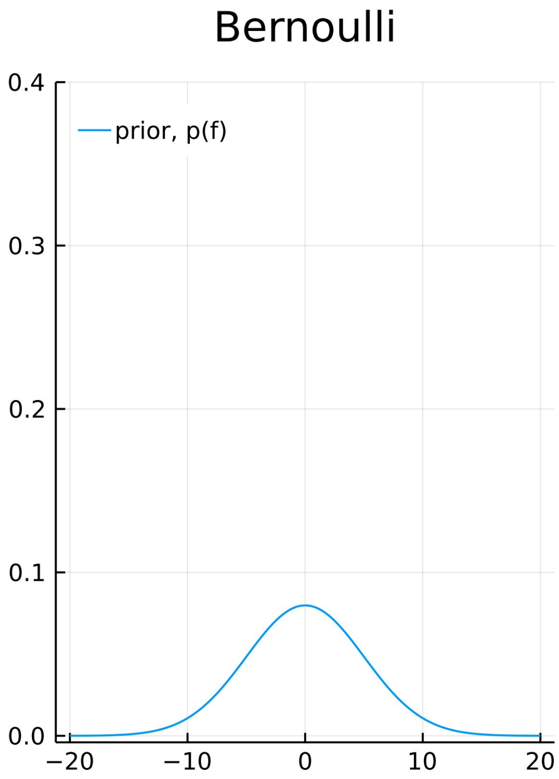
GP classification: inference

$$f(\cdot) \sim \mathcal{GP}(0, k)$$
$$y_i | f(x_i) \sim \text{Bernoulli}\big(\sigma(f(x_i))\big)$$

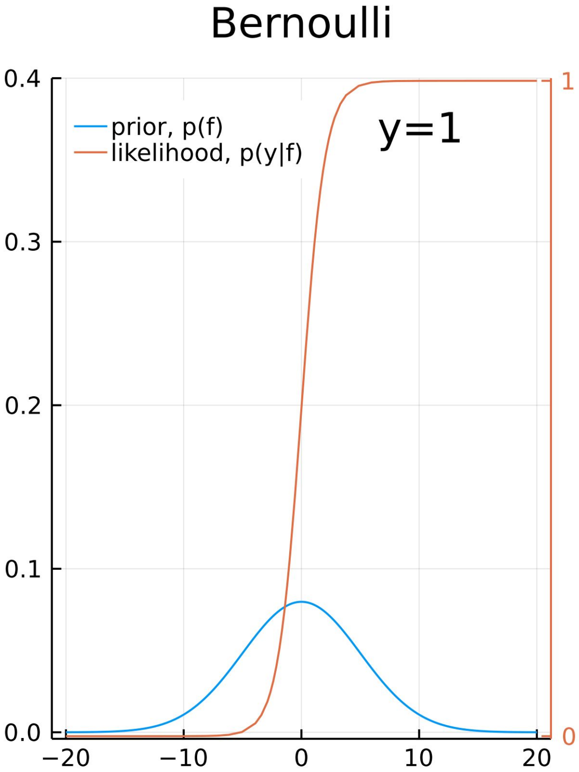
y(x*) ∈ {0, 1}



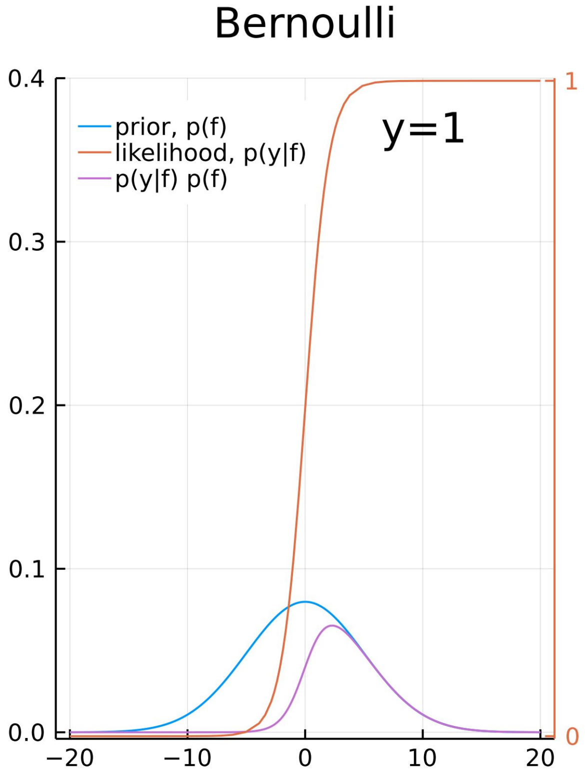
Non Gaussian likelihoods - what happens to the posterior?



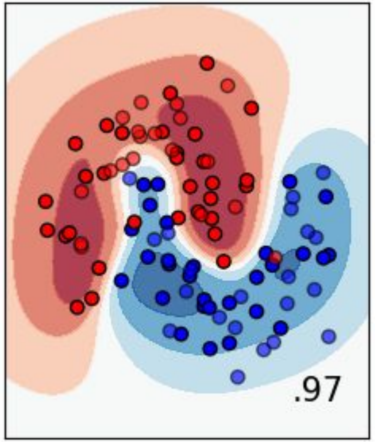
Non Gaussian likelihoods - what happens to the posterior?

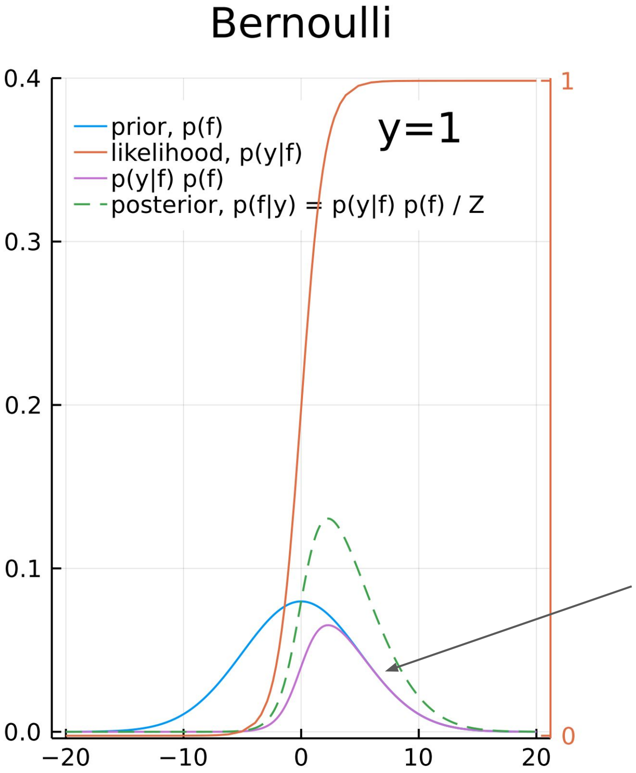


Non Gaussian likelihoods - what happens to the posterior?



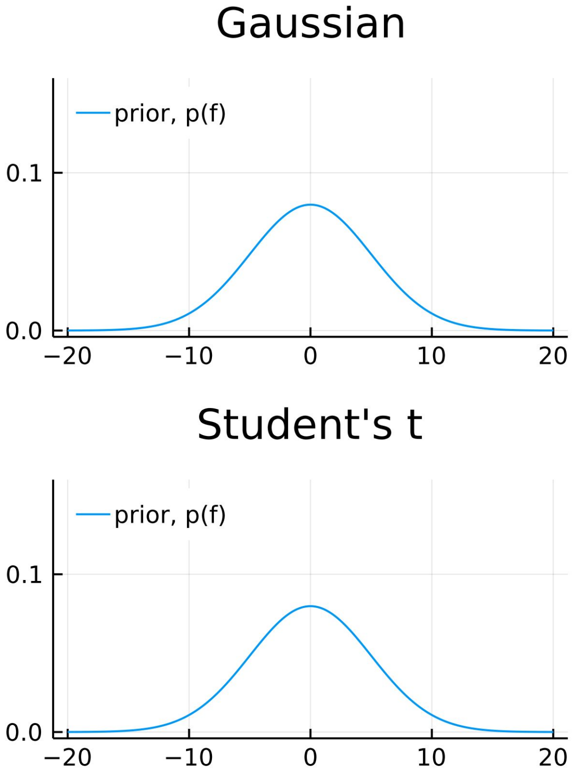
Non Gaussian likelihoods - what happens to the posterior?



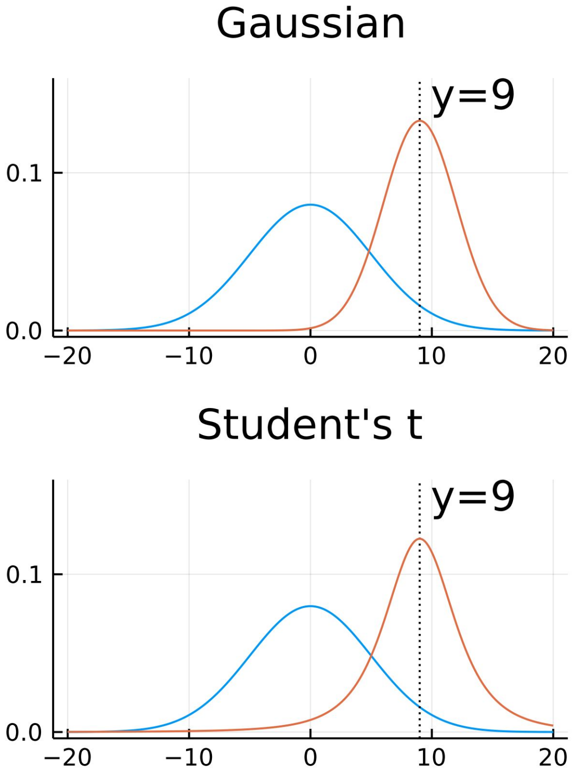


Not a standard distribution!

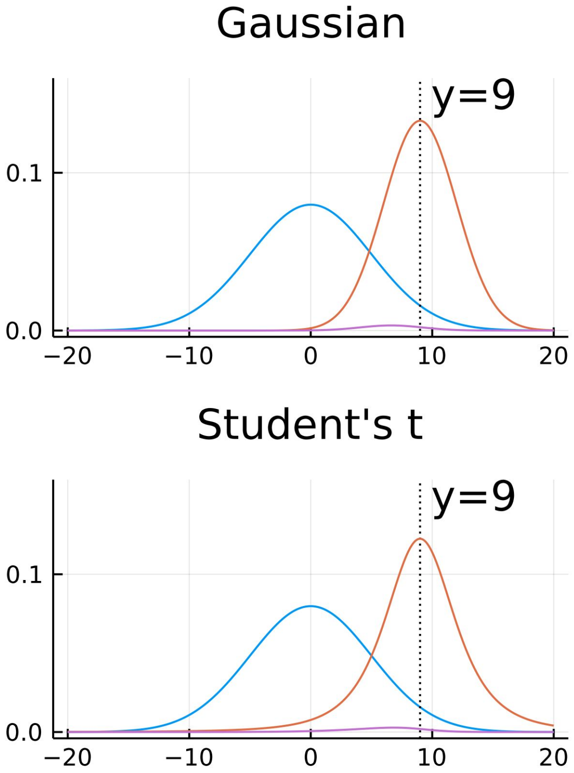
Non Gaussian likelihoods - what happens to the posterior?



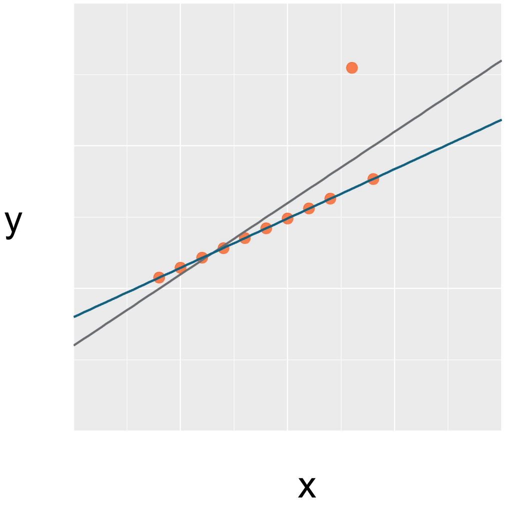
Non Gaussian likelihoods - what happens to the posterior?

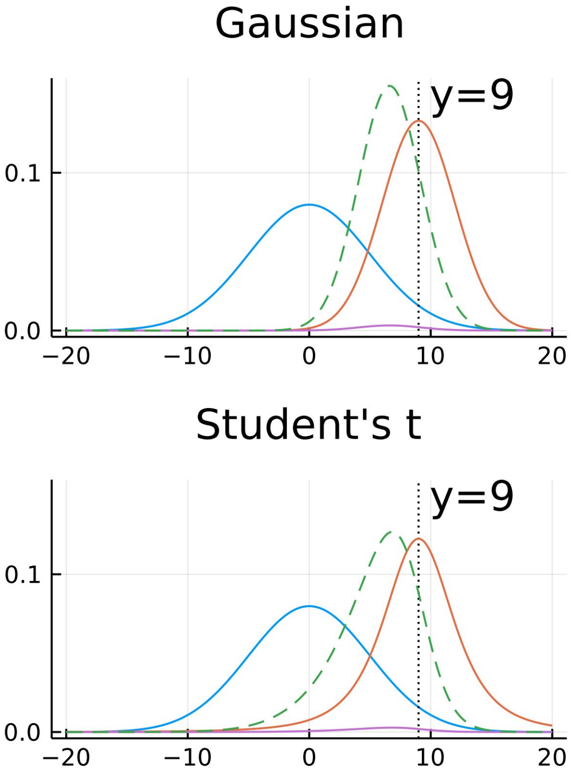


Non Gaussian likelihoods - what happens to the posterior?



Non Gaussian likelihoods - what happens to the posterior?





Why is it a problem?

For learning

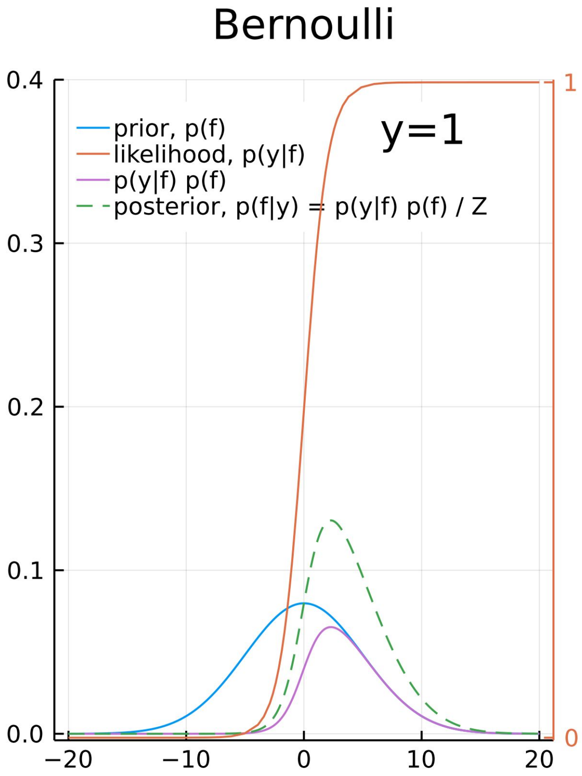
$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})\mathrm{d}\mathbf{f}$$

For inference

$$p(\mathbf{f} \mid \mathbf{y}) = \frac{p(\mathbf{f})p(\mathbf{y} \mid \mathbf{f})}{p(\mathbf{y})}$$

For predictions (or any posterior expectation)

$$p(f(x^*)) = \int p(f(x^*)|\mathbf{f})p(\mathbf{f}|\mathbf{y})\mathrm{d}\mathbf{f}$$



How to approximate the intractable posterior?

Parametric approximations

Most common: approximate the posterior as a ***Gaussian***

- Laplace approximation
- Variational inference
- Expectation propagation

Stochastic approximations

Draw ***samples*** *from the posterior*

Monte carlo Markov chains - *I won't cover today*

Why gaussian approximations to the posterior

Posterior approximation

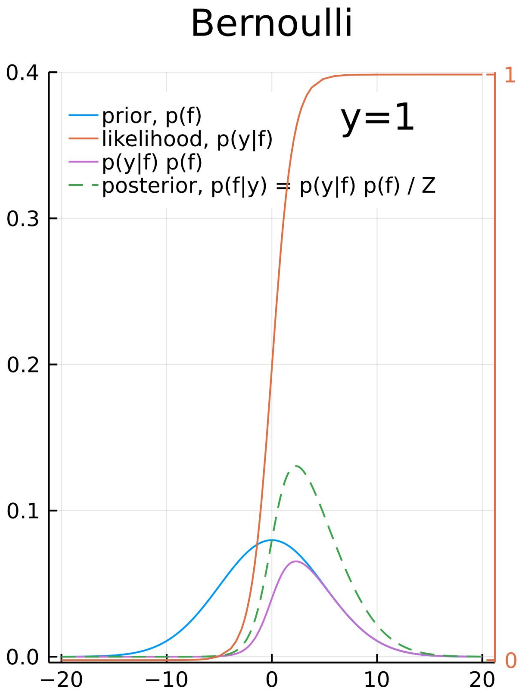
$$p(\mathbf{f}|\mathbf{y}) \approx q(\mathbf{f}) = \mathcal{N}(\mathbf{f}; \mathbf{m_f}, \mathbf{S_f})$$

For predictions (or any posterior expectation)

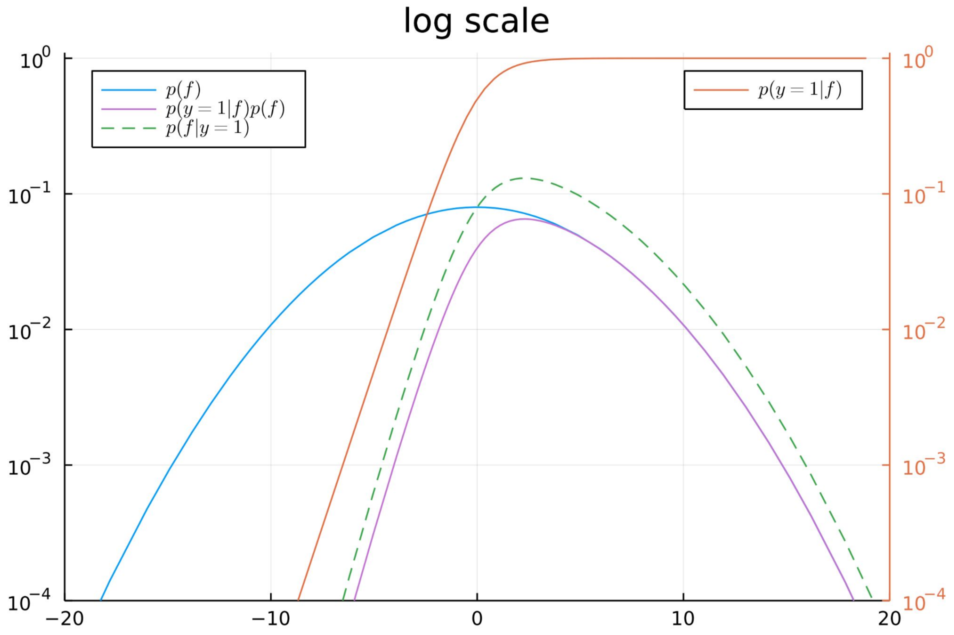
$$\begin{aligned} p(f(x^*)) &= \int p(f(x^*)|\mathbf{f})p(\mathbf{f}|\mathbf{y})\mathrm{d}\mathbf{f} \\ &\approx \int p(f(x^*)|\mathbf{f})q(\mathbf{f})\mathrm{d}\mathbf{f} \\ &= \mathcal{N}\left(f^* \,|\, \mathbf{b}_*^\top \mathbf{m_f}, \kappa_{**} - \mathbf{b}_*^\top (\mathbf{K_{ff}} - \mathbf{S_f})\mathbf{b}_*\right) \end{aligned}$$

$$\mathbf{b}_*^\top = \mathbf{k}_{*\mathbf{f}}\mathbf{K}_{\mathbf{ff}}^{-1}$$

Laplace approximation: the idea

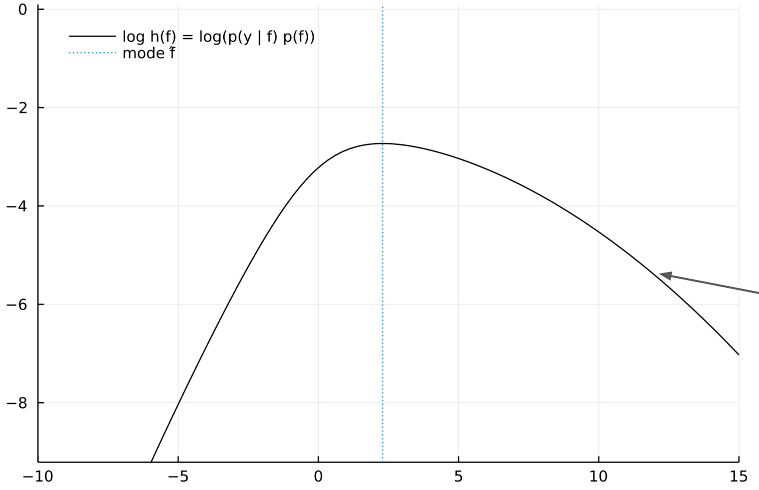


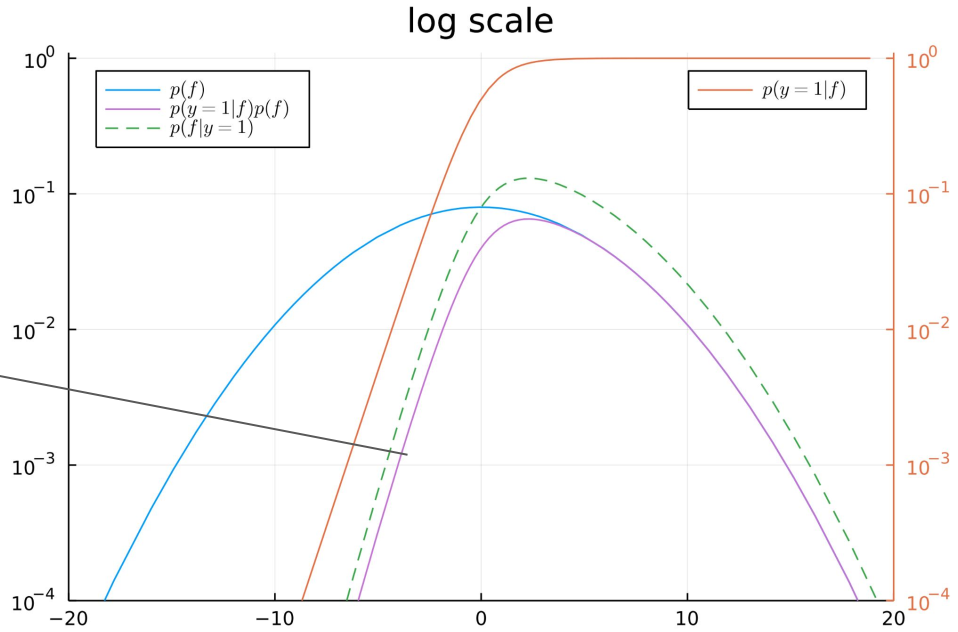
log



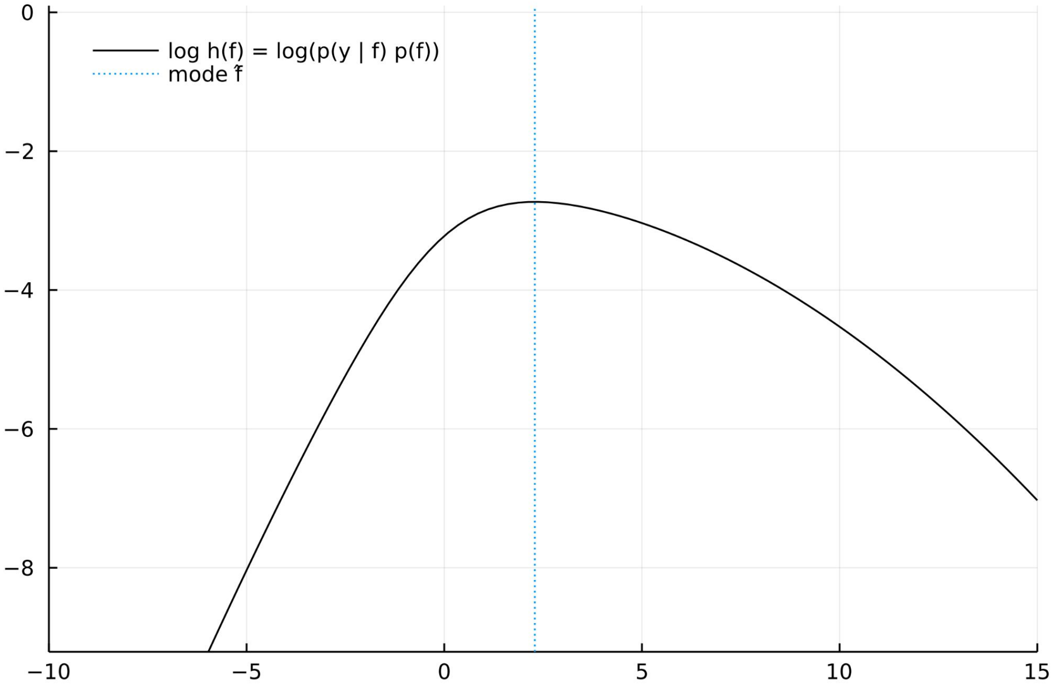
Laplace approximation: the idea

Posterior

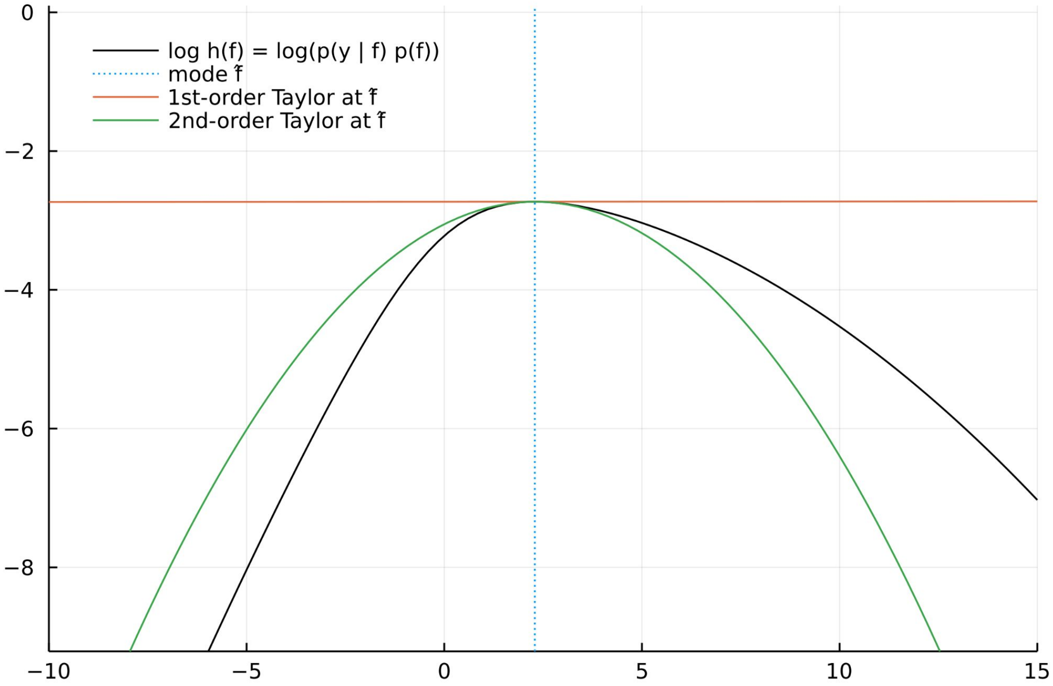




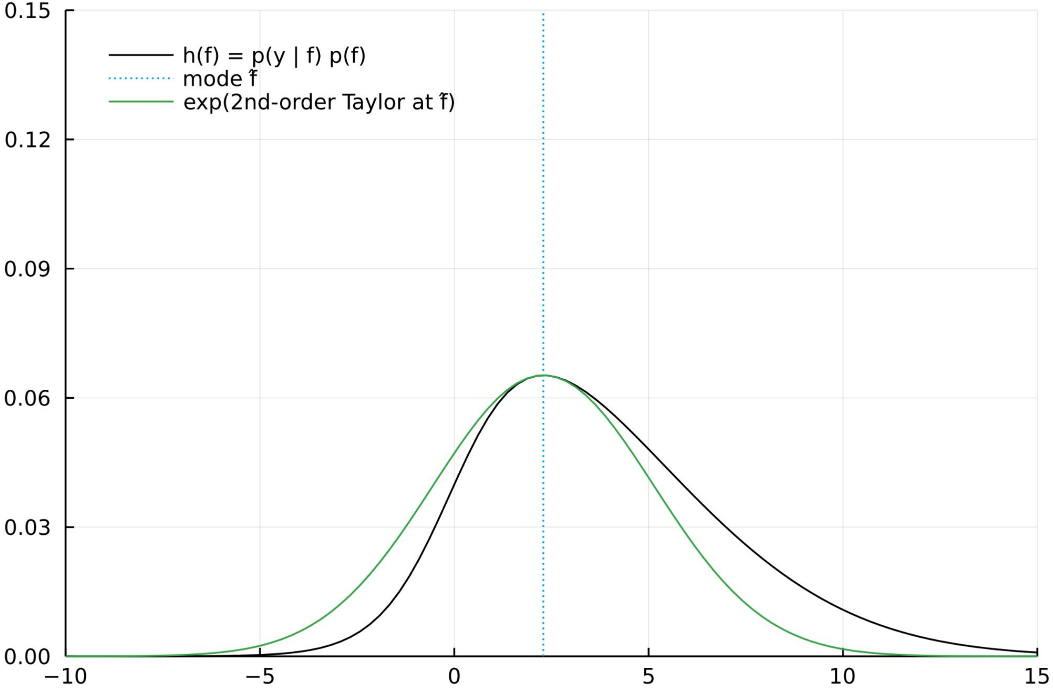
Laplace approximation: the idea



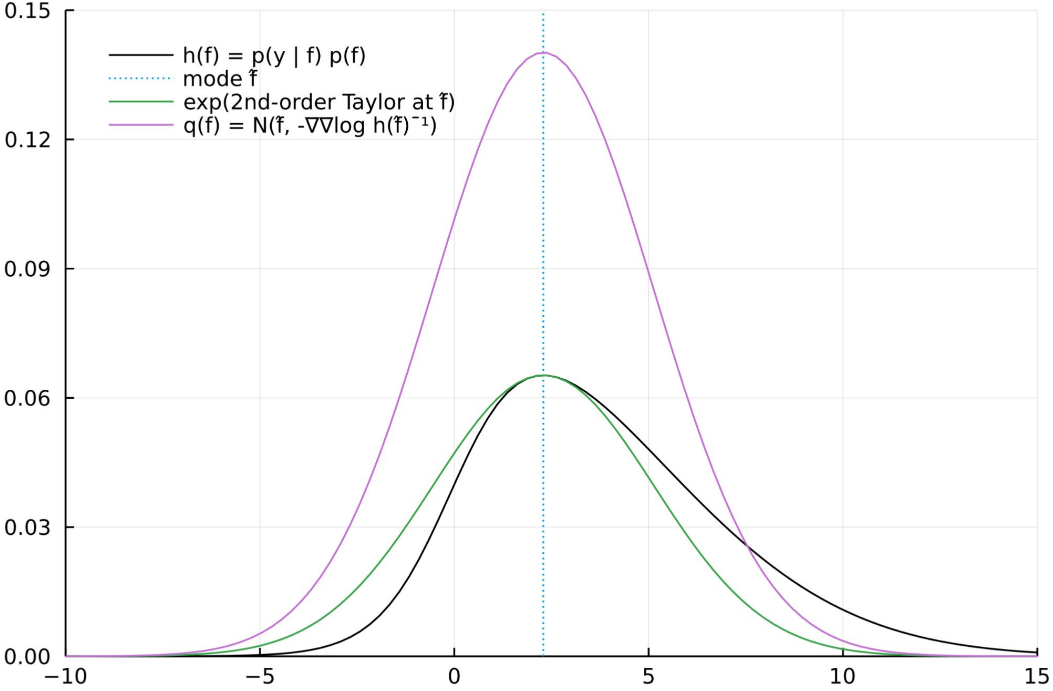
Laplace approximation: the idea



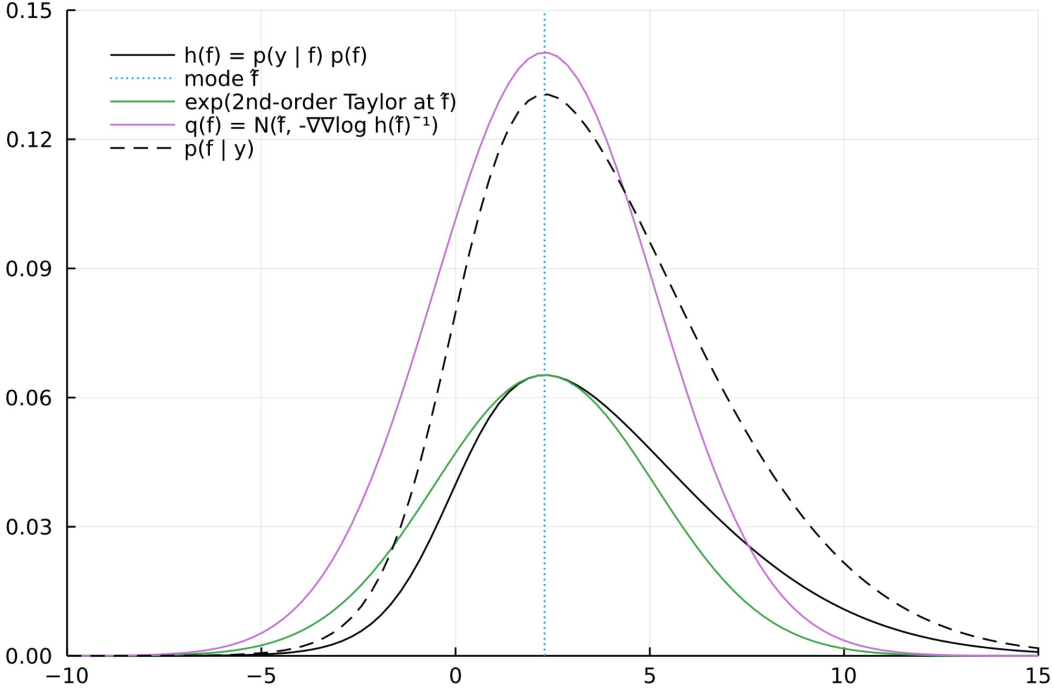
Laplace approximation: the idea



Laplace approximation: the idea



Laplace approximation: the idea



Laplace Approximate: the maths

$$p(\mathbf{f}\,|\,\mathbf{y}) = \frac{p(\mathbf{f})p(\mathbf{y}\,|\,\mathbf{f})}{Z}$$

$$\log\, p(\mathbf{f}\,|\,\mathbf{y}) = -\log\, Z + \log\, p(\mathbf{f}) + \log p(\mathbf{y}\,|\,\mathbf{f}) = h(\mathbf{f})$$

$$h(\mathbf{f}) \quad \underbrace{\approx}_{\text{Taylor at }\mathbf{f}^*} \quad h(\mathbf{f}^*) + \underbrace{\nabla_{\mathbf{f}} h^\top}_{0}(\mathbf{f} - \mathbf{f}^*) + \frac{1}{2}(\mathbf{f} - \mathbf{f}^*)^\top H_{\mathbf{ff}}[h](\mathbf{f} - \mathbf{f}^*)$$

$$p(\mathbf{f}\,|\,\mathbf{y}) \approx \exp\left(\frac{1}{2}(\mathbf{f} - \mathbf{f}^*)^\top H_{\mathbf{ff}}[h](\mathbf{f} - \mathbf{f}^*)\right) = \mathcal{N}(\mathbf{f}; \mathbf{f}^*, -H_{\mathbf{ff}}[h]^{-1})$$

Laplace Approximation: pros and cons

fast and easy to implement ✘

Poor posterior if mode is not representative ✓

Variational inference

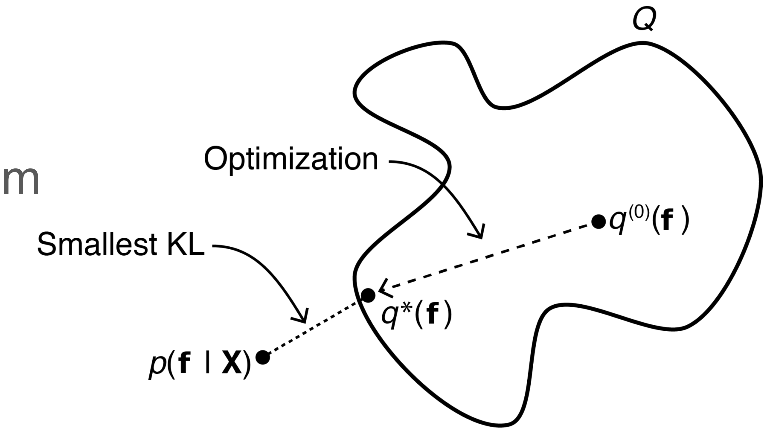
Turning inference into an **optimization** problem

A “distance”

$$\underset{q \in \mathcal{Q}}{\arg\min} \;\; \mathrm{D}_{\mathrm{KL}}[q(\mathbf{f}) \,\|\, p(\mathbf{f} \,|\, \mathbf{x}, \mathbf{y})]$$

Tractable set

Intractable target



Searching for the best Gaussian approximation for the **KL divergence**

$$\mathrm{D}_{\mathrm{KL}}[q(\mathbf{f}) \,\|\, p(\mathbf{f})] = \mathbb{E}_{q(\mathbf{f})} \log \frac{q(\mathbf{f})}{p(\mathbf{f})}$$

Variational inference

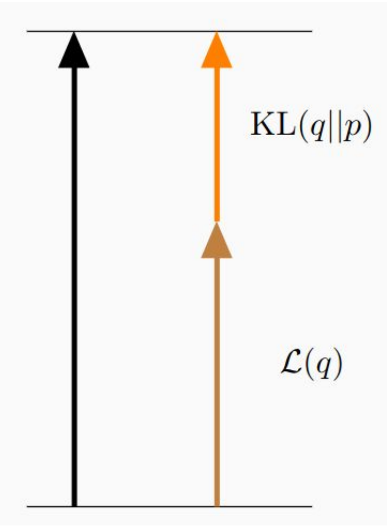
A lower bound to the marginal likelihood

$$\begin{aligned}
\log p(\mathbf{y}) &= \log \int p(\mathbf{f}, \mathbf{y}) \mathrm{d}\mathbf{f} \\
&= \log \int q(\mathbf{f}) \frac{p(\mathbf{f}, \mathbf{y})}{q(\mathbf{f})} \mathrm{d}\mathbf{f} \\
&\underbrace{\geq}_{Jensen} \int q(\mathbf{f}) \log \left(\frac{p(\mathbf{f}, \mathbf{y})}{q(\mathbf{f})} \right) \mathrm{d}\mathbf{f} \\
&= \int q(\mathbf{f}) \log p(\mathbf{y}|\mathbf{f}) \mathrm{d}\mathbf{f} - \mathrm{D}_{\mathrm{KL}}[q(\mathbf{f}) \,\|\, p(\mathbf{f})] = \mathcal{L}(q)
\end{aligned}$$

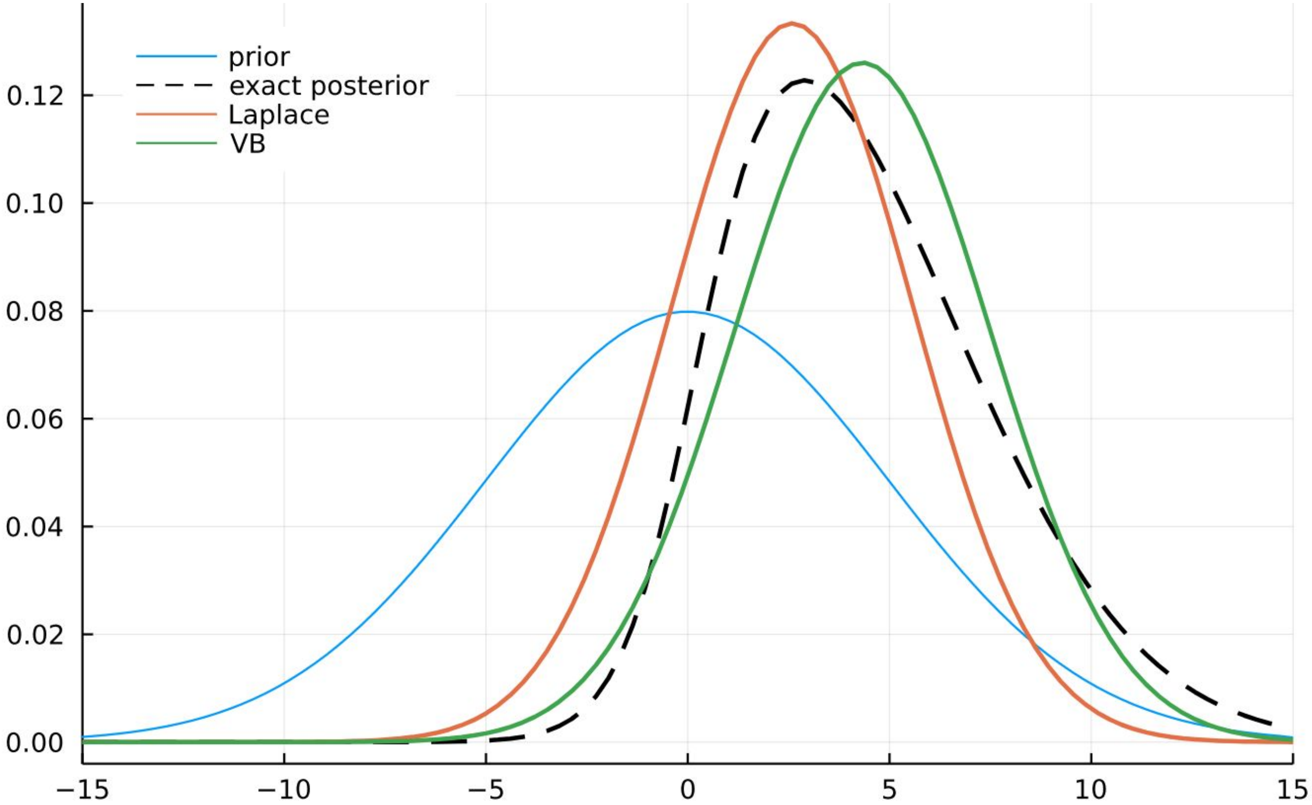
A bound related to the objective

$$\log p(\mathbf{y}) - \mathcal{L}(q) = \mathrm{D}_{\mathrm{KL}}[q(\mathbf{f}) \,\|\, p(\mathbf{f} \,|\, \mathbf{x}, \mathbf{y})]$$

constant



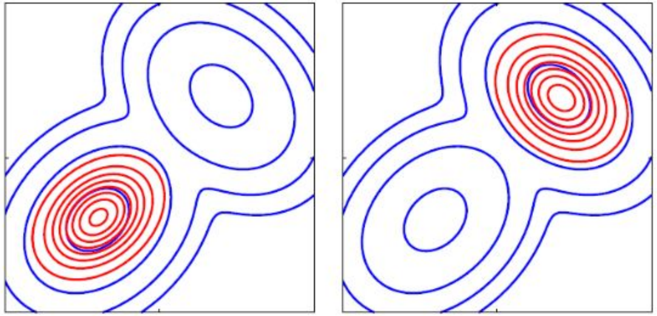
Variational inference



Variational inference: pros and cons

Properties

- A lower bound to the log marginal likelihood ✓
- Inference + learning with a single objective ✓
- Mode matching behavior ✘
- Some theoretical guarantees ✓



Variational inference: details and extensions

- Different **parameterizations** and **optimization** schemes
- VI can be adapted **to more complex likelihoods**
- Using different divergences (instead of the KL)

PART 2 - Scaling up Gaussian process regression

Reminder : Gaussian Process Regression

Problem: Cubic scaling of computation

inverse!

$$\mathbf{y}_i | f_i = f_i + \mathcal{N}(0, \sigma^2)$$

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; 0, \mathbf{K}_{\mathbf{ff}} + \sigma^2 \mathbf{I})$$

$$p(\mathbf{f}_* | \mathbf{y}) = \mathcal{N}(\mathbf{f}_*; \mathbf{K}_{\mathbf{f}_*\mathbf{f}} \mathbf{K}_{\mathbf{ff}}^{-1} \mathbf{y}, \mathbf{K}_{\mathbf{f}_*\mathbf{f}_*} - \mathbf{K}_{\mathbf{f}_*\mathbf{f}} (\mathbf{K}_{\mathbf{ff}} + \sigma^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{f}\mathbf{f}_*})$$

Two main families of approximations

- **Conjugate gradient methods**

  Approximate the computations

- **Inducing point methods (a.k.a sparse methods)**

  Approximate the posterior (by one simpler to compute)

Conjugate Gradient methods

Expression of the Log marginal likelihood and its gradient

$$\hat{\mathbf{K}}_{\mathbf{ff}} = \mathbf{K}_{\mathbf{ff}} + \sigma^2 \mathbf{I}$$

$$\log p_{\boldsymbol{\theta}}(\mathbf{y}) \propto \log |\hat{\mathbf{K}}_{\mathbf{ff}}| - \mathbf{y}^\top \hat{\mathbf{K}}_{\mathbf{ff}}^{-1} \mathbf{y}$$

$$\frac{\mathrm{d} \log p_{\boldsymbol{\theta}}(\mathbf{y})}{\mathrm{d}\boldsymbol{\theta}} = \mathbf{y}^\top \left(\hat{\mathbf{K}}_{\mathbf{ff}}^{-1} \frac{\mathrm{d}\hat{\mathbf{K}}_{\mathbf{ff}}}{\mathrm{d}\boldsymbol{\theta}} \right) \mathbf{y} + \mathrm{Tr} \left(\hat{\mathbf{K}}_{\mathbf{ff}}^{-1} \frac{\mathrm{d}\hat{\mathbf{K}}_{\mathbf{ff}}}{\mathrm{d}\boldsymbol{\theta}} \right)$$

Replace (matrix inverse) **x** (vector) **by** a few (matrix) **x** (vector)

$O(N^3)$ **$O(KN^2)$ K<<N**

Conjugate Gradient methods

(Matrix inverse) **x** (vector)

$$\mathbf{a} = \mathbf{A}^{-1}\mathbf{b}$$

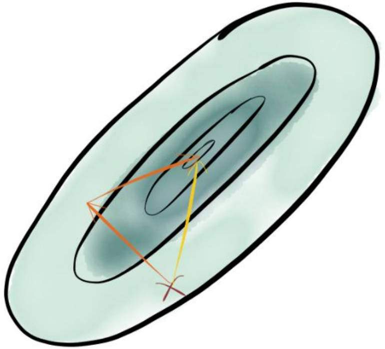
Minimizing a quadratic form

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{b}^\top \mathbf{x} + \mathbf{c}$$

$$\mathbf{a} = \arg\min_{\mathbf{x}} f(\mathbf{x})$$

Following a gradient based procedure

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$$



Conjugate Gradient methods: Idea

Basis of **conjugate** vectors

$$\mathbf{p}_i^\top \mathbf{A} \mathbf{p}_j = 0$$

$$\mathbf{x}^* = \sum_k \alpha_k \mathbf{p}_k$$

Compute the coefficients

$$\begin{aligned} \mathbf{A}\mathbf{x}^* &= \mathbf{b} \\ \mathbf{p}_i^\top \mathbf{A}\mathbf{x}^* &= \mathbf{p}_i^\top \mathbf{b} \\ &= \mathbf{p}_i^\top \mathbf{A} \sum_k \alpha_k \mathbf{p}_k = \alpha_i \mathbf{p}_i^\top \mathbf{A} \mathbf{p}_i \end{aligned}$$

$$\alpha_i = \frac{\mathbf{p}_i^\top \mathbf{b}}{\mathbf{p}_i^\top \mathbf{A} \mathbf{p}_i}$$

How to find the basis of conjugate vectors?

Conjugate Gradient methods: Iterative procedure

Initialize $\mathbf{p}_0 = \nabla_{\mathbf{x}} f(\mathbf{x}_0) = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$

First iteration: follow the gradient

$$\mathbf{x}_1 = \mathbf{x}_0 + \beta_0 \mathbf{p}_0$$

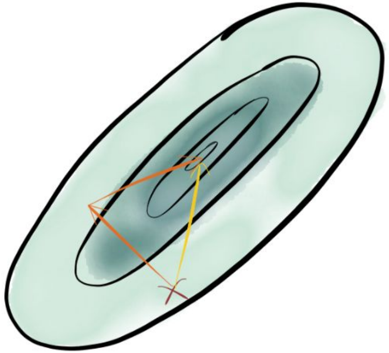
$$\min_{\beta} f(\mathbf{x}_0 + \beta \mathbf{p}_0)$$

$$\beta_0 = \frac{\mathbf{p}_0^\top \mathbf{b}}{\mathbf{p}_0^\top \mathbf{A}\mathbf{p}_0}$$

Next iteration $\hat{\mathbf{p}}_1 = \nabla_{\mathbf{x}} f(\mathbf{x}_1) = \mathbf{A}\mathbf{x}_1 - \mathbf{b}$

$$\mathbf{p}_1 = \hat{\mathbf{p}}_1 - \frac{\hat{\mathbf{p}}_1 \mathbf{A}\mathbf{p}_0}{\hat{\mathbf{p}}_0 \mathbf{A}\mathbf{p}_0}$$

Gram-Schmidt orthogonalization



Carry until gradient small enough

Hopefully stops after **K<<N iterations**

Conjugate gradient methods

- Efficient methods to approximate the log det and trace terms + parallelization
- Efficiency depends on conditioning of **A : preconditioning** helps
- In practice **$O(N^2)$** is still big!

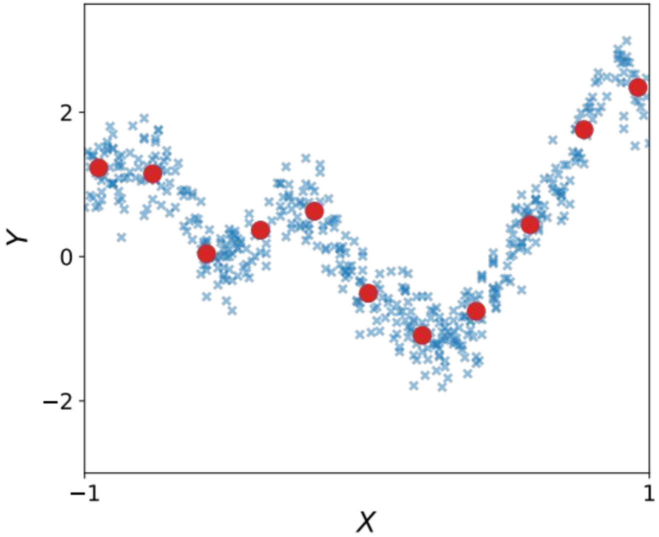
Inducing point: intuition

Gaussian Process regression: posterior mean

$$f^*(x) = \sum_n \alpha_n \, k(x, x_n)$$

Getting rid of the redundant information

$$f^*(x) \approx \sum_m \alpha_m \, k(x, z_m)$$



From non-parametric (**N**) *back* to parametric (**M**)

IDEA: Inference on **f(z)** instead of **f(x)**

Inducing point: variational approach

Reminder of the objective

$$\underset{q \in \mathcal{Q}}{\arg\min}\ \mathrm{D}_{\mathrm{KL}}[q(\mathbf{f}) \,\|\, p(\mathbf{f} \,|\, \mathbf{x}, \mathbf{y})]$$

Choice of *Q*: **q(f(z))** instead of **q(f(x))**

$$q(f(\cdot)) = \int p(f(\cdot) \,|\, f(\mathbf{z}) = \mathbf{u})\, q(\mathbf{u})\, \mathrm{d}\mathbf{u}$$

$$q(f_i) = \mathcal{N}\left(f_i \,|\, \mathbf{a}_i^\top \mathbf{m}_{\mathbf{u}}, \kappa_{ii} - \mathbf{a}_i^\top (\mathbf{K}_{\mathbf{uu}} - \mathbf{S}_{\mathbf{u}})\mathbf{a}_i\right)$$

$$\mathbf{a}_i^\top = \mathbf{k}_{i\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1}$$

$$\mathcal{L}(q) = \mathbb{E}_{q(\mathbf{f})}\left[\log p(\mathbf{y} \,|\, \mathbf{f})\right] - \mathrm{D}_{\mathrm{KL}}[q(\mathbf{u}) \,\|\, p(\mathbf{u})]$$

$O(M^3 + NM^2)$

$$q(f(\cdot)) = \int p(f(\cdot) \,|\, f(\mathbf{x}) = \mathbf{f})\, q(\mathbf{f})\, \mathrm{d}\mathbf{f}$$

$$q(f_i) = \mathcal{N}\left(f_i \,|\, \mathbf{b}_i^\top \mathbf{m}_{\mathbf{f}}, \kappa_{ii} - \mathbf{b}_i^\top (\mathbf{K}_{\mathbf{ff}} - \mathbf{S}_{\mathbf{f}})\mathbf{b}_i\right)$$

$$\mathbf{b}_i^\top = \mathbf{k}_{i\mathbf{f}} \mathbf{K}_{\mathbf{ff}}^{-1}$$

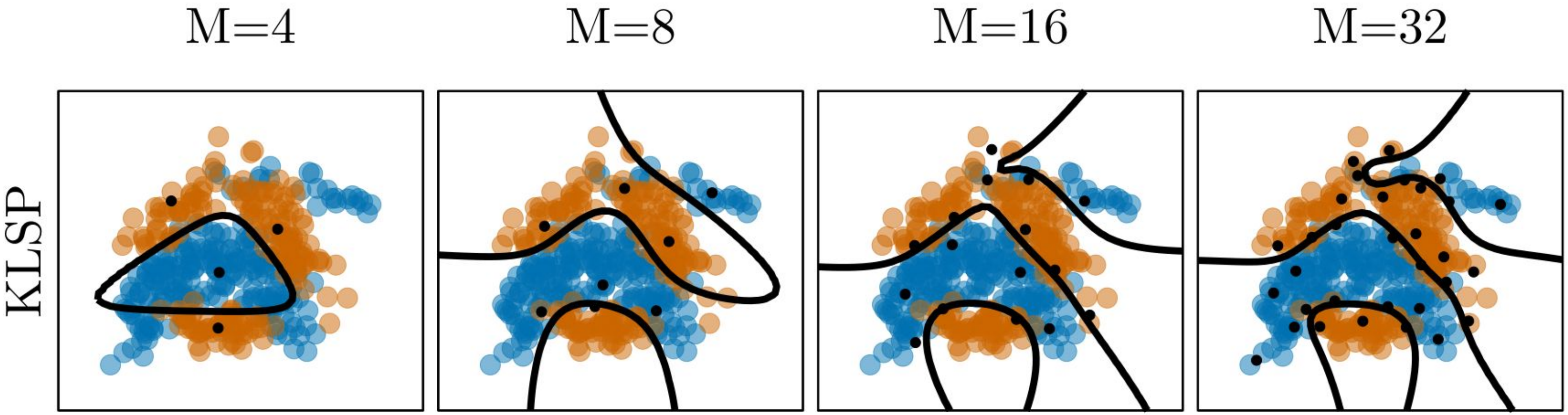
$$\mathcal{L}(q) = \mathbb{E}_{q(\mathbf{f})}\left[\log p(\mathbf{y} \,|\, \mathbf{f})\right] - \mathrm{D}_{\mathrm{KL}}[q(\mathbf{f}) \,\|\, p(\mathbf{f})]$$

$O(N^3 + N)$

Inducing point: variational approach

Reminder of the objective $$\arg\min_{q \in \mathcal{Q}} \; \mathrm{D}_{\mathrm{KL}}[q(\mathbf{f}) \,\|\, p(\mathbf{f} \,|\, \mathbf{x}, \mathbf{y})]$$

Choice of *Q*: **q(f(z))** instead of **q(f(x))**



Hensman et al, AISTATS 2015

Inducing points: going further

- Gaussian case: closed form solution for **q*** and **L(q*)**

$$\mathcal{L}(q^*) = \log \mathcal{N}(\mathbf{y}; 0, \sigma^2\mathbf{I} + \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}) - \frac{1}{2}\mathrm{Tr}\left[\mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\right]$$

- Interdomain approach: other choice for **u=ϕ(f)**

$$\mathbf{u}_m = \int f(x)e^{imx}dx$$

- Mini-batching: stochastic evaluation of the loss

$$\mathcal{L}(q) = \sum_{i=1}^{n} \mathbb{E}_{q(f_i)}\left[\log p(y_i \mid f_i)\right] - \mathrm{D_{KL}}[q(\mathbf{f}) \parallel p(\mathbf{f})]$$

O(NM² + M³)

$$\approx \frac{n}{|\mathcal{B}|}\sum_{j\in\mathcal{B}} \mathbb{E}_{q(f_j)}\left[\log p(y_j \mid f_j)\right] - \mathrm{D_{KL}}[q(\mathbf{f}) \parallel p(\mathbf{f})]$$

O(N_{batch}M² + M³)

Mixing the two parts?

Computationally efficiciency

+

Non conjugacy

Questions ?

References

Conjugate gradient method

- Gardner, Jacob, et al. "Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration." *Advances in neural information processing systems* 31 (2018).
- Artemev, Artem, David R. Burt, and Mark van der Wilk. "Tighter bounds on the log marginal likelihood of Gaussian process regression using conjugate gradients." *International Conference on Machine Learning*. PMLR, 2021.

Sparse GPs

- Hensman, James, Nicolò Fusi, and Neil D. Lawrence. "Gaussian processes for big data." *Conference on Uncertainty in Artificial Intelligence* (2013): 282-290.
- Titsias, Michalis. "Variational learning of inducing variables in sparse Gaussian processes." *Artificial intelligence and statistics*. PMLR, 2009.
- Wild, Veit, Motonobu Kanagawa, and Dino Sejdinovic. "Connections and equivalences between the nystr\" om method and sparse variational gaussian processes." *arXiv preprint arXiv:2106.01121* (2021).
- Bui, Thang D., Josiah Yan, and Richard E. Turner. "A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation." *The Journal of Machine Learning Research* 18.1 (2017): 3649-3720.
- Burt, David R., Carl Edward Rasmussen, and Mark Van Der Wilk. "Convergence of sparse variational inference in Gaussian processes regression." *The Journal of Machine Learning Research* 21.1 (2020): 5120-5182.

Non conjugate Regression

- Kuss, Malte, Carl Edward Rasmussen, and Ralf Herbrich. "Assessing Approximate Inference for Binary Gaussian Process Classification." *Journal of machine learning research* 6.10 (2005).
- Gardner, Jacob, et al. "Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration." *Advances in neural information processing systems* 31 (2018).