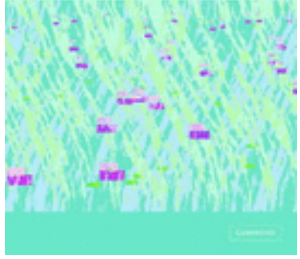
Rubén Martínez Cantín
Dpto. Informática e Ingeniería de Sistemas.

- Markov Chain Monte Carlo
- **Bayesian decision making**
  - Active learning
  - Bayesian optimization

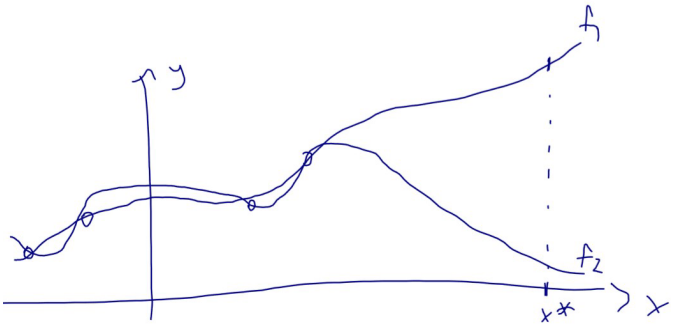
- Phillip Henning, Probabilistic Machine Learning, https://uni-tuebingen.de/en/180804



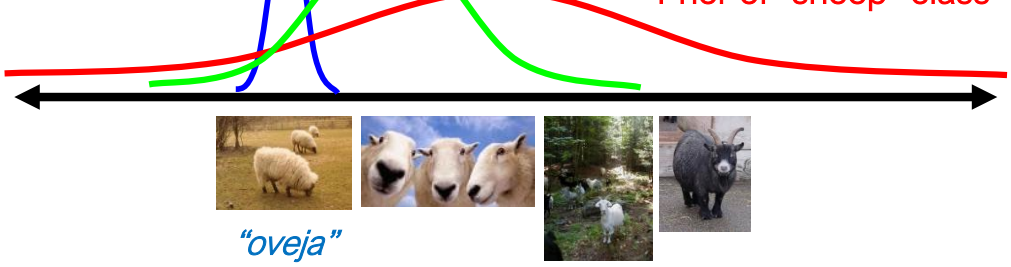
- Python resources
  - In the labs: GPy, GPyOpt , scikit-learn.
  - Gpyflow, PyMC3…

Universidad Zaragoza

engineer to use for your diagnosis?



Universidad Zaragoza
1542



"oveja"

Images: Wikipedia, Nando de Freitas.

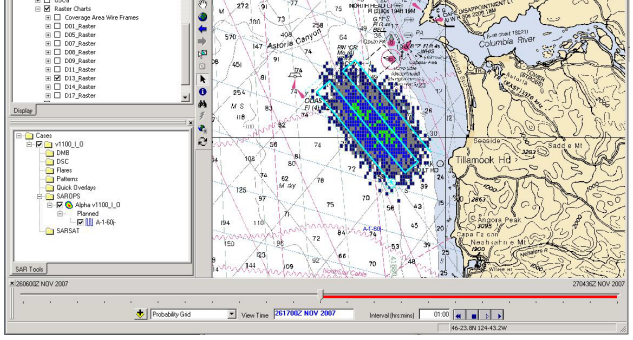
- Bayes Rule

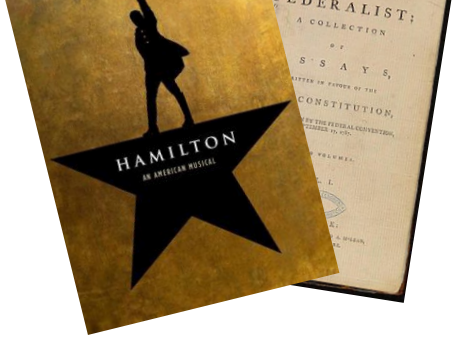
$$P(A|B) = \frac{P(B|A)P(A)}{\int_{\mathcal{A}} P(B|A)P(A)dA}$$

- Expectation

$$\mathbb{E}_{P(A)} f(A) = \int_{\mathcal{A}} f(A)P(A)dA$$

- Which animal do you think is deadlier (higher probability of dying if bitten)?
  - p(shark bite)?
  - p(dead | shark bite)?
  - p(shark bite | dead)?



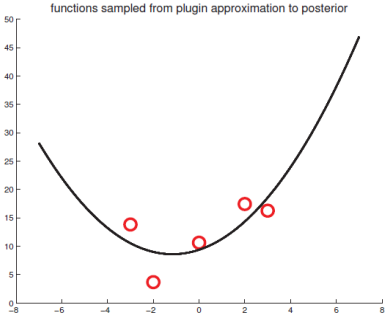


Universidad Zaragoza

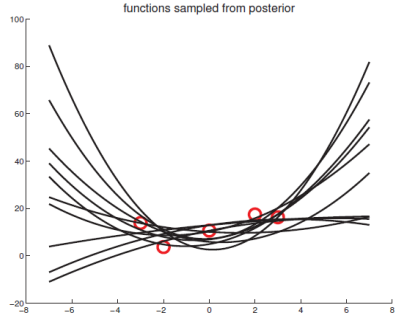
1542

3G/4G

Images: Wikipedia, Judea Pearl, Katie Bowman, Claude Berrou, David J.C. Mackay.



(c)



(d)

Image: Kevin P. Murphy

$p(w)$

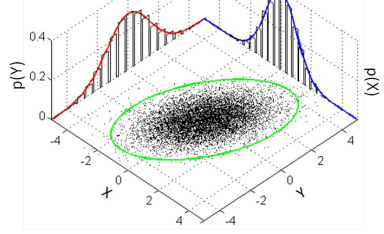
- Step 3: Compute the posterior:

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)}$$

- What is the prior?
  - For a linear-Gaussian likelihood model, the conjugate prior is a Gaussian.
  - The posterior is also Gaussian → Closed-form, easy, allows recurrence.

$$p(w) = N(w|w_0, \Sigma_0)$$

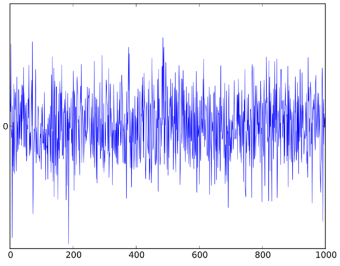
- p(x) and p(y) are Gaussian.
- p(x|y) and p(y|x) are Gaussian.
- Linear: a * p(x) + b * p(y) is a Gaussian.
- Product: p(x)*p(y) is a Gaussian:
- Convolution: p(x + y) is a Gaussian



Images: Wikipedia

$$p(\mathbf{x}_1) = \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$$
$$p(\mathbf{x}_2) = \mathcal{N}(\mathbf{x}_2|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22})$$

$$\begin{aligned} p(\mathbf{x}_1|\mathbf{x}_2) &= \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \\ \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ &= \boldsymbol{\Sigma}_{1|2}\left(\boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2)\right) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} = \boldsymbol{\Lambda}_{11}^{-1} \end{aligned}$$



- Noise is usually Gaussian

Universidad Zaragoza

Image: Wikipedia

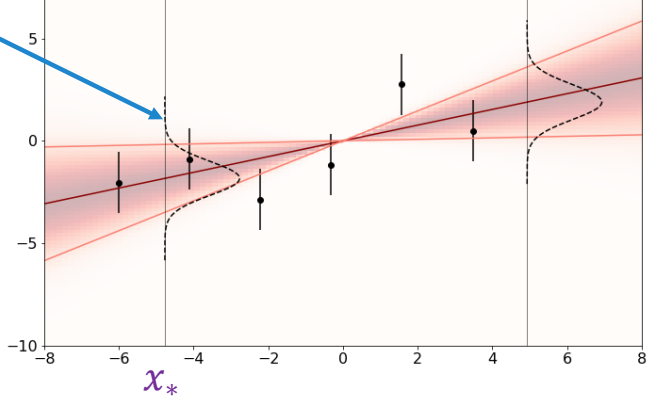
- If we use an isotropic Gaussian prior: $p(w) = N(w|0, \tau^2 I_d)$ then

Remember

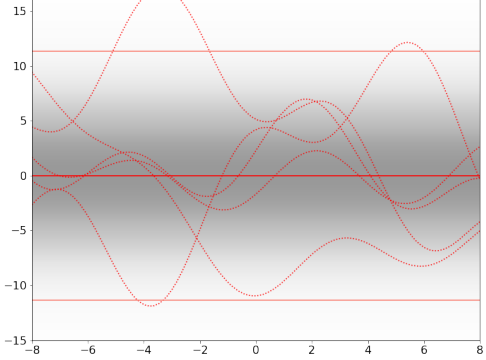
$y = x^T w + \epsilon$

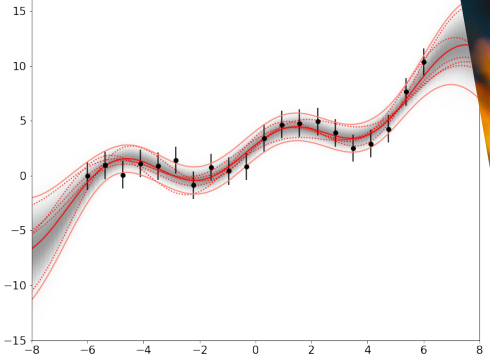
$\mu(x_*) = ?$

$\sigma_n^2(x_*) = ?$

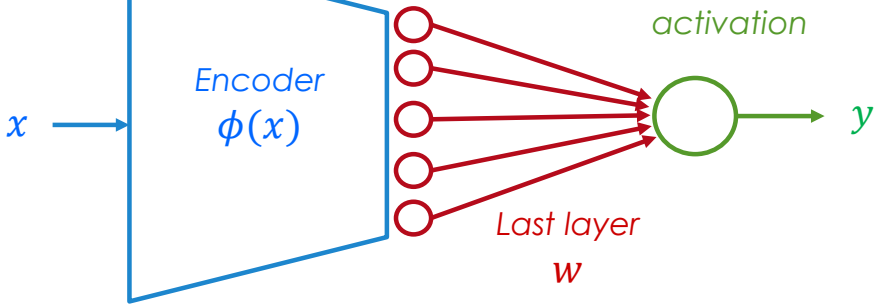


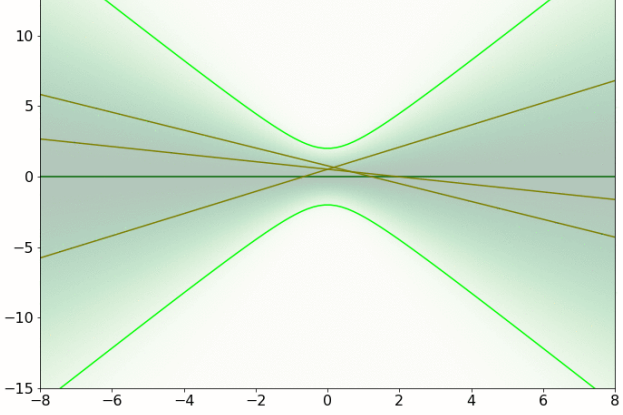
Universidad Zaragoza

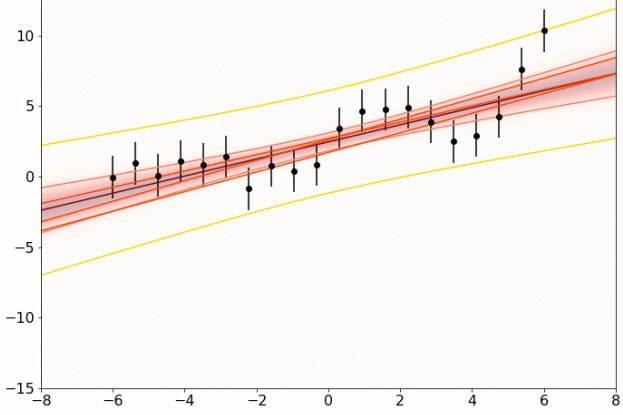

15
10
5
0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8

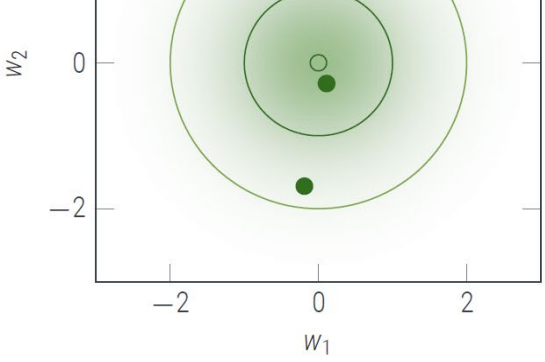

15
10
5
0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8

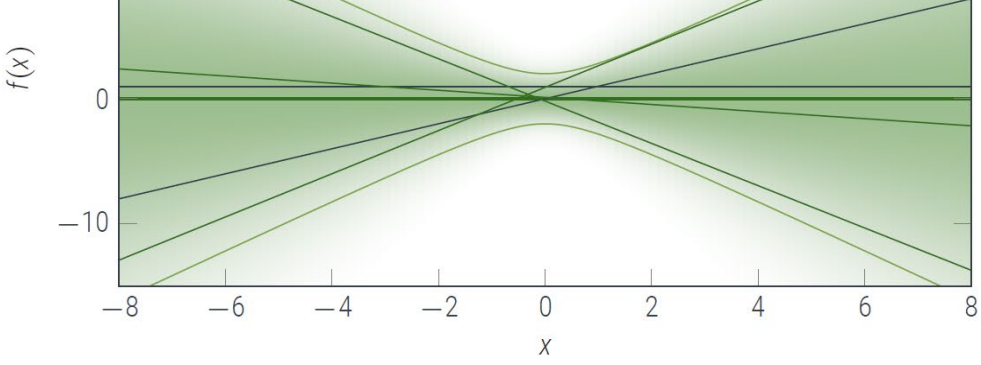

GAUSSIAN


activation
Encoder
$\phi(x)$
x
y
Last layer
w


10
5
0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8

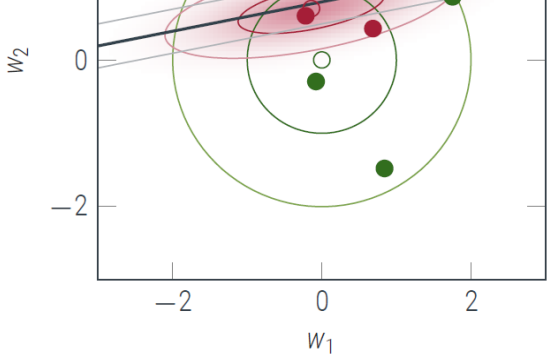

10
5
0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8

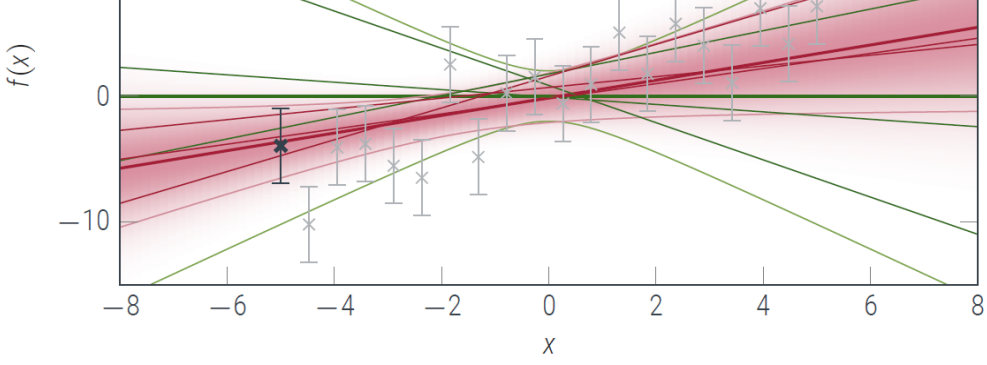




Universidad Zaragoza

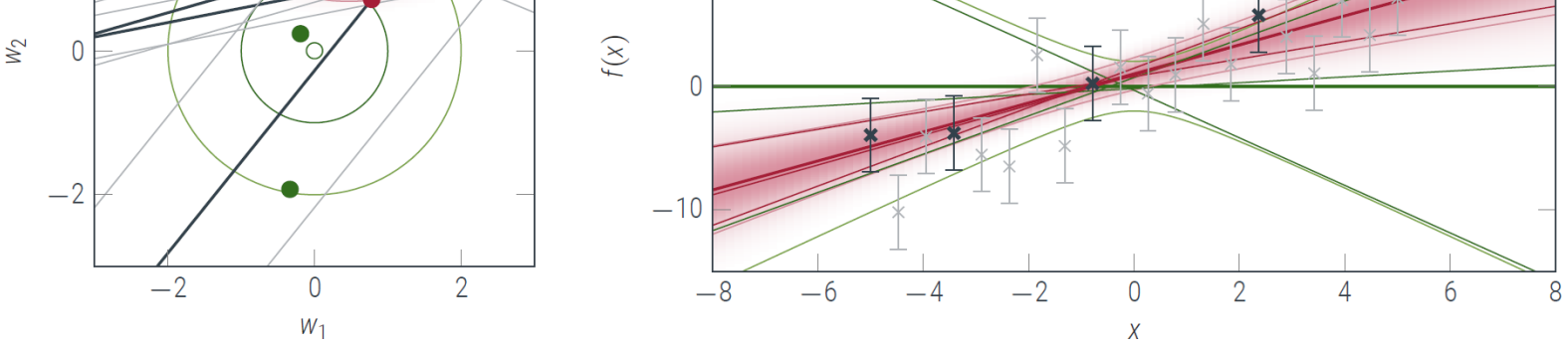
Image: Philipp Hennig





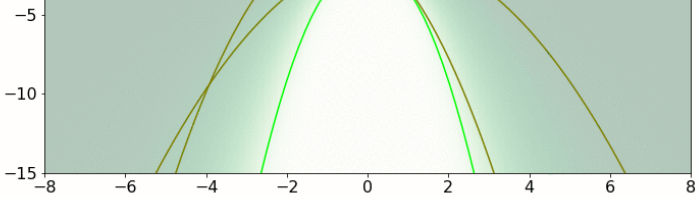
Universidad Zaragoza

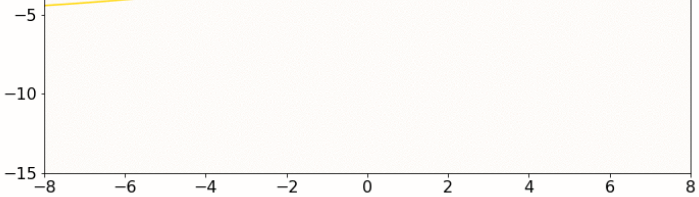
Image: Philipp Hennig



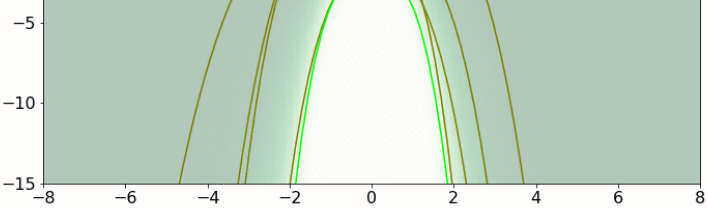
Universidad Zaragoza

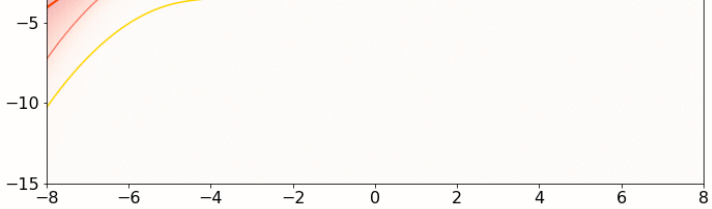
Image: Philipp Hennig



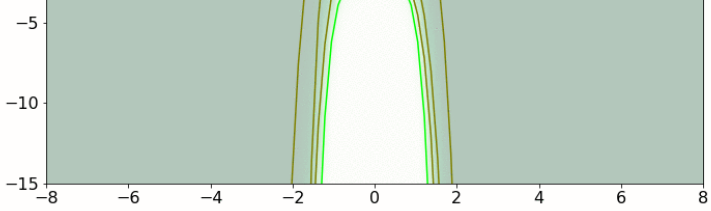


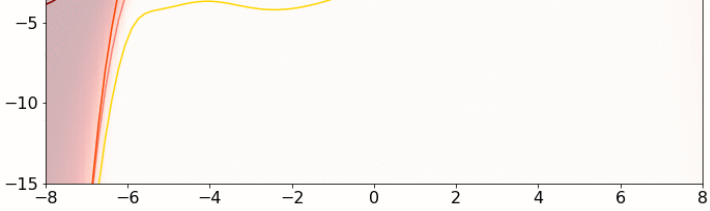
Universidad Zaragoza

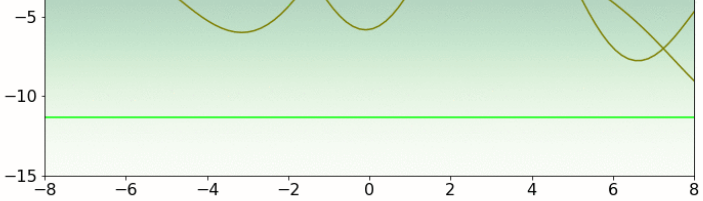


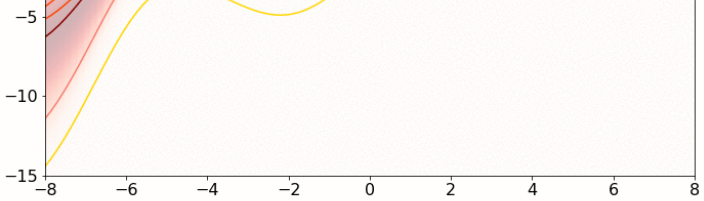


Universidad Zaragoza

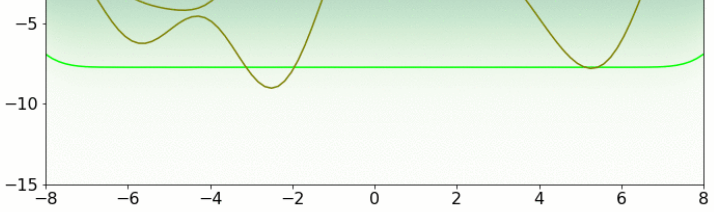

−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8

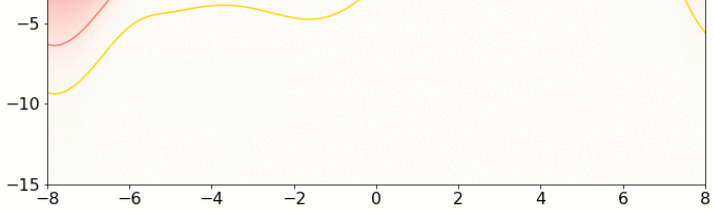

−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8

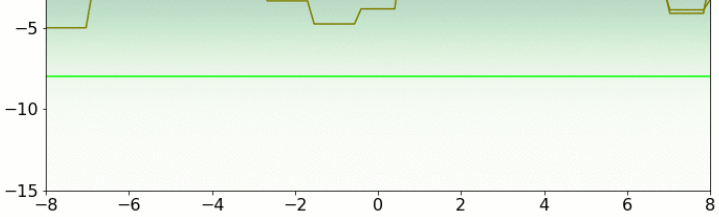


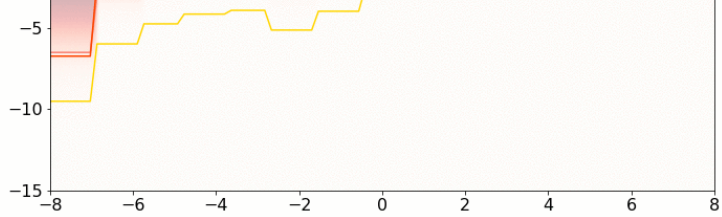


Universidad Zaragoza

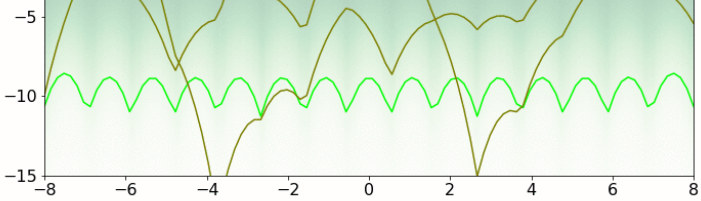

−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8

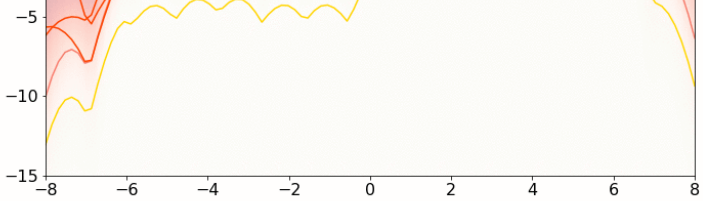

−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8





Universidad Zaragoza





Universidad Zaragoza

- Hyperparameters are nonlinear.
  - No closed-form. No Gaussian…

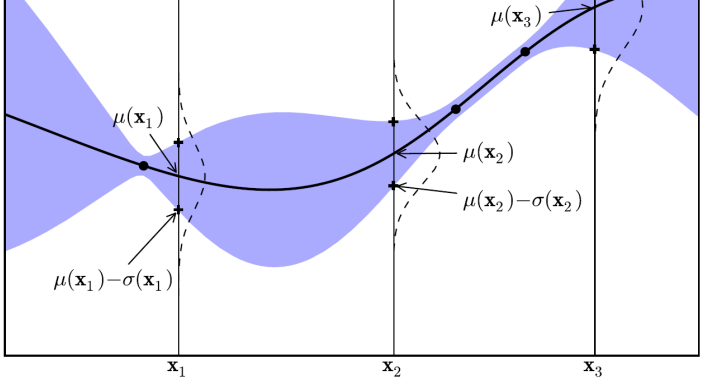
- We'll get back to it latter.

result is always a scalar.

- Kernel function: $k(x_1, x_2) = \phi(x_1)^T \Sigma \phi(x_2)$
  - This type of function is called a Mercer kernel.

- The predictive posterior is a conditional Gaussian of:

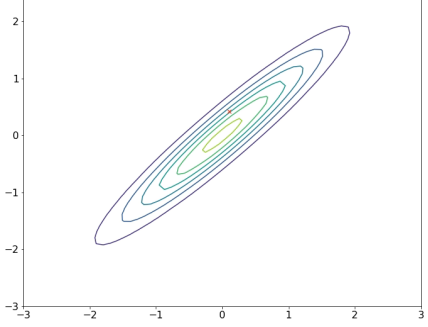
$$\begin{pmatrix} \boldsymbol{y} \\ f_* \end{pmatrix} \sim N\left(0, \begin{bmatrix} k(X,X) + \sigma^2 I & k(X,x_*) \\ k(x_*,X) & k(x_*,x_*) \end{bmatrix}\right)$$

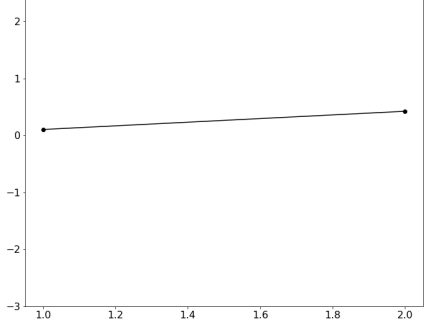


Universidad Zaragoza

Image: Eric Brochu

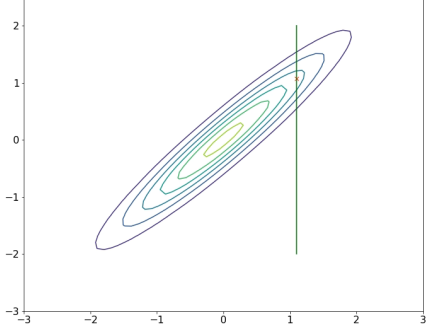
Large correlation

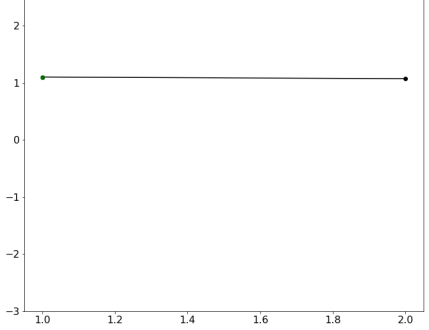




Universidad Zaragoza

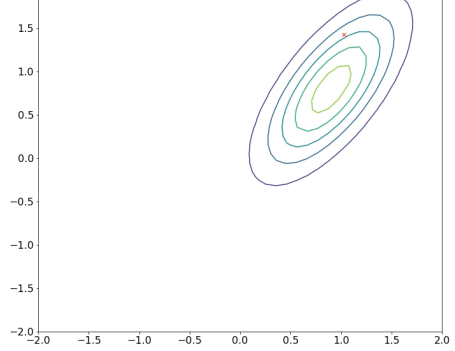
Large correlation

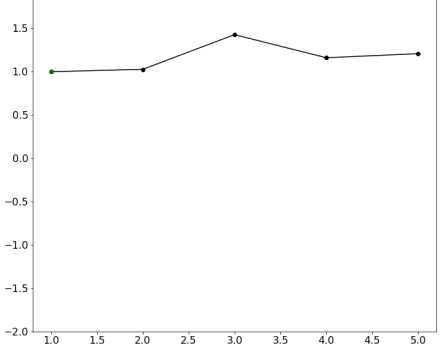




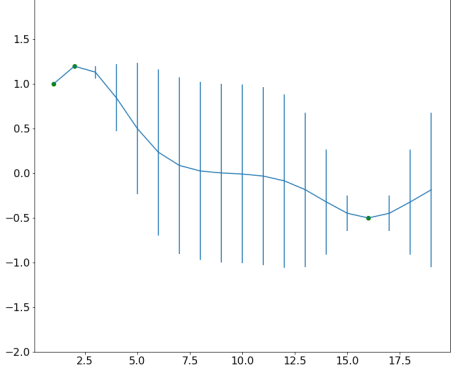
Universidad Zaragoza
1542

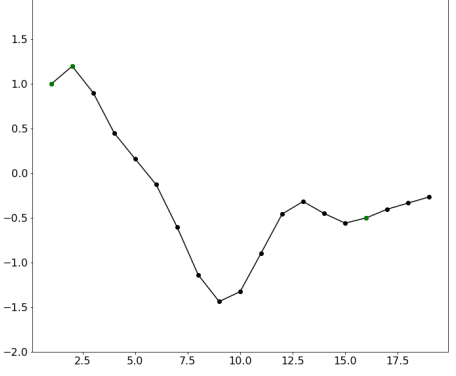
Conditional





Conditional





Universidad Zaragoza

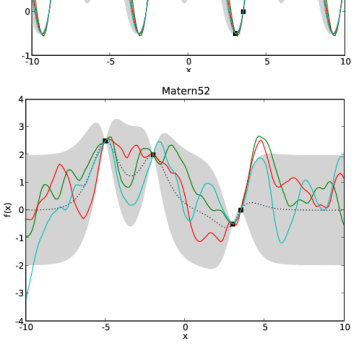
data points.

- GPs are distributions over functions:

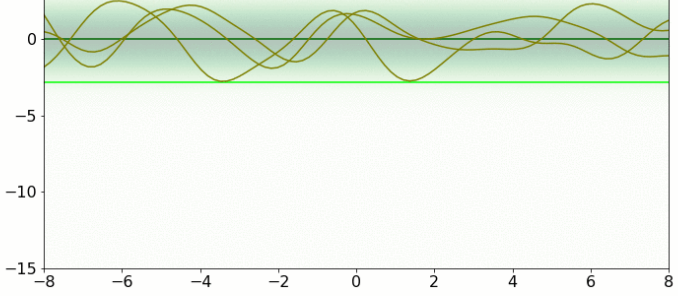
$$f(x) \sim \mathcal{GP}\big(m(x), k(x, x')\big)$$

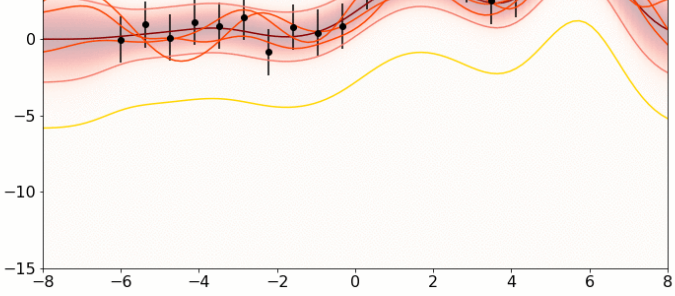
$$m(x) = \mathbb{E}[f(x)]$$
$$k(x, x') = \mathbb{E}\Big[\big(f(x) - m(x)\big)\big(f(x) - m(x)\big)^T\Big]$$

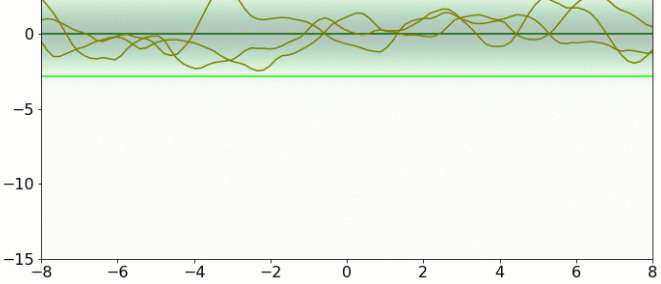
rational quadratic	$(1 + \frac{r^2}{2\alpha\ell^2})^{-\alpha}$
neural network	$\sin^{-1}\left(\frac{2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^\top \Sigma \tilde{\mathbf{x}}')}}\right)$

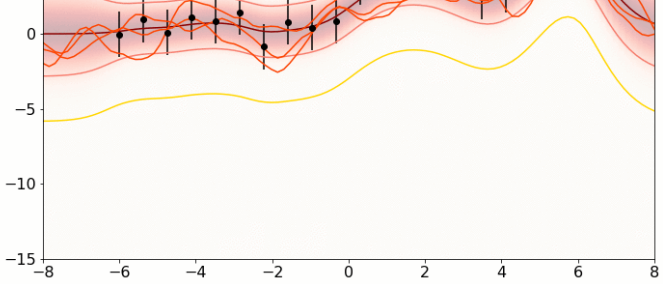


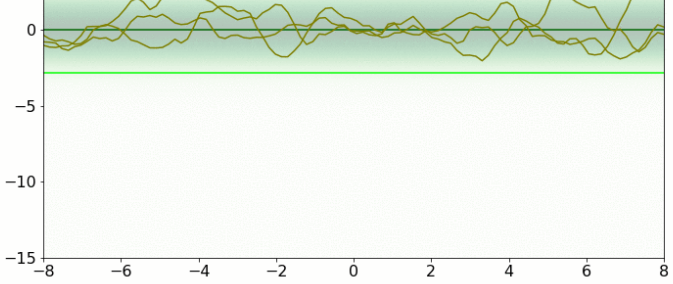
- Others: Wiener process, splines…
- Not only real vectors: strings, paths…

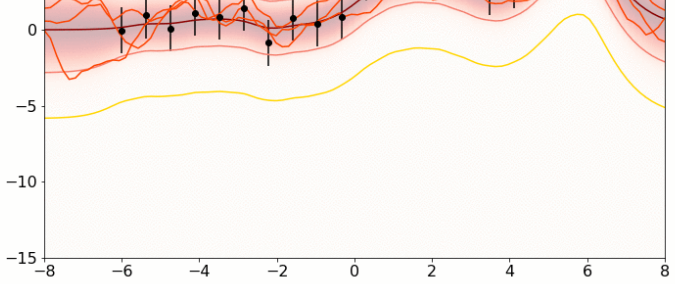

0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8


0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8


0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8


0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8

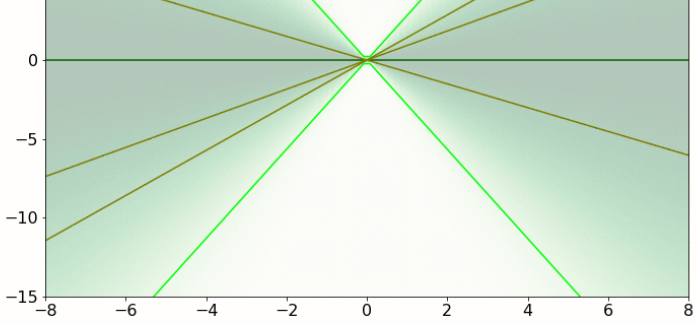


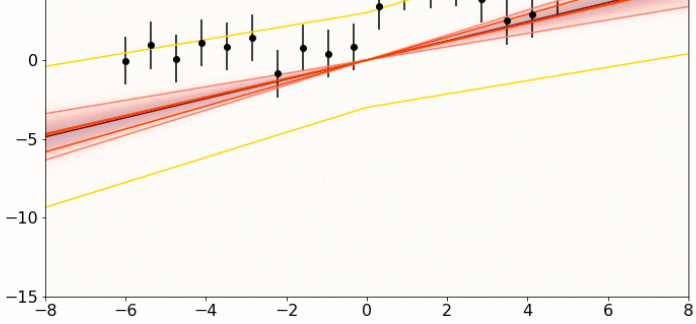


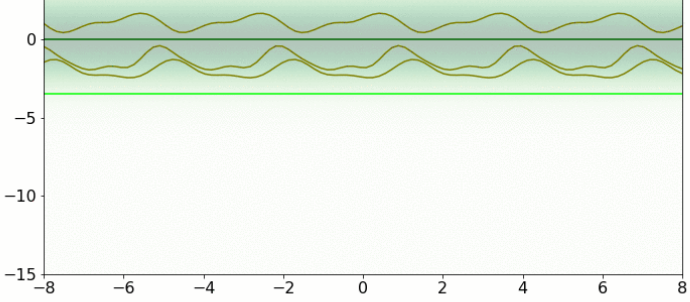
Universidad Zaragoza

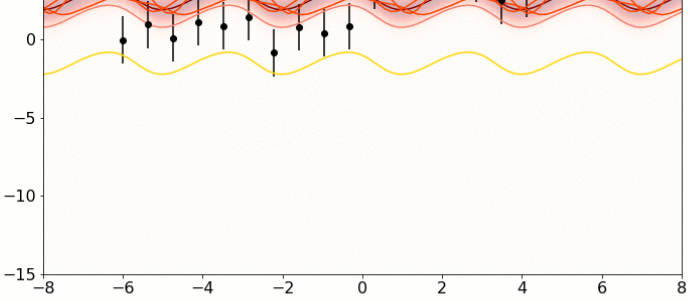
Matérn $\nu \to \infty$ → Squared exponential

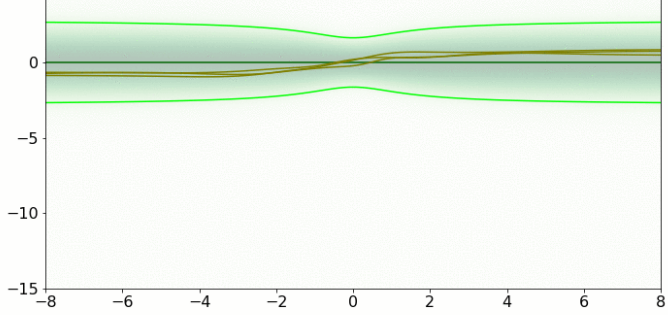
Matérn v=1/2 → Brownian motion model

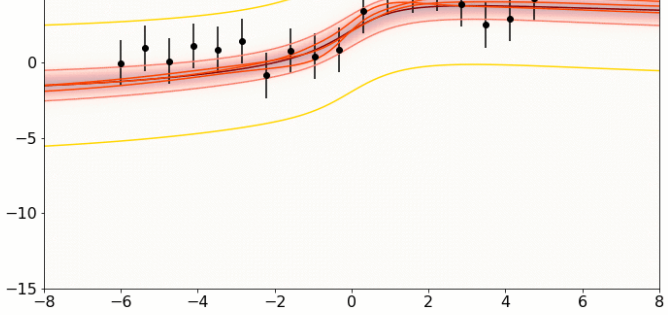

0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8


0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8

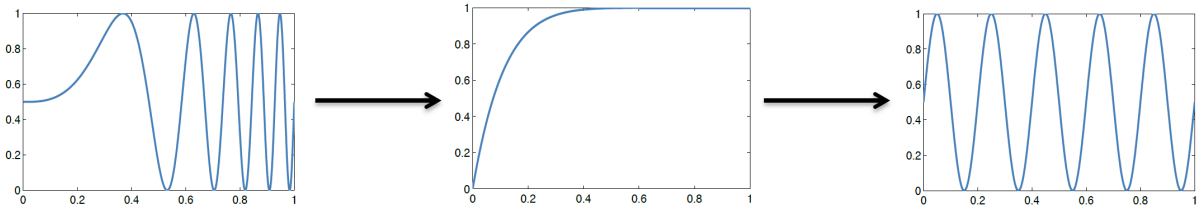

0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8


0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8


0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8

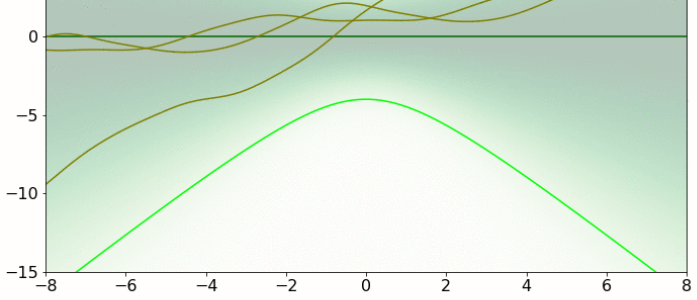

0
−5
−10
−15
−8
−6
−4
−2
0
2
4
6
8

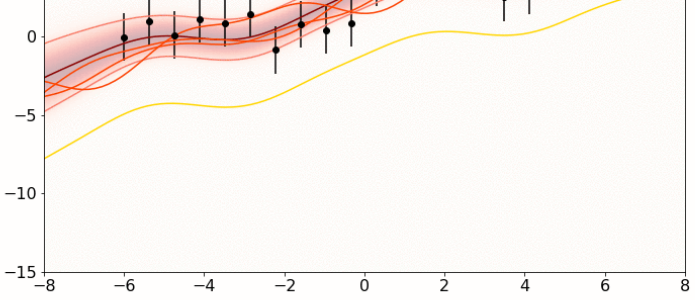
- Mapped inputs: $k(\psi(x), \psi(x'))$



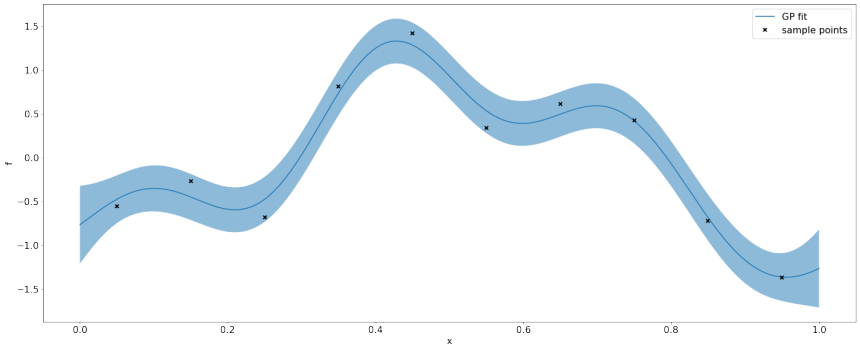
Universidad Zaragoza

Image: Wil Ward & Jasper Snoek





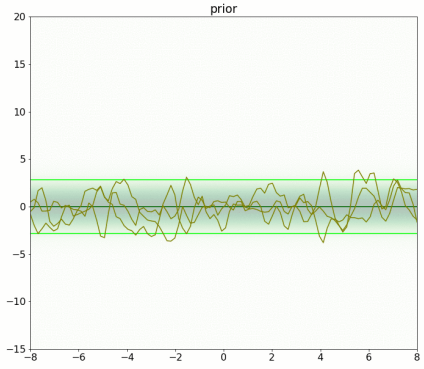
Universidad Zaragoza

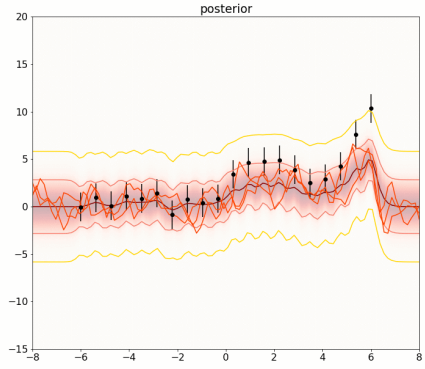


Universidad Zaragoza

Universidad Zaragoza

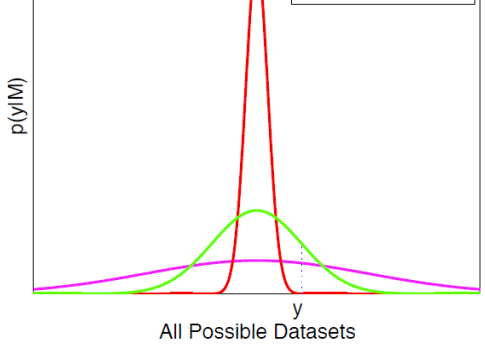
l=0.3

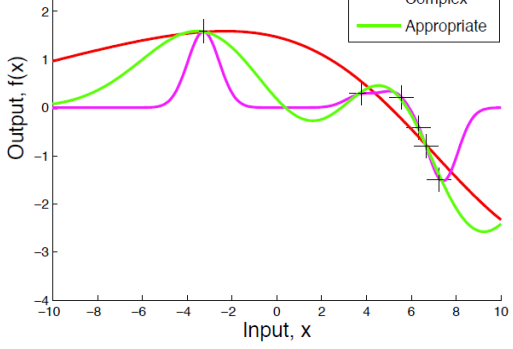




similar alternative: Cross-validation.

- Sampling (e.g.: MCMC – next lesson):
  - *Full Bayesian*
  - Result in a mixture of GPs
  - Much, much more expensive


p(y|M)
y
All Possible Datasets


Complex
Appropriate
Output, f(x)
Input, x

- Learn – Maximize the marginal log-likelihood:

$$\log p(\boldsymbol{y} \mid X, \theta) = -\frac{1}{2}\boldsymbol{y}^T K_\theta^{-1}\boldsymbol{y} - \frac{1}{2}\log|K_\theta| - \frac{n}{2}\log 2\pi$$

$\mathbf{x}_*$ (test input)

2: $L := \text{cholesky}(K + \sigma_n^2 I)$

$\boldsymbol{\alpha} := L^\top \backslash (L \backslash \mathbf{y})$ } predictive mean eq. (2.25)

4: $\bar{f}_* := \mathbf{k}_*^\top \boldsymbol{\alpha}$

$\mathbf{v} := L \backslash \mathbf{k}_*$ } predictive variance eq. (2.26)

6: $\mathbb{V}[f_*] := k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^\top \mathbf{v}$

$\log p(\mathbf{y}|X) := -\frac{1}{2}\mathbf{y}^\top \boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{n}{2} \log 2\pi$ eq. (2.30)

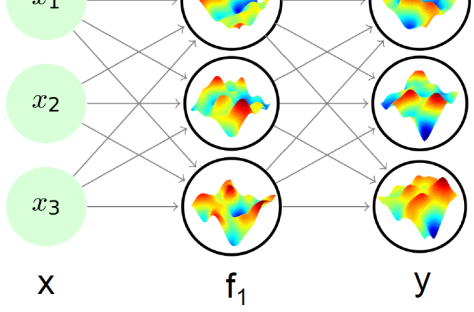
8: **return**: $\bar{f}_*$ (mean), $\mathbb{V}[f_*]$ (variance), $\log p(\mathbf{y}|X)$ (log marginal likelihood)

Image: Rasmussen and Williams

- $\boldsymbol{\alpha} = (\mathbf{L} \backslash \boldsymbol{k}_*^T)^T$
- $\mu = \boldsymbol{\alpha} \boldsymbol{y}$
- $\sigma_n^2 = k(x_*, x_*) - \boldsymbol{\alpha} \boldsymbol{k}_*^T$
- $\boldsymbol{\beta} = (\mathbf{L} \backslash \boldsymbol{y})$
- $\log\big(p(y|X)\big) = -\frac{1}{2} \boldsymbol{\beta}^T \boldsymbol{\beta} - \sum_{i=1}^{N} \log(L_{ii}) - \frac{N}{2} \log(2\pi)$

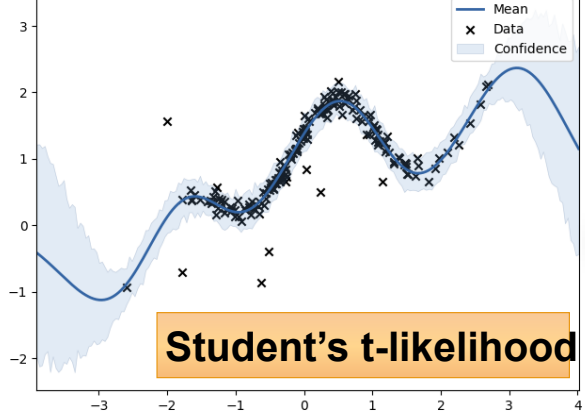
course

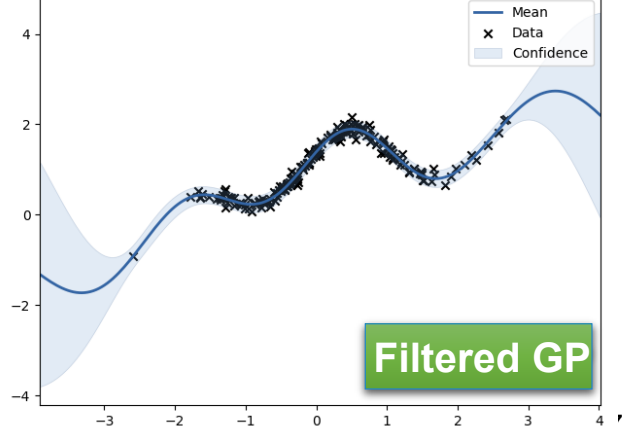
- Deep GPs
  - Build a model made of layers of GPs
- Robust regression:
  - Use non-Gaussian likelihoods (e.g.: Student-t)



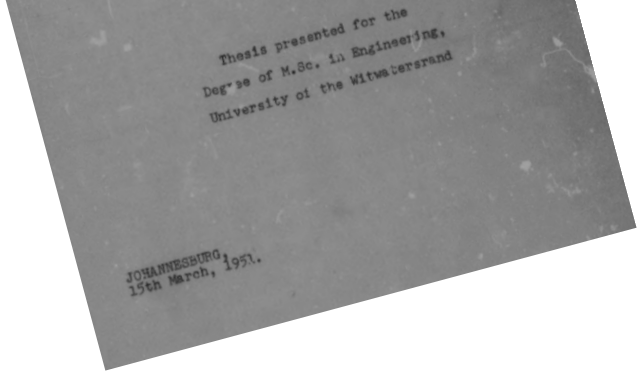
Universidad Zaragoza

Images: M. Havasi et al.,

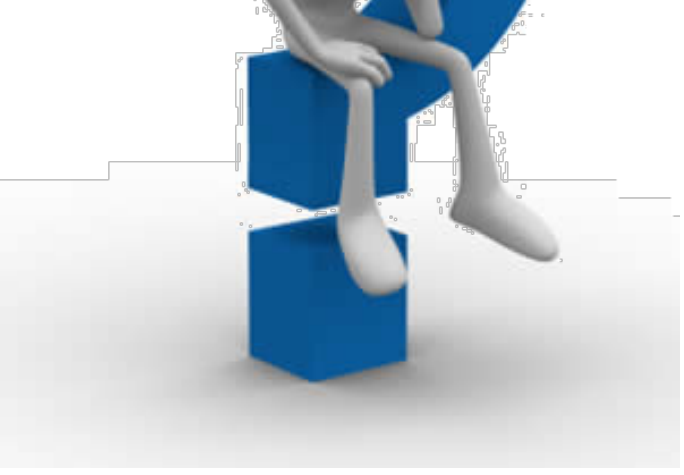

Mean
Data
Confidence
Student's t-likelihood


Mean
Data
Confidence
Filtered GP

- GPs for decision making
  - GPs in reinforcement learning/bandits
  - GPs for dynamic systems
  - GPs for experimental design and optimization → Meta-learning


Thesis presented for the
Degree of M.Sc. in Engineering,
University of the Witwatersrand

JOHANNESBURG
15th March, 1951.



and Computer Vision