Gaussian Processes for Regression, Classification and Dimensionality Reduction

Advanced Machine Learning

École Polytechnique Fédérale de Lausanne, Switzerland

6 May 2020

Learning Outcomes

- Probabilistic Linear Regression
- Maximum Likelihood and Maximum a posteriori estimations
- Bayesian Model Selection
- Non-linear regression with GPs
- Classification with GPs
- Dimensionality reduction with GPLVM

Probabilistic Linear Regression

Creating the model

Observations y are modeled as linear combination of weights w and inputs x contaminated with noise ϵ

$$\hat{y} = f(x) = w^T x + \epsilon \tag{1}$$

The probabilistic regression model is created by handling noise as random variable (RV):

- $\epsilon \sim \mathcal{N}\left(0, \sigma_\epsilon^2\right)$

Gaussian properties:

- The addition of a Normal RV to a constant variable yields a Normal random variable

$$p(\hat{y}) = \mathcal{N}\left(\hat{w}^T x, \sigma_\epsilon^2\right) \tag{2}$$

Probabilistic Linear Regression

Maximum Likelihood Estimation

Given training set:$\{\mathbf{y}, \mathbf{X}\} = \{y_i, \mathbf{x}_i\}_{i=1}^{M}$ and σ_ϵ is assumed known (Hyperparameter), we can define the likelihood as:

$$p(\mathbf{y}|\mathbf{X}, \hat{\mathbf{w}}, \sigma_\epsilon^2) = \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y_i - \hat{\mathbf{w}}^T \mathbf{x_i}}{2\sigma_\epsilon^2}\right) \tag{3}$$

$$\hat{\mathbf{w}}_{\text{MLE}}, \sigma_\epsilon^2 = \arg\max_{\hat{\mathbf{w}}, \sigma_w^2} p(\mathbf{y}|\mathbf{X}, \hat{\mathbf{w}}, \sigma_\epsilon^2 \mathbf{I}_M)$$

Closed form solution

$\hat{\mathbf{w}}_{\text{MLE}} = \left(\mathbf{X}\mathbf{X}^T\right)^{-1} \mathbf{X}\mathbf{y} \rightarrow$ Same with the OLS estimator

$\hat{\sigma}^2_{\text{MLE}} = \frac{1}{N}\left(\hat{\mathbf{w}}^T_{\text{MLE}}\mathbf{X} - \mathbf{y}\right)^T \left(\hat{\mathbf{w}}^T_{\text{MLE}}\mathbf{X} - \mathbf{y}\right) \rightarrow$ Average of squared deviations

Probabilistic Linear Regression

Maximum a Posteriori (MAP) estimation

Weights are also handled as RVs

Prior distribution over weights: $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{w}}, \boldsymbol{\Sigma}_{\mathbf{w}})$

Model Likelihood: $p(\mathbf{y}|\mathbf{X}, \hat{\mathbf{w}}, \sigma_\epsilon^2 \mathbf{I}_M)$

Bayes Rule: $\text{posterion} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X}, \hat{\mathbf{w}}, \sigma_\epsilon^2) p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})} \tag{4}$$

Prior and Likelihood are conjugate distributions[1]. The posterior has a closed form solution

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{w}|\mathbf{X},\mathbf{y}}, \mathbf{A}^{-1})$$

[1] See Conjugate Bayesian analysis of the Gaussian distribution by Murphy

Probabilistic Linear Regression

Maximum a Posteriori (MAP) estimation

MAP estimation derives from the expected value of the posterior normal distribution

$$\begin{aligned} \hat{\mathbf{w}}_{\text{MAP}} &= \mathbb{E}\left\{p\left(\mathbf{w}|\mathbf{x},\mathbf{Y}\right)\right\} \\ \hat{\mathbf{\Sigma}}_{\text{MAP}}^{-1} &= \mathbf{A} = \frac{1}{\sigma_\epsilon^2}\mathbf{X}\mathbf{X}^T + \mathbf{\Sigma}_{\mathbf{w}}^{-1} \\ \hat{\mathbf{w}}_{\text{MAP}} &= \boldsymbol{\mu}_{\mathbf{w}|\mathbf{X},\mathbf{y}} = \mathbf{A}^{-1}\mathbf{\Sigma}_{\mathbf{w}}^{-1}\boldsymbol{\mu}_{\mathbf{w}} + \frac{1}{\sigma_\epsilon^2}\mathbf{A}^{-1}\mathbf{X}\mathbf{y} \end{aligned} \tag{5}$$

For the special of $\boldsymbol{\mu}_{\boldsymbol{w}} = \mathbf{0}$ and $\mathbf{\Sigma}_w = \tau\mathbf{I}$ the Bayesian regression reduces to ridge regression with $\lambda = \frac{\sigma_\epsilon^2}{\tau}$

Probabilistic Linear Regression

Posterior Predictive Distribution

The predictive distribution is a distribution over output y and allows to make predictions given a set of testing data $\mathbf{x}_*$

$$p\left(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}\right) = \int \underbrace{p(\mathbf{y}|\mathbf{X}, \hat{\mathbf{w}}, \sigma_\epsilon^2 \mathbf{I}_M)}_{\text{Likelihood}} \underbrace{p\left(\mathbf{w}|\mathbf{X}, \mathbf{y}\right)}_{\text{Posterior}} \partial \mathbf{w} \tag{6}$$

The integral has a closed form solution in the case of Normal distributions

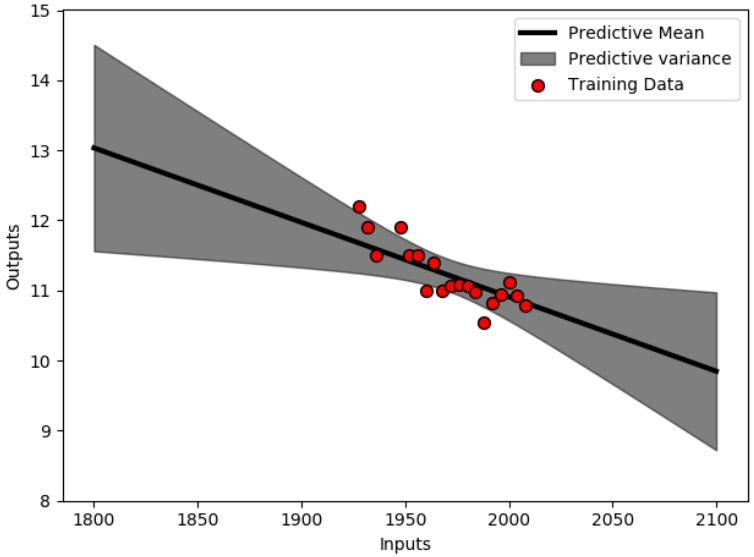
$$p\left(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}\right) \sim \mathcal{N}\left(\mu_{y^*}, \sigma^2_{\mathbf{y}^*}\right)$$

assuming $\mu_w = 0$

$$\begin{aligned} \mu_{y^*} &= \mathbf{x}_*^T \hat{\mathbf{w}}_{\text{MAP}} = \frac{1}{\sigma_\epsilon^2} \mathbf{x}_*^T \mathbf{A}^{-1} \mathbf{X}\mathbf{y} \\ \sigma^2_{y^*} &= \mathbf{x}_*^T \mathbf{A}^{-1} \mathbf{x}_* \end{aligned} \tag{7}$$

Probabilistic Linear Regression

Posterior Predictive Distribution



$$p\left(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}\right) \sim \mathcal{N}\left(\mu_{y^*}, \sigma^2_{\mathbf{y}^*}\right)$$

$$\mu_{y^*}\left(\mathbf{x}_*\right) = \mathbf{x}_*^T \hat{\mathbf{w}}_{\text{MAP}} = \frac{1}{\sigma_\epsilon^2} \mathbf{x}_*^T \mathbf{A}^{-1} \mathbf{X}\mathbf{y}$$

$$\sigma^2_{y^*}\left(\mathbf{x}_*\right) = \mathbf{x}_*^T \mathbf{A}^{-1} \mathbf{x}_*$$

The variance gives a measure of the uncertainty of the prediction

Probabilistic Linear Regression

Marginal Likelihood

Bayesian Regression depends on hyper-parameters
The marginal likelihood provides a metric for optimization
Bayes Rule:

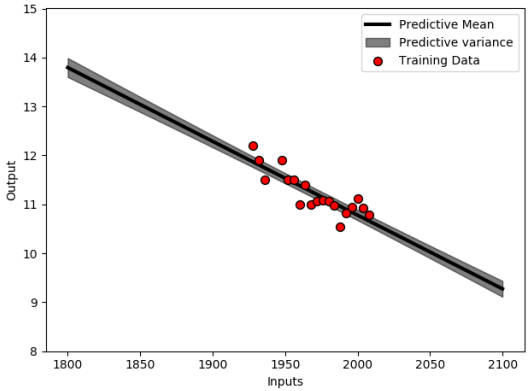
$$p\left(\mathbf{w}|\mathbf{X},\mathbf{y}\right) = \frac{p(\mathbf{y}|\mathbf{X},\hat{\mathbf{w}},\sigma_\epsilon^2)p\left(\mathbf{w}\right)}{\underbrace{p\left(\mathbf{y}|\mathbf{X}\right)}_{MarginalLikelihood}}$$

$$p\left(\mathbf{y}|\mathbf{X}\right) = \int p(\mathbf{y}|\mathbf{X},\hat{\mathbf{w}},\sigma_\epsilon^2)p\left(\mathbf{w}\right)d\mathbf{w} \tag{8}$$

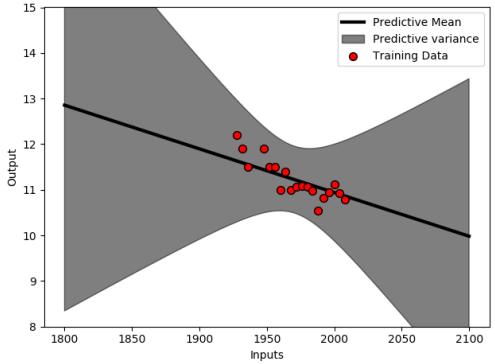
Measures how well our model explains the observed data. For the current case, it has a closed form solution

Probabilistic Linear Regression

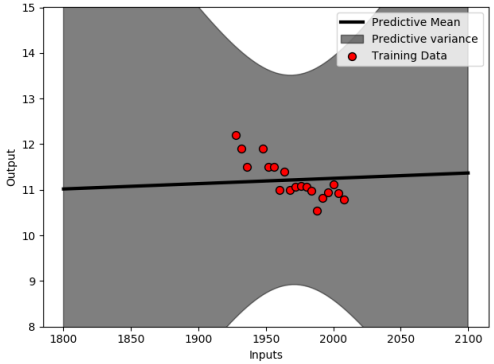
Marginal Likelihood



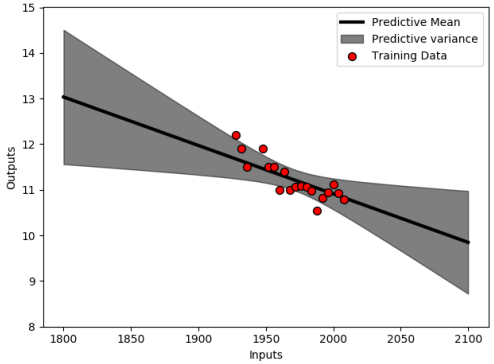
(a) 33.38



(b) 47.19



(c) 67.19



(d) 27.32

Figure: Negative Log-Marginal Likelihood for different models

Gaussian Process for Regression

From probabilistic linear to nonlinear regression

Starting with the predictive distribution of Bayesian linear regression, Apply non-linear mapping $\phi(\mathbf{x})$ to the feature space.

$$p(y_*|\boldsymbol{\phi}_*, \boldsymbol{\Phi}, \mathbf{y}) \sim \mathcal{N}\left(\mu_{y^*}, \sigma^2_{y^*}\right)$$

$$\mu_{y^*}(\boldsymbol{\phi}_*) = \frac{1}{\sigma^2_\epsilon}\boldsymbol{\phi}_*^T \mathbf{A}^{-1}\boldsymbol{\Phi}\mathbf{y}$$

$$\sigma^2_{y^*}(\boldsymbol{\phi}_*) = \boldsymbol{\phi}_*^T \mathbf{A}^{-1}\boldsymbol{\phi}_*$$

$$\mathbf{A} = \frac{1}{\sigma^2_\epsilon}\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \boldsymbol{\Sigma}_\mathbf{w}^{-1}$$

The $p(y_*|\boldsymbol{\phi}_*, \boldsymbol{\Phi}, \mathbf{y})$ is a Normal distribution over functions.

$$\left.\begin{array}{l}\boldsymbol{\phi}_*^T \mathbf{A}^{-1}\boldsymbol{\Phi} \\ \boldsymbol{\phi}_*^T \mathbf{A}^{-1}\boldsymbol{\phi}_*\end{array}\right\} \text{ Inner product in feature space}$$

Gaussian Process for Regression

From probabilistic linear to nonlinear regression

Define kernel as: $k(x, x') = \boldsymbol{\phi}(\mathbf{x}) \boldsymbol{\Sigma}_{\mathbf{w}}^{-1} \boldsymbol{\phi}(\mathbf{x}')$ and apply on the mean and variance[2]

$$\mathbb{E}\{p(y_* | \boldsymbol{\phi}_*, \boldsymbol{\Phi}, \mathbf{y})\} = \sum_{i=1}^{M} \alpha_i k(\mathbf{x}_i, \mathbf{x}_*)$$

$$\text{where: } \mathbf{a} = \left[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_y^2 \mathbf{I}\right]^{-1} \mathbf{y}$$

$$\text{Var}\{p(y_* | \boldsymbol{\phi}_*, \boldsymbol{\Phi}, \mathbf{y})\} = k(\mathbf{x}_*, \mathbf{x}_*) + \mathbf{k}(\mathbf{x}_*, \mathbf{X}) \left[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_y^2 \mathbf{I}\right]^{-1} \mathbf{k}(\mathbf{X}, \mathbf{x}_*)$$

In GPs all the training data are used for predictions!

[2] See supplement material for steps

Gaussian Process for Regression

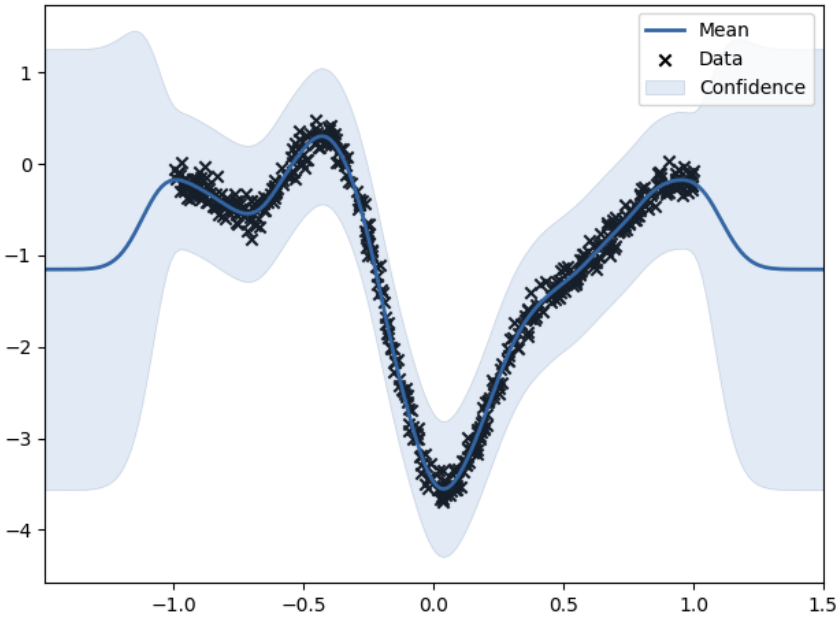
Kernels

Kernel functions:

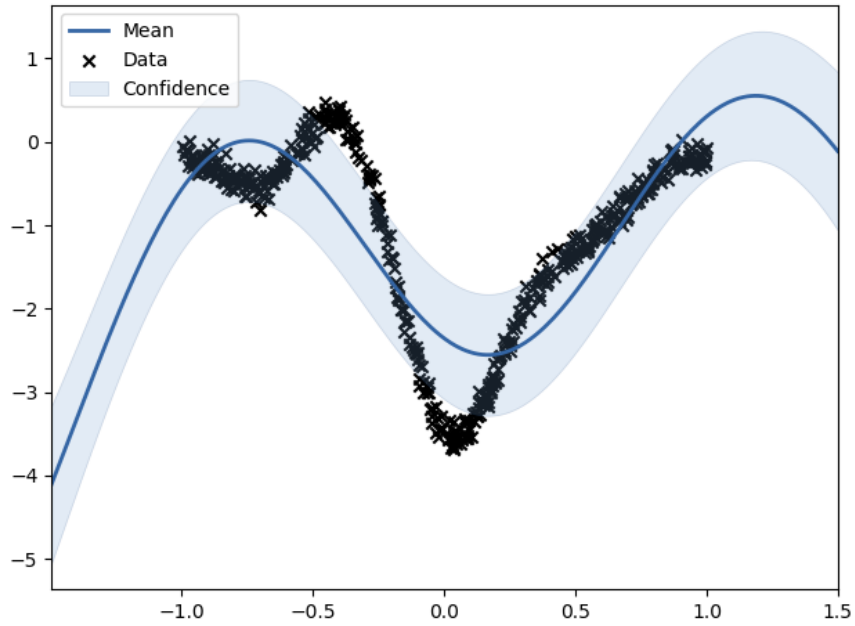
- Linear
  $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}' + c$ where c the intercept $\equiv$ Bayes regression
- Gaussian (RBF)
  $k(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{||\mathbf{x}-\mathbf{x}'||}{\ell}\right)$ where ℓ the kernel width (lengthscale)
- Polynomial
  $k(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T \mathbf{x}' + c\right)^d$ where d is the degree of polynomial
- Periodic
  $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{2\sin^2(\pi||\mathbf{x}-\mathbf{x}'||/p)}{\ell}\right)$ si relation périodique
  where p is the period.

Gaussian Process for Regression

Kernels – RBF



(a) $\ell = 0.1$

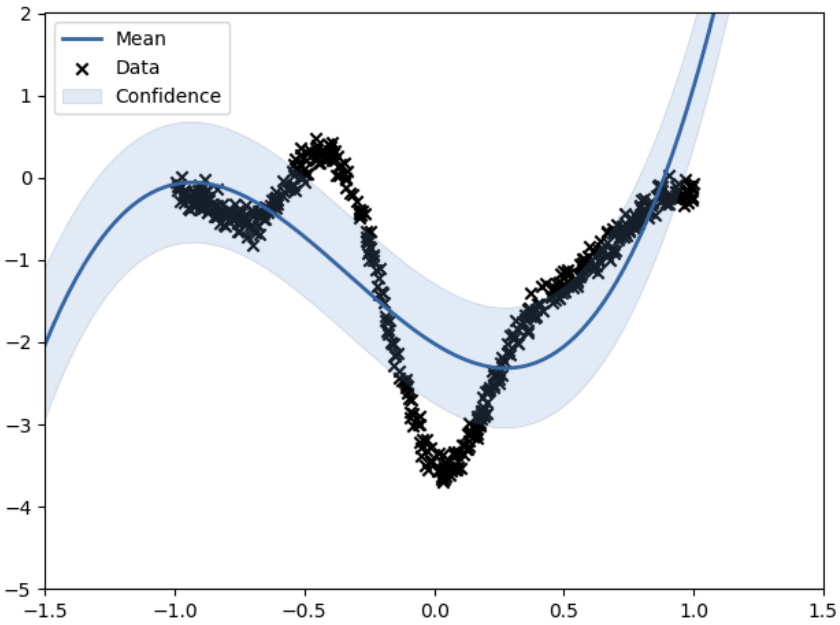


(b) $\ell = 1$

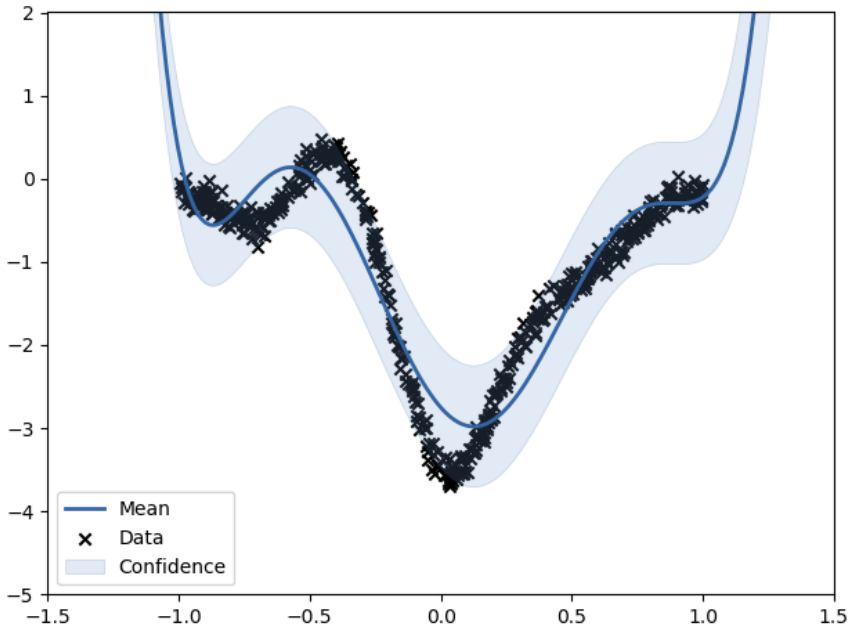
Figure: Impact of the width of RBF kernel

Gaussian Process for Regression

Kernels – Polynomial



(a) $d = 3$

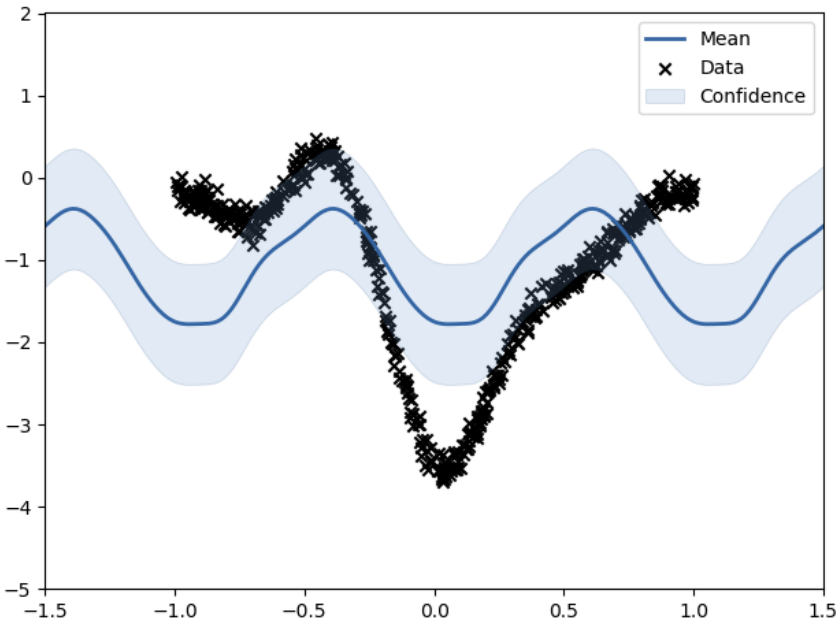


(b) $d = 6$

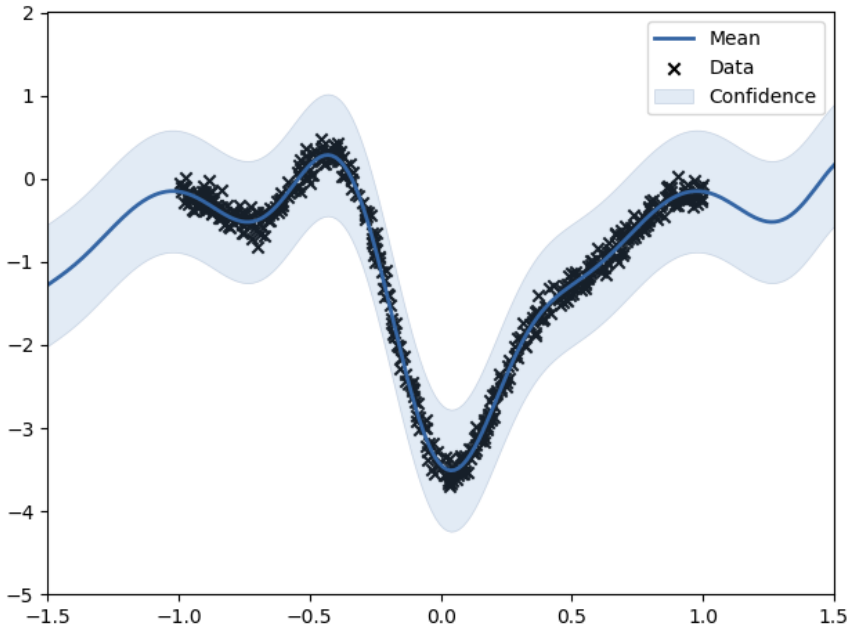
Figure: Impact of the degree of polynomial kernel

Gaussian Process for Regression

Kernels – Periodic



(a) $p = 1$



(b) $p = 2$

Figure: Impact of the period of periodic kernel

Gaussian Process for Regression

Construction of new kernels

Not all kernels are appropriate for the all datasets.
There exists many kernels in literature.
As seen in kernel lecture,new kernels can be created by combination of kernels:

valid kernel: symmetric and positive definite

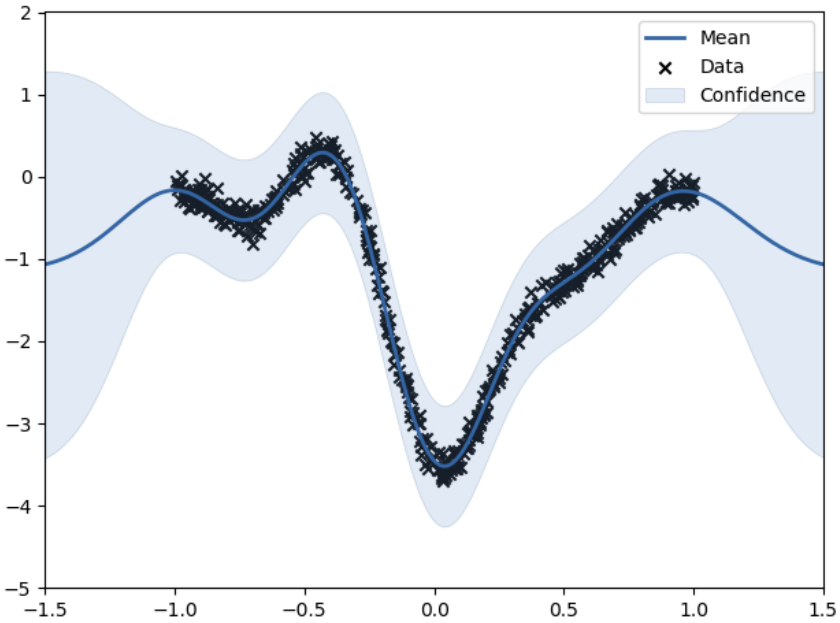
$$k\left(\mathbf{x}, \mathbf{x}'\right) = ck_1\left(\mathbf{x}, \mathbf{x}'\right) \text{ where } c > 0$$
$$k\left(\mathbf{x}, \mathbf{x}'\right) = f(\mathbf{x})k_1\left(\mathbf{x}, \mathbf{x}'\right) f(\mathbf{x}') \text{ where } f\left(\cdot\right) \text{ any function}$$
$$k\left(\mathbf{x}, \mathbf{x}'\right) = k_1\left(\mathbf{x}, \mathbf{x}'\right) + k_2\left(\mathbf{x}, \mathbf{x}'\right)$$
$$k\left(\mathbf{x}, \mathbf{x}'\right) = k_1\left(\mathbf{x}, \mathbf{x}'\right) k_2\left(\mathbf{x}, \mathbf{x}'\right)$$
$$k\left(\mathbf{x}, \mathbf{x}'\right) = \exp(k_1\left(\mathbf{x}, \mathbf{x}'\right))$$
$$k\left(\mathbf{x}, \mathbf{x}'\right) = k_1\left(\mathbf{x}_a, \mathbf{x}'_a\right) + k_2\left(\mathbf{x}_b, \mathbf{x}'_b\right) \text{ where } a, b \text{ dimensions of } \mathbf{x}$$
$$k\left(\mathbf{x}, \mathbf{x}'\right) = k_1\left(\mathbf{x}_a, \mathbf{x}'_a\right) k_2\left(\mathbf{x}_b, \mathbf{x}'_b\right)$$

↑ peut avoir diff kernel sur diff dimensions

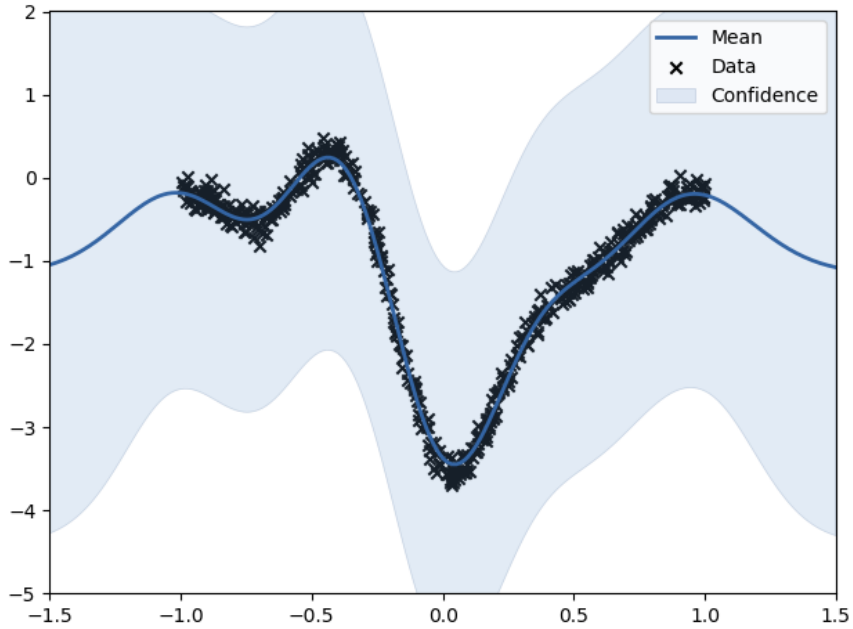
Gaussian Process for Regression

The noise

The larger the noise, the more uncertainty



(a) $\sigma_y^2 = 0.1$



(b) $\sigma_y^2 = 1$

Figure: Impact of noise variance at the confidence interval

Gaussian Process for Regression

Hyper-parameter tuning

Hyper-parameters: kernel's parameters and noise variance

Tuning:

- Cross-validation
- Minimize Marginal-likelihood (local optima)
  s.t. hyper-parameters, where $\mathbf{K}_{\sigma_y^2} = \left[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_y^2 \mathbf{I}\right]$:

$$-\log(\mathbf{y}|\mathbf{X}) = 0.5(\underbrace{\mathbf{y}^T \mathbf{K}_{\sigma_y^2}^{-1} \mathbf{y}}_{\text{Fit}} + \underbrace{\log|\mathbf{K}_{\sigma_y^2}|}_{\text{Complexity}} + \log 2\pi) \tag{9}$$

p ex RBF σ petit good fit mais |K| ↗

Automatically provides trade-off between model fit and complexity[3]

[3] See supplement material

Gaussian Process for Regression

Zero Mean Prior

Prior expectations

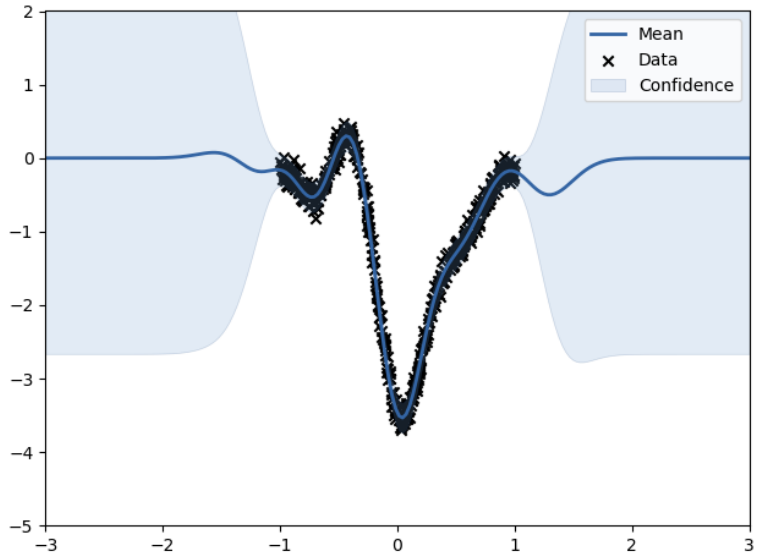
Linear Model:

$$\mathbb{E}\{y\} = \mathbb{E}\left\{w^T x + \mathcal{N}\left(0, \sigma_y^2\right)\right\}$$
$$= \mathbb{E}\left\{w^T\right\} x + 0$$

Non-Linear Model:

$$\mathbb{E}\{y\} = \mathbb{E}\left\{w^T\right\} \phi(x) + \mathcal{N}\left(0, \sigma_y^2\right)$$
$$= \mathbb{E}\left\{w^T\right\} \phi(x) + 0$$

Both models have a zero mean Gaussian prior.
GPs with RBF kernel, predict zero far from the data



Gaussian Process for Regression

Mean function

Instead of zero mean one can use a mean function $m(\cdot) : x \mapsto y$.

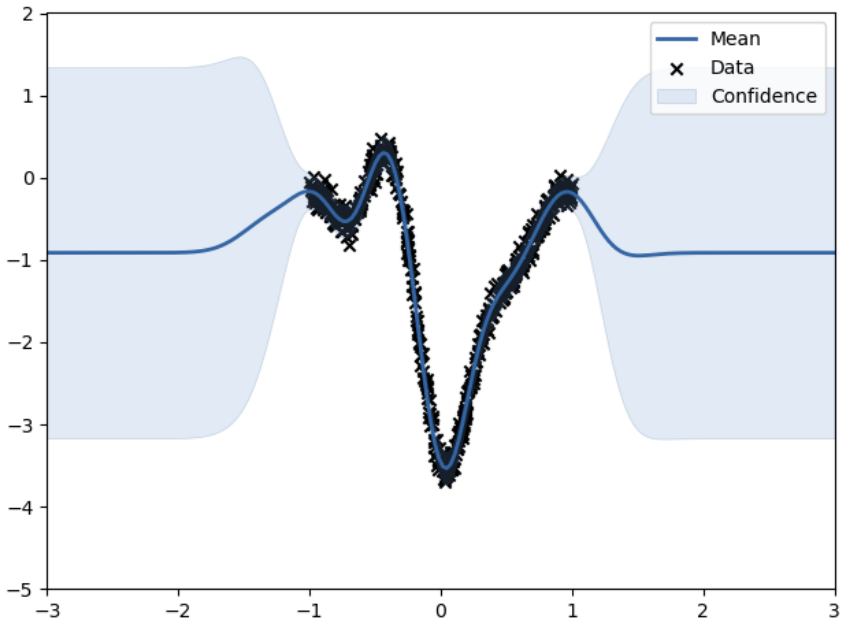
$$\mathbb{E}\left\{p\left(y_* | \boldsymbol{\phi}_*, \boldsymbol{\Phi}, \mathbf{y}\right)\right\} = \sum_{i=1}^{M} m\left(\mathbf{x}_*\right) + \alpha_i k\left(\mathbf{x}_i, \mathbf{x}_*\right) \tag{10}$$

$$\text{where: } \mathbf{a} = \left[\mathbf{K}\left(\mathbf{X}, \mathbf{X}\right) + \sigma_y^2 \mathbf{I}\right]^{-1} \left(\mathbf{y} - m\left(\mathbf{x}_*\right)\right)$$

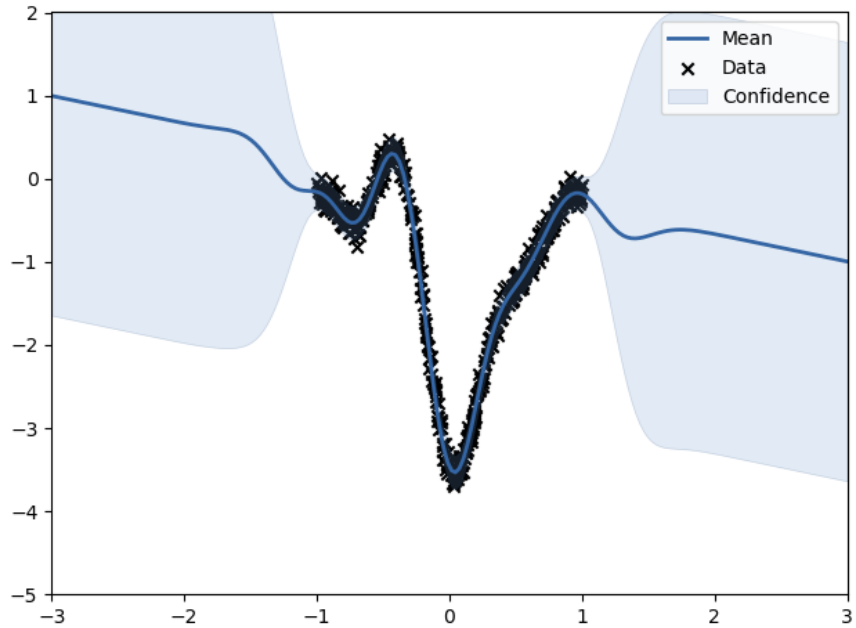
The parameters of the mean can be auto-tuned based on the marginal likelihood

Gaussian Process for Regression

Impact of mean function



(a) Constatnt mean



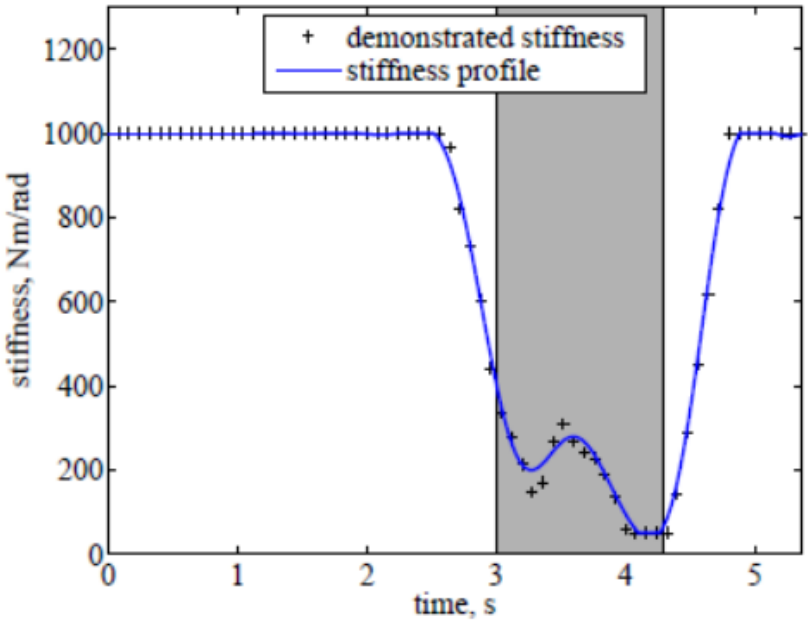
(b) Linear Mean

Figure: GPs with different mean functions

Gaussian Process for Regression

Applications – Learning from demonstration

Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction[4]



The stiffness profile is encoded as a time-varying input using GPR. The shaded area corresponds to the striking phase.

[4] Kronander, K. and Billard, A. (2013) Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction. IEEE Transactions on Haptics. 10.1109/TOH.2013.54.

Gaussian Process for Regression

Applications – Surface models

GPR can be used to model the shape of objects[5]

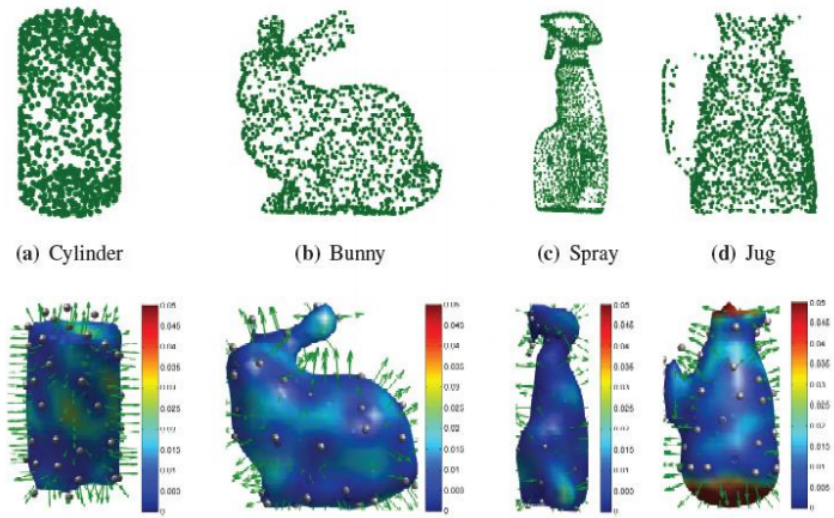


Figure: Top: 3D points sampled either from a camera or from tactile sensing. Bottom: 3D shape reconstructed by GPR. The arrows represent the predicted normals at the surface

[5] El Khoury, S., Li, M., and Billard, A. (2013). On the generation of a variety of grasps. Robotics and Autonomous Systems, 61(12):1335–34

Gaussian Process for Classification

GPs from the Bayesian perspective

Given training data $\mathbf{X}, \mathbf{y}$, infer continuous function f such as:

$$y = f(\mathbf{x}) + \epsilon \text{ ,where: } \epsilon \sim \mathcal{N}\left(0, \sigma_y^2\right)$$

GPs are jointly normal distributions over the outputs of the latent function $p(f) \sim \mathcal{GP}(m(x), k(x, x'))$.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ m(x_2) \\ \vdots \\ m(x_3) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) + \sigma_y^2 & k(x_1, x_2) & \dots & k(x_1, x_M) \\ k(x_2, x_1) & k(x_2, x_2) + \sigma_y^2 & \dots & k(x_2, x_M) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_M, x_1) & k(x_M, x_2) & \dots & k(x_M, x_M) + \sigma_y^2 \end{bmatrix} \right)$$

The prediction for f_* for a novel input x_* can also be made by conditioning at the joint Normal $p(f_* | \mathbf{f}, \mathbf{X}, \mathbf{y})$

Gaussian Process for Classification

GPs from the Bayesian perspective

Given training data $\mathbf{X}, \mathbf{y}$, infer a continuous function f such as:

$$y = f(\mathbf{x}) + \epsilon \text{ ,where: } \epsilon \sim \mathcal{N}(0, \sigma_y^2)$$

GPs are probability distributions over a latent function $p(f)$.
Bayes rule for GPs:

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

Prior :	$p(\mathbf{f}) \sim \mathcal{GP}(0, k(x, x'))$
Likelihood:	$p(\mathbf{y}|\mathbf{f}, \mathbf{X})$ Normal for regression
Posterior:	$p(\mathbf{f}|\mathbf{X}, \mathbf{y}) \sim \mathcal{GP}(m_{\text{post}}, k_{\text{post}}(x, x'))$

For Normal likelihood and a GP prior the posterior has a closed form solution

Gaussian Process for Classification

GPs from the Bayesian perspective

Given data $\mathbf{X}, \mathbf{y} \in [0; 1]$, for a binary classification problem, infer a continuous function f such as:

$$y = \lambda\left(f\left(\mathbf{x}\right) + \epsilon\right) \text{ ,where: } \epsilon \sim \mathcal{N}\left(0, \sigma_y^2\right) \text{ and} \lambda\left(f\left(\mathbf{x}\right) + \epsilon\right) = \frac{1}{1 + \exp\left(y_i f_i + \epsilon\right)}$$

$$p\left(\mathbf{f} | \mathbf{X}, \mathbf{y}\right) = \frac{p\left(\mathbf{y} | \mathbf{f}, \mathbf{X}\right) p\left(\mathbf{f}\right)}{p\left(\mathbf{y} | \mathbf{X}\right)}$$

Prior :	$p\left(\mathbf{f}\right) \sim \mathcal{GP}\left(0, k\left(x, x'\right)\right)$
Likelihood:	$p\left(\mathbf{y} | \mathbf{f}, \mathbf{X}\right) \sim \lambda\left(y_i f_i + \epsilon\right)$ if $y_i = 1$ otherwise $1 - \lambda\left(y_i f_i + \epsilon\right)$
Posterior:	$p\left(\mathbf{f} | \mathbf{X}, \mathbf{y}\right)$ No closed form solution!!!

The logistic likelihood and the GP prior are not conjugate

Gaussian Process for Classification

Approximation Methods

Many methods exist to approximate the posterior:

- Laplace approximation
  Taylor series expansion about mode of posterior
- Expectation propagation
  Minimize divergence between approximated and true distribution
- Markov Chain Monte Carlo
  Take a large amount of samples from posterior (more computationally expensive)

Gaussian Process for Classification

Laplace Approximation

Approximate as a Normal distribution:

$$\tilde{p}\left(\mathbf{f}|\mathbf{X},\mathbf{y}\right) \sim \mathcal{N}\left(\tilde{\mathbf{f}}, \mathbf{C}^{-1}\right)$$

$$\text{where: } \tilde{\mathbf{f}} = \arg\max_{\mathbf{f}} p\left(\mathbf{f}|\mathbf{X},\mathbf{y}\right) \tag{11}$$

$$\mathbf{C} = -\nabla^2 \log p\left(\mathbf{f}|\mathbf{X},\mathbf{y}\right)|_{\mathbf{f}=\tilde{\mathbf{f}}}$$

↑ curvature

Gaussian Process for Classification

Laplace approximation of the posterior

Find $\tilde{\mathbf{f}}$ by maximizing log-posterior :

$$\log p\left(\mathbf{f}|\mathbf{X},\mathbf{y}\right) = \log p\left(\mathbf{y}|\mathbf{f},\mathbf{X}\right) + \log p\left(\mathbf{f}\right)$$

Differentiating w.r.t $\mathbf{f}$

$$\begin{aligned}\nabla \log p\left(\mathbf{f}|\mathbf{X},\mathbf{y}\right) =& \nabla \log p\left(\mathbf{y}|\mathbf{f},\mathbf{X}\right) - \mathbf{K}^{-1}\mathbf{f} \\ \nabla^2 \log p\left(\mathbf{f}|\mathbf{X},\mathbf{y}\right) =& \nabla^2 \log p\left(\mathbf{y}|\mathbf{f},\mathbf{X}\right) - \mathbf{K}^{-1} \\ =& -\mathbf{D} - \mathbf{K}^{-1}\end{aligned}$$

$$\tilde{p}\left(\mathbf{f}|\mathbf{X},\mathbf{y}\right) \sim \mathcal{N}\left(\tilde{\mathbf{f}}, \left(\mathbf{D} + \mathbf{K}^{-1}\right)^{-1}\right) \tag{12}$$

Gaussian Process for Classification

Making predictions

Posterior predictive distribution:

$$\tilde{p}(\mathbf{f}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \int p\left(\mathbf{f}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}, \mathbf{f}\right) \tilde{p}\left(\mathbf{f}|\mathbf{X}, \mathbf{Y}\right) d\mathbf{f}$$

$$\mathbb{E}_{\tilde{p}}\left[\mathbf{f}_*|\,\mathbf{X}, \mathbf{Y}, \mathbf{x}_*\right] = \sum_{i=1}^{M} \alpha_i \mathbf{k}\left(\mathbf{x}_*, \mathbf{x}_i\right), \text{where: } \mathbf{a} = \left[\mathbf{K}\left(\mathbf{X}, \mathbf{X}\right) + \sigma_y^2 \mathbf{I}\right]^{-1} \tilde{\mathbf{f}}$$

$$\mathrm{Var}_{\tilde{p}}\left[\mathbf{f}_*|\,\mathbf{X}, \mathbf{Y}, \mathbf{x}_*\right] = k\left(\mathbf{x}_*\mathbf{x}_*\right) - \mathbf{k}\left(\mathbf{X}\mathbf{x}_*\right)^T \left(\mathbf{D} + \mathbf{K}^{-1}\right)^{-1} \mathbf{k}\left(\mathbf{X}, \mathbf{x}_*\right)$$

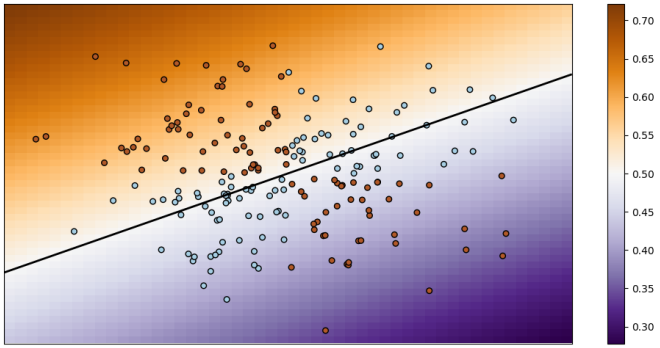
Class predictions:

$$p\left(y_* = 1|\mathbf{x}_*, \mathbf{X}_*, \mathbf{Y}\right) = \int p\left(y_* = 1|\mathbf{f}_*\right) \tilde{p}(\mathbf{f}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) d\mathbf{f}_*$$

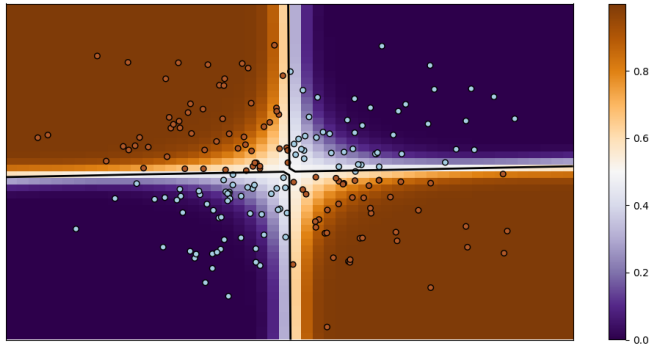
No closed form solution, has to approximated

Gaussian Process for Classification

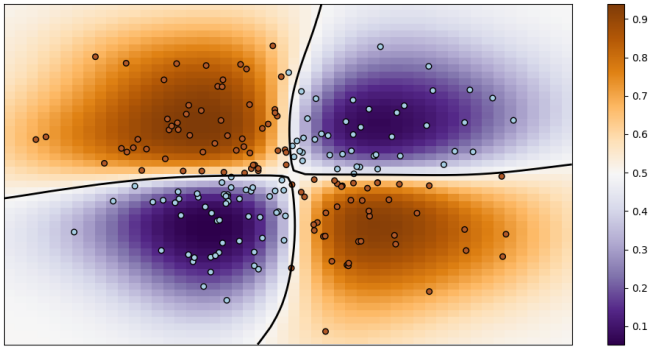
Toy Examples



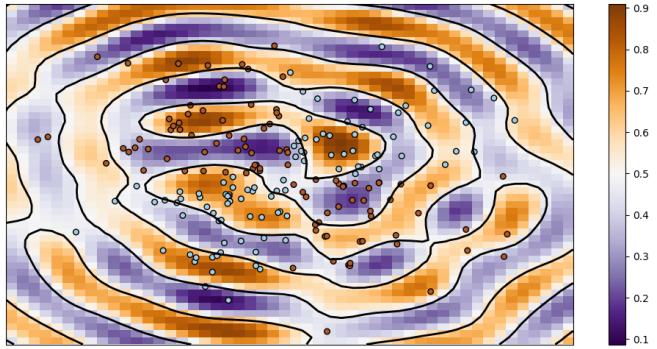
(a) Linear



(b) Second degree polynomial



(c) RBF

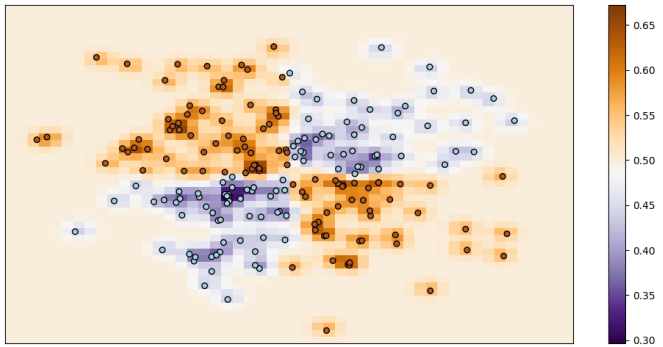


(d) Periodic

Figure: GP classification of a XOR dataset for different kernels

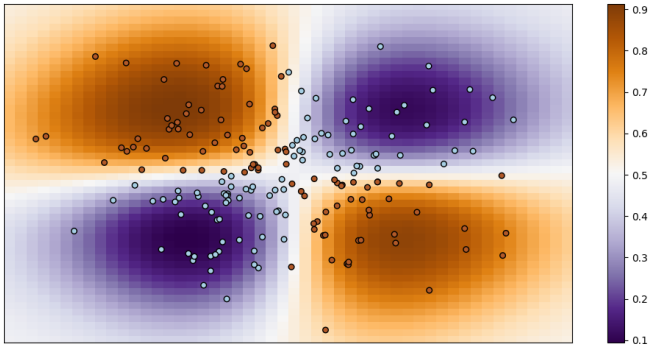
Gaussian Process for Classification

Impact of kernel width

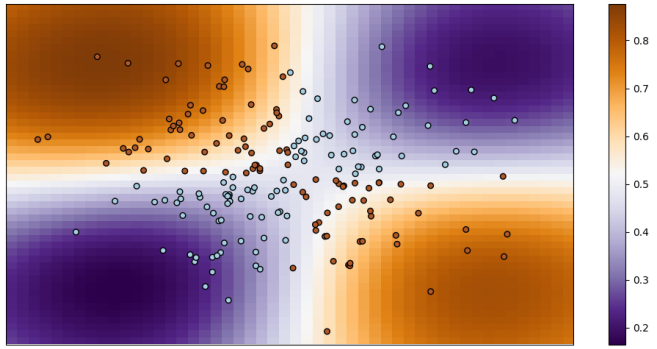


kernel on every points

(a) width = 0.1



(b) width = 1



(c) width = 2

Figure: GP classification of a XOR dataset using RBF kernel with various widths

GPLVM

Outline

- GPLVM is an unsupervised variation of GPs
- Probabilistic dimensionality reduction

Derivation:

- PCA to Probabilistic PCA
- Dual Formulation of Probabilistic PCA
- GPLVM through kernel trick

GPLVM

From PCA to Probabilistic PCA

M data $\mathbf{Y} \in \Re^d$ at the original space and $\mathbf{X}$ their projection to the laten space.

PCA: $\mathbf{y} = \mathbf{A}\mathbf{x}$

Propbabilistic PCA

- Introduce stochasticity by adding noise variable $\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \sigma^2 \mathbf{I}\right)$
  $\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}$
- Likelihood:
  $p\left(\mathbf{y}|\mathbf{A}, \mathbf{x}\right) \sim \mathcal{N}\left(\mathbf{A}\mathbf{x}, \sigma^2 \mathbf{I}\right)$
- Integrate over $\mathbf{x}$ to get marginal likelihood
  $p\left(\mathbf{y}|\mathbf{A}\right) = \int p\left(\mathbf{y}|\mathbf{A}, \mathbf{x}\right) p\left(\mathbf{x}\right)$
- Requires prior over $\mathbf{x}$:
  $p\left(\mathbf{x}\right) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$ ← un informative et besoin GP

GPLVM

From PCA to Probabilistic PCA

Marginal likelihood has a closed form solution:

$$p(\mathbf{y}|\mathbf{A}) = \mathcal{N}\left(\mathbf{0}, \mathbf{A}\mathbf{A}^T + \sigma^2\mathbf{X}\right)$$

Assuming $\mathbf{Y}$ are i.i.d:

$$p(\mathbf{Y}|\mathbf{A}) = \prod_{i=1}^{M} p(\mathbf{y}_n|\mathbf{A})$$

Maximizing marginal likelihood wrt to $\mathbf{A}$ yields unique optimal solution[6]

[6] Tipping and Bishop. "Probabilistic principal component analysis."

GPLVM

Dual Probabilistic PCA

Probabilistic PCA: Integrate over $\mathbf{x}$ and optimize w.r.t to $\mathbf{A}$

$$p(\mathbf{y}|\mathbf{A}) = \int p(\mathbf{y}|\mathbf{A},\mathbf{x})\, p(\mathbf{x})$$

Dual Probabilistic PCA Integrate over $\mathbf{A}$ and optimize w.r.t to $\mathbf{x}$

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{A},\mathbf{x})\, p(\mathbf{A})$$

Prior over $\mathbf{A}$:

$$p(\mathbf{A}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{a}_d; \mathbf{0}, \mathbf{I})$$

GPLVM

Dual Probabilistic PCA

Marginal likelihood:

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{d=1}^{D} \mathcal{N}\left(\mathbf{y}_d; \mathbf{0}, \mathbf{X}^T\mathbf{X} + \sigma^2\mathbf{I}\right)$$

GPLVM:

- $\mathbf{X}^T\mathbf{X} = \mathbf{K}(\mathbf{x}, \mathbf{x}')$ Inner product allows to apply kernel trick
- $\mathcal{N}\left(\mathbf{y}_d; \mathbf{0}, \mathbf{K}(\mathbf{x}, \mathbf{x}') + \sigma^2\mathbf{I}\right)$ A Gaussian process on each dimension of $\mathbf{Y}$
- Minimize negative log-marginal likelihood w.r.t $\mathbf{X}$

$$-\log p(\mathbf{y}_d|\mathbf{X}) = \frac{1}{2}\left(DN\log 2\pi + D\log|\mathbf{K}| + \mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}^T\mathbf{Y}\right)\right)$$

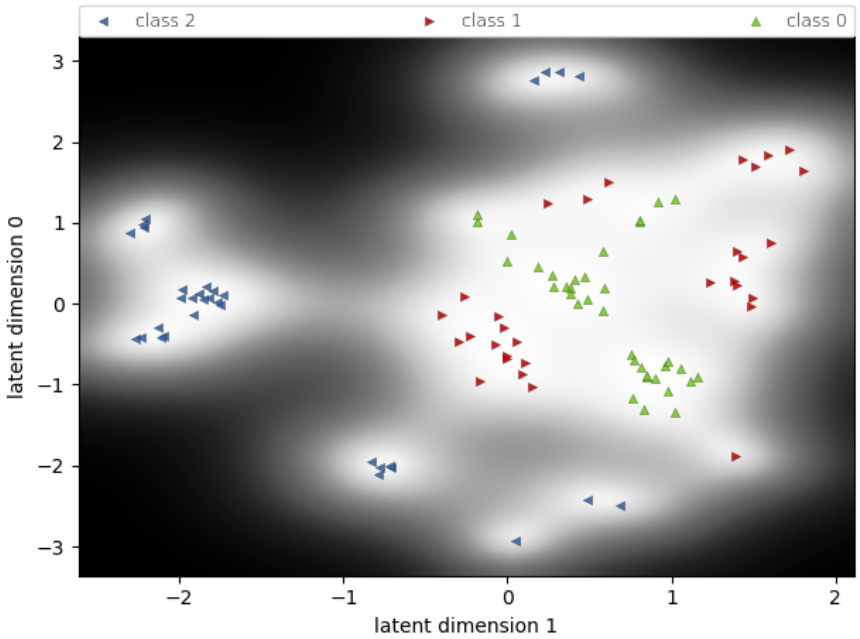
GPLVM

Comaprison with kPCA

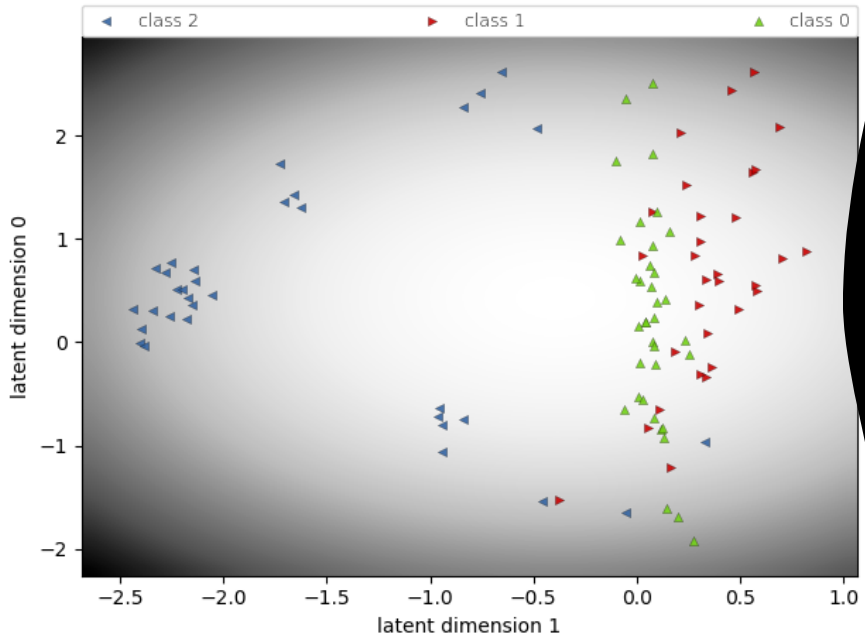
Method	Proximity	Mapping	Non-Linear	Probabilistic
GPLVM	YES	$\mathbf{X} \rightarrow \mathbf{Y}$	YES	YES
kPCA	YES	$\mathbf{Y} \rightarrow \mathbf{X}$	YES	NO

GPLVM

Examples



(a) RBF kernel



(b) Linear Kernel

Figure: GPLVM with different kernels

GPLVM

Choosing latent dimensions

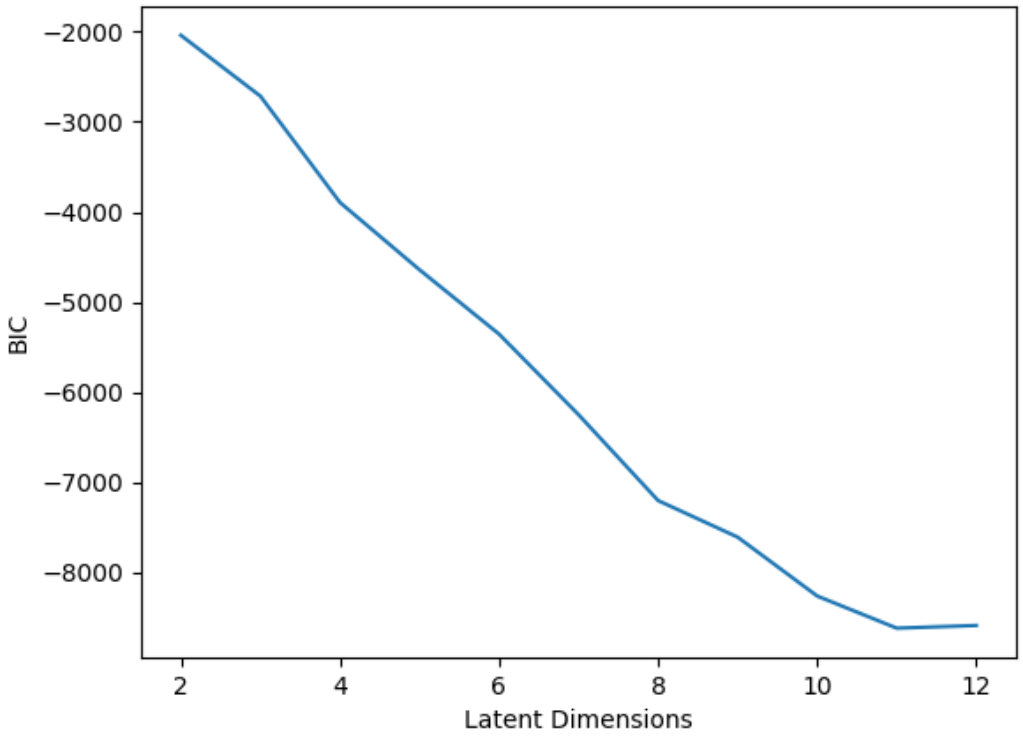


Figure: BIC as function of the number of latent dimensions

$$\text{BIC} = M \log(k) - 2 \log(p(\mathbf{Y}|\mathbf{X})) \tag{13}$$

GPLVM

Choosing latent dimensions

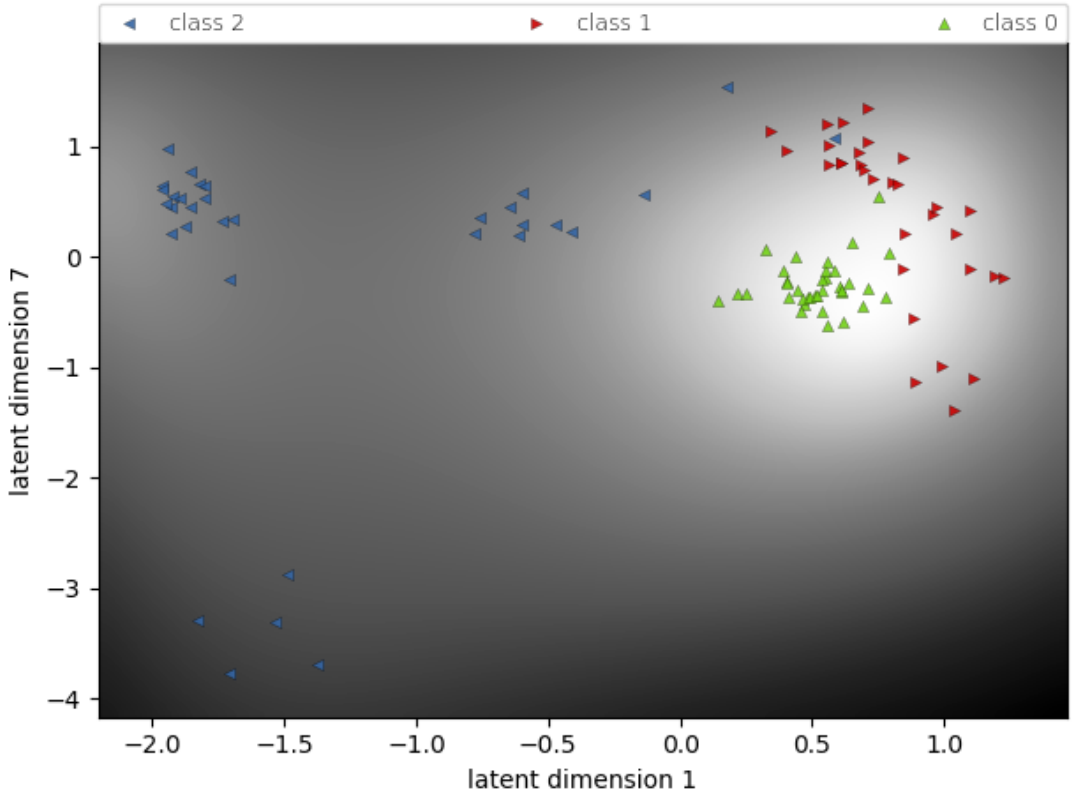


Figure: Combination of latent dimensions which increase separability

Gaussian Processes

Discussion

- Advantages
  - Generalization
  - Accuracy
  - Metric of Uncertainty
  - Auto-tuning of hyper-parameters
- Disadvantages
  - Computational complexity
- Recommended packages
  - GPy
  - scikit-learn