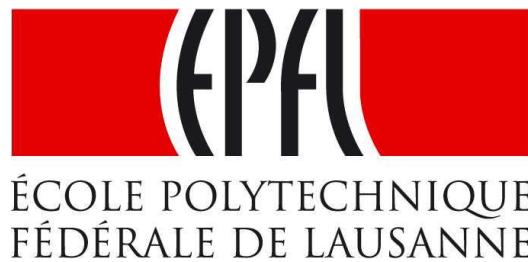


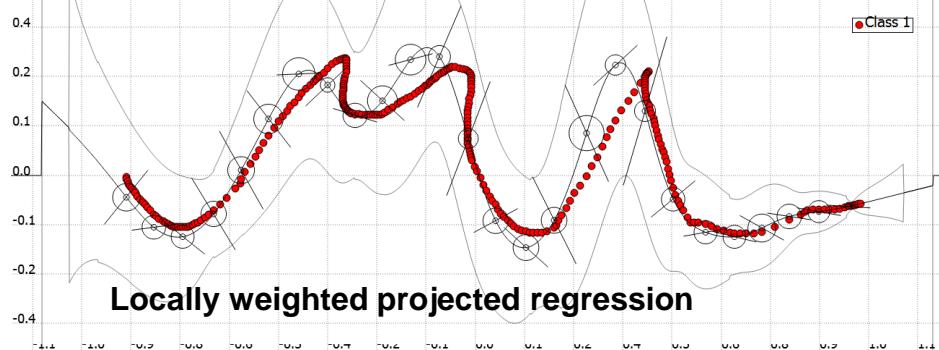
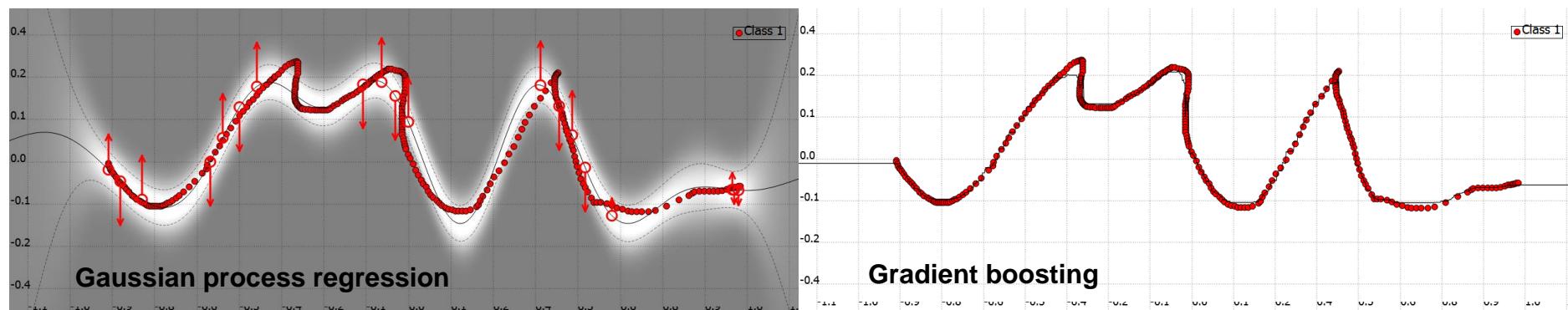
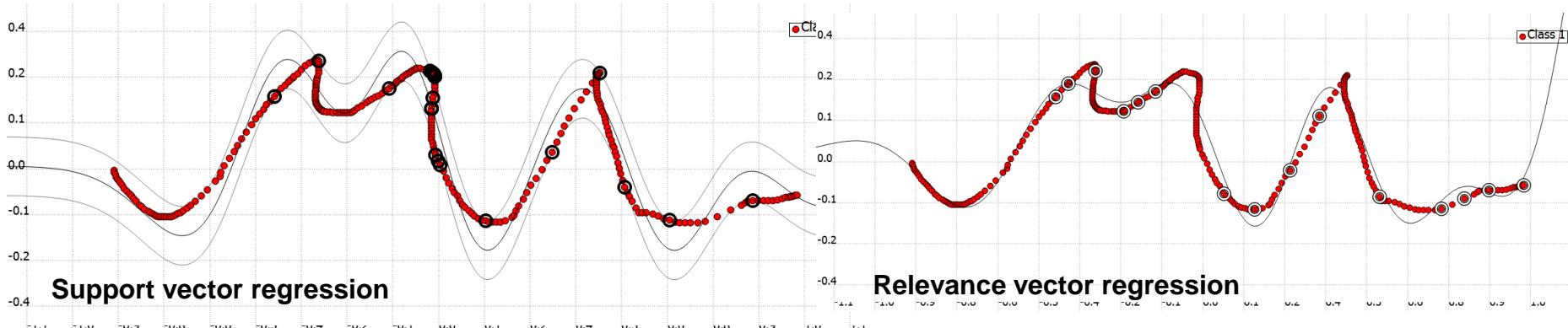
# ***ADVANCED MACHINE LEARNING***

## **Non-linear regression techniques**

### **Part - II**

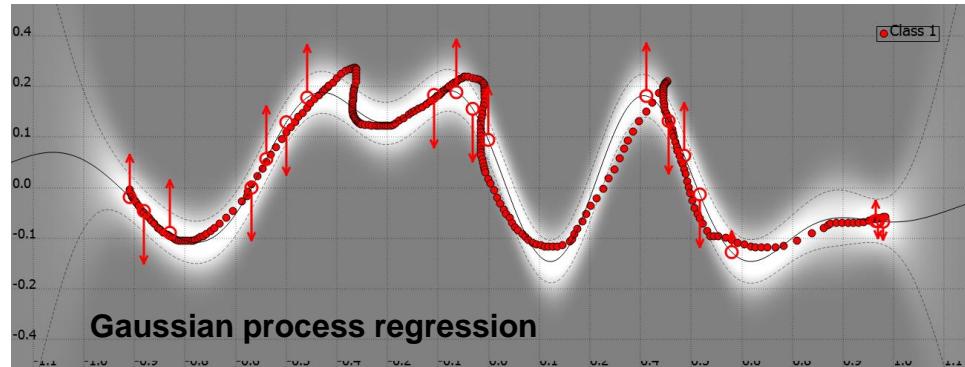


# Regression Algorithms in this Course



Not covered – replaced by one hour to answer questions about mini-project

# Regression Algorithms in this Course



# Probabilistic Regression (PR)

PR is a statistical approach to classical linear regression that estimates the relationship between zero-mean variables  $y$  and  $x$  by building a linear model of the form:

$$y = f(x, w) = w^T x, \quad w, x \in \mathbb{R}^N$$

If one assumes that the observed values of  $y$  differ from  $f(x)$  by an additive noise  $\varepsilon$  that follows a zero-mean Gaussian distribution (such an assumption consists of putting a *prior distribution* over the noise), then:

$$y = w^T x + \varepsilon, \quad \text{with } \varepsilon = N(0, \sigma^2)$$

Where have we seen this before?

# Probabilistic Regression

Training set of  $M$  pairs of data points  $\{X, \mathbf{y}\} = \left\{x^i, y^i\right\}_{i=1}^M$

Likelihood of the regressive model

$$\begin{aligned} \mathbf{y} &= w^T X + N(0, \sigma^2) \\ \Rightarrow \mathbf{y} &\sim p(\mathbf{y} | X, w, \sigma) \end{aligned}$$

**Parameters of the model**

Data points are independently and identically distributed (i.i.d)

$$\begin{aligned} p(\mathbf{y} | X, w, \sigma) &\sim \prod_{i=1}^M p(y^i | x^i, w, \sigma) \\ &= \prod_{i=1}^M \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^i - w^T x^i)^2}{2\sigma^2}\right) \end{aligned}$$

# Probabilistic Regression

Training set of  $M$  pairs of data points  $\{X, \mathbf{y}\} = \{x^i, y^i\}_{i=1}^M$

Likelihood of the regressive model

$$\mathbf{y} = \mathbf{w}^T \mathbf{X} + N(0, \sigma^2)$$
$$\Rightarrow \mathbf{y} \sim p(\mathbf{y} | \mathbf{X}, \mathbf{w}, \sigma)$$

Parameters of the model

Hyperparameters Given by user

Prior model on distribution of parameter  $w$ :

$$p(w) = N(0, \Sigma_w) \propto \exp\left(-\frac{1}{2} w^T \Sigma_w^{-1} w\right)$$

# Probabilistic Regression

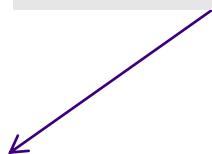
Prior on  $w$ :  $p(w) = N(0, \Sigma_w) \propto \exp\left(-\frac{1}{2} w^T \Sigma_w^{-1} w\right)$

Estimates conditional distribution on  $w$  given the data using Bayes' rule.

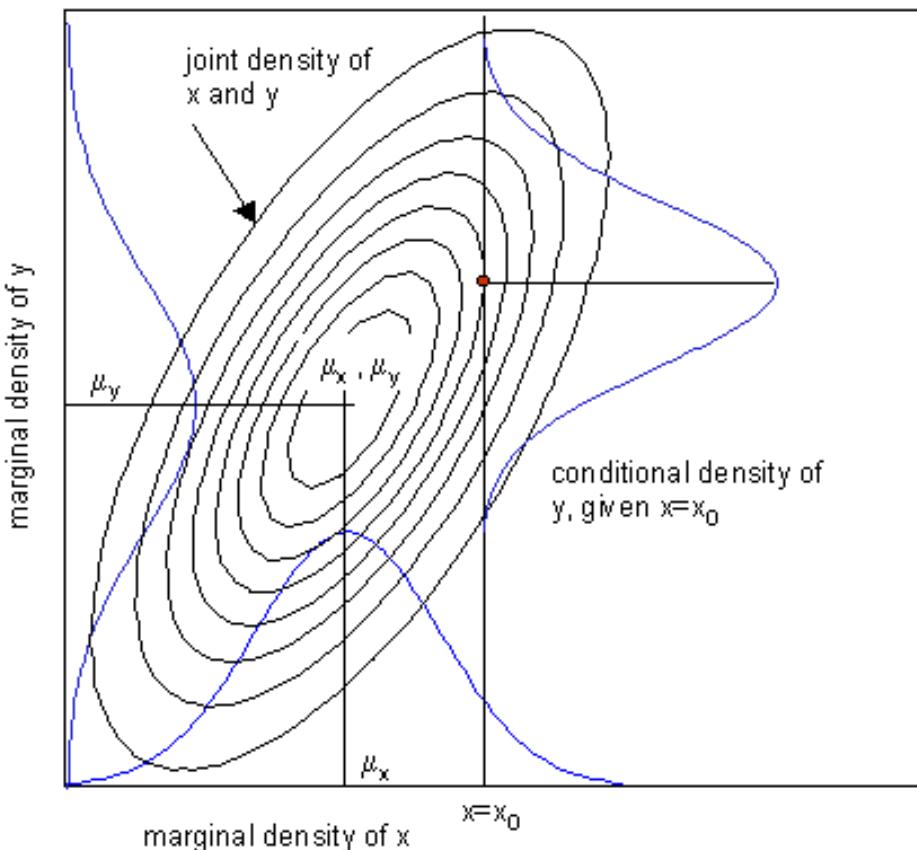
$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}} \quad (\text{drop } \sigma, \text{ not a variable})$$

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

Posterior distribution on  $w$   
is Gaussian.

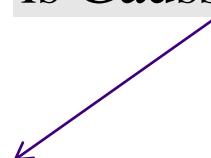


$$\Rightarrow p(w | \mathbf{X}, \mathbf{y}) \propto N\left(\frac{1}{\sigma^2} \left( \frac{1}{\sigma^2} \mathbf{X} \mathbf{X}^T + \Sigma_w^{-1} \right)^{-1} \mathbf{X} \mathbf{y}, \left( \frac{1}{\sigma^2} \mathbf{X} \mathbf{X}^T + \Sigma_w^{-1} \right)^{-1}\right)$$



The conditional distribution of a Gaussian distribution is also Gaussian (image from Wikipedia)

Posterior distribution on  $w$   
is Gaussian.



$$\Rightarrow p(w | X, \mathbf{y}) \propto N\left( \frac{1}{\sigma^2} \left( \frac{1}{\sigma^2} \mathbf{X}\mathbf{X}^T + \Sigma_w^{-1} \right)^{-1} \mathbf{X}\mathbf{y}, \left( \frac{1}{\sigma^2} \mathbf{X}\mathbf{X}^T + \Sigma_w^{-1} \right)^{-1} \right)$$

# Probabilistic Regression

The expectation over the posterior distribution gives the best estimate:

$$E\{ p(w | X, \mathbf{y}) \} = \frac{1}{\sigma^2} \underbrace{\left( \frac{1}{\sigma^2} \mathbf{X}\mathbf{X}^T + \boldsymbol{\Sigma}_w^{-1} \right)^{-1}}_A \mathbf{X}\mathbf{y}.$$

This is called the *maximum a posteriori (MAP)* estimate of  $w$ .

$$\Rightarrow p(w | X, \mathbf{y}) \propto N\left( \frac{1}{\sigma^2} \left( \frac{1}{\sigma^2} \mathbf{X}\mathbf{X}^T + \boldsymbol{\Sigma}_w^{-1} \right)^{-1} \mathbf{X}\mathbf{y}, \left( \frac{1}{\sigma^2} \mathbf{X}\mathbf{X}^T + \boldsymbol{\Sigma}_w^{-1} \right)^{-1} \right)$$

# Probabilistic Regression

$$p(y | x, X, \mathbf{y}) = N\left(\frac{1}{\sigma^2} x^T A^{-1} X \mathbf{y}, x^T A^{-1} x\right)$$

with  $A = \frac{1}{\sigma^2} X X^T + \Sigma_w^{-1}$

We can now compute the posterior distribution on  $y$ :

$$p(y | x, X, \mathbf{y}) = \int p(y | x, w) p(w | X, \mathbf{y}) dw$$

$$\Rightarrow p(w | X, \mathbf{y}) \propto N\left(\frac{1}{\sigma^2} \left(\frac{1}{\sigma^2} X X^T + \Sigma_w^{-1}\right)^{-1} X \mathbf{y}, \left(\frac{1}{\sigma^2} X X^T + \Sigma_w^{-1}\right)^{-1}\right)$$

# Probabilistic Regression

$$p(y | \mathbf{x}, \mathbf{X}, \mathbf{y}) = N\left(\frac{1}{\sigma^2} \mathbf{x}^T A^{-1} \mathbf{Xy}, \mathbf{x}^T A^{-1} \mathbf{x}\right)$$

Testing point

Training datapoints

$$\text{with } A = \frac{1}{\sigma^2} \mathbf{XX}^T + \Sigma_w^{-1}$$

The estimate of  $y$  given a test point  $x$  is given by :

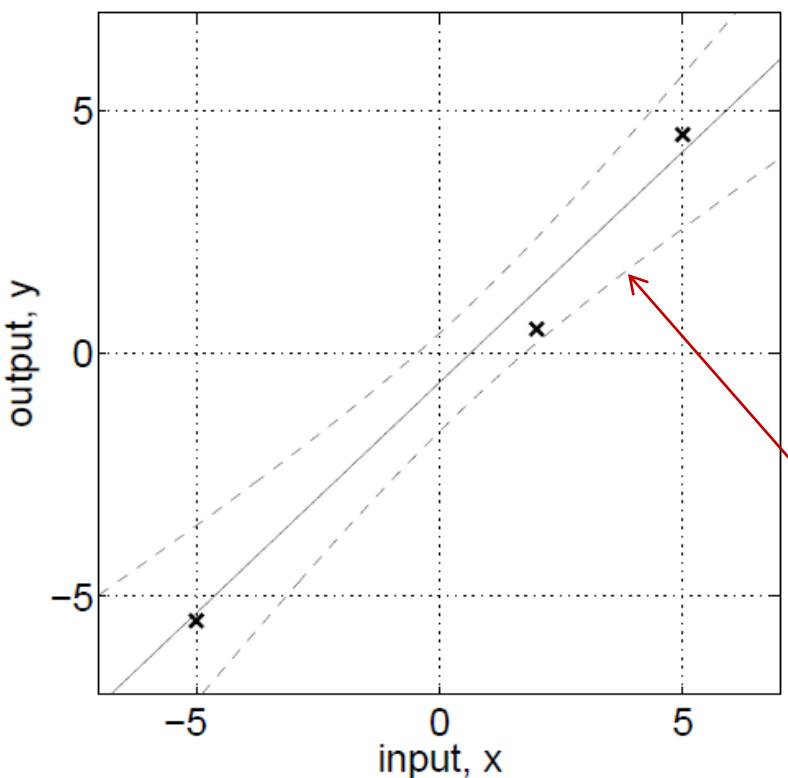
$$y = E\{p(y | \mathbf{x})\} = \frac{1}{\sigma^2} \mathbf{x}^T A^{-1} \mathbf{Xy}$$

# Probabilistic Regression

$$p(y | \mathbf{x}, \mathbf{X}, \mathbf{y}) = N\left(\frac{1}{\sigma^2} \mathbf{x}^T A^{-1} \mathbf{X} \mathbf{y}, \mathbf{x}^T A^{-1} \mathbf{x}\right)$$

$$E\{p(y | x, X, y)\} = \frac{1}{\sigma^2} \mathbf{x}^T A^{-1} \mathbf{X} \mathbf{y}$$

with  $A = \frac{1}{\sigma^2} \mathbf{X} \mathbf{X}^T + \Sigma_w^{-1}$



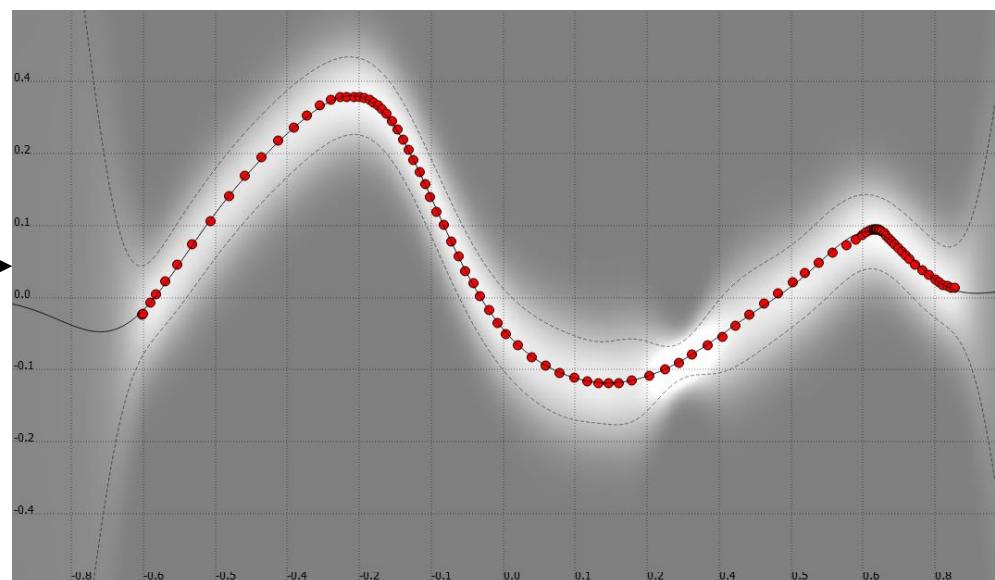
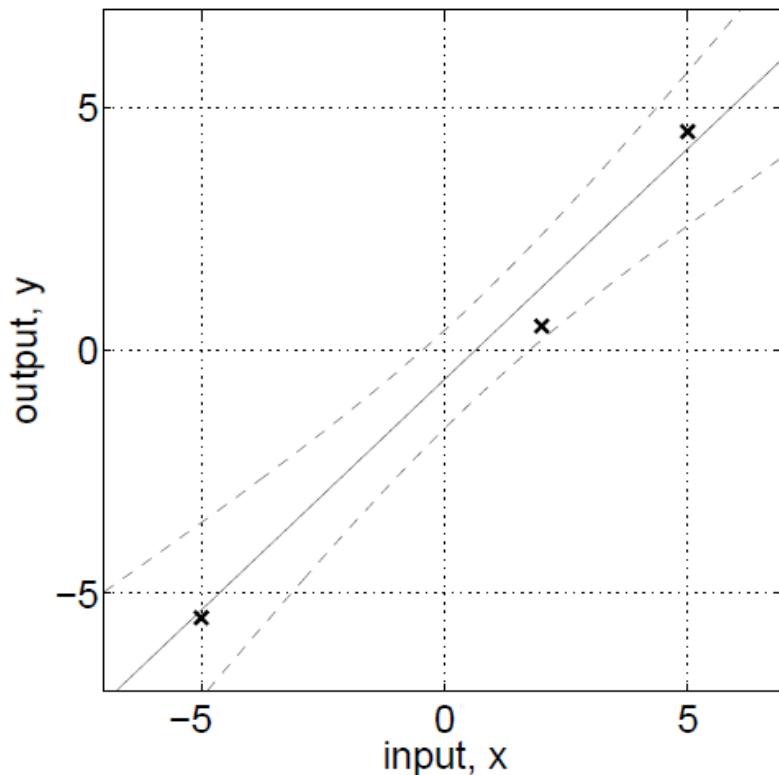
The variance gives a measure of the uncertainty of the prediction:

$$\text{var}\{p(y | x)\} = \mathbf{x}^T A^{-1} \mathbf{x}$$

# Gaussian Process Regression

How to extend the simple linear Bayesian regressive model for nonlinear regression ?

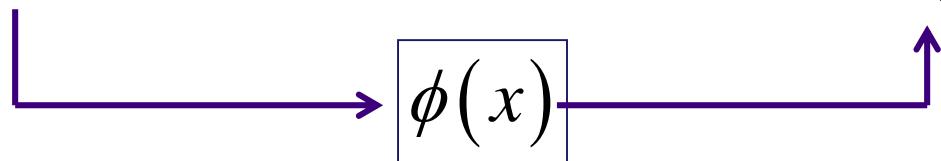
$$y = w^T x + N(0, \sigma^2)$$



# Gaussian Process Regression

How to extend the simple linear Bayesian regressive model for nonlinear regression ?

$$y = w^T x + N(0, \sigma^2)$$



$$y = w^T \phi(x) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

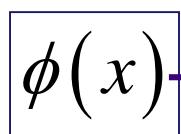
# Gaussian Process Regression

How to extend the simple linear Bayesian regressive model for nonlinear regression ?

$$y = w^T x + N(0, \sigma^2)$$



$$y = w^T \phi(x) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$



Distribution over functions

# Gaussian Process Regression

How to extend the simple linear Bayesian regressive model for nonlinear regression ?

$$y = w^T x + N(0, \sigma^2)$$



$$p(y | x, X, \mathbf{y}) = N\left(\frac{1}{\sigma^2} x^T A^{-1} X \mathbf{y}, x^T A^{-1} x\right), \quad A = \frac{1}{\sigma^2} X X^T + \Sigma_w^{-1}$$


$$\phi(x)$$

Non-Linear Transformation

$$p(y | x, X, \mathbf{y}) = N\left(\frac{1}{\sigma^2} \phi(x)^T A^{-1} \Phi(X) \mathbf{y}, \phi(x)^T A^{-1} \phi(x)\right)$$

$$\text{with } A = \sigma^{-2} \Phi(X) \Phi(X)^T + \Sigma_w^{-1}$$

# Gaussian Process Regression

Again, a Gaussian distribution.

$$p(y | x, X, \mathbf{y}) = N\left(\frac{1}{\sigma^2} \boldsymbol{\phi}(x)^T A^{-1} \Phi(X) \mathbf{y}, \boldsymbol{\phi}(x)^T A^{-1} \boldsymbol{\phi}(x)\right)$$

$$\text{with } A = \sigma^{-2} \Phi(X) \Phi(X)^T + \Sigma_w^{-1}$$

# Gaussian Process Regression

$$p(y | x, X, \mathbf{y}) = N\left( \frac{1}{\sigma^2} \boldsymbol{\phi}(x)^T \Sigma_w \Phi(X) \left[ K(X, X) + \sigma^2 I \right]^{-1} \mathbf{y}, \right.$$

$$\left. \boldsymbol{\phi}(x)^T \Sigma_w \boldsymbol{\phi}(x) - \boldsymbol{\phi}(x)^T \Sigma_w \Phi(X) \left[ K(X, X) + \sigma^2 I \right]^{-1} \Phi(X)^T \Sigma_w \boldsymbol{\phi}(x) \right)$$

See supplement  
for steps

Define the kernel as:  $k(x, x') = \boldsymbol{\phi}(x)^T \Sigma_w \boldsymbol{\phi}(x')$

## Inner product in feature space

$$p(y | x, X, \mathbf{y}) = N\left( \frac{1}{\sigma^2} \boldsymbol{\phi}(x)^T A^{-1} \Phi(X) \mathbf{y}, \boldsymbol{\phi}(x)^T A^{-1} \boldsymbol{\phi}(x) \right)$$

$$\text{with } A = \sigma^{-2} \Phi(X) \Phi(X)^T + \Sigma_w^{-1}$$

# Gaussian Process Regression

$$y = E\{y | x, X, \mathbf{y}\} = \sum_{i=1}^M \alpha_i k(x, x^i)$$

$$\text{with } \boldsymbol{\alpha} = [K(X, X) + \sigma^2 I]^{-1} \mathbf{y}$$

Define the kernel as:  $k(x, x') = \phi(x)^T \Sigma_w \phi(x')$

See supplement  
for steps

## Inner product in feature space

$$p(y | x, X, \mathbf{y}) = N\left(\frac{1}{\sigma^2} \phi(x)^T A^{-1} \Phi(X) \mathbf{y}, \phi(x)^T A^{-1} \phi(x)\right)$$

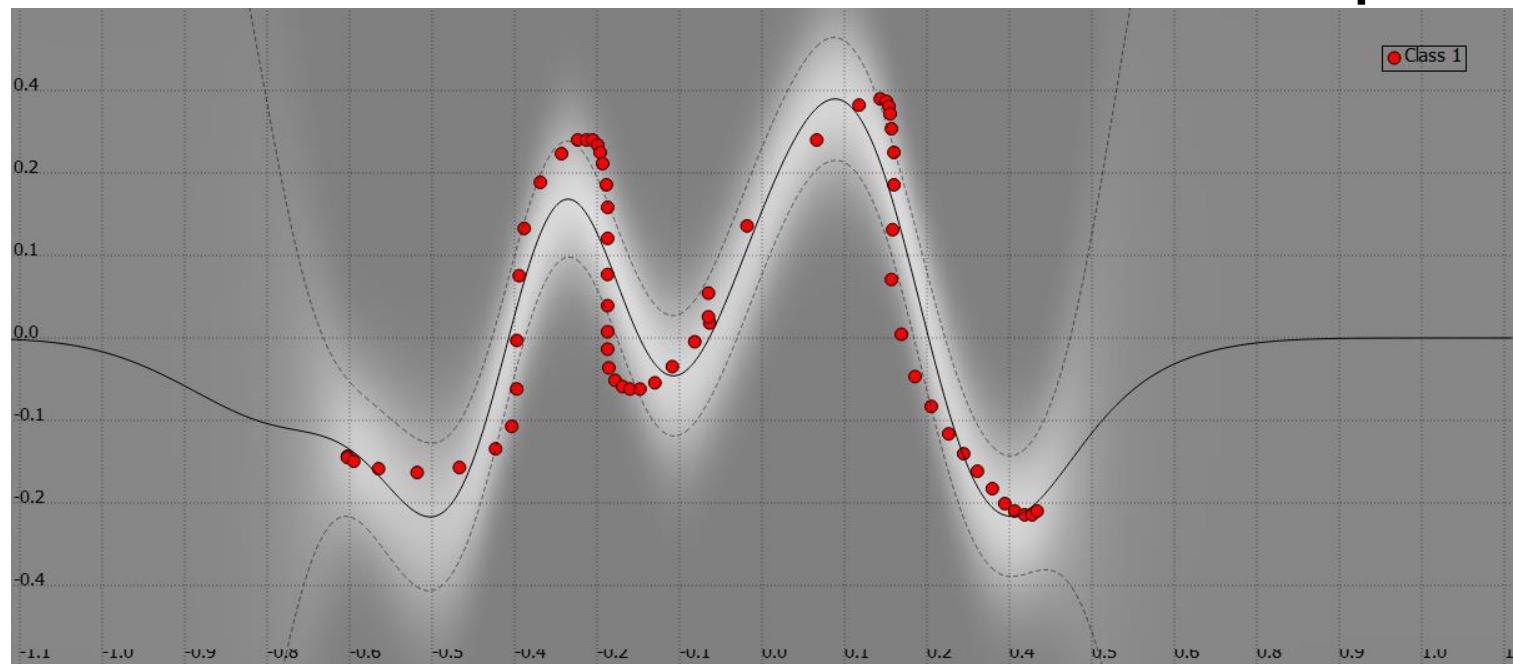
$$\text{with } A = \sigma^{-2} \Phi(X) \Phi(X)^T + \Sigma_w^{-1}$$

# Gaussian Process Regression

$$y = E\{y | x, X, \mathbf{y}\} = \sum_{i=1}^M \alpha_i k(x, x^i)$$

$$\text{with } \boldsymbol{\alpha} = [K(X, X) + \sigma^2 I]^{-1} \mathbf{y}$$

$\alpha_i > 0$   
→ All datapoints are used in the computation!

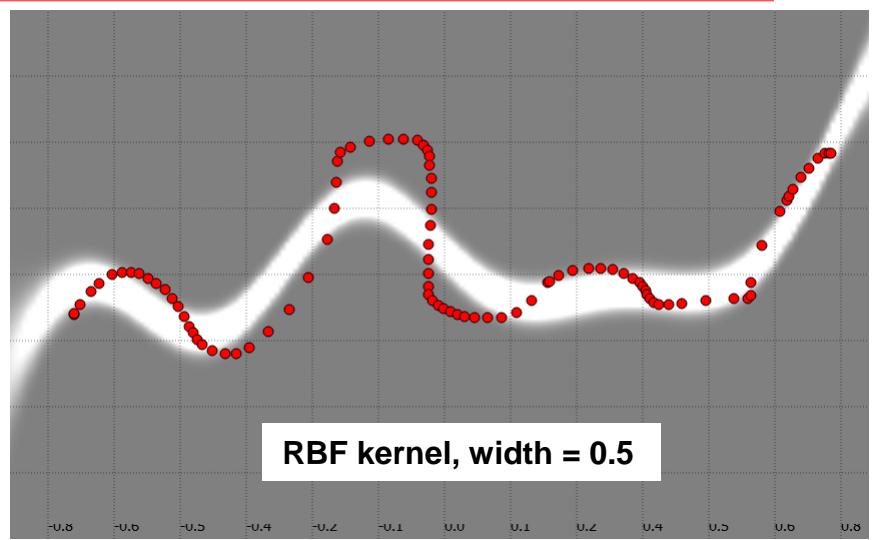
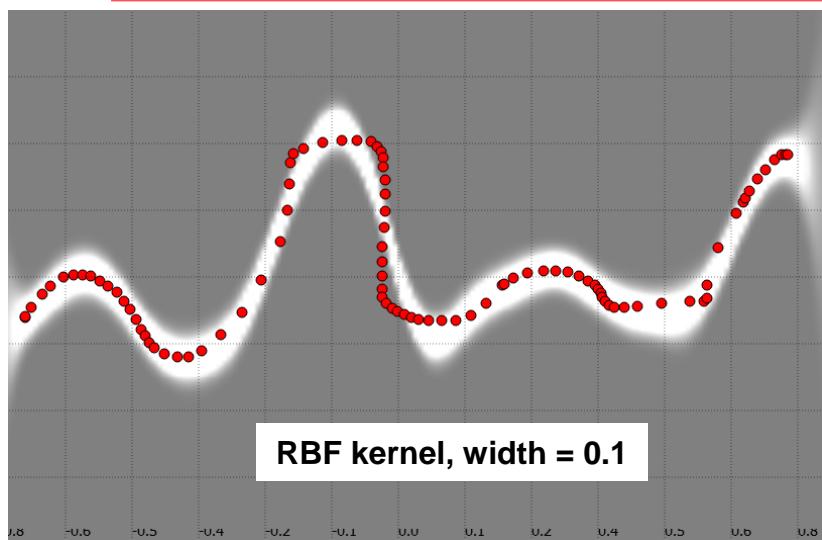


# Gaussian Process Regression

$$y = E\{y | x, X, \mathbf{y}\} = \sum_{i=1}^M \alpha_i k(x, x^i)$$

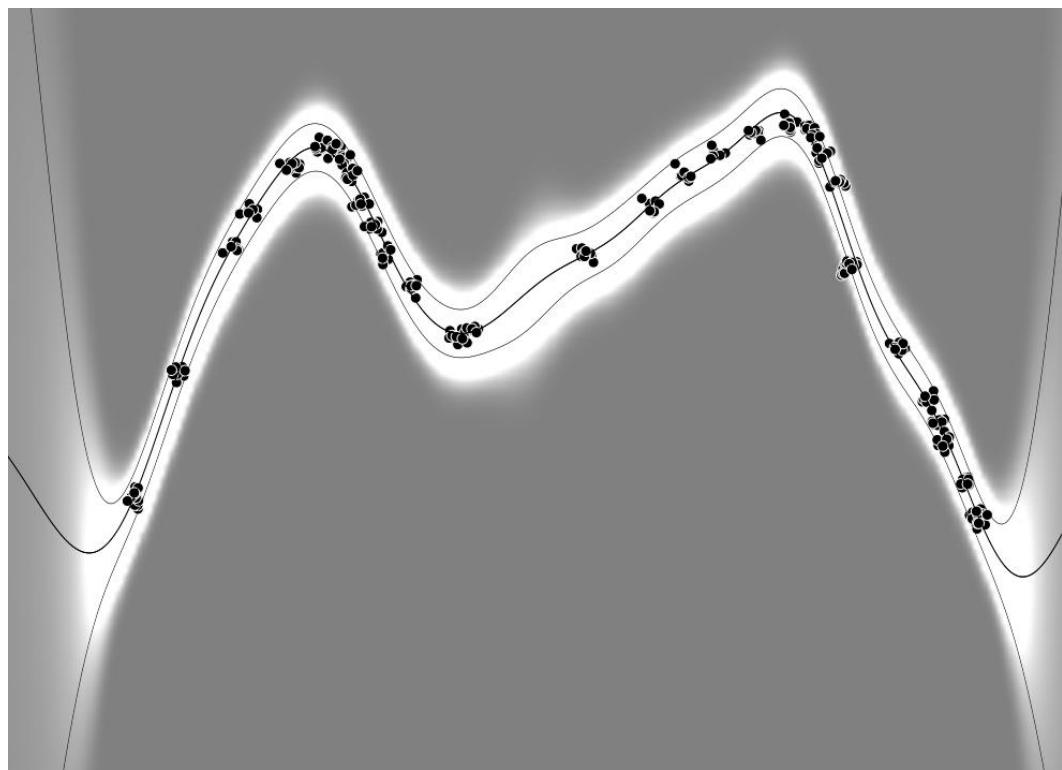
$$\text{with } \alpha = [K(X, X) + \sigma^2 I]^{-1} \mathbf{y}$$

The kernel and its hyperparameters are given by the user.  
These can be optimized through maximum likelihood over the marginal likelihood, see class's supplement



# Gaussian Process Regression

Sensitivity to the choice of *kernel width* (called *lengthscale* in most books) when using Gaussian kernels (also called RBF or square exponential).

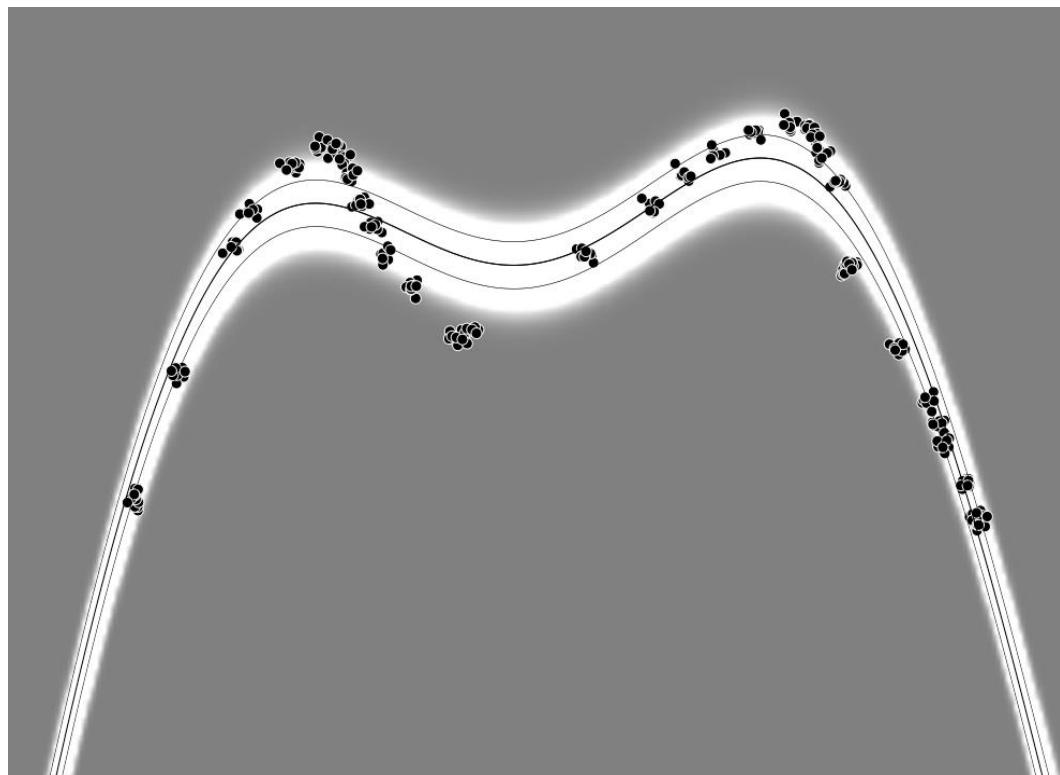


$$k(x, x') = e^{-\frac{\|x-x'\|}{l}}$$

Kernel Width=0.1

# Gaussian Process Regression

Sensitivity to the choice of *kernel width* (called *lengthscale* in most books) when using Gaussian kernels (also called RBF or square exponential).



$$k(x, x') = e^{-\frac{\|x-x'\|}{l}}$$

Kernel Width=0.5

A red arrow points from the variable  $l$  in the equation to the text "Kernel Width=0.5", indicating that the value of  $l$  corresponds to a kernel width of 0.5.

# Gaussian Process Regression

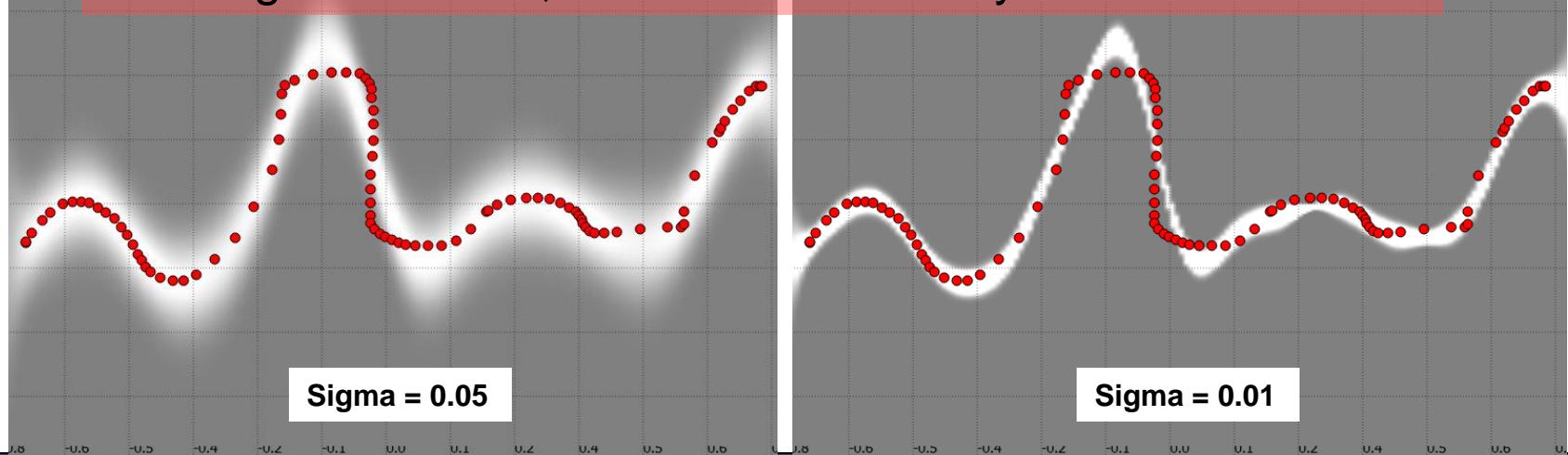
$$y = E\{y | x, X, \mathbf{y}\} = \sum_{i=1}^M \alpha_i k(x, x^i)$$

$$\text{with } \alpha = [K(X, X) + \sigma^2 I]^{-1} \mathbf{y}$$

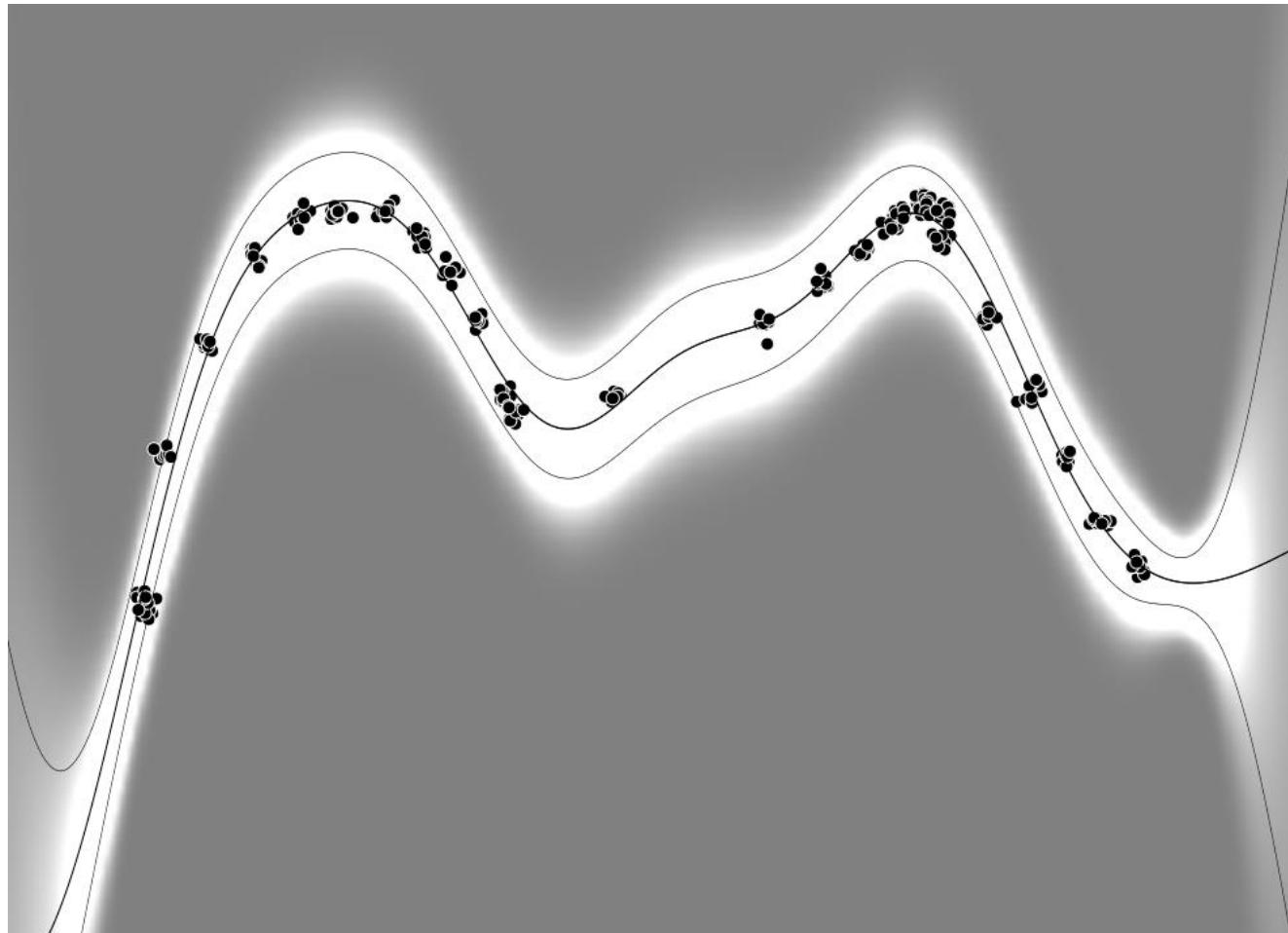
The value for the noise needs to be pre-set by hand.

$$\text{cov}(p(y | x)) = K(x, x) - K(x, X) [K(X, X) + \sigma^2 I]^{-1} K(X, x)$$

The larger the noise, the more uncertainty. The noise is  $\leq 1$ .

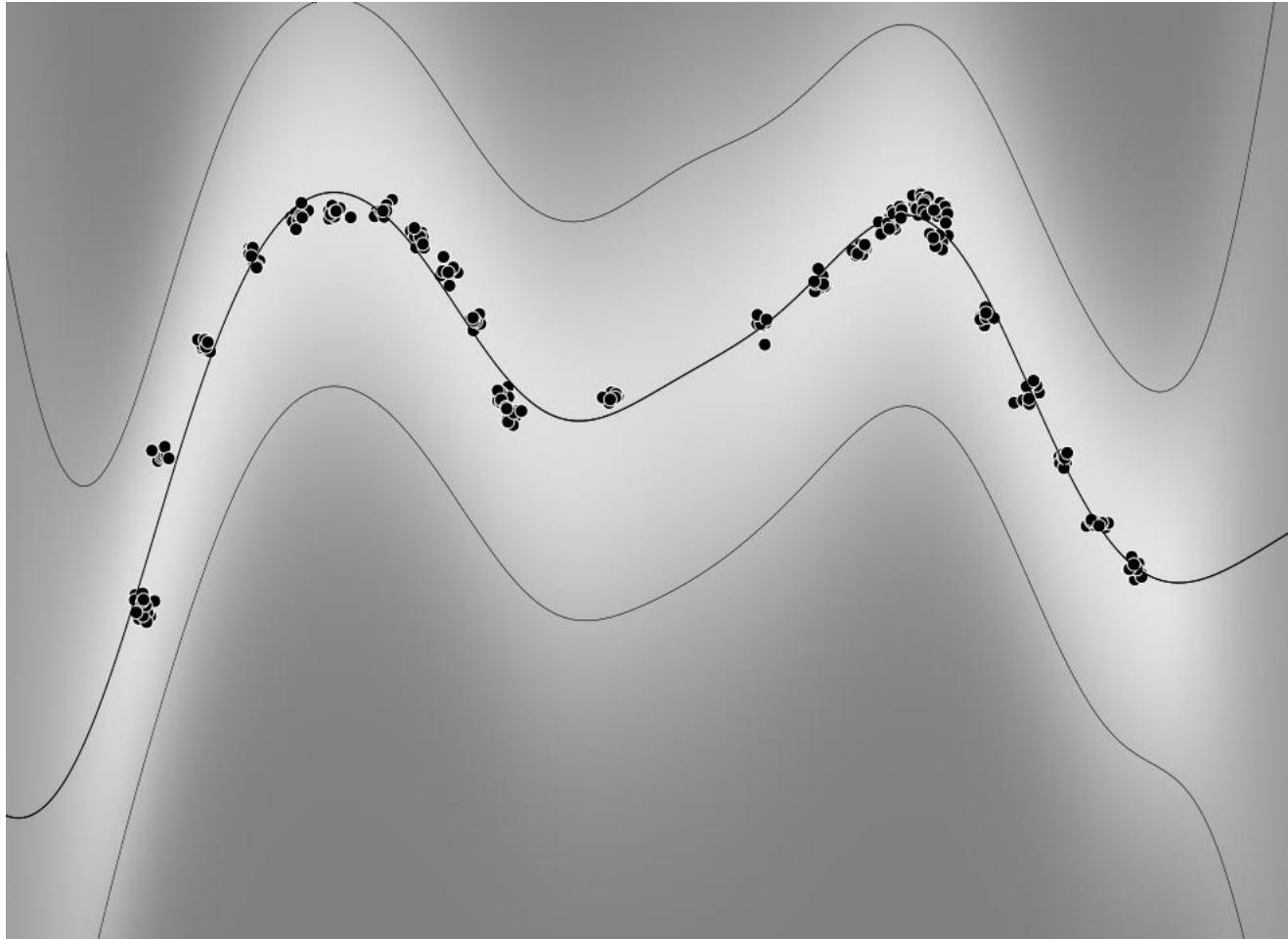


# Gaussian Process Regression



Low noise:  $\sigma=0.05$

# Gaussian Process Regression



High noise:  $\sigma=0.2$

# Gaussian Process Regression

$$y = E\{y | x, X, \mathbf{y}\} = \sum_{i=1}^M \alpha_i k(x, x^i)$$

$$\text{with } \alpha = [K(X, X) + \sigma^2 I]^{-1} \mathbf{y}$$

Kernel is usually Gaussian kernel with *stationary* covariance function  
 → Non-Stationary Covariance Functions can encapsulate local variations in the density of the datapoints

$$k(x, x') = \prod_{i=1}^N \left( \frac{2l_i(x)2l_i(x')}{l_i(x)+l_i(x')} \right)^{\frac{1}{2}} \exp \left( -\sum_{i=1}^N \frac{(x_i - x'_i)^2}{l_i^2(x) + l_i^2(x')} \right)$$

Gibbs' non stationary covariance function (length-scale a function of x):

# Gaussian Process Regression

## Linear Model

$$y = w^T x + N(0, \sigma^2)$$

## Non-Linear Model

$$y = w^T \phi(x) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

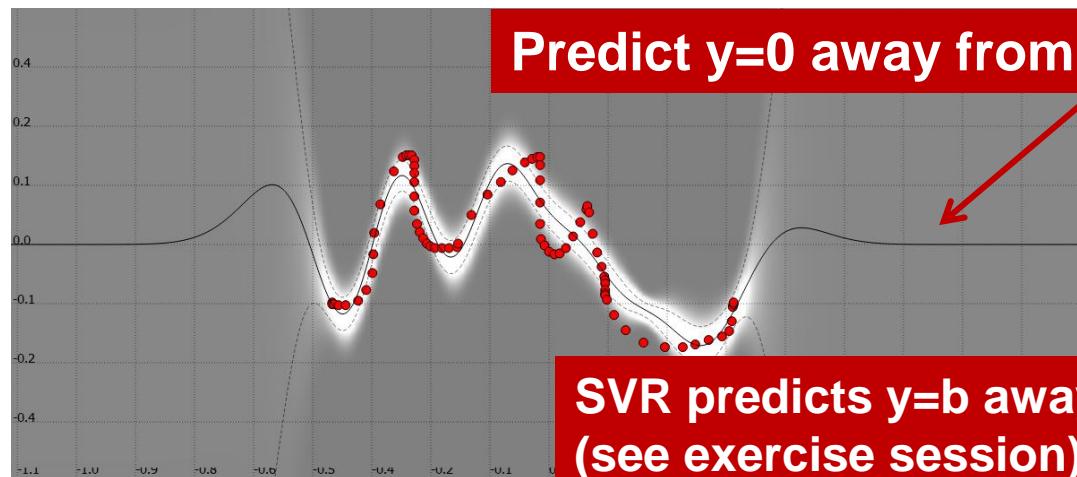
**Both models follow a zero mean Gaussian distribution!**

$$E\{y\} = E\{w^T x + N(0, \sigma^2)\}$$

$$= E\{w^T\} x + 0 = 0$$

$$E\{y\} = E\{w^T \phi(x) + N(0, \sigma^2)\}$$

$$= E\{w^T\} \phi(x) + 0 = 0$$



# Examples of application of GPR



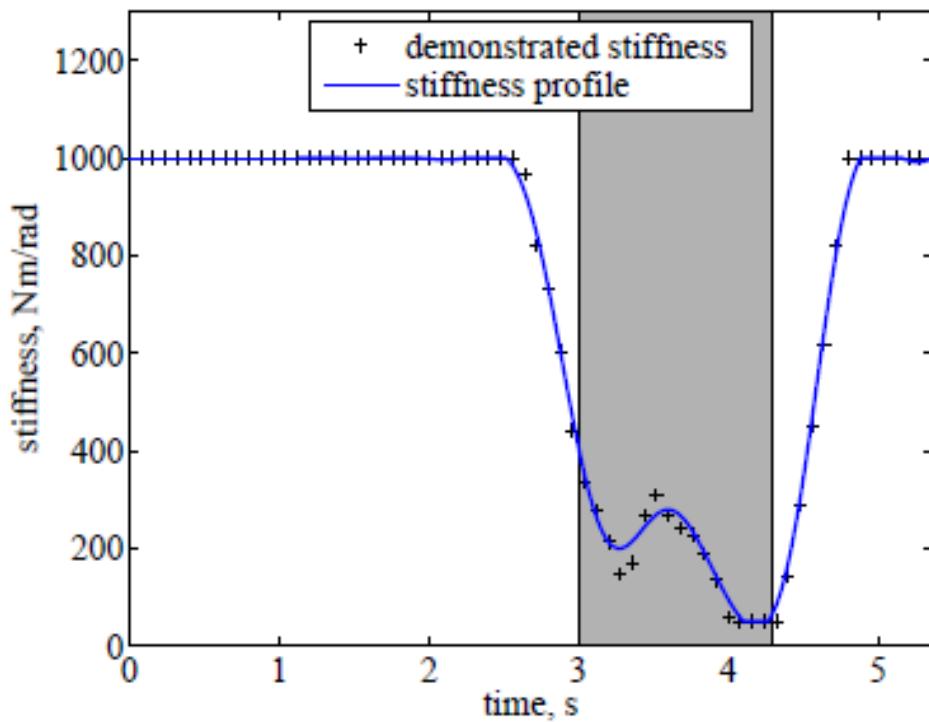
**Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction:  
the match lighting task.**

Klas Kronander and Aude Billard

Striking a match is a task that requires careful control of the force in interaction to push enough to light the match but not too much in order not to break it.

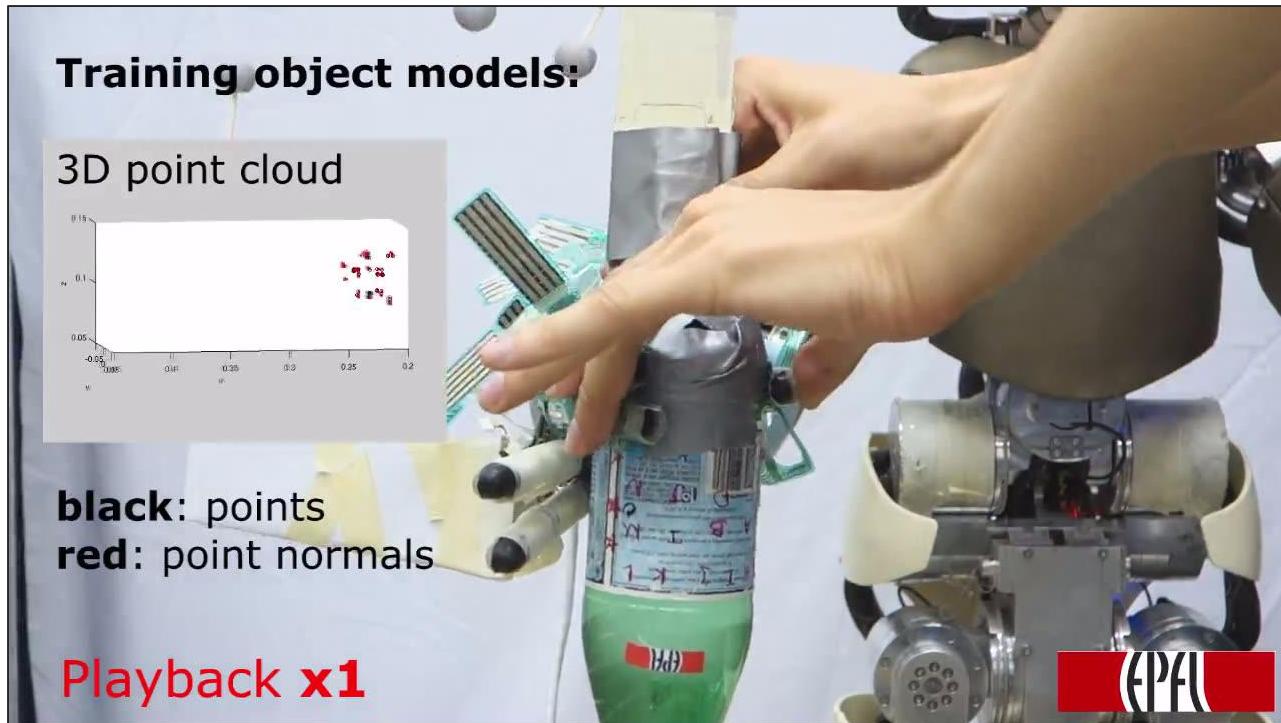
Kronander, K. and Billard, A. (2013) Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction. *IEEE Transactions on Haptics*. *10.1109/TOH.2013.54*.

# Examples of application of GPR



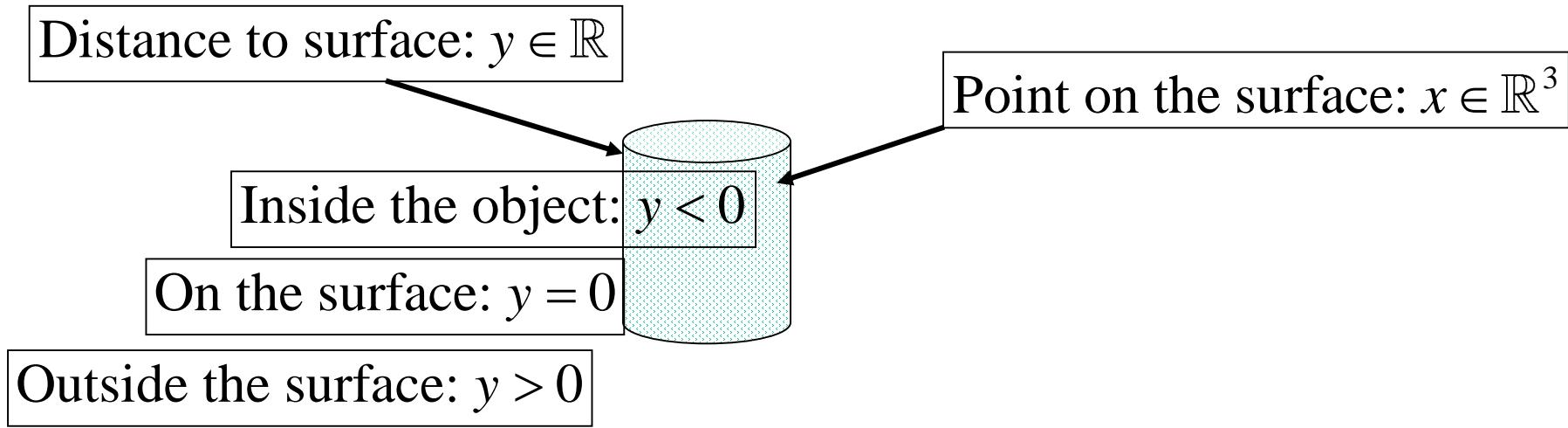
The stiffness profile is encoded as a time-varying input using GPR. The shaded area corresponds to the striking phase. We see that the stiffness must be decreased just before entering into contact and again during contact when the match lights up. Stiffness can increase again when the robot moves back into free space

# Examples of application of GPR



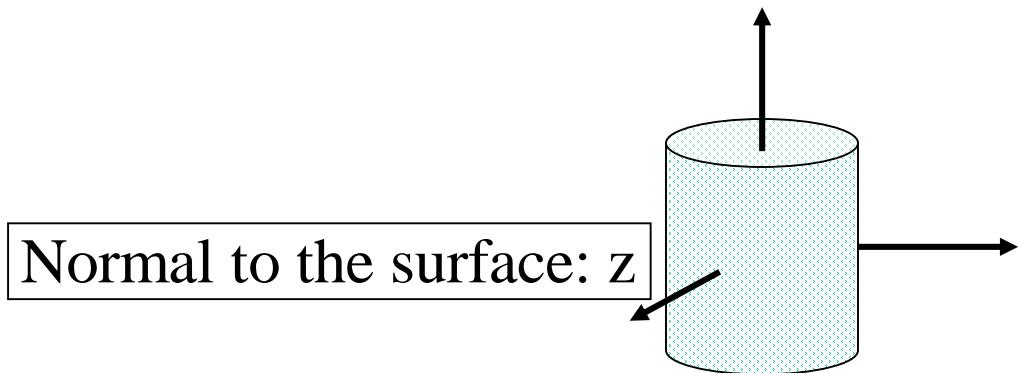
Building a 3D model of an object from tactile information can be useful to guide manipulation of object when the object is no longer visible.

# Examples of application of GPR



Learn a mapping  $y = f(x)$  with GPR  
to determine how far one is from the surface.

# Examples of application of GPR

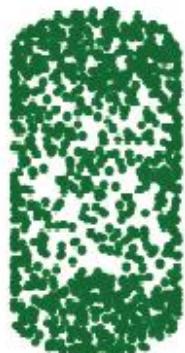


Normal to the surface:  $z$

Learn a mapping  $z = g(x)$  with GPR to determine the normal from the surface (need 3 GPRs for each of the coordinate of the vector  $z$ ).

The distance and normal to the surfaces can be used in an optimization framework to determine the optimal posture of robot fingers on the object.

# Examples of application of GPR



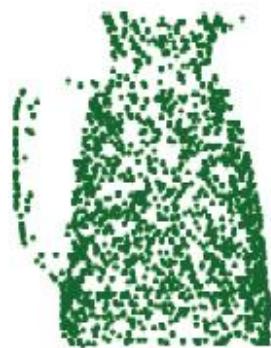
(a) Cylinder



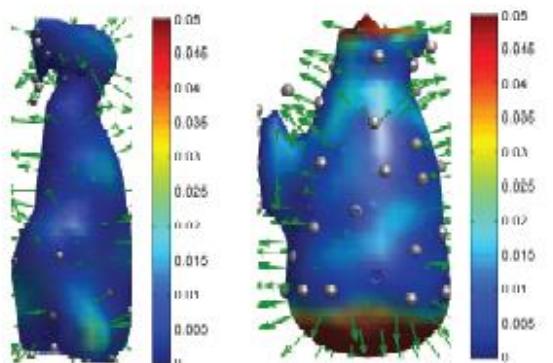
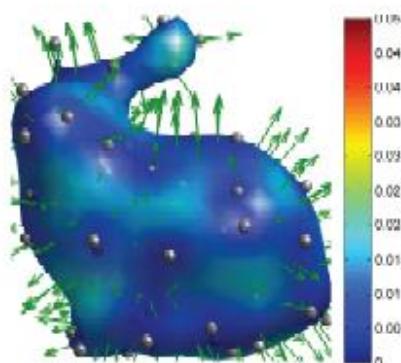
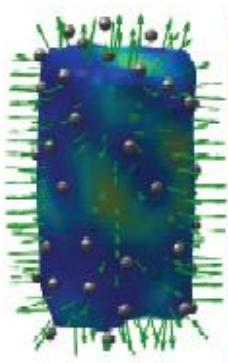
(b) Bunny



(c) Spray

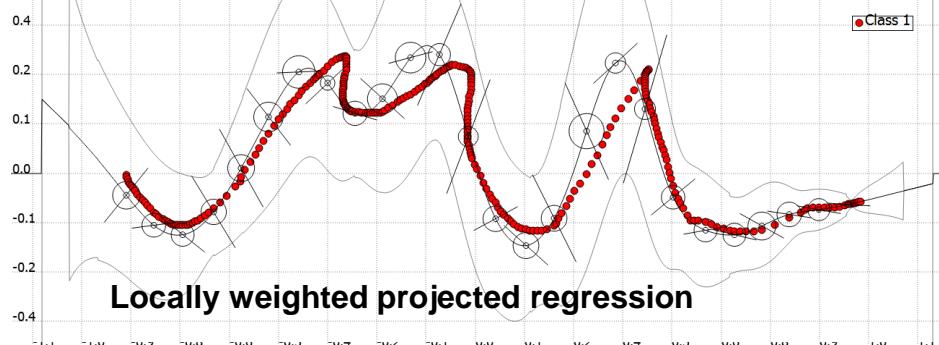
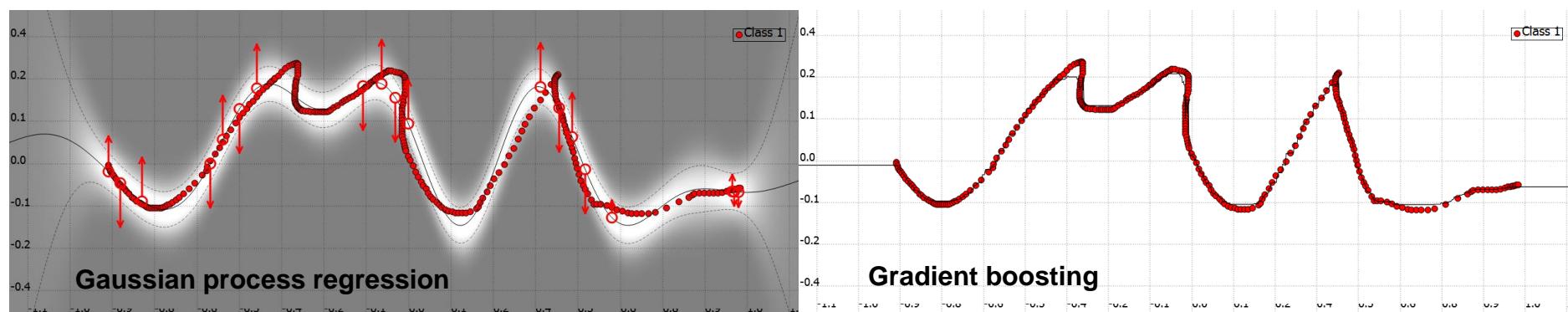
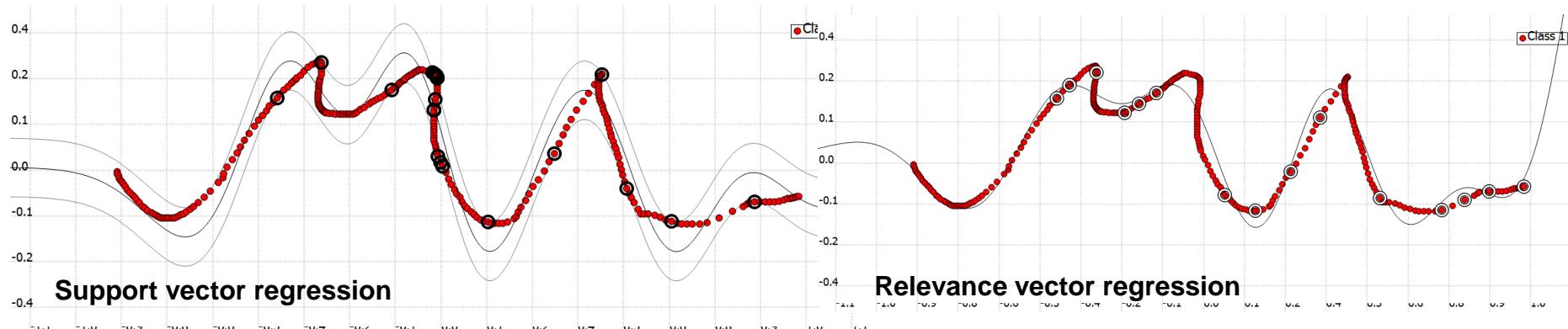


(d) Jug

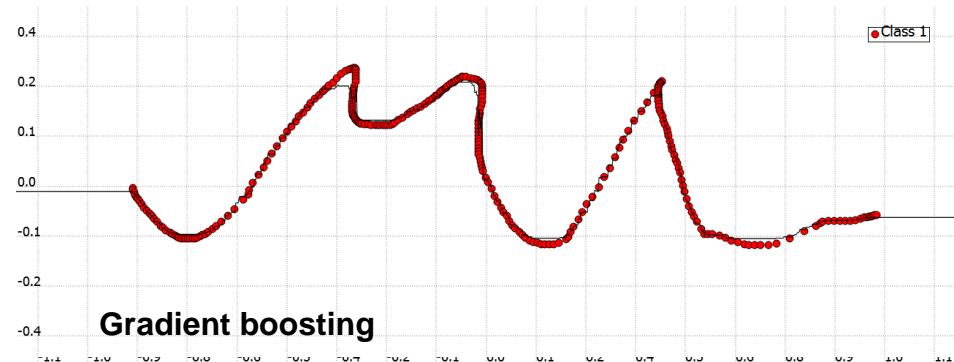


GPR can be used to model the shape of objects. Top: 3D points sampled either from a camera or from tactile sensing. Bottom: 3D shape reconstructed by GPR. The arrows represent the predicted normals at the surface ([El Khoury, S., Li, M., and Billard, A. \(2013\). On the generation of a variety of grasps. Robotics and Autonomous Systems, 61\(12\):1335–34](#))

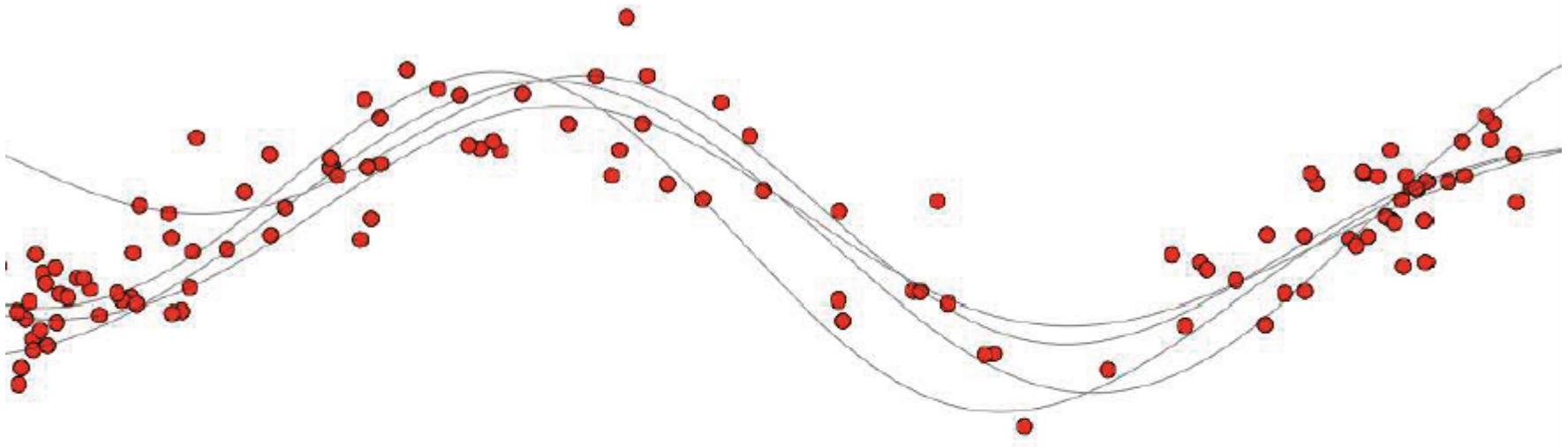
# Regression Algorithms in this Course



# Regression Algorithms in this Course



# Gradient Boosting



- We select some input samples
- We learn a regression model
- Very sensitive to the input selection
- $m$  training sets =  $m$  different models

$$(x_1^{(1)}, y_1^{(1)}), \dots, (x_n^{(1)}, y_n^{(1)}) \rightarrow \hat{f}^{(1)}(x) = Y^{(1)}$$

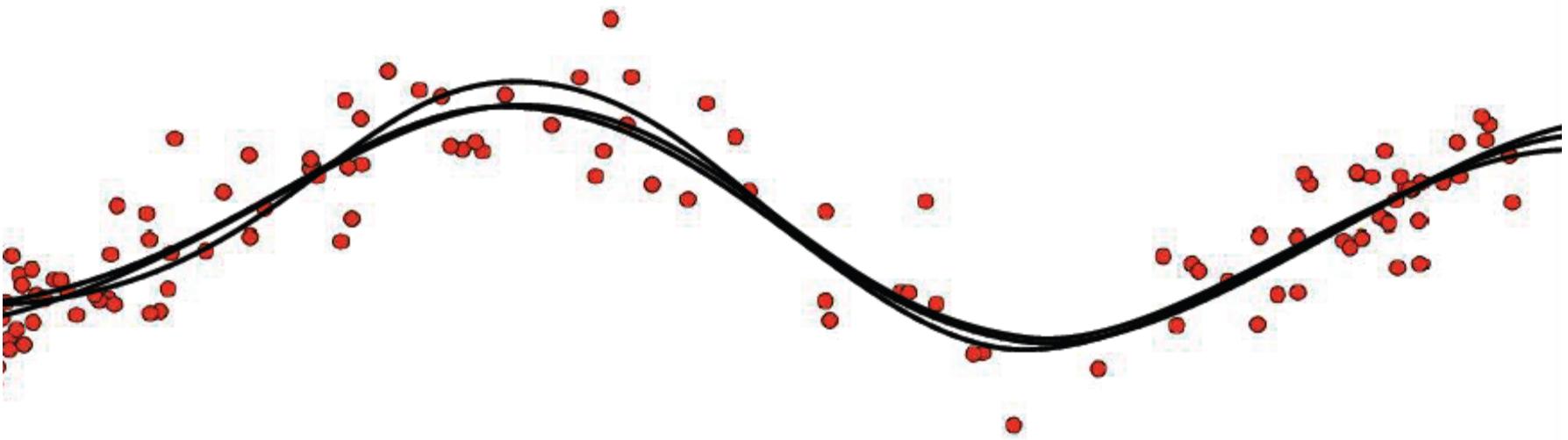
$$(x_1^{(2)}, y_1^{(2)}), \dots, (x_n^{(2)}, y_n^{(2)}) \rightarrow \hat{f}^{(2)}(x) = Y^{(2)}$$

$$\hat{f}^{(1)}, \dots, \hat{f}^{(m)} \rightarrow Y^{(1)}, \dots, Y^{(m)}$$

Choose some regressive technique (any we have seen sofar)

Apply boosting to train and combine the set of estimates  $\hat{f}^1, \hat{f}^2, \dots, \hat{f}^m$ .

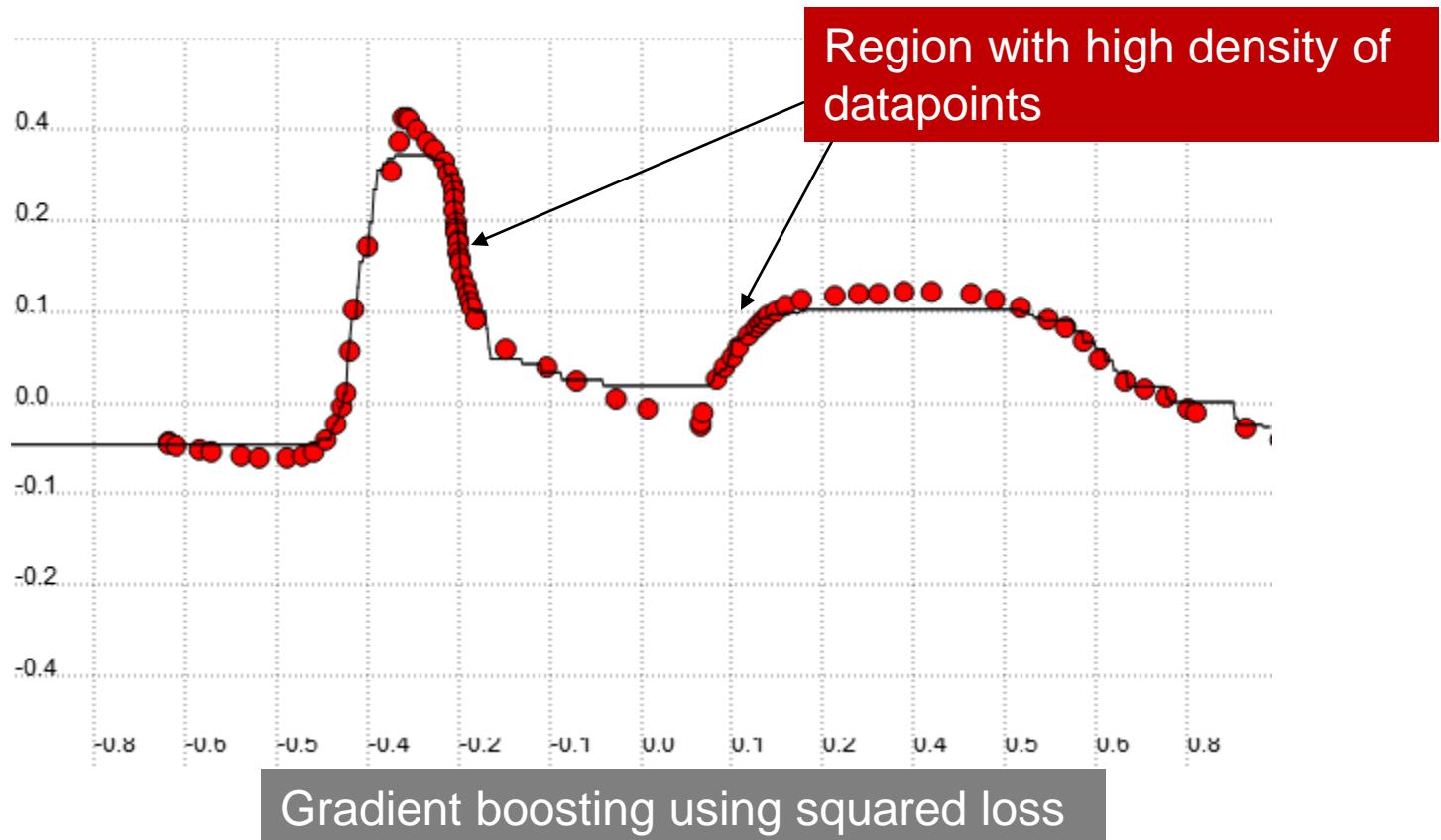
# Gradient Boosting



- Linear combination of simple models
- More examples = Better model
- We can stop when we're satisfied

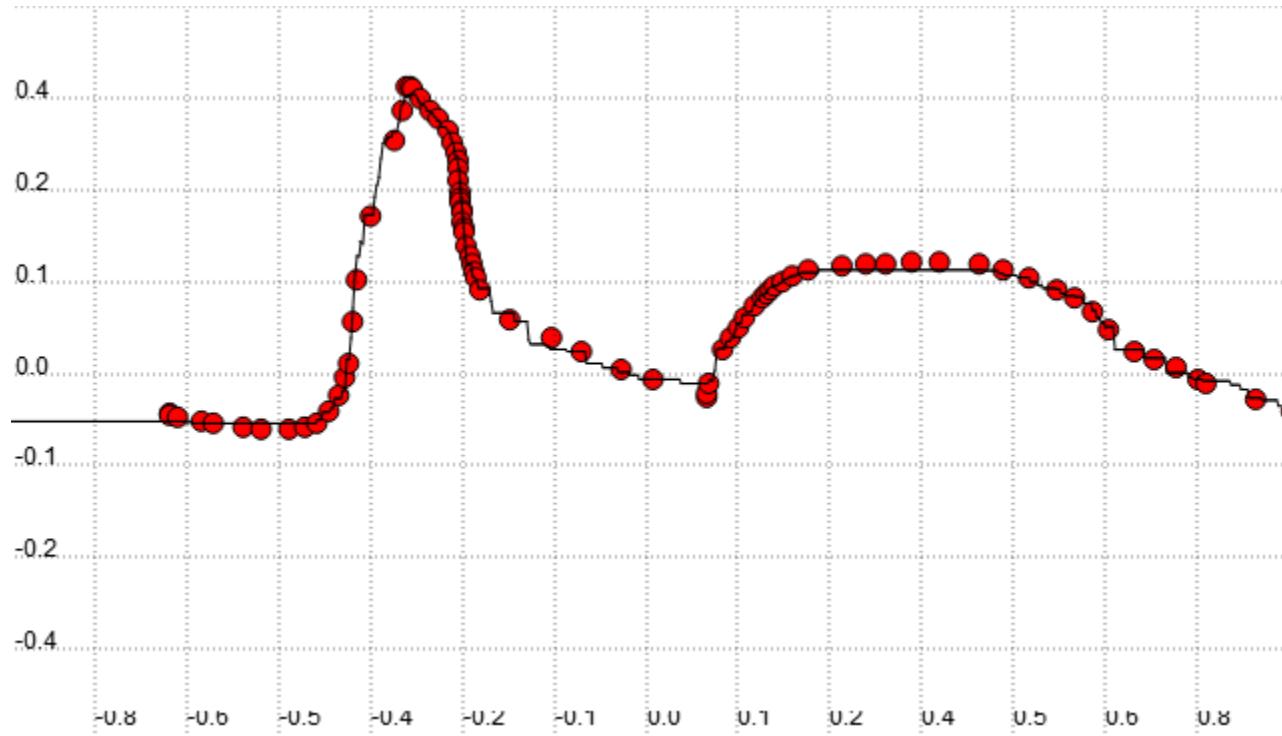
Aggregate to get the final estimate  $\hat{f} = \frac{1}{m} \sum_{i=1}^m \hat{f}^i$

# Gradient Boosting



Typical example of dataset with **imbalanced data**. Data was hand-drawn. There are more datapoints in regions where the hand slowed down. As a result, the fit is very good in these regions and less good in other regions.

# Gradient Boosting



Gradient boosting using squared loss and using twice more functions

Better results, i.e. smoother fit, are obtained when increasing the number of functions for the fit

# Summary

We have seen a few different techniques to perform non-linear regression in machine learning.

The techniques differ in their algorithm and in the number of hyperparameters.

Some techniques (GP, RVR) provide a metric of uncertainty of the model, which can be used to determine when inference is trustable.

Some techniques ( $\nu$ -SVR, RVR, LWPR) are designed to be computationally cheap at retrieval (very few support vectors, few models).

Other techniques (GP) are meant to provide very accurate estimate of the data, at the cost of retaining all datapoints for retrieval.