Advanced Techniques for Mobile Robotics

Gaussian Processes in Robotics

Wolfram Burgard, Cyrill Stachniss,

Kai Arras, Maren Bennewitz

Overview

- Regression problem
- Gaussian process models
- Learning GPs
- Applications
- Summary

The Regression Problem

- Given n observed points

$$\mathcal{X} = \{(x_1, t_1), \ldots, (x_n, t_n)\}, \quad x_i \in \mathbb{R}^p,\ t_i \in \mathbb{R}$$

- Assuming the dependency

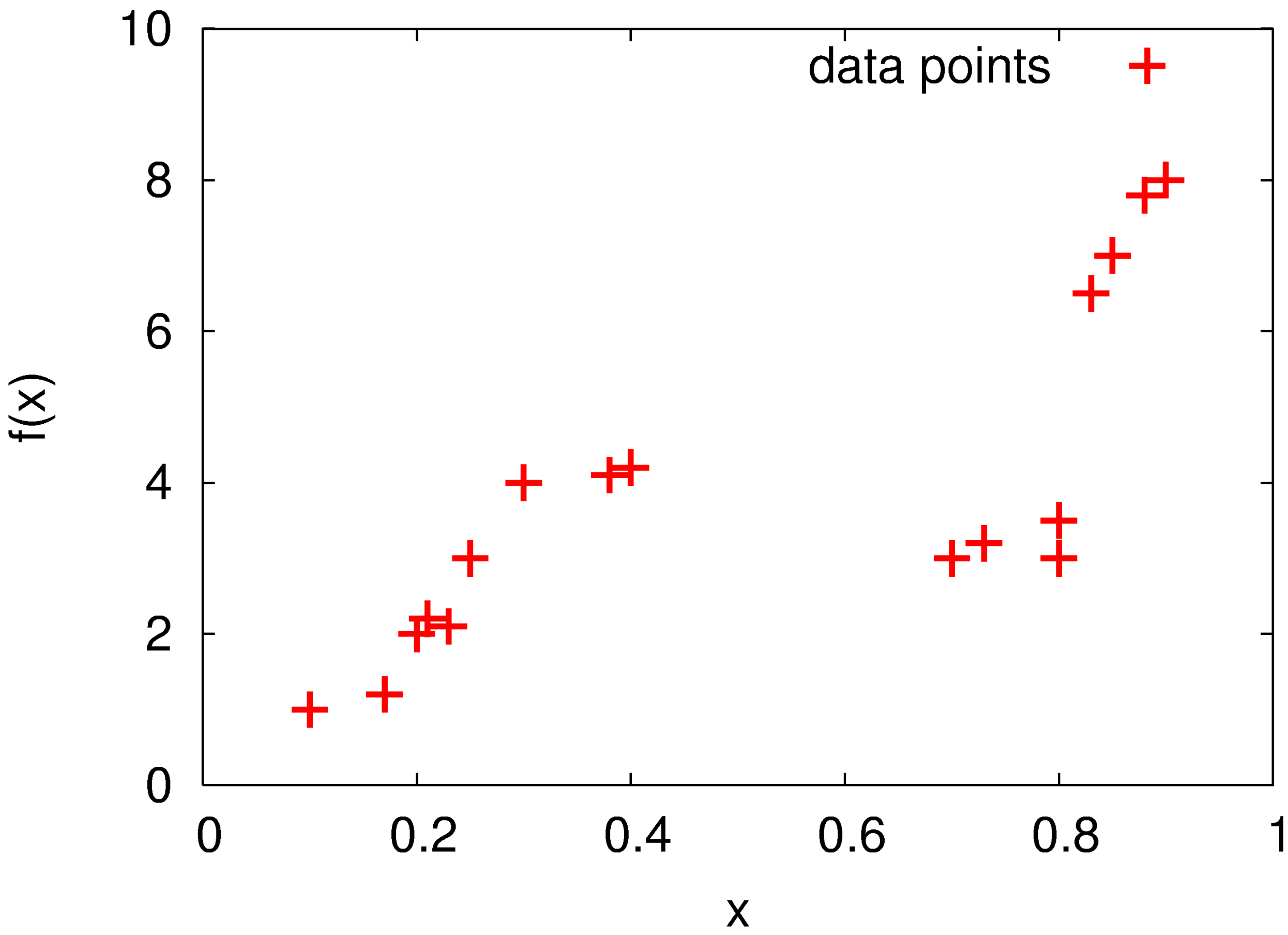
$$t_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2),\ i.i.d$$

- How to predict new points

$$p(t_{n+1} \mid x_{n+1}, \mathcal{X})$$

The Regression Problem

- Given n observed points

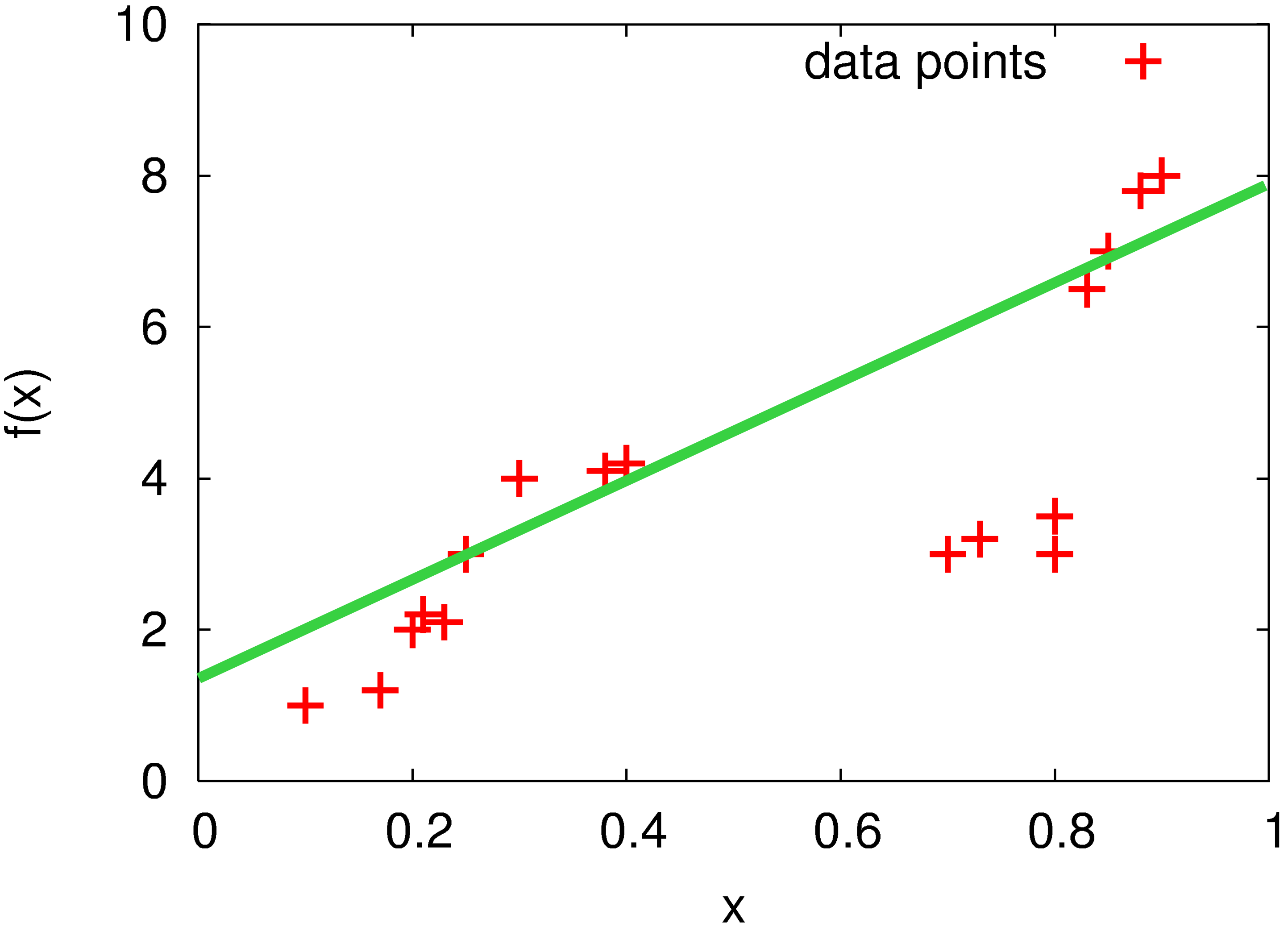


The Regression Problem

- Solution 1: Parametric models
  - Linear $f(x_i) = c_0 + c_1 x_i + \epsilon_i$
  - Quadratic $f(x_i) = c_0 + c_1 x_i + c_2 x_i^2 + \epsilon_i$
  - Higher order polynomials
  - …
- Learning: optimizing the parameters

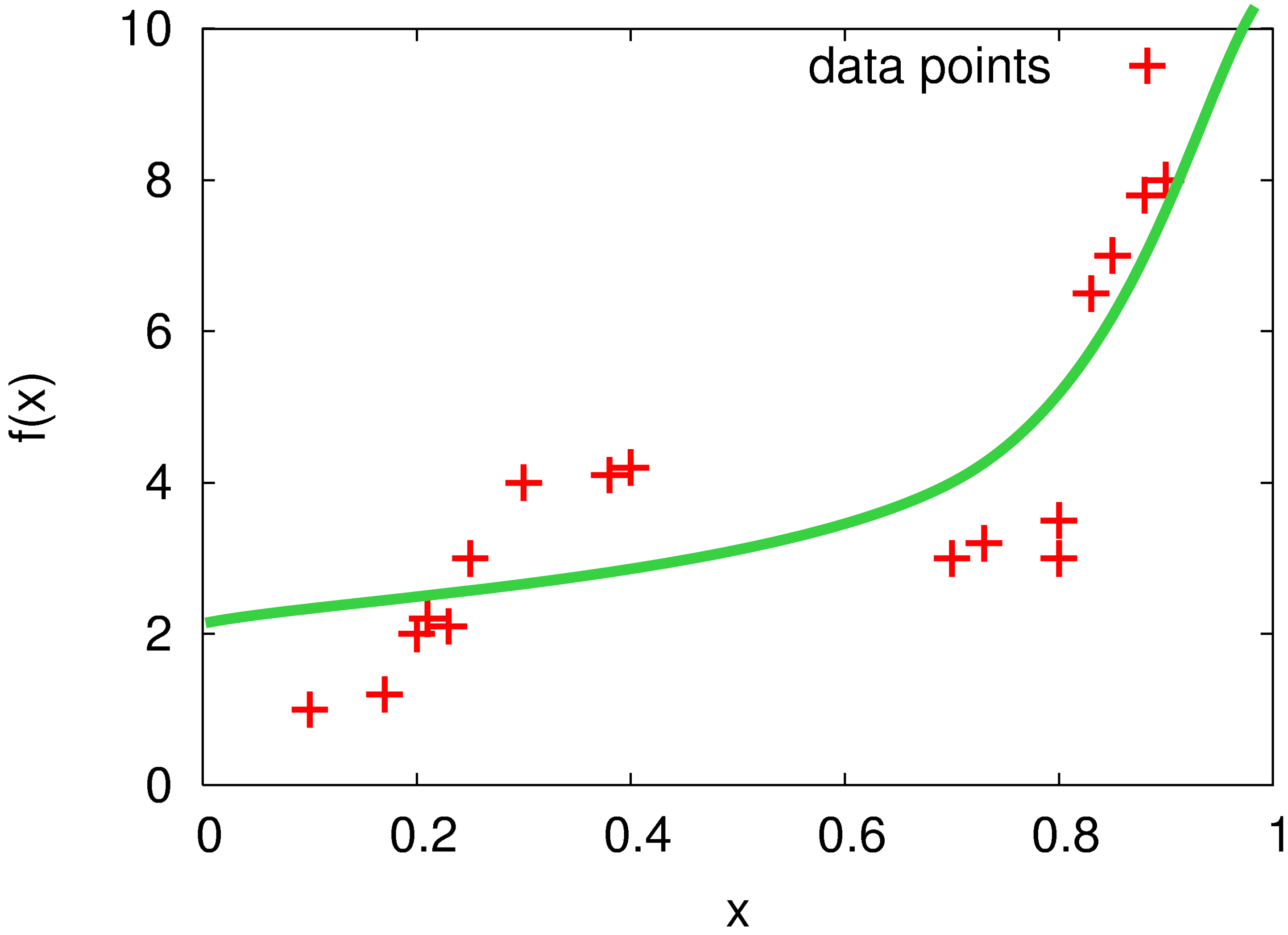
The Regression Problem

- Solution 1: Parametric models



The Regression Problem

- Solution 1: Parametric models



The Regression Problem

- Solution 2: Non-parametric models
  - Radial Basis functions

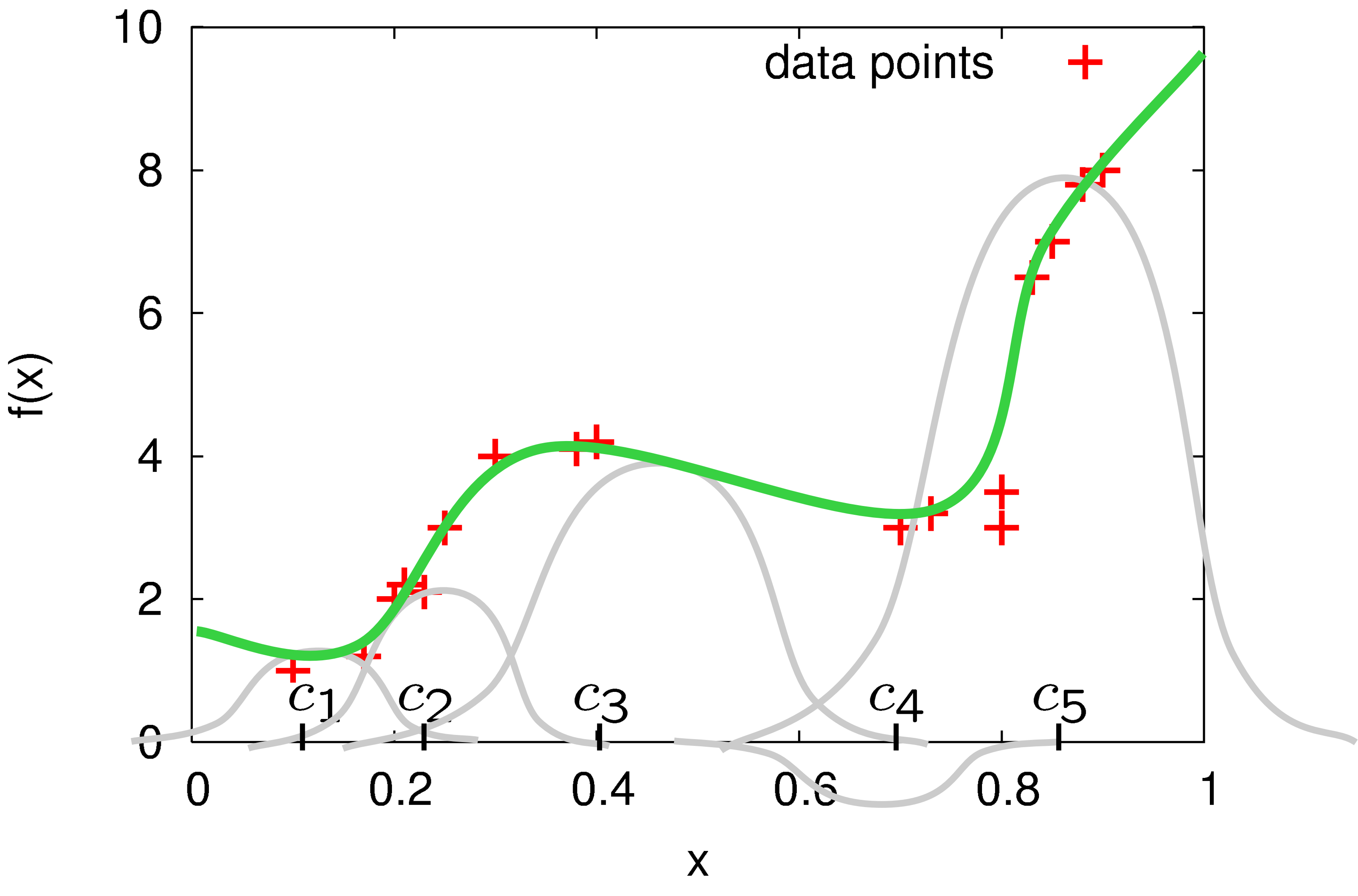
$$f(x_i) = \sum_j w_j \; k(\| x_i - c_j \|)$$

$$k(\| x - c \|) \propto e^{-\beta \|x-c\|^2}$$

  - Histograms, Splines, Support Vector Machines …
- Learning: finding the structure of the model and optimize its parameters

The Regression Problem

- Given n observed points



The Regression Problem

- Solution 3: Express $t_i = f(x_i) + \epsilon_i$ directly in terms of the data points
- Idea: Any finite set of values t_i sampled from $(t_1, \ldots, t_n) \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$ has a **joint Gaussian distribution** with a covariance matrix K given by

$$
\begin{aligned}
k_{ij} = \mathrm{cov}(t_i, t_j) &= \mathrm{cov}(f(x_i), f(x_j)) \\
&=: c(x_i, x_j)
\end{aligned}
$$

Gaussian Process Models

- Then, the n+1 dimensional vector

$$\left(f(x_1),\ldots,f(x_n),f(x_{n+1})\right),$$

  which includes the new target to be predicted $t_{n+1} = f(x_{n+1})$, comes from an n+1 dimensional Gaussian

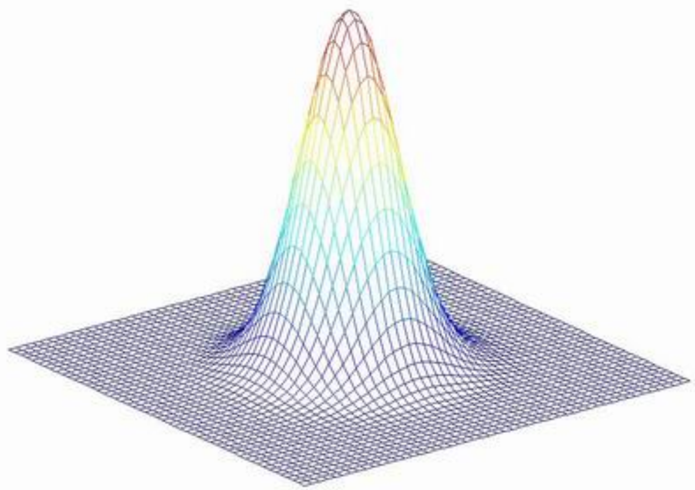
- The predictive distribution for the new target $p(t_{n+1} \mid x_{n+1}, \mathcal{X})$ is a 1-dimensional Gaussian

Gaussian Process Model

- Given the n observed points
- **Squared exponential** covariance function

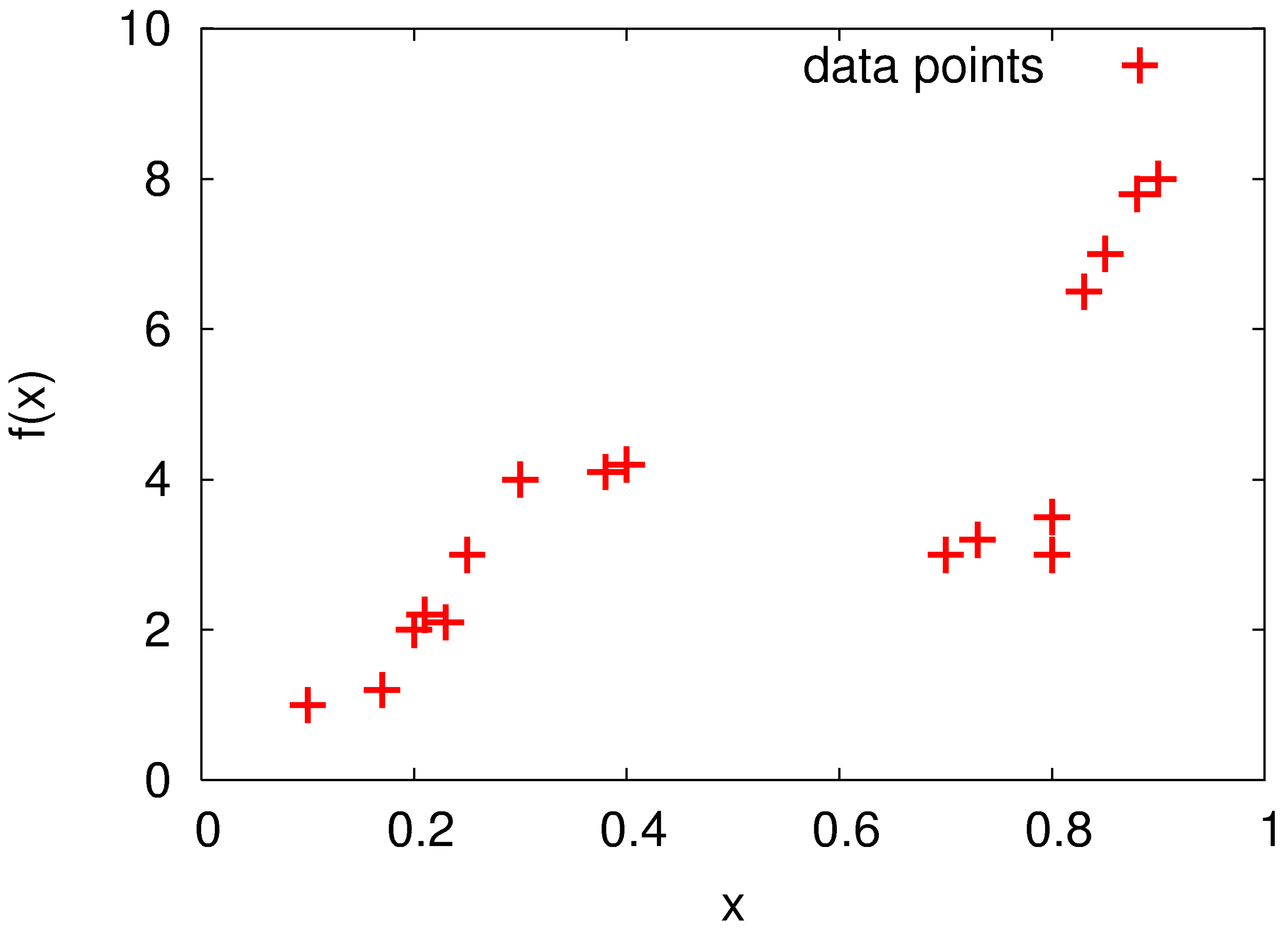
$$c(x_i, x_j) = \sigma_f^2 \cdot \exp\left(-\frac{(x_i - x_j)^2}{\ell^2}\right) + \delta_{(i=j)}\sigma_n^2$$

- with $\sigma_f = \frac{1}{6},\ \ell = 5,$
- and a noise level σ_n^2



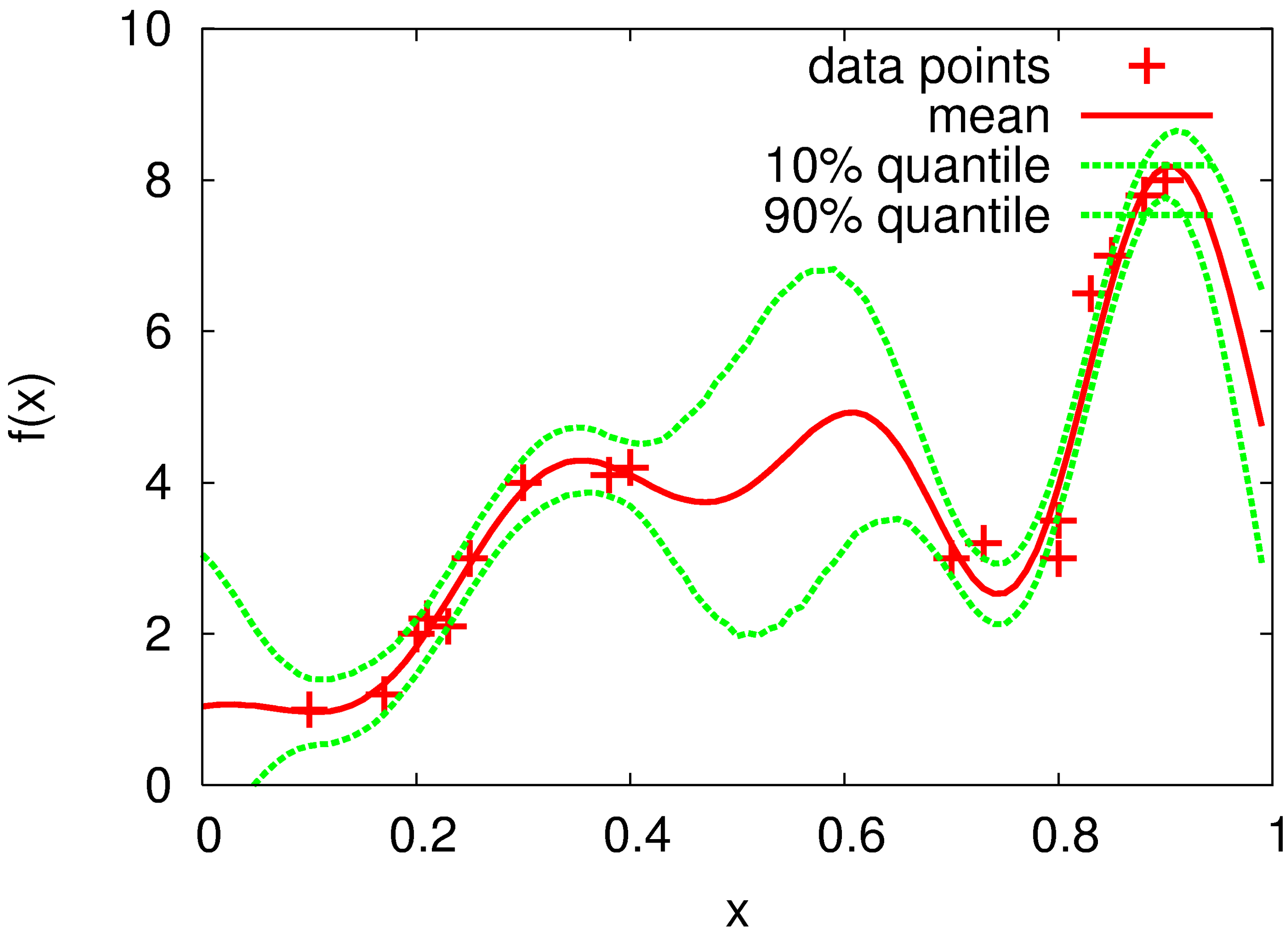
The Regression Problem

- Given n observed points



Gaussian Process Models

- GP model



Learning GPs

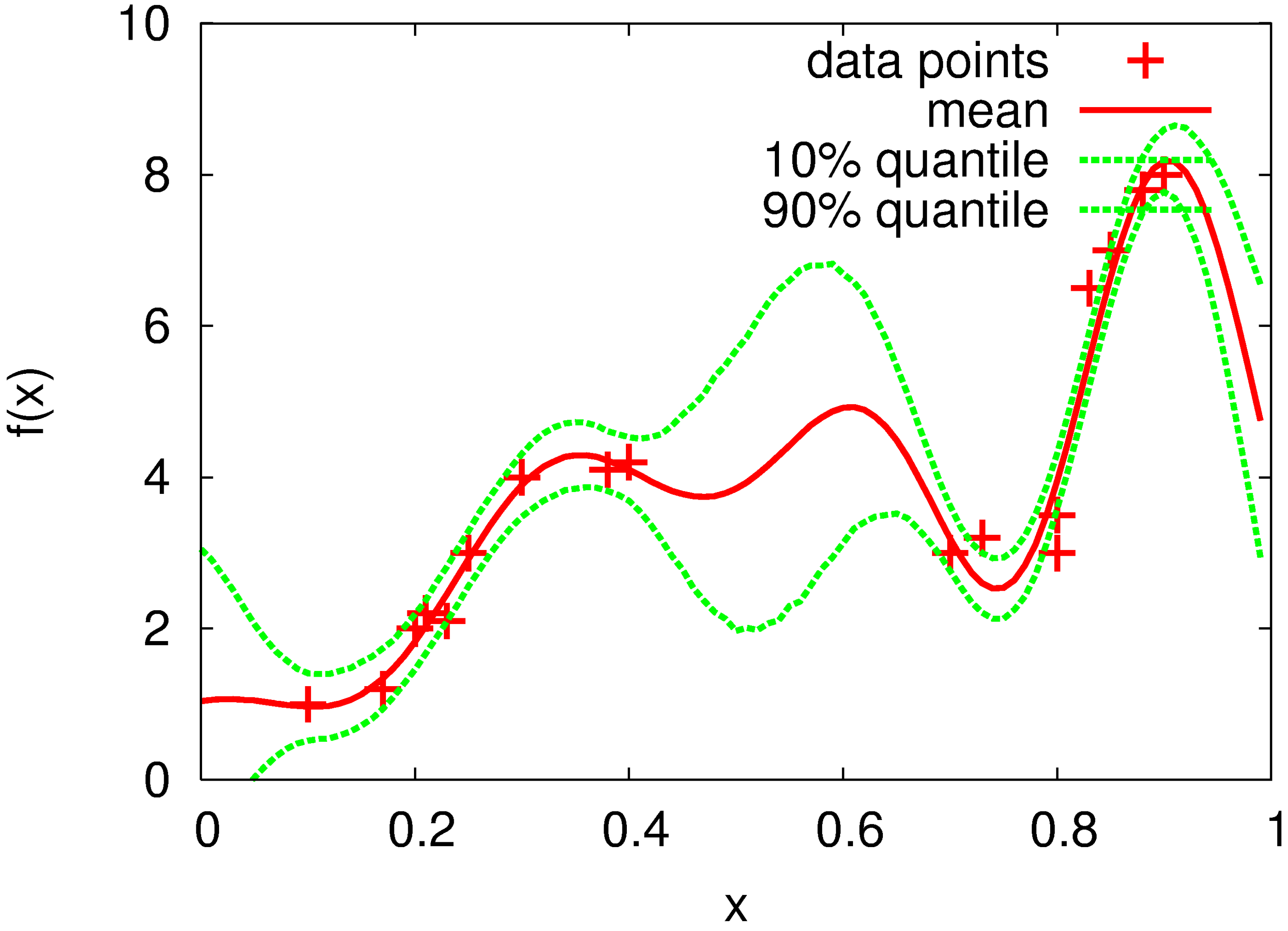
- The **squared exponential** covariance function:

$$c(x_i, x_j) = \underbrace{\sigma_f^2}_{\text{amplitude}} \cdot \exp\left(-\frac{\overbrace{(x_i - x_j)^2}^{\text{index/input distance}}}{\underbrace{\ell^2}_{\text{characteristic lengthscale}}}\right) + \delta_{(i=j)}\underbrace{\sigma_n^2}_{\text{noise level}}$$

- Easy to interpret parameters

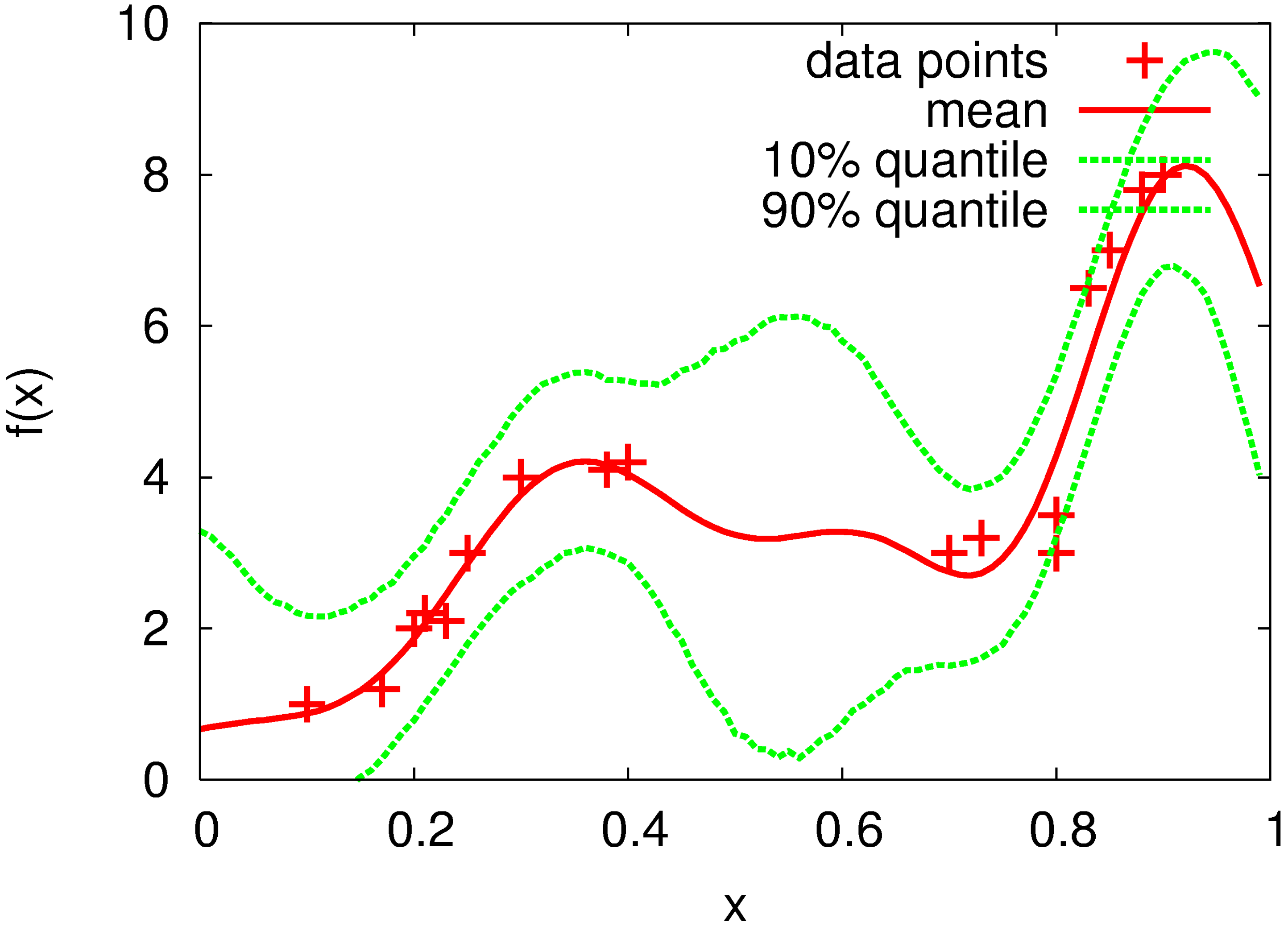
Learning GPs

- Example: low noise



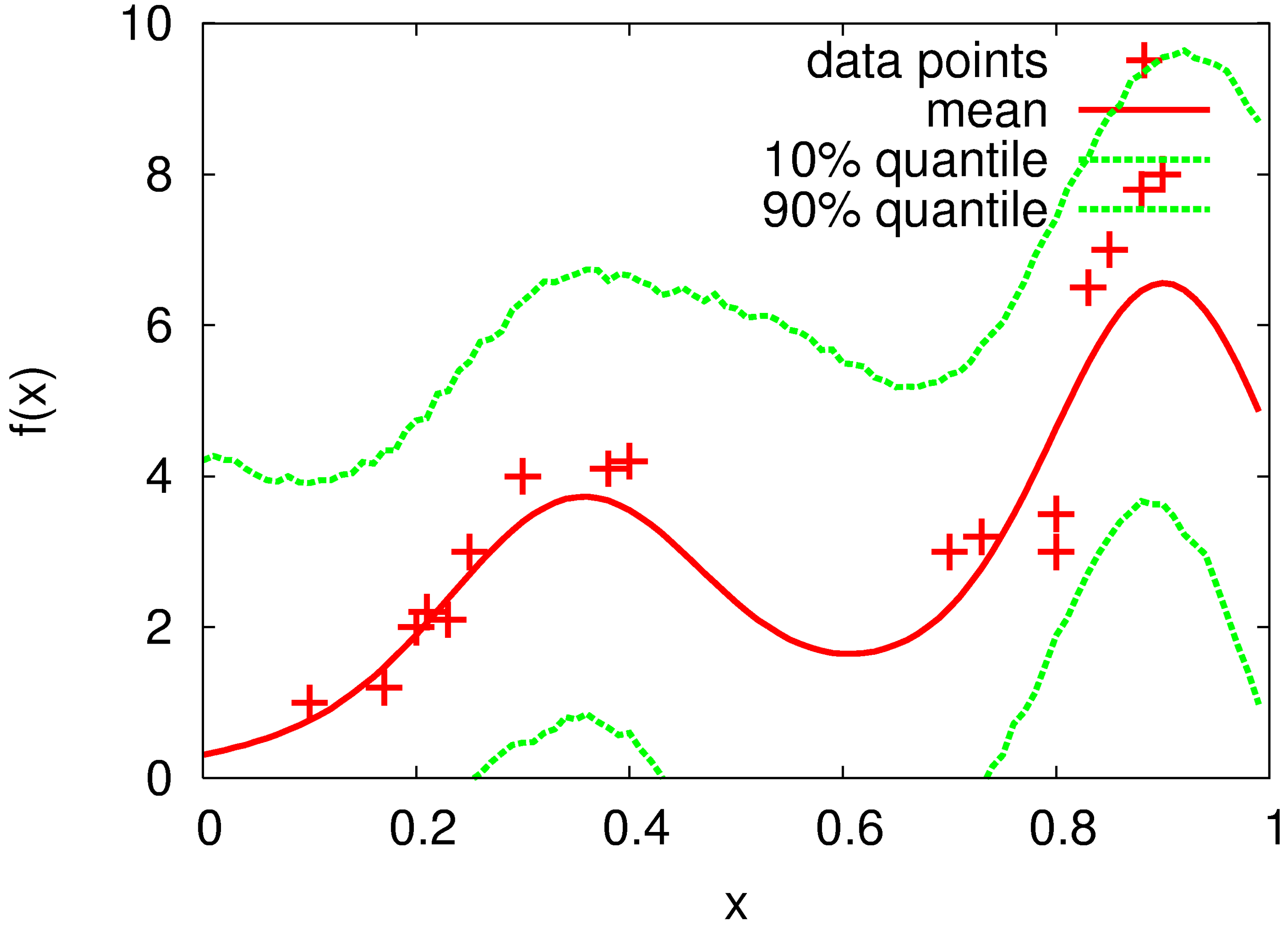
Learning GPs

- Example: medium noise



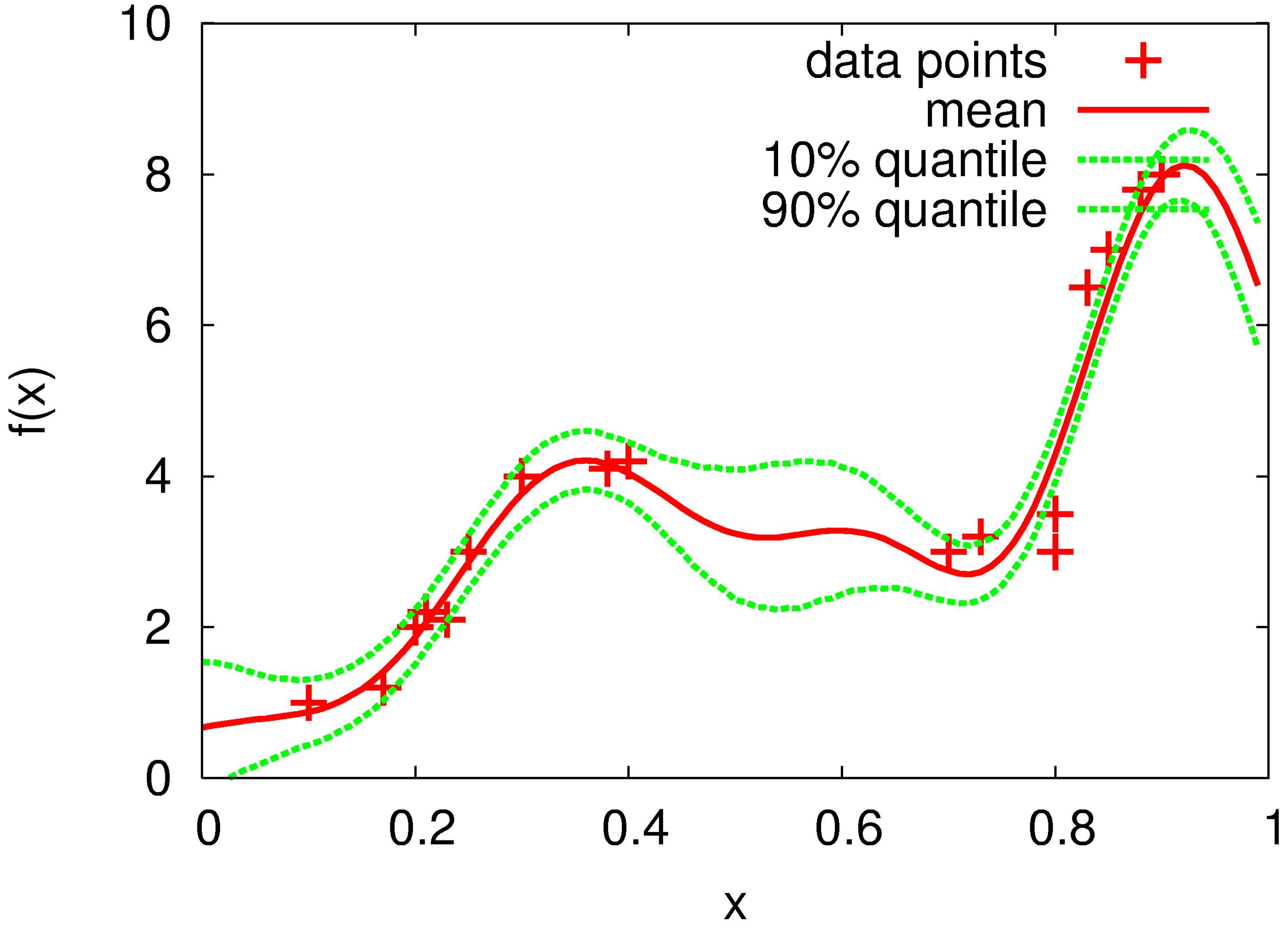
Learning GPs

- Example: high noise



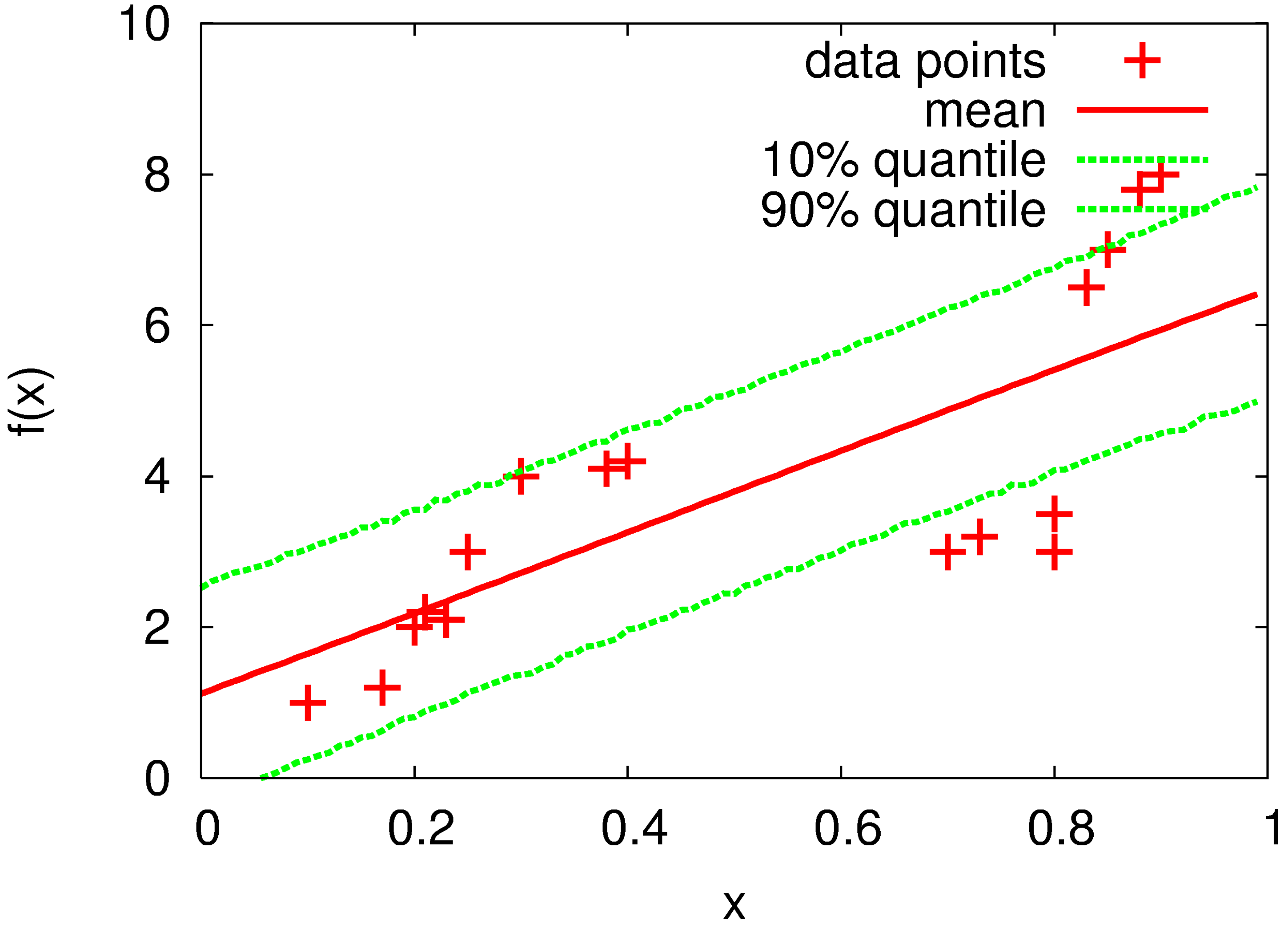
Learning GPs

- Example: small lengthscale



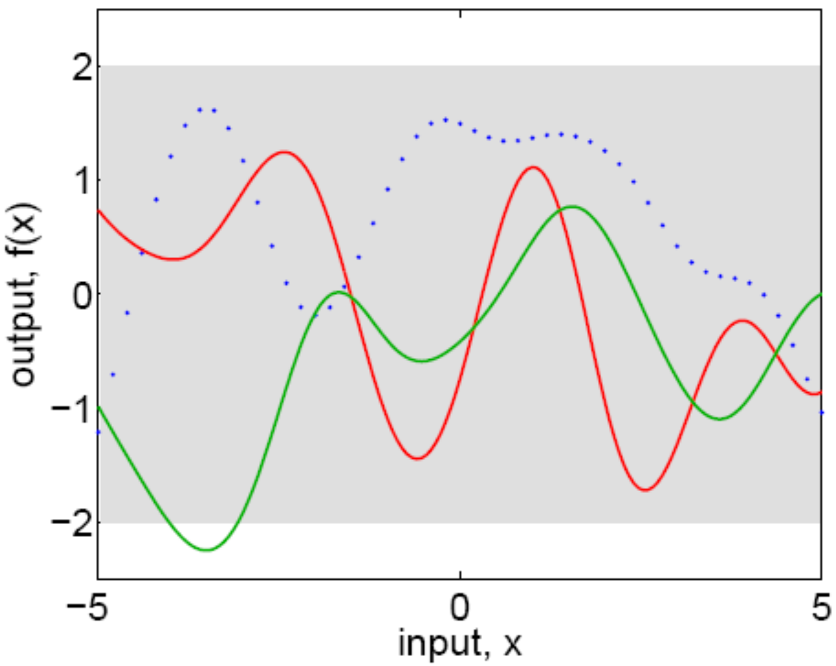
Learning GPs

- Example: large lengthscale

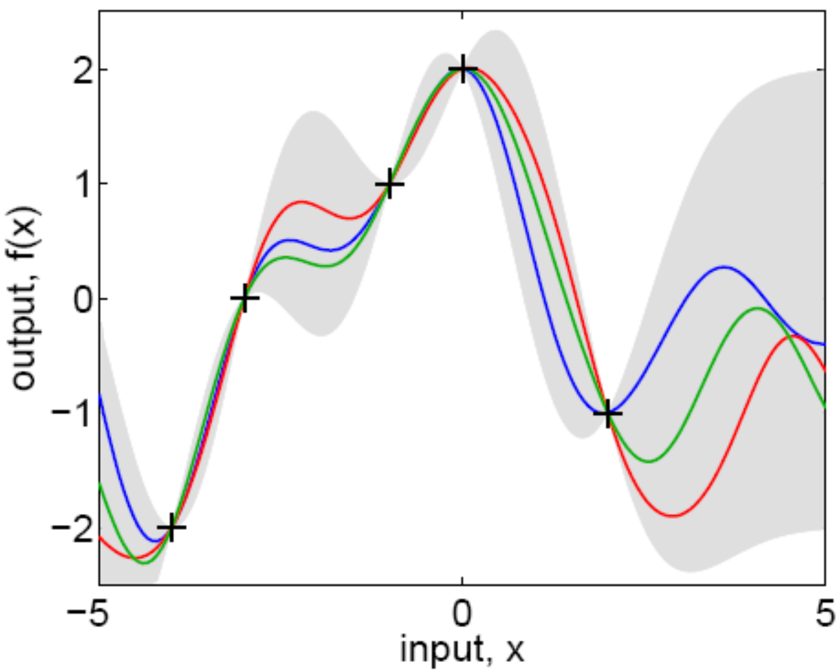


Learning GPs

- Covariance function specifies the **prior**



prior



posterior

Gaussian Process Models

- Recall, the n+1 dimensional vector

$$\Big(f(x_1), \ldots, f(x_n), f(x_{n+1})\Big),$$

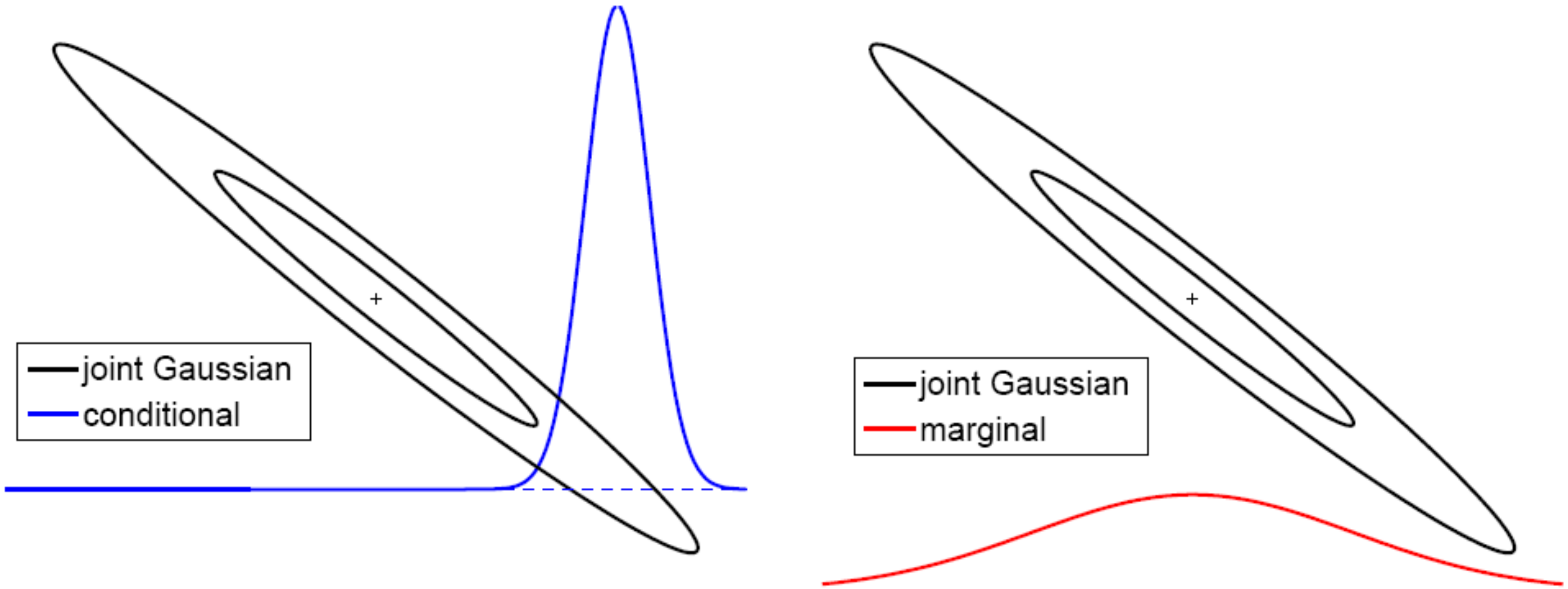
  comes from an n+1 dimensional normal distribution

- The predictive distribution for the new target $p(t_{n+1} \mid x_{n+1}, \mathcal{X})$ is a 1-dimensional Gaussian.

- **Why?**

The Gaussian Distribution

- Recall the 2-dimensional joint Gaussian:

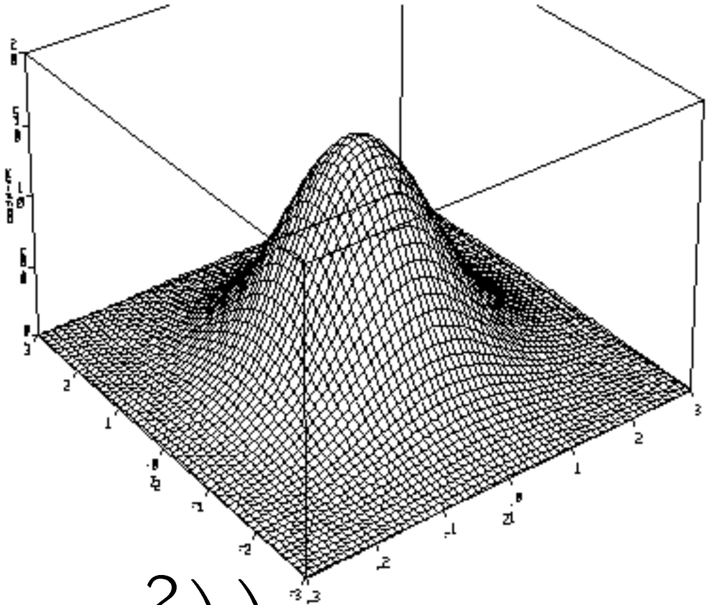


- The conditionals and the marginals are also Gaussians

Figure taken from
Carl E. Rasmussen:
NIPS 2006 Tutorial

The Gaussian Distribution

- Simple bivariate example:



$$p(x, y) = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$$

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}$$

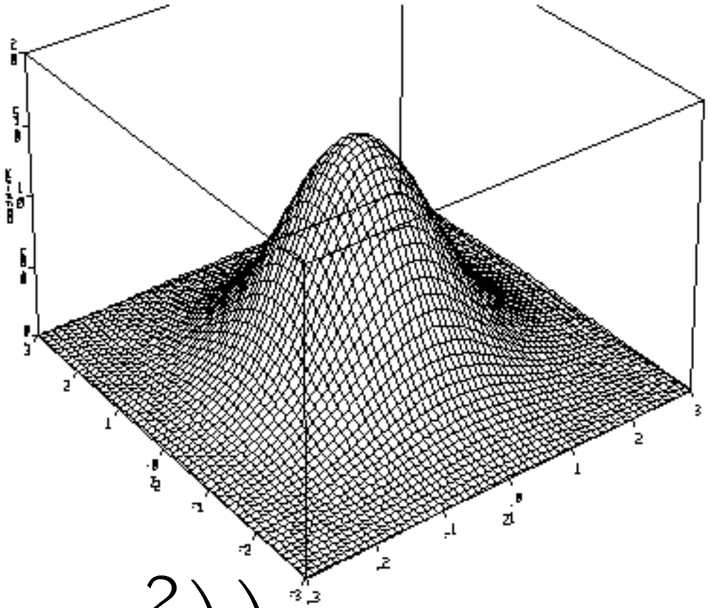
$$p(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) \right)$$

$$p(x \mid y) =$$

The Gaussian Distribution

- Simple bivariate example:

$$p(x, y) = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$$



$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}$$

$$p(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right)$$

$$p(x \mid y) = \frac{p(x, y)}{p(y)}$$

joint

marginal

conditional

The Gaussian Distribution

- Marginalization:

$$
\begin{aligned}
p(y) &= \int p(x,y)\ dx \\
&= \int \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right)\ dx \\
&= \underbrace{\frac{1}{\sigma_y\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{y^2}{\sigma_y^2}\right)}_{\mathcal{N}(0,\sigma_y^2)}
\end{aligned}
$$

The Gaussian Distribution

- The conditional:

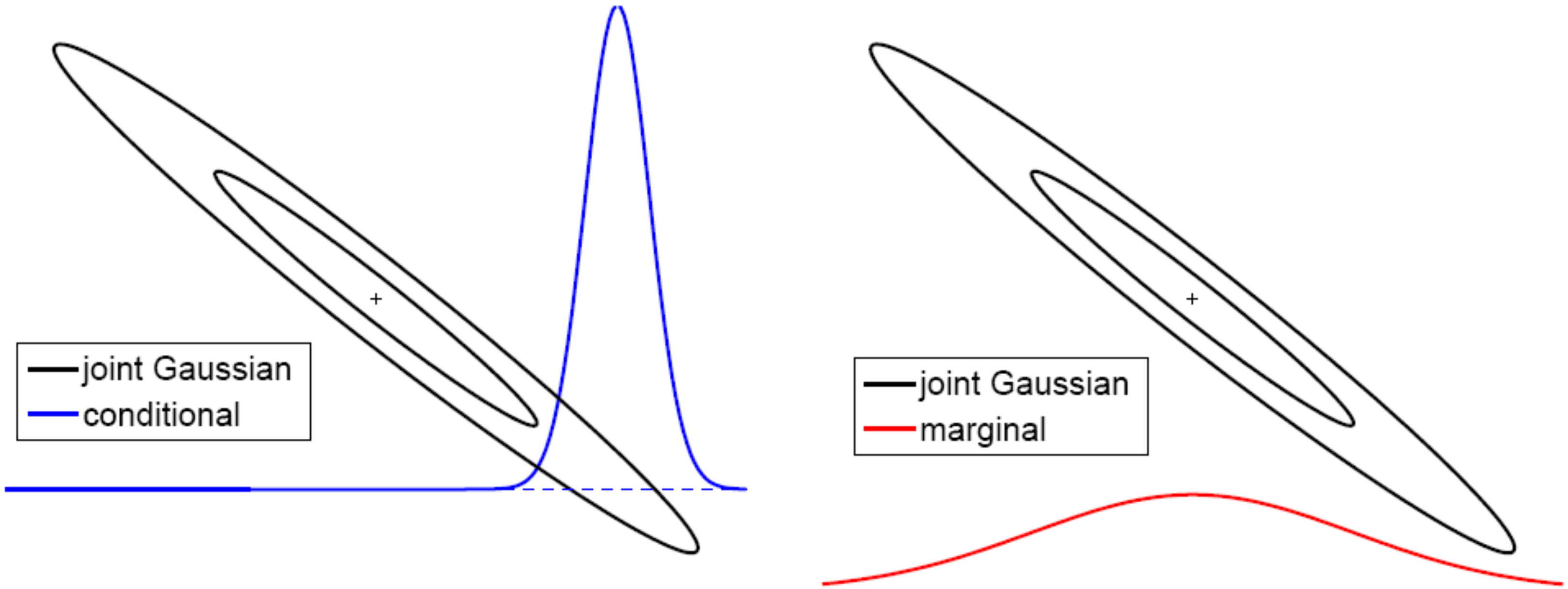
$$p(x \mid y) = \frac{p(x,y)}{p(y)} =$$

$$\underbrace{\frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)\right)}_{p(x,y)} \cdot \underbrace{\left(\frac{1}{\sigma_y\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{y^2}{\sigma_y^2}\right)\right)^{-1}}_{p(y)^{-1}}$$

$$= \underbrace{\frac{1}{\sigma_x\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{x^2}{\sigma_x^2}\right)}_{\mathcal{N}(0,\sigma_x^2)}$$

The Gaussian Distribution

- Slightly more complicated in the general case:



- The conditionals and the marginals are also Gaussians

Figure taken from
Carl E. Rasmussen:
NIPS 2006 Tutorial

The Gaussian Distribution

- Conditioning the joint Gaussian in general

$$p(x,y) = \mathcal{N}\left(\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}, \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix}\right)$$

$$p(x \mid y) = \mathcal{N}(\mathbf{a} + \mathbf{B}\mathbf{C}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T)$$

- In case of zero mean:

$$p(x,y) = \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix}\right)$$

$$p(x \mid y) = \mathcal{N}(\mathbf{B}\mathbf{C}^{-1}\mathbf{y}, \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T)$$

Gaussian Process Models

- Recall the GP assumption

$$\mathbf{t} = (t_1, \ldots, t_n)^T$$

$$\mathbf{t} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$$

$$\begin{pmatrix} \mathbf{t} \\ t_{n+1} \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{pmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & v \end{pmatrix})$$

$$t_{n+1} \mid \mathbf{t} \sim \mathcal{N}(\mu^*, \sigma^*)$$

Gaussian Process Models

- **Noise-free** mean and variance of the predictive distribution have the form

$$\mu^* = E(t_{n+1} \mid t_1, \cdots, t_n) = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{t}$$

$$\sigma^* = V(t_{n+1} \mid t_1, \cdots, t_n) = v - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}$$

- with

$$\mathbf{K} = \begin{pmatrix} c(x_1, x_1) & \ldots & c(x_1, x_n) \\ \ldots & \ldots & \\ & \ldots & c(x_n, x_n) \end{pmatrix} \qquad k = \begin{pmatrix} c(x_1, x_{n+1}) \\ \ldots \\ c(x_n, x_{n+1}) \end{pmatrix}$$

$$v = c(x_{n+1}, x_{n+1}) \qquad t = \begin{pmatrix} t_1 \\ \ldots \\ t_n \end{pmatrix}$$

Gaussian Process Models

- Mean and variance of the predictive distribution then lead to

$$\mu^* = \mathbf{k}^T(\mathbf{K} + \mathbf{I}\sigma_n^2)^{-1}\mathbf{t}$$

$$\sigma^* = c(x_{n+1}, x_{n+1}) - \mathbf{k}^T(\mathbf{K} + \mathbf{I}\sigma_n^2)^{-1}\mathbf{k}$$

- with

$$\mathbf{K} = \begin{pmatrix} c(x_1, x_1) & \dots & c(x_1, x_n) \\ \dots & \dots & \\ & \dots & c(x_n, x_n) \end{pmatrix} \quad k = \begin{pmatrix} c(x_1, x_{n+1}) \\ \dots \\ c(x_n, x_{n+1}) \end{pmatrix}$$

Learning GPs

- Learning a Gaussian process means
  - choosing a covariance function
  - finding its parameters and the noise level

- **What is the objective?**

Learning GPs

- The hyperparameters

$$\boldsymbol{\theta} = \left\{\sigma_f, (\ell_1, \ldots, \ell_n), \sigma_n^2\right\}$$

can be found by maximizing the likelihood of the training data

$$\boldsymbol{\theta} = \text{argmax}_\theta \ \log \ p(t_1, \ldots, t_n \mid x_1, \ldots, x_n, \boldsymbol{\theta}),$$

e.g., using gradient methods

Learning GPs

- Objective: high data likelihood

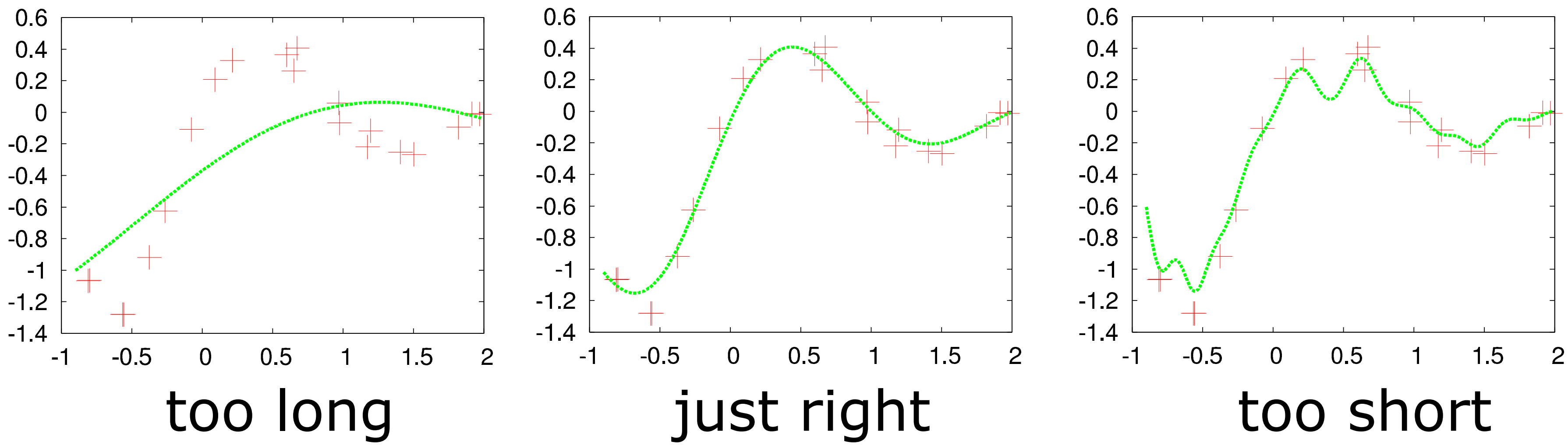
$$\log p(\mathbf{t} \mid \mathbf{x}) = -\frac{1}{2}\mathbf{t}^T(\mathbf{K}+\sigma_n^2)^{-1}\mathbf{t} - \frac{1}{2}\log|\mathbf{K}+\sigma_n^2\mathbf{I}| - \frac{n}{2}\log 2\pi$$

data fit — complexity penalty — const.

- Due to the Gaussian assumption, GPs have Occam's razor built in

Occam's Razor

- Use the simplest explanation that is needed to describe the data

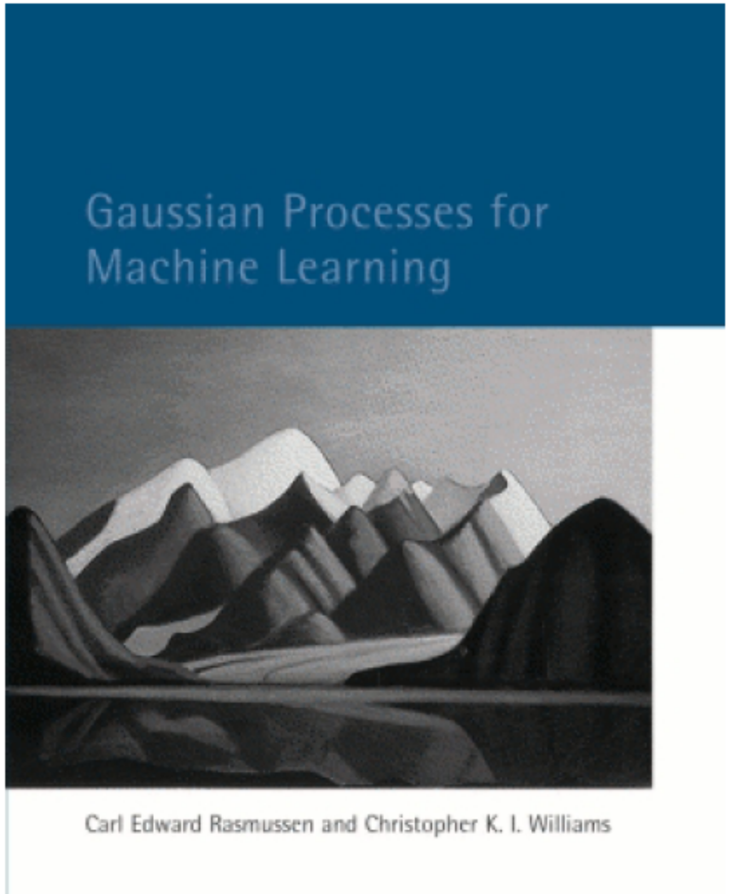


- **Data-fit** favors overfitting
- **Complexity penalty** favors simplicity

Advanced Topics / Extensions

- Classification/non-Gaussian noise
- Sparse GPs: Approximations for large data sets
- Heteroscedastic GPs: Modeling non-constant noise
- Nonstationary GPs: Modeling varying smoothness (lengthscales)
- Mixtures of GPs
- Uncertain inputs
- …

Further Reading



Rasmussen and Williams
Gaussian Processes for Machine Learning,
MIT Press, 2006.
`http://www.GaussianProcess.org/gpml`

Gaussian process web (code, papers, etc): `http://www.GaussianProcess.org`

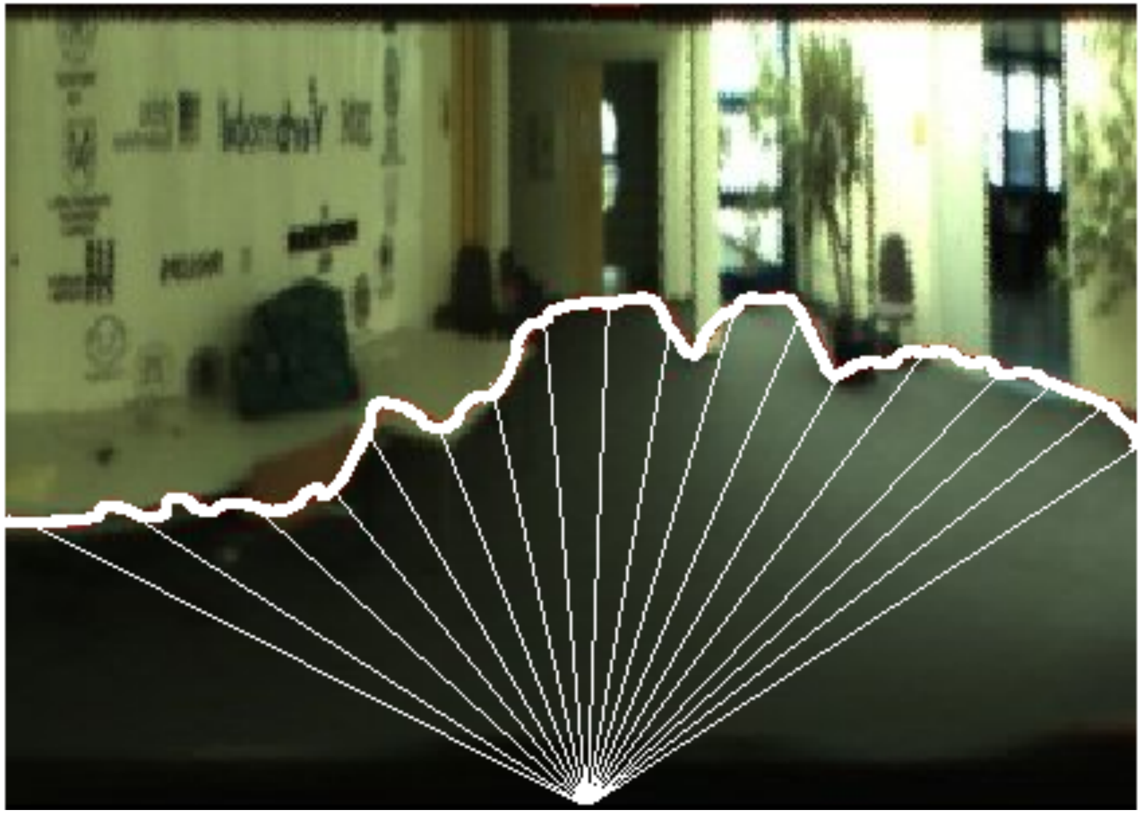
Applications in Robotics

- Monocular range sensing
- Terrain modeling
- Learning sensor models
- Learning to control a blimp
- Localization in cellular networks
- Time-series forecasting
- …

Applications in Robotics

- **Monocular range sensing**
- **Terrain modeling**
- Learning sensor models
- Learning to control a blimp
- Localization in cellular networks
- Time-series forecasting
- …

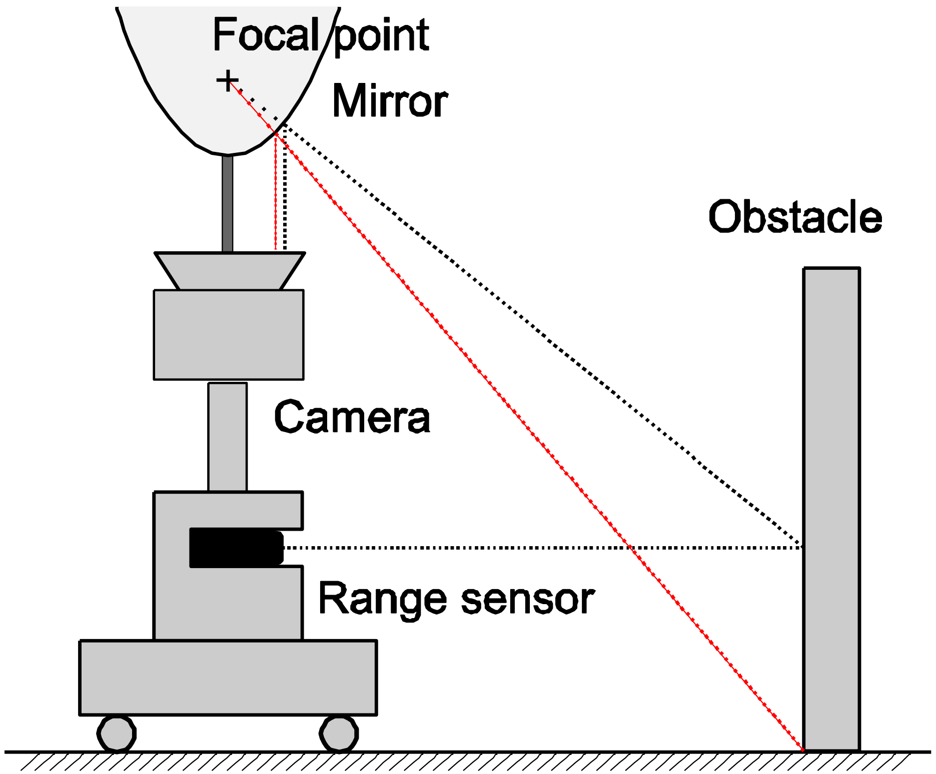
Monocular Range Sensing



- Can we learn range from single, monocular camera images?

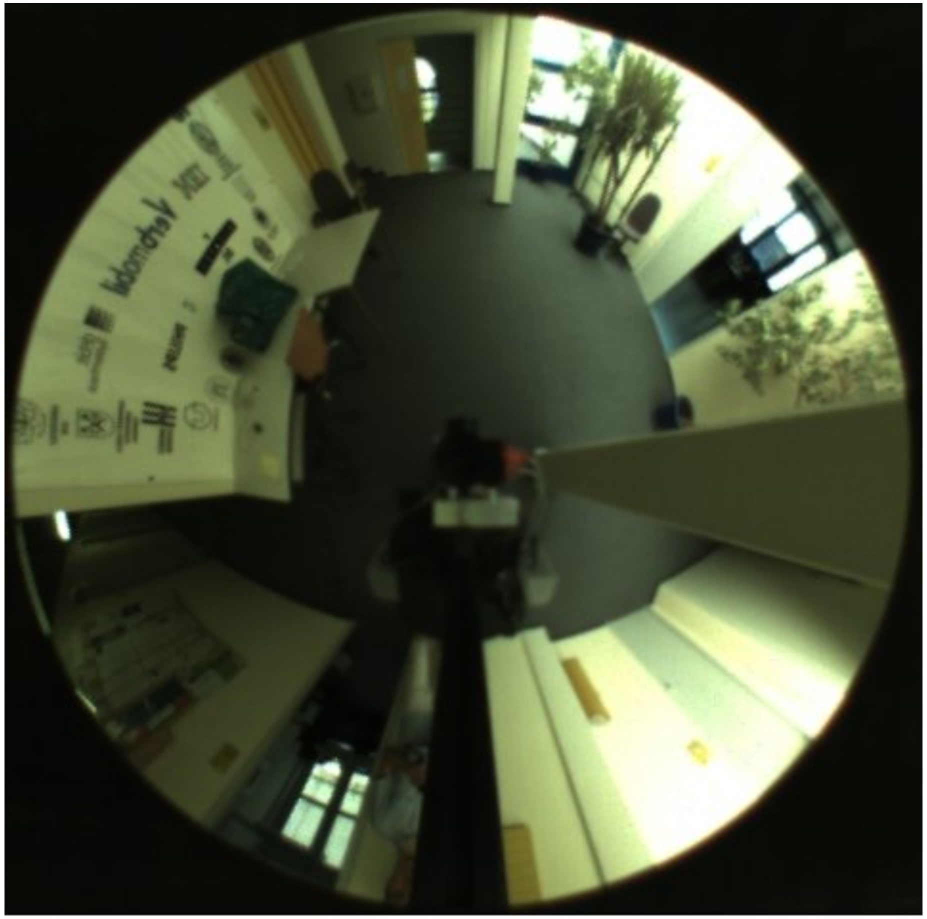
Training Setup

- Mobile robot + laser range finder
- Omni-directional monocular camera



Training Setup

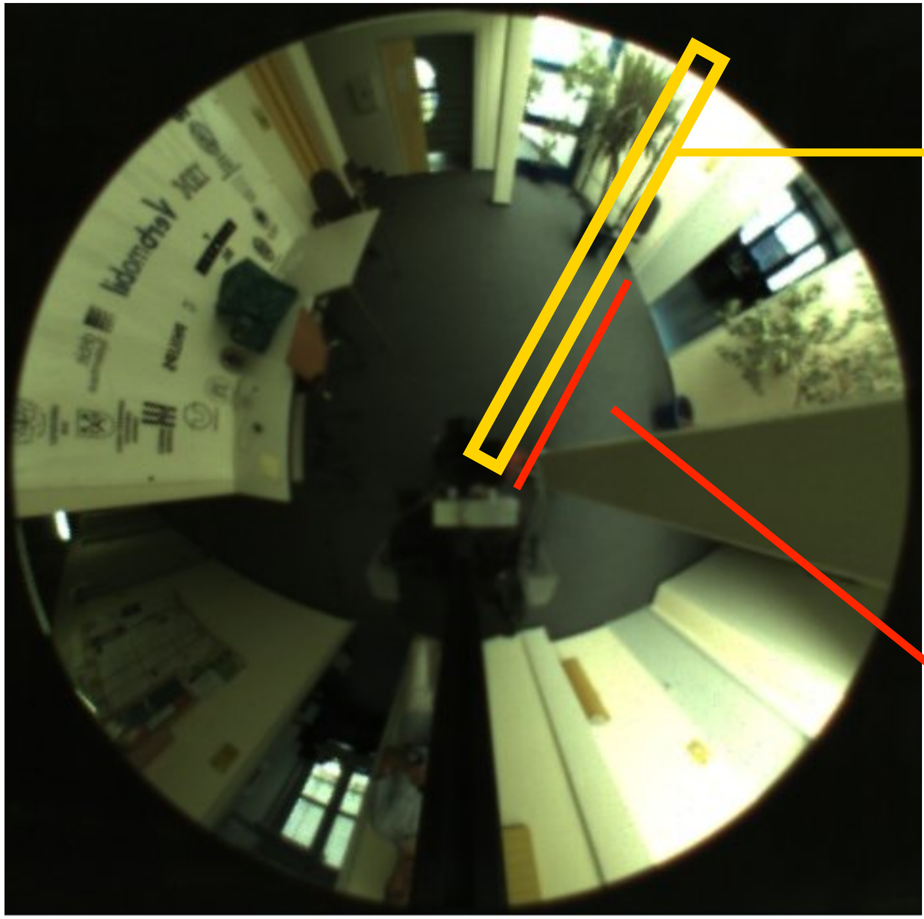
DFKI Saarbrücken



University of Freiburg

Learning Range from Vision

- Associate (polar) pixel columns with ranges



$$\mathbf{p}_i \in \mathbb{R}^{420}$$

Extract features

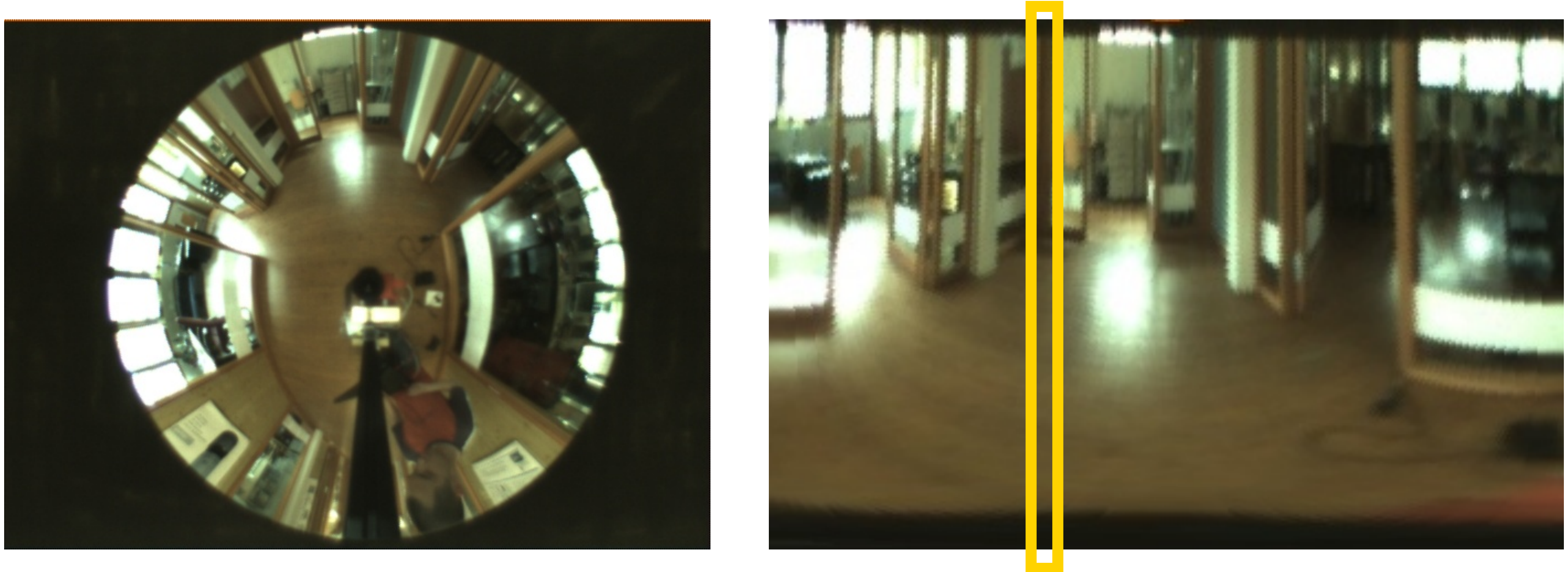
$$\mathbf{x}_i = f(\mathbf{p}_i)$$

Associate with ranges

$$r_i = r(\mathbf{x}_i) \in \mathbb{R}$$

Pre-processing

- Warp images into a panoramic view



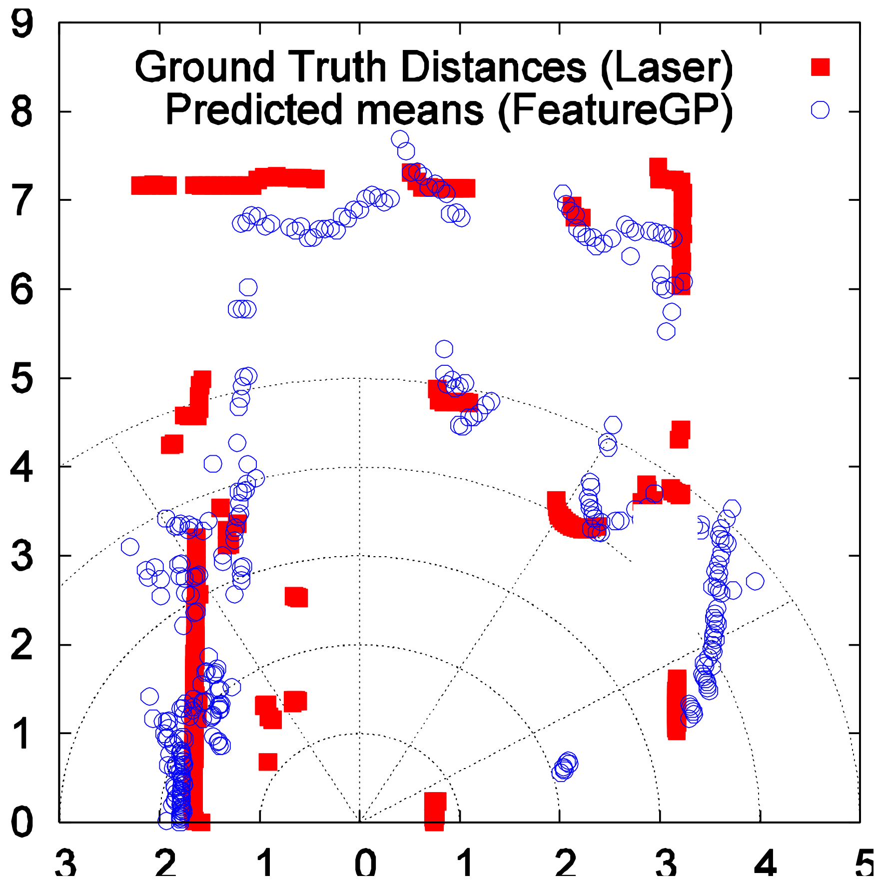
- 120 pixels per column
- Transform to HSV -> 420 dimensions

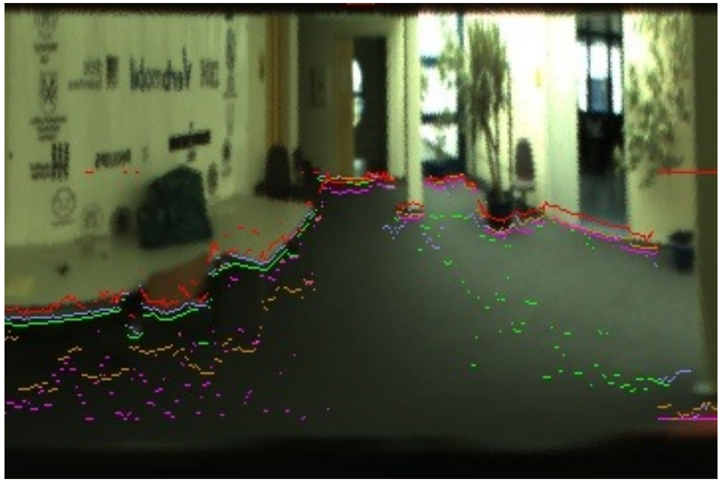
Visual Features

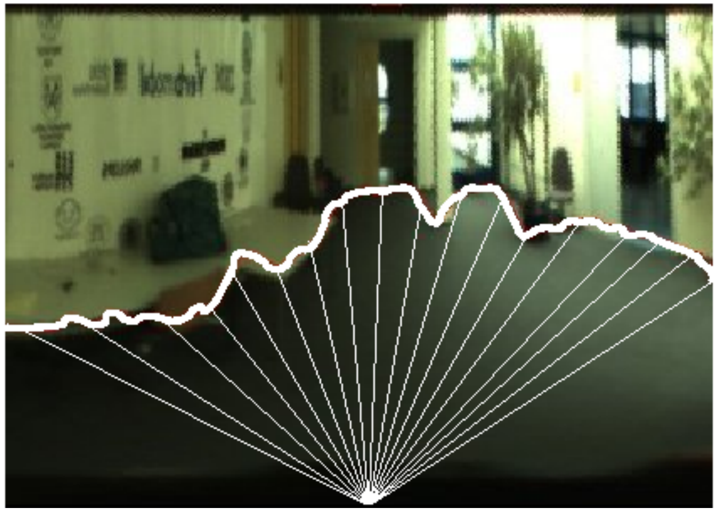
- Two types of features
  1. No human engineering: Principle components analysis (PCA) on raw input
  2. Use of domain specific knowledge: Edge features that shall correspond to floor boundaries

Experiments

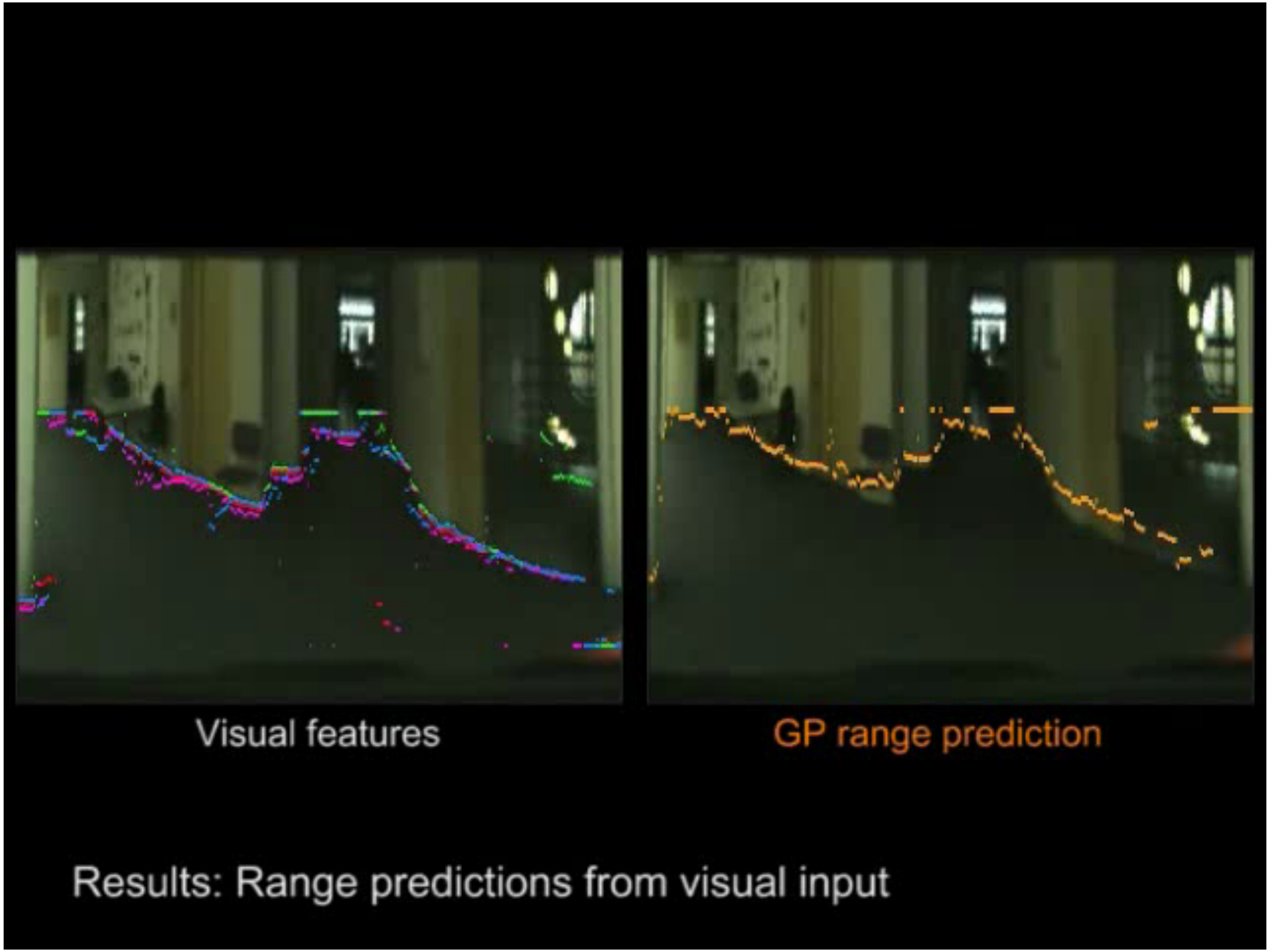
Typical 180° scan







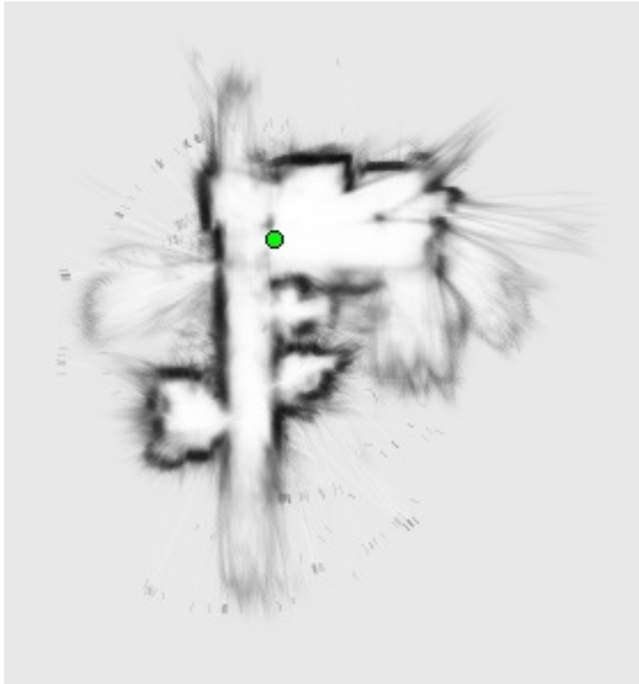
Online Prediction



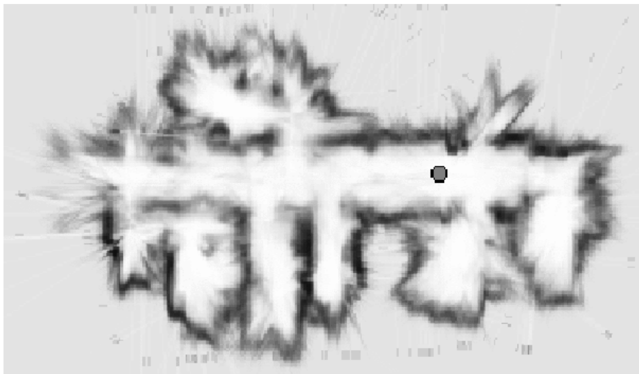
Mapping Results

Laser-based | Vision-based

Saarbrücken:

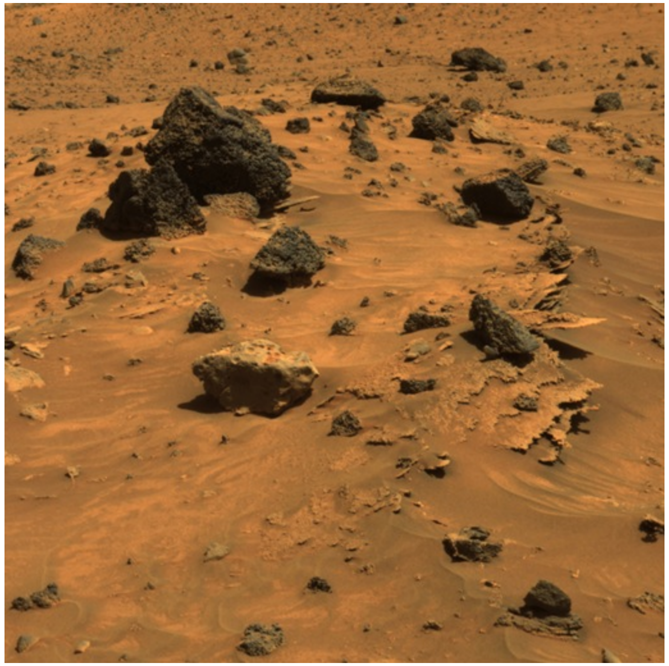


Freiburg:



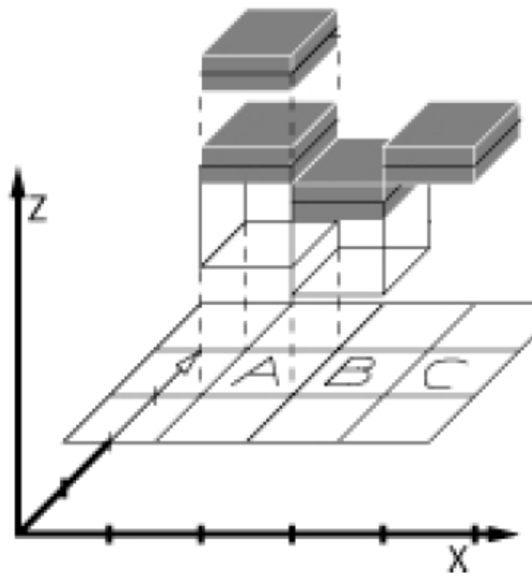
GP-based Terrain Modeling

- **3D terrain models** are important in many tasks in outdoor robotics



Terrain Modeling

- **Given:** observations of the terrain surface
- **Task:** Learn a predictive model
- Classic Approach: Elevation grid maps



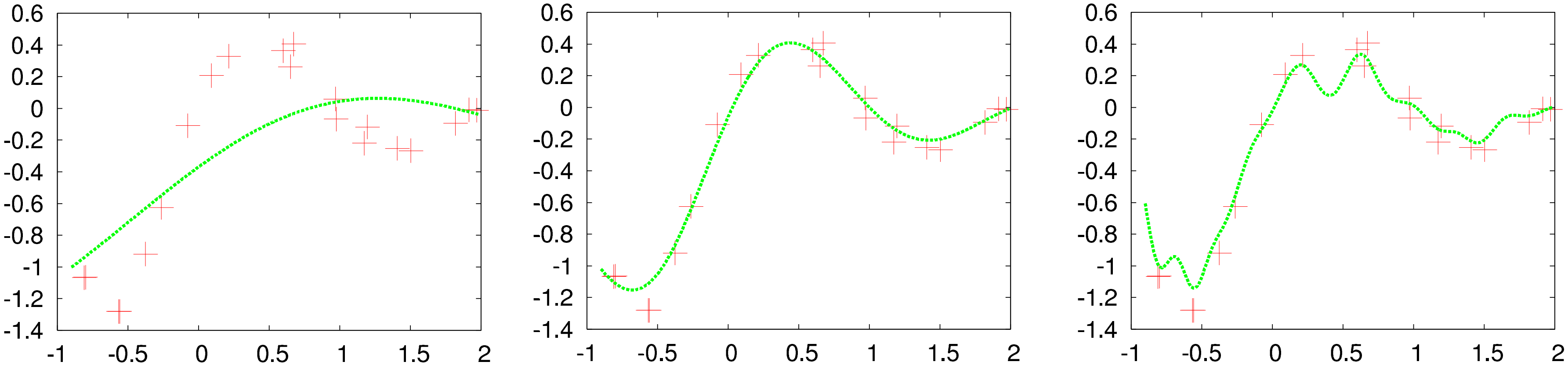
GP-Based Approach

- Generalize the grid-based model to **fully continuous spaces** by viewing the problem as function regression

- Requirements
  - **Probabilistic** formulation to handle uncertainty
  - Ability to **adapt** to the spatial structures

Covariance Function

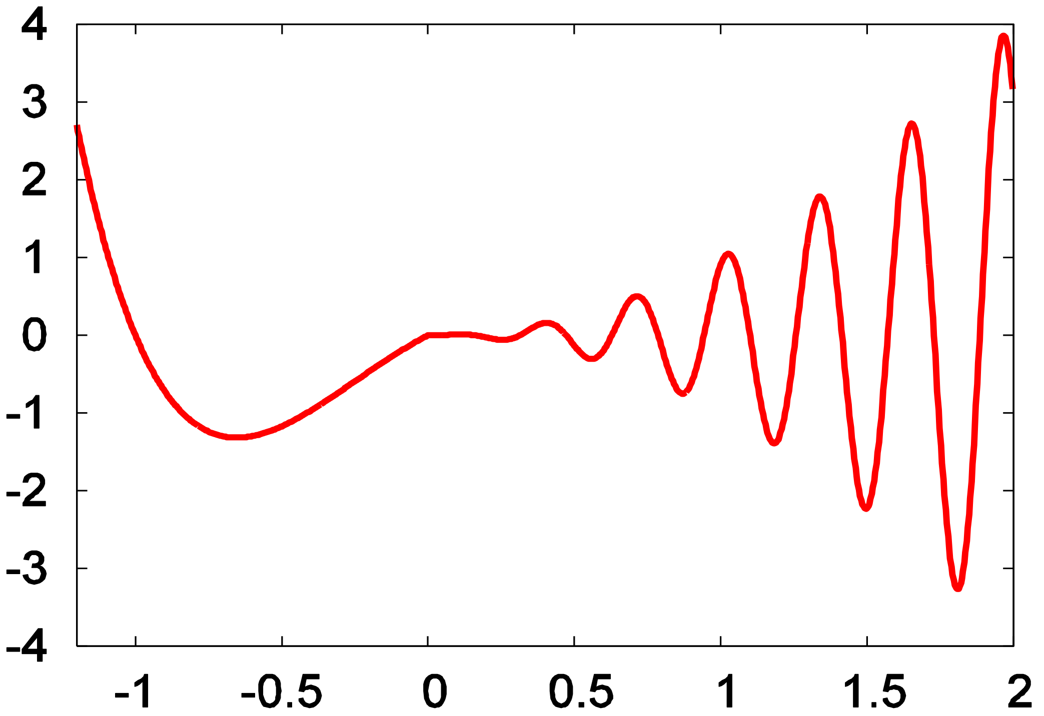
- Standard covariance function have limited flexibility to adapt to the local spatial structure



strong smoothing | medium smoothing | little smoothing

Covariance Function

- What is **optimal** in this case?



Local Kernel Adaptation

- Adapt kernels based on the **terrain gradients**
- Covariance is adjusted according to the change in terrain elevation in the local neighborhood

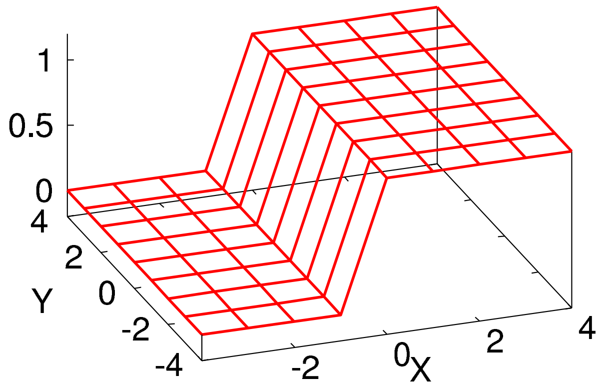
local average

$$\Sigma_i = EST(\mathbf{x_i})^{-1} = \overline{(\nabla \mathbf{y}(\mathbf{x_i}))(\nabla \mathbf{y}(\mathbf{x_i}))^{\mathbf{T}}}$$

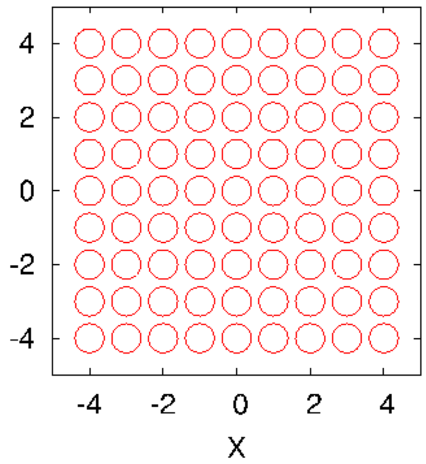
elevation gradient

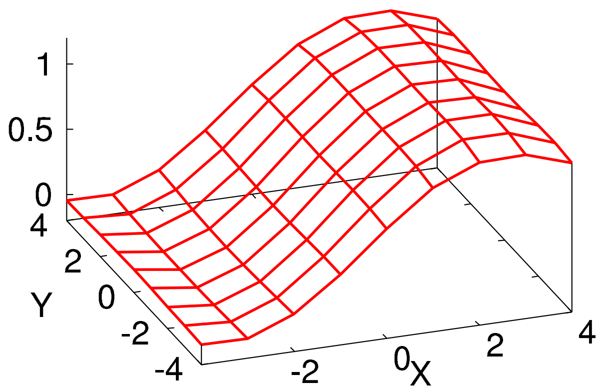
Adapting to Local Structures

Ground truth

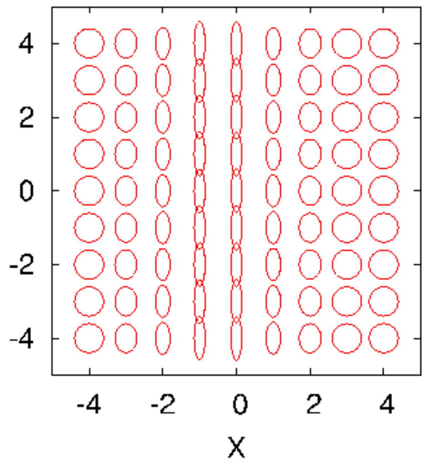


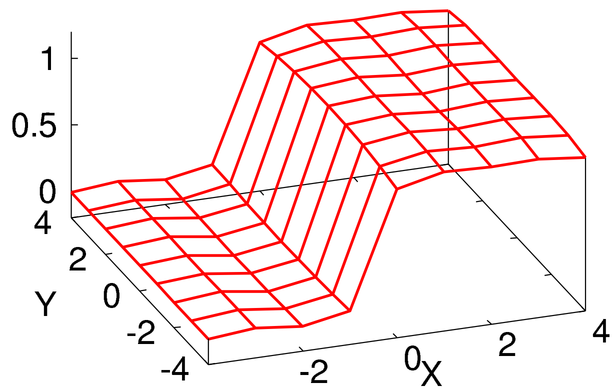
Stationary GP





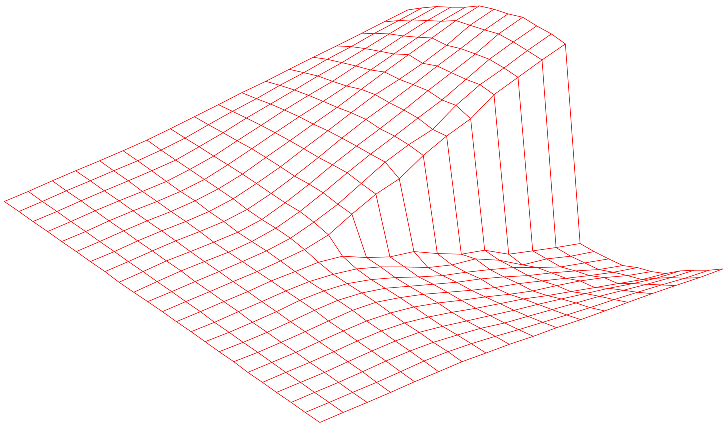
Non-stationary GP

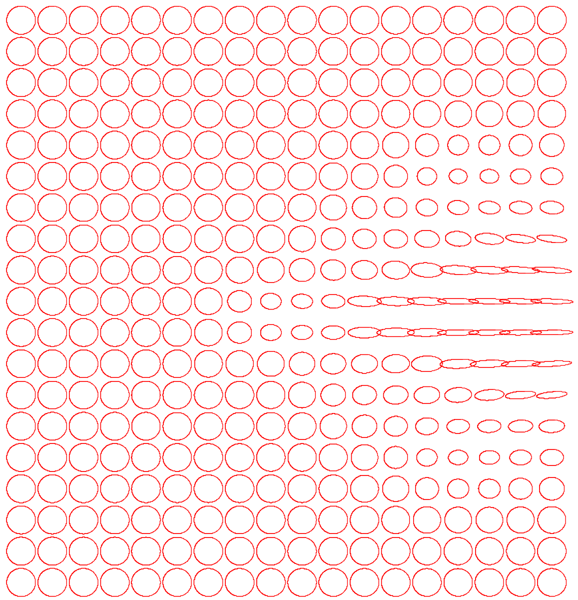




Adapting to Local Structure

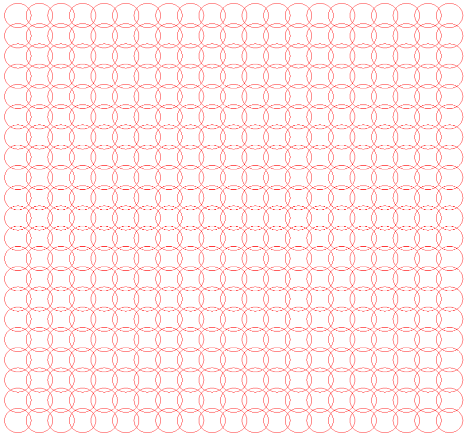
- Model to deal with slowly changing characteristics and strong discontinuities

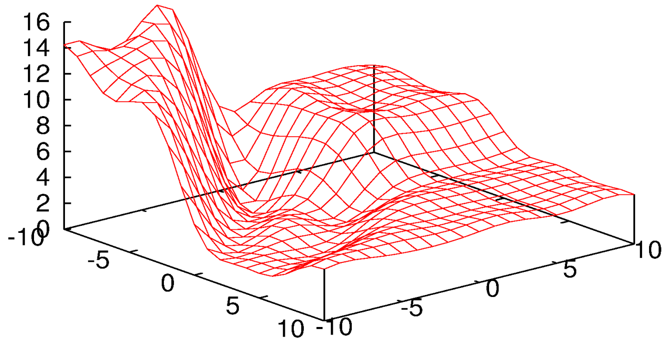


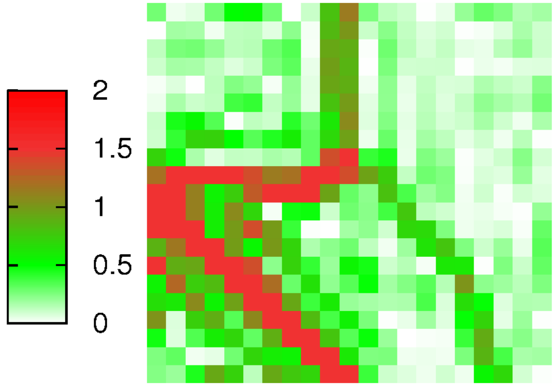


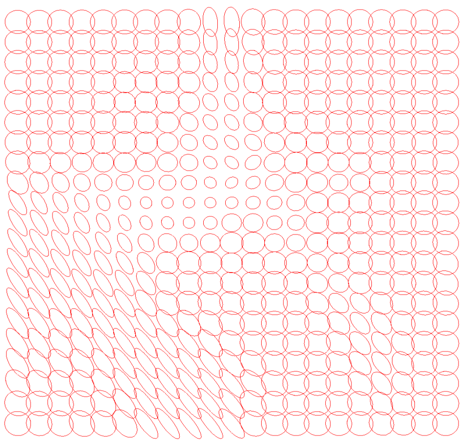
Experiments

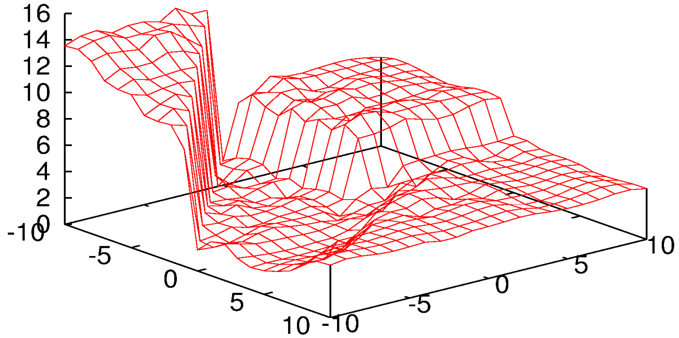
standard

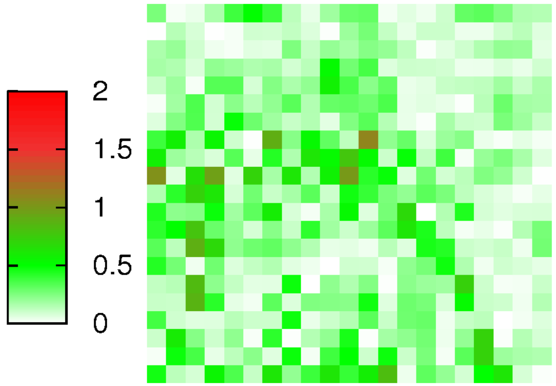






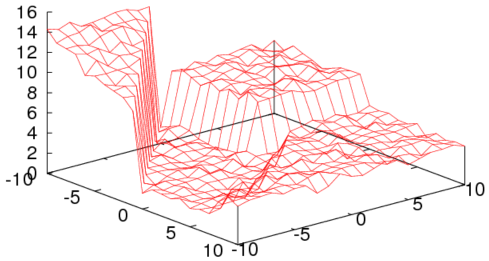






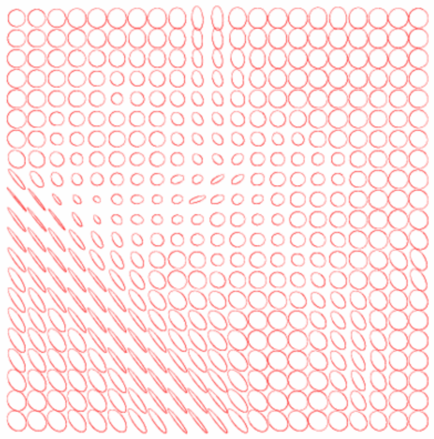
adaptive

Experiments

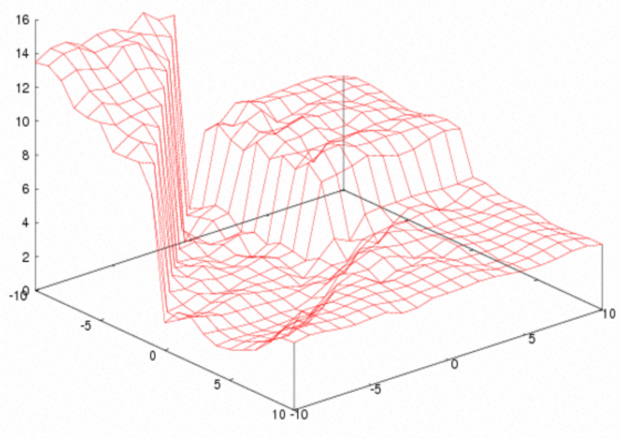


Observation (with white noise σ=0.3)

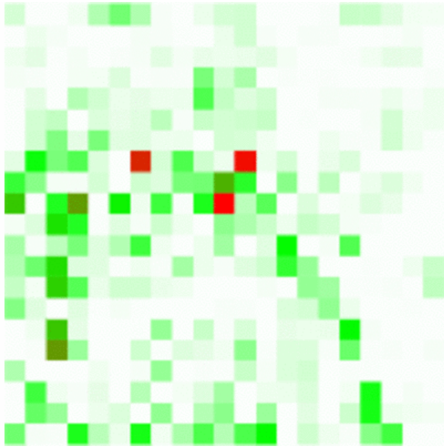
Kernels



Predicted Map

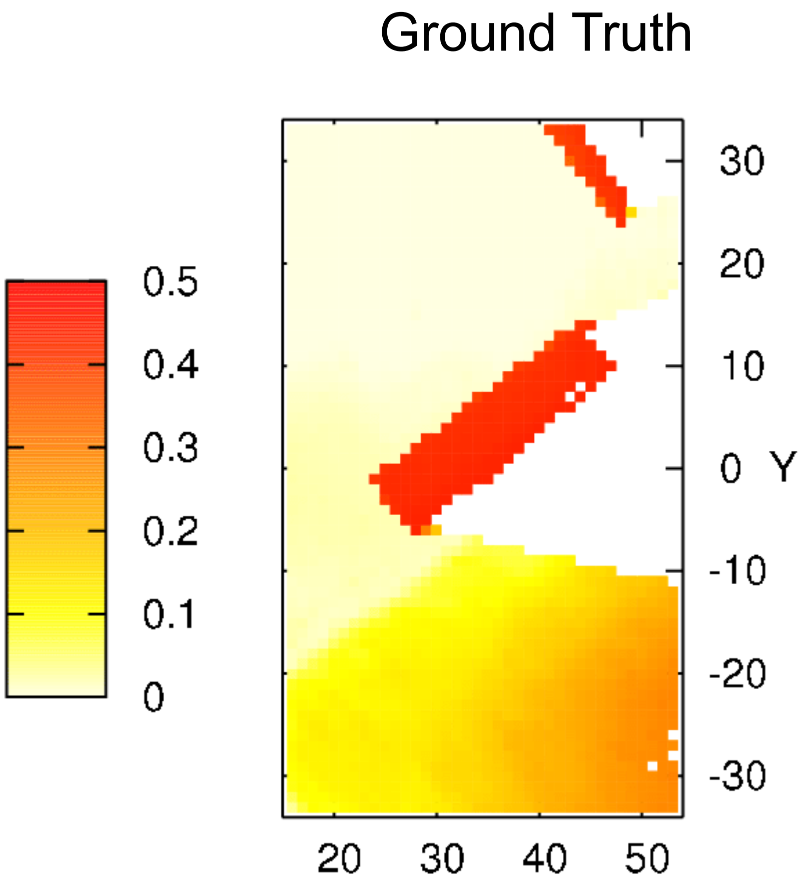


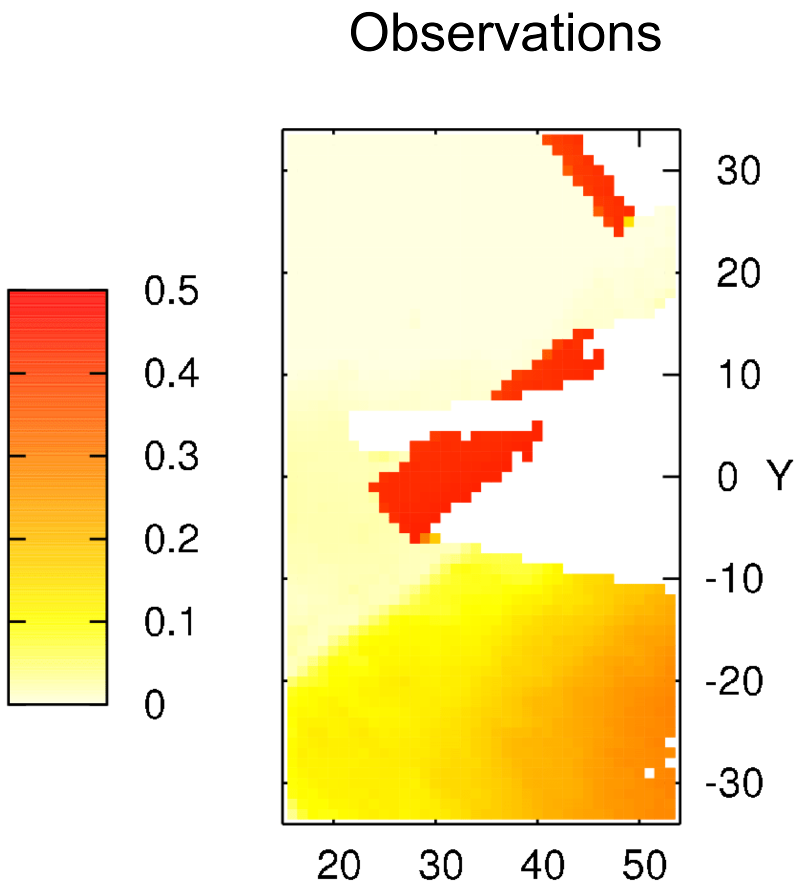
Local errors



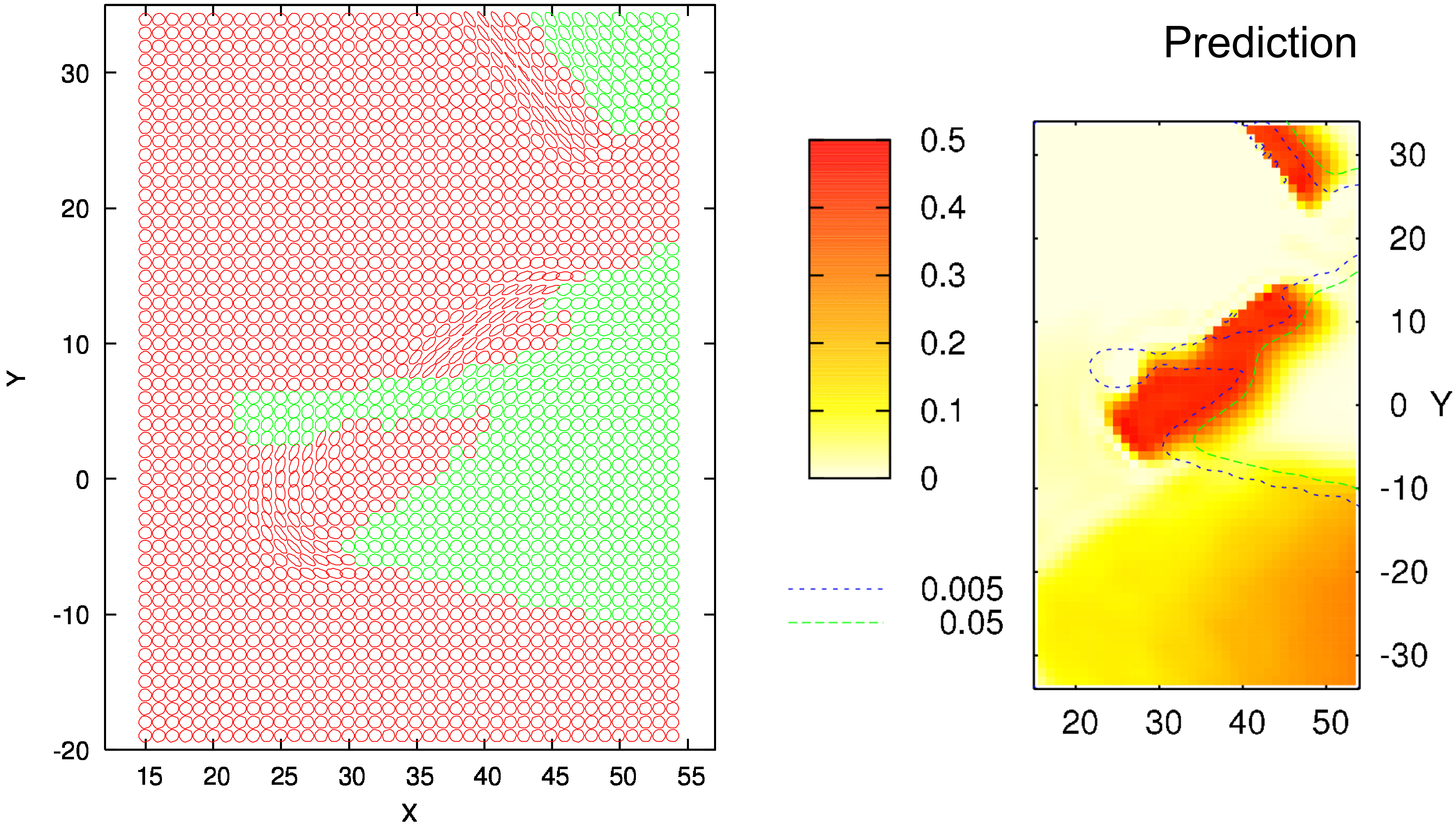
Experiments – Stone Block

Experiments – Stone Block



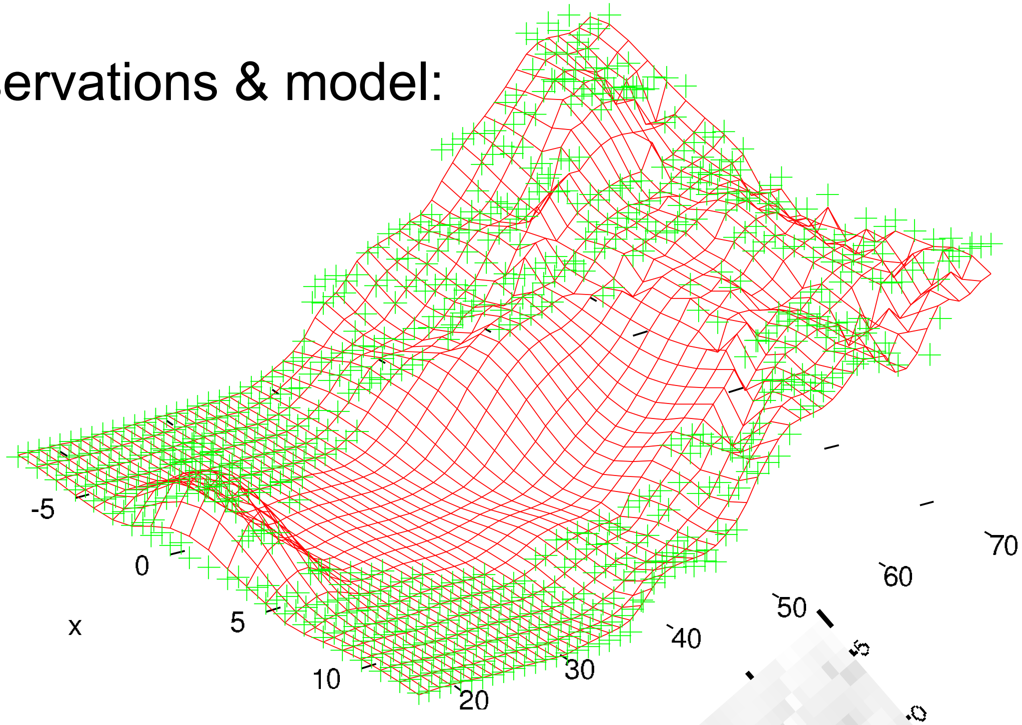


Experiments – Stone Block

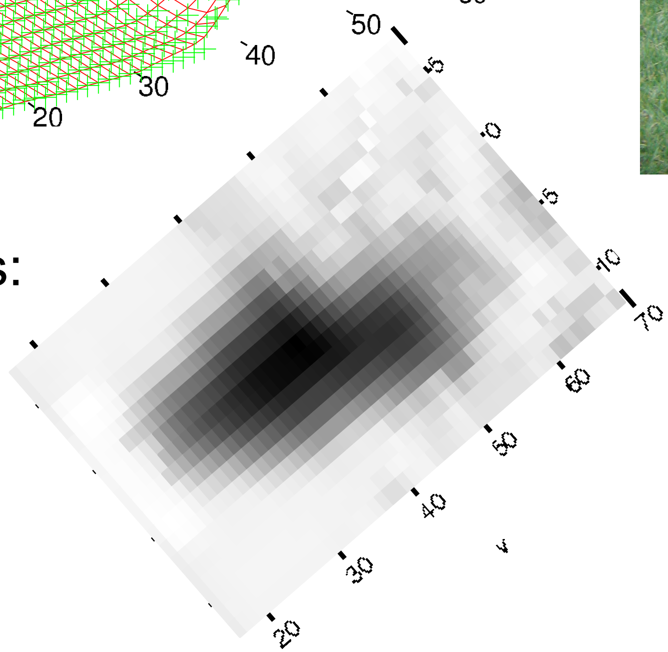


Experiments – Slope

Observations & model:



Uncertainties:



Summary

- GPs are a flexible and practical approach to Bayesian regression
- Prior knowledge is encoded in a human understandable way
- Learned models can be interpreted
- Efficiency mainly depends on the number of training points
- Real-world problem sizes require approximations/sparsity/…