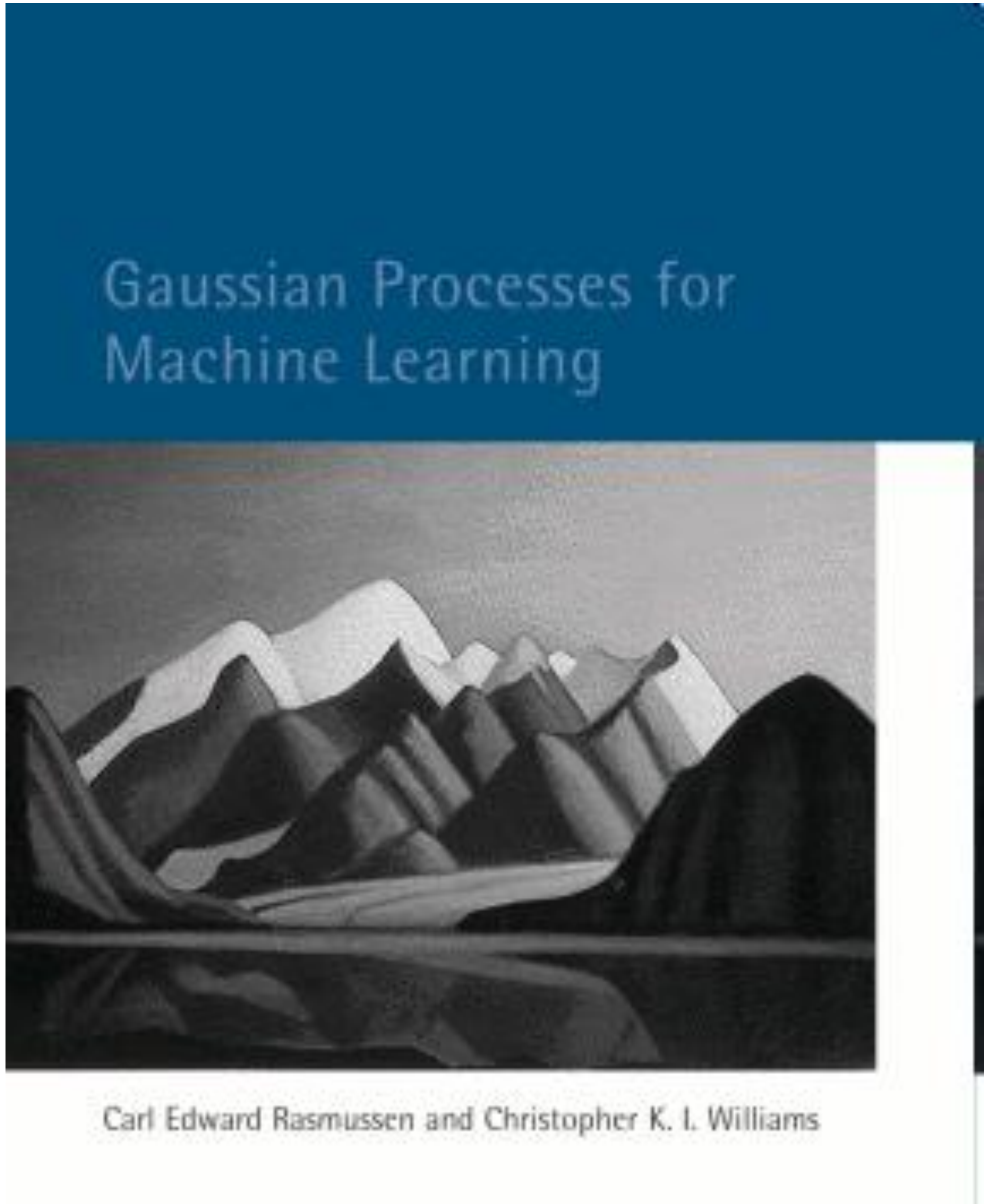
Advanced Introduction to Machine Learning CMU-10715

Gaussian Processes

Barnabás Póczos

MACHINE LEARNING DEPARTMENT
Carnegie Mellon.
School of Computer Science



http://www.gaussianprocess.org/

Some of these slides in the intro are taken from D. Lizotte, R. Parr, C. Guesterin

Contents

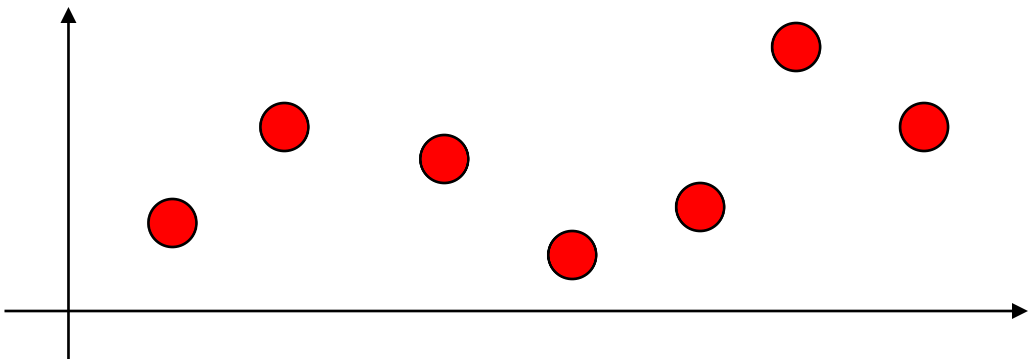
- Introduction
- Ridge Regression
- Gaussian Processes
  - Weight space view
    - Bayesian Ridge Regression + Kernel trick
  - Function space view
    - Prior distribution over functions + calculation posterior distributions

Contents

- Introduction
- Ridge Regression
- Gaussian Processes
  - Weight space view
    - Bayesian Ridge Regression + Kernel trick
  - Function space view
    - Prior distribution over functions + calculation posterior distributions

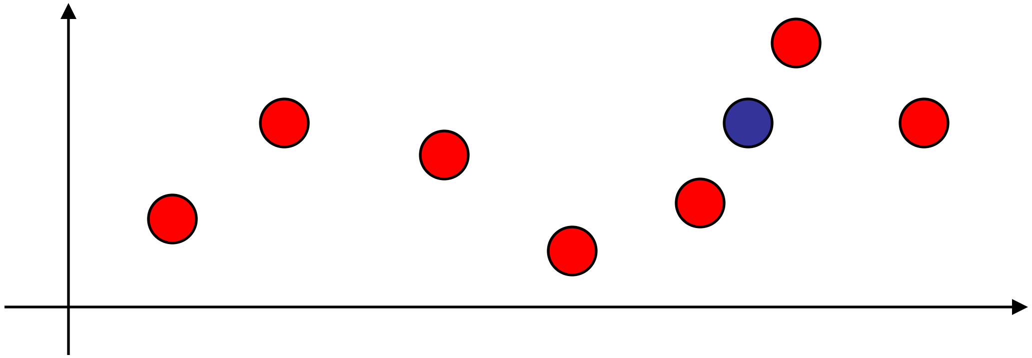
Why GPs for Regression?

Here are some data points! What function did they come from?



- I have *no idea*.

Oh. Okay. Uh, you think this point is likely in the function, too?



- I still have *no idea*.

Why GPs for Regression?

- You can't get anywhere without making some assumptions

- GPs are a nice way of expressing this 'prior on functions' idea.

- Can be used in many applications:
  - **Regression**
  - Classification
  - Optimization

Why GPs for Regression?

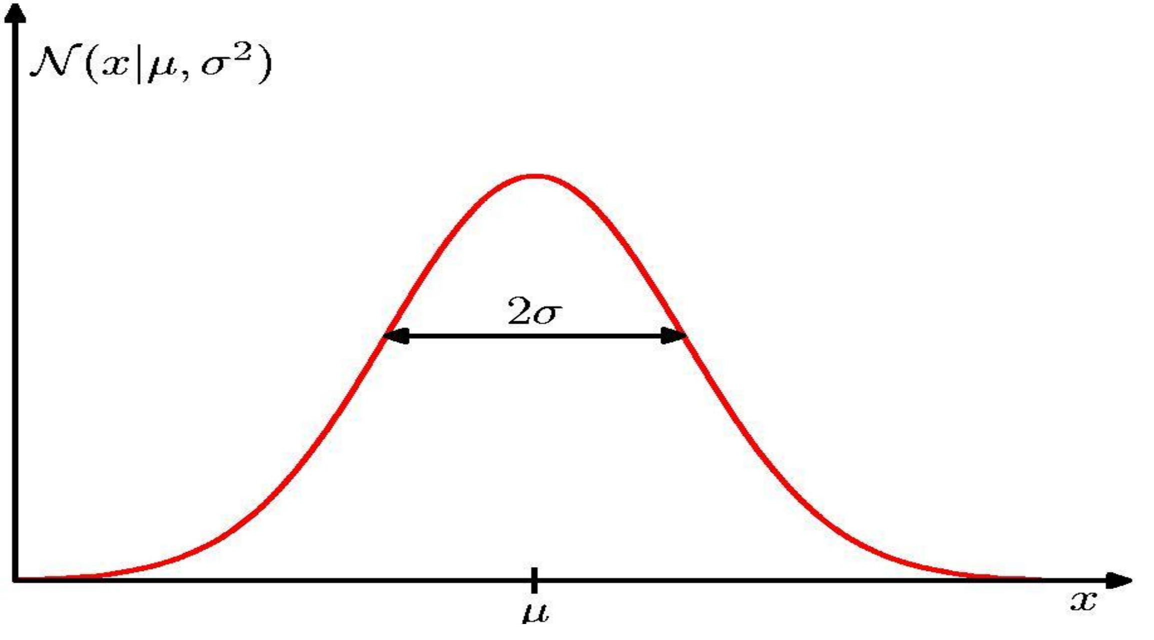
Under certain assumptions GPs can answer the following questions

- Here's where the function will **most likely be.** (expected function)
- Here are some **examples** of what it might look like. (sampling from the posterior distribution)
- Here is a prediction of what you'll see if you evaluate your function at x', **with confidence**

1D Gaussian Distribution

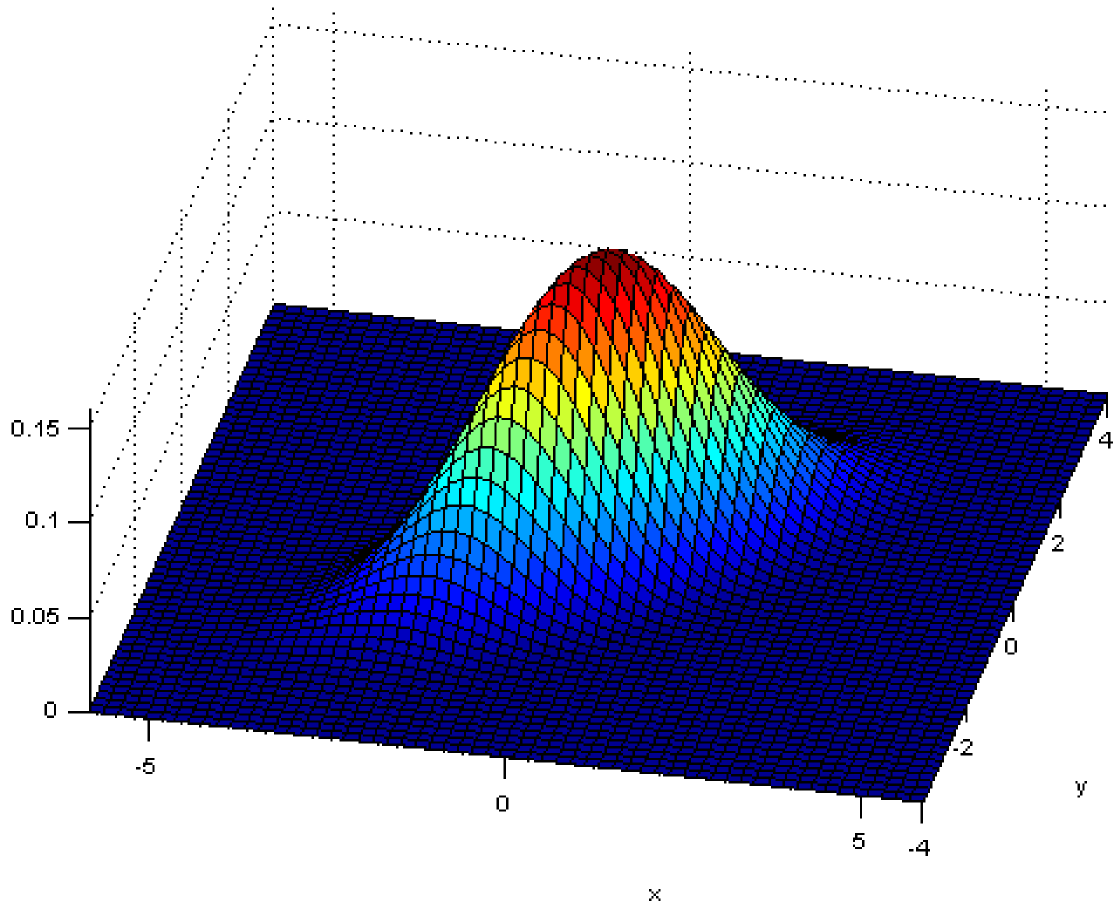
Parameters

- Mean, μ
- Variance, σ^2



$$P(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Multivariate Gaussian



$$P(\mathbf{x} \mid \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\{\frac{-1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})\}$$

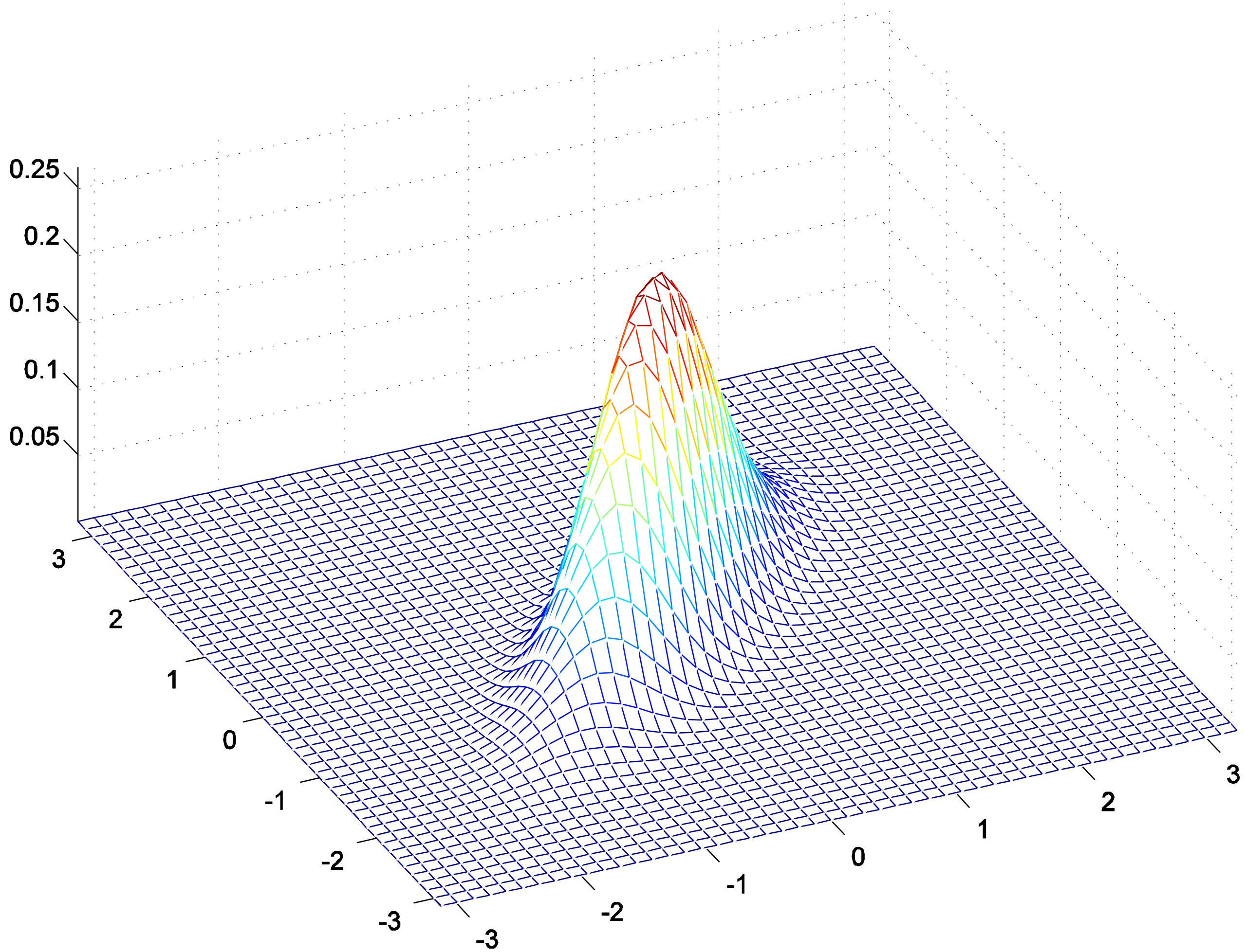
Multivariate Gaussian

- A 2-dimensional Gaussian is defined by
  - a mean vector $\mu = [\ \mu_1, \mu_2\ ]$
  - a covariance matrix: $$\Sigma = \begin{bmatrix} \sigma_{1,1}^2 & \sigma_{2,1}^2 \\ \sigma_{1,2}^2 & \sigma_{2,2}^2 \end{bmatrix}$$

  where $\sigma_{i,j}^2 = E[\ (x_i - \mu_i)\ (x_j - \mu_j)\ ]$
  is (co)variance

- Note: Σ is symmetric,
  "positive semi-definite": $\forall x$: $x^T \Sigma x \geq 0$

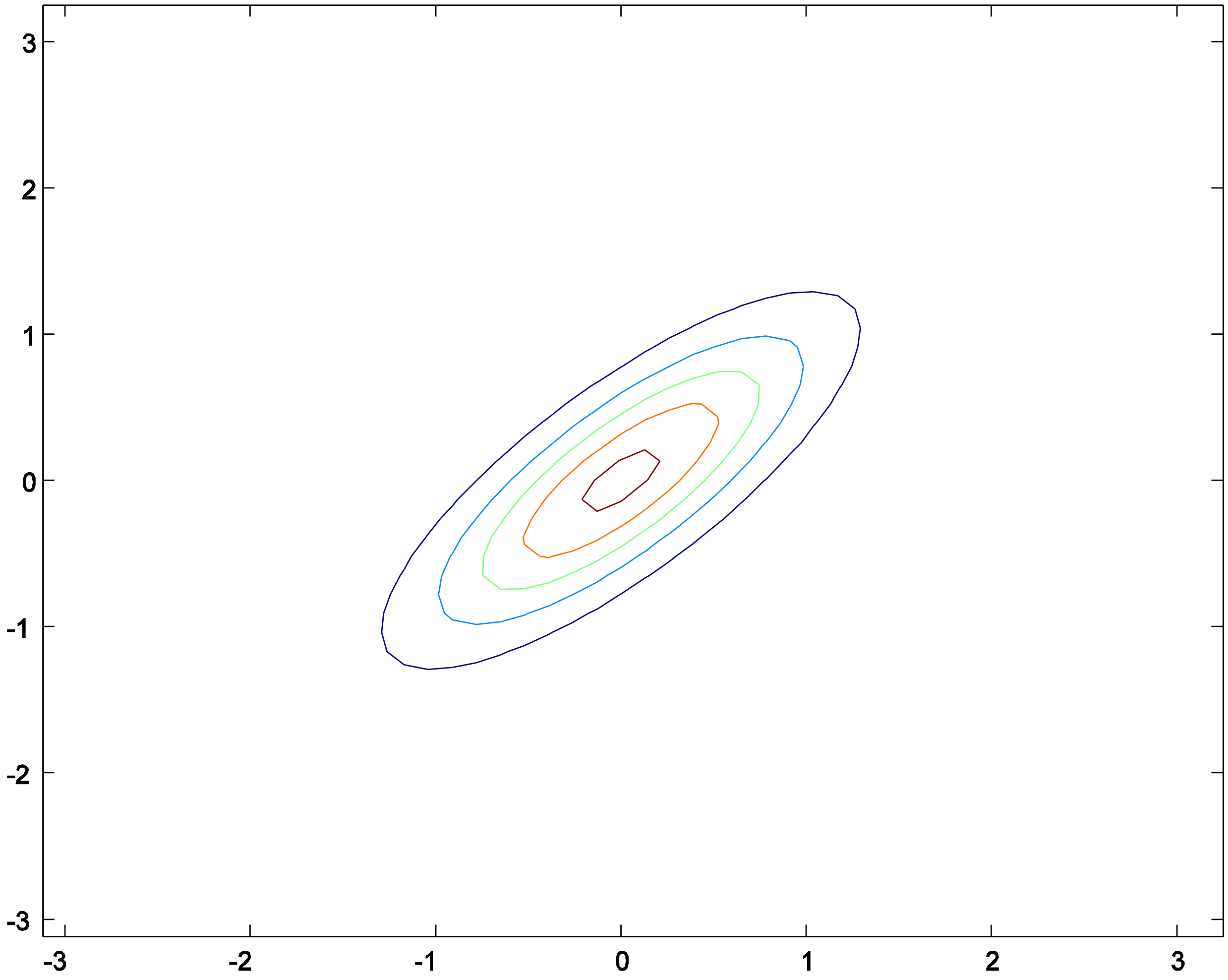
Multivariate Gaussian examples



μ = (0,0)

$$\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

Multivariate Gaussian examples



$$\mu = (0,0) \qquad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

Useful Properties of Gaussians

- Marginal distributions of Gaussians are Gaussian

- Given:

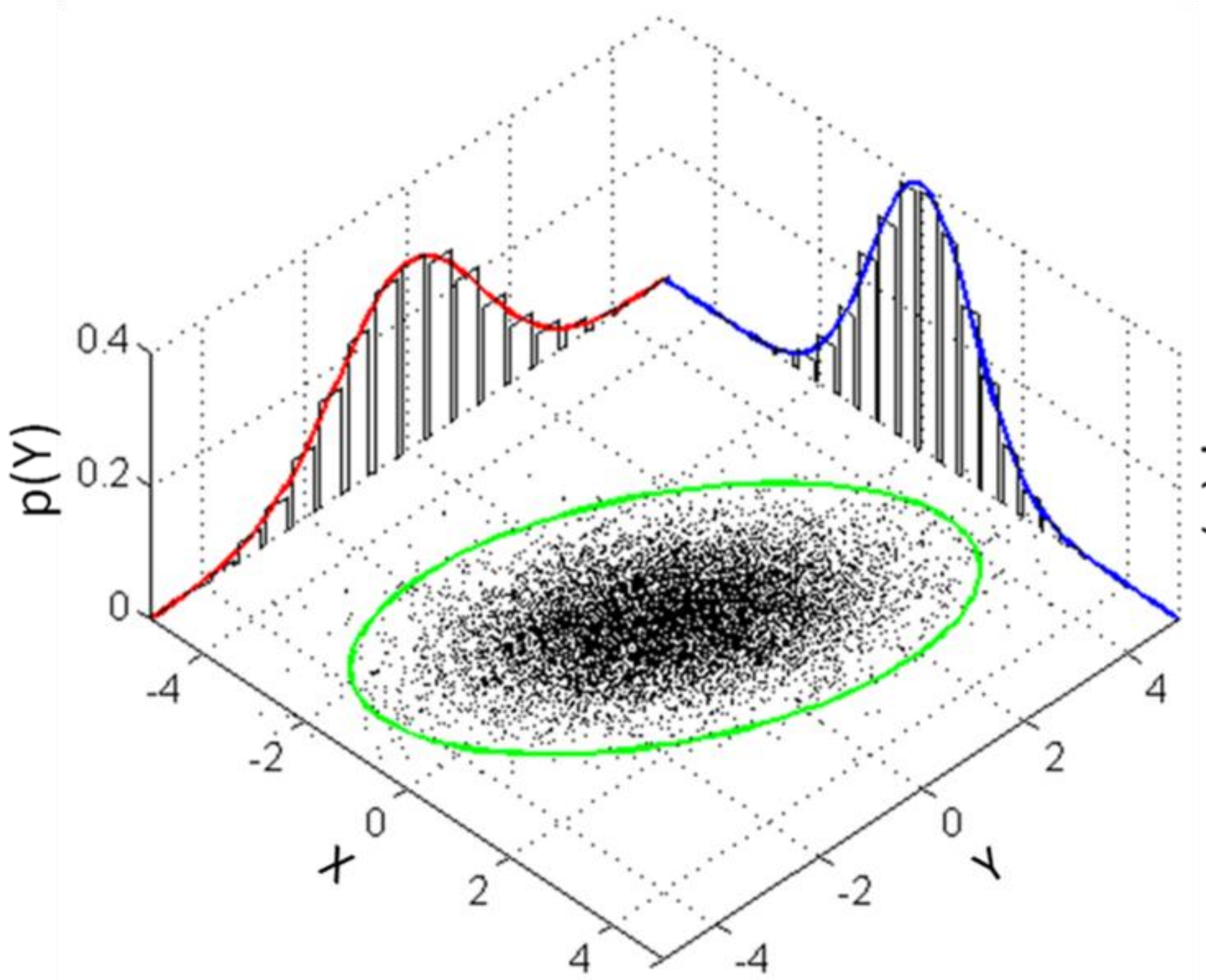
$$x = (x_a, x_b), \mu = (\mu_a, \mu_b)$$

$$\Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

- Marginal Distribution:

$$p(X_a) = \mathcal{N}(x_a \mid \mu_a, \Sigma_{aa})$$

Marginal distributions of Gaussians are Gaussian



Useful Properties of Gaussians

- Conditional distributions of Gaussians are Gaussian
- Notation:

$$\Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix} \qquad \Lambda = \Sigma^{-1} = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix}$$

- Conditional Distribution:

$$p(X_a|X_b) = \mathcal{N}(x_a \mid \mu_{a|b}, \Lambda_{aa}^{-1})$$

$$\mu_{a|b} = \mu_a - \Lambda_{aa}^{-1}\Lambda_{ab}(\mathbf{x}_b - \mu_a) = \mu_a - \Sigma_{ab}\Sigma_{bb}^{-1}(\mathbf{x}_b - \mu_a)$$

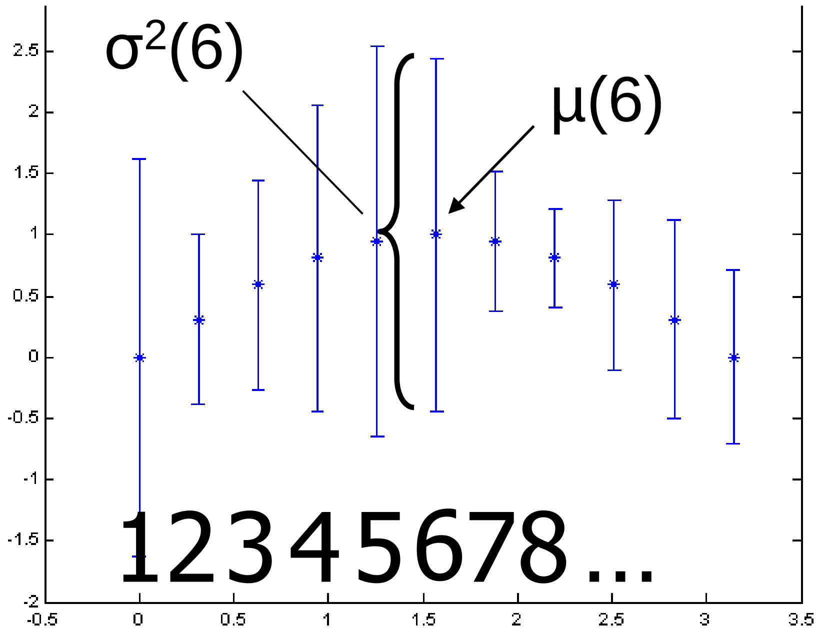
$$\Lambda_{aa}^{-1} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$$

Schur complement of Σ_{bb} in Σ

Higher Dimensions

- **Visualizing > 3 dimensions is… difficult**
- Means and marginals are practical, but then we don't see correlations between those variables
- Marginals are Gaussian, e.g., f(6) ~ N(μ(6), σ^2(6))

Visualizing a multivariate Gaussian f:

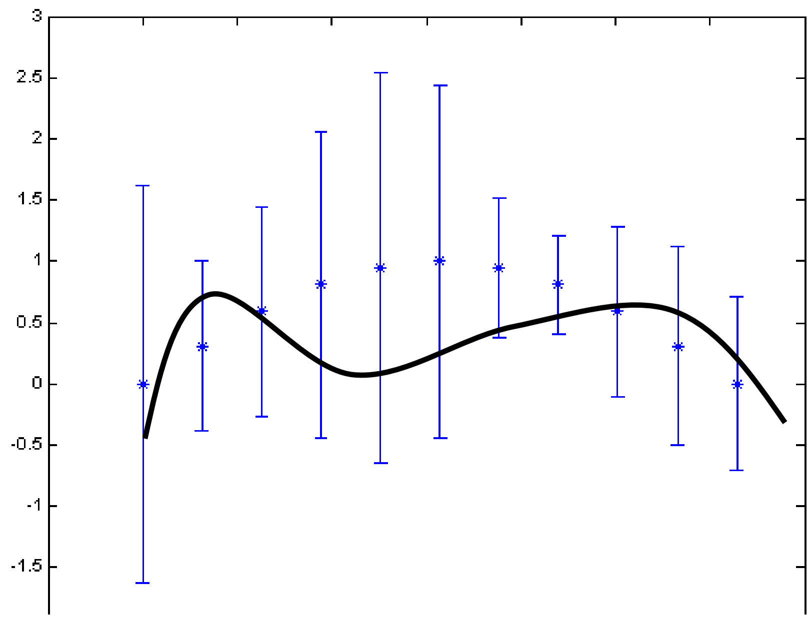


Yet Higher Dimensions

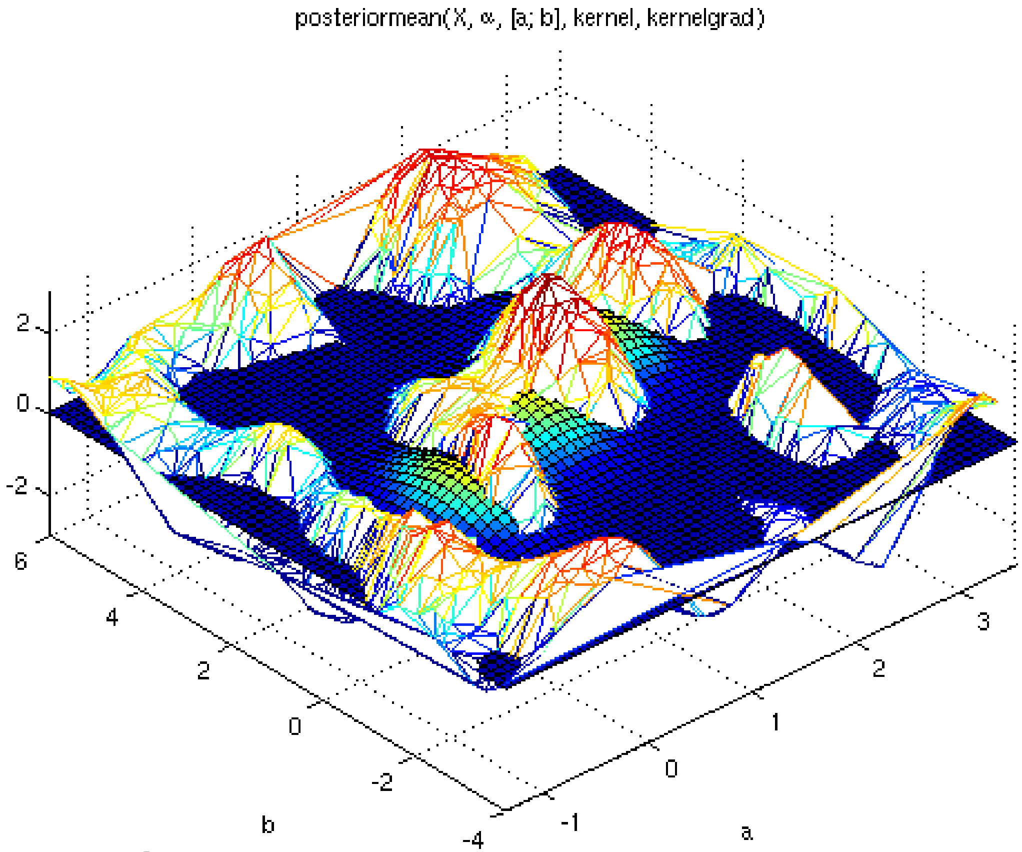
Why stop there?

- We indexed before with $\mathbb{Z}$, why not with $\mathbb{R}$?
- Need functions $\mu(x), k(x,z)$, $\forall x, z \in \mathbb{R}$
- x and z are indexes over the random variables
- f is now an uncountably infinite dimensional vector

Don't panic: It's just a function



Getting Ridiculous



Why stop there?

- We indexed before with $\mathbb{R}$, why not with $\mathbb{R}^D$?
- Need functions $\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{z})$, $\forall \mathbf{x}, \mathbf{z} \in \mathbb{R}^D$

Gaussian Process

Definition:

- Probability distribution *indexed by* an arbitrary set (integer, real, finite dimensional vector, etc)
- Each element gets a Gaussian distribution over the reals with mean $\mu(x)$
- These distributions are dependent/correlated as defined by $k(x,z)$
- Any finite subset of indices defines a multivariate Gaussian distribution

Gaussian Process

- Distribution over *functions*

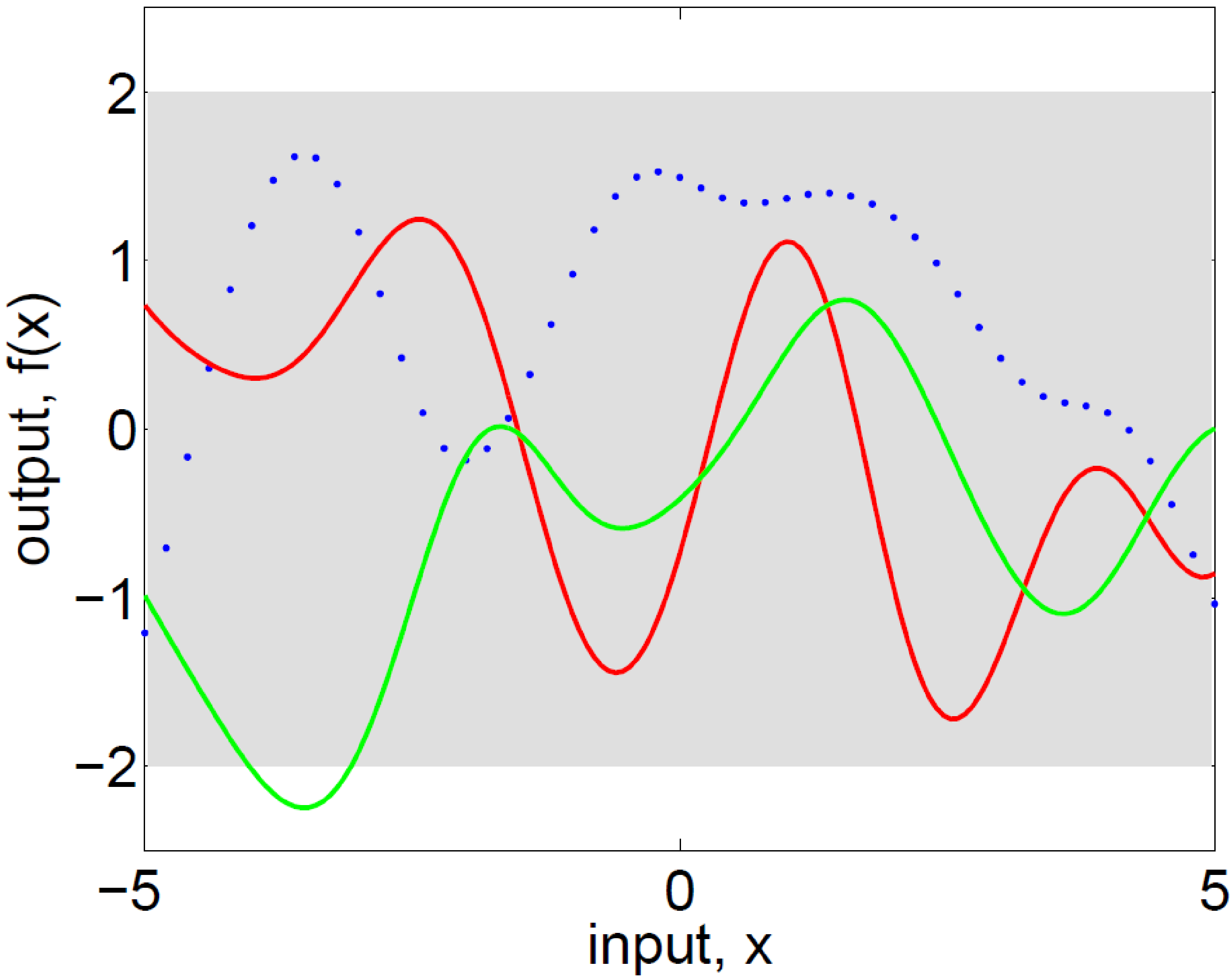
- Domain (index set) of the functions can be pretty much whatever
  - Reals
  - Real vectors
  - Graphs
  - Strings
  - Sets
  - …

- Most interesting structure is in k(x,z), the ‘kernel.’

Bayesian Updates for GPs

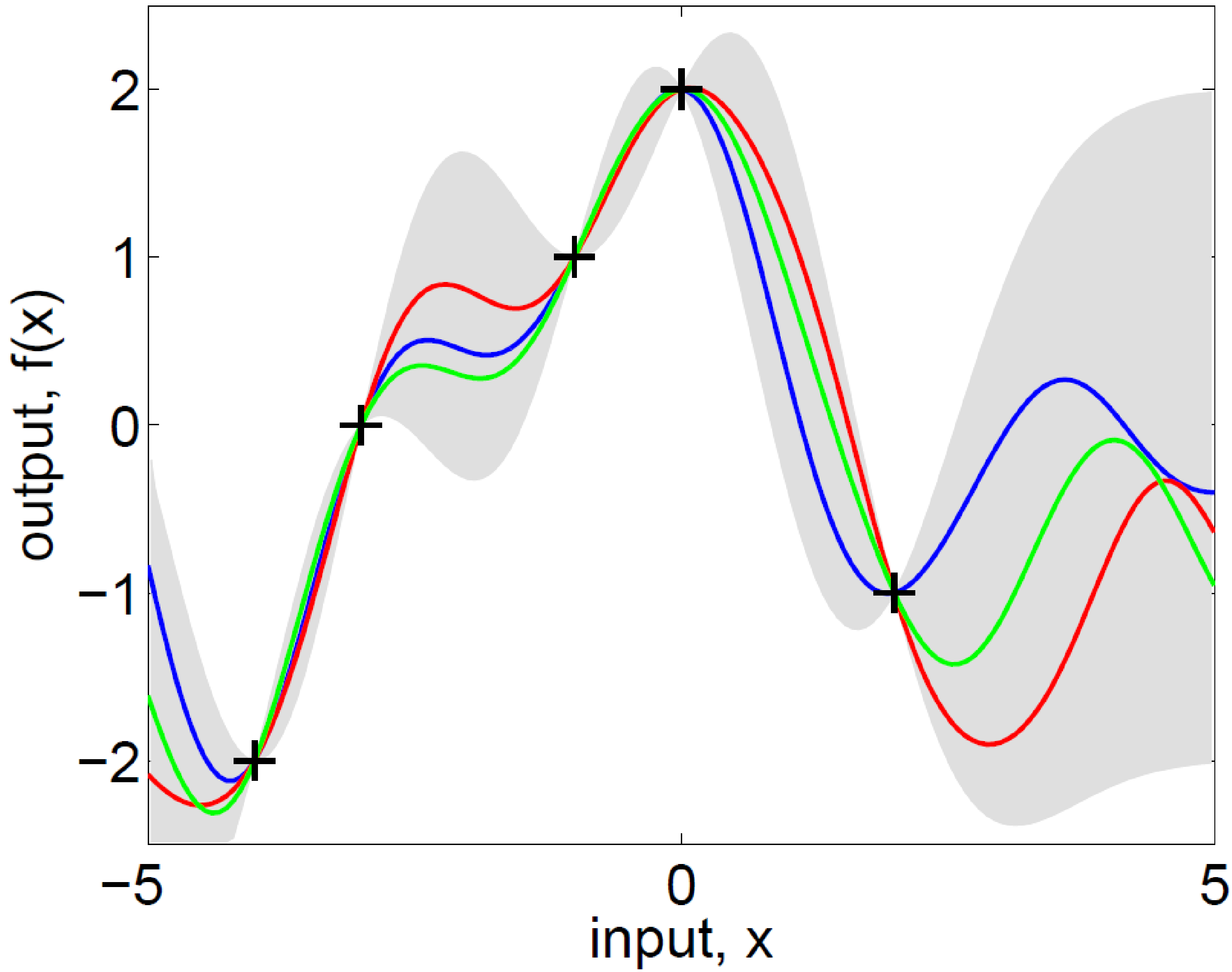
- How do Bayesians use a Gaussian Process?
  - Start with GP prior
  - Get some data
  - Compute a posterior
- Ask interesting questions about the posterior

Samples from the prior distribution



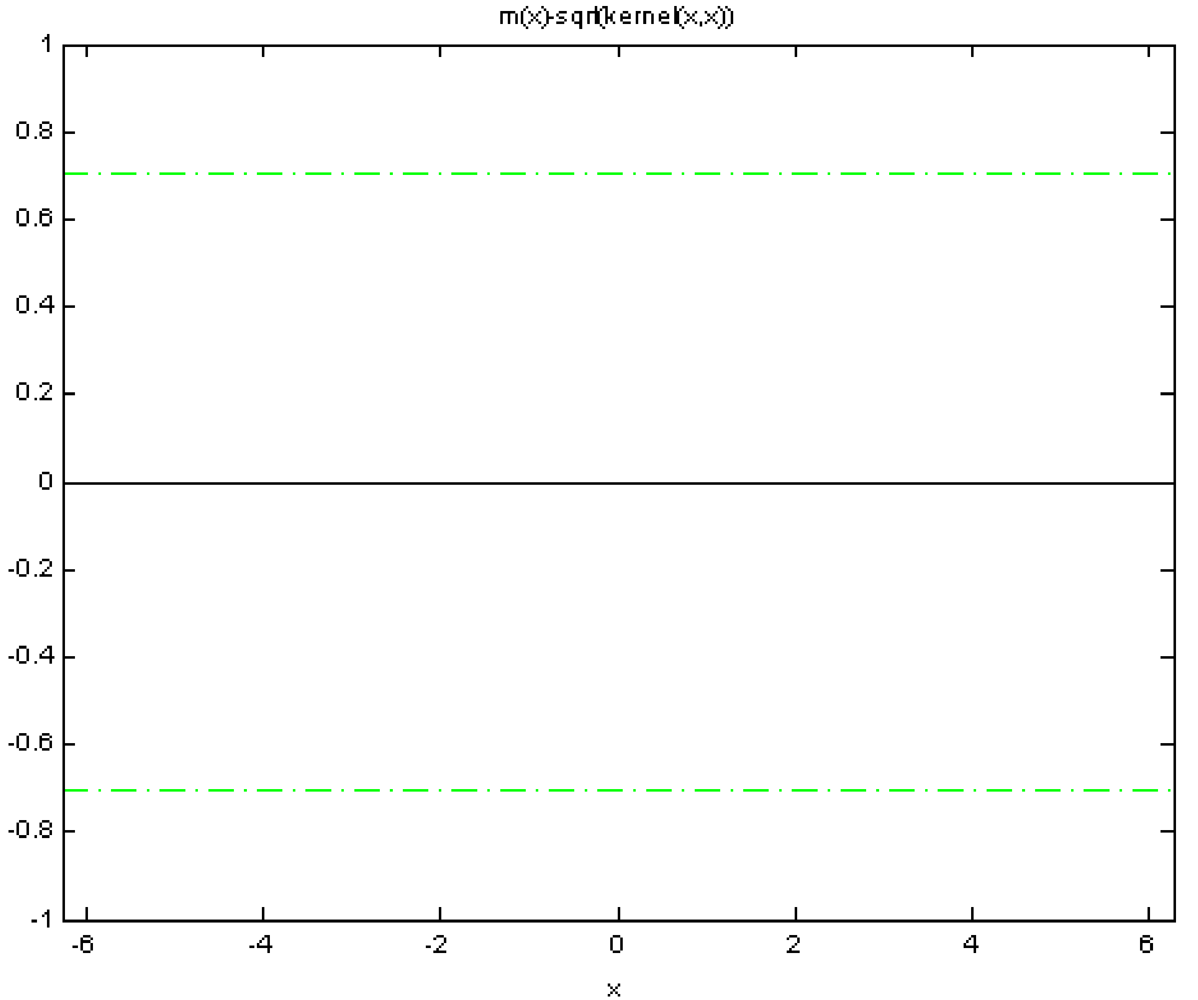
Picture is taken from Rasmussen and Williams

Samples from the posterior distribution

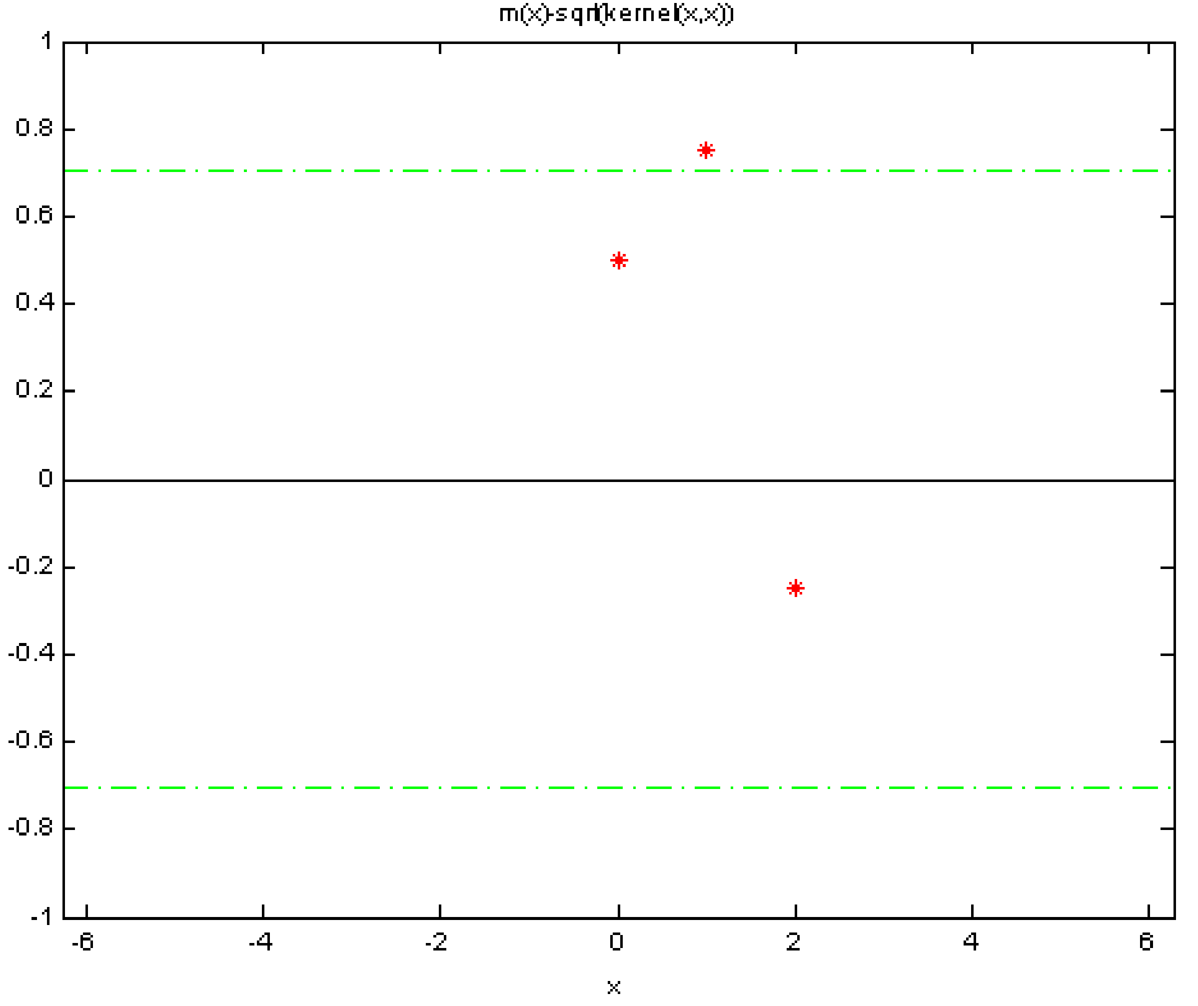


Picture is taken from Rasmussen and Williams

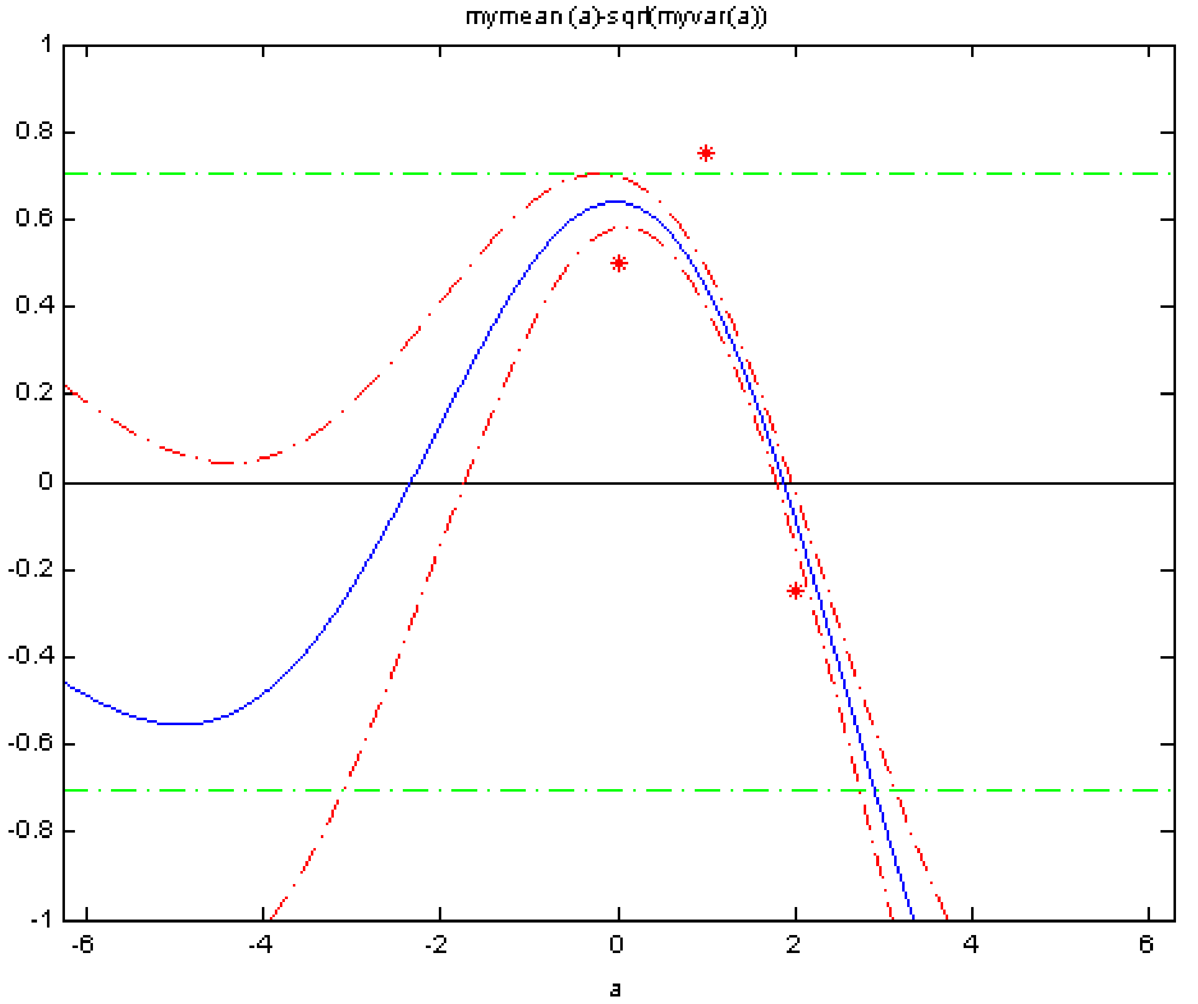
Prior



Data



Posterior



Contents

- Introduction
- Ridge Regression
- Gaussian Processes
  - Weight space view
    - Bayesian Ridge Regression + Kernel trick
  - Function space view
    - Prior distribution over functions
      + calculation posterior distributions

Ridge Regression

Training set: $D = \{(x_i, y_i) | i = 1, \ldots, n\}$

Linear regression: $f(x) = \langle \mathbf{w}, \phi(x) \rangle$

Ridge regression:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w} \in \mathcal{K}} \sum_{i=1}^{m} (y_i - \langle \underbrace{\phi(x_i)}_{\mathbf{x}_i}, \mathbf{w} \rangle)^2 + \lambda \|\mathbf{w}\|^2$$

The Gaussian Process is a Bayesian Generalization of the Ridge regression

Contents

- Introduction
- Ridge Regression
- Gaussian Processes
  - Weight space view
    - Bayesian Ridge Regression + Kernel trick
  - Function space view
    - Prior distribution over functions
      + calculation posterior distributions

Weight Space View

GP = Bayesian ridge regression in feature space
+ Kernel trick to carry out computations

Training set: $D = \{(\mathbf{x}_i, y_i) | i = 1, \ldots, n\}$

$$X = \begin{bmatrix} \mathbf{x}_1 \Big| & \ldots & \Big| \mathbf{x}_n \end{bmatrix} \in \mathbb{R}^{D \times n}, \text{ design matrix}$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n$$

The training data

Bayesian Analysis of Linear Regression with Gaussian noise

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} \in \mathbb{R},\ \mathbf{x}, \mathbf{w} \in \mathbb{R}^D$$

$$y = f(\mathbf{x}) + \epsilon = \mathbf{x}^T\mathbf{w} + \epsilon \in \mathbb{R}$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2) \in \mathbb{R}$$

(*Homoscedastic* noise, the same for all $\mathbf{x}$)

Let us calculate the likelihood:

$$P(\mathbf{y}|X, \mathbf{w}) = \prod_{i=1}^{n} P(y_i|\mathbf{x}_i^T\mathbf{w})$$

and then put $\mathbf{w} \sim \mathcal{N}_{\mathbf{w}}(0, \Sigma_p)$ prior over parameters $\mathbf{w}$.

Bayesian Analysis of Linear Regression with Gaussian noise

The likelihood:

$$
\begin{aligned}
P(\mathbf{y}|X,\mathbf{w}) &= \prod_{i=1}^{n} P(y_i|\mathbf{x}_i^T\mathbf{w}) \\
&= \prod_{i=1}^{n} \mathcal{N}_{y_i}(\mathbf{x}_i^T\mathbf{w},\sigma^2) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-(y_i-\mathbf{x}_i^T\mathbf{w})^2}{2\sigma^2}\right] \\
&= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left[\frac{-1}{2\sigma^2}\|\mathbf{y}-X^T\mathbf{w}\|^2\right] \\
&= \mathcal{N}_{\mathbf{y}}(X^T\mathbf{w},\sigma^2\mathbf{I}_n)
\end{aligned}
$$

Bayesian Analysis of Linear Regression with Gaussian noise

The prior:

$$\mathbf{w} \sim \mathcal{N}_{\mathbf{w}}(0, \Sigma_p)$$

Now, we can calculate the posterior:

$$\begin{aligned}
P(\mathbf{w}|X,\mathbf{y}) &= \frac{P(\mathbf{y}|X,\mathbf{w})P(\mathbf{w})}{P(\mathbf{y}|X)} \\
&= \frac{P(\mathbf{y}|X,\mathbf{w})P(\mathbf{w})}{\int P(\mathbf{y}|X,\mathbf{w})d\mathbf{w}} \\
&= \frac{\mathcal{N}_{\mathbf{y}}(X^T\mathbf{w},\sigma^2\mathbf{I}_n)\mathcal{N}_{\mathbf{w}}(0,\Sigma_p)}{\int \mathcal{N}_{\mathbf{y}}(X^T\mathbf{w},\sigma^2\mathbf{I}_n)\mathcal{N}_{\mathbf{w}}(0,\Sigma_p)\, d\mathbf{w}} \\
&\sim \mathcal{N}_{\mathbf{y}}(X^T\mathbf{w},\sigma^2\mathbf{I}_n)\mathcal{N}_{\mathbf{w}}(0,\Sigma_p)
\end{aligned}$$

Bayesian Analysis of Linear Regression with Gaussian noise

Ridge Regression

$$
\begin{aligned}
P(\mathbf{w}|X,\mathbf{y}) &\sim \mathcal{N}_{\mathbf{y}}(X^T\mathbf{w},\sigma^2\mathbf{I}_n)\mathcal{N}_{\mathbf{w}}(0,\Sigma_p) \\
&\sim \exp\{\frac{-1}{2\sigma^2}(\mathbf{y}-X^T\mathbf{w})^T(\mathbf{y}-X^T\mathbf{w})\}\exp\{\frac{-1}{2}\mathbf{w}^T\Sigma_p^{-1}\mathbf{w}\} \\
&\sim \exp\{\frac{-1}{2}(\mathbf{w}-\bar{\mathbf{w}})^T\underbrace{\left(\frac{1}{\sigma^2}XX^T+\Sigma_p^{-1}\right)}_{A}(\mathbf{w}-\bar{\mathbf{w}})\} \\
&\sim \boxed{\mathcal{N}_{\mathbf{w}}(\bar{\mathbf{w}},A^{-1})}
\end{aligned}
$$

After "completing the square"

where $\boxed{\bar{\mathbf{w}} \doteq \sigma^{-2}\underbrace{\left(\sigma^{-2}XX^T+\Sigma_p^{-1}\right)^{-1}}_{A^{-1}\in\mathbb{R}^{D\times D}}X\mathbf{y}\in\mathbb{R}^D}$

MAP estimation

$$\boxed{A \doteq \left(\sigma^{-2}XX^T+\Sigma_p^{-1}\right)\in\mathbb{R}^{D\times D}}$$

Bayesian Analysis of Linear Regression with Gaussian noise

We want to use $P(\mathbf{w}|X, \mathbf{y}) = N_{\mathbf{w}}(\bar{\mathbf{w}}, A^{-1})$ posterior for predicting f in a test point $\mathbf{x}_*$.

$$f_* \doteq f(\mathbf{x}_*) \qquad f(\mathbf{x}) = \mathbf{x}^T \mathbf{w} \in \mathbb{R}, \ \mathbf{x}, \mathbf{w} \in \mathbb{R}^D$$

$$y = f(\mathbf{x}) + \epsilon = \mathbf{x}^T \mathbf{w} + \epsilon \in \mathbb{R}$$

$$P(\underbrace{f_*}_{\mathbf{x}_*^T \mathbf{w}} | \mathbf{x}_*, X, \mathbf{y}) = \int \underbrace{P(f_* | \mathbf{x}_*, \mathbf{w})}_{\delta(f_*, \mathbf{x}_*^T \mathbf{w})} \underbrace{P(\mathbf{w} | \mathbf{y}, X)}_{\mathcal{N}_{\mathbf{w}}(\bar{\mathbf{w}}, A^{-1})} \, d\mathbf{w}$$

$$= \mathcal{N}_{f_*}(\mathbf{x}_*^T \bar{\mathbf{w}}, \mathbf{x}_*^T A^{-1} \mathbf{x}_*)$$

This posterior covariance matrix doesn't depend on the observations **y**,

A strange property of Gaussian Processes $\quad \mathbf{y}^T = [y_1, \ldots, y_n]$

Projections of Inputs into Feature Space

The reviewed Bayesian linear regression suffers from

limited expressiveness

$\Downarrow$

To overcome the problem $\Rightarrow$
go to a feature space and do linear regression there

a., **explicit** features $\phi(\mathbf{x}) = [x_1, x_1 x_2^2, x_1 - x_2, \ldots]^T$

b., **implicit** features (kernels) $k(\vec{x}, \vec{y}) = \exp(-\|\vec{x} - \vec{y}\|^2)$

Explicit Features

$$\phi(\mathbf{x}) = [x_1, x_1 x_2^2, x_1 - x_2, \ldots]^T \in \mathbb{R}^N$$

$$\phi(X) = \left[\phi(\mathbf{x}_1)\middle|\phi(\mathbf{x}_2)\middle| \quad \ldots \quad \middle|\phi(\mathbf{x}_n)\right] \in \mathbb{R}^{N \times n}$$

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w} \in \mathbb{R}, \qquad \phi(\mathbf{x}), \mathbf{w} \in \mathbb{R}^N$$

$$y = f(\mathbf{x}) + \epsilon = \phi(\mathbf{x})^T \mathbf{w} + \epsilon \in \mathbb{R}$$

Linear regression in the feature space

Explicit Features

The predictive distribution after feature map:

$$P(\underbrace{f_*}_{\phi(\mathbf{x}_*)^T\mathbf{w}}|\mathbf{x}_*, X, \mathbf{y}) = \mathcal{N}_{f_*}\left(\phi(x_*)^T\bar{\mathbf{w}}, \phi(x_*)^T A^{-1}\phi(x_*)\right)$$

where $\boxed{\bar{\mathbf{w}} \doteq \sigma^{-2}\underbrace{\left(\sigma^{-2}\phi(X)\phi(X)^T + \Sigma_p^{-1}\right)^{-1}}_{A^{-1}\in\mathbb{R}^{N\times N}}\phi(X)\mathbf{y} \in \mathbb{R}^D}$

$$A \doteq \boxed{\left(\sigma^{-2}\phi(X)\phi(X)^T + \Sigma_p^{-1}\right) \in \mathbb{R}^{N\times N}}$$

Explicit Features

Shorthands:

$$\phi_* \doteq \phi(\mathbf{x}_*) \in \mathbb{R}^N \qquad N = \text{dim of feature space}$$

$$\phi \doteq \phi(X) = \left[\phi(\mathbf{x}_1) \middle| \phi(\mathbf{x}_2) \middle| \quad \ldots \quad \middle| \phi(\mathbf{x}_n)\right] \in \mathbb{R}^{N \times n}$$

$$A \doteq \left(\sigma^{-2}\phi\phi^T + \Sigma_p^{-1}\right) \in \mathbb{R}^{N \times N}$$

$$\bar{\mathbf{w}} \doteq \sigma^{-2} \underbrace{\left(\sigma^{-2}\phi\phi^T + \Sigma_p^{-1}\right)^{-1}}_{A^{-1} \in \mathbb{R}^{N \times N}} \phi \mathbf{y} \in \mathbb{R}^N$$

The predictive distribution after feature map:

$$P(\underbrace{f_*}_{\phi_*^T \mathbf{w}} | \mathbf{x}_*, X, \mathbf{y}) = \mathcal{N}_{f_*}\left(\phi_*^T \bar{\mathbf{w}}, \phi_*^T A^{-1} \phi_*\right)$$

Explicit Features

The predictive distribution after feature map:

$$P(\underbrace{f_*}_{\phi_*^T \mathbf{w}} | \mathbf{x}_*, X, \mathbf{y}) = \mathcal{N}_{f_*}\left(\phi_*^T \bar{\mathbf{w}}, \phi_*^T A^{-1} \phi_*\right) \quad (*)$$

$$= \mathcal{N}_{f_*}\left(\sigma^{-2}\phi_*^T\left[\sigma^{-2}\phi\phi^T + \Sigma_p^{-1}\right]^{-1}\phi\mathbf{y}, \phi_*^T \overbrace{\left[\sigma^{-2}\phi\phi^T + \Sigma_p^{-1}\right]^{-1}}^{\mathbb{R}^{N\times N}}\phi_*\right)$$

A problem with (*) is that it needs an NxN matrix inversion...

Let $K \doteq \phi^T \Sigma_p \phi \in \mathbb{R}^{n\times n}$

(*) can be rewritten: $P(f_*|\mathbf{x}_*, X, \mathbf{y}) =$

$$\mathcal{N}_{f_*}\left(\underbrace{(\phi_*^T\Sigma_p\phi)}_{\mathbb{R}^{n\times n}}\underbrace{(K+\sigma^2\mathbf{I}_n)^{-1}}_{\mathbb{R}^{n\times n}}\mathbf{y}, \underbrace{(\phi_*^T\Sigma_p\phi_*)}_{\mathbb{R}^{n\times n}} - \underbrace{(\phi_*^T\Sigma_p\phi)}_{\mathbb{R}^{n\times n}}\underbrace{(K+\sigma^2\mathbf{I}_n)^{-1}}_{\mathbb{R}^{n\times n}}\underbrace{(\phi^T\Sigma_p\phi_*)}_{\mathbb{R}^{n\times n}}\right)$$

Proofs

- **Mean expression.** We need:

$$\sigma^{-2}\phi_*^T \underbrace{\left[\sigma^{-2}\phi\phi^T + \Sigma_p^{-1}\right]^{-1}}_{A^{-1}} \phi\mathbf{y} = (\phi_*^T \underbrace{\Sigma_p\phi)(K+\sigma^2\mathbf{I}_n)^{-1}}_{\sigma^{-2}A^{-1}\phi}\mathbf{y}$$

Lemma:

$$\sigma^{-2}\phi(K+\sigma^2\mathbf{I}_n) = \sigma^{-2}\phi(\phi^T\Sigma_p\phi + \sigma^2\mathbf{I}_n) = A\Sigma_p\phi$$

- **Variance expression.** We need:

$$\phi_*^T\left[\sigma^{-2}\phi\phi^T + \Sigma_p^{-1}\right]^{-1}\phi_* = (\phi_*^T\Sigma_p\phi_*) - (\phi_*^T\Sigma_p\phi)(K+\sigma^2\mathbf{I}_n)^{-1}(\phi^T\Sigma_p\phi_*)$$

Matrix inversion Lemma:

$$(\underbrace{U}_{\phi}\underbrace{W}_{\sigma^{-2}}\underbrace{V^T}_{\phi^T} + \underbrace{Z}_{\Sigma_p^{-1}})^{-1} = Z^{-1} - Z^{-1}U(W^{-1} + \underbrace{V^TZ^{-1}U}_{K})^{-1}V^TZ^{-1}$$

From Explicit to Implicit Features

$$P(f_*|\mathbf{x}_*, X, \mathbf{y}) =$$

$$\mathcal{N}_{f_*}\Big(\underbrace{(\phi_*^T\Sigma_p\phi)}_{\mathbb{R}^{n\times n}}\underbrace{(K+\sigma^2\mathbf{I}_n)^{-1}}_{\mathbb{R}^{n\times n}}\mathbf{y}, \underbrace{(\phi_*^T\Sigma_p\phi_*)}_{\mathbb{R}^{n\times n}} - \underbrace{(\phi_*^T\Sigma_p\phi)}_{\mathbb{R}^{n\times n}}\underbrace{(K+\sigma^2\mathbf{I}_n)^{-1}}_{\mathbb{R}^{n\times n}}\underbrace{(\phi^T\Sigma_p\phi_*)}_{\mathbb{R}^{n\times n}}\Big)$$

We have to work only with $n \times n$ matrices, and not by $N \times N$

The feature space always enters in the form of:

$(\phi_*^T\Sigma_p\phi_*)$, $(\phi_*^T\Sigma_p\phi)$, $(\phi^T\Sigma_p\phi)$, ($\in \mathbb{R}^{n\times n}$ matrices)

Let $k(x,\tilde{x}) \doteq \phi(x)^T\Sigma_p\phi(x)$

Lemma:

$k(x,\tilde{x})$ is an inner product in the feature space: $\psi(x) \doteq \Sigma_p^{1/2}\phi(x)$

Contents

- Introduction
- Ridge Regression
- Gaussian Processes
  - Weight space view
    - Bayesian Ridge Regression + Kernel trick
  - Function space view
    - Prior distribution over functions
      + calculation posterior distributions

Function Space View

- An alternative way to get the previous results

- Inference directly in function space

Definition: (Gaussian Processes)

GP is a collection of random variables, s.t. any finite number of them have a joint Gaussian distribution

Function Space View

Notations:

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \tilde{\mathbf{x}})) \in \mathbb{R},\ \mathbf{x} \in \mathbb{R}^D$$

$$m(\mathbf{x}) = \mathbb{E}[f(x)] \in \mathbb{R},\ \text{(mean function)}$$

$$k(\mathbf{x}, \tilde{\mathbf{x}}) = \mathbb{E}[(f(x) - m(\mathbf{x}))(f(\tilde{\mathbf{x}}) - m(\tilde{\mathbf{x}}))^T] \in \mathbb{R}$$

(covariance function)

GP is **completely specified** by its

mean function $m(\mathbf{x})$, and

covariance function $k(\mathbf{x}, \tilde{\mathbf{x}})$

Function Space View

Gaussian Processes:

For each $\mathbf{x} \in \mathbb{R}^D$ we associate a Gaussian variable $f(\mathbf{x})$ such that $\mathbb{R} \ni f(\mathbf{x}) \sim \mathcal{N}_{f(\mathbf{x})}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}))$, and its correalation with other $f(\tilde{\mathbf{x}})$ variables is $k(\mathbf{x}, \tilde{\mathbf{x}})$.

$$\mathbb{R} \ni f(\mathbf{x}) \sim \mathcal{N}_{f(\mathbf{x})}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}))$$

$$\begin{bmatrix} f(\mathbf{x}) \\ f(\tilde{\mathbf{x}}) \end{bmatrix} \sim \mathcal{N}_{\begin{bmatrix} f(\mathbf{x}) \\ f(\tilde{\mathbf{x}}) \end{bmatrix}} \left\{ \begin{bmatrix} m(\mathbf{x}) \\ m(\tilde{\mathbf{x}}) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}, \mathbf{x}) & k(\tilde{\mathbf{x}}, \mathbf{x}) \\ k(\mathbf{x}, \tilde{\mathbf{x}}) & k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) \end{bmatrix} \right\}$$

Function Space View

The Bayesian linear regression is an example of GP

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w} \in \mathbb{R},\ \phi(\mathbf{x}), \mathbf{w} \in \mathbb{R}^N \quad \mathbf{w} \sim \mathcal{N}_{\mathbf{w}}(0, \Sigma_p)$$

$\Rightarrow [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_k)]$ are jointly Gaussian $\forall \mathbf{x}_1, \ldots, \mathbf{x}_k$
thus f is GP.

$$\mathbb{E}[f(\mathbf{x})] = \phi(\mathbf{x})^T \mathbb{E}[\mathbf{w}] = 0 \Rightarrow m(\mathbf{x}) = 0$$

$$\mathbb{E}[f(x) f(\tilde{\mathbf{x}})^T] = \phi(\mathbf{x})^T \underbrace{\mathbb{E}[\mathbf{w}\mathbf{w}^T]}_{\Sigma_p} \phi(\tilde{\mathbf{x}}) = k(\mathbf{x}, \tilde{\mathbf{x}})$$

Function Space View

Special case

$$\left.\begin{array}{l} m(\mathbf{x}) = 0 \\ k(\mathbf{x}, \tilde{\mathbf{x}}) = \exp\left(-\frac{1}{2}\|\mathbf{x} - \tilde{\mathbf{x}}\|^2\right) \end{array}\right\} \Rightarrow f \text{ GP is given}$$

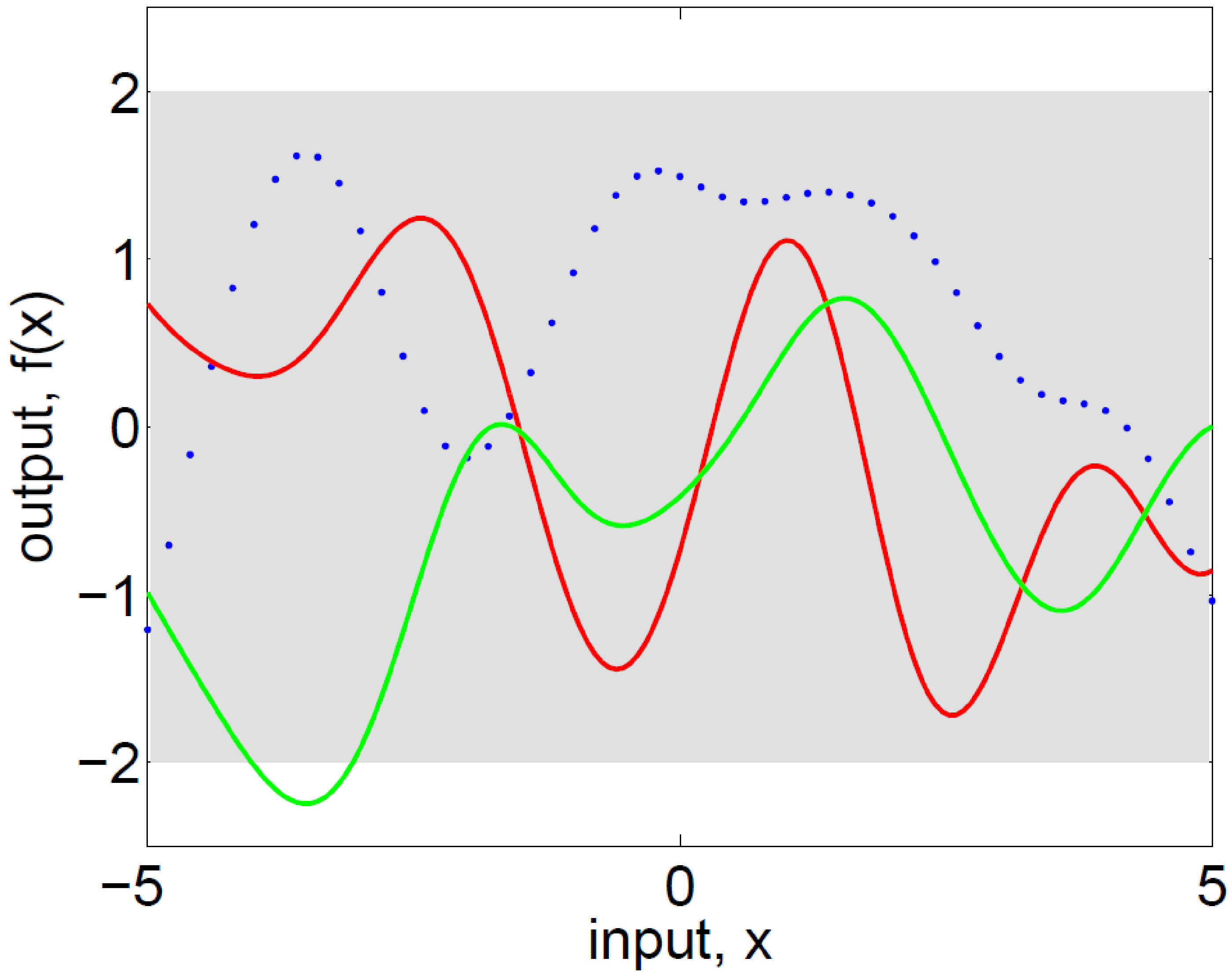
$\Rightarrow$ implies a distribution over functions.

$$\text{Let } X_* = \begin{bmatrix} \mathbf{x}_{*1}^T \\ \vdots \\ \mathbf{x}_{*m}^T \end{bmatrix} \; m \text{ input points}$$

$$\Rightarrow \mathbb{R}^m \ni f_* \sim \mathcal{N}_{f_*}(\underbrace{\mathbf{0}}_{\in \mathbb{R}^m}, \underbrace{k(X_*, X_*)}_{\in \mathbb{R}^{m \times m}})$$

At arbitray $\mathbf{x}_{*1}, \ldots, \mathbf{x}_{*m}$ places, we can generate m points from f (denoted by f_*) and plot them .

Function Space View



Picture is taken from Rasmussen and Williams

Function Space View

Observation

The plotted $f(\mathbf{x}_{*1}), \ldots, f(\mathbf{x}_{*m})$ function looks smooth.

Explanation

$$k(\mathbf{x}, \tilde{\mathbf{x}}) = \exp\left(-\frac{1}{2}\|\mathbf{x} - \tilde{\mathbf{x}}\|^2\right)$$

Thus if $\|\mathbf{x}_{*i} - \mathbf{x}_{*j}\|$ is small, then $corr(f(\mathbf{x}_{*i}), f(\mathbf{x}_{*j}))$ is high.

Prediction with noise free observations

Training set: $D = \{(\mathbf{x}_i, f_i) | i = 1, \ldots, n\}$

noise free observations

$$X = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} \in \mathbb{R}^{n \times D},\ m \text{ training inputs}$$

$$X_* = \begin{bmatrix} \mathbf{x}_{*1}^T \\ \vdots \\ \mathbf{x}_{*m}^T \end{bmatrix} \in \mathbb{R}^{m \times D},\ m \text{ test inputs}$$

$$f = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} \in \mathbb{R}^{n},\ n \text{ training targets}$$

$$f_* = \begin{bmatrix} f_{*1} \\ \vdots \\ f_{*m} \end{bmatrix} \in \mathbb{R}^{m},\ m \text{ test targets}$$

Prediction with noise free observations

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}_{\begin{bmatrix} f \\ f_* \end{bmatrix}} \left\{ \begin{bmatrix} \mathbf{0}_n \\ \mathbf{0}_m \end{bmatrix}, \underbrace{\begin{bmatrix} k(X,X) & k(X,X_*) \\ k(X_*,X) & k(X_*,X_*) \end{bmatrix}}_{\in \mathbb{R}^{(m+n)\times(m+n)}} \right\}$$

Goal:

We want to calculate the posterior distribution $f_*|X_*, X, f$

Prediction with noise free observations

Lemma:

$$P(f_*|X_*, X, f) =$$

$$\mathcal{N}_{f_*}\left(k(X_*, X)k(X, X)^{-1}f, k(X_*, X_*) - k(X_*, X)k(X, X)^{-1}k(X, X_*)\right)$$

Proofs: a bit of calculation using the joint (n+m) dim density

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}_{\begin{bmatrix} f \\ f_* \end{bmatrix}} \left\{ \begin{bmatrix} \mathbf{0}_n \\ \mathbf{0}_m \end{bmatrix}, \begin{bmatrix} k(X, X) & k(X, X_*) \\ k(X_*, X) & k(X_*, X_*) \end{bmatrix} \right\}$$

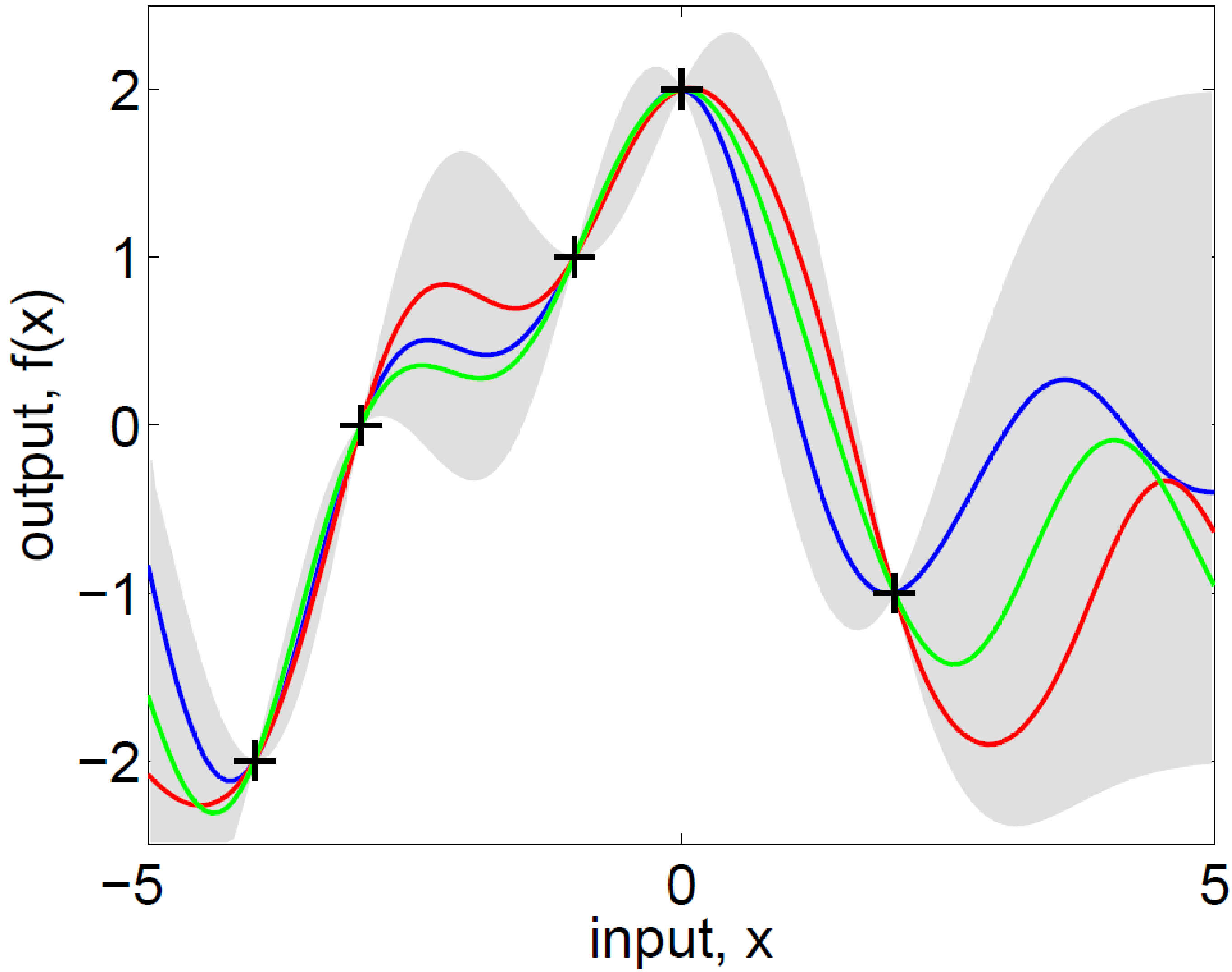
Remarks:

- If $X_* = X \Rightarrow f_* = f$ and the cov is 0. (nosie free observations)
- $P(f_*|X_*, X, f)$ is similar to the previous results:

$$P(f_*|\mathbf{x}_*, X, f) =$$

$$\mathcal{N}_{f_*}\left((\phi_*^T \Sigma_p \phi)(K + \sigma^2 \mathbf{I}_n)^{-1} f, (\phi_*^T \Sigma_p \phi_*) - (\phi_*^T \Sigma_p \phi)(K + \sigma^2 \mathbf{I}_n)^{-1}(\phi^T \Sigma_p \phi_*)\right)$$

Prediction with noise free observations



Picture is taken from Rasmussen and Williams

Prediction using noisy observations

$$y = f(\mathbf{x}) + \epsilon \in \mathbb{R}$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2) \in \mathbb{R}$$

(Homoscedastic noise, the same for all $\mathbf{x}$)

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}_{\begin{bmatrix} f \\ f_* \end{bmatrix}} \left\{ \begin{bmatrix} \mathbf{0}_n \\ \mathbf{0}_m \end{bmatrix}, \begin{bmatrix} k(X,X) & k(X,X_*) \\ k(X_*,X) & k(X_*,X_*) \end{bmatrix} \right\}$$

$$\Rightarrow cov(y_p, y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma^2 \delta_{p,q}$$

$$\Rightarrow cov([y_1, \ldots, y_n]) = k(X,X) + \sigma^2 \mathbf{I}_n \in \mathbb{R}^{n \times n}$$

The joint distribution:

$$\Rightarrow \begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}_{\begin{bmatrix} y \\ f_* \end{bmatrix}} \left\{ \begin{bmatrix} \mathbf{0}_n \\ \mathbf{0}_m \end{bmatrix}, \begin{bmatrix} k(X,X) + \sigma^2 \mathbf{I}_n & k(X,X_*) \\ k(X_*,X) & k(X_*,X_*) \end{bmatrix} \right\}$$

Prediction using noisy observations

The posterior for the noisy observations:

$$P(f_*|X,\mathbf{y},X_*) = \mathcal{N}_{f_*}(\bar{f}_*, cov(f_*))$$

where

$$\bar{f}_* = \mathbb{E}[f_*|X,\mathbf{y},X_*] = k(X_*,X)[k(X,X)+\sigma^2 I_n]^{-1}\mathbf{y} \in \mathbb{R}^m$$

$$cov(f_*) = k(X_*,X_*) - k(X_*,X)[k(X,X)+\sigma^2 I_n]^{-1}K(X,X_*) \in \mathbb{R}^{m\times m}$$

In the weight space view we had:

$$\bar{f}_* = (\phi_*^T \Sigma_p \phi)(\phi^T \Sigma_p \phi + \sigma^2 \mathbf{I}_n)^{-1}\mathbf{y}$$

$$cov(f_*) = (\phi_*^T \Sigma_p \phi_*) - (\phi_*^T \Sigma_p \phi)(\phi^T \Sigma \phi + \sigma^2 \mathbf{I}_n)^{-1}(\phi^T \Sigma_p \phi_*)$$

If $k(\mathbf{x},\tilde{\mathbf{x}}) = \phi(x)^T \Sigma_p \phi(\tilde{x})$, then they are the same.

Prediction using noisy observations

Short notations:

$$K = k(X, X) \in \mathbb{R}^{n \times n}$$

$$K_* = k(X, X_*) \in \mathbb{R}^{n \times m}$$

$$k(\mathbf{x}_*) = k_* = k(X, \mathbf{x}_*) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_*) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}_*) \end{bmatrix} \in \mathbb{R}^n$$

$\Rightarrow$ for a single test point $\mathbf{x}_*$:

$$\bar{f}_* = \underbrace{k_*^T}_{\mathbb{R}^{1 \times n}} \underbrace{[K + \sigma^2 I_n]^{-1}}_{\mathbb{R}^{n \times n}} \underbrace{\mathbf{y}}_{\mathbb{R}^n} \in \mathbb{R}$$

$$cov(f_*) = \underbrace{k(\mathbf{x}_*, \mathbf{x}_*)}_{\mathbb{R}} - \underbrace{k_*^T}_{\mathbb{R}^{1 \times n}} \underbrace{[K + \sigma^2 I_n]^{-1}}_{\mathbb{R}^{n \times n}} \underbrace{k_*}_{\mathbb{R}^n} \in \mathbb{R}$$

Prediction using noisy observations

$$\bar{f}_* = \underbrace{k_*^T}_{\mathbb{R}^{1\times n}} \underbrace{[K + \sigma^2 I_n]^{-1}}_{\mathbb{R}^{n\times n}} \underbrace{\mathbf{y}}_{\mathbb{R}^n} \in \mathbb{R}$$

Two ways to look at it:

- **Linear predictor**

$$\bar{f}_* = \boldsymbol{\beta}^T \mathbf{y} = \beta_1 y_1 + \ldots + \beta_n y_n$$

where $\boldsymbol{\beta}^T = k_*^T [K + \sigma^2 I_n]^{-1} \in \mathbb{R}1 \times n$

- **Manifestation of the Representer Theorem**

$$\bar{f}_* = \boldsymbol{\alpha}^T k_* = \alpha_1 k(\mathbf{x}_1, \mathbf{x}_*) + \ldots + \alpha_n k(\mathbf{x}_n, \mathbf{x}_*)$$

where $\boldsymbol{\alpha} = [K + \sigma^2 I_n]^{-1} \mathbf{y}$

$\bar{f}_*$ is a linear combination of n kernel values.

Prediction using noisy observations

$$\bar{f}_* = \underbrace{k_*^T}_{\mathbb{R}^{1\times n}} \underbrace{[K + \sigma^2 I_n]^{-1}}_{\mathbb{R}^{n\times n}} \underbrace{\mathbf{y}}_{\mathbb{R}^n} \in \mathbb{R}$$

Remarks:

- While the GP in general is quite complex, for the prediction of $\bar{f}_* = f(\mathbf{x}_*)$ we need only the (n+1) dimensional joint Gaussian distibution of $[y_1, \ldots, y_n, f(\mathbf{x}_*)]$

- The posterior covariance of

$$cov(f_*|X, \mathbf{y}, X_*) = k(X_*, X_*) - k(X_*, X)[k(X, X) + \sigma^2 I_n]^{-1} K(X, X_*)$$

does not depend on the observed targets $\mathbf{y}$.
This is a peculiarity of GP.

GP pseudo code

Inputs:

$$X = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} \in \mathbb{R}^{n \times D}, \; n \text{ training inputs}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^{n}, \; n \text{ training targets}$$

$k(\cdot,\cdot) : \mathbb{R}^{D \times D} \rightarrow \mathbb{R}$ covariance function (kernel)

$\mathbf{x}_*$ test input

σ^2 noise level on the observations

$$[y(\mathbf{x}) = f(\mathbf{x}) + \epsilon, \; \epsilon \sim \mathcal{N}(0, \sigma^2)]$$

GP pseudo code (continued)

1., $K \in \mathbb{R}^{n \times n}$ Gram matrix. $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$

$$k(\mathbf{x}_*) = k_* = k(X, \mathbf{x}_*) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_*) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}_*) \end{bmatrix} \in \mathbb{R}^n$$

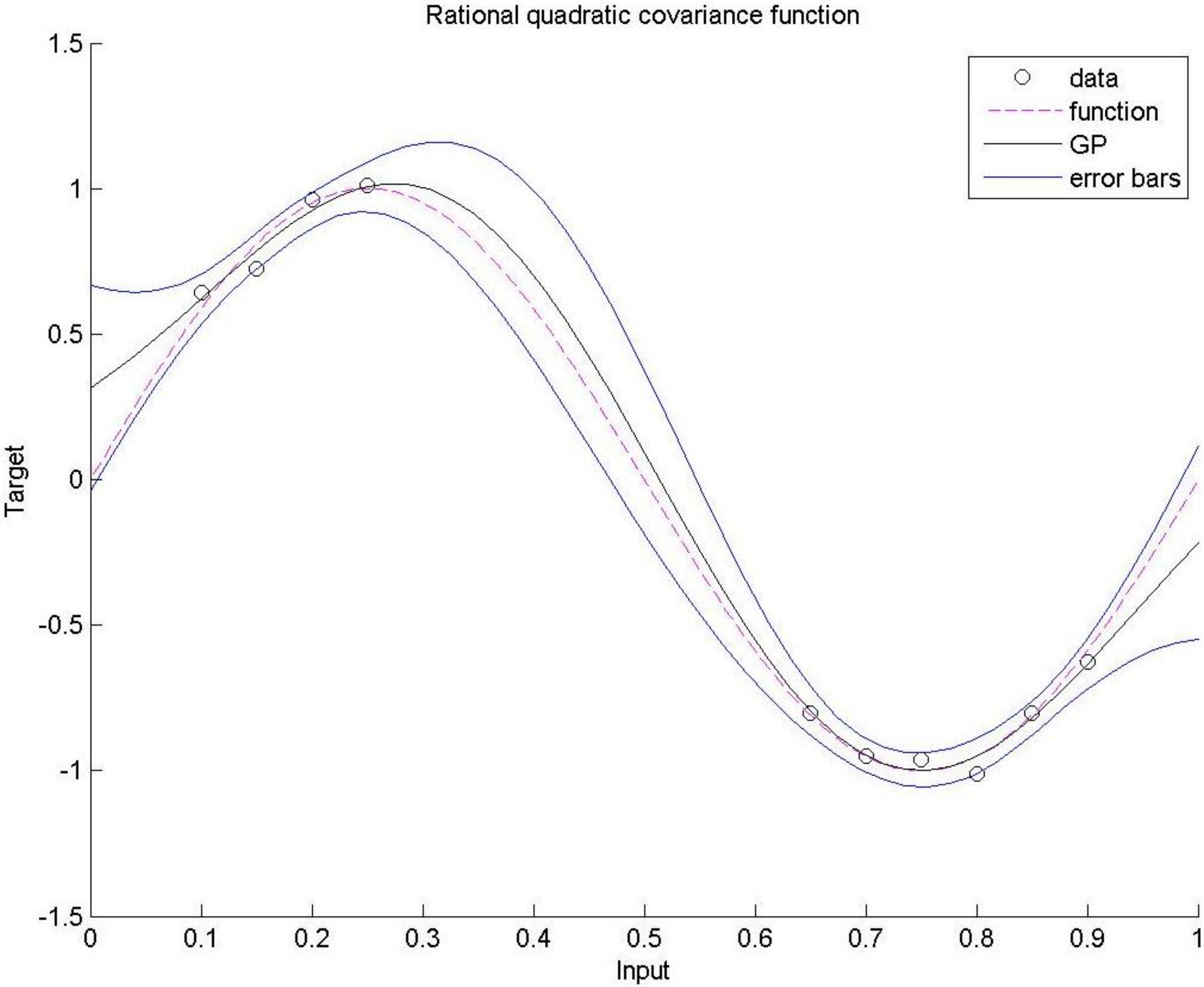
2., $\boldsymbol{\alpha} = (K + \sigma^2 \mathbf{I}_n)^{-1} \mathbf{y}$

3., $\bar{f}_* = k_*^T \boldsymbol{\alpha} \in \mathbb{R}$

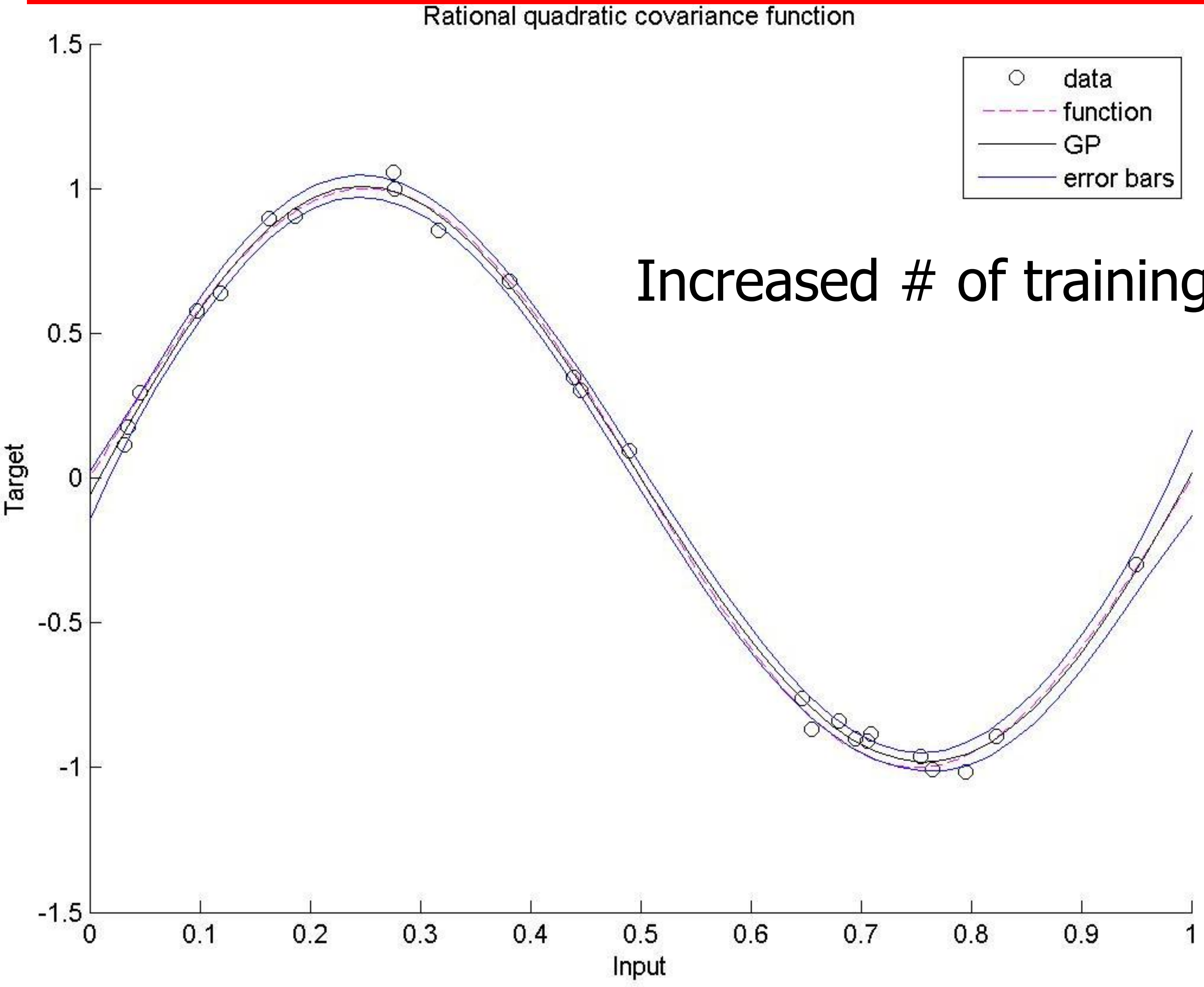
4., $cov(f_*) = \underbrace{k(\mathbf{x}_*, \mathbf{x}_*)}_{\mathbb{R}} - \underbrace{k_*^T}_{\mathbb{R}^{1 \times n}} \underbrace{[K + \sigma^2 I_n]^{-1}}_{\mathbb{R}^{n \times n}} \underbrace{k_*}_{\mathbb{R}^n} \in \mathbb{R}$

Outputs: $\bar{f}_*,\ cov(f_*)$

Results using Netlab , Sin function

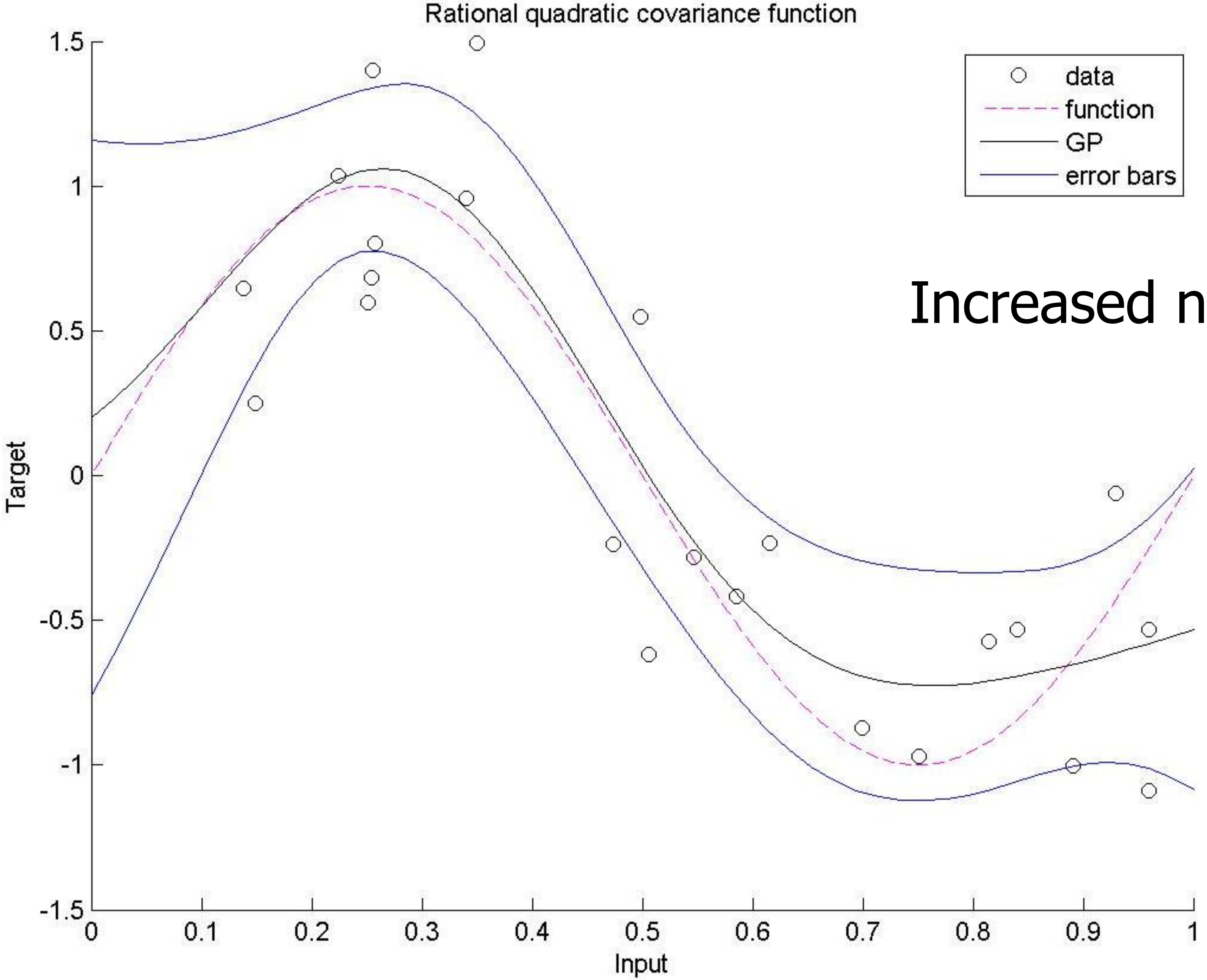


Results using Netlab, Sin function



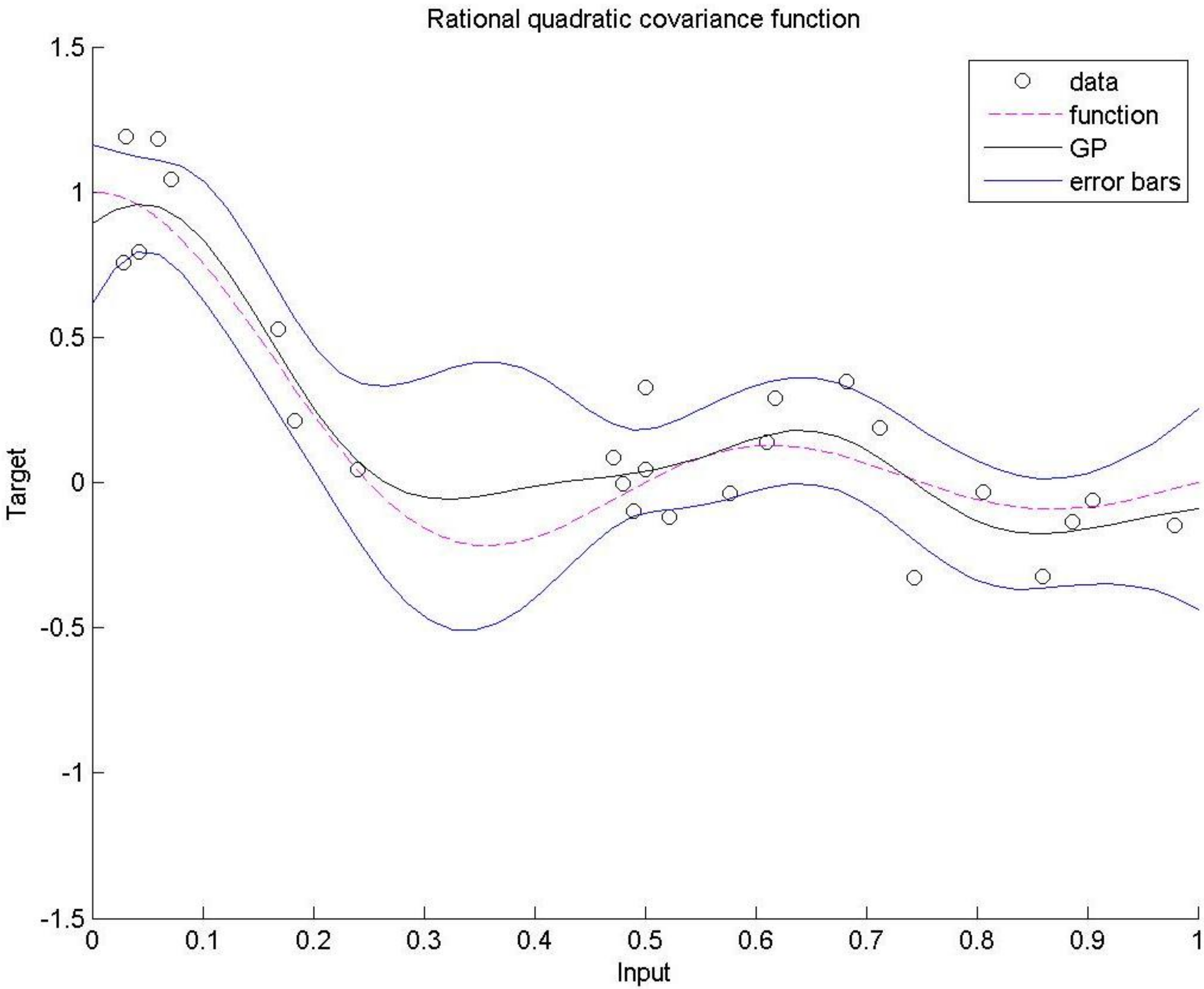
Increased # of training points

Results using Netlab, Sin function



Increased noise

Results using Netlab, Sinc function



Thanks for the Attention! ☺