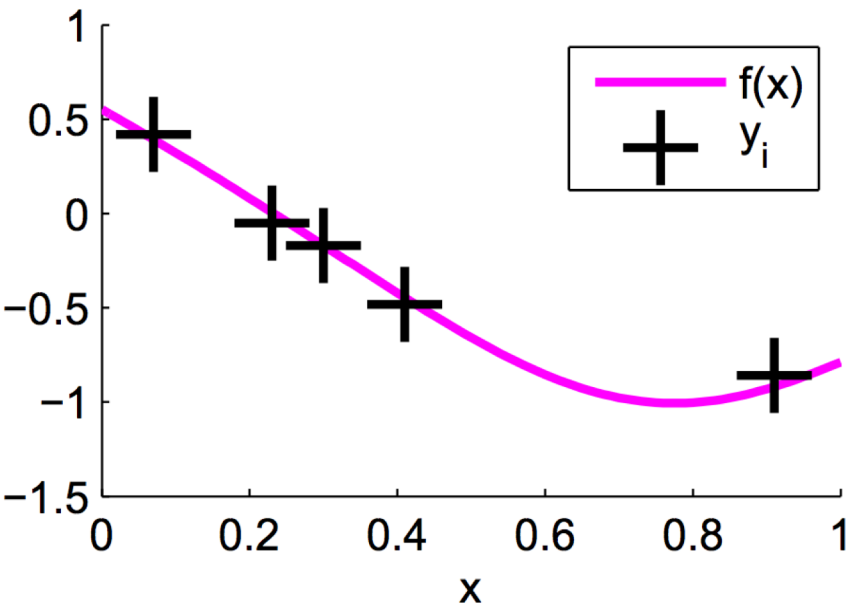
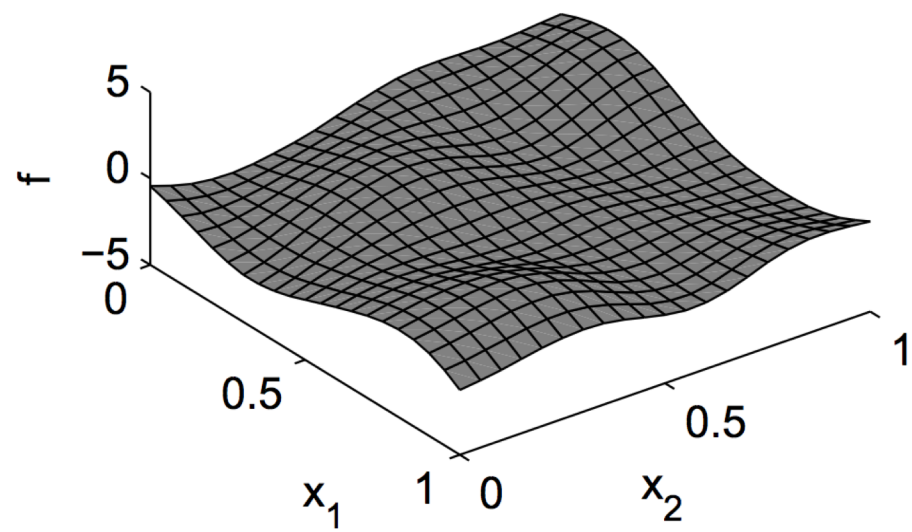
Gaussian Processes

Kayhan Batmanghelich

Goal: Learning a Function

Learn scalar function of vector values $f(\mathbf{x})$



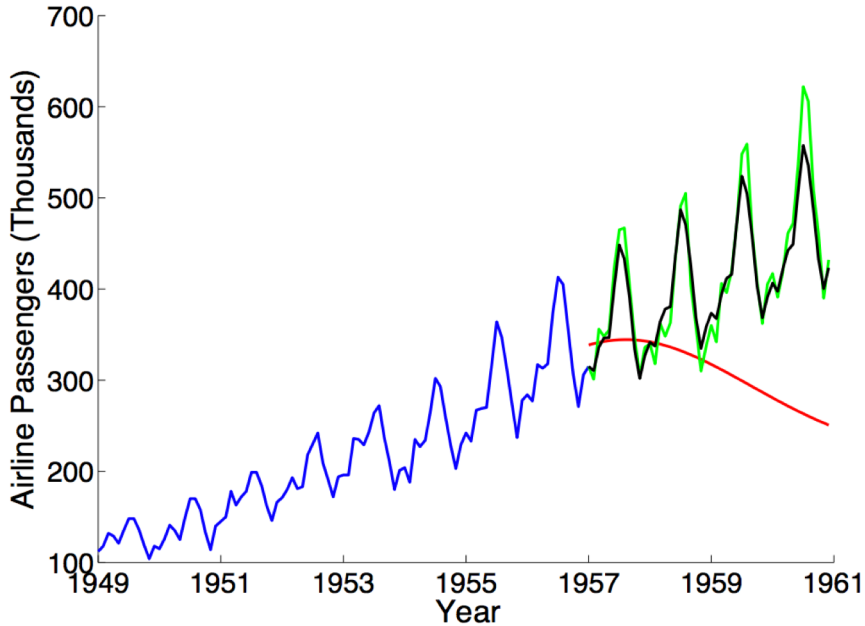


We have (possibly noisy) observations $\{\mathbf{x}_i, y_i\}_{i=1}^n$

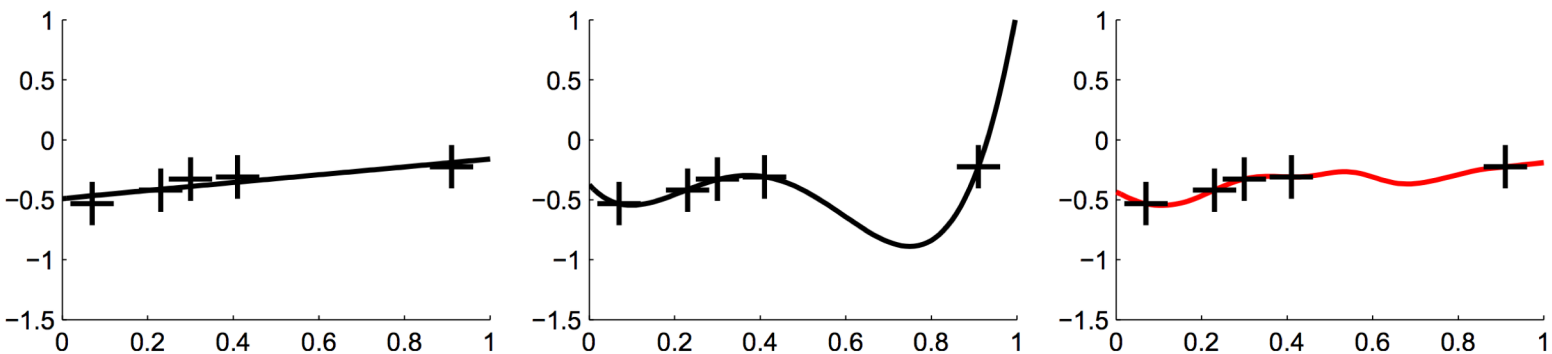
Slides: Iain Murray

A Regression Example

Predict the future:



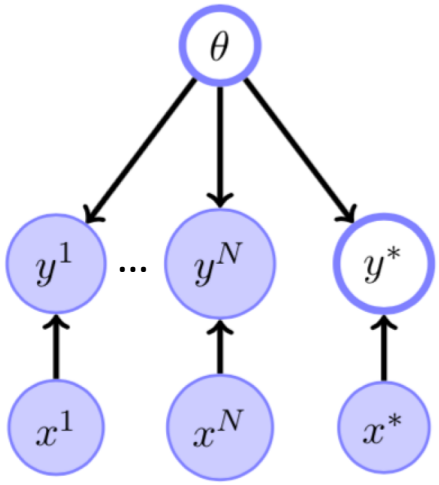
The world is often complicated:

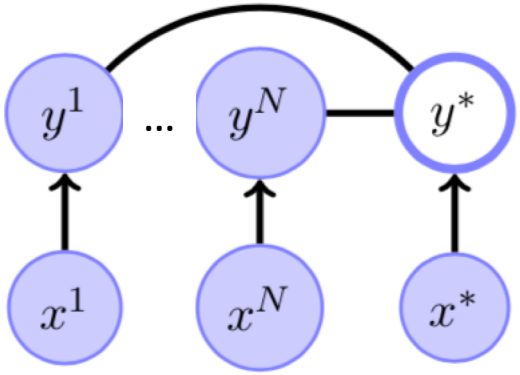


simple fit complex fit truth

Recap: Non-Parametric Approach

Let's revisit prediction (at test time):





Non-parametric methods directly model the joint conditional distribution!

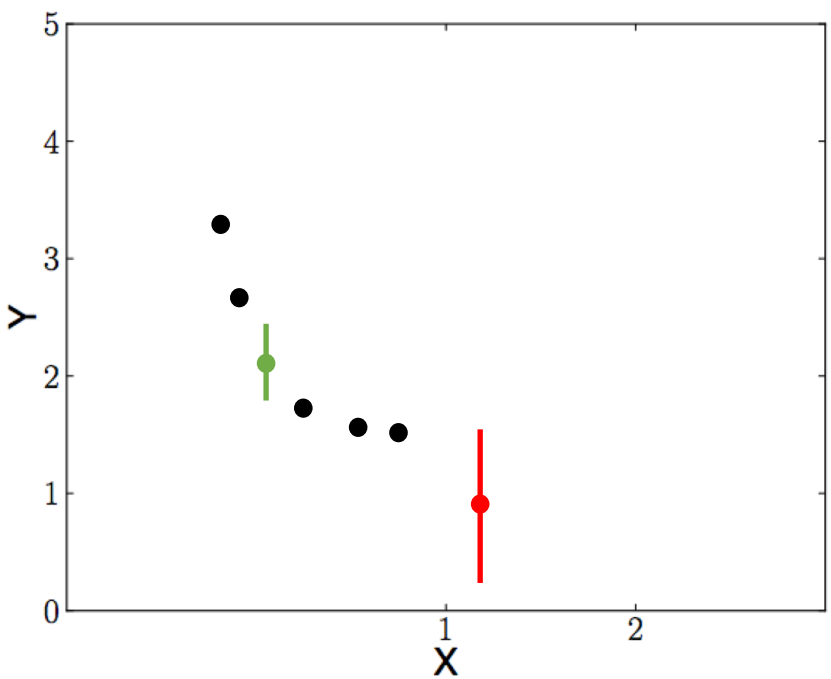
$$p(y^*, \mathcal{Y} | x^*, \mathcal{X}) = \int_\theta p(y^* | x^*, \theta) p(\theta) \prod_n p(y^n | \theta, x^n)$$

Recap: Revisiting Linear Regression

Linear Regression with no noise:

$$y = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \qquad p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}})$$

Feature vector



- In many applications, one relies on generic **smoothness** assumptions:
  - If two inputs x and x' then outputs (y and y') should be similar.

Recap: Revisiting Linear Regression

Linear Regression with no noise:

$$y = \mathbf{w}^T \boxed{\boldsymbol{\phi}(\mathbf{x})} \qquad p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}})$$

Feature vector

Let's integrate *w* out to get $p(y^*, \mathcal{Y}|x^*, \mathcal{X})$:

$$\mathbf{y} = [y^1, \cdots, y^N] \qquad \boldsymbol{\Phi} = [\boldsymbol{\phi}(x^1), \ldots, \boldsymbol{\phi}(x^N)]^\mathsf{T}$$

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

$$[\mathbf{K}]_{n,n'} = \boldsymbol{\phi}(x^n)^\mathsf{T}\boldsymbol{\phi}(x^{n'}) = k(x^n, x^{n'})$$ *covariance function*

What if instead of $\phi(x)$, we specify $k(x', x)$ directly?

Covariance Function

- The **covariance function** controls the behavior of the prediction function ($x \to y$) implicitly.

- Examples of $k(x, x')$:

$$v_0 \exp\left\{-\frac{1}{2}\sum_{l=1}^{D}\lambda_l\left(x_l - x_l'\right)^2\right\} \qquad \|x - x'\|^{\nu} K_{\nu}(\|x - x'\|) \qquad \mathbf{x}^{\mathsf{T}}\mathbf{x}'$$

Squared Exponential | Matern | Linear

Stationary | Non-Stationary

Jupyter Demo!

Covariance Function

- The **covariance function** controls the behavior of the prediction function ($x \to y$) implicitly.

- The $k(x, x')$ specifies the prior over functions.

- Potentially, the $\phi(x)$ function can be infinite dimensional which can be viewed as basis function for $f(x)$.

- *k* is a covariance ⇒ *k* is a positive semi-definite function

$$\int_{\mathcal{X}^2} k(x, x') f(x) f(x') \, d\mu(x) d\mu(x') \geq 0$$

Theorem (Loeve)

k corresponds to the covariance of a GP

$$\Updownarrow$$

k is a (symmetric) positive definite function

Kernel Matrix

$$K_{XX} = \begin{bmatrix} \langle \phi(x_1), \phi(x_1) \rangle & \cdots & \boxed{\langle \phi(x_1), \phi(x_n) \rangle} \\ \vdots & \ddots & \vdots \\ \langle \phi(x_n), \phi(x_1) \rangle & \cdots & \langle \phi(x_n), \phi(x_n) \rangle \end{bmatrix}$$

$k(x_1, x_n)$

N

N

Implicit Feature Vector

- ϕ_X is an **operator** that maps R^N to functions $\mathcal{F}$.
- $K_{XX} = \Phi_X^T \Phi_X$

$$\mathbf{\Phi_X} = \begin{bmatrix} \phi(x_1) & \phi(x_2) & \cdots & \phi(x_N) \end{bmatrix}$$

∞

N

Gaussian Process

- The Gaussian process (GP) as a prior on **functions**.
- *Covariance function* and hyperparameters reflect the prior belief on function smoothness, length scales etc.
- A GP is a collection of random variables, any **finite number** of which have a joint **Gaussian** distribution.

Infinite dimension:

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

Finite dimension:

$$[f(x_1), \ldots, f(x_N)] \sim \mathcal{N}(\boldsymbol{\mu}, K)$$

$$\boldsymbol{\mu}_i = m(x_i)$$

$$K_{ij} = k(x_i, x_j)\ ,$$

Gaussian Process

- The Gaussian process (GP) as a prior on **functions**.
- *Covariance function* and hyperparameters reflect the prior belief on function smoothness, length scales etc.
- A GP is a collection of random variables, any **finite number** of which have a joint **Gaussian** distribution.
- Easy inference for the regression.

Inference in GP: Regression

Given observed noisy data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, the joint probability over latent function values $\mathbf{f}$ and $\mathbf{f}^\star$ given $\mathbf{y}$ is

Regression Case:

$$p([\mathbf{f}, \mathbf{f}^\star] \,|\, \mathbf{X}, \mathbf{X}^\star, \mathbf{y}, \boldsymbol{\theta}_\mathsf{K}, \sigma^2) \propto \overbrace{\mathcal{N}\left([\mathbf{f}, \mathbf{f}^\star] \,\middle|\, \mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{X},\mathbf{X}} & \mathbf{K}_{\mathbf{X},\mathbf{X}^\star} \\ \mathbf{K}_{\mathbf{X}^\star,\mathbf{X}} & \mathbf{K}_{\mathbf{X}^\star,\mathbf{X}^\star} \end{bmatrix}\right)}^{\text{Prior}} \times \underbrace{\prod_{n=1}^{N} \mathcal{N}\left(y_n \,|\, f_n, \sigma^2\right)}_{\text{Likelihood}},$$

$$\mathbf{f}(\mathbf{X}) = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \end{bmatrix}$$

$$\mathbf{f}^*(\mathbf{X}^*) = \begin{bmatrix} f^*(\mathbf{x}_1^*) \\ f^*(\mathbf{x}_2^*) \\ \vdots \end{bmatrix}$$

Inference in GP: Regression

Given observed noisy data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, the joint probability over latent function values $\mathbf{f}$ and $\mathbf{f}^\star$ given $\mathbf{y}$ is

Regression Case:

$$p([\mathbf{f}, \mathbf{f}^\star] \,|\, \mathbf{X}, \mathbf{X}^\star, \mathbf{y}, \boldsymbol{\theta}_\mathsf{K}, \sigma^2) \propto \overbrace{\mathcal{N}\left([\mathbf{y}, \mathbf{f}^\star] \,\middle|\, \mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{X},\mathbf{X}} + \sigma^2\mathbf{I} & \mathbf{K}_{\mathbf{X},\mathbf{X}^\star} \\ \mathbf{K}_{\mathbf{X}^\star,\mathbf{X}} & \mathbf{K}_{\mathbf{X}^\star,\mathbf{X}^\star} \end{bmatrix}\right)}^{\text{Prior}} \times \underbrace{\prod_{n=1}^{N} \mathcal{N}\left(y_n \,|\, f_n, \sigma^2\right)}_{\text{Likelihood}},$$

$$p(\mathbf{f}^\star \,|\, \mathbf{X}, \mathbf{y}, \mathbf{X}^\star, \boldsymbol{\theta}_\mathsf{K}, \sigma^2) = \mathcal{N}\left(\mathbf{f}^\star \,|\, \boldsymbol{\mu}^\star, \boldsymbol{\Sigma}^\star\right) \text{ with}$$

$$\boldsymbol{\mu}^\star = \mathbf{K}_{\mathbf{X}^\star,\mathbf{X}} \left[\mathbf{K}_{\mathbf{X},\mathbf{X}} + \sigma^2\mathbf{I}\right]^{-1} \mathbf{y}$$

$$\boldsymbol{\Sigma}^\star = \mathbf{K}_{\mathbf{X}^\star,\mathbf{X}^\star} - \mathbf{K}_{\mathbf{X}^\star,\mathbf{X}} \left[\mathbf{K}_{\mathbf{X},\mathbf{X}} + \sigma^2\mathbf{I}\right]^{-1} \mathbf{K}_{\mathbf{X},\mathbf{X}^\star}$$

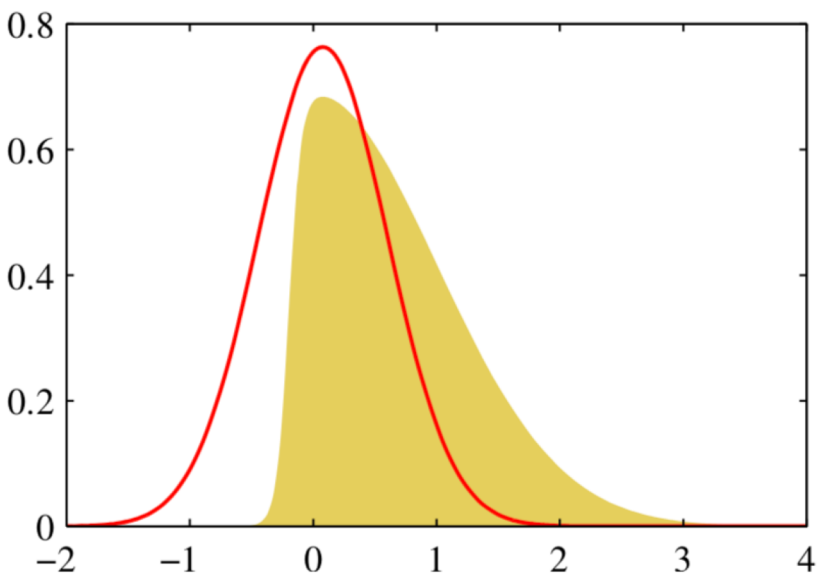
Inference in GP: Classification

Given observed noisy data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, the joint probability over latent function values $\mathbf{f}$ and $\mathbf{f}^\star$ given $\mathbf{y}$ is

Classification Case:

$$p([\mathbf{f}, \mathbf{f}^\star] \,|\, \mathbf{X}, \mathbf{X}^\star, \mathbf{y}, \boldsymbol{\theta}_\mathsf{K}, \sigma^2) \propto \overbrace{\mathcal{N}\left([\mathbf{f}, \mathbf{f}^\star] \,\middle|\, \mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{X},\mathbf{X}} & \mathbf{K}_{\mathbf{X},\mathbf{X}^\star} \\ \mathbf{K}_{\mathbf{X}^\star,\mathbf{X}} & \mathbf{K}_{\mathbf{X}^\star,\mathbf{X}^\star} \end{bmatrix}\right)}^{\text{Prior}}$$

$$\times \underbrace{\prod_{n=1}^{N} \mathcal{N}\left(y_n \,|\, f_n, \sigma^2\right)}_{\text{Likelihood}},$$



$$p(y_n = 1 \,|\, f_n) = \frac{1}{1 + exp(-f_n)}$$

What if we approximate the likelihood term to look like a Gaussian?

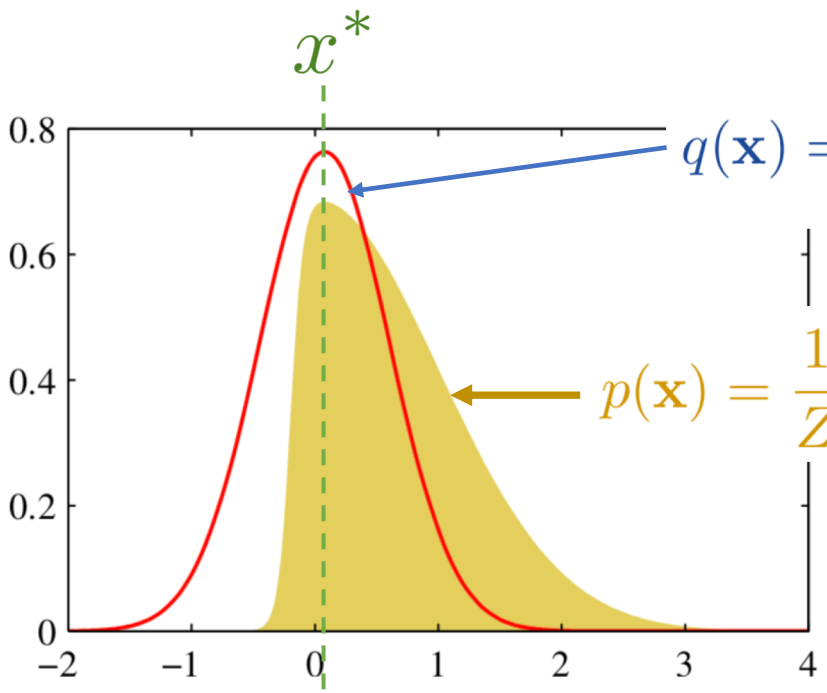
Not closed form!

Laplace Method

Given observed noisy data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, the joint probability over latent function values $\mathbf{f}$ and $\mathbf{f}^\star$ given $\mathbf{y}$ is

Use **Taylor** expansion around x^*:

$$E(\mathbf{x}) \approx E(\mathbf{x}^*) + (\mathbf{x} - \mathbf{x}^*)^\mathsf{T} \nabla E|_{\mathbf{x}^*} + \frac{1}{2} (\mathbf{x} - \mathbf{x}^*)^\mathsf{T} \mathbf{H} (\mathbf{x} - \mathbf{x}^*)$$



$$q(\mathbf{x}) = \frac{1}{Z_q} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^\mathsf{T}\mathbf{H}(\mathbf{x}-\mathbf{x}^*)} = \mathcal{N}\left(\mathbf{x}|\mathbf{x}^*, \mathbf{H}^{-1}\right)$$

$$p(\mathbf{x}) = \frac{1}{Z} e^{-E(\mathbf{x})}$$

Learning

Using maximum likelihood to learn the parameters of the kernel (θ)?

$$p(\mathbf{y}|\mathbf{X}) = \int \boxed{p(\mathbf{y}|\mathbf{f},\mathbf{X})}\boxed{p(\mathbf{f}|\mathbf{X})}d\mathbf{f}$$

$$p(\mathbf{y}|\mathbf{f}) = \prod_i \mathcal{N}(y_i|f_i,\sigma_y^2) \qquad p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\mathbf{0},\mathbf{K})$$

$$\log p(\mathbf{y}|\mathbf{X}) = \log \mathcal{N}(\mathbf{y}|\mathbf{0},\mathbf{K}_y) = -\frac{1}{2}\mathbf{y}\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_y| - \frac{N}{2}\log(2\pi)$$

K_y absorbs the effect of noise variance

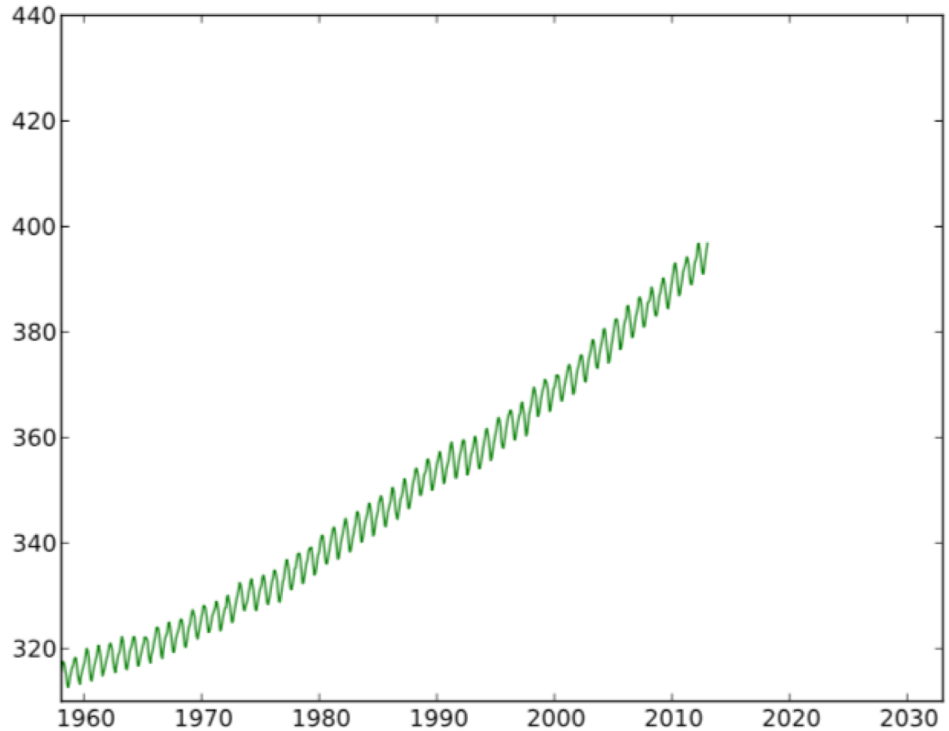
$$\frac{\partial}{\partial\theta_j}\log p(\mathbf{y}|\mathbf{X}) = \frac{1}{2}\mathbf{y}^T\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}(\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j})$$

How to Choose $k(x, x')$?

Example:

Data: This famous dataset compiles the monthly CO_2 concentration in Hawaii since 1958.

Task: Predict the concentration for the next 20 years.



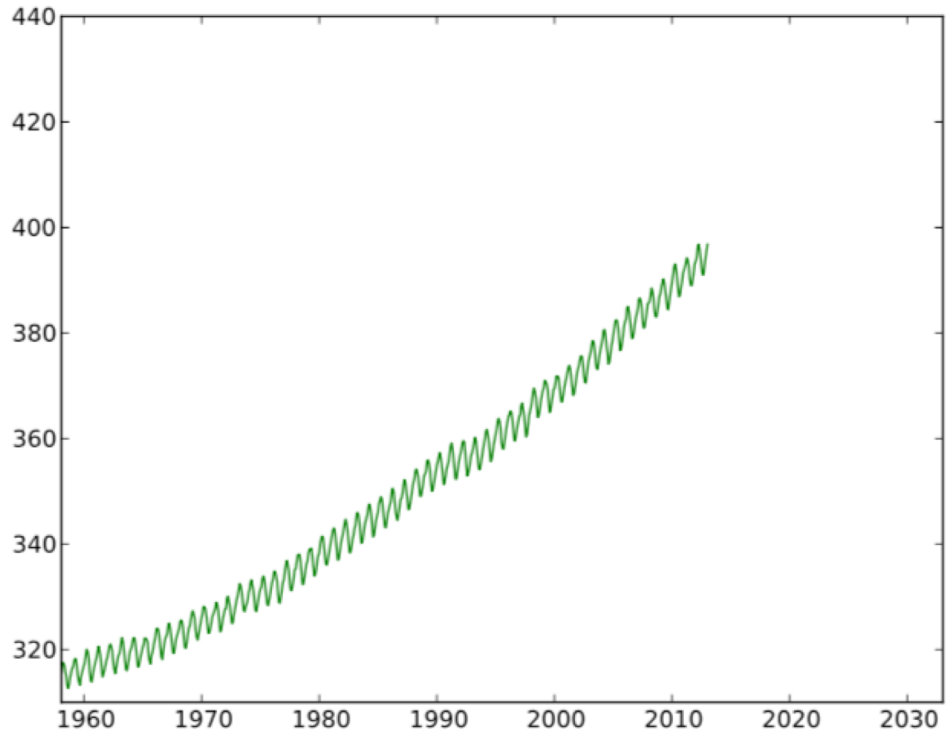
Figures Credit: Nicolas Durrande

How to Choose $k(x, x')$?

Example:

Data: This famous dataset compiles the monthly CO_2 concentration in Hawaii since 1958.

Task: Predict the concentration for the next 20 years.

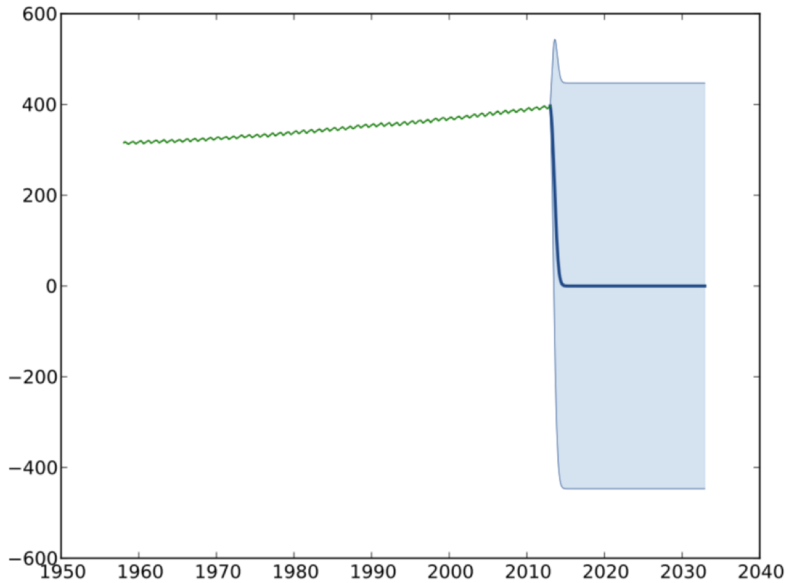


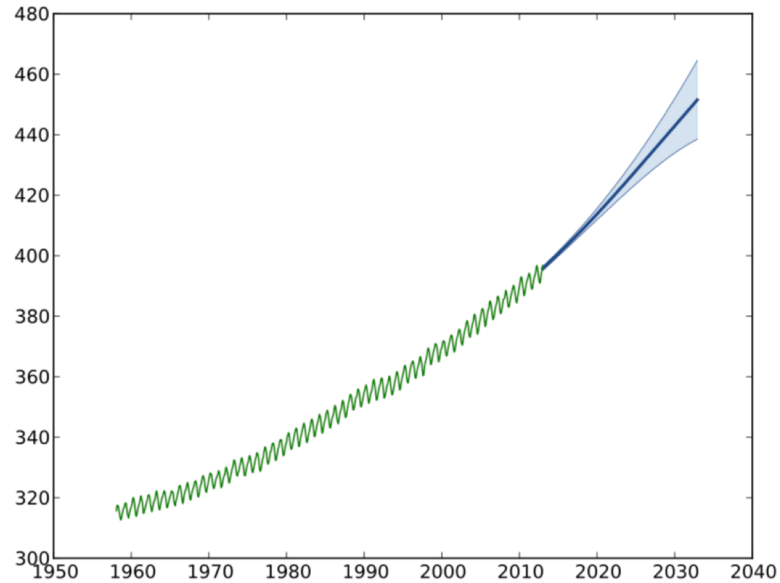
Figures Credit: Nicolas Durrande

How to Choose $k(x, x')$?

$$k(x, y) = \sigma^2 \exp\left(-\frac{(x-y)^2}{\theta^2}\right)$$

Terrible!



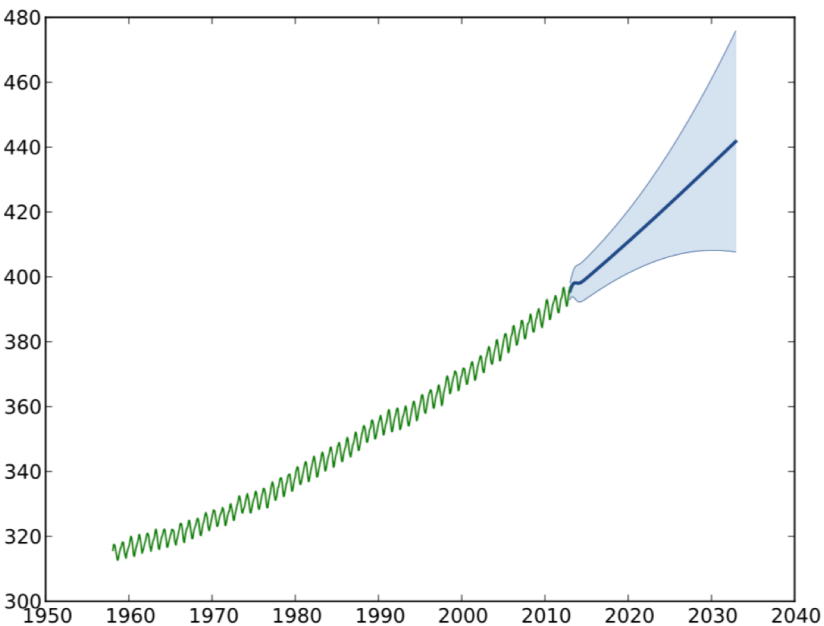


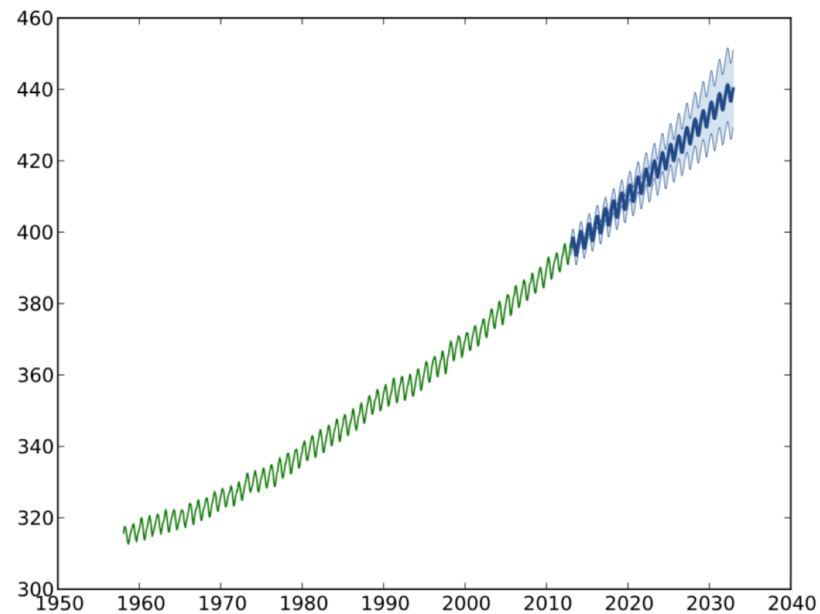
Figures Credit: Nicolas Durrande

How to Choose $k(x, x')$?

$$k(x, y) = k_{rbf1}(x, y) + k_{rbf2}(x, y)$$

$$k(x, y) = \sigma_0^2 x^2 y^2 + k_{rbf1}(x, y) + k_{rbf2}(x, y) + k_{per}(x, y)$$





Meh…!

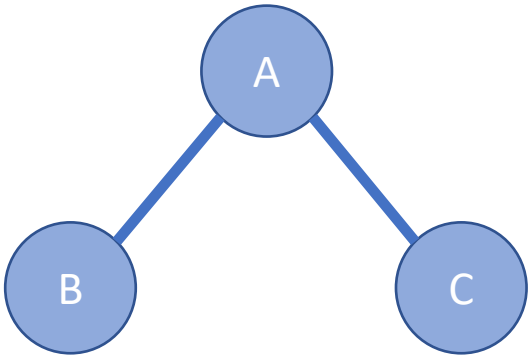
What about the oscillations?

Much better!

Figures Credit: Nicolas Durrande

How can we extend this idea to Graphical Model?

General Ideas



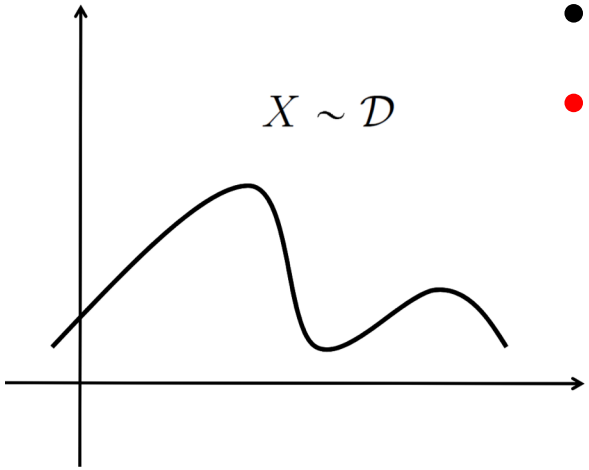
- Parametric forms of conditional

Discrete

$\mathbb{P}[C=0\|A=0]$	$\mathbb{P}[C=0\|A=1]$
$\mathbb{P}[C=1\|A=0]$	$\mathbb{P}[C=1\|A=1]$

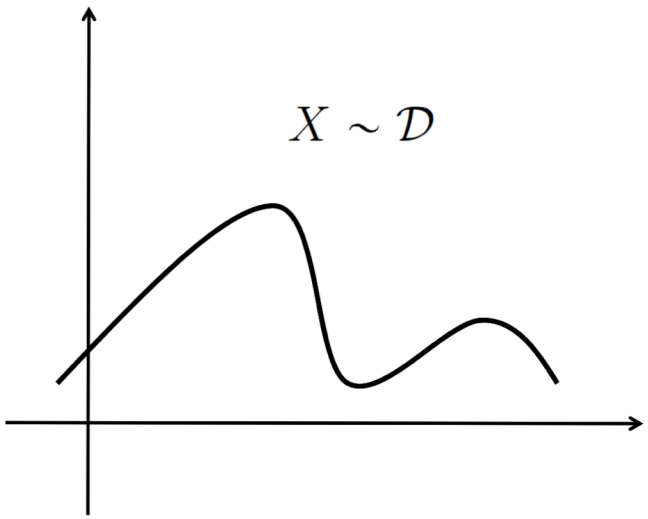
Continuous

$$\mathcal{N}(\mu, \sigma)$$



- Can we make it **non-parametric**?
- **Key idea**: represent this distribution with a small vector μ_X

Key Idea



$$\mu_X = \begin{pmatrix} \mathbb{E}[X] \\ \mathbb{E}[X^2] \\ \mathbb{E}[X^3] \\ \vdots \end{pmatrix}$$

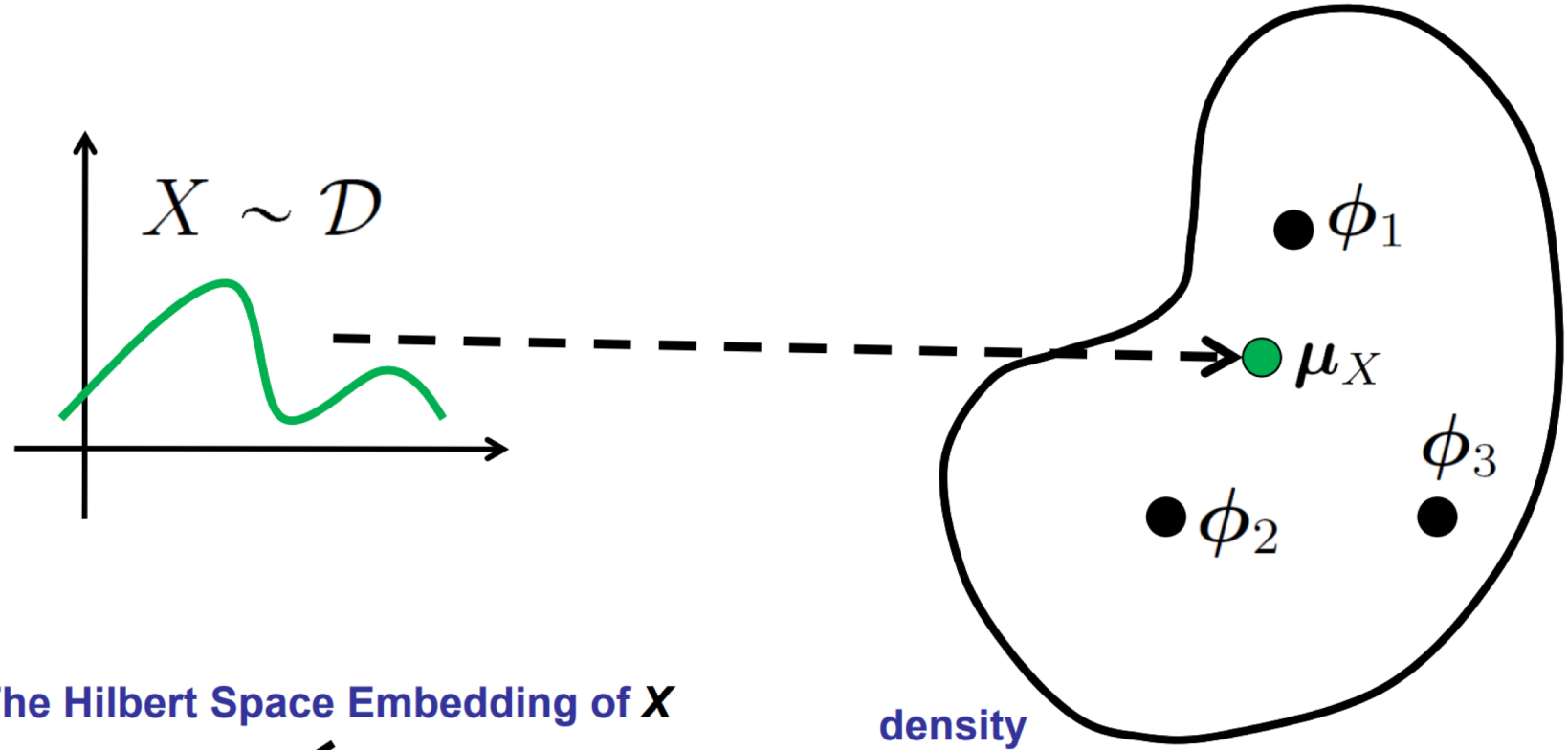
Not enough!

$$\mathbb{E}[\phi_X]$$

Right idea but not exactly….

- ϕ_X is an infinite dimensional vector space.

Embedding Distributions



The Hilbert Space Embedding of *X*

density

$$\boldsymbol{\mu}_X(\cdot) = \mathbb{E}_{X\sim\mathcal{D}}[\phi_X] = \int \boldsymbol{p}_{\mathcal{D}}(X)\phi_X(\cdot)\,dX$$

Embedding Distributions

We leave detail discussion to a future. Here is a general idea:

Original Space	RKHS Space
X	$\phi_X \in \mathcal{F}$
Y	$\psi_Y \in \mathcal{G}$
(X, Y)	$(\phi_X, \psi_Y) \in \mathcal{F} \times \mathcal{G}$
$\mathbb{P}(X)$	$\mu_X = \mathbb{E}[\phi_X]$
$\mathbb{P}(Y)$	$\mu_Y = \mathbb{E}[\psi_Y]$
$\mathbb{P}(X, Y)$	$\mathcal{C}_{YX} = \mathbb{E}[\psi_Y \otimes \phi_X]$

cross-covariance **operator**

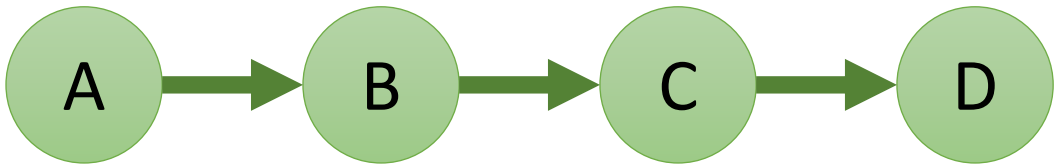
Embedding Distributions

We leave detail discussion to a future. Here is a general idea:

Original Space	RKHS Space
$\mathbb{P}(X, Y)$	$\mathcal{C}_{YX} = \mathbb{E}[\psi_Y \otimes \phi_X]$
$\mathrm{Diag}[P(X)]$	$\mathcal{C}_{XX} = \mathbb{E}[\phi_X \otimes \phi_X]$
$\mathbb{P}[Y\|X] = \mathbb{P}[Y, X] \times \mathrm{Diag}(\mathbb{P}[X])^{-1}$	$\boldsymbol{\mathcal{C}}_{\boldsymbol{Y}\|\boldsymbol{X}} = \boldsymbol{\mathcal{C}}_{YX}\boldsymbol{\mathcal{C}}_{XX}^{-1}$
$\mathbb{P}[X] = \int_Y \mathbb{P}[X, Y] = \int_Y \mathbb{P}[X\|Y]\mathbb{P}[Y]$	$\boldsymbol{\mu}_X = \boldsymbol{\mathcal{C}}_{X\|Y}\boldsymbol{\mu}_Y$
$\mathbb{P}[X, Y] = \mathbb{P}[X\|Y]\mathbb{P}[Y]$	$\boldsymbol{\mathcal{C}}_{YX} = \boldsymbol{\mathcal{C}}_{Y\|X}\boldsymbol{\mathcal{C}}_{XX}$

Kernel Graphical Models

- Replacing CPT with RKHS operators/function



$$\mathcal{C}_{AA} \quad \mathcal{C}_{B|A} \quad \mathcal{C}_{C|B} \quad \mathcal{C}_{D|C}$$

Sum Product

With discrete variables:

$$\mathcal{P}(A, D) = \mathcal{P}(\oslash A)\mathcal{P}(B|A)^\top \mathcal{P}(C|B)^\top \mathcal{P}(D|C)^\top$$

With kernels:

$$\mathcal{C}_{AD} = \mathcal{C}_{AA}\mathcal{C}_{B|A}^\top \mathcal{C}_{B|C}^\top \mathcal{C}_{C|D}^\top$$

Summary

- Hilbert Space Embedding provides a way to create a "sufficient statistic" for an arbitrary distribution.

- Can embed marginal, joint, and conditional distributions into the RKHS.

- **More references:**
  - Smola, A. J., Gretton, A., Song, L., and Schölkopf, B., A Hilbert Space Embedding for Distributions, Algorithmic Learning Theory, E. Takimoto (Eds.), Lecture Notes on Computer Science, Springer, 2007.
  - L. Song. Learning via Hilbert space embedding of distributions. PhD Thesis 2008.
  - Song, L., Huang, J., Smola, A., and Fukumizu, K., Hilbert space embeddings of conditional distributions, International Conference on Machine Learning, 2009.