EECS 545: Machine Learning

Lecture 9. Kernel methods: Gaussian Processes

Honglak Lee

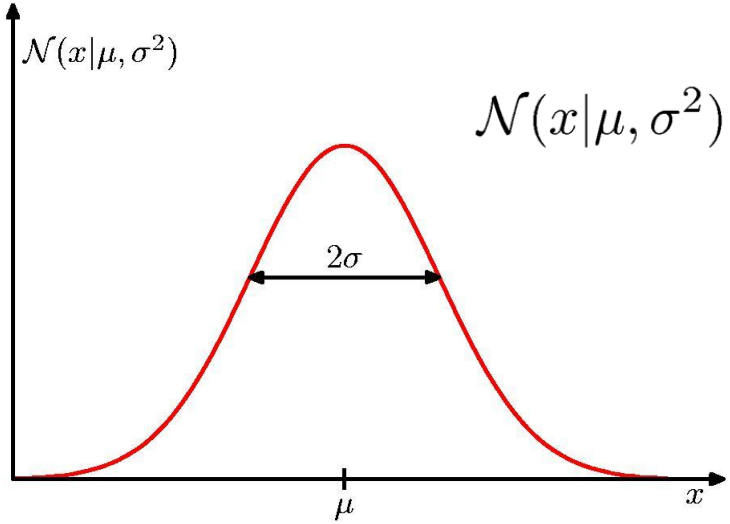
2/7/2011

Outline

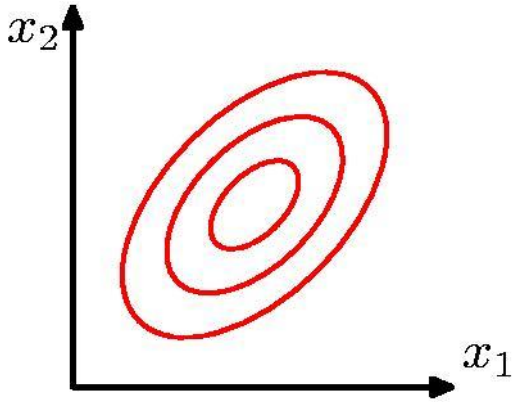
- More about Multivariate Gaussians
  - Marginal and conditional distributions
- Bayesian Linear Regression
- Gaussian Processes
  - GP for Regression

Multivariate Gaussians – marginal and conditional distributions

The Gaussian Distribution



$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

Multivariate Gaussian (mean)

$$\begin{aligned}\mathbb{E}[\mathbf{x}] &= \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\int \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}\mathbf{x}\,\mathrm{d}\mathbf{x} \\ &= \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\int \exp\left\{-\frac{1}{2}\mathbf{z}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{z}\right\}(\mathbf{z}+\boldsymbol{\mu})\,\mathrm{d}\mathbf{z}\end{aligned}$$

Thanks to the anti-symmetry of **z**, we get:

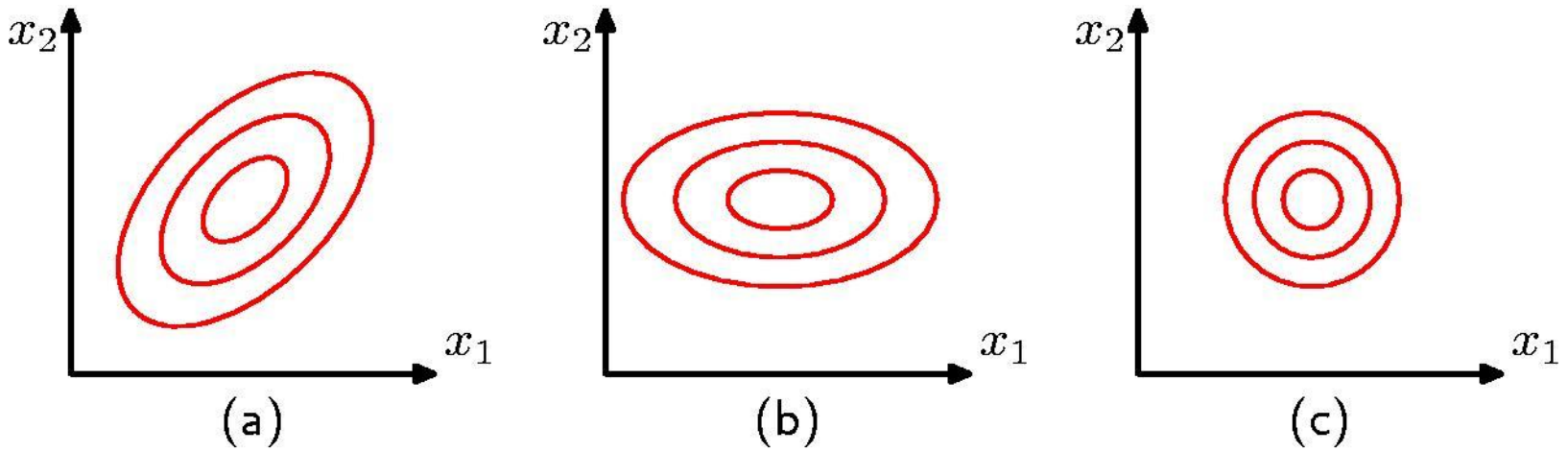
$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$$

Multivariate Gaussian (covariance)

$$\mathbb{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}] = \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}} + \boldsymbol{\Sigma}$$

$$\mathrm{cov}[\mathbf{x}] = \mathbb{E}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^{\mathrm{T}}\right] = \boldsymbol{\Sigma}$$

Types of covariance matrices:



General Diagonal const*Identity

Partitioned Gaussian Distributions

- Multivariate Gaussian distribution for x

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Partitioning x into x_a and x_b.
  - Mean and covariance

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix} \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}$$

  - Precision matrix

$$\boldsymbol{\Lambda} \equiv \boldsymbol{\Sigma}^{-1} \qquad \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix}$$

Partitioned Conditionals and Marginals

- Conditional distribution

$$p(\mathbf{x}_a|\mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Sigma}_{a|b})$$

$$\begin{aligned}
\boldsymbol{\Sigma}_{a|b} &= \boldsymbol{\Lambda}_{aa}^{-1} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \\
\boldsymbol{\mu}_{a|b} &= \boldsymbol{\Sigma}_{a|b}\left\{\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b)\right\} \\
&= \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\
&= \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b)
\end{aligned}$$

- Marginal distribution

$$\begin{aligned}
p(\mathbf{x}_a) &= \int p(\mathbf{x}_a, \mathbf{x}_b)\,\mathrm{d}\mathbf{x}_b \\
&= \mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})
\end{aligned}$$

Derivation of conditional Gaussian

- From the exponent of the Gaussian

$$-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) =$$
$$-\frac{1}{2}(\mathbf{x}_a-\boldsymbol{\mu}_a)^{\mathrm{T}}\boldsymbol{\Lambda}_{aa}(\mathbf{x}_a-\boldsymbol{\mu}_a)-\frac{1}{2}(\mathbf{x}_a-\boldsymbol{\mu}_a)^{\mathrm{T}}\boldsymbol{\Lambda}_{ab}(\mathbf{x}_b-\boldsymbol{\mu}_b)$$
$$-\frac{1}{2}(\mathbf{x}_b-\boldsymbol{\mu}_b)^{\mathrm{T}}\boldsymbol{\Lambda}_{ba}(\mathbf{x}_a-\boldsymbol{\mu}_a)-\frac{1}{2}(\mathbf{x}_b-\boldsymbol{\mu}_b)^{\mathrm{T}}\boldsymbol{\Lambda}_{bb}(\mathbf{x}_b-\boldsymbol{\mu}_b).$$

- Note that general Gaussians have the following form:

$$-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = -\frac{1}{2}\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x}+\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}+\mathrm{const}$$

- Treat x_b's as constant and rearrange terms for x_a's.

Derivation of conditional Gaussian

- Second order term:

$$-\frac{1}{2}\mathbf{x}_a^{\mathrm{T}}\boldsymbol{\Lambda}_{aa}\mathbf{x}_a \quad \Rightarrow \quad \boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Lambda}_{aa}^{-1}.$$

- First order term:

$$\mathbf{x}_a^{\mathrm{T}}\left\{\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b)\right\}$$

– From $\Sigma_{a|b}^{-1}\mu_{a|b} = \Lambda_{aa}\mu_a - \Lambda_{ab}(x_b - \mu_b)$

$$\begin{aligned}\boldsymbol{\mu}_{a|b} &= \boldsymbol{\Sigma}_{a|b}\left\{\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b)\right\} \\ &= \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b)\end{aligned}$$

Derivation of conditional Gaussian

- Expressing in terms of covariance matrix

$$\begin{aligned}\boldsymbol{\mu}_{a|b} &= \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\ \boldsymbol{\Sigma}_{a|b} &= \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}.\end{aligned}$$

  – where $\begin{pmatrix}\boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb}\end{pmatrix}^{-1} = \begin{pmatrix}\boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb}\end{pmatrix}$

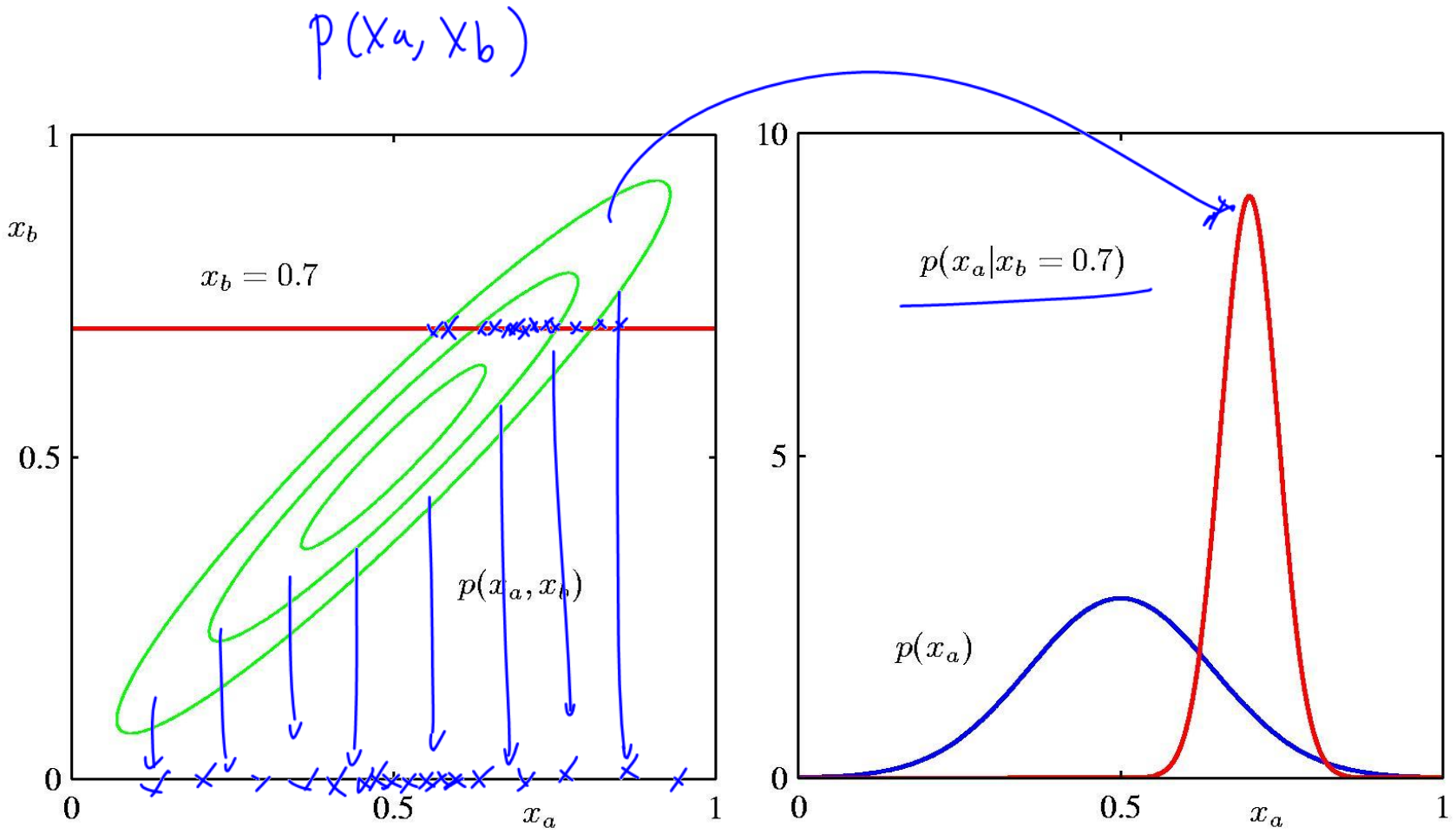
$$\begin{aligned}\boldsymbol{\Lambda}_{aa} &= (\boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba})^{-1} \\ \boldsymbol{\Lambda}_{ab} &= -(\boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba})^{-1}\boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}.\end{aligned}$$

- Here, we used matrix inversion lemma

$$\begin{pmatrix}\mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D}\end{pmatrix}^{-1} = \begin{pmatrix}\mathbf{M} & -\mathbf{M}\mathbf{B}\mathbf{D}^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}\mathbf{M} & \mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{C}\mathbf{M}\mathbf{B}\mathbf{D}^{-1}\end{pmatrix}$$

  – where $\mathbf{M} = (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}.$

Partitioned Conditionals and Marginals



Linear Gaussian Distributions

- Linear combination of Gaussians is also a Gaussian distribution.
  - Its marginal distribution is also a Gaussian
  - Its conditional distribution is also a Gaussian

Bayes' Theorem for Gaussian Variables

- Given

$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right)$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}\left(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b}, \mathbf{L}^{-1}\right)$$

$p(x, y)$

- we have

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}+\mathbf{b}, \mathbf{L}^{-1}+\mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}})$$
$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y}-\mathbf{b})+\boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma})$$

- where

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}$$

Bayesian Linear Regression

Regression

- Given a set of observations: x = { x_1 . . . x_N }
  - And corresponding target values: **t** = { t_1 . . . t_N }

- We want to learn a function $y(x)=t$ to predict future values.
  - We have just learned to find the maximum likelihood weights $\mathbf{w}_{ML}$, to predict $y(x,\mathbf{w}_{ML})$.

$$\mathbf{w}_{ML} = (\lambda\mathbf{I} + \mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{t}$$

$P(w|D) \propto P(w, D) = P(D|w)P(w)$

Gaussian Prior on w

$t_n \in \mathbb{R} \sim \mathcal{N}(w^T \phi(x_n), \beta^{-1})$

- With a likelihood:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T\phi(\mathbf{x}_n), \beta^{-1})$$

$P(D|w)$

- and a prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$P(w)$

- we get a posterior $p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$
- where $\mathbf{m}_N = \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\mathbf{\Phi}^T\mathbf{t})$
- and $\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\mathbf{\Phi}^T\mathbf{\Phi}$

Simplify the Prior

- Zero-mean isotropic Gaussian:

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$

$\mathbf{w} \in \mathbb{R}^M$

$\mathbf{I}_M \in \mathbb{R}^{M \times M}$

- The corresponding posterior is:

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$

- where $\mathbf{m}_N = \beta \mathbf{S}_N \mathbf{\Phi}^T \mathbf{t}$
- and $\mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \mathbf{\Phi}^T \mathbf{\Phi}$

Derivation of posterior distribution

- Use linear Gaussian distributions

$$p(w) = N(w|0, \alpha^{-1}I)$$
$$p(t|w, x) = N(t|\Phi w, \beta^{-1}I)$$

- Conditional distribution

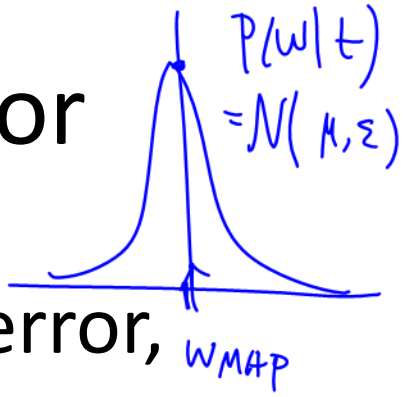
$$p(w|t, x) = N(w|\Sigma(\beta\Phi^T t), \Sigma^{-1})$$

where $\Sigma = (\alpha I + \beta\Phi^T\Phi)^{-1}$

Recall:

$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right)$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}\left(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}\right)$$
$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}})$$
$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\{\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma})$$
$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}$$

Maximize the (log) Posterior



P(w|t)

- Same as minimizing sum-of-squared error, with regularization term:

$\propto P(w)\,P(t|w)$

$$\ln p(\mathbf{w}|\mathbf{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(x_n)\}^2 - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w} + \text{ const}$$

- This is the same as w_{ML}, with $\lambda = \frac{\alpha}{\beta}$
- So the solution is (same as w_{ML})

$$\mathbf{w}_{MAP} = (\lambda\mathbf{I} + \mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{t}$$

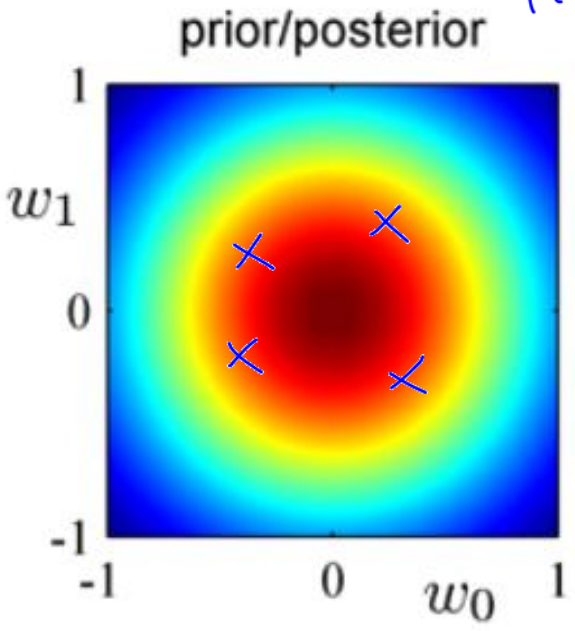
- But now we have the variance on w, as well!

Sequential Bayesian Learning

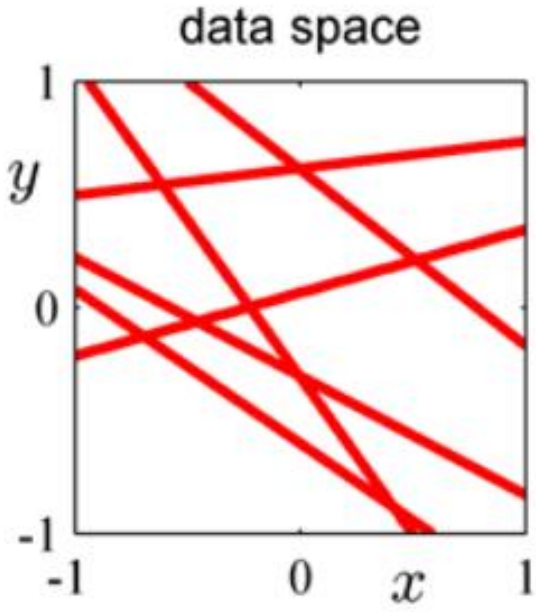
- Simple model: $y(x,\mathrm{w}) = w_0 + w_1 x$.
  - **Prior * Likelihood = Posterior**

likelihood ~ 1

$p(w) \propto \exp\left(-\frac{\alpha w^T w}{2}\right)$

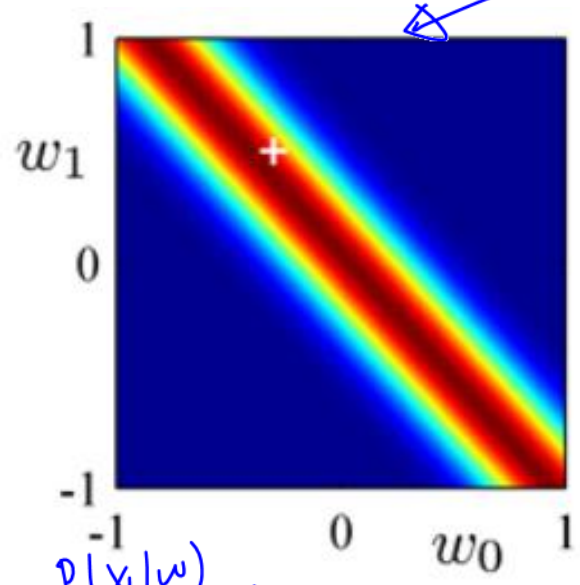


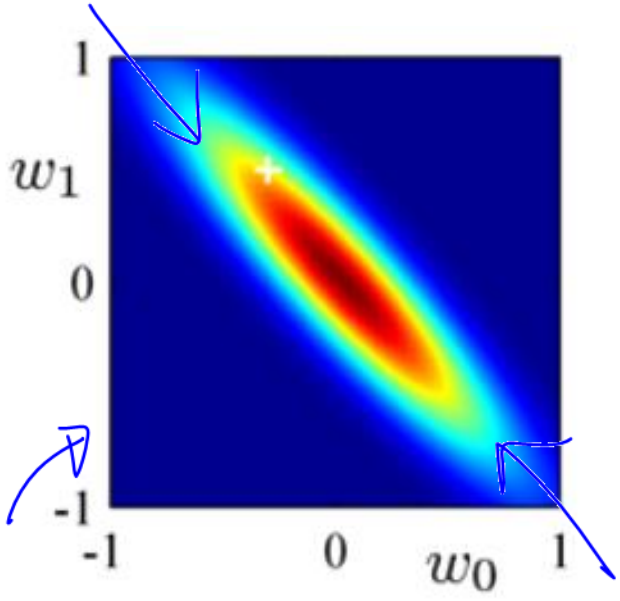
$\log p(w)$

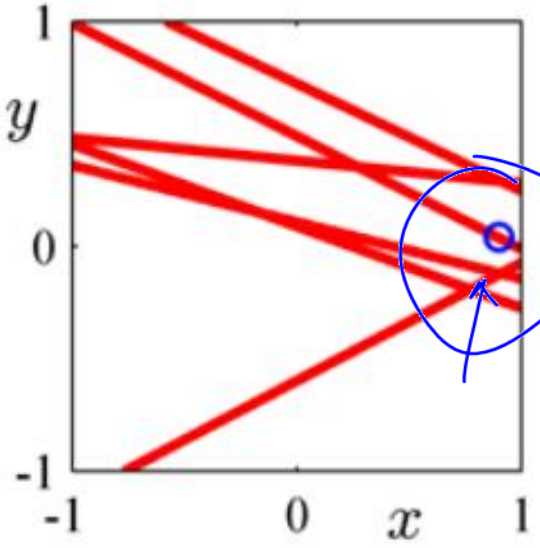


Sequential Bayesian Learning

- Likelihood of most recent data point.
  - **Posterior = likelihood * prior**
- Samples drawn from posterior.





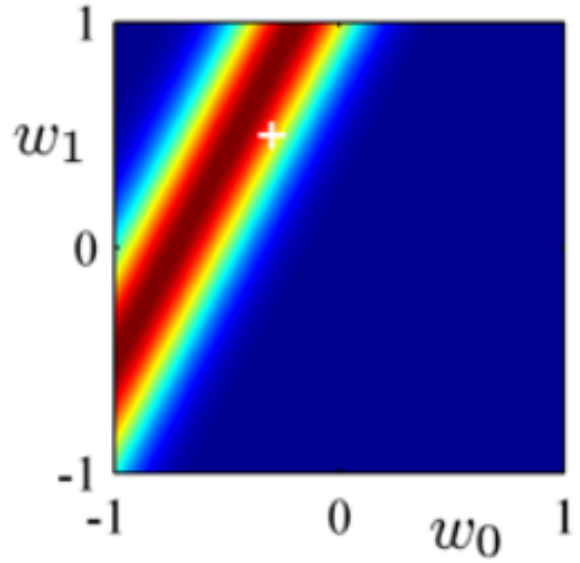


$p(x_i|w)$

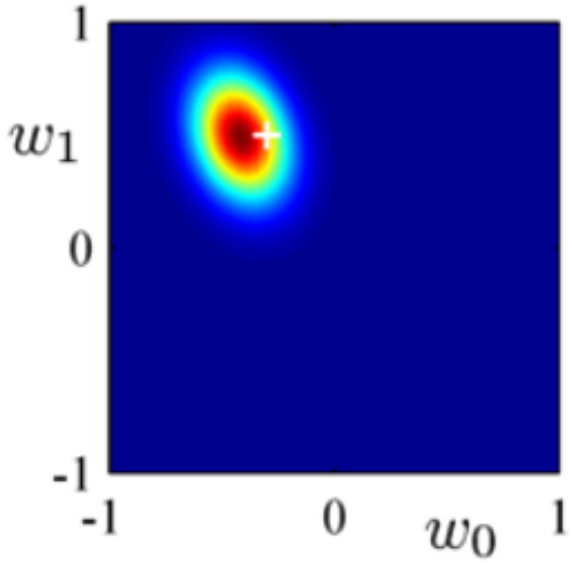
$\ln p(t|w) \propto \left(w^T \phi(x) - t\right)^2 + \frac{\alpha}{2} w^T w$

Sequential Bayesian Learning

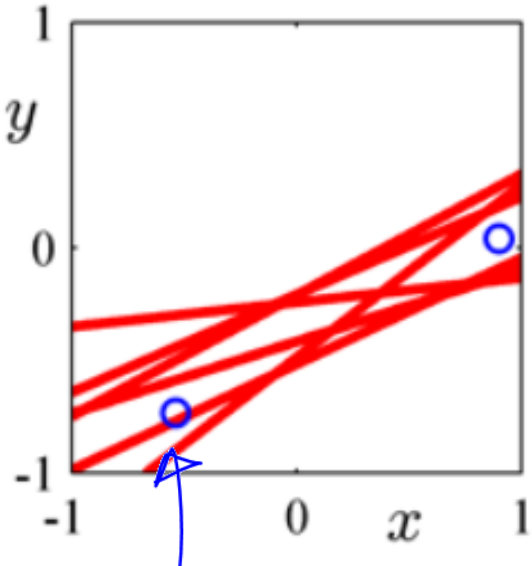
- Likelihood of second data point (only).
  - **Posterior = Likelihood * Prior**
- Sample lines drawn from posterior.



$P(x_2|w)$

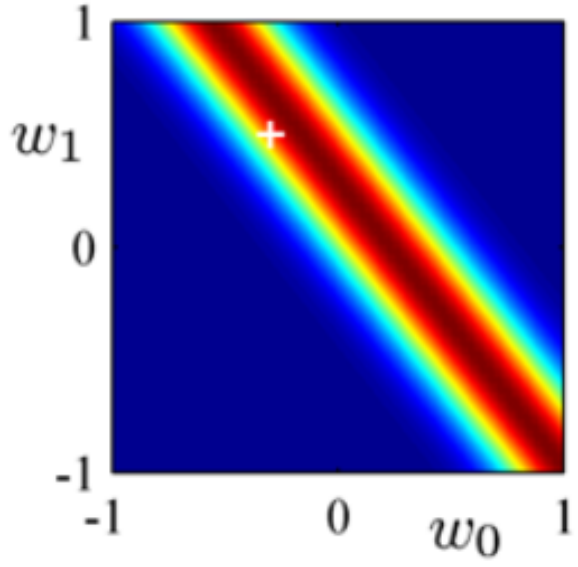


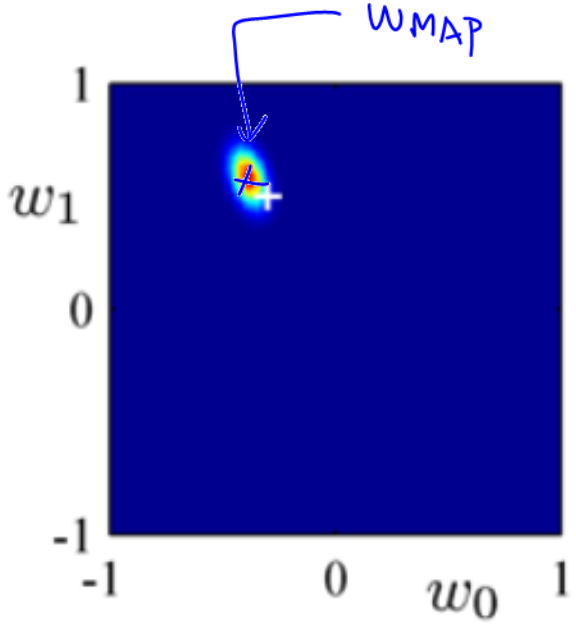
$\ln P(x_1|w) + \ln P(x_2|w) + \ln P(w)$

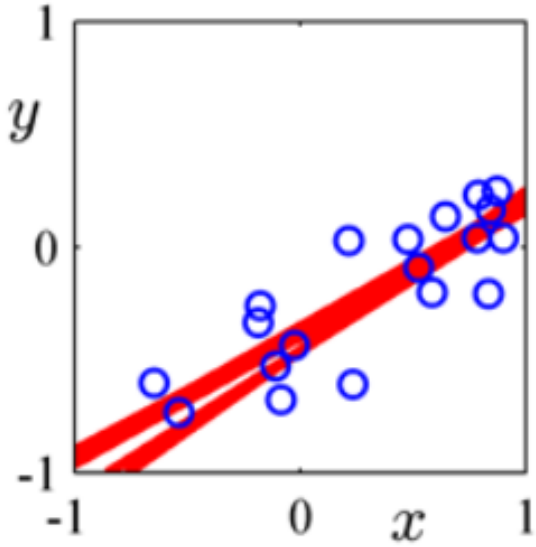


Sequential Bayesian Learning

- Likelihood of third data point (only).
  - **Posterior = Likelihood * Prior**
- Sample lines drawn from posterior







$$\ln P(w|D) = \sum_{i=1}^{N} \ln P(x_i|w) + \ln P(w)$$

$p(w|D) \rightarrow p(t,w|D) \rightarrow p(t|D)$ (marginalize w.)

Predictive Distribution

- Our real goal is to predict *t* given new x^*, so we evaluate the *predictive distribution*:

$$p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \int p(t^*|\mathbf{x}^*, \mathbf{w}, \beta)\, p(\mathbf{w}|\mathbf{t}, \alpha, \beta)\, d\mathbf{w}$$

$\mathcal{N}(w^T \phi(x^*), \beta^{-1})$ — $p(\mathbf{w}|\mathbf{t}, \alpha, \beta)$: Gaussian

- where

$$p(t|\mathbf{x}^*, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}^*, \mathbf{w}), \beta^{-1})$$

- and we just derived

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$

$$\mathbf{m}_N = \beta \mathbf{S}_N \mathbf{\Phi}^T \mathbf{t} \qquad \mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \mathbf{\Phi}^T \mathbf{\Phi}$$

Predictive Distribution

- The integral is a convolution of Gaussians, so the result is:

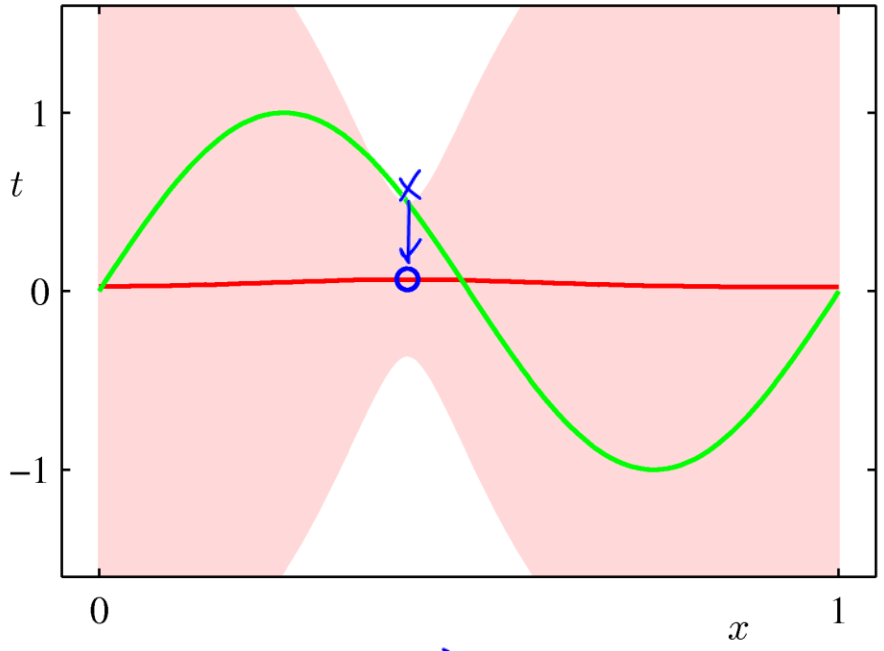
$$p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(t|\mathbf{m}_N^T \phi(\mathbf{x}), \sigma_N^2(\mathbf{x}))$$

- where

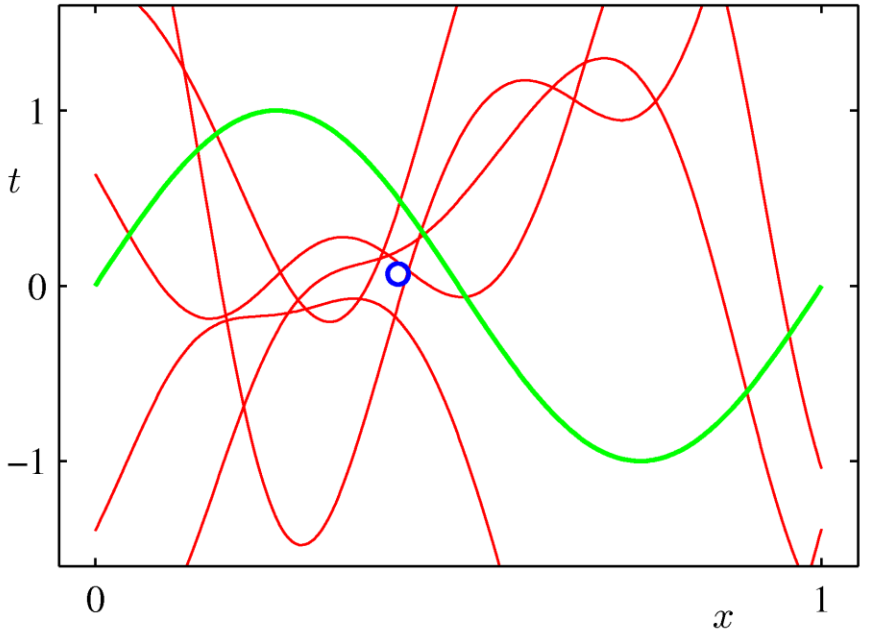
$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \phi(\mathbf{x})^T \mathbf{S}_n \phi(\mathbf{x})$$

- = noise in the data + uncertainty in w

Predictive Distribution & Samples



$p(w|D)$

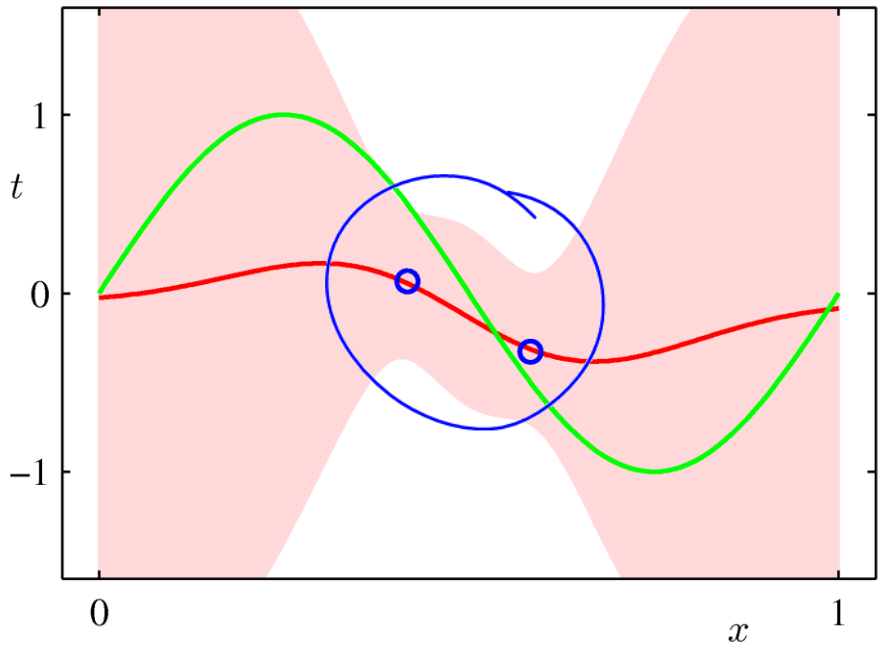


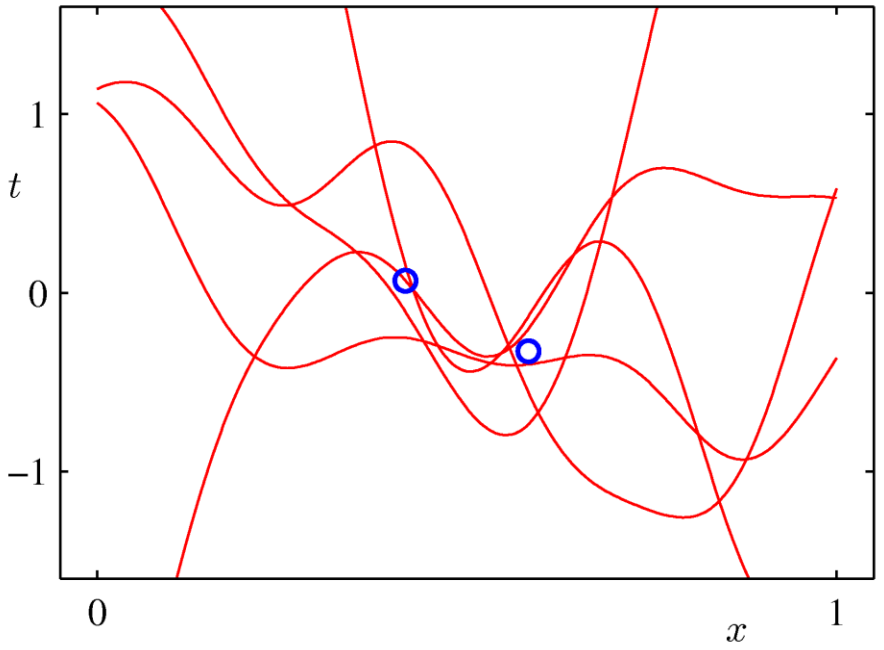
Using 9 Gaussian basis functions

- N=1 observed point

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

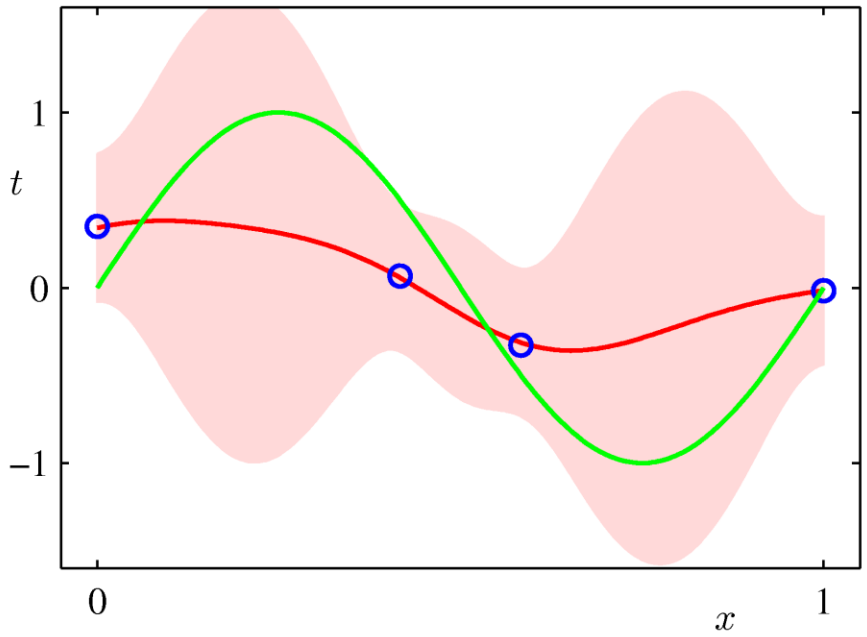
Predictive Distribution & Samples

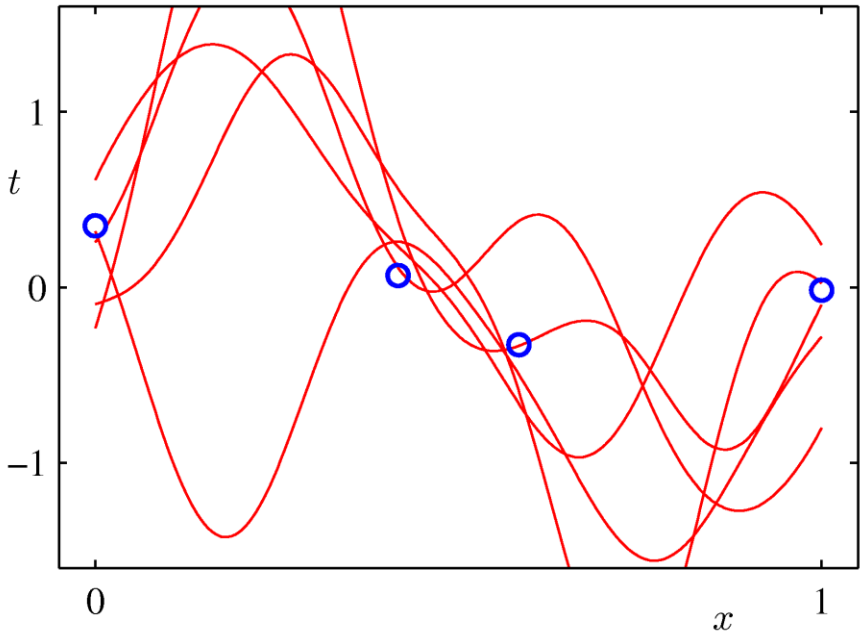




- N=2 observed points

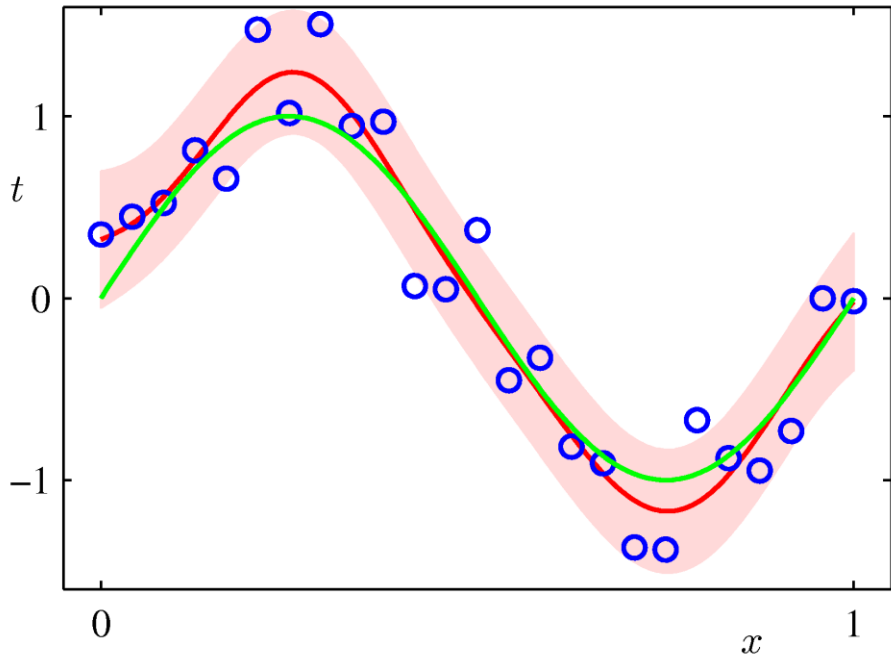
Predictive Distribution & Samples

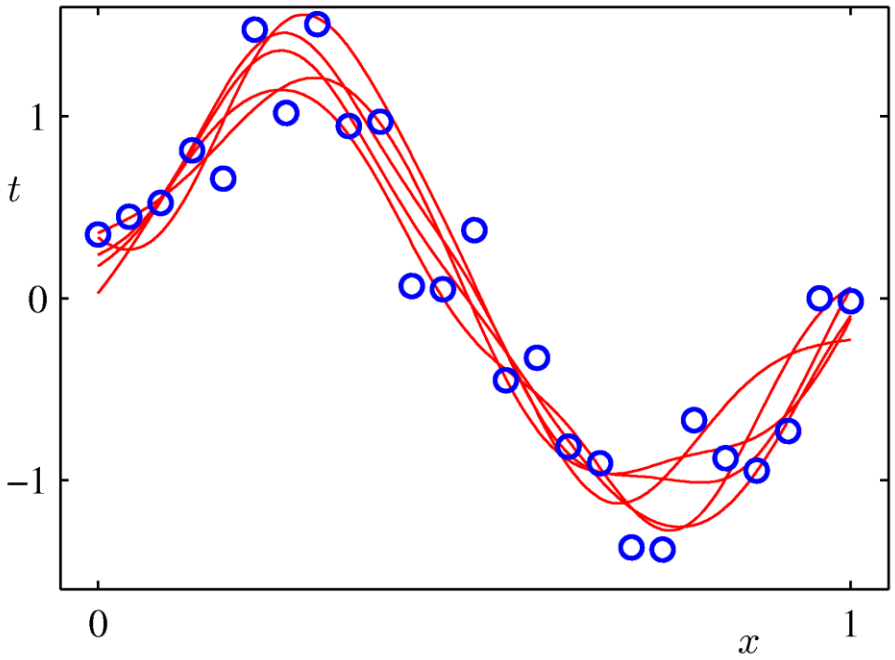




- N=4 observed points

Predictive Distribution & Samples



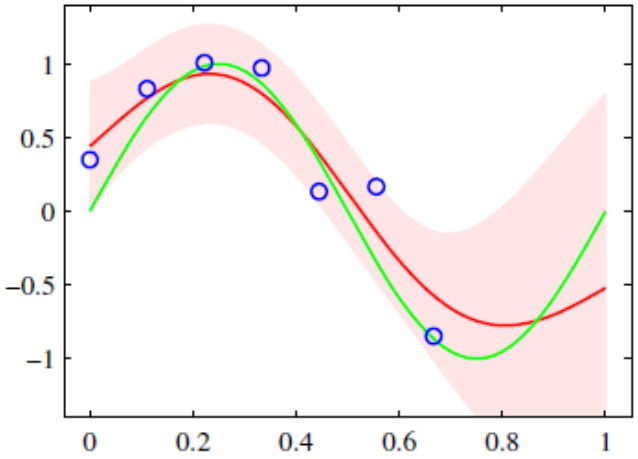


- N=25 observed points

Gaussian Processes

Why GPs?

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



- GPs are a nice way of expressing this “prior on functions” idea.
- Applications:
  - Regression
  - Classification

Linear Regression Revisited

- Linear regression model: Combination of M fixed basis functions given by $\phi(x)$, so that

$$y(x) = w^T \phi(x)$$

- Prior distribution $p(w) = N(w \mid 0, \alpha^{-1} I)$
- Given training data points $x_1, \ldots, x_n$, what is the joint distribution of $y(x_1), \ldots, y(x_n)$?
  - y is the vector with elements $y_n = y(x_n)$, this vector is given by

$$y = \Phi w$$

  - where Φ is the design matrix with elements $\Phi_{nk} = \Phi_k(x_n)$

Linear Regression Revisited

- $y = \Phi w$: y is a linear combination of Gaussian distributed variables w, hence itself is Gaussian.
- Mean and covariance

$$E[y] = \Phi E[w] = 0$$

$$\mathrm{cov}[y] = E[y y^T] = \Phi E[w w^T] \Phi^T = \frac{1}{\alpha} \Phi \Phi^T = K$$

where K is the Gram matrix with elements

$$K_{nm} = k(x_n, x_m) = \frac{1}{\alpha} \Phi(x_n)^T \Phi(x_m)$$

and $k(x, x')$ is the kernel function.

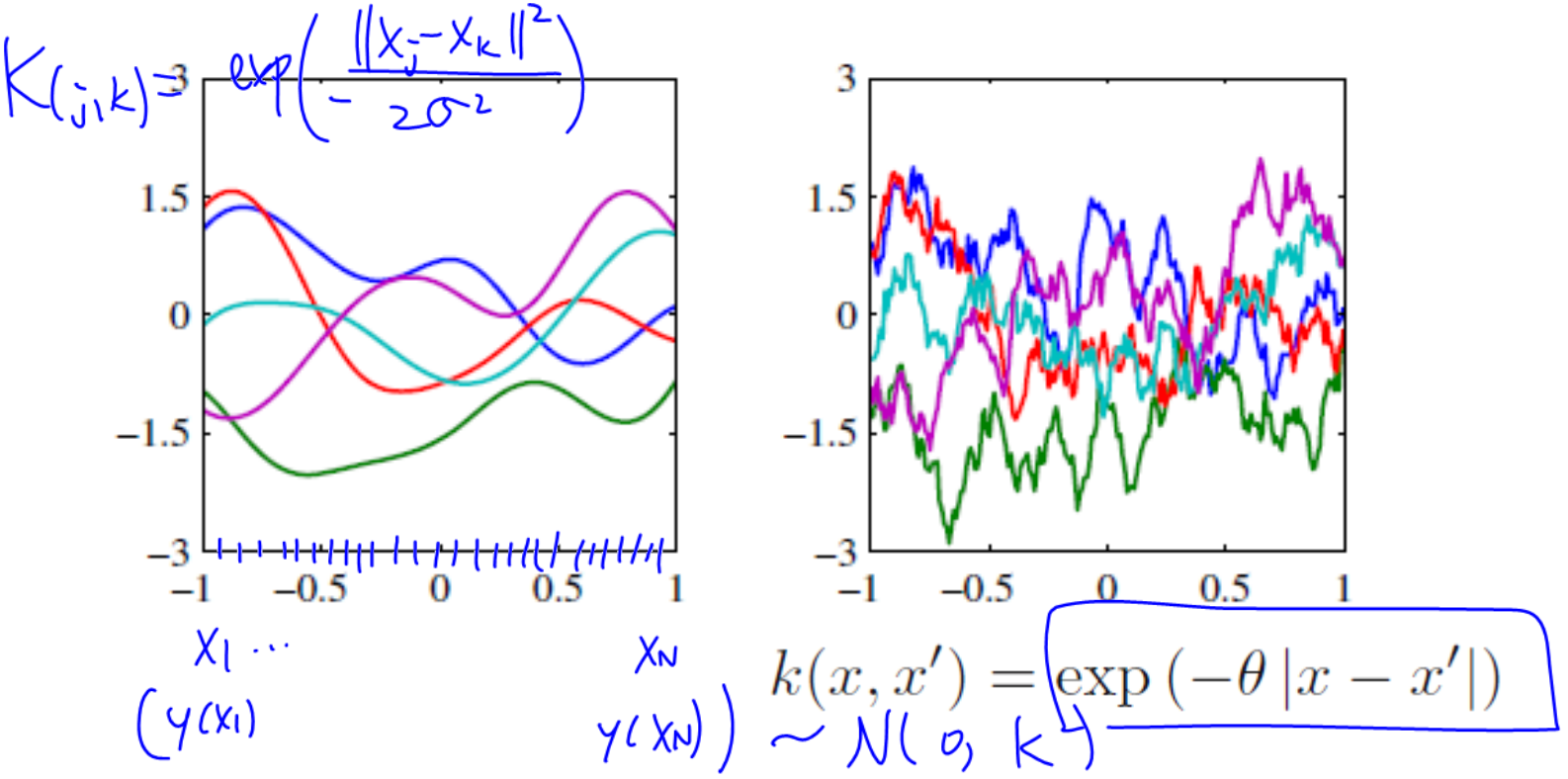
Definition of GP

- A Gaussian process is defined as a probability distribution over functions y(x), such that the set of values of y(x) evaluated at an arbitrary set of points x1,.. Xn jointly have a Gaussian distribution.
  - Any finite subset of indices defines a multivariate Gaussian distribution
- What determines the GP is
  - The mean function μ(x) = E(y(x))
  - The covariance function (kernel)

    $$k(x,x')=E(y(x)y(x'))$$
  - In most applications, we take μ(x)=0. Hence the prior is represented by the kernel.

Covariance function of GP defines prior

- The figure show samples of functions drawn from Gaussian processes for two different choices of kernel functions
  - Gaussian kernel (left) vs. exponential kernel (right)



Linear regression updated by GP

- Specific case of a Gaussian Process
- It is defined by the linear regression model

$$y(x) = w^T \phi(x)$$

- with a weight prior

$$p(w) = N(w \mid 0, \alpha^{-1} I)$$

- the kernel function is given by

$$k(x_n, x_m) = \frac{1}{\alpha} \phi(x_n)^T \phi(x_m)$$

GP for regression

- Take into account of the noise on the observed target values, which are given by

$$t_n = y_n + \varepsilon_n$$

where $y_n = y(x_n)$, and ε_n is a random noise variable

Here we consider noise processes that have a Gaussian distribution, so that

$$p(t_n \mid y_n) = N(t_n \mid y_n, \beta^{-1})$$

where β is a hyperparameter representing the precision of the noise.

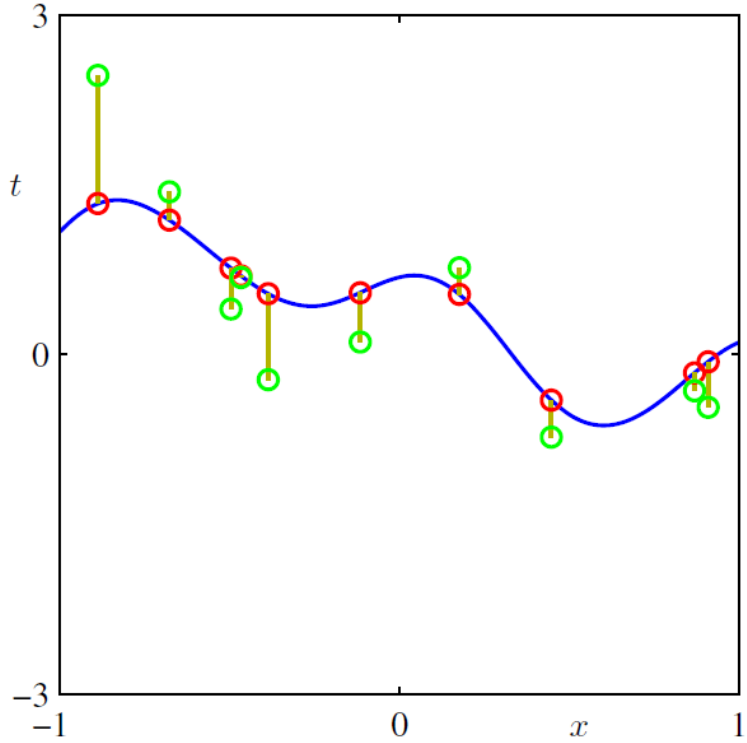
Because the noise is independent, the joint distribution of $t = (t_1, ..., t_n)^T$ conditioned on $y = (y_1, ..., y_n)^T$ is given by

$$p(t \mid y) = N(t \mid y, \beta^{-1} I_n)$$

Example: sampling data points

- Sample function from GP (blue): $y(x) \sim GP(\mu, \mathrm{K})$
- Sample points from GP (red): $(x_1, y(x_1)), (x_2, y(x_2)), (x_N, y(x_N)) \sim GP(\mu, K)$
- Add noise (green): $t_n(x) \sim y_n(x) + N(0, \beta^{-1})$

$(y(x_1), \ldots, y(x_N)) \sim \mathcal{N}(\mu, K)$



GP for regression

- From the definition of GP, the marginal distribution p(y) is given by

$$p(y) = N(y \mid 0, K)$$

- The marginal distribution of t is given by

$$p(t) = \int p(t \mid y) p(y) dy = N(t \mid 0, C)$$

marginal of linear gaussian.

- Where the covariance matrix C has elements

$$C(x_n, x_m) = k(x_n, x_m) + \beta^{-1} \delta_{nm}$$

$$\delta_{nm} = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{o.w} \end{cases}$$

GP for Regression

- We have used GP to build a model of the joint distribution over sets of data points
- Goal: Given training points $\mathrm{t}_n = (t_1,..,t_n)^T$, input values $\mathrm{x}_1,...,\mathrm{x}_n$, predict t_{n+1} for a new input x_{n+1}
- To find $p(t_{n+1} \mid t)$, we begin by writing down the joint distribution

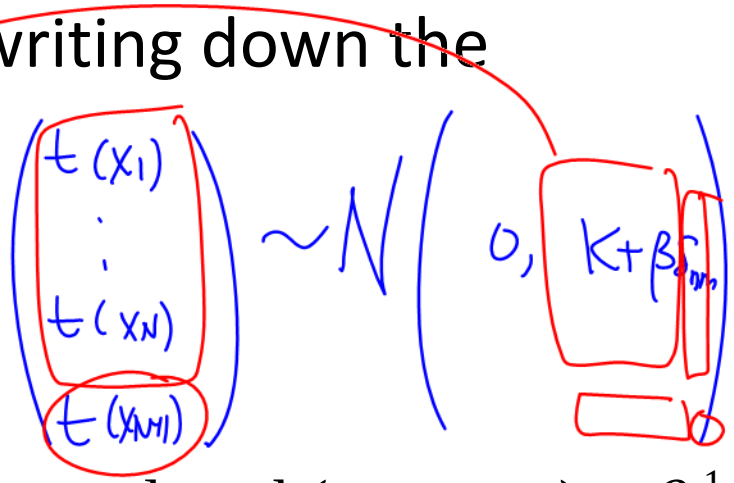
$$p(t_{n+1}) = N(t_{n+1} \mid 0, C_{n+1})$$

where C_{n+1} is $(\mathrm{n}+1)\times(\mathrm{n}+1)$ matrix,

$$C_{n+1} = \begin{pmatrix} C_n & k \\ k^T & \mathrm{c} \end{pmatrix}, \text{ where } C_n \text{ is } \mathrm{n}\times\mathrm{n} \text{ matrix, and } \mathrm{c} = \mathrm{k}(\mathrm{x}_{n+1}, \mathrm{x}_{n+1}) + \beta^{-1}$$

$(C_n)_{jk} = (K_n)_{jk} + \beta^{-1}\delta_{jk}$.

$P(t(X_{N+1}) \mid t(X_1) \cdots t(X_N))$



GP for Regression

- The conditional distribution $p(t_{n+1} \mid t)$ is a Gaussian distribution with mean and covariance given by

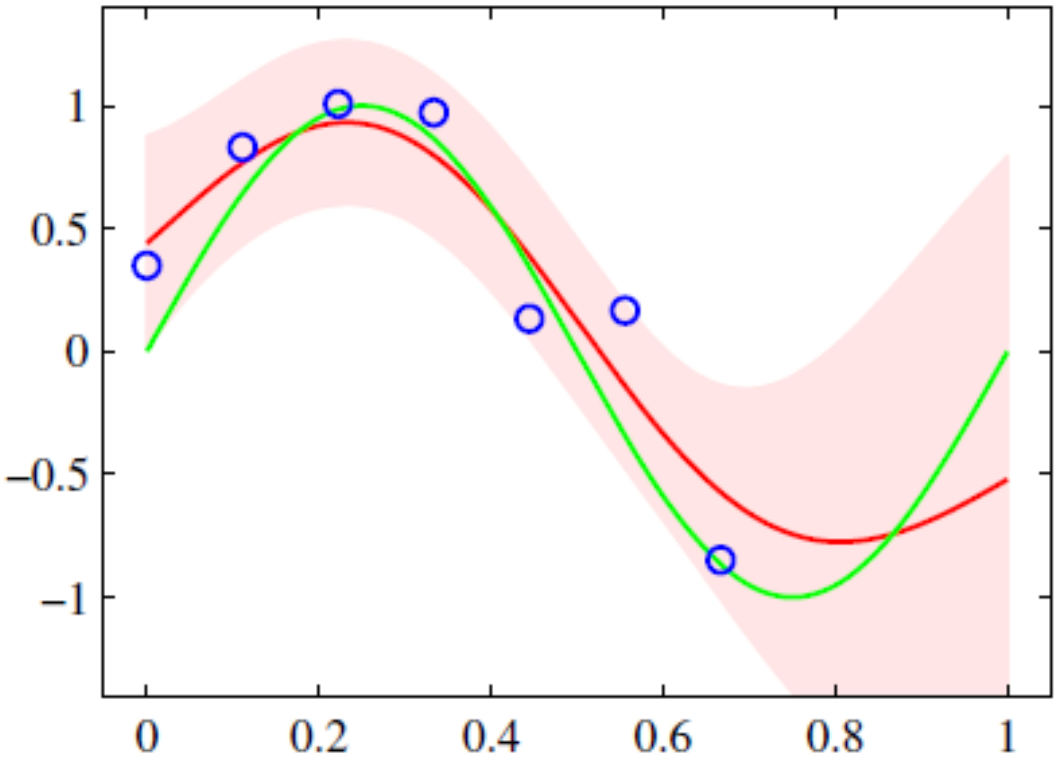
$$m(x_{n+1}) = k^T C_n^{-1} t$$

$$\sigma^2(x_{n+1}) = c - k^T C_n^{-1} k$$

- These are the key results that define Gaussian process regression.
- The predictive distribution is a Gaussian whose mean and variance both depend on x_{n+1}

An Example of GP regression

- Green: underlying function (sine function)
- Blue: samples from GP (with noise)
- Red: prediction from GP regression) with “error bars”



GP for Regression

- The only restriction on the kernel is that the covariance matrix given by

$$C(x_n, x_m) = k(x_n, x_m) + \beta^{-1}\delta_{nm}$$

must be positive definite.

- GP will involve a matrix of size N*N, for which require $O(N^3)$ computations.

Learning Hyperparameters

- Log likelihood

θ = param. for Kernel matrix.

$$\ln p(\mathbf{t}|\boldsymbol{\theta}) = -\frac{1}{2}\ln|\mathbf{C}_N| - \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{C}_N^{-1}\mathbf{t} - \frac{N}{2}\ln(2\pi).$$

- Gradient Ascent for parameter θ

$$\frac{\partial}{\partial\theta_i}\ln p(\mathbf{t}|\boldsymbol{\theta}) = -\frac{1}{2}\mathrm{Tr}\left(\mathbf{C}_N^{-1}\frac{\partial\mathbf{C}_N}{\partial\theta_i}\right) + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{C}_N^{-1}\frac{\partial\mathbf{C}_N}{\partial\theta_i}\mathbf{C}_N^{-1}\mathbf{t}.$$

  – where we used the following:

$$\frac{\partial}{\partial x}\left(\mathbf{A}^{-1}\right) = -\mathbf{A}^{-1}\frac{\partial\mathbf{A}}{\partial x}\mathbf{A}^{-1}$$

$$\frac{\partial}{\partial x}\ln|\mathbf{A}| = \mathrm{Tr}\left(\mathbf{A}^{-1}\frac{\partial\mathbf{A}}{\partial x}\right)$$

Conclusion

- Distribution over functions
- GP generates a jointly have a Gaussian distribution where the input data can be:
  - scalar
  - real vectors
  - graphs
  - strings
  - …
- Most interesting structure is in k(x,x'), the 'kernel.'
- GP can be used for regression to predict the target for a new input