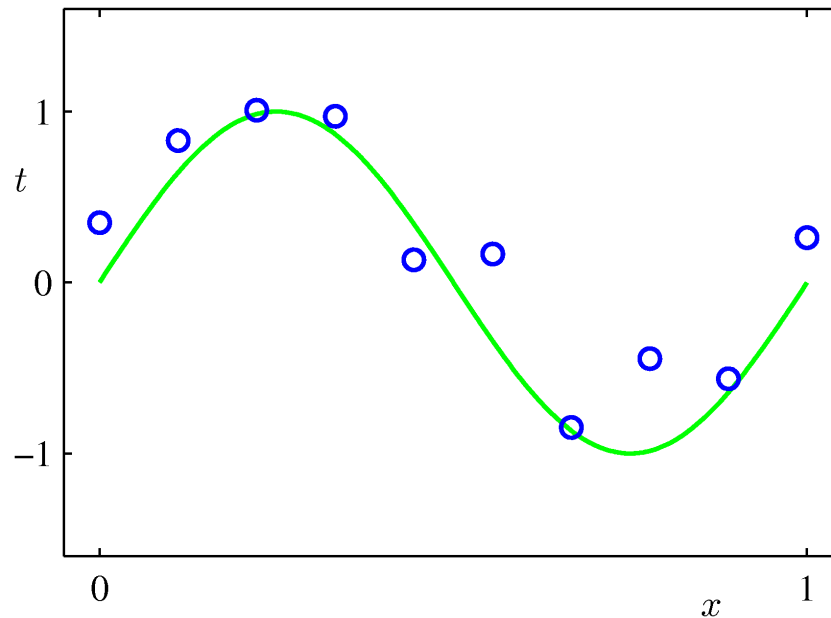

PATTERN RECOGNITION AND MACHINE LEARNING

CHAPTER 3: LINEAR MODELS FOR REGRESSION

Linear Basis Function Models (1)

Example: Polynomial Curve Fitting



$$y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M = \sum_{j=0}^M w_jx^j$$

Linear Basis Function Models (2)

Generally

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

where $\phi_j(\mathbf{x})$ are known as *basis functions*.

Typically, $\phi_0(\mathbf{x}) = 1$, so that w_0 acts as a bias.

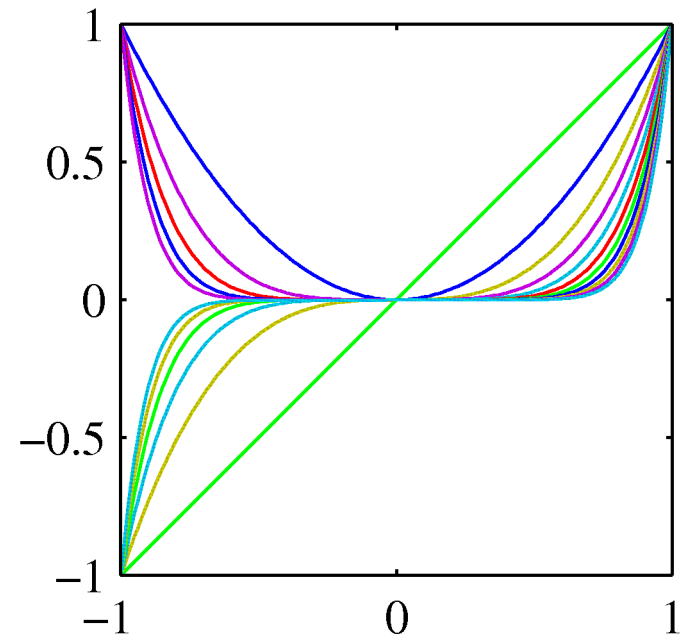
In the simplest case, we use linear basis functions : $\phi_d(\mathbf{x}) = x_d$.

Linear Basis Function Models (3)

Polynomial basis functions:

$$\phi_j(x) = x^j.$$

These are global; a small change in x affect all basis functions.

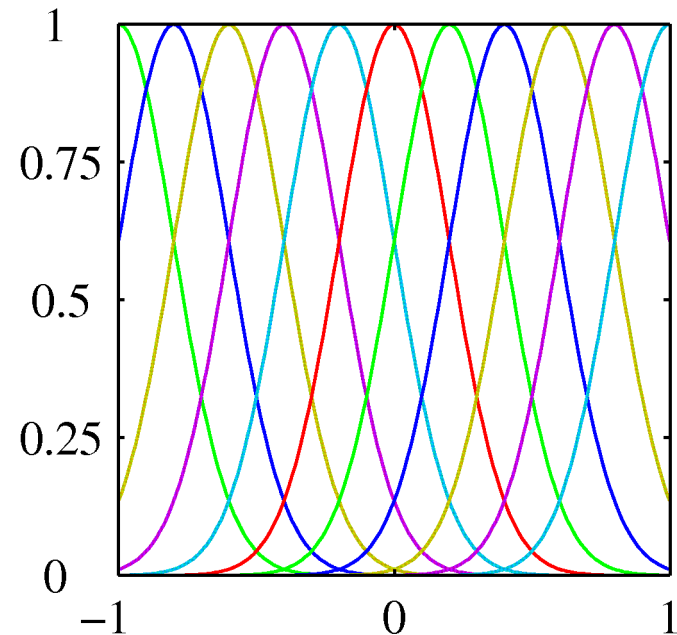


Linear Basis Function Models (4)

Gaussian basis functions:

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

These are local; a small change in x only affect nearby basis functions. μ_j and s control location and scale (width).



Linear Basis Function Models (5)

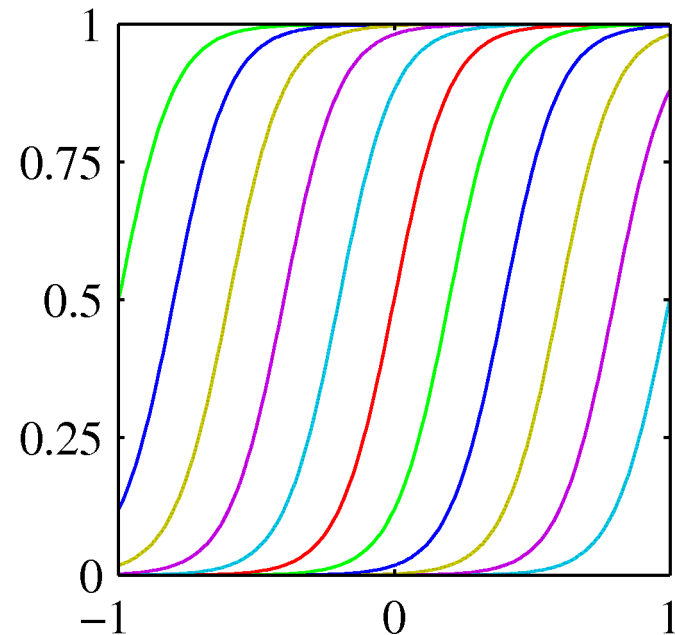
Sigmoidal basis functions:

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right)$$

where

$$\sigma(a) = \frac{1}{1 + \exp(-a)}.$$

Also these are local; a small change in x only affect nearby basis functions. μ_j and s control location and scale (slope).



Maximum Likelihood and Least Squares (1)

Assume observations from a deterministic function with added Gaussian noise:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \quad \text{where} \quad p(\epsilon|\beta) = \mathcal{N}(\epsilon|0, \beta^{-1})$$

which is the same as saying,

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

Given observed inputs, $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, and targets, $\mathbf{t} = [t_1, \dots, t_N]^T$, we obtain the likelihood function

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

Maximum Likelihood and Least Squares (2)

Taking the logarithm, we get

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{w}, \beta) &= \sum_{n=1}^N \ln \mathcal{N}(t_n | \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \\ &= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(\mathbf{w})\end{aligned}$$

where

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

is the sum-of-squares error.

Maximum Likelihood and Least Squares (3)

Computing the gradient and setting it to zero yields

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t}|\mathbf{w}, \beta) = \beta \sum_{n=1}^N \{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n)\} \boldsymbol{\phi}(\mathbf{x}_n)^T = \mathbf{0}.$$

Solving for \mathbf{w} , we get

$$\mathbf{w}_{\text{ML}} = \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^T \mathbf{t}$$

The Moore-Penrose
pseudo-inverse, $\boldsymbol{\Phi}^\dagger$.

where

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

Geometry of Least Squares

Consider

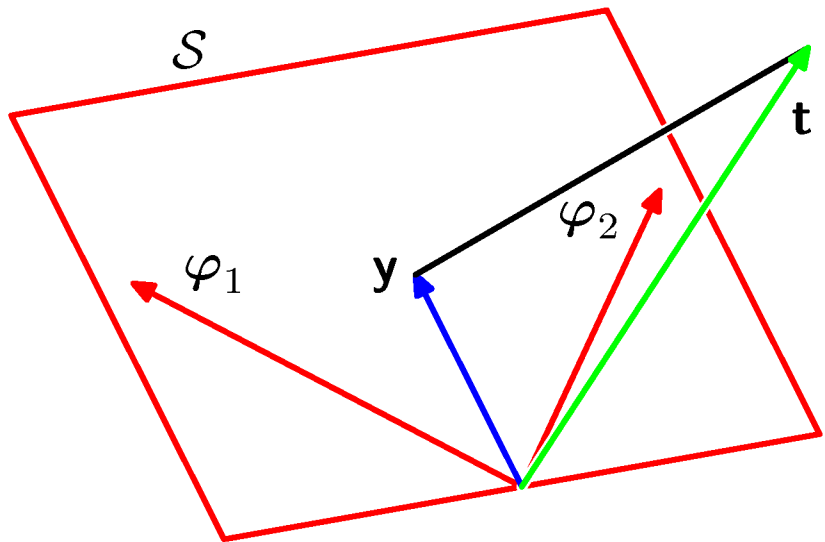
$$\mathbf{y} = \Phi \mathbf{w}_{\text{ML}} = [\varphi_1, \dots, \varphi_M] \mathbf{w}_{\text{ML}}.$$

$$\mathbf{y} \in \mathcal{S} \subseteq \mathcal{T} \quad \mathbf{t} \in \mathcal{T}$$

$\begin{array}{c} \uparrow \\ M\text{-dimensional} \end{array}$ $\begin{array}{c} \uparrow \\ N\text{-dimensional} \end{array}$

\mathcal{S} is spanned by $\varphi_1, \dots, \varphi_M$.

\mathbf{w}_{ML} minimizes the distance between \mathbf{t} and its orthogonal projection on \mathcal{S} , i.e. \mathbf{y} .



Sequential Learning

Data items considered one at a time (a.k.a. online learning); use stochastic (sequential) gradient descent:

$$\begin{aligned}\mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} - \eta \nabla E_n \\ &= \mathbf{w}^{(\tau)} + \eta (t_n - \mathbf{w}^{(\tau)\top} \phi(\mathbf{x}_n)) \phi(\mathbf{x}_n).\end{aligned}$$

This is known as the *least-mean-squares (LMS) algorithm*. Issue: how to choose η ?

Regularized Least Squares (1)

Consider the error function:

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Data term + Regularization term

With the sum-of-squares error function and a quadratic regularizer, we get

$$\frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$$

which is minimized by

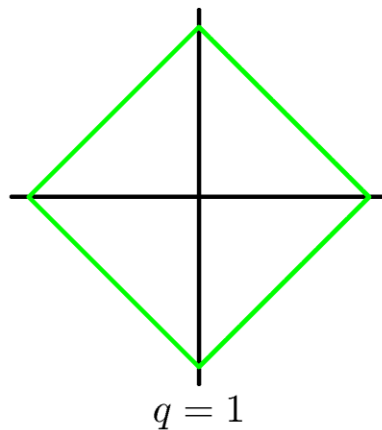
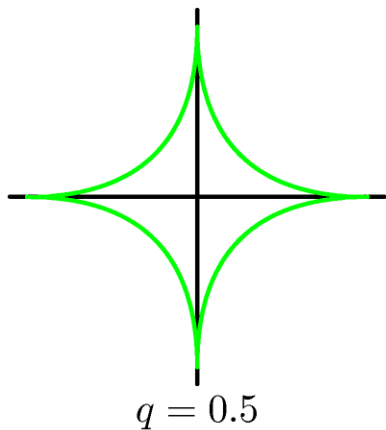
$$\mathbf{w} = \left(\lambda \mathbf{I} + \Phi^T \Phi \right)^{-1} \Phi^T \mathbf{t}.$$

λ is called the regularization coefficient.

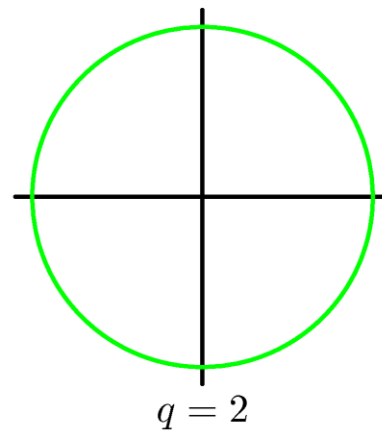
Regularized Least Squares (2)

With a more general regularizer, we have

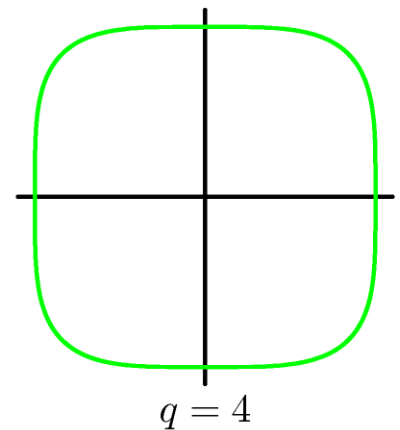
$$\frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \sum_{j=1}^M |w_j|^q$$



Lasso

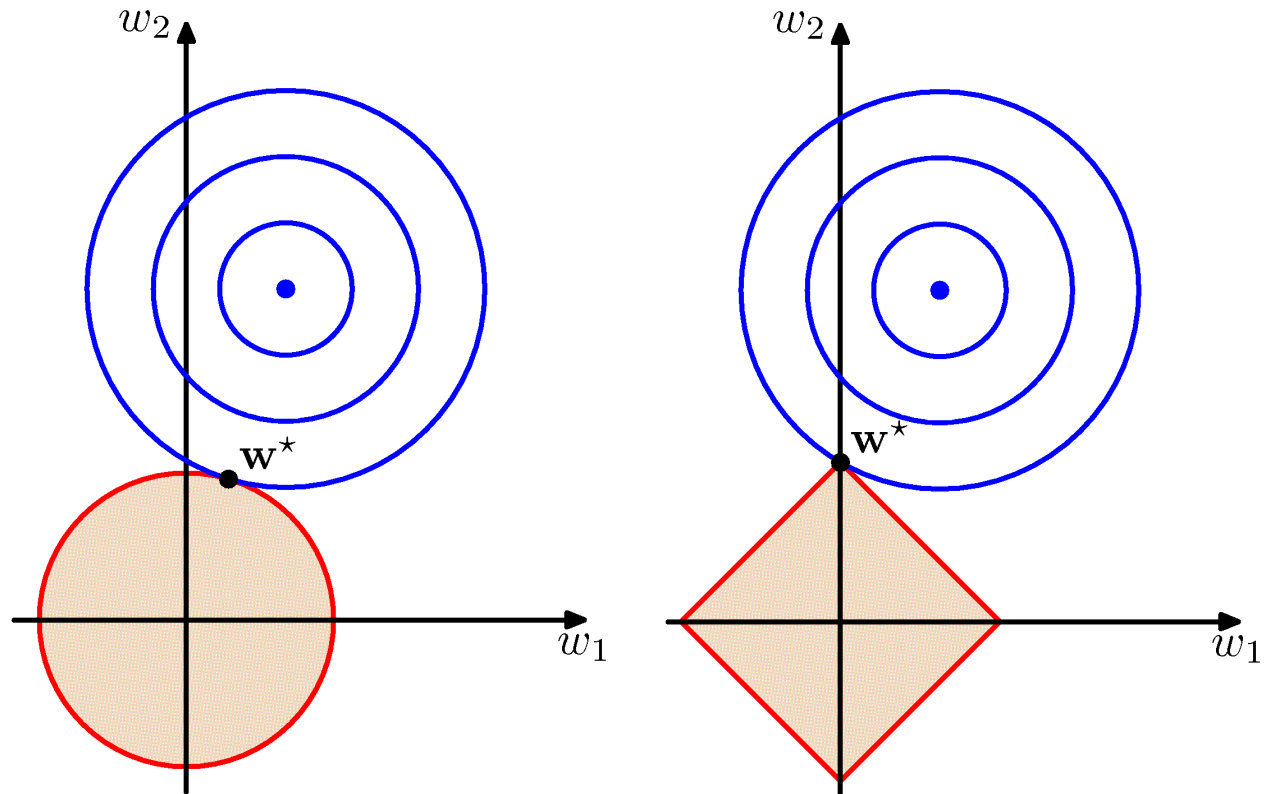


Quadratic



Regularized Least Squares (3)

Lasso tends to generate sparser solutions than a quadratic regularizer.



Multiple Outputs (1)

Analogously to the single output case we have:

$$\begin{aligned} p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) &= \mathcal{N}(\mathbf{t}|\mathbf{y}(\mathbf{W}, \mathbf{x}), \beta^{-1}\mathbf{I}) \\ &= \mathcal{N}(\mathbf{t}|\mathbf{W}^T\phi(\mathbf{x}), \beta^{-1}\mathbf{I}). \end{aligned}$$

Given observed inputs, $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, and targets, $\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_N]^T$, we obtain the log likelihood function

$$\begin{aligned} \ln p(\mathbf{T}|\mathbf{X}, \mathbf{W}, \beta) &= \sum_{n=1}^N \ln \mathcal{N}(\mathbf{t}_n|\mathbf{W}^T\phi(\mathbf{x}_n), \beta^{-1}\mathbf{I}) \\ &= \frac{NK}{2} \ln \left(\frac{\beta}{2\pi} \right) - \frac{\beta}{2} \sum_{n=1}^N \|\mathbf{t}_n - \mathbf{W}^T\phi(\mathbf{x}_n)\|^2. \end{aligned}$$

Multiple Outputs (2)

Maximizing with respect to \mathbf{W} , we obtain

$$\mathbf{W}_{\text{ML}} = \left(\Phi^T \Phi \right)^{-1} \Phi^T \mathbf{T}.$$

If we consider a single target variable, t_k , we see that

$$\mathbf{w}_k = \left(\Phi^T \Phi \right)^{-1} \Phi^T \mathbf{t}_k = \Phi^\dagger \mathbf{t}_k$$

where $\mathbf{t}_k = [t_{1k}, \dots, t_{Nk}]^T$, which is identical with the single output case.

The Bias-Variance Decomposition (1)

Recall the *expected squared loss*,

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x}) d\mathbf{x} + \underbrace{\iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) d\mathbf{x} dt}_{\text{noise}}$$

where

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int tp(t|\mathbf{x}) dt.$$

The second term of $\mathbb{E}[L]$ corresponds to the noise inherent in the random variable t .

What about the first term?

The Bias-Variance Decomposition (2)

Suppose we were given multiple data sets, each of size N . Any particular data set, \mathcal{D} , will give a particular function $y(\mathbf{x}; \mathcal{D})$. We then have

$$\begin{aligned} & \{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2 \\ &= \{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\ &= \{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\ &\quad + 2\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}. \end{aligned}$$

The Bias-Variance Decomposition (3)

Taking the expectation over \mathcal{D} yields

$$\begin{aligned} & \mathbb{E}_{\mathcal{D}} [\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2] \\ &= \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}} [\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2]}_{\text{variance}}. \end{aligned}$$



The Bias-Variance Decomposition (4)

Thus we can write

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

where

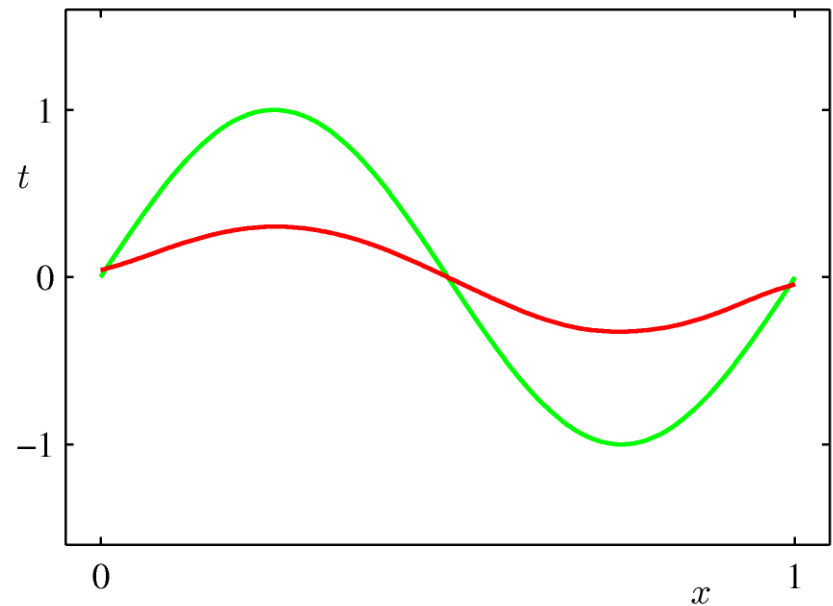
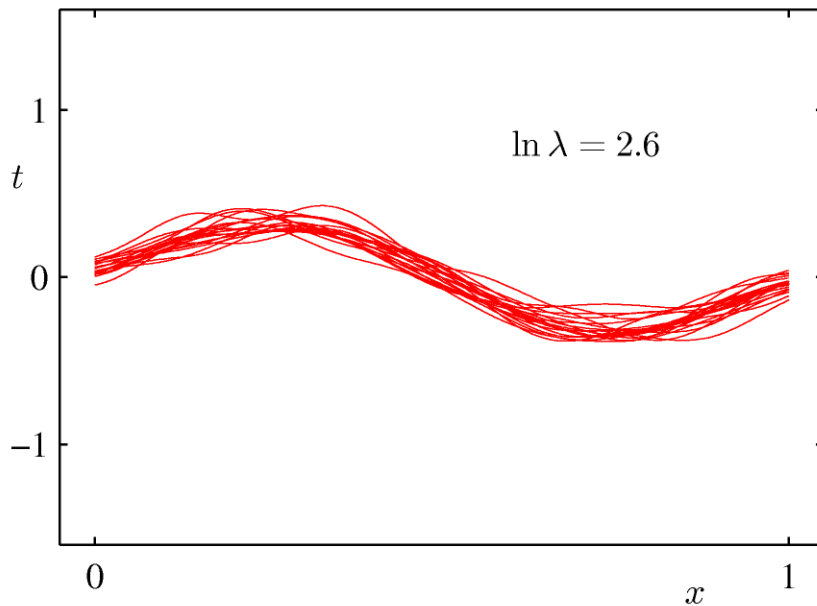
$$(\text{bias})^2 = \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x}) \, d\mathbf{x}$$

$$\text{variance} = \int \mathbb{E}_{\mathcal{D}} [\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2] p(\mathbf{x}) \, d\mathbf{x}$$

$$\text{noise} = \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) \, d\mathbf{x} \, dt$$

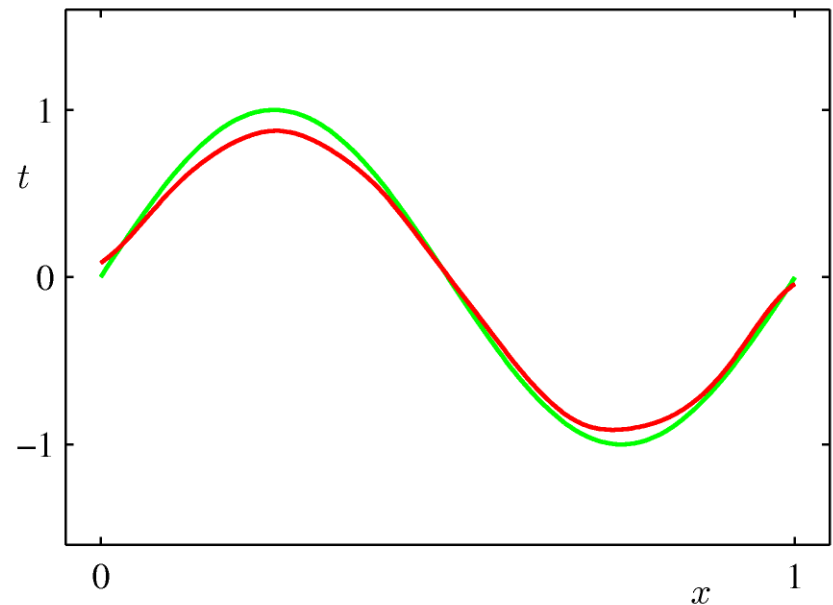
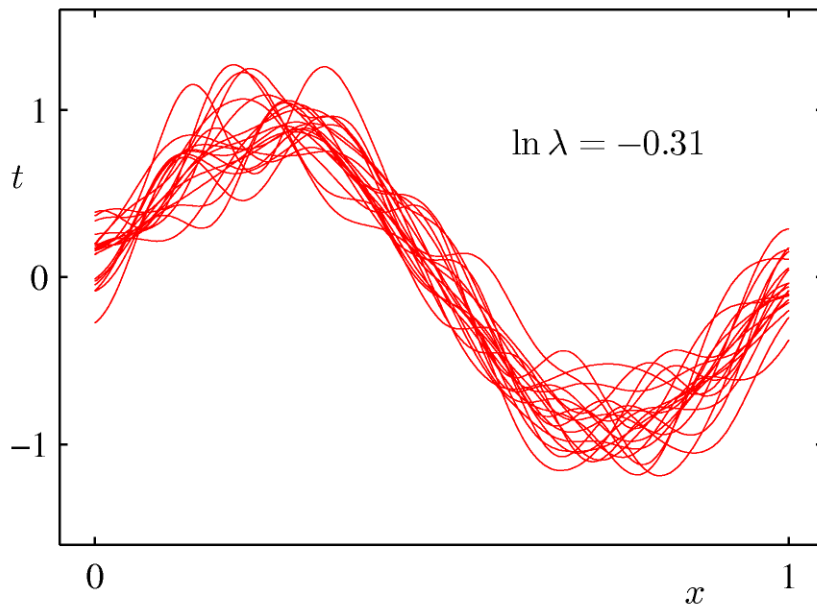
The Bias-Variance Decomposition (5)

Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ .



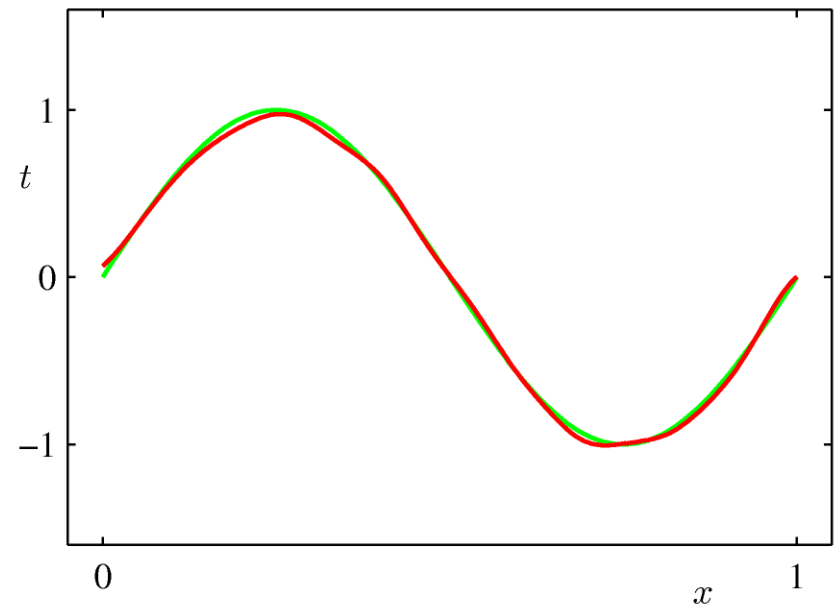
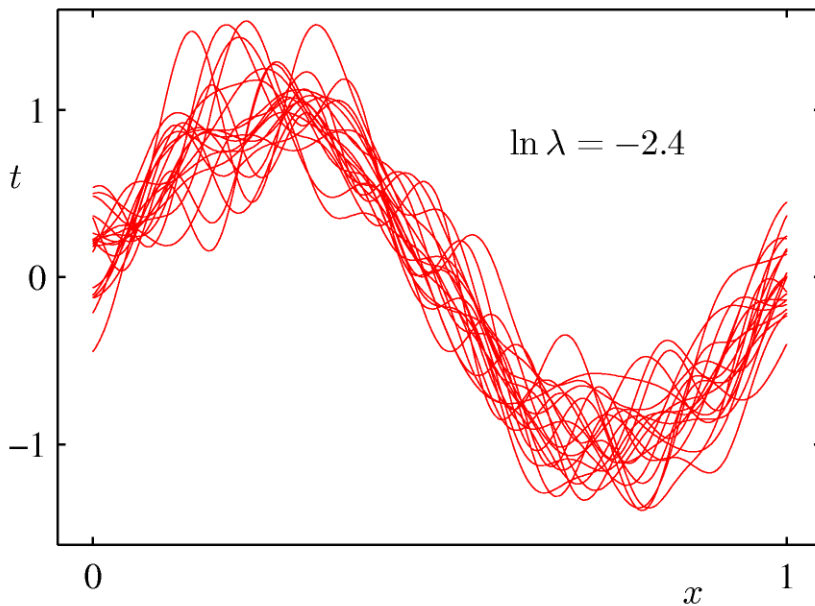
The Bias-Variance Decomposition (6)

Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ .



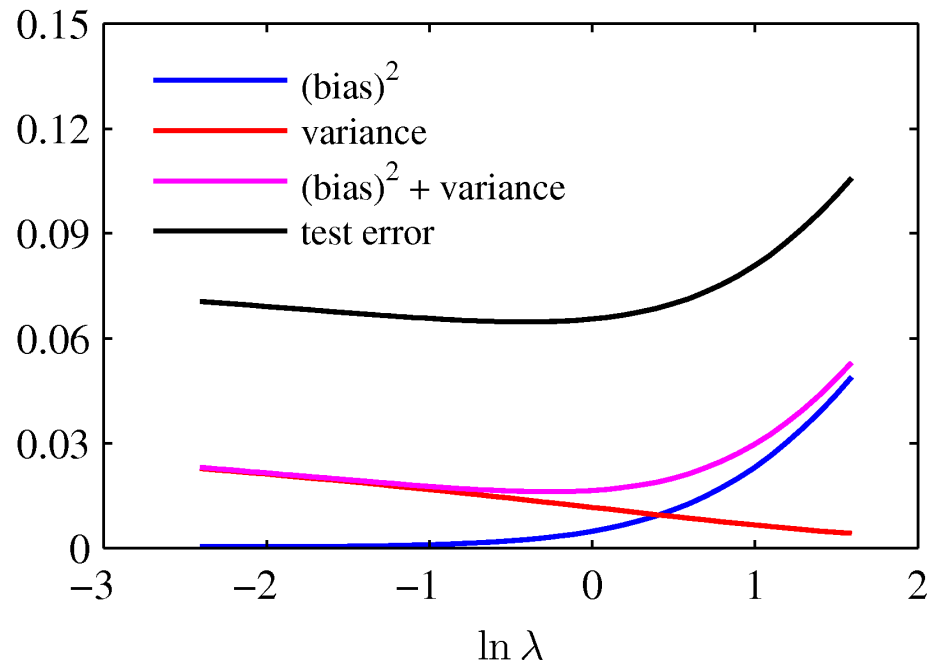
The Bias-Variance Decomposition (7)

Example: 25 data sets from the sinusoidal, varying the degree of regularization, λ .



The Bias-Variance Trade-off

From these plots, we note that an over-regularized model (large λ) will have a high bias, while an under-regularized model (small λ) will have a high variance.



Bayesian Linear Regression (1)

Define a conjugate prior over \mathbf{w}

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

Combining this with the likelihood function and using results for marginal and conditional Gaussian distributions, gives the posterior

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

where

$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N \left(\mathbf{S}_0^{-1} \mathbf{m}_0 + \beta \Phi^T \mathbf{t} \right) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta \Phi^T \Phi. \end{aligned}$$

Bayesian Linear Regression (2)

A common choice for the prior is

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

for which

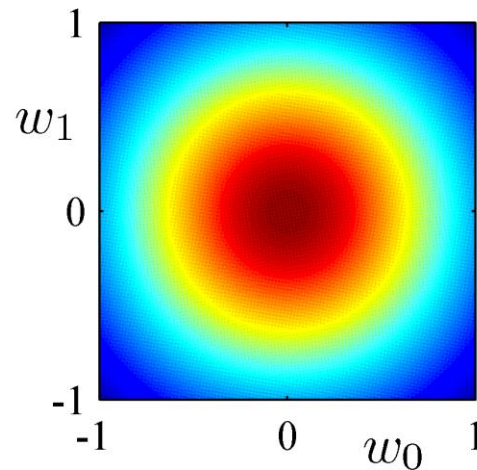
$$\begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \Phi^T \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \Phi^T \Phi. \end{aligned}$$

Next we consider an example ...

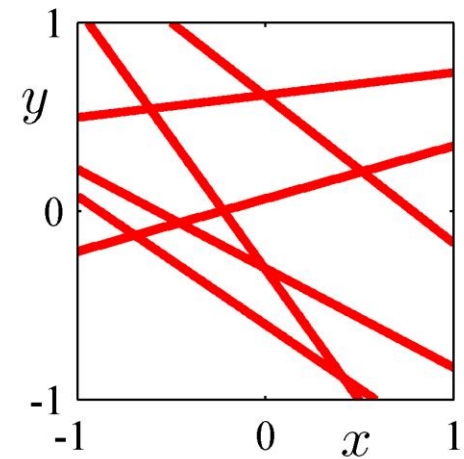
Bayesian Linear Regression (3)

0 data points observed

Prior



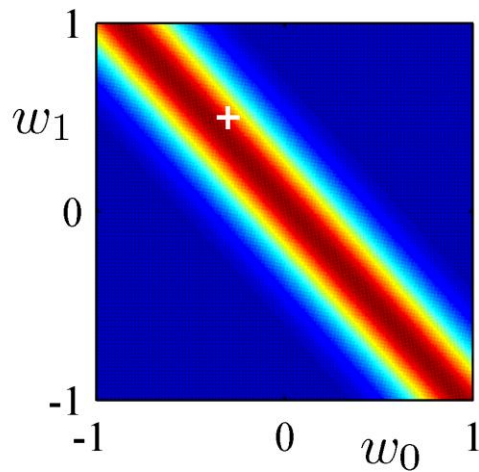
Data Space



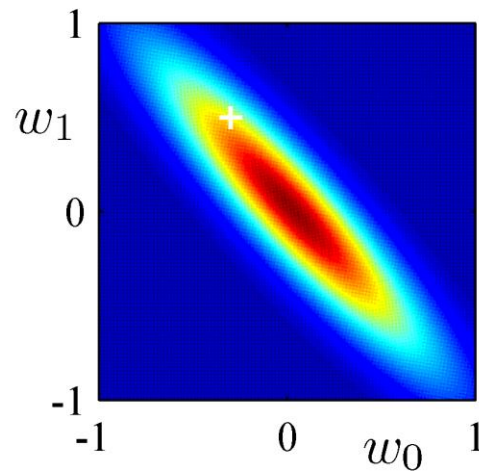
Bayesian Linear Regression (4)

1 data point observed

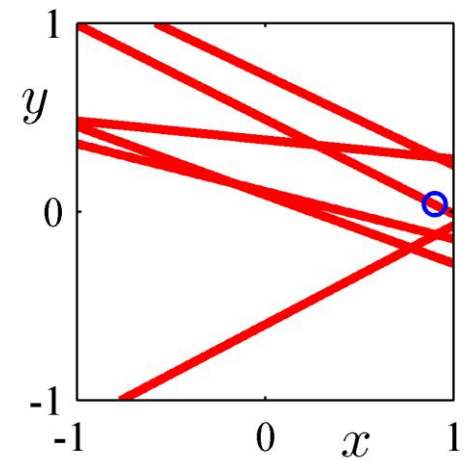
Likelihood



Posterior



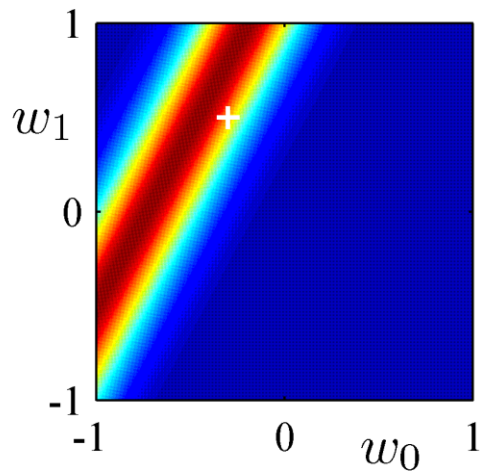
Data Space



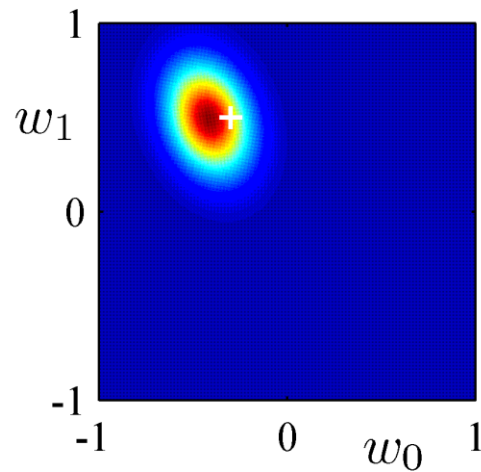
Bayesian Linear Regression (5)

2 data points observed

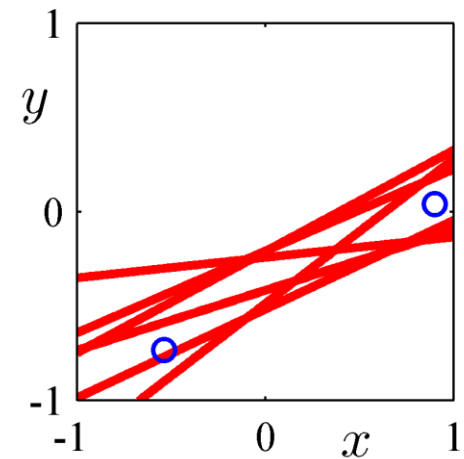
Likelihood



Posterior



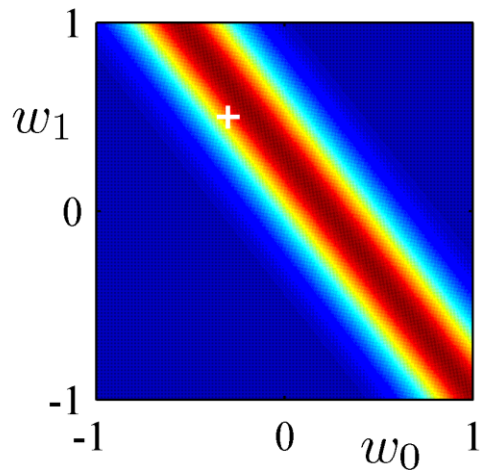
Data Space



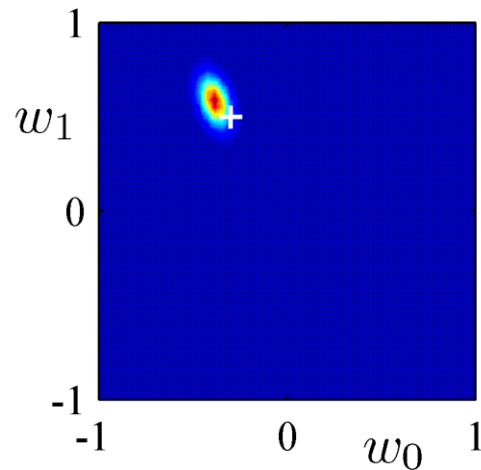
Bayesian Linear Regression (6)

20 data points observed

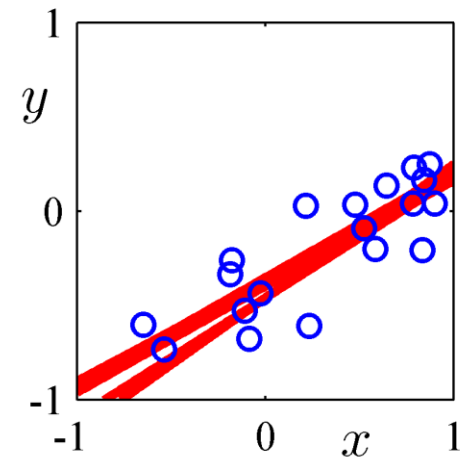
Likelihood



Posterior



Data Space



Predictive Distribution (1)

Predict t for new values of \mathbf{x} by integrating over \mathbf{w} :

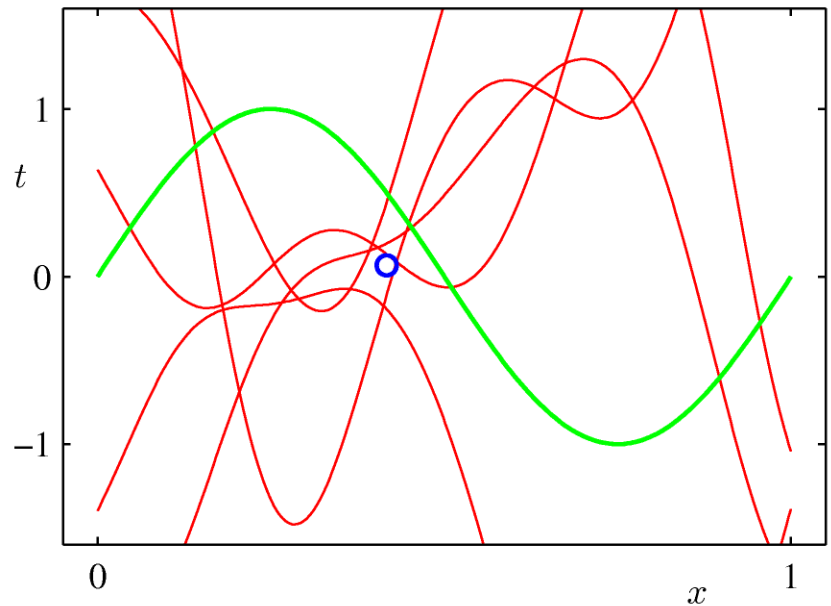
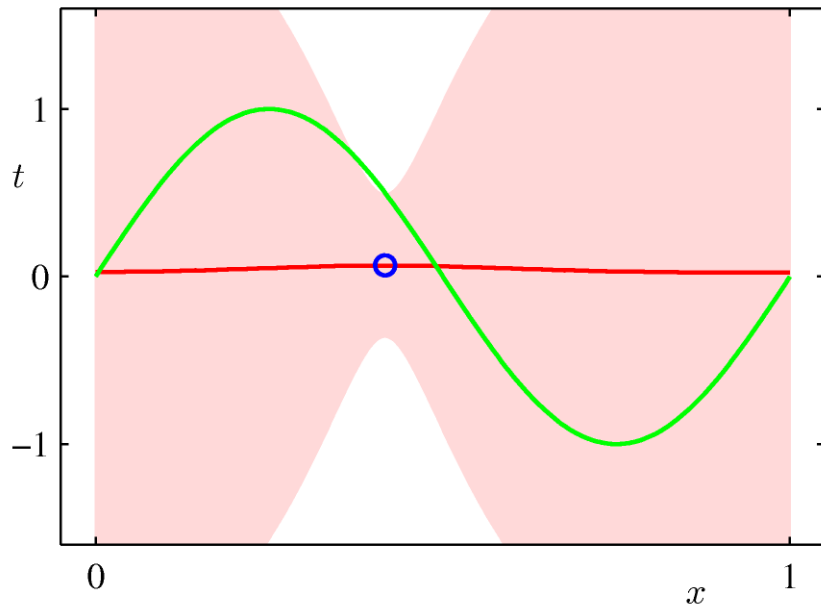
$$\begin{aligned} p(t|\mathbf{t}, \alpha, \beta) &= \int p(t|\mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \alpha, \beta) d\mathbf{w} \\ &= \mathcal{N}(t|\mathbf{m}_N^T \phi(\mathbf{x}), \sigma_N^2(\mathbf{x})) \end{aligned}$$

where

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

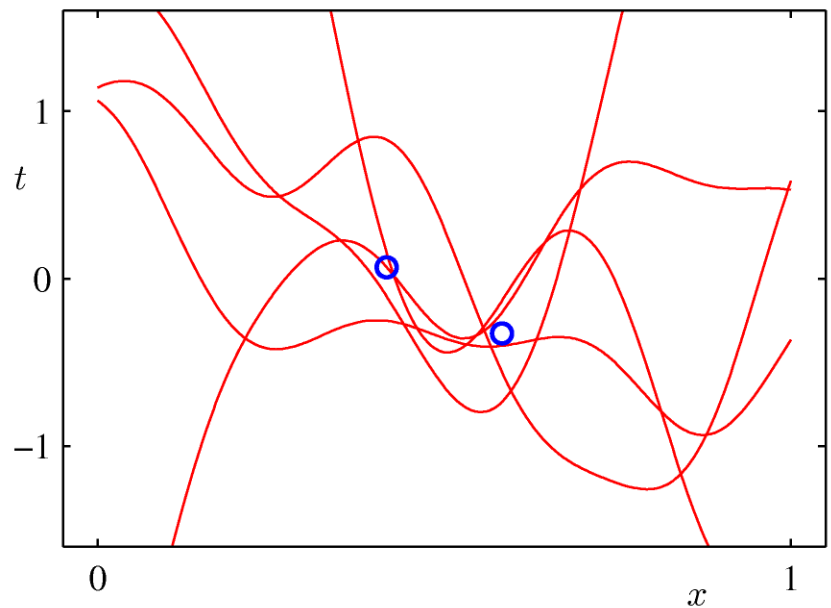
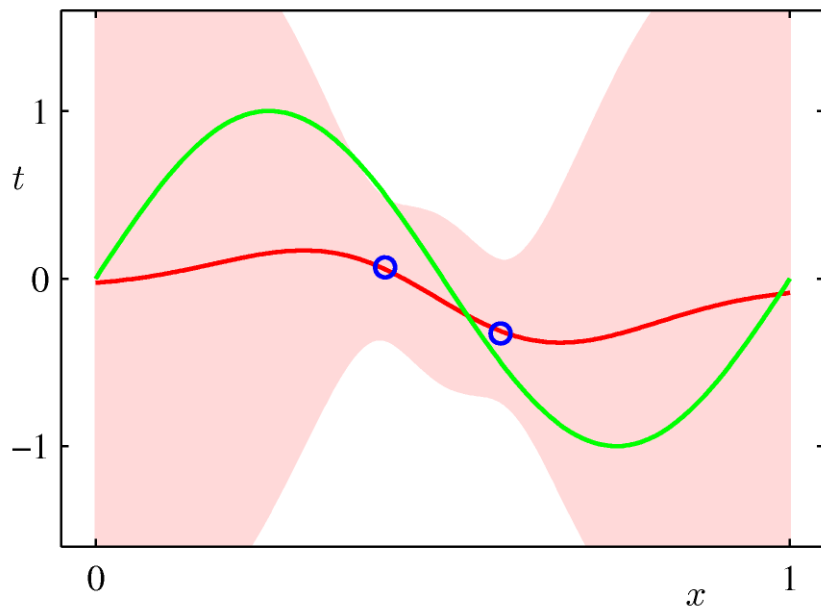
Predictive Distribution (2)

Example: Sinusoidal data, 9 Gaussian basis functions,
1 data point



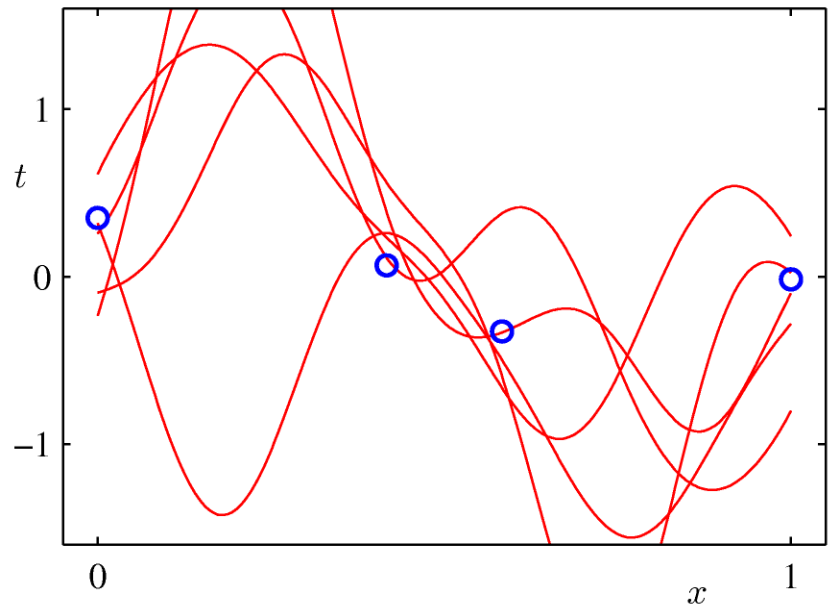
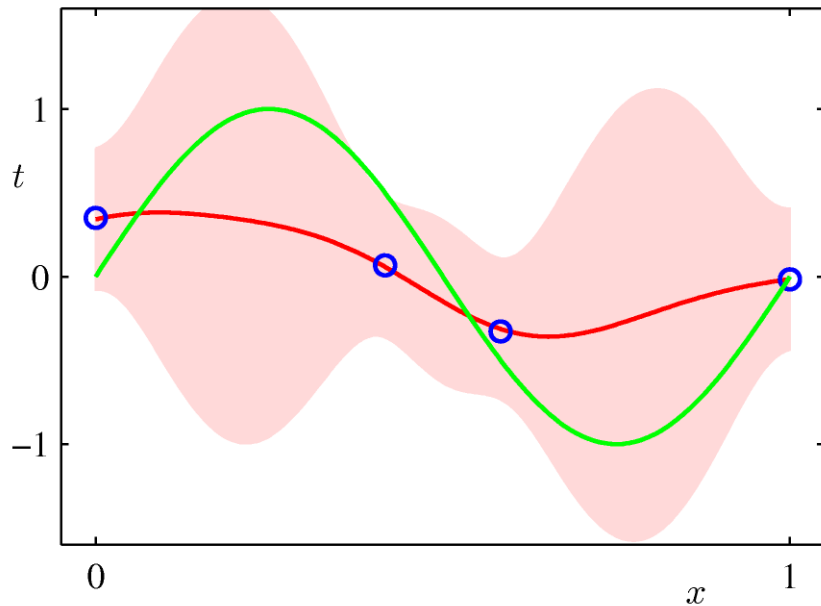
Predictive Distribution (3)

Example: Sinusoidal data, 9 Gaussian basis functions, 2 data points



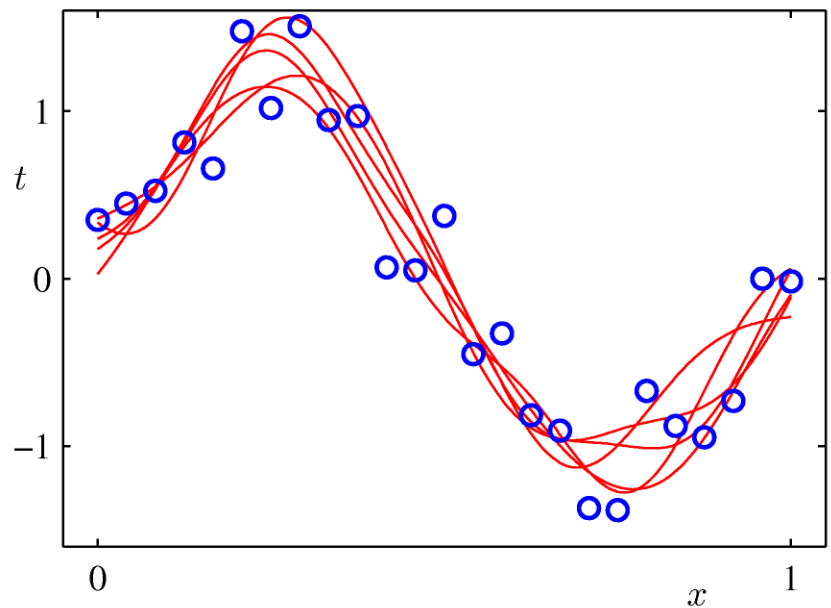
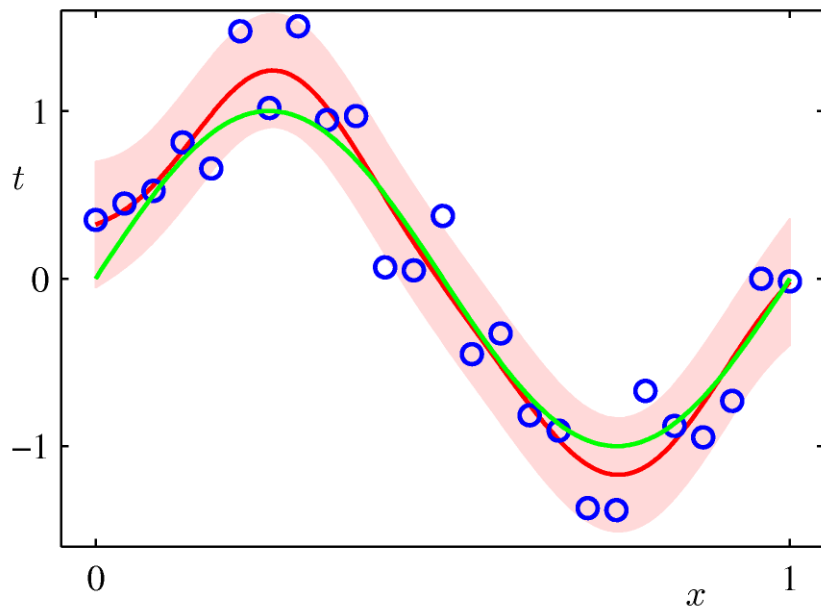
Predictive Distribution (4)

Example: Sinusoidal data, 9 Gaussian basis functions, 4 data points



Predictive Distribution (5)

Example: Sinusoidal data, 9 Gaussian basis functions, 25 data points



Equivalent Kernel (1)

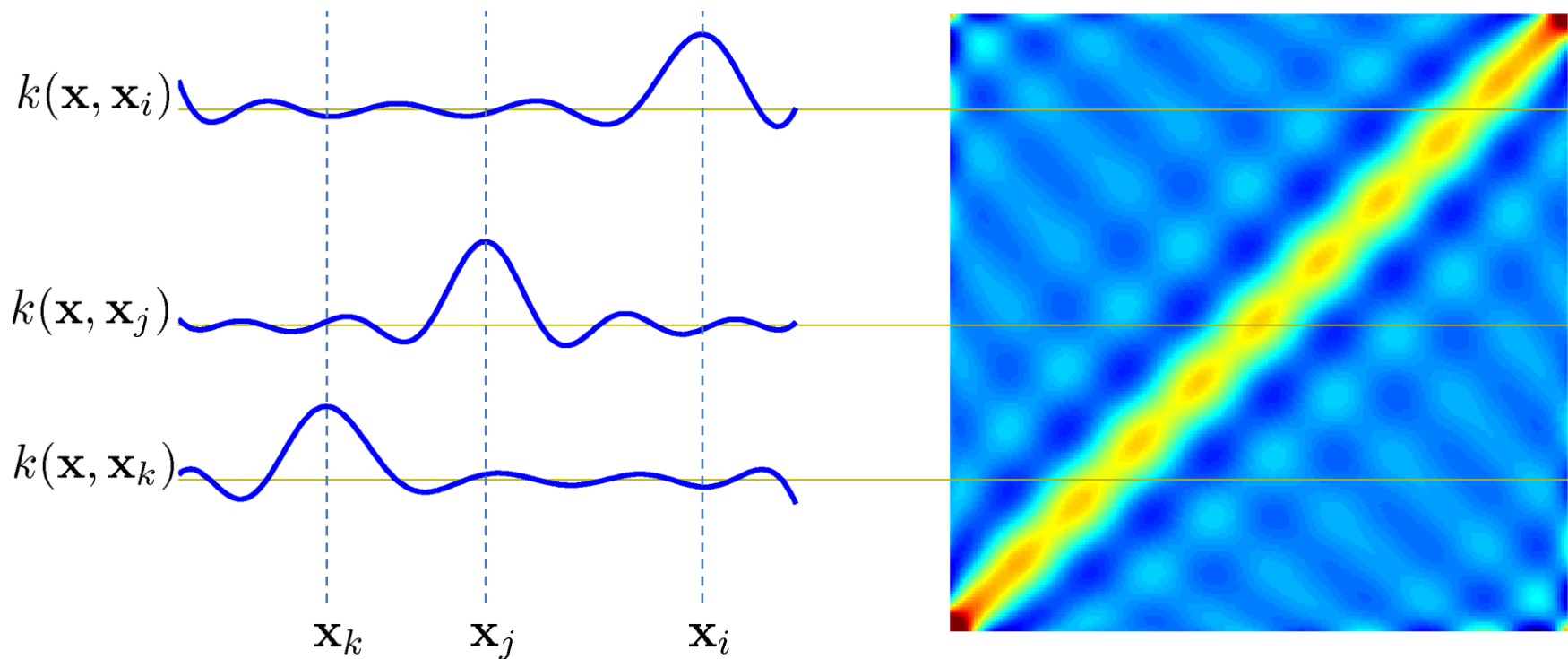
The predictive mean can be written

$$\begin{aligned}y(\mathbf{x}, \mathbf{m}_N) &= \mathbf{m}_N^T \phi(\mathbf{x}) = \beta \phi(\mathbf{x})^T \mathbf{S}_N \Phi^T \mathbf{t} \\ &= \sum_{n=1}^N \underbrace{\beta \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}_n)}_{k(\mathbf{x}, \mathbf{x}_n)} t_n \\ &= \sum_{n=1}^N k(\mathbf{x}, \mathbf{x}_n) t_n.\end{aligned}$$

Equivalent kernel or smoother matrix.

This is a weighted sum of the training data target values, t_n .

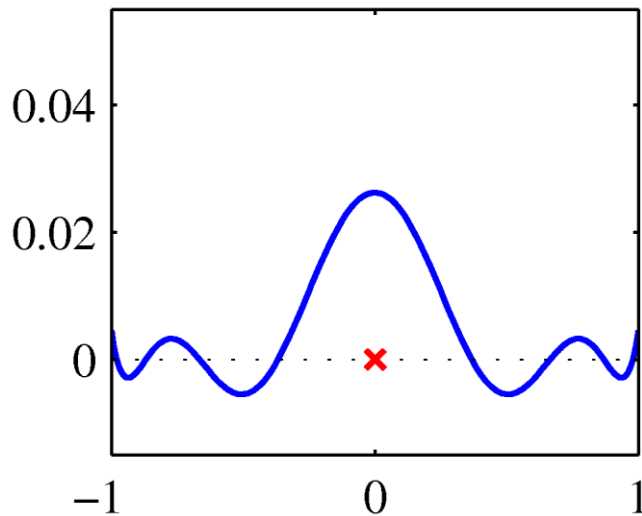
Equivalent Kernel (2)



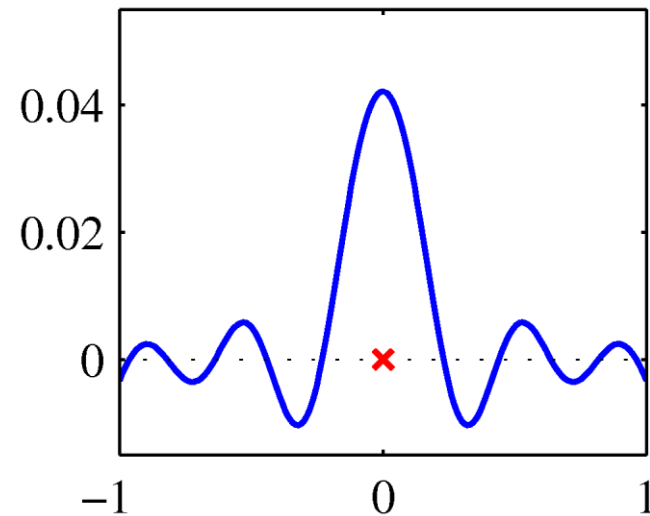
Weight of t_n depends on distance between \mathbf{x} and \mathbf{x}_n ;
nearby \mathbf{x}_n carry more weight.

Equivalent Kernel (3)

Non-local basis functions have local equivalent kernels:



Polynomial



Sigmoidal

Equivalent Kernel (4)

The kernel as a covariance function: consider

$$\begin{aligned}\text{cov}[y(\mathbf{x}), y(\mathbf{x}')] &= \text{cov}[\boldsymbol{\phi}(\mathbf{x})^T \mathbf{w}, \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}')] \\ &= \boldsymbol{\phi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}') = \beta^{-1} k(\mathbf{x}, \mathbf{x}').\end{aligned}$$

We can avoid the use of basis functions and define the kernel function directly, leading to *Gaussian Processes* (Chapter 6).

Equivalent Kernel (5)

$$\sum_{n=1}^N k(\mathbf{x}, \mathbf{x}_n) = 1$$

for all values of \mathbf{x} ; however, the equivalent kernel may be negative for some values of \mathbf{x} .

Like all kernel functions, the equivalent kernel can be expressed as an inner product:

$$k(\mathbf{x}, \mathbf{z}) = \boldsymbol{\psi}(\mathbf{x})^T \boldsymbol{\psi}(\mathbf{z})$$

where $\boldsymbol{\psi}(\mathbf{x}) = \beta^{1/2} \mathbf{S}_N^{1/2} \boldsymbol{\phi}(\mathbf{x})$.

Bayesian Model Comparison (1)

How do we choose the ‘right’ model?

Assume we want to compare models \mathcal{M}_i , $i=1, \dots, L$, using data \mathcal{D} ; this requires computing

$$p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_i).$$

Posterior

Prior

*Model evidence or
marginal likelihood*

Bayes Factor: ratio of evidence for two models

$$\frac{p(\mathcal{D}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_j)}$$

Bayesian Model Comparison (2)

Having computed $p(\mathcal{M}_i|\mathcal{D})$, we can compute the predictive (mixture) distribution

$$p(t|\mathbf{x}, \mathcal{D}) = \sum_{i=1}^L p(t|\mathbf{x}, \mathcal{M}_i, \mathcal{D})p(\mathcal{M}_i|\mathcal{D}).$$

A simpler approximation, known as *model selection*, is to use the model with the highest evidence.

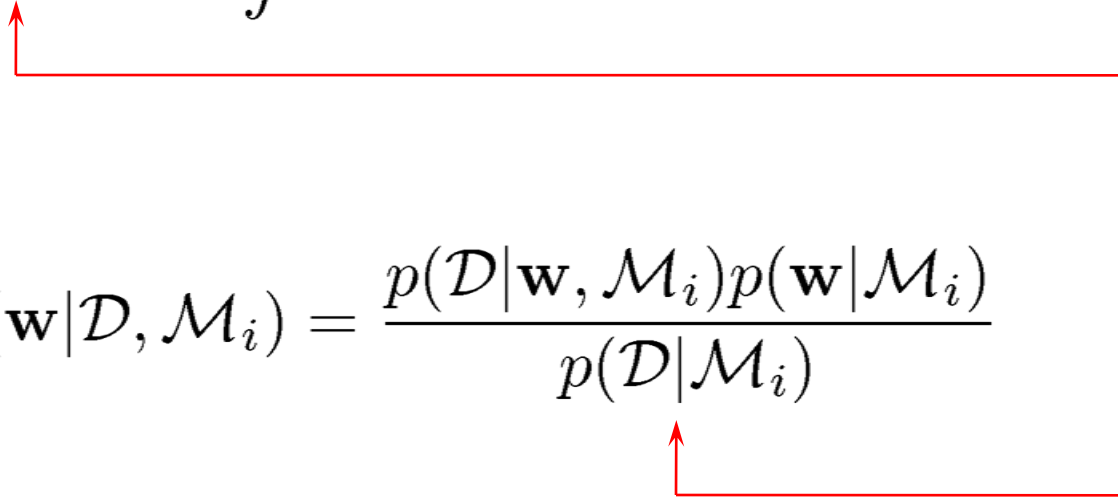
Bayesian Model Comparison (3)

For a model with parameters \mathbf{w} , we get the model evidence by marginalizing over \mathbf{w}

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w}, \mathcal{M}_i)p(\mathbf{w}|\mathcal{M}_i) d\mathbf{w}.$$

Note that

$$p(\mathbf{w}|\mathcal{D}, \mathcal{M}_i) = \frac{p(\mathcal{D}|\mathbf{w}, \mathcal{M}_i)p(\mathbf{w}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_i)}$$

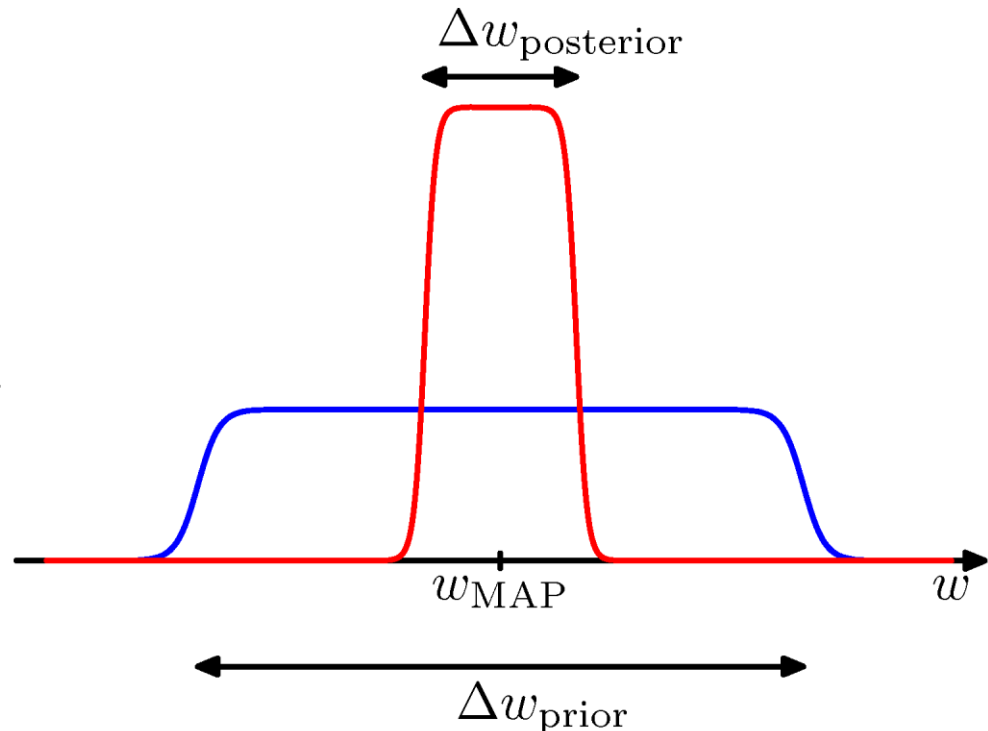


Bayesian Model Comparison (4)

For a given model with a single parameter, w , consider the approximation

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w) dw$$
$$\simeq p(\mathcal{D}|w_{\text{MAP}}) \frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}$$

where the posterior is assumed to be sharply peaked.



Bayesian Model Comparison (5)

Taking logarithms, we obtain

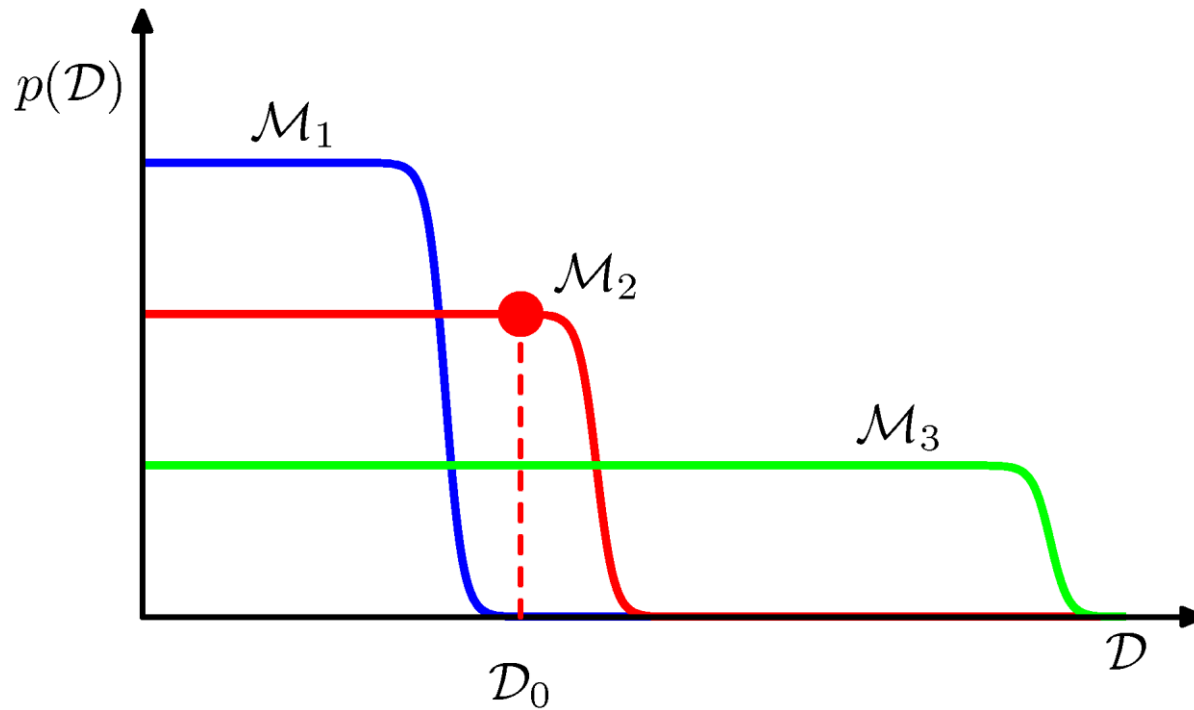
$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|w_{\text{MAP}}) + \underbrace{\ln \left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}} \right)}_{\text{Negative}}.$$

With M parameters, all assumed to have the same ratio $\Delta w_{\text{posterior}}/\Delta w_{\text{prior}}$, we get

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\mathbf{w}_{\text{MAP}}) + \underbrace{M \ln \left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}} \right)}_{\text{Negative and linear in } M}.$$

Bayesian Model Comparison (6)

Matching data and model complexity



The Evidence Approximation (1)

The fully Bayesian predictive distribution is given by

$$p(t|\mathbf{t}) = \iiint p(t|\mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \alpha, \beta) p(\alpha, \beta|\mathbf{t}) d\mathbf{w} d\alpha d\beta$$

but this integral is intractable. Approximate with

$$p(t|\mathbf{t}) \simeq p\left(t|\mathbf{t}, \hat{\alpha}, \hat{\beta}\right) = \int p\left(t|\mathbf{w}, \hat{\beta}\right) p\left(\mathbf{w}|\mathbf{t}, \hat{\alpha}, \hat{\beta}\right) d\mathbf{w}$$

where $(\hat{\alpha}, \hat{\beta})$ is the mode of $p(\alpha, \beta|\mathbf{t})$, which is assumed to be sharply peaked; a.k.a. *empirical Bayes, type II* or *generalized maximum likelihood, or evidence approximation*.

The Evidence Approximation (2)

From Bayes' theorem we have

$$p(\alpha, \beta | \mathbf{t}) \propto p(\mathbf{t} | \alpha, \beta) p(\alpha, \beta)$$

and if we assume $p(\alpha, \beta)$ to be flat we see that

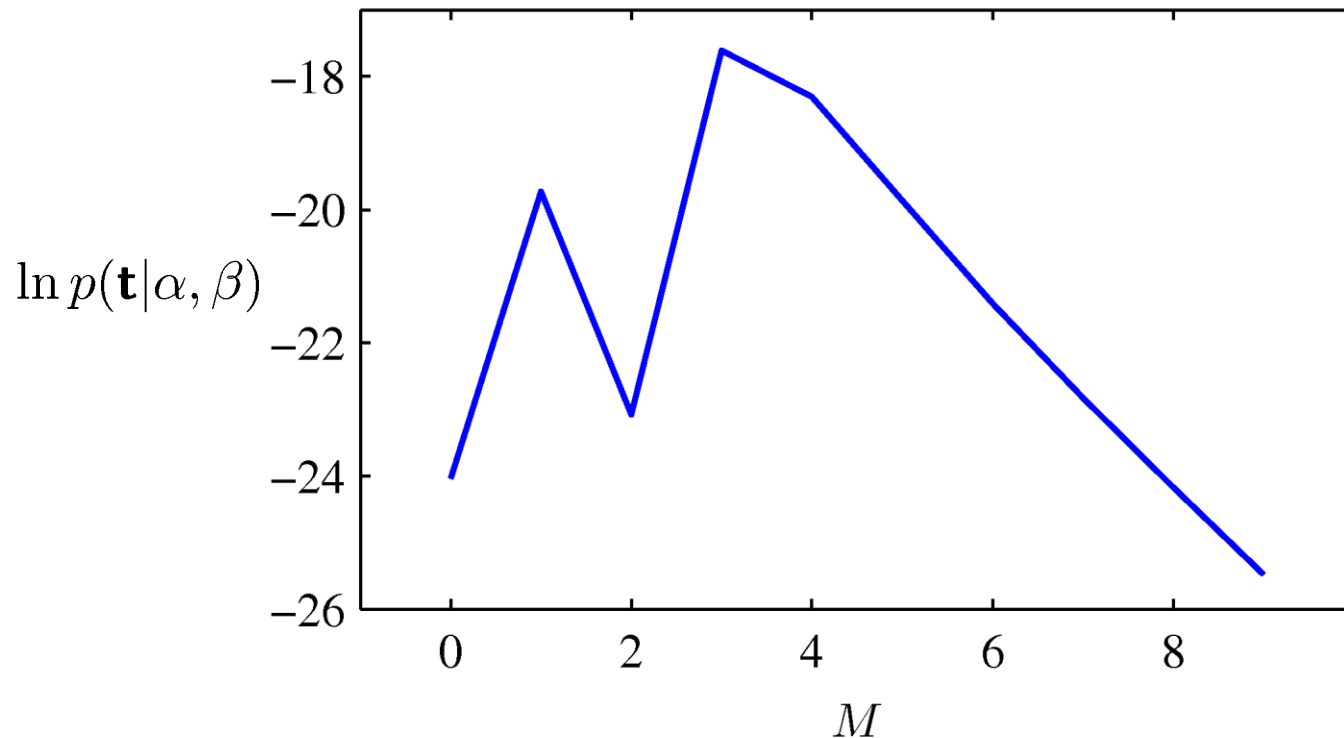
$$\begin{aligned} p(\alpha, \beta | \mathbf{t}) &\propto p(\mathbf{t} | \alpha, \beta) \\ &= \int p(\mathbf{t} | \mathbf{w}, \beta) p(\mathbf{w} | \alpha) d\mathbf{w}. \end{aligned}$$

General results for Gaussian integrals give

$$\ln p(\mathbf{t} | \alpha, \beta) = \frac{M}{2} \ln \alpha + \frac{N}{2} \ln \beta - E(\mathbf{m}_N) + \frac{1}{2} \ln |\mathbf{S}_N| - \frac{N}{2} \ln(2\pi).$$

The Evidence Approximation (3)

Example: sinusoidal data, M^{th} degree polynomial,
 $\alpha = 5 \times 10^{-3}$



Maximizing the Evidence Function (1)

To maximise $\ln p(\mathbf{t}|\alpha, \beta)$ w.r.t. α and β , we define the eigenvector equation

$$\left(\beta\Phi^T\Phi\right)\mathbf{u}_i = \lambda_i\mathbf{u}_i.$$

Thus

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

has eigenvalues $\lambda_i + \alpha$.

Maximizing the Evidence Function (2)

We can now differentiate $\ln p(\mathbf{t}|\alpha, \beta)$ w.r.t. α and β , and set the results to zero, to get

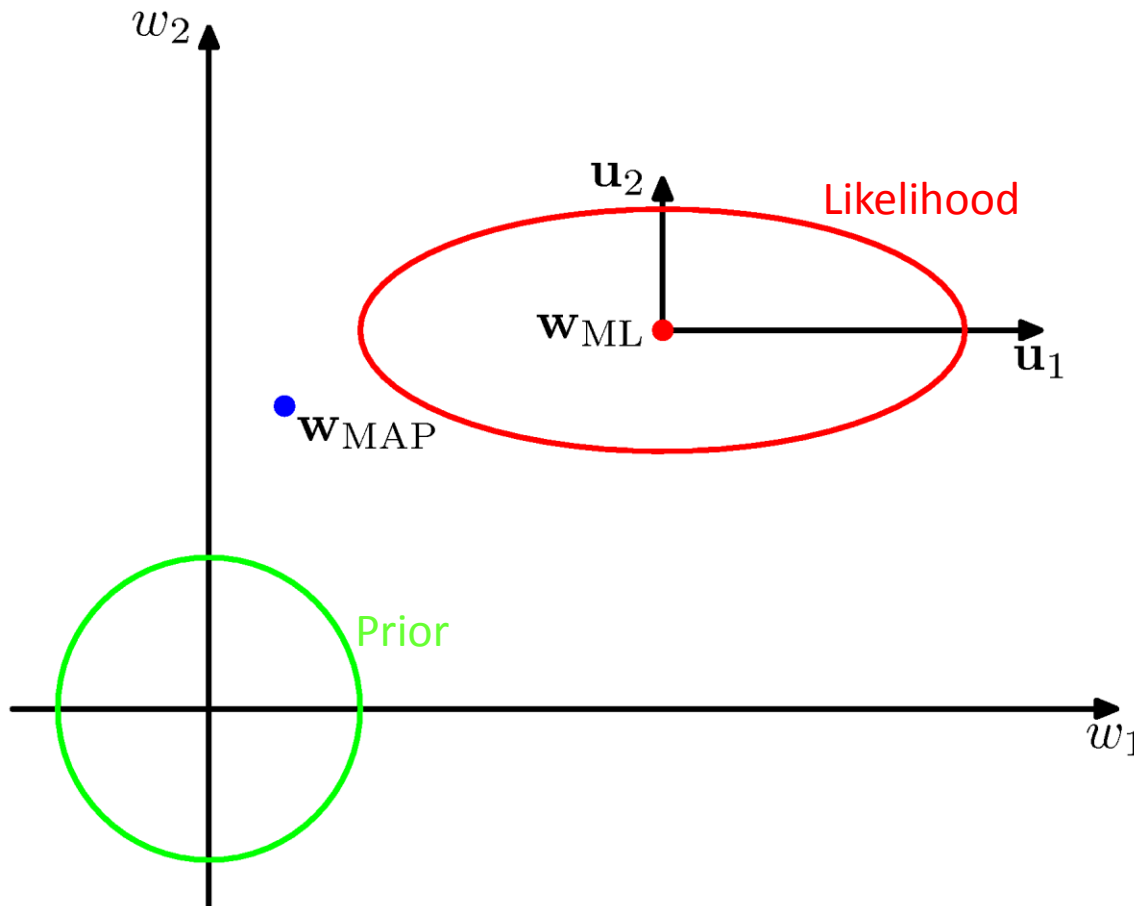
$$\alpha = \frac{\gamma}{\mathbf{m}_N^T \mathbf{m}_N}$$
$$\frac{1}{\beta} = \frac{1}{N - \gamma} \sum_{n=1}^N \{t_n - \mathbf{m}_N^T \phi(\mathbf{x}_n)\}^2$$

where

$$\gamma = \sum_i \frac{\lambda_i}{\alpha + \lambda_i}.$$

N.B. γ depends on both α and β .

Effective Number of Parameters (3)



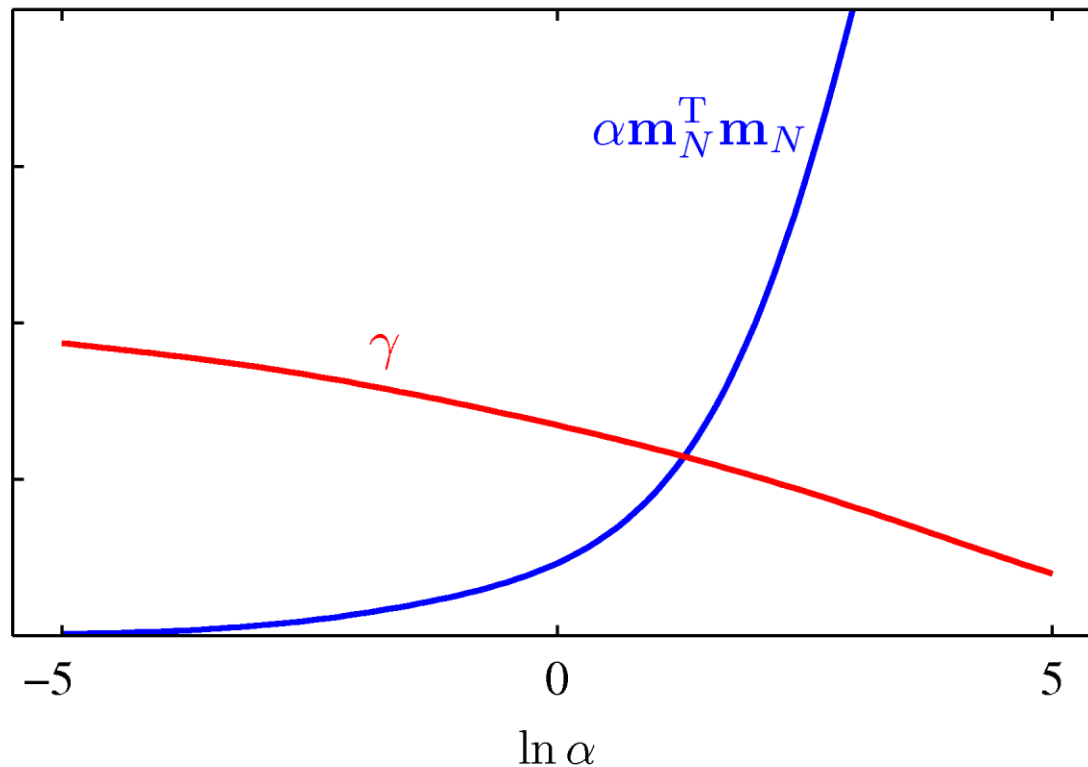
$\lambda_1 \ll \alpha$
 w_1 is not well
determined by the
likelihood

$\lambda_2 \gg \alpha$
 w_2 is well determined
by the likelihood

γ is the number of well
determined parameters

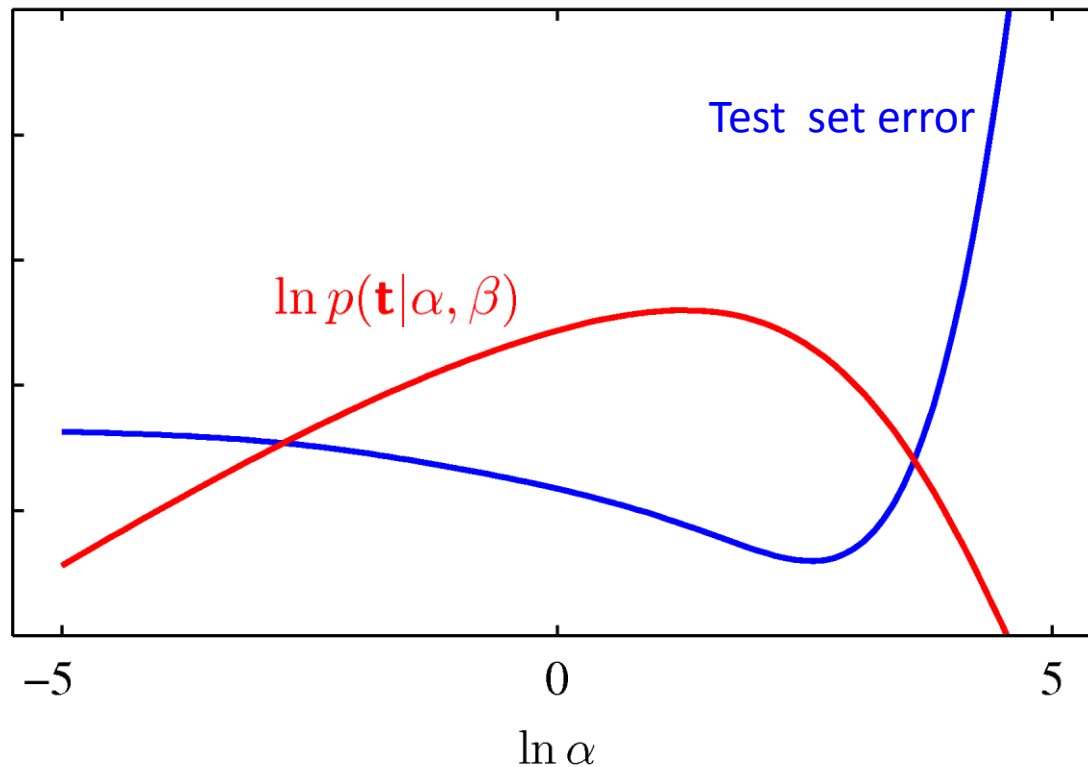
Effective Number of Parameters (2)

Example: sinusoidal data, 9 Gaussian basis functions,
 $\beta = 11.1$.



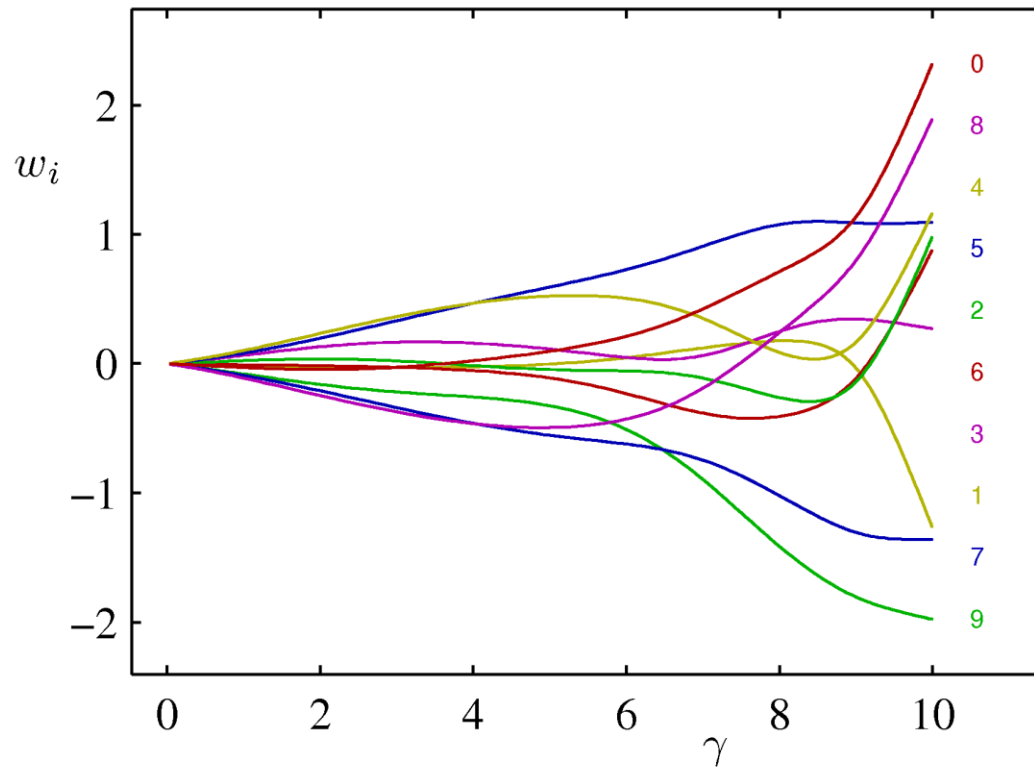
Effective Number of Parameters (3)

Example: sinusoidal data, 9 Gaussian basis functions,
 $\beta = 11.1$.



Effective Number of Parameters (4)

Example: sinusoidal data, 9 Gaussian basis functions,
 $\beta = 11.1$.



Effective Number of Parameters (5)

In the limit $N \gg M$, $\gamma = M$ and we can consider using the easy-to-compute approximation

$$\alpha = \frac{M}{\mathbf{m}_N^T \mathbf{m}_N}$$
$$\frac{1}{\beta} = \frac{1}{N} \sum_{n=1}^N \{t_n - \mathbf{m}_N^T \phi(\mathbf{x}_n)\}^2.$$



Limitations of Fixed Basis Functions

- M basis function along each dimension of a D -dimensional input space requires M^D basis functions: the curse of dimensionality.
 - In later chapters, we shall see how we can get away with fewer basis functions, by choosing these using the training data.
-