Linear Models for Regression

Chapter 3

Prof. Reza Azadeh

University of Massachusetts Lowell

Supervised Learning - Regression

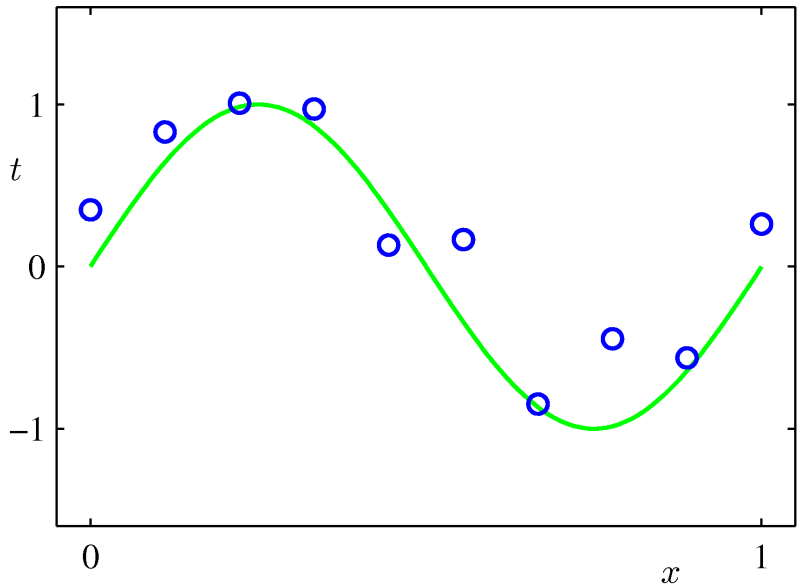
Goal: predict value of one or more continuous **target** variables t given the value of a D-dimensional vector $\mathbf{x}$ of **input** variables.

Problem: Given a training dataset comprising N observations $\{\mathbf{x}_n\}$, where $n = 1, \ldots, N$, together with the corresponding target values $\{t_n\}$, the goal is to predict the value of t fro a new value of $\mathbf{x}$.

- deterministic solution: $y(\mathbf{x})$
- predictive distribution: $p(t|\mathbf{x})$

Linear Basis Function Models

Example: Polynomial Curve Fitting



$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Linear Basis Function Models

Generally

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

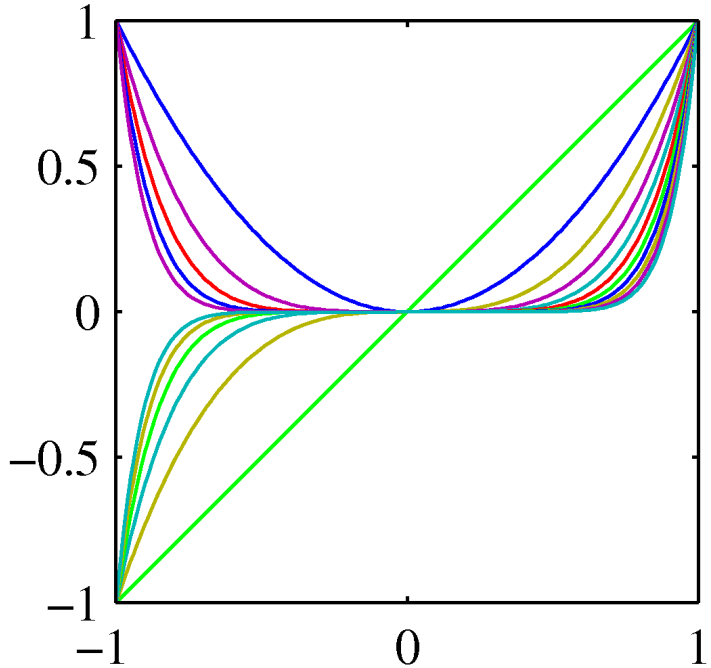
where $\mathbf{x} = (x_1, \ldots, x_D)^\top$, $\mathbf{w} = (w_0, \ldots, w_{M-1})^\top$, and $\boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^\top$.

In the simplest case, we can have $\phi_j(\mathbf{x}) = x_j$, which makes a linear combination of the input variables.

Basis Functions: Polynomial

Polynomial basis functions:

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

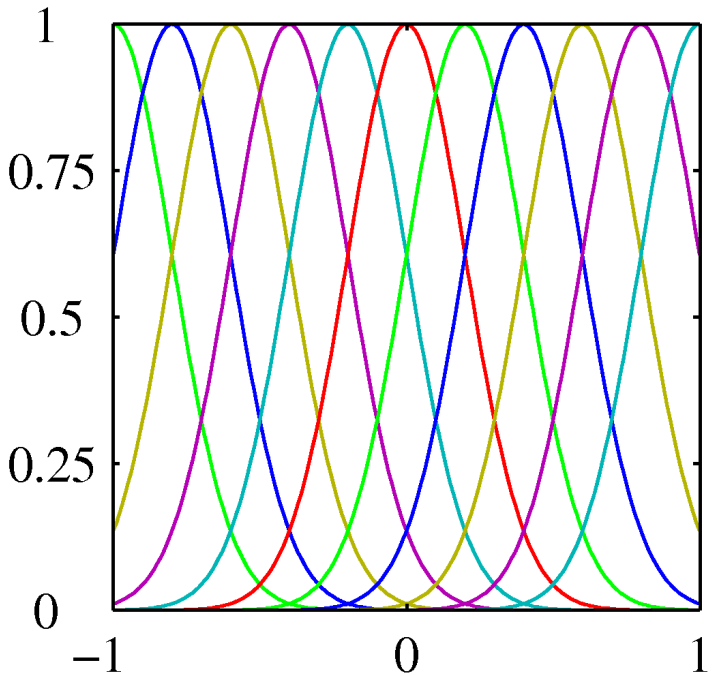


- these are global; a small change in x affects all basis functions.
- can be extended to spline basis

Basis Functions: Gaussian

Gaussian basis functions:

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



- these are local; a small change in x only affects nearby basis functions.
- μ_j and s control the location and scale (width), respectively.

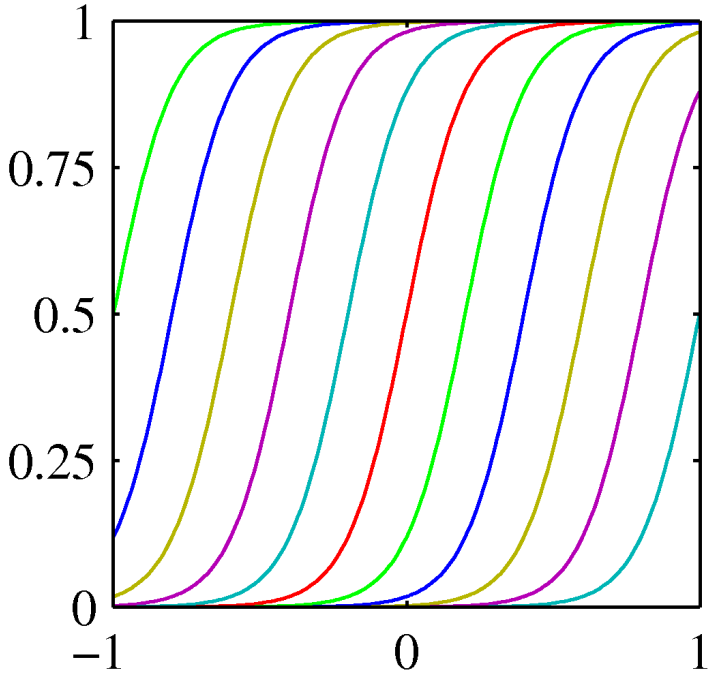
Basis functions: Sigmoidal

Sigmoidal basis functions:

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

where $\sigma(a)$ is the logistic sigmoid function defined as:

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



- these are also local; a small change in x only affects nearby basis functions.
- μ_j and s control the location and scale (slope), respectively.

Basis functions: others

- tanh basis functions where $\tanh(a) = 2\sigma(a) - 1$.
- Fourier basis functions
- ...

Maximum Likelihood & Least Squares

Assume observation from a deterministic function with added Gaussian noise:

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

which is the same as saying

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

- because Gaussian, the conditional mean simplifies to $\mathbf{E}[t|\mathbf{x}] = \int tp(t|\mathbf{x})dt = y(\mathbf{x}, \mathbf{w})$.
- Note: Gaussian noise assumption implies that the conditional distribution is unimodal which might not be a correct assumption for some applications.

Maximum Likelihood & Least Squares

Consider dataset:

$$\text{observed inputs: } \mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$$
$$\text{targets: } \mathbf{t} = [t_1, \ldots, t_N]^\top$$

with the assumption that these data points are drawn independently from the distribution $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}, \beta^{-1})$, we can write the likelihood function as

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

Note: we used the linear model here to represent $y(\mathbf{x}, \mathbf{w})$.

Maximum Likelihood & Least Squares

Take logarithm:

$$\ln p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}), \beta^{-1})$$
$$= \frac{N}{2} \ln \beta - \frac{N}{2} \ln (2\pi) - \beta E_D(\mathbf{w})$$

where

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

is the sum-of-squares error.

Maximum Likelihood & Least Squares

Maximizing w.r.t $\mathbf{w}$: Compute gradient and setting it to zero,

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

Solving for $\mathbf{w}$, we get

$$\mathbf{w}_{\mathrm{ML}} = \boldsymbol{\Phi}^\dagger \mathbf{t}$$

where $\boldsymbol{\Phi}^\dagger = (\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{\Phi} \in \mathbf{R}^{M \times N}$ and $\Phi_{nj} = \phi_j(\mathbf{x}_n)$.

Maximum Likelihood & Least Squares

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

is called the *design matrix*.

Maximum Likelihood & Least Squares

We can also maximize w.r.t noise β,

$$\frac{1}{\beta_{\text{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{t_n - \mathbf{w}_{\text{ML}}^{\top}\phi(\mathbf{x}_n)\}^2$$

We see that the inverse of the noise precision is given by the residual variance of the target values around the regression function.

Sequential Learning

- Batch solution: we use all data at once; costly and not applicable in real-time.
- Online learning: data items are used one at a time

Use Stochastic 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 **Least-Mean-Squares (LMS)** algorithm. Note: the learning rate, η needs to be chosen with care to ensure that the algorithm converges.

Regularized Least Squares

Consider the error function

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

$$\text{data term} + \text{regularization term}$$

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

$$\begin{aligned} E &= E_D(\mathbf{w}) + \lambda E_w \\ &= \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w} \end{aligned}$$

where λ is the regularization coefficient.

Regularized Least Squares

We can minimize the error function by calculating the gradient and setting it equal to 0, $\nabla_{\mathbf{w}} E = 0$ which results in

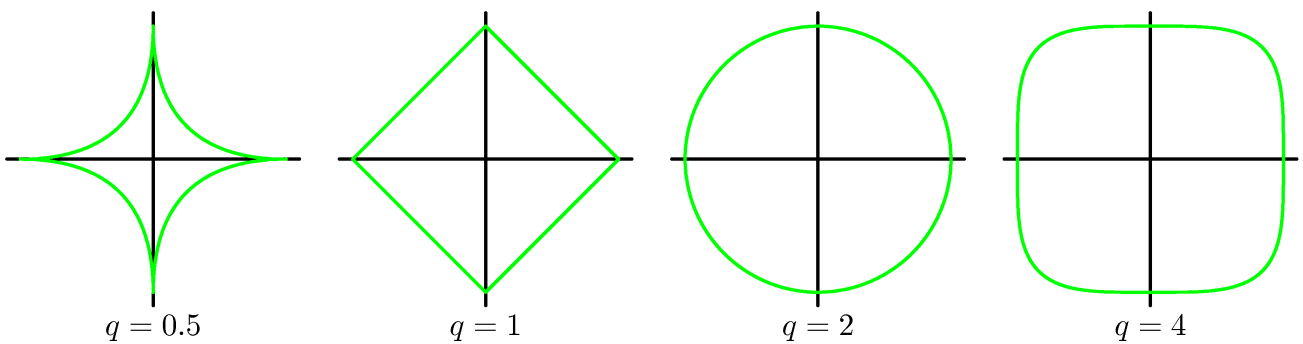
$$\mathbf{w} = \left(\lambda \mathbf{I} + \mathbf{\Phi}^\top \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^\top \mathbf{t}$$

Regularized Least Squares

With a more general regularizer, we have

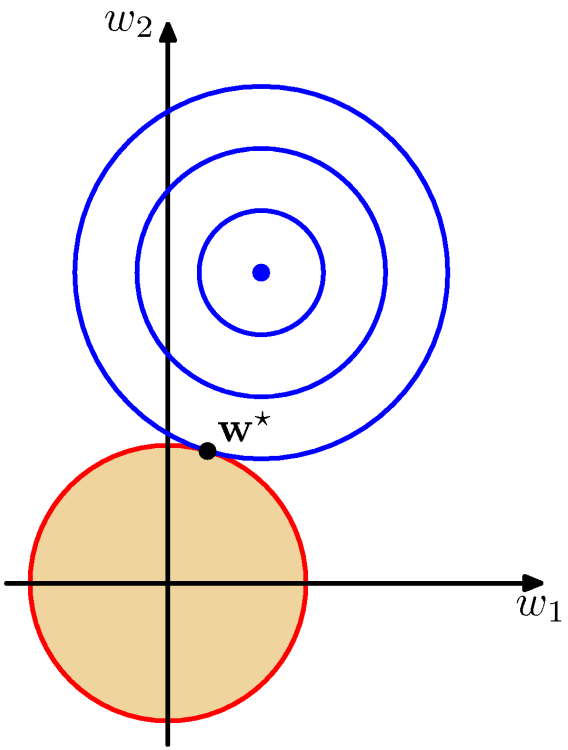
$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^\top \phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$

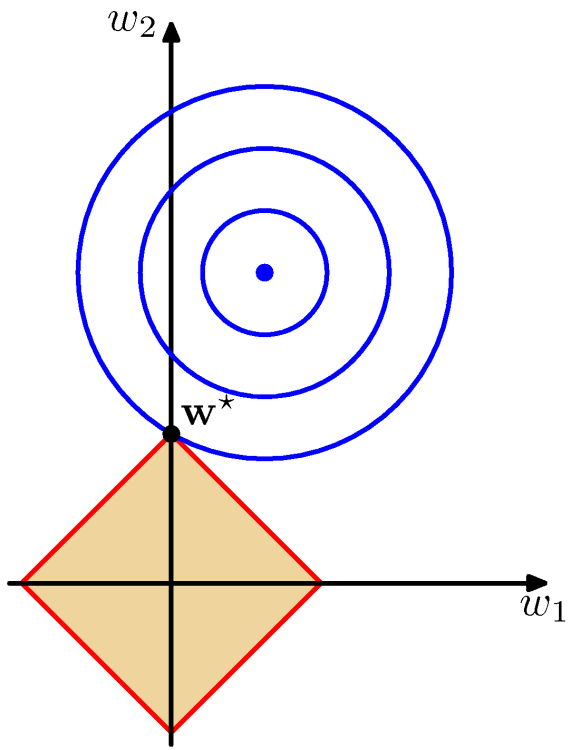
where $q = 1$ and $q = 2$ correspond to the *lasso* and quadratic regularizers, respectively.



Regularized Least Squares

Lasso tends to generate sparse solutions compared to a quadratic regularizer





Multiple Outputs

- If we want to predict $K > 1$ target variables, we can define a different set of basis functions for each target variable, leading to multiple independent regression problems.
- A more common approach is to use the same set of basis functions to model all of the components of the target vector so that $\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{W}^\top \phi(\mathbf{x})$, where $\mathbf{y}$ is a K-dimensional column vector, $\mathbf{W}$ is an $M \times K$ matrix of parameters, and $\phi(\mathbf{x})$ is an M-dimensional column vector with elements $\phi_j(\mathbf{x})$, with $\phi_0(\mathbf{x}) = 1$.

Multiple Outputs

Analogously to the single output case, we have

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

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

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

Multiple Outputs

Maximizing w.r.t $\mathbf{W}$, we obtain

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

if we consider a single target variable $\mathbf{t}_k$, we see that

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

where $\mathbf{t}_k = [t_{1k}, \ldots, t_{Nk}]^\top$. This is identical to the single output case.

Bayesian Linear Regression

Main issues of Maximum Likelihood:

- **Model complexity:** model complexity, governed by the number of basis functions, needs to be controlled according to the size of data set. This can be dealt with by using a regularization term.
- **Over-fitting:** Although the choice of the number and form of the basis functions is still important in determining the overall behavior of the model

Can we do better? Yes! Both problems can be avoided using *Bayesian Linear Regression*.

Bayesian Linear Regression

Define a conjugate prior over the model parameters $\mathbf{w}$ as

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

having mean $\mathbf{m}_0$ and covariance $\mathbf{S}_0$.
Combining this with the likelihood function and using results of marginal and conditional Gaussian distributions, gives the posterior

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

where

$$\mathbf{m}_N = \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\mathbf{\Phi}^\top\mathbf{t})$$
$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\mathbf{\Phi}^\top\mathbf{\Phi}.$$

Bayesian Linear Regression

Notes:

- because the posterior distribution is Gaussian, its mode coincides with its mean. Thus the maximum posterior weight vector is simply given by $\mathbf{w}_{MAP} = \mathbf{m}_N$.
- If we consider an infinitely broad prior $\mathbf{S}_0 = \alpha^{-1}\mathbf{I}$ with $\alpha \to 0$, the mean $\mathbf{m}_N$ of the posterior distribution reduces to the maximum likelihood value $\mathbf{w}_{ML}$.
- if $N = 0$, then the posterior distribution reverts to the prior.
- if data points arrive sequentially, then the posterior distribution at any stage acts as the prior distribution for the subsequent data point.

Bayesian Linear Regression

A common choice for the prior is

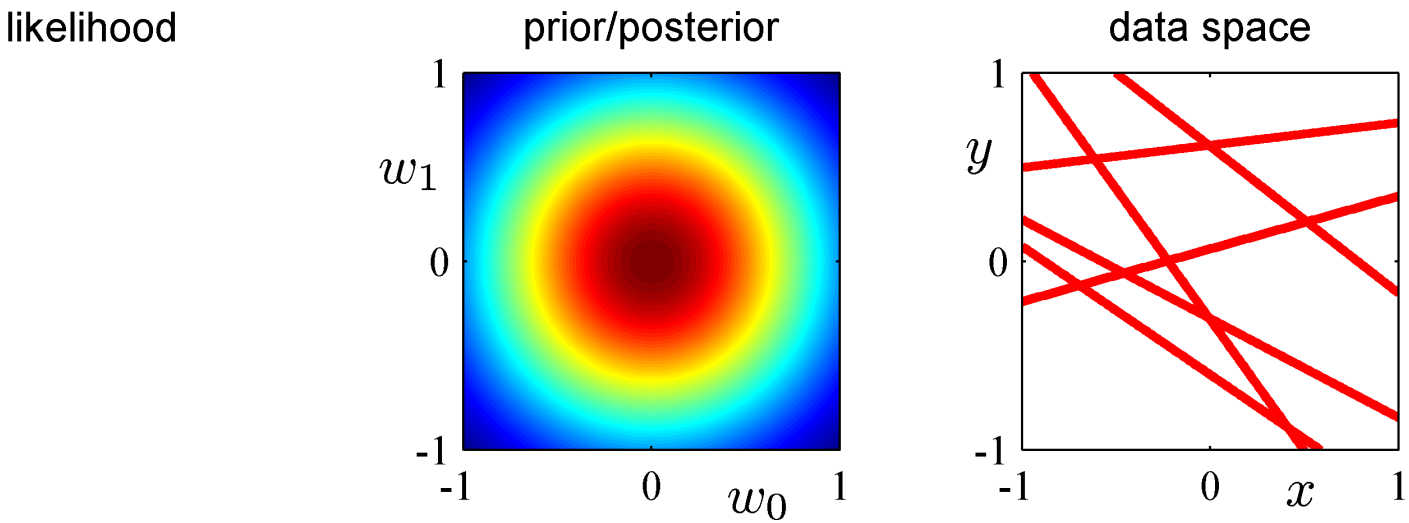
$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I})$$

for which

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

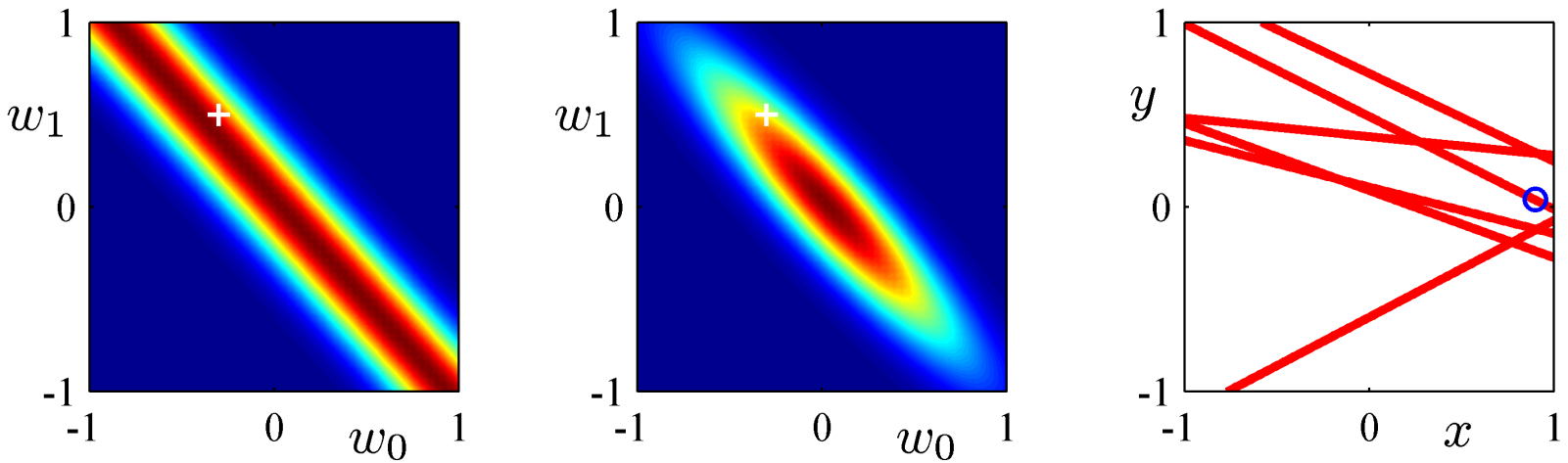
Bayesian Linear Regression - Example

0 data points observed



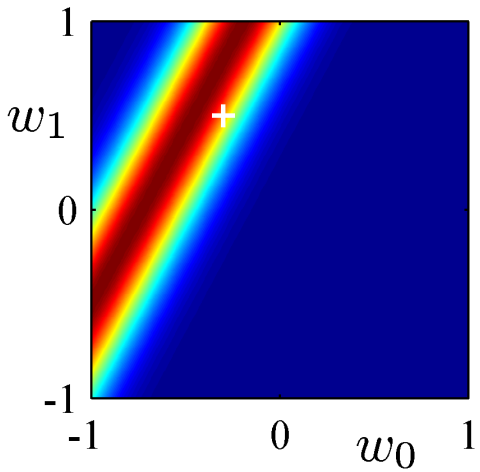
Bayesian Linear Regression - Example

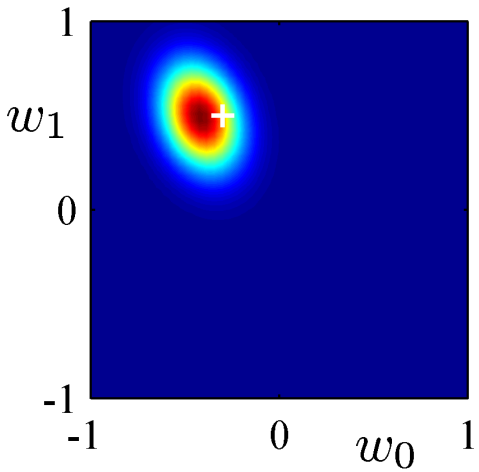
1 data points observed

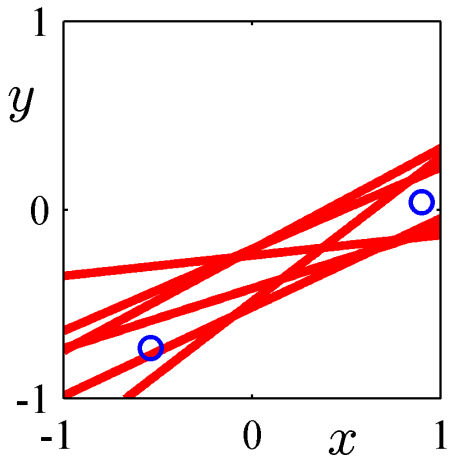


Bayesian Linear Regression - Example

2 data points observed

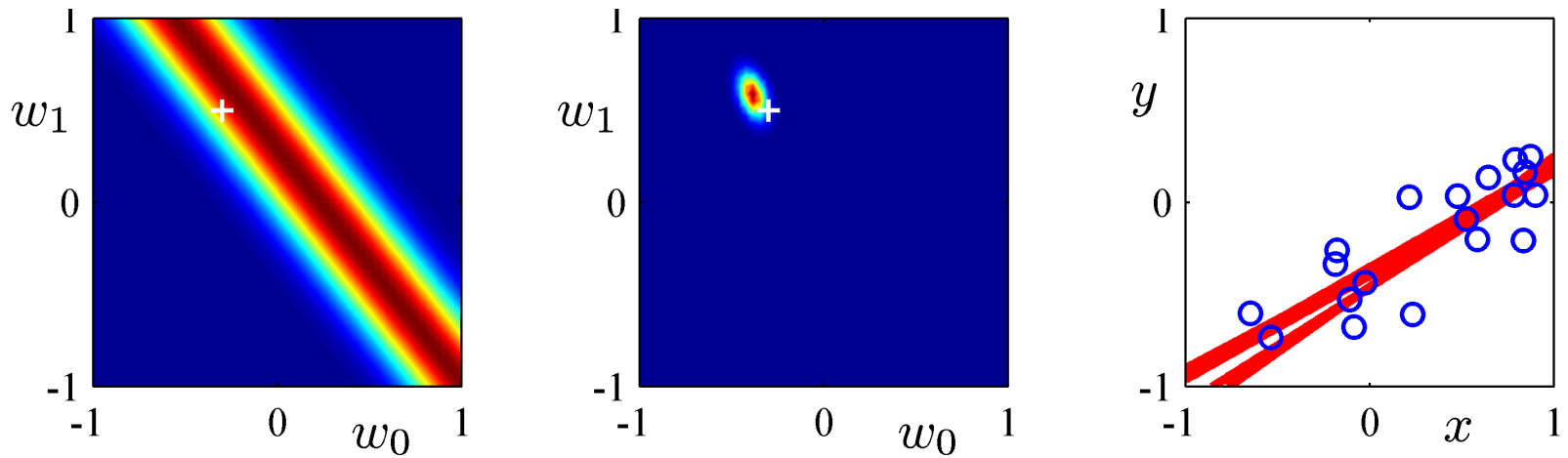






Bayesian Linear Regression - Example

20 data points observed



Predictive Distribution (1)

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

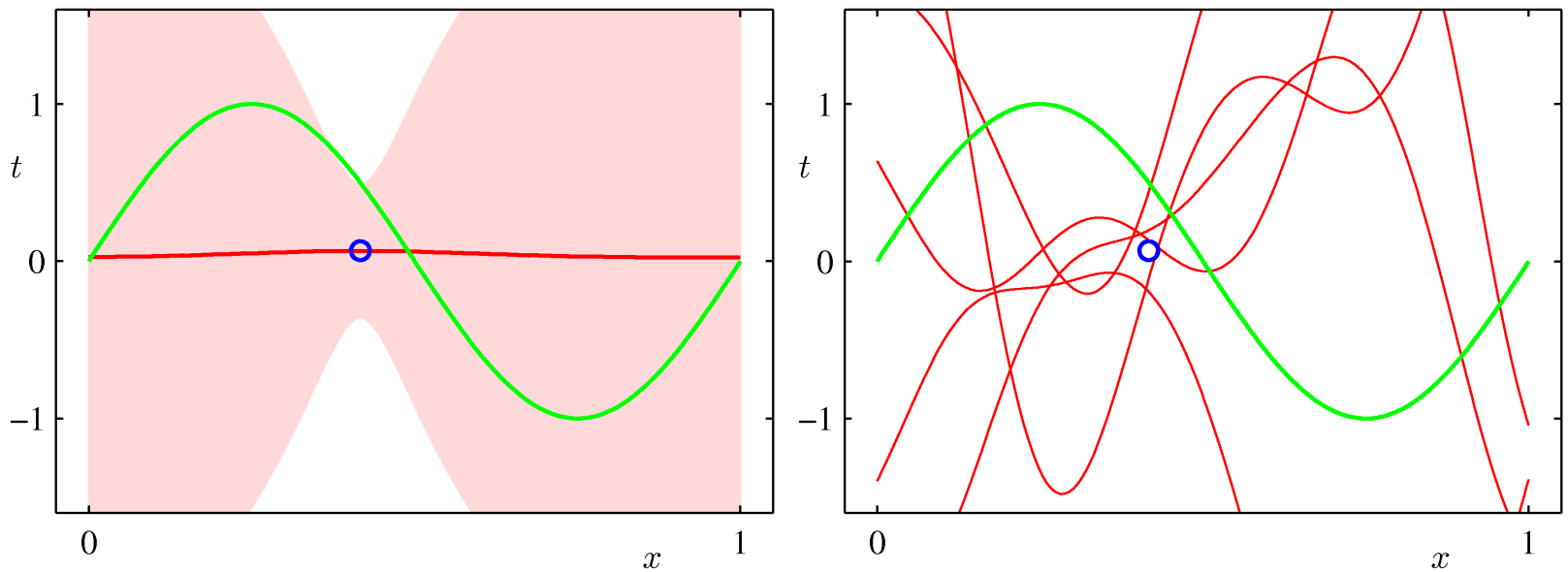
$$\begin{aligned} p(t|\mathbf{t}, \alpha, \beta) &= \int p(t|\mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \alpha, \beta) d\mathbf{w} \\ &= \mathcal{N}(t|\mathbf{m}_N^\top \boldsymbol{\phi}(\mathbf{x}), \sigma_N^2(\mathbf{x})) \end{aligned}$$

where

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

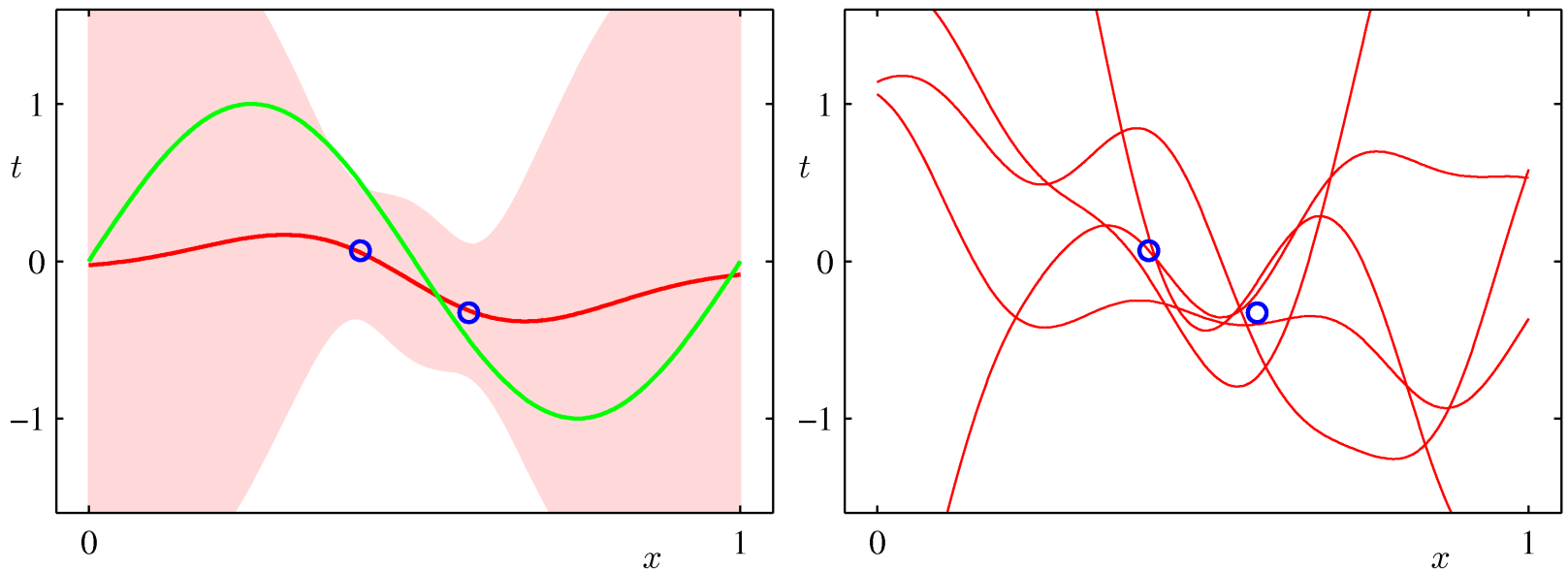
Predictive Distribution (2)

Example: sinusoidal data, 9 Gaussian basis functions, 1 data points



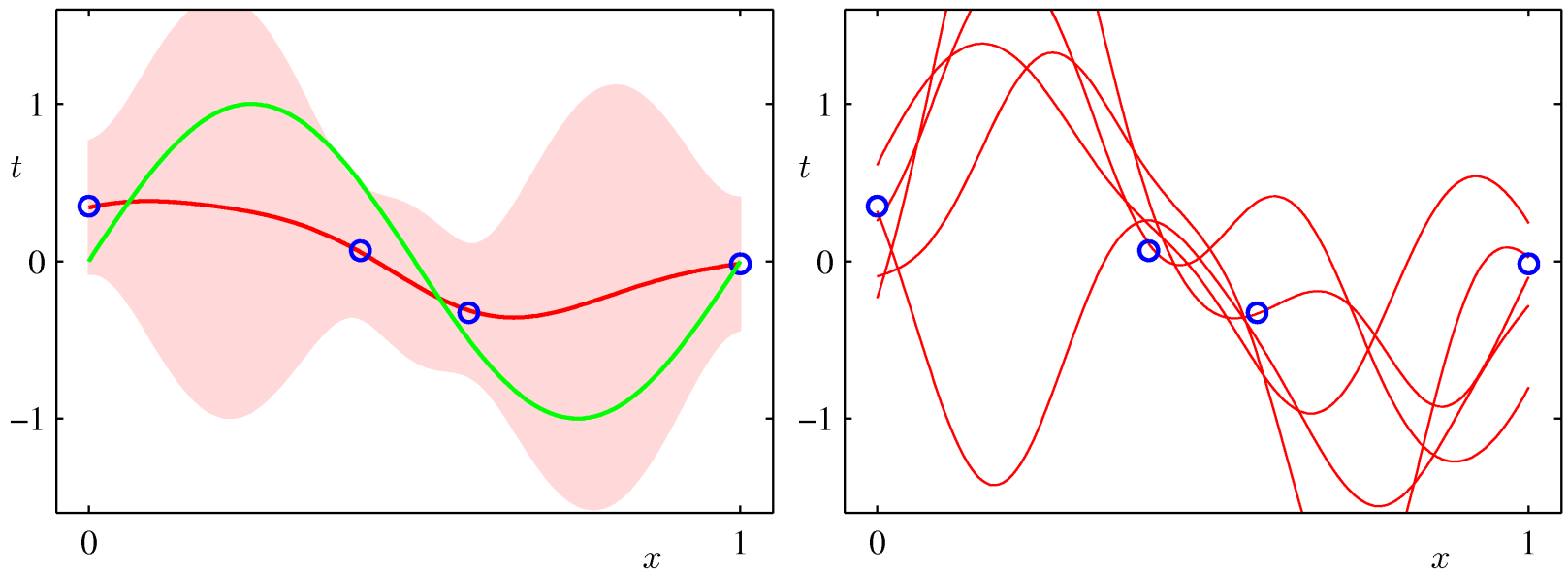
Predictive Distribution (3)

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



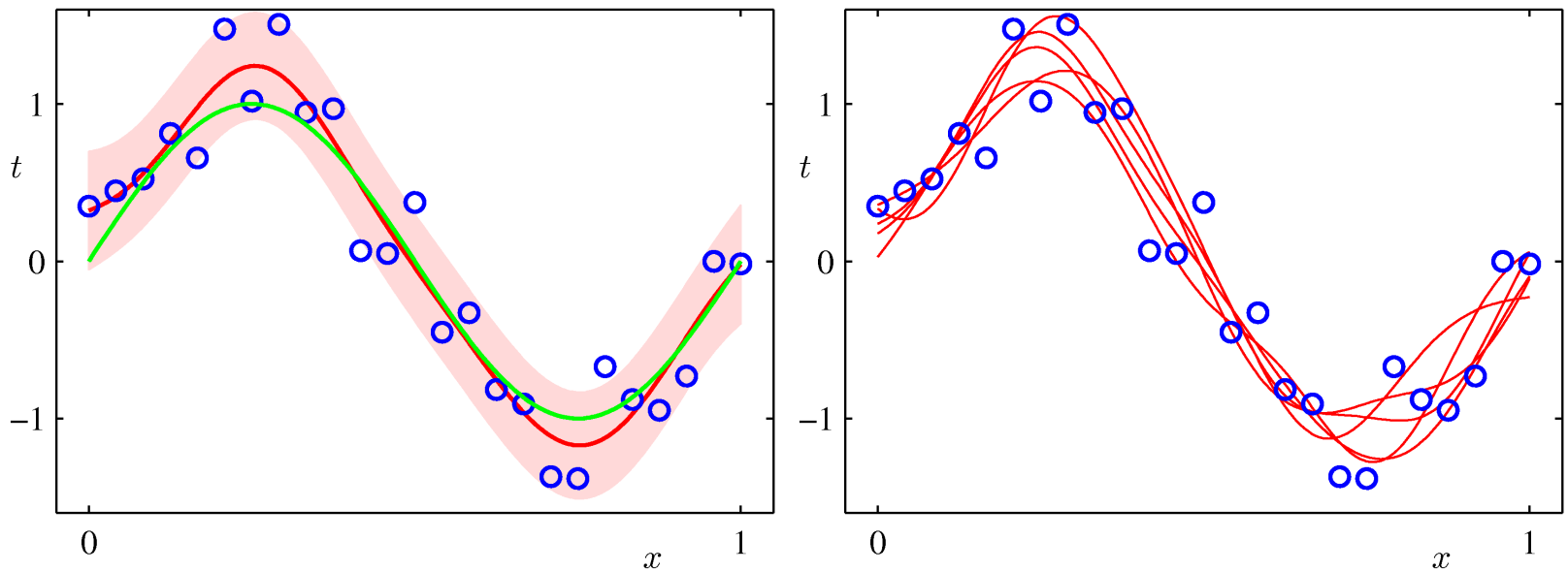
Predictive Distribution (4)

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



Predictive Distribution (5)

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



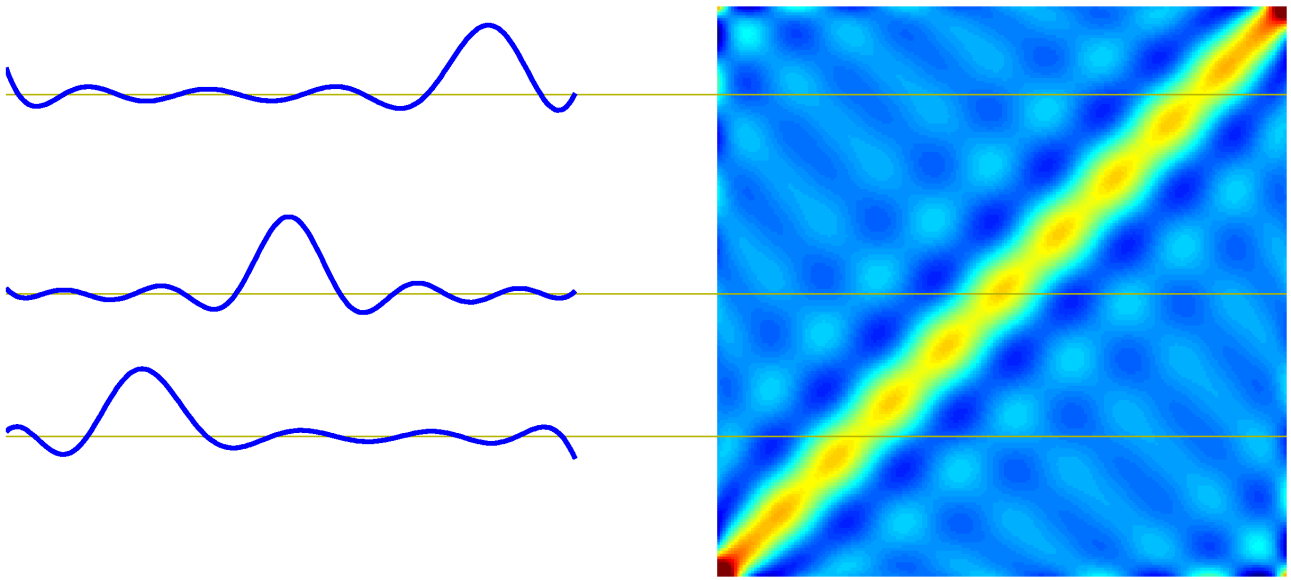
Equivalent Kernel (1)

The predictive mean can be written

$$\begin{aligned} y(\mathbf{x}, \mathbf{m}_N) &= \mathbf{m}_N \boldsymbol{\phi}(\mathbf{x}) = \beta \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{S}_N \boldsymbol{\Phi}^\top \mathbf{t} \\ &= \sum_{n=1}^{N} \beta \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}_n)^\top t_n \\ &= \sum_{n=1}^{N} \kappa(\mathbf{x}, \mathbf{x}_n) t_n \end{aligned}$$

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

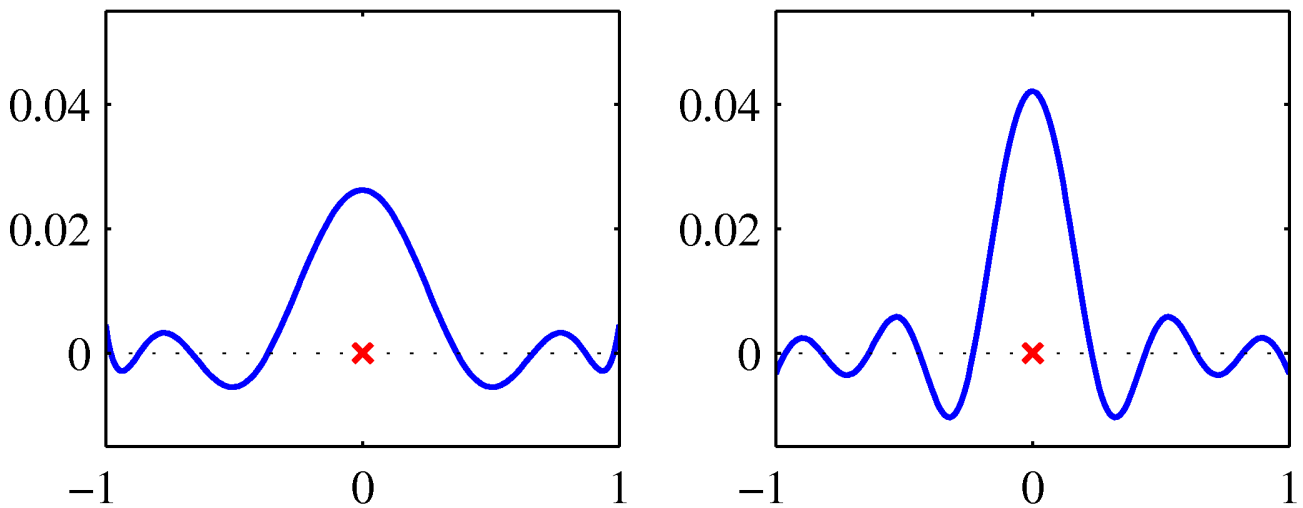
Equivalent Kernel (2)



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:



(left) Polynomial, (right) Sigmoid

Equivalent Kernel (4)

The kernel as a covariance function: Consider

$$\begin{aligned} cov[y(\mathbf{x}), y(\mathbf{x}')] &= cov[\boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w}, \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}')] \\ &= \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}') = \beta^{-1} \kappa(\mathbf{x}, \mathbf{x}') \end{aligned}$$

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

Equivalent Kernel (5)

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

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

$$\kappa(\mathbf{x}, \mathbf{z}) = \boldsymbol{\psi}(\mathbf{x})^{\top} \boldsymbol{\psi}(\mathbf{z})$$

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

Bayesian Model Comparison (1)

How do we choose the *right* model? Assume we want to compare models $\mathcal{M}_i$, $i = 1, \ldots, L$, 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 $\propto$ prior $\times$ 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

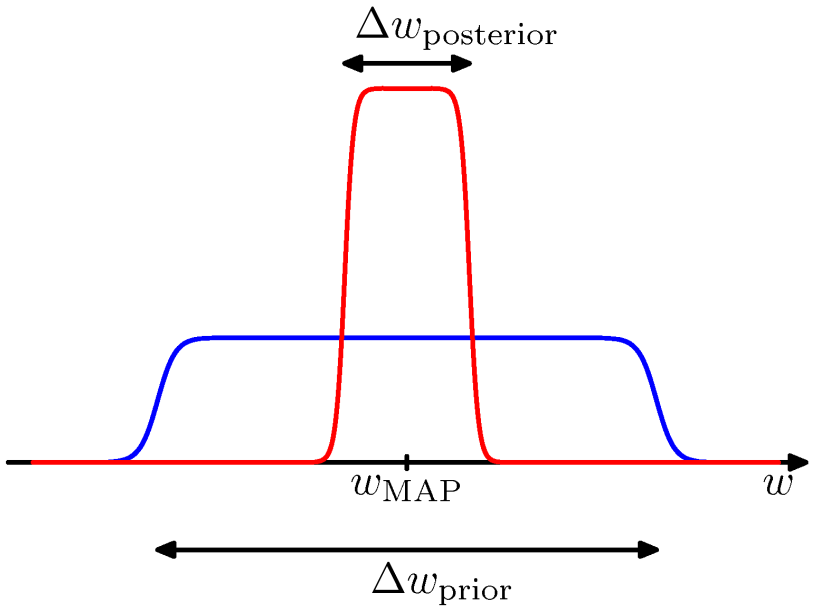
$$p(\mathbf{w}|\mathcal{D}, \mathcal{M}_i) = \frac{p(\mathcal{D}|\mathbf{W}, \mathcal{M}_i) p(\mathbf{w}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_i)}$$

Bayesian Model Comparison (4)

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

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

where the posterior is assumed to be sharply peaked.



Bayesian Model Comparison (5)

Taking logarithms, we get

$$\ln p(\mathcal{D}) \approx \ln p(\mathcal{D}|w_{\text{MAP}}) + \ln(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}})$$

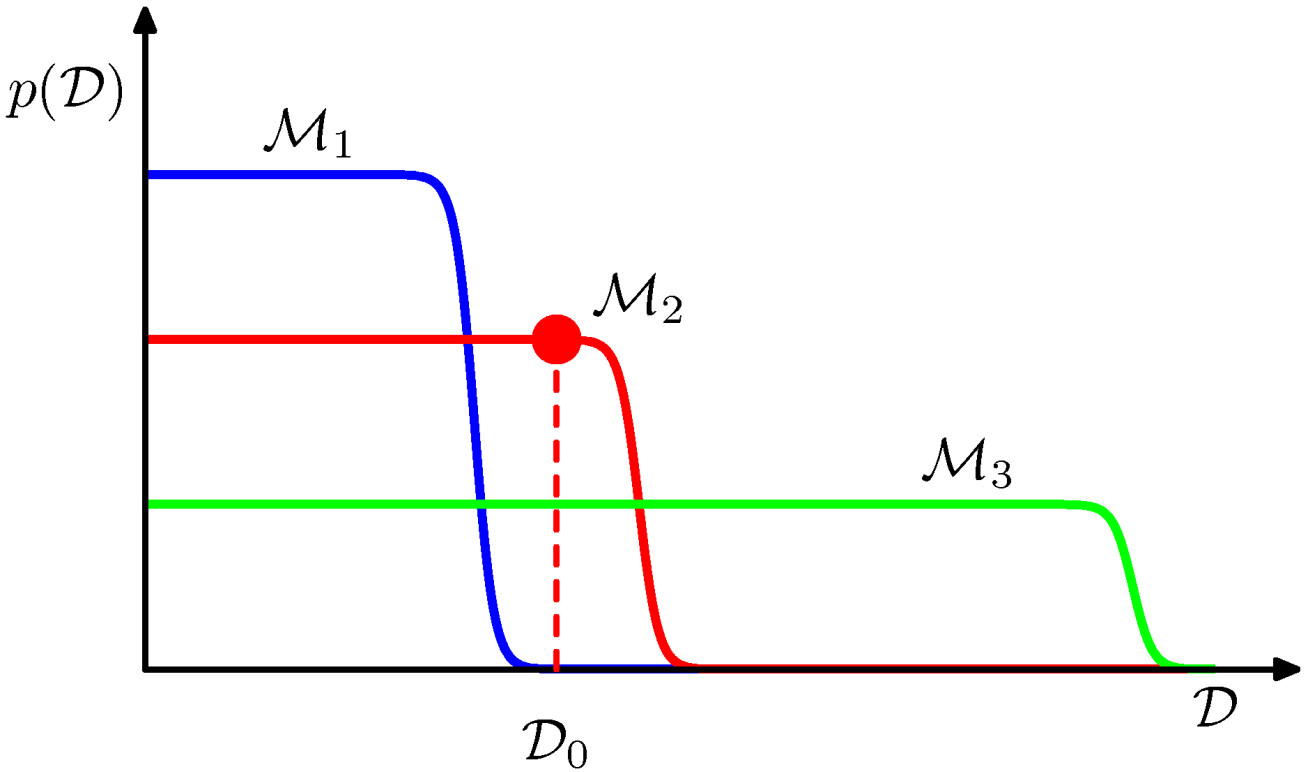
where the last term is negative. With M parameters, all assumed to have the same ratio $\Delta w_{\text{posterior}}/\Delta w_{\text{prior}}$, we get

$$\ln p(\mathcal{D}) \approx \ln p(\mathcal{D}|\mathbf{w}_{\text{MAP}}) + \ln(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}})$$

where the last term is 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

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

But this integral is intractable. Approximate with:

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

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

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

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

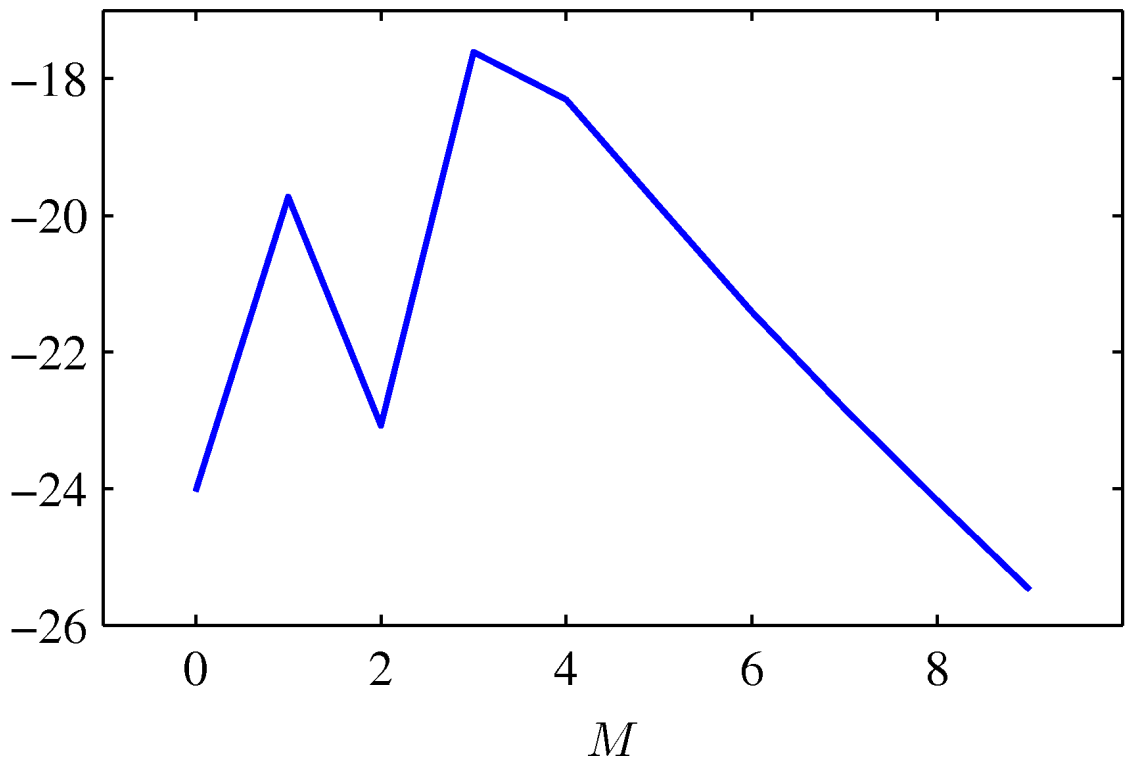
$$p(\alpha, \beta|\mathbf{t}) \propto p(\mathbf{t}|\alpha, \beta) = \int p(\mathbf{t}|\mathbf{w}, \beta)p(\mathbf{w}|\alpha)d\mathbf{w}$$

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} order polynomial, $\alpha = 5 \times 10^{-3}$



y-axis shows $\ln p(\mathbf{t}|\alpha, \beta)$.

Maximizing the Evidence Function (1)

To maximize $\ln p(\mathbf{t}|\alpha, \beta)$ with respect to α and β we define the eigenvector equation

$$(\beta \mathbf{\Phi}^\top \mathbf{\Phi})\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

Thus

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

has eigenvalues $\lambda_i + \alpha$.

Maximizing the Evidence Function (2)

We can now differentiate $\ln p(\mathbf{t}|\alpha, \beta)$ with respect to α and β, and set to results to zero, we get

$$\alpha = \frac{\gamma}{\mathbf{m}_N^\top \mathbf{m}_N}$$

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

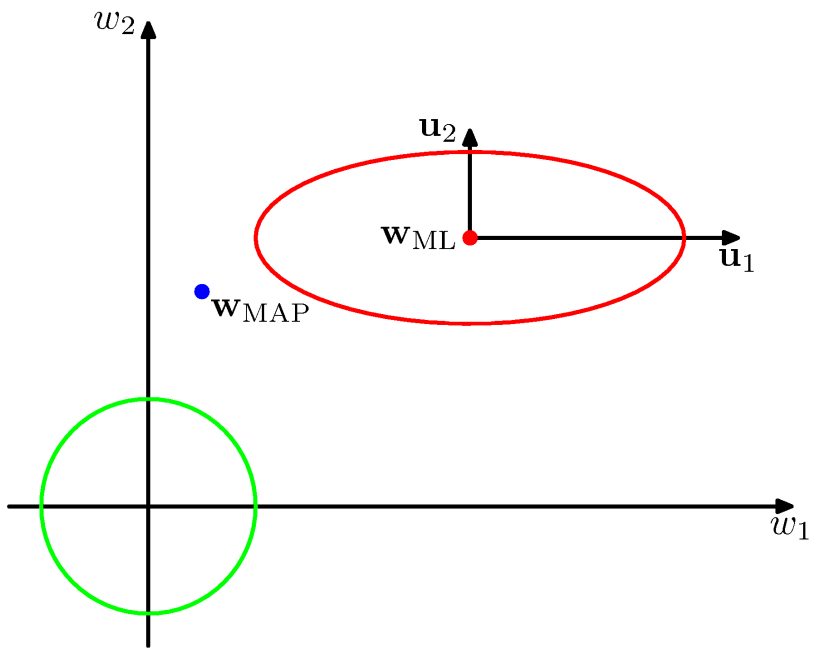
where

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

Note that γ depends on both α and β.

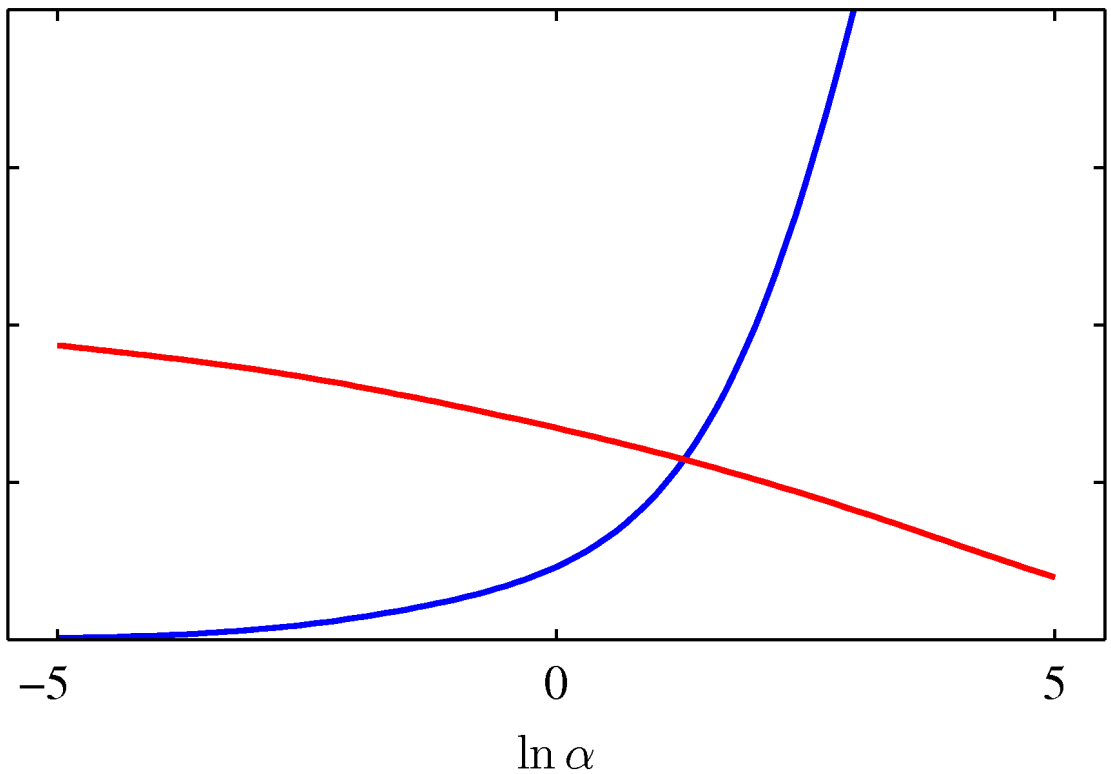
Effective Number of Parameters (1)

$\lambda_1 \ll \alpha$: 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.



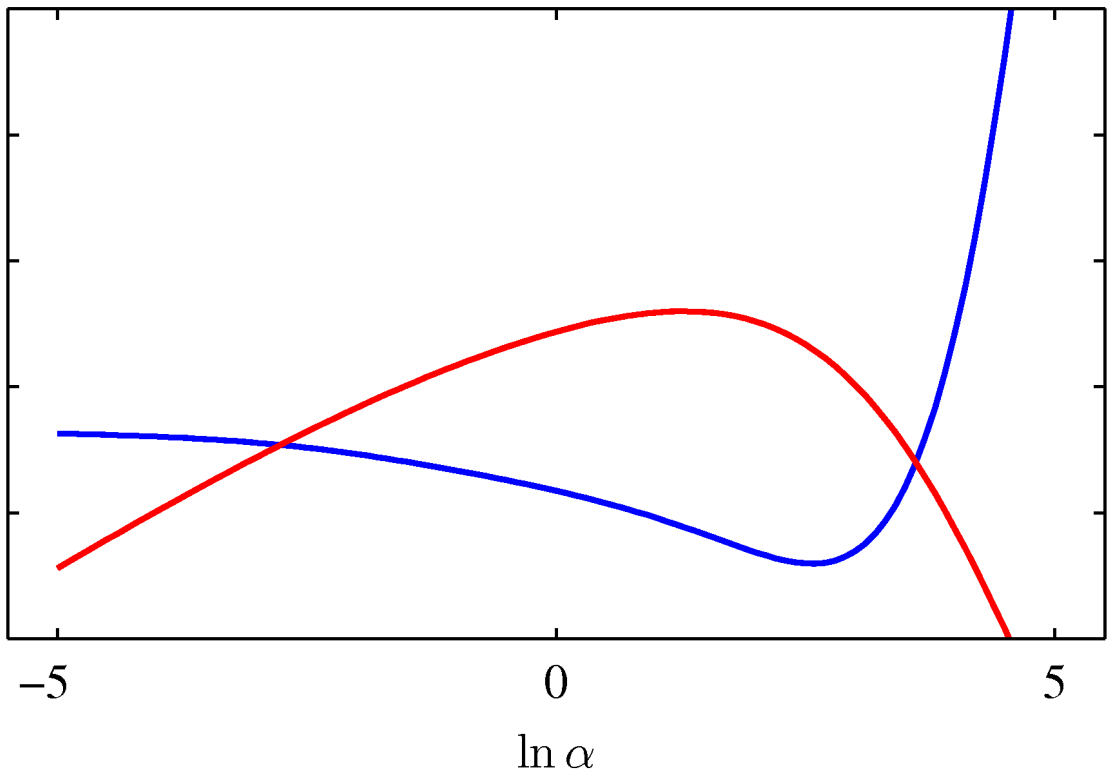
Effective Number of Parameters (2)

Example: sinusoidal data, 9 Gaussian basis functions, $\beta = 11.1$. γ (red) and $\alpha \mathbf{m}_N^\top \mathbf{m}_N$ (blue).



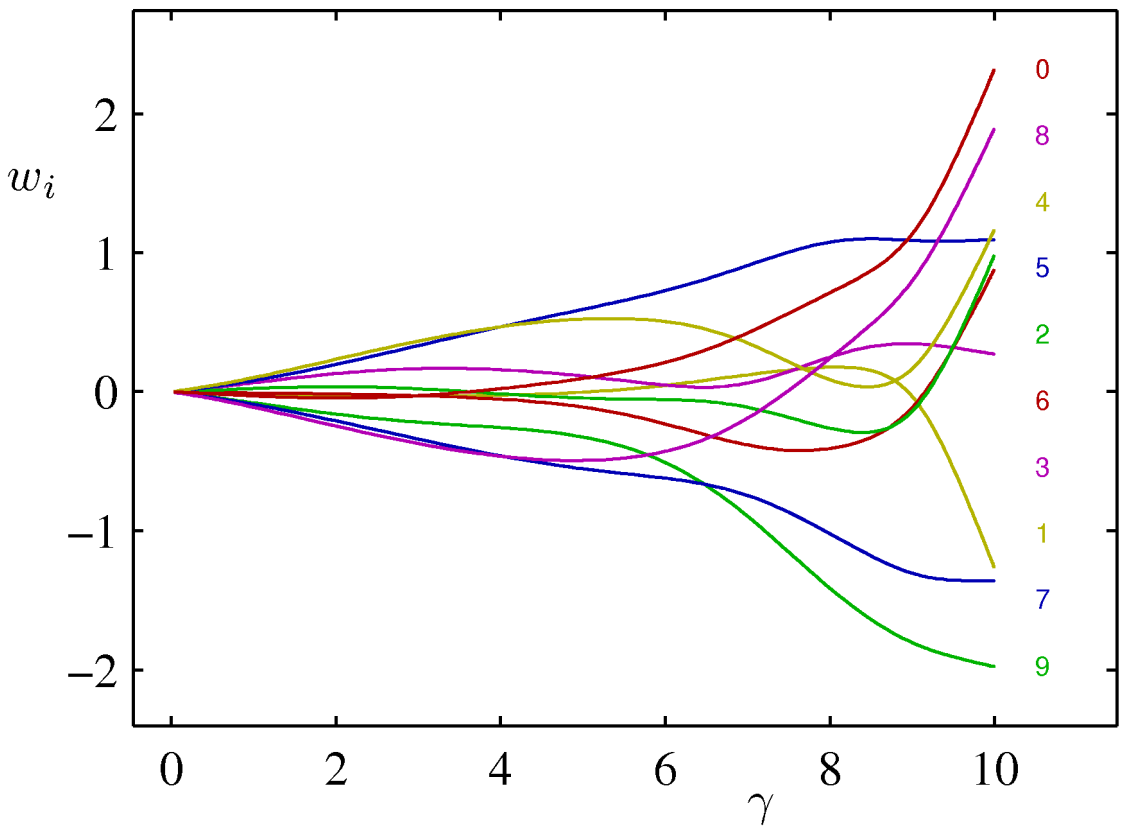
Effective Number of Parameters (3)

Example: sinusoidal data, 9 Gaussian basis functions, $\beta = 11.1$. Test set error (blue) and $\ln p(\mathbf{t}|\alpha, \beta)$ (red).



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

$$\alpha = \frac{M}{\mathbf{m}_N^\top \mathbf{m}_N}$$

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

Limitations of Fixed Basis Functions

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