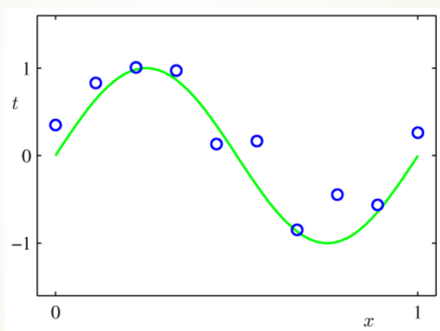
Machine Learning

Chapter 3: Linear Models for Regression

林嘉文 (Chia-Wen Lin)
清華大學電機系
cwlin@ee.nthu.edu.tw

Linear Basis Function Models (1)

Example: Polynomial Curve Fitting



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

3

Linear Basis Function Models (2)

Generally

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

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 : $\phi_d(\mathbf{x}) = x_d$.

3/20/2021

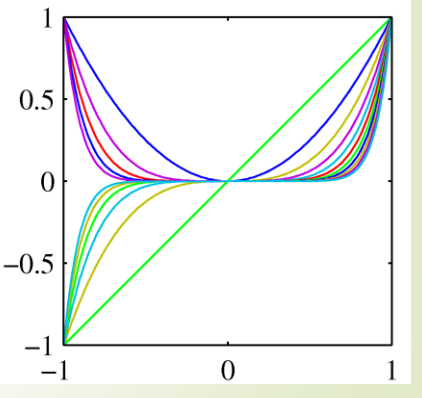
4

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.



3/20/2021

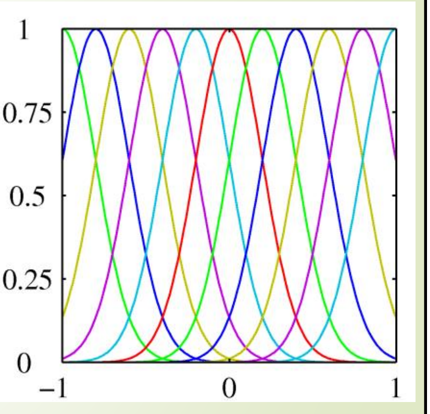
5

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).



6

Linear Basis Function Models (5)

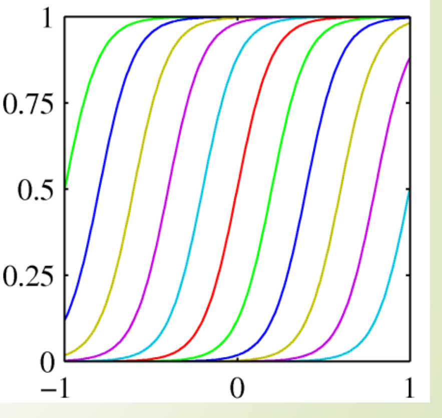
Sigmoidal basis functions:

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

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).



7

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

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

3/20/2021

8

Maximum Likelihood and Least Squares (2)

Taking the logarithm, we get

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

where

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

is the sum-of-squares error.

3/20/2021

9

Maximum Likelihood and Least Squares (3)

Computing the gradient and setting it to zero yields

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

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

$$\mathbf{w}_{\mathrm{ML}} = \underbrace{\left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}}_{\text{The Moore-Penrose pseudo-inverse, } \boldsymbol{\Phi}^{\dagger}.}\mathbf{t}$$

where

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

3/20/2021

11

Geometry of Least Squares

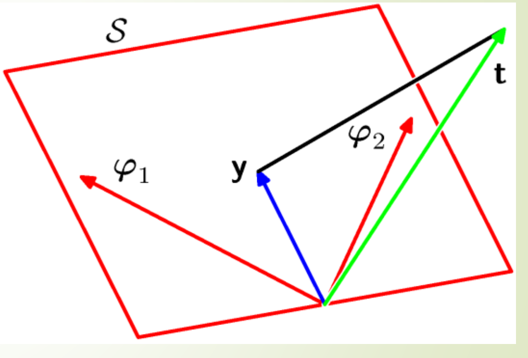
Consider

$$\mathbf{y} = \boldsymbol{\Phi}\mathbf{w}_{\mathrm{ML}} = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M]\,\mathbf{w}_{\mathrm{ML}}.$$

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

$\mathcal{T}$: N-dimensional

$\mathcal{S}$: M-dimensional



$\mathcal{S}$ is spanned by $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M$.

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

3/20/2021

12

Sequential Learning

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

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

This is known as the **least-mean-squares (LMS)** algorithm.

Issue: how to choose η?

3/20/2021

13

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

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

λ is called the regularization coefficient.

which is minimized by

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

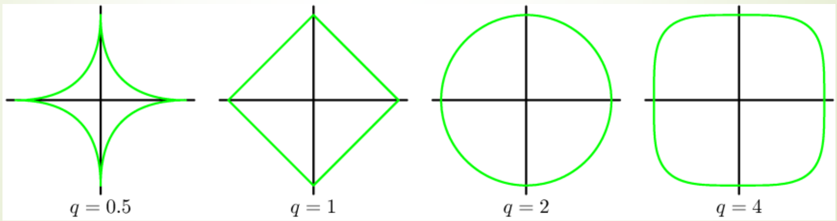
3/20/2021

14

Regularized Least Squares (2)

With a more general regularizer, we have

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



Lasso

Quadratic

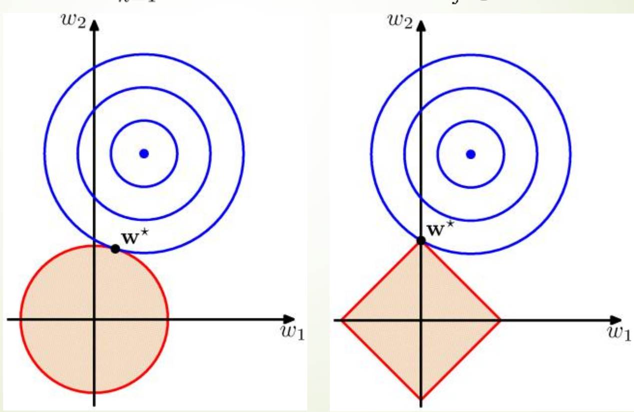
3/20/2021

15

Regularized Least Squares (3)

Lasso tends to generate sparser solutions than a quadratic regularizer.

Minimize $\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$ s.t. $\sum_{j=1}^{M}|w_j|^q \le \eta$



3/20/2021

16

Multiple Outputs (1)

Analogously to the single output case we have:

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

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

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

3/20/2021

17

Multiple Outputs (2)

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

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

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

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

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

3/20/2021

18

The Bias-Variance Decomposition (1)

Recall the **expected squared loss**,

$$\mathbb{E}[L] = \iint \{y(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\, \mathrm{d}\mathbf{x}\, \mathrm{d}t$$

$$\begin{aligned} \{y(\mathbf{x}) - t\}^2 &= \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}] + \mathbb{E}[t|\mathbf{x}] - t\}^2 \\ &= \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}^2 + 2\{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}\{\mathbb{E}[t|\mathbf{x}] - t\} + \{\mathbb{E}[t|\mathbf{x}] - t\}^2 \end{aligned}$$

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

where $h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x})\, \mathrm{d}t.$

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

What about the first term?

19

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

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

20

The Bias-Variance Decomposition (3)

Taking the expectation over $\mathcal{D}$ yields

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

3/20/2021

21

The Bias-Variance Decomposition (4)

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

Thus we can write

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

where

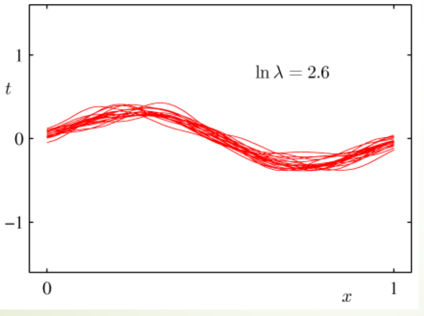
$$\begin{aligned}(\text{bias})^2&=\int\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]-h(\mathbf{x})\}^2p(\mathbf{x})\,\mathrm{d}\mathbf{x}\\\text{variance}&=\int\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D})-\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right]p(\mathbf{x})\,\mathrm{d}\mathbf{x}\\\text{noise}&=\iint\{h(\mathbf{x})-t\}^2p(\mathbf{x},t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\end{aligned}$$

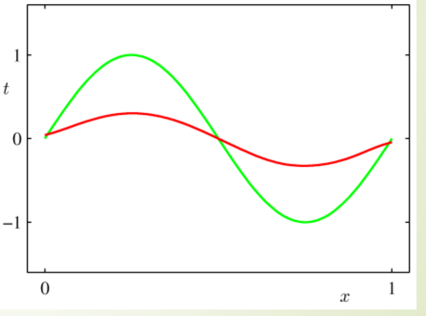
3/20/2021

22

The Bias-Variance Decomposition (5)

Example: 100 data sets, each with 25 samples, from the sinusoidal, varying the degree of regularization, λ.



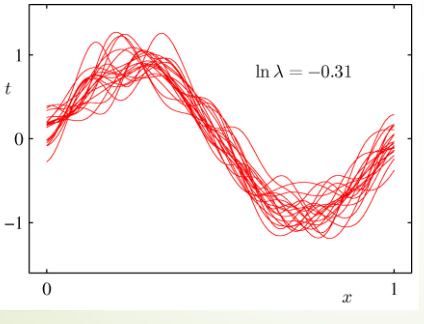


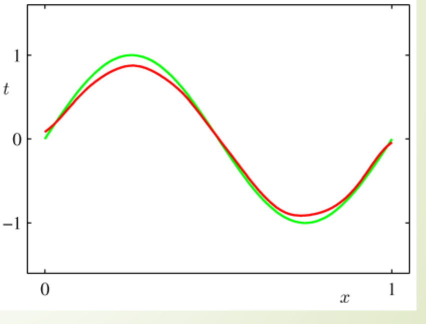
3/20/2021

23

The Bias-Variance Decomposition (6)

Example: 100 data sets, each with 25 samples, from the sinusoidal, varying the degree of regularization, λ.



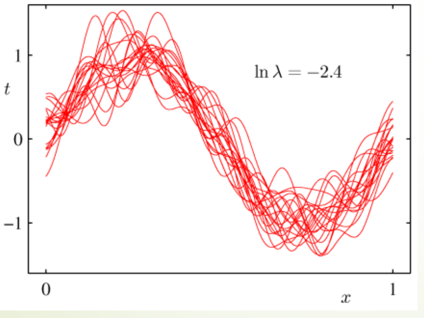


3/20/2021

24

The Bias-Variance Decomposition (7)

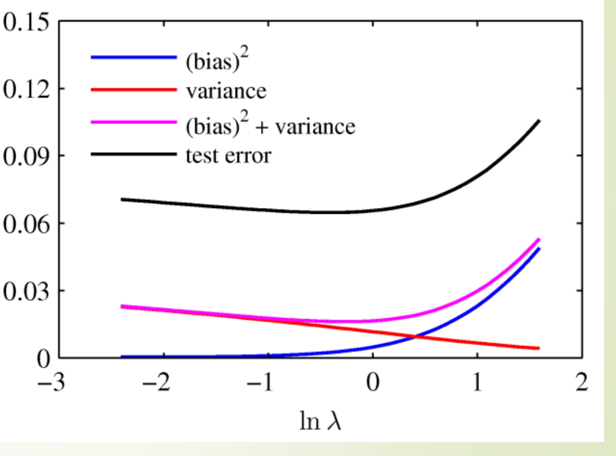
Example: 100 data sets, each with 25 samples, from the sinusoidal, varying the degree of regularization, λ.



25

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.



26

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

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

3/20/2021

27

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

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

Next we consider an example …

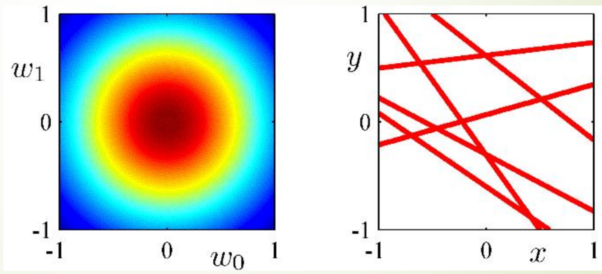
3/20/2021

28

Bayesian Linear Regression (3)

0 data points observed

Prior Data Space



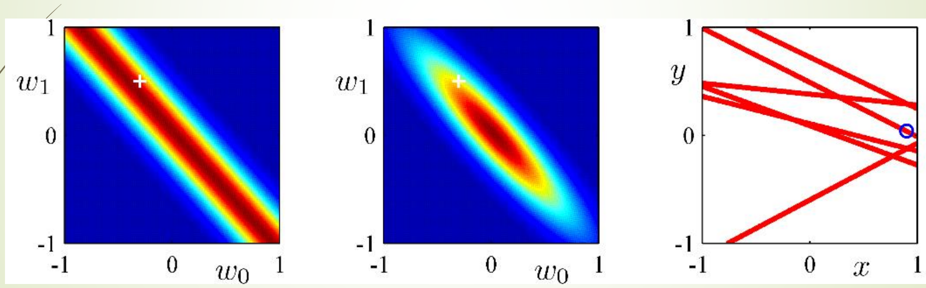
3/20/2021

29

Bayesian Linear Regression (4)

1 data point observed

Likelihood Posterior Data Space



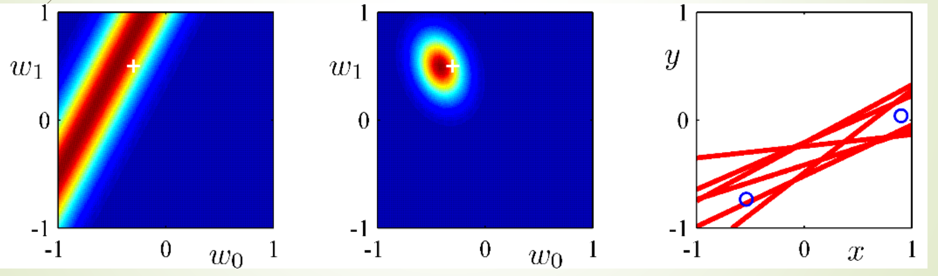
3/20/2021

30

Bayesian Linear Regression (5)

2 data point observed

Likelihood Posterior Data Space

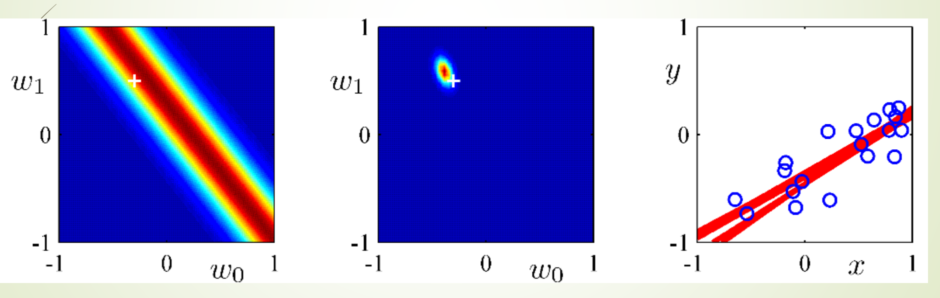


31

Bayesian Linear Regression (6)

20 data point observed

Likelihood Posterior Data Space



32

Predictive Distribution (1)

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

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

where

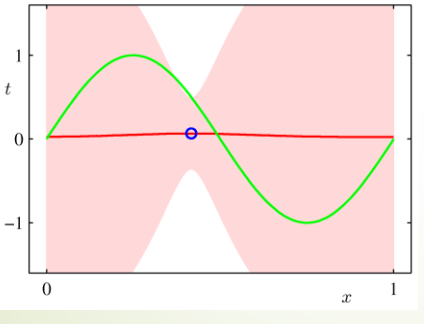
$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}).$$

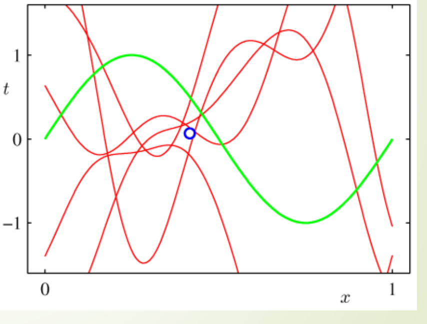
3/20/2021

33

Predictive Distribution (2)

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



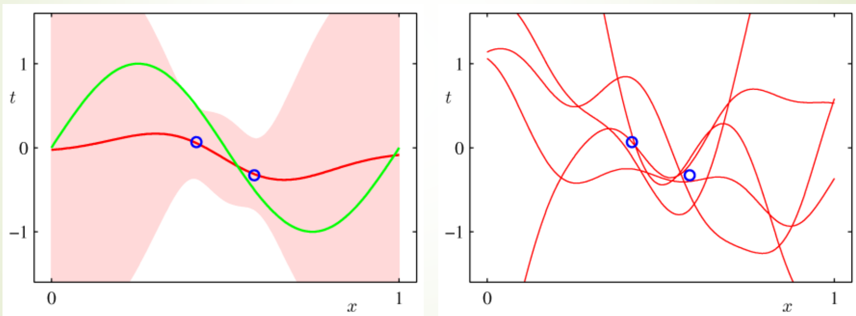


3/20/2021

34

Predictive Distribution (3)

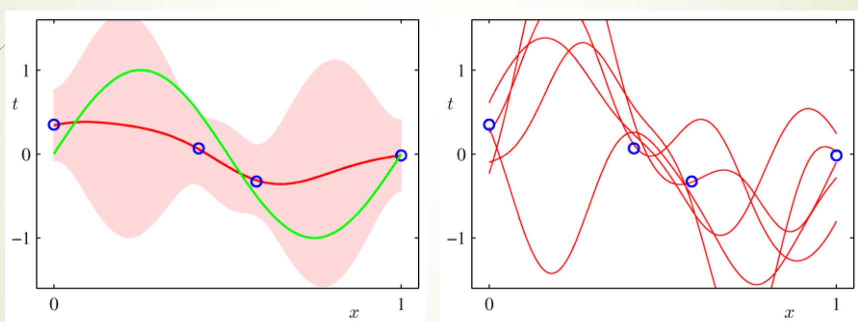
Example: Sinusoidal data, 9 Gaussian basis functions, 2 data points



35

Predictive Distribution (4)

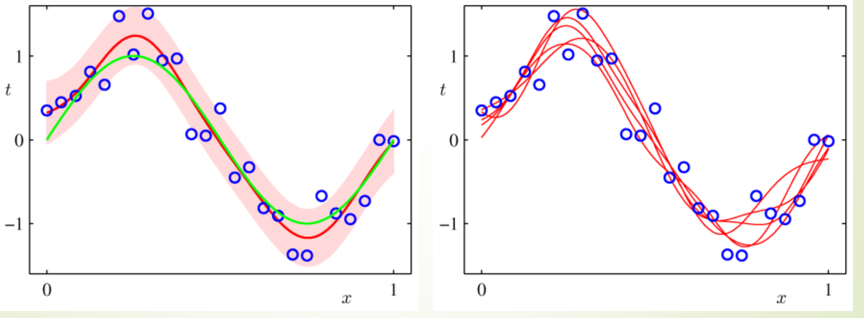
Example: Sinusoidal data, 9 Gaussian basis functions, 4 data points



36

Predictive Distribution (5)

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



37

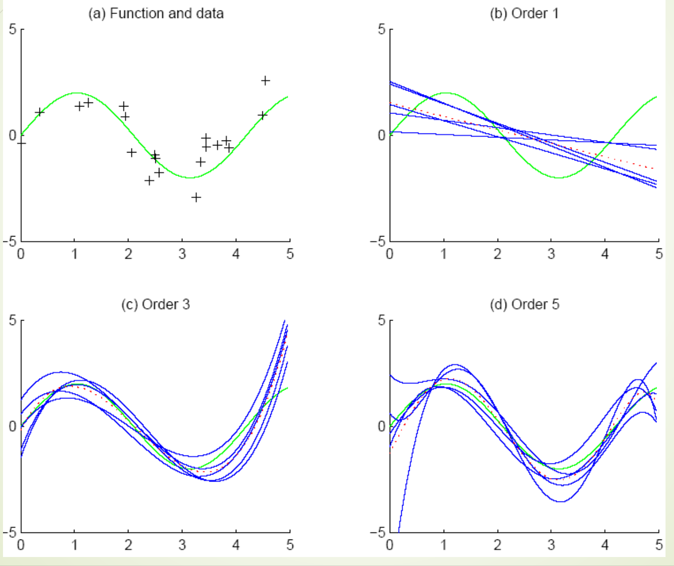
Bias/Variance Dilemma (1)

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

- Bias: the difference between the expected value and the true value
- Variance: variations of estimated values
- Given training set D, as we increase complexity,
  - bias decreases (a better fit to data) and variance increases (fit varies more with data)
- High bias means usually low variance, and vice versa
- Bias/Variance dilemma (Geman et al., 1992)

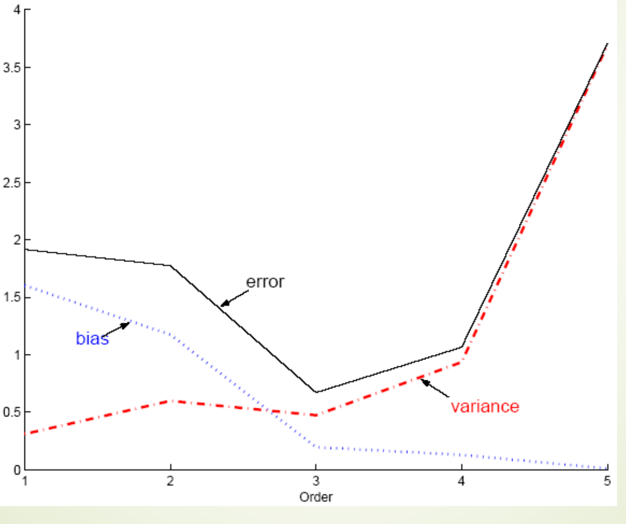
38

Bias/Variance Dilemma (2)



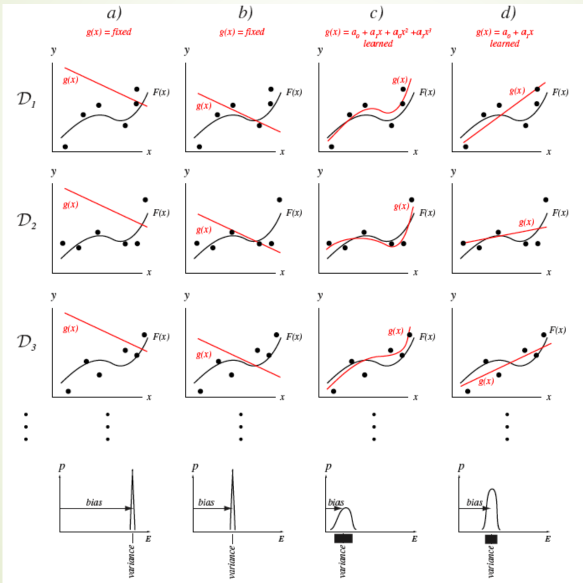
39

Bias/Variance Dilemma (2)



40

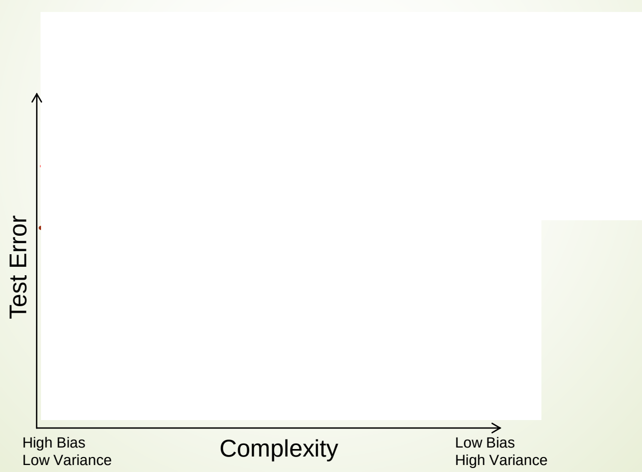
Bias/Variance Dilemma (4)



41

Bias/Variance Dilemma (5)

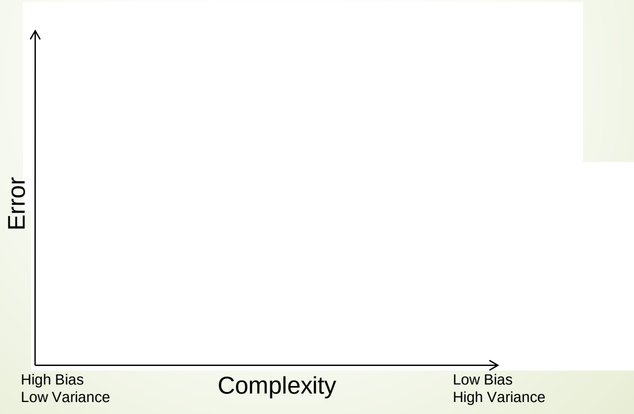
Error = noise^2 + bias^2 + variance



42

Bias/Variance Dilemma (6)

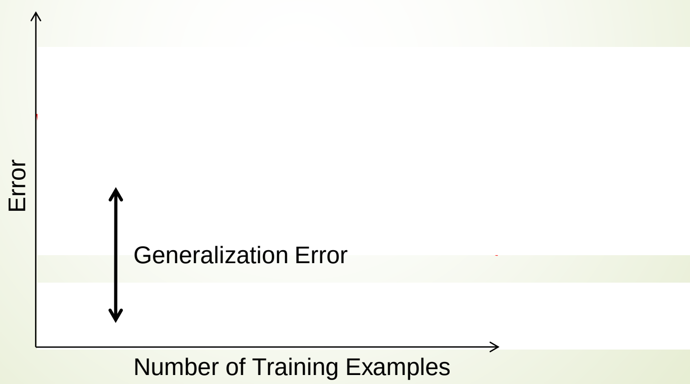
- Need validation set
- Validation set is separate from the test set



43

Bias/Variance Dilemma (7)

Fixed classifier



44

Equivalent Kernel (1)

The predictive mean can be written

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

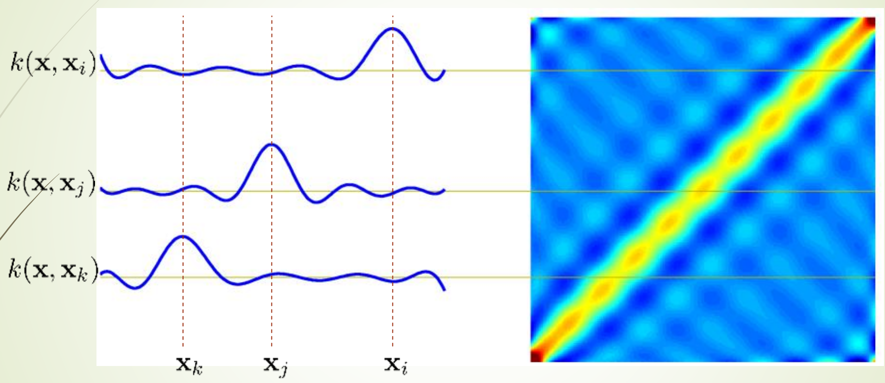
Equivalent kernel or smoother matrix.

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

3/20/2021

45

Equivalent Kernel (2)



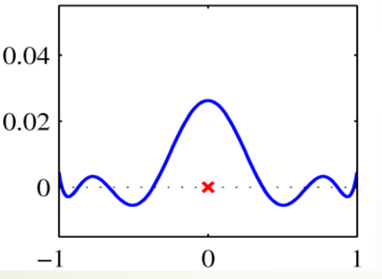
Weight of t_n depends on distance between x and x_n; nearby x_n carry more weight.

3/20/2021

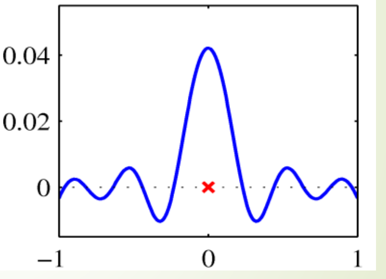
46

Equivalent Kernel (3)

Non-local basis functions have local equivalent kernels:



Polynomial



Sigmoidal

47

Equivalent Kernel (4)

The kernel as a covariance function: consider

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

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

48

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:

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

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

49

Bayesian Model Comparison (1)

How do we choose the 'right' model?

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

$$\underbrace{p(\mathcal{M}_i|\mathcal{D})}_{\text{Posterior}} \propto \underbrace{p(\mathcal{M}_i)}_{\text{Prior}}\underbrace{p(\mathcal{D}|\mathcal{M}_i)}_{\boxed{\text{Model evidence or marginal likelihood}}}.$$

Bayes Factor: ratio of evidence for two models

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

50

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.

3/20/2021

51

Bayesian Model Comparison (3)

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

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

Note that

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

3/20/2021

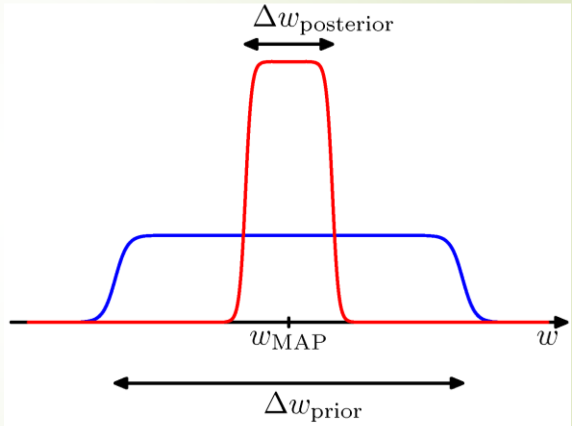
52

Bayesian Model Comparison (4)

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

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

where the posterior is assumed to be sharply peaked.



3/20/2021

53

Bayesian Model Comparison (5)

Taking logarithms, we obtain

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

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

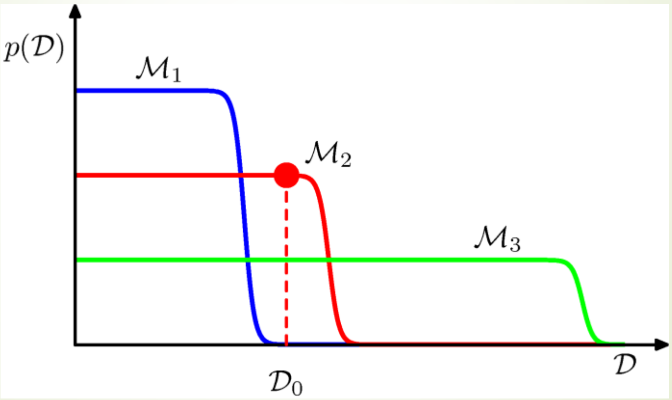
$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\mathbf{w}_{\mathrm{MAP}}) + \underbrace{M\ln\left(\frac{\Delta w_{\mathrm{posterior}}}{\Delta w_{\mathrm{prior}}}\right)}_{\text{Negative and linear in } M}.$$

3/20/2021

54

Bayesian Model Comparison (6)

Matching data and model complexity



3/20/2021

55

The Evidence Approximation (1)

The fully Bayesian predictive distribution is given by

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

but this integral is intractable. Approximate with

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

where $\left(\widehat{\alpha},\widehat{\beta}\right)$ 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*.

3/20/2021

56

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

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

General results for Gaussian integrals give

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

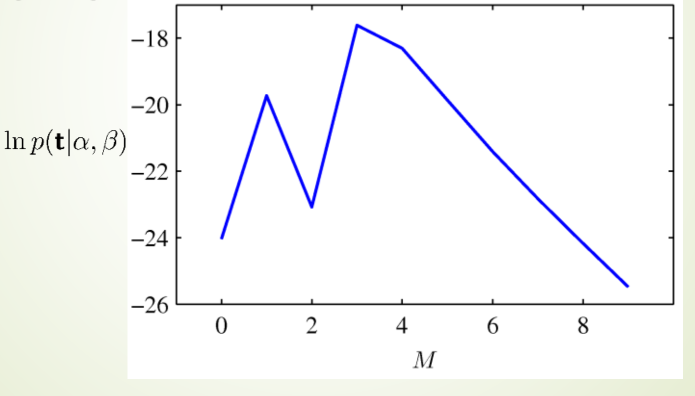
3/20/2021

57

The Evidence Approximation (3)

Example: sinusoidal data, M^{th} degree polynomial,

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



3/20/2021

58

Maximizing the Evidence Function (1)

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

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

Thus

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

has eigenvalues $\lambda_i + \alpha$.

3/20/2021

59

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

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

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

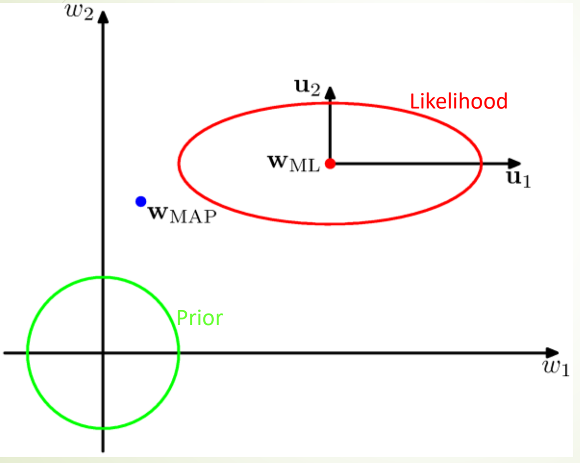
where $\gamma = \sum_i \frac{\lambda_i}{\alpha + \lambda_i}.$

N.B. ° depends on both α and β.

3/20/2021

60

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

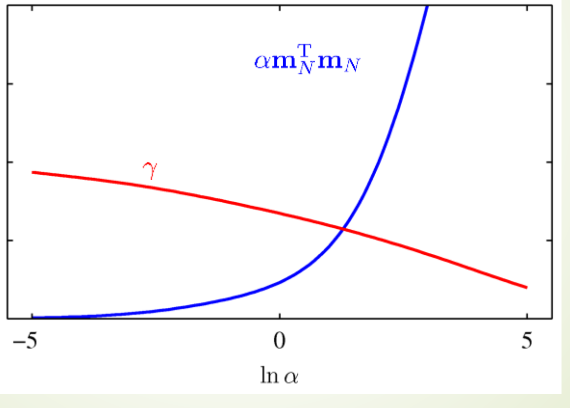
° is the number of well determined parameters

3/20/2021

61

Effective Number of Parameters (2)

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

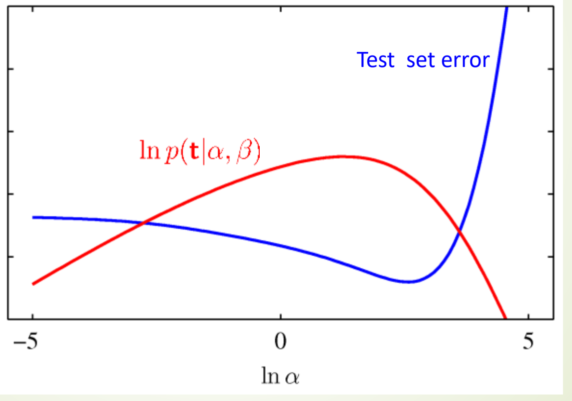


3/20/2021

62

Effective Number of Parameters (3)

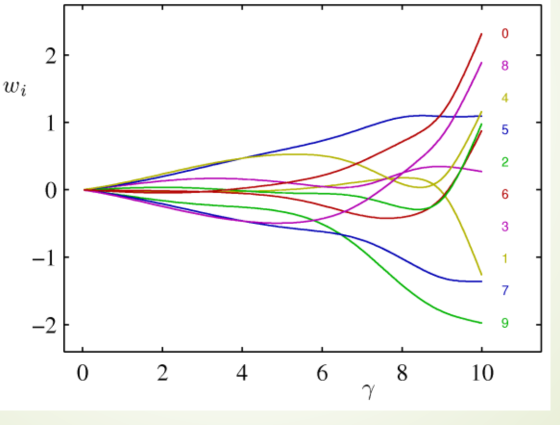
Example: sinusoidal data, 9 Gaussian basis functions, $\beta = 11.1$.



63

Effective Number of Parameters (4)

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



64

Effective Number of Parameters (5)

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

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

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

65

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.