



Pattern Recognition

Linear Models for Regression

林彥宇 教授

Yen-Yu Lin, Professor

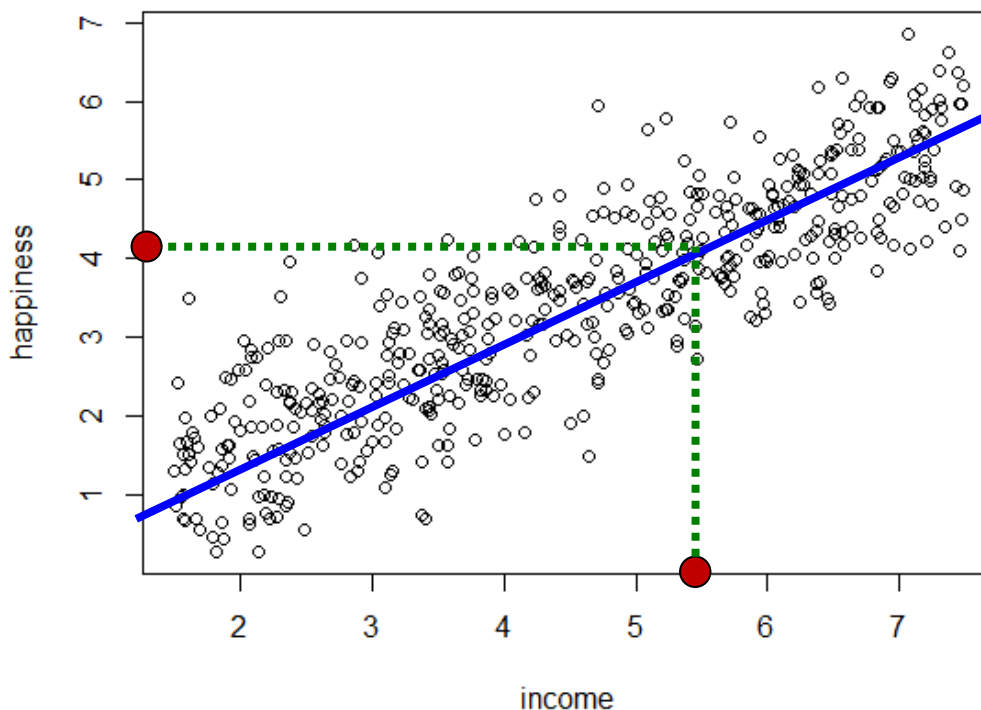
國立陽明交通大學 資訊工程學系

Computer Science, National Yang Ming Chiao Tung University

Some slides are modified from Prof. Sheng-Jyh Wang
and Prof. Hwang-Tzong Chen

Regression

- Given a training data set comprising N observations $\{\mathbf{x}_n\}_{n=1}^N$ and the corresponding target values $\{t_n\}_{n=1}^N$, the goal of regression is to predict the value of t for a new value of \mathbf{x}



<https://www.scribbr.com/statistics/linear-regression-in-r/>

A simple regression model

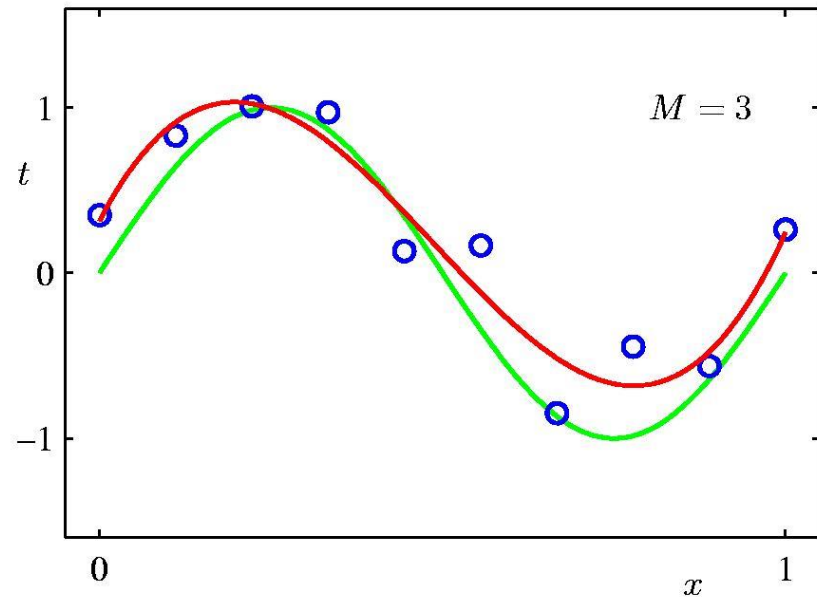
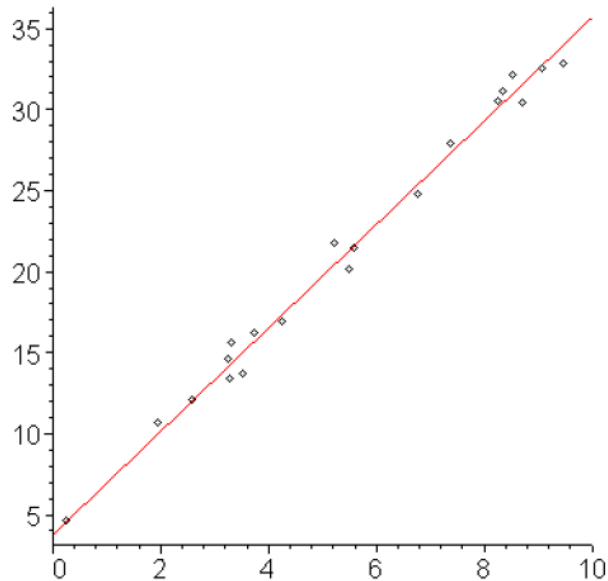
- A simple linear model:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1x_1 + \dots + w_Dx_D$$

- Each observation is in a D -dimensional space $\mathbf{x} = (x_1, \dots, x_D)^T$
- y is a regression model parametrized by $\mathbf{w} = (w_0, \dots, w_D)^T$
- The output is a **linear combination of the input variables**
- It is a **linear function of parameters**
- The fitting power is quite limited. Seeking **a nonlinear extension** for the input variables

An example

- A regressor in the form of $y(\mathbf{x}, \mathbf{w}) = w_0 + w_1x_1 + \dots + w_Dx_D$
 - A straight line in this case -> Insufficient fitting power
 - Nonlinear feature transforms before linear regression



Linear regression with nonlinear basis functions

- Simple linear model:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1x_1 + \dots + w_Dx_D$$

- A linear model with **nonlinear basis functions**

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

where $\{\phi_j\}_{j=1}^{M-1}$: nonlinear basis functions

M : the number of parameters

w_0 : the bias parameter allowing a fixed offset

- The regression output is a **linear combination** of **nonlinear basis functions of the inputs**

Linear regression with nonlinear basis functions

- A linear model with nonlinear basis functions

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

- Let $\phi_0(\mathbf{x}) = 1$, a dummy basis function. The regression function is equivalently expressed as

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

where $\mathbf{w} = (w_0, \dots, w_{M-1})^T$ and $\boldsymbol{\phi} = (\phi_0, \dots, \phi_{M-1})^T$

Examples of basis functions

- Polynomial basis function: taking the form of powers of x

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

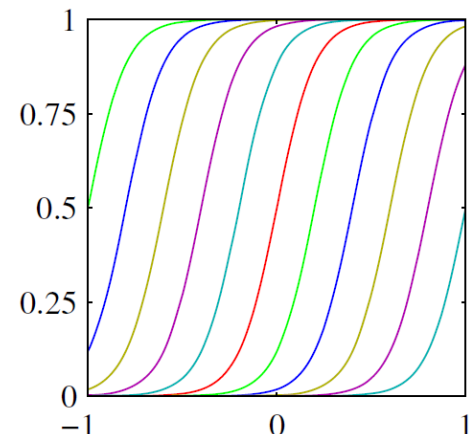
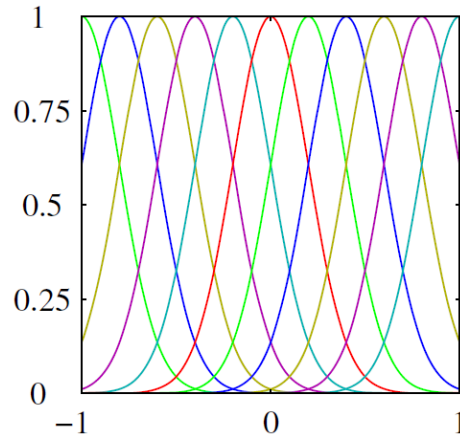
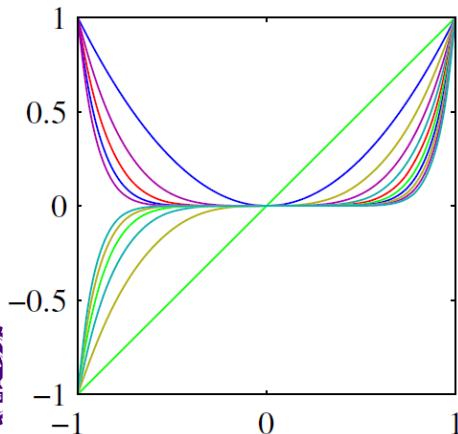
- Gaussian basis function: governed by μ_j and s

➤ μ_j governs the location while s governs the scale

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

- Sigmoidal basis function: governed by μ_j and s

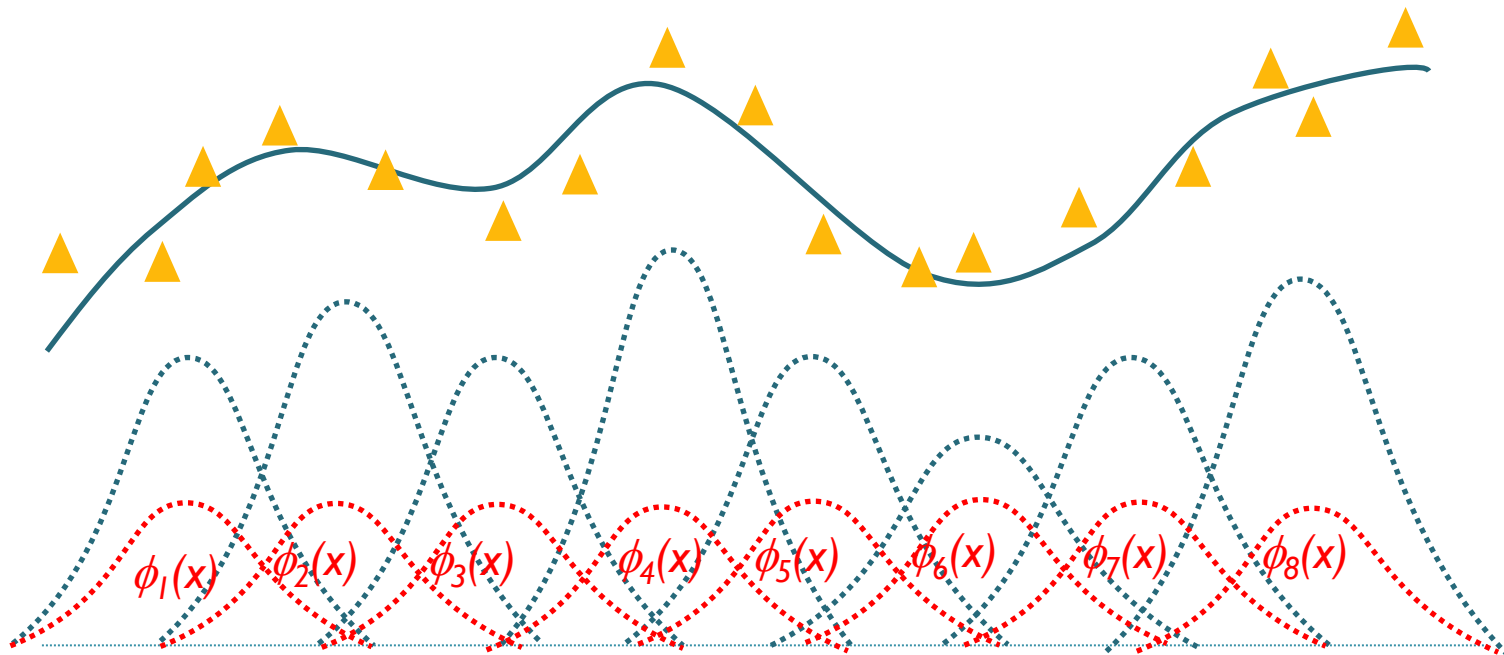
$$\phi_j(x) = \sigma \left(\frac{x - \mu_j}{s} \right) \text{ where } \sigma(a) = \frac{1}{1 + \exp(-a)}$$



How basis functions work

- Take Gaussian basis functions as an example

$$y = w_0 + w_1\phi_1(\mathbf{x}) + w_2\phi_2(\mathbf{x}) + \dots + w_{M-1}\phi_{M-1}(\mathbf{x})$$



Maximum likelihood and least squares

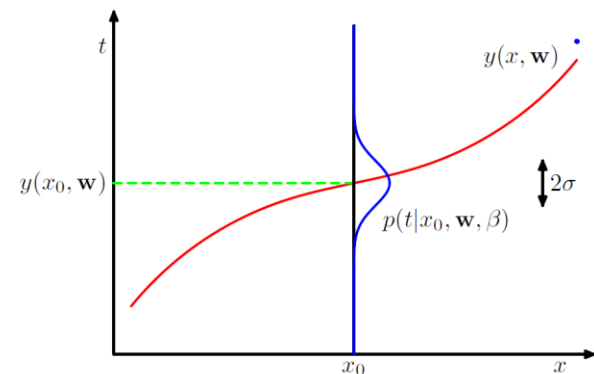
- Assume each observation is sampled from a deterministic function with an added Gaussian noise

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon$$

where ϵ is a zero mean Gaussian and precision (inverse variance) is β

- Thus, we have the conditional probability

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



Maximum likelihood and least squares

- Given a data set of inputs $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ with corresponding target values t_1, \dots, t_N , we have the **likelihood function**

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

- The **log likelihood function** is

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


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

Maximum likelihood and least squares

- Given a data set of inputs $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ with corresponding target values t_1, \dots, t_N , we have the **likelihood function**

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^N \mathcal{N}(t_n | \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1})$$
$$\mathcal{N}(x | \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2} (x - \mu)^2 \right\}$$
$$= \frac{\sqrt{\beta}}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \beta (x - \mu)^2 \right\}$$

- The **log likelihood function**

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


How?

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

Maximum likelihood and least squares

- Gaussian noise likelihood \Leftrightarrow sum-of-squares error function
- Maximum likelihood solution: Optimize \mathbf{w} by maximizing the log likelihood function
- Step 1: Compute the **gradient** of log likelihood w.r.t. \mathbf{w}

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

- Step 2: **Set the gradient to zero**, which gives

$$0 = \sum_{n=1}^N t_n \phi(\mathbf{x}_n)^T - \mathbf{w}^T \left(\sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \right)$$

Maximum likelihood and least squares

- Define the **design matrix** in this task

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

- It has N rows, one for each training sample
- It has M columns, one for each basis function

Maximum likelihood and least squares

- Setting the gradient to zero

$$0 = \sum_{n=1}^N t_n \phi(\mathbf{x}_n)^T - \mathbf{w}^T \left(\sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \right)$$

we have

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

- How to derive?

➤ Hint 1: $\sum_{n=1}^N t_n \phi^T(\mathbf{x}_n) = (\Phi^T \mathbf{t})^T$

➤ Hint 2:

$$\sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T = \sum_{n=1}^N \begin{pmatrix} \phi_0(\mathbf{x}_n) \\ \phi_1(\mathbf{x}_n) \\ \vdots \end{pmatrix} (\phi_0(\mathbf{x}_n), \phi_1(\mathbf{x}_n), \dots) = \Phi^T \Phi$$



Maximum likelihood and least squares

- The ML solution

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

- $\Phi^\dagger \equiv (\Phi^T \Phi)^{-1} \Phi^T$ is known as the **Moore-Penrose pseudo-inverse** of the design matrix Φ
- Φ has linearly independent columns. Why is $\Phi^T \Phi$ invertible?

Suppose that $\Phi^T \Phi$ is not invertible. $\exists \mathbf{v} \neq 0$ such that $\Phi^T \Phi \mathbf{v} = 0$.

$$\mathbf{v}^T \Phi^T \Phi \mathbf{v} = 0$$

$$\|\Phi \mathbf{v}\|^2 = 0$$

$$\Phi \mathbf{v} = 0$$

$\Rightarrow \Phi$ columns linearly dependent. A contradiction.



Maximum likelihood and least squares

- Similarly, β is optimized by maximizing the log likelihood

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

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

- We get

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

Regression for a new data point

- The conditional probability (likelihood function)

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

- After learning, we get $\mathbf{w} \leftarrow \mathbf{w}_{\text{ML}}$ and $\beta \leftarrow \beta_{\text{ML}}$
- Specify the prediction of a data point \mathbf{x} in the form of a Gaussian distribution with mean $y(\mathbf{x}, \mathbf{w}_{\text{ML}})$ and variance β_{ML}^{-1}

Regularized least squares

- Add a regularization term helps alleviate over-fitting

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

- The simplest form of the regularization term

$$E_W(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}$$

- The total error function becomes

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

- Setting the gradient of the function w.r.t. \mathbf{w} to 0, we have

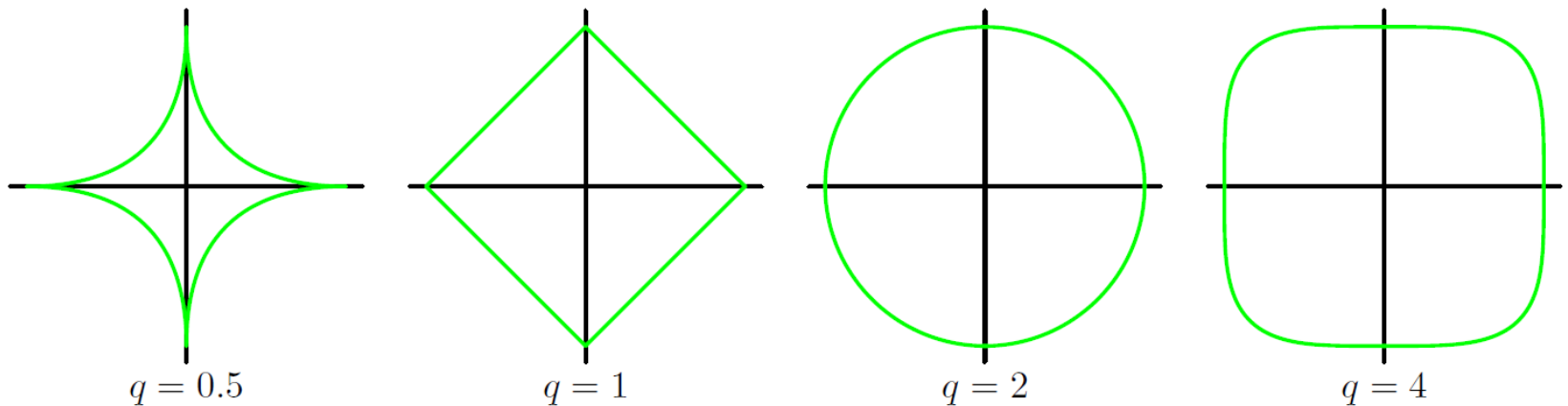
$$\mathbf{w} = (\lambda \mathbf{I} + \Phi^T \Phi)^{-1} \Phi^T \mathbf{t}.$$

Regularized least squares

- A more general regularizer

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

- $q=2 \rightarrow$ quadratic regularizer
- $q=1 \rightarrow$ the **lasso** in the statistics literature
- Contours of the regularization term



Multiple outputs

- In some applications, we wish to predict $K > 1$ target values
 - One target value: Income -> Happiness
 - Multiple target values: Income -> Happiness, Hours of duty, Health
- Recall the one-dimensional case

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

- With the same basis functions, the regression approach becomes

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{W}^T \boldsymbol{\phi}(\mathbf{x})$$

where \mathbf{W} is a $M \times K$ matrix, M is the number of basis functions, and K is the number of target values

Multiple outputs

- The conditional probability of a single observation is

$$p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) = \mathcal{N}(\mathbf{t}|\mathbf{W}^T \phi(\mathbf{x}), \beta^{-1} \mathbf{I})$$

- An isotropic Gaussian, i.e., with a diagonal covariance matrix
 - Each pair of variables are independent
- The log likelihood function is

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



Multiple outputs: Maximum likelihood solution

- Setting the gradient of the log likelihood function w.r.t. \mathbf{W} to 0, we have

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

- Consider the k th column of \mathbf{W}_{ML}

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

where \mathbf{t}_k is a N -dimensional vector with components $[t_{nk}]$

- It leads to K independent regression problems

Sequential learning

- The maximum likelihood derivation is a **batch** technique
 - It takes all training data into account at the same time
 - Case 1: The training data set is sufficiently large
 - Case 2: Data points are arriving in a continuous stream
- For the two cases, it is worthwhile to use **sequential algorithms, or on-line algorithms**, in which the data points are considered one by one, and the model parameters are updated incrementally



Sequential learning

- Stochastic gradient descent

- Error function comprises a sum over data points $E = \sum_n E_n$

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

- Given data point \mathbf{x}_n , the parameter vector \mathbf{w} is updated by

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n$$

where τ is the iteration number and η is the **learning rate**

- In the case of sum-of-squares error, it is

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)T} \phi_n) \phi_n$$

Maximum a posterior

- Likelihood function

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

- Let's consider a prior function, which is a Gaussian

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

where \mathbf{m}_0 is the mean and \mathbf{S}_0 is the covariance matrix

- The posterior function is also a Gaussian

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

where $\mathbf{m}_N = \mathbf{S}_N (\mathbf{S}_0^{-1} \mathbf{m}_0 + \beta \Phi^T \mathbf{t})$ is the mean

and $\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta \Phi^T \Phi$ is the covariance

How to derive the mean and covariance in posterior

- According to the marginal and conditional Gaussians on page 93 of the PRML textbook

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{Ax} + \mathbf{b}, \mathbf{L}^{-1})$$

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

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



A zero-mean isotropic Gaussian prior

- A general Gaussian **prior** function

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

where \mathbf{m}_0 is the mean and \mathbf{S}_0 is the covariance matrix

- A widely used Gaussian prior

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

- Mean and covariance of the resulting posterior function

$$\begin{array}{ll} \mathbf{m}_N &= \mathbf{S}_N (\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\Phi^T\mathbf{t}) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta\Phi^T\Phi \end{array} \quad \Rightarrow \quad \begin{array}{ll} \mathbf{m}_N &= \beta\mathbf{S}_N\Phi^T\mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha\mathbf{I} + \beta\Phi^T\Phi \end{array}$$

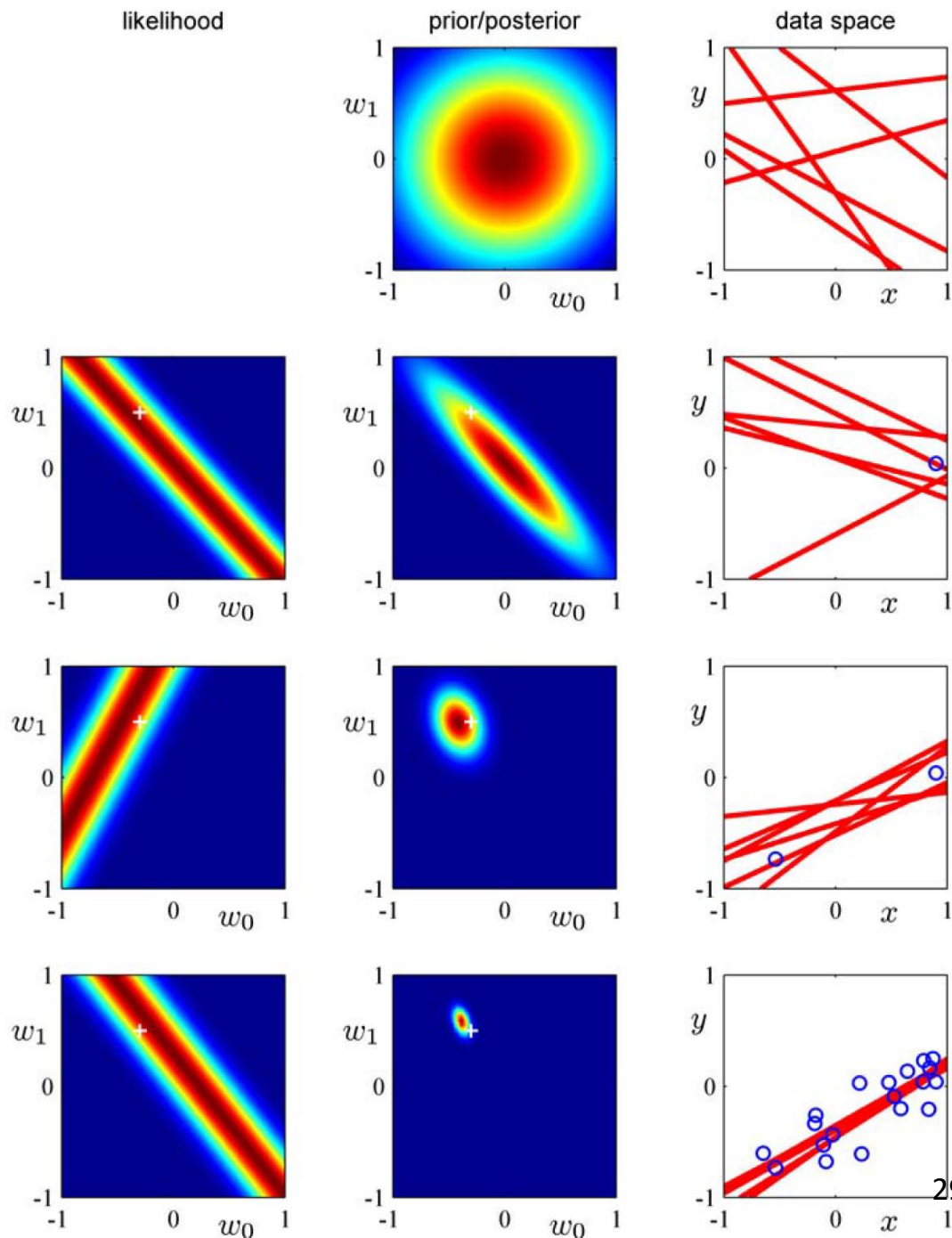
Sequential Bayesian learning: An example

- Data, including observations and target values, are given one-by-one
- Data are in a one-dimensional space
- Data are sampled from the function $f(x, \mathbf{a}) = a_0 + a_1x$, where $a_0 = -0.3$ and $a_1 = 0.5$, and added by a Gaussian noise
 - Note that the function is **unknown**
 - We have just the observations and the target values

An example

- Regression function

$$y(x, \mathbf{w}) = w_0 + w_1 x$$



likelihood

An example

- Regression function

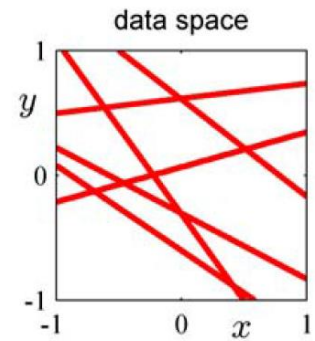
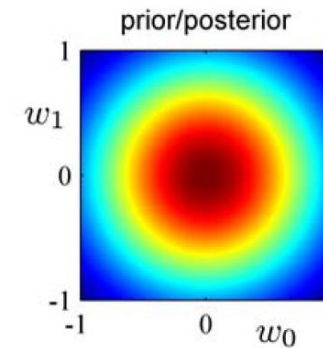
$$y(x, \mathbf{w}) = w_0 + w_1 x$$

- In the beginning, no data are available

- Constant likelihood

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

- Sample 6 curves for function according to posterior distribution



An example

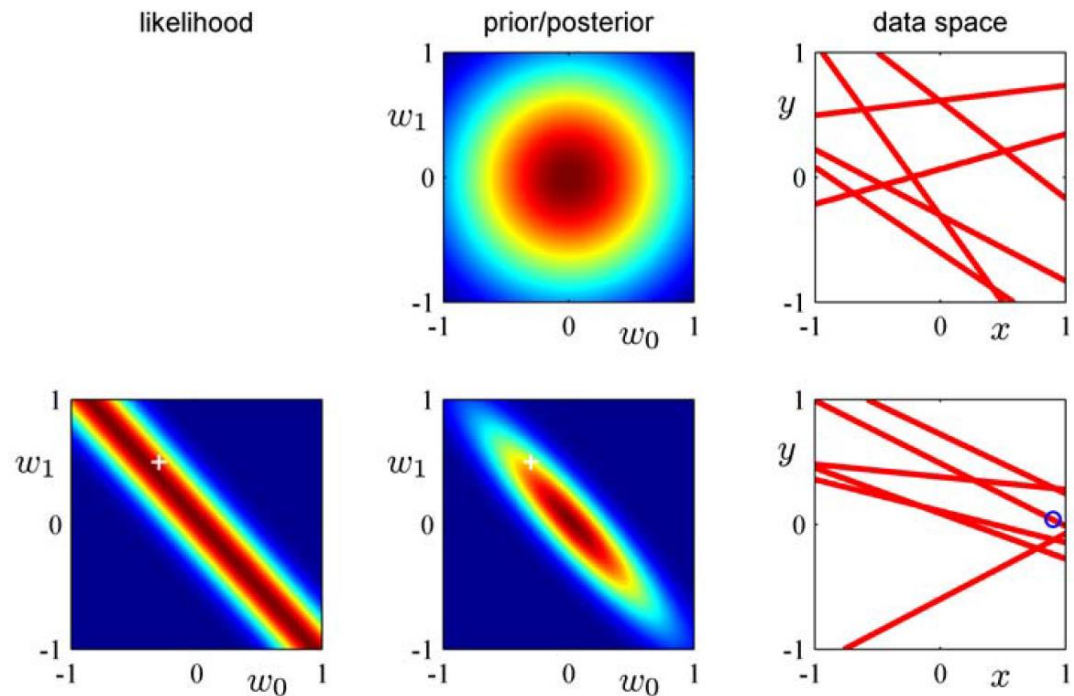
- Regression function

$$y(x, \mathbf{w}) = w_0 + w_1 x$$

- One data (blue circle) sample is given

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

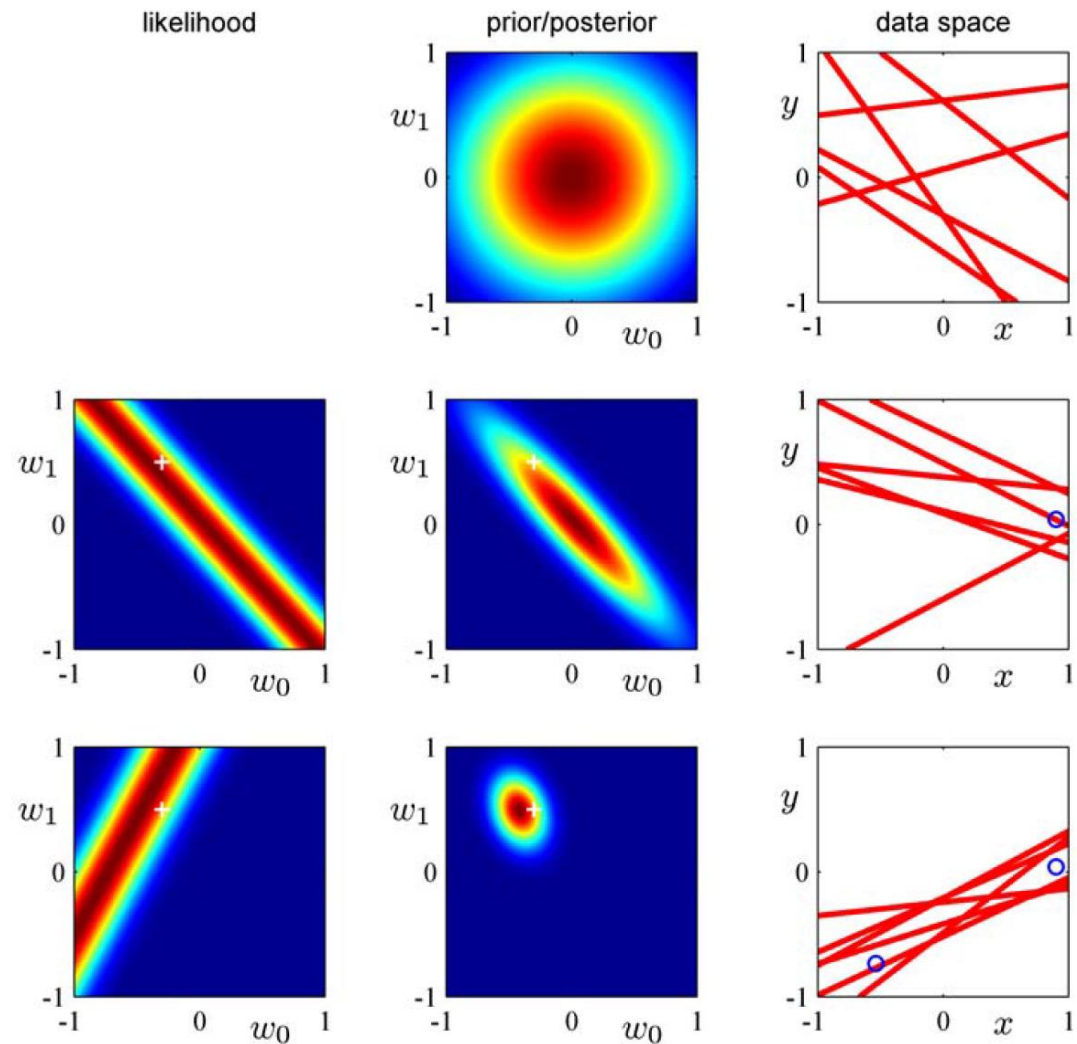
- White cross
- Posterior proportional to likelihood x prior
- Sample 6 curves according to posterior



An example

- Regression function

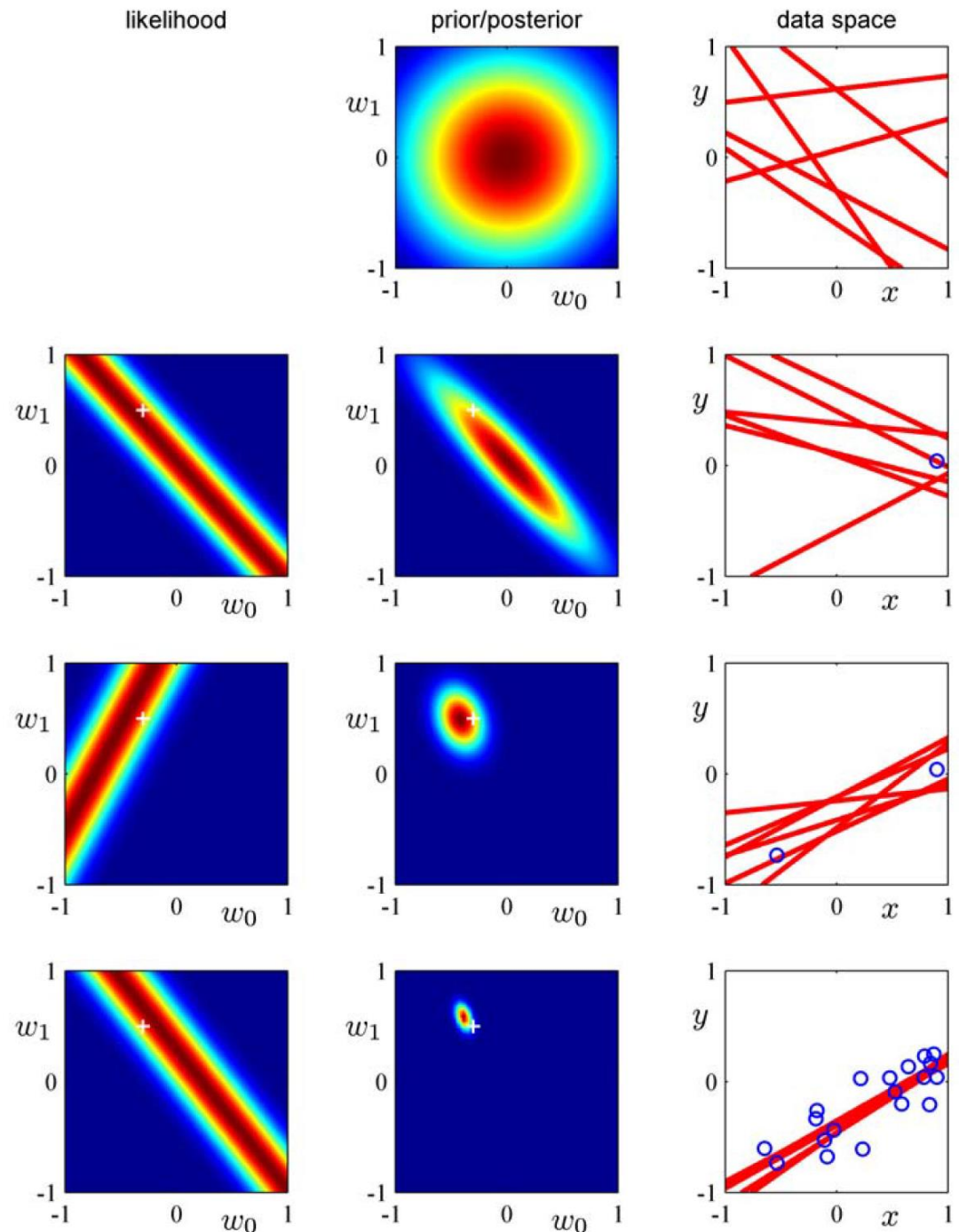
$$y(x, \mathbf{w}) = w_0 + w_1 x$$
- Second data (blue circle) sample is given
- Likelihood for the second sample
- White cross
- Posterior proportional to likelihood x prior
- Sample 6 curves according to posterior



An example

- Regression function

$$y(x, \mathbf{w}) = w_0 + w_1 x$$
- 20 data (blue circle) sample are given
- Likelihood for the 20th sample
- White cross
- Posterior proportional to likelihood x prior
- Sample 6 curves according to posterior



Predictive distribution

- Recall the posterior function

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

where $\mathbf{m}_N = \mathbf{S}_N (\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\Phi^T\mathbf{t})$ and $\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\Phi^T\Phi$

- Given \mathbf{w} , we regress a data sample via

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

- In Bayesian treatment, the predictive distribution is

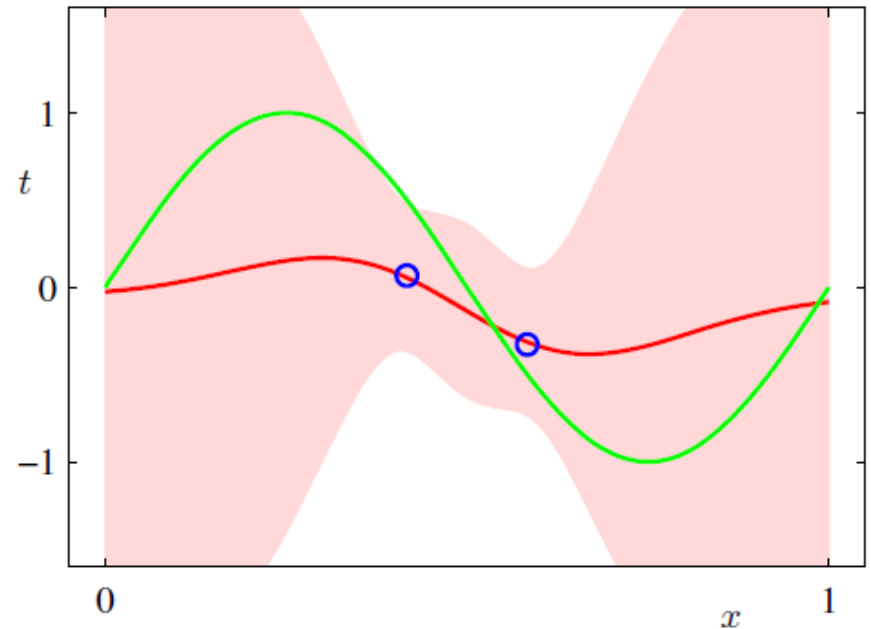
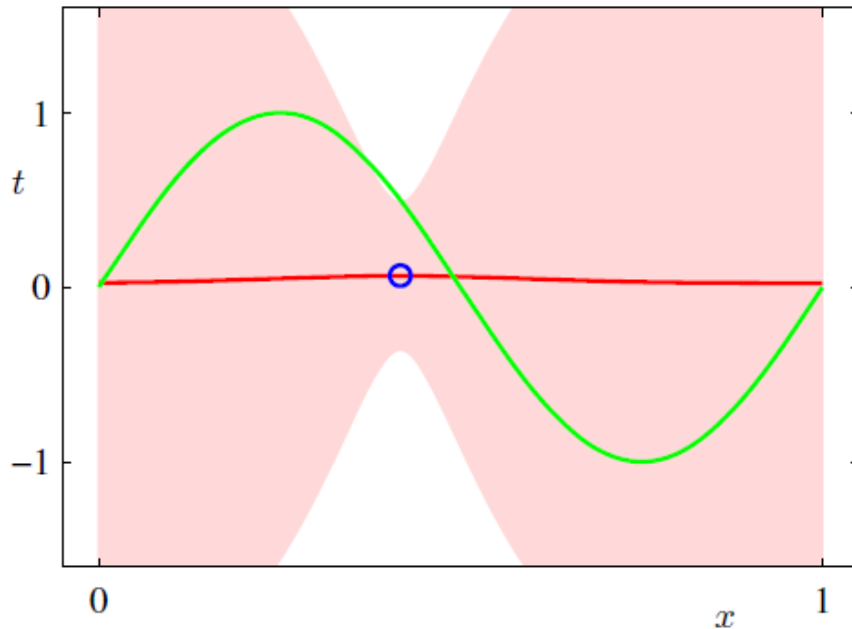
$$p(t|\mathbf{t}, \alpha, \beta) = \int p(t|\mathbf{w}, \beta)p(\mathbf{w}|\mathbf{t}, \alpha, \beta) d\mathbf{w}$$

- Then we have

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

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



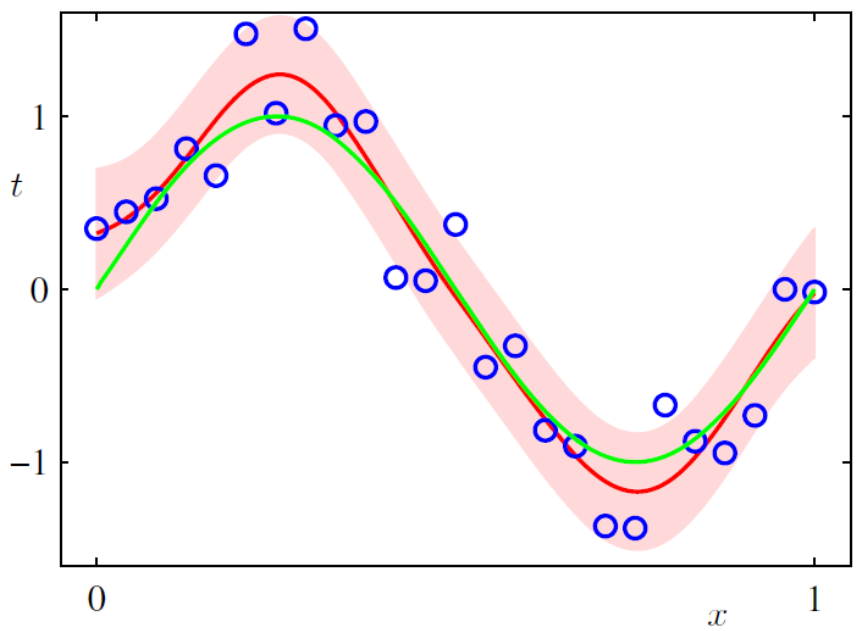
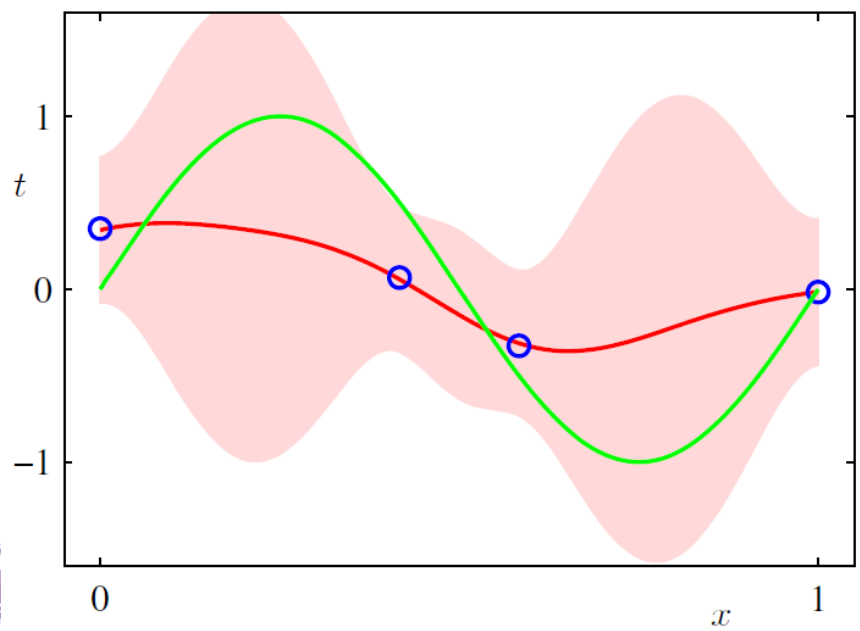
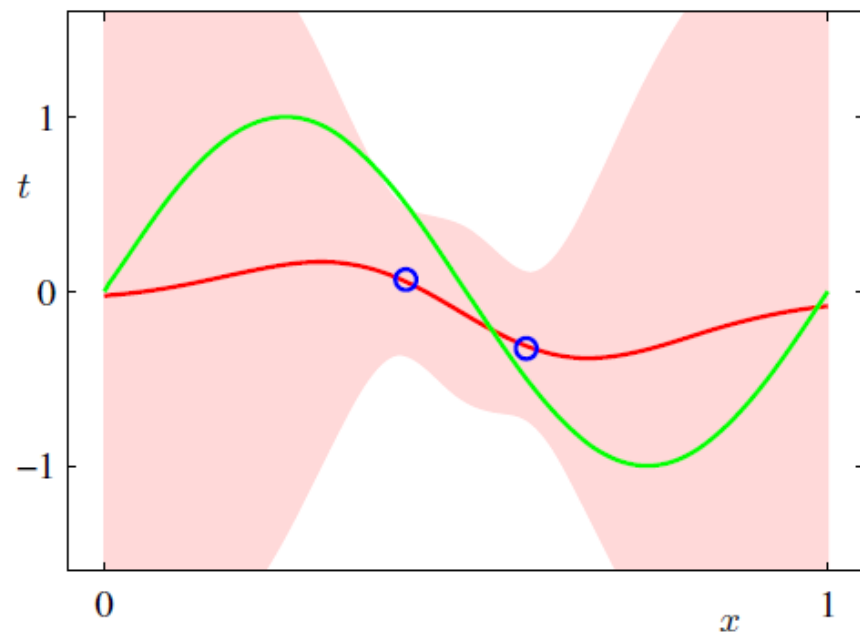
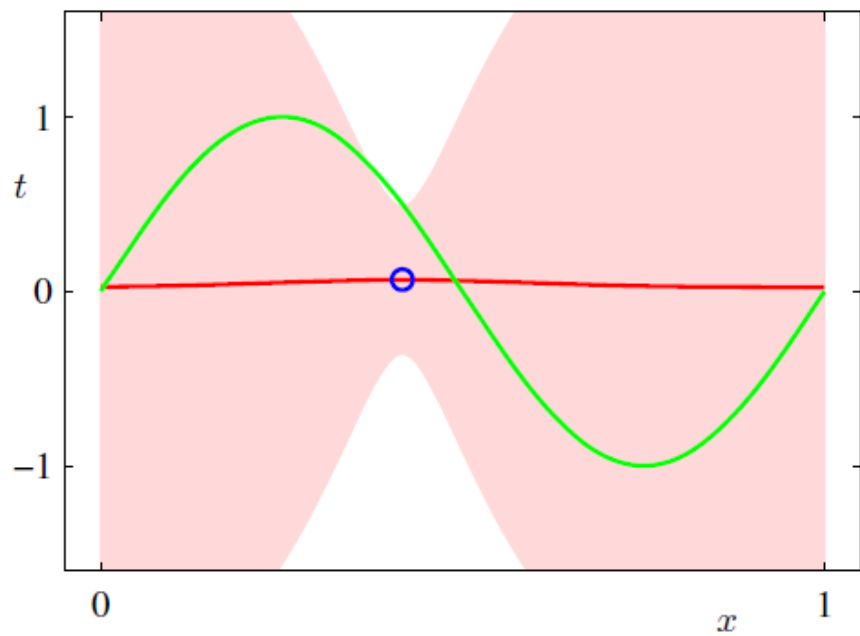


- **Green curve** $\sin(2\pi x)$ is used to sample data. It is unknown
- **Blue circle**: a sampled data
- After learning, the predictive distribution

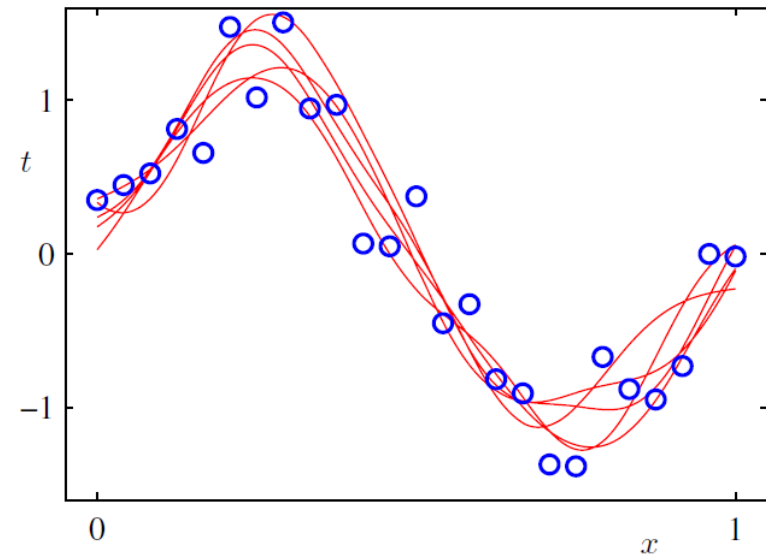
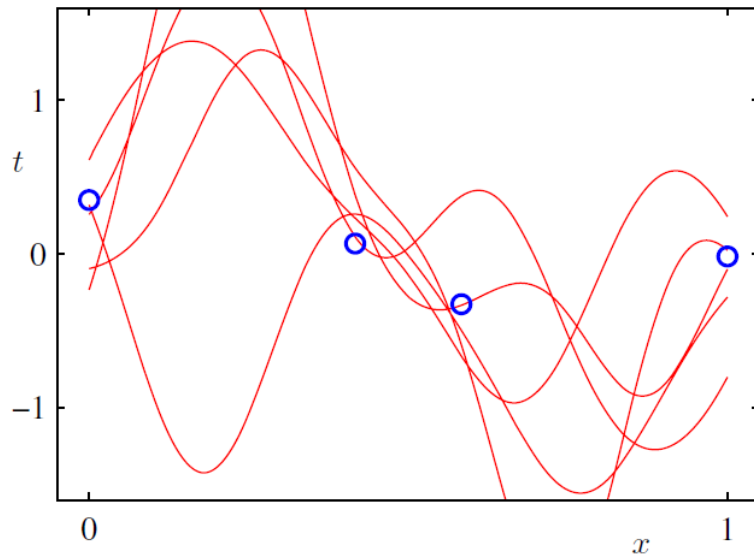
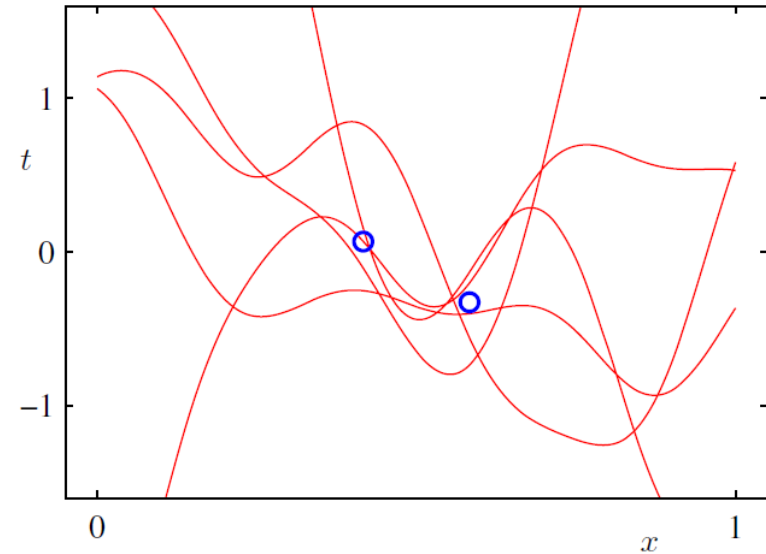
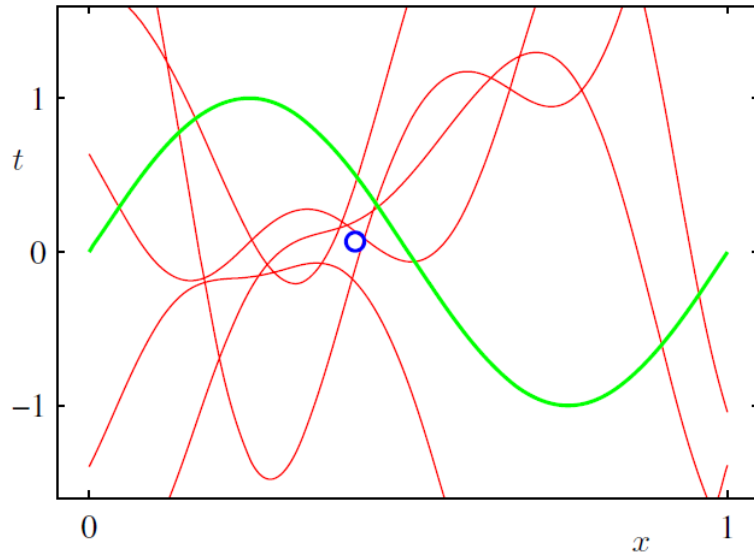
$$p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(t|\mathbf{m}_N^T \phi(\mathbf{x}), \sigma_N^2(\mathbf{x}))$$

- **Red curve**: the mean of the Gaussian above
- **Red shaded region**: One standard deviation on either side of mean





- Sample 5 points of \mathbf{w} according to the posterior function
- Plot the corresponding regression functions $y(x, \mathbf{w})$



References

- Chapters 3.1 and 3.3 in the PRML textbook

Thank You for Your Attention!

THANK YOU FOR YOUR ATTENTION!

Yen-Yu Lin (林彥宇)

Email: lin@cs.nctu.edu.tw

URL: <https://www.cs.nycu.edu.tw/members/detail/lin>

