Artificial Intelligence

Linear Models for Regression

Donghui Wang
AI Institute@ZJU
2015.03

Contents

- Linear basis function models
- Bayesian linear regression

References:

1. *Bishop. "Pattern Recognition and Machine Learning", Chapter 3. 2006.*

Linear basis function models

Linear basis function models

- Regression:

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

- Linear regression:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D \qquad \text{where } \mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$$

- Linear basis function model:
  - Linear combinations of fixed nonlinear functions of the input variables

Bias parameter

Basis function

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

$$\phi_0(\mathbf{x}) = 1$$

$$\boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^{\mathrm{T}}$$

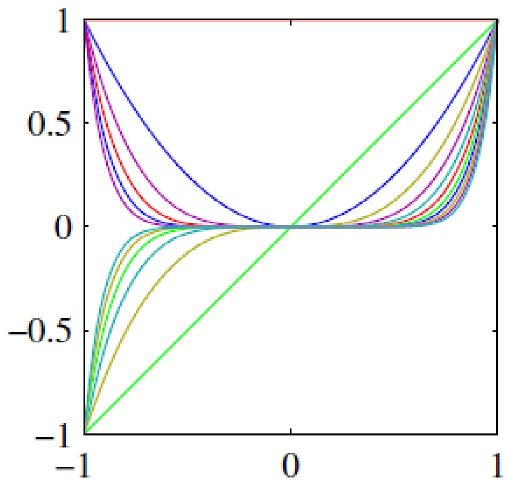
$$\mathbf{w} = (w_0, \ldots, w_{M-1})^{\mathrm{T}}$$

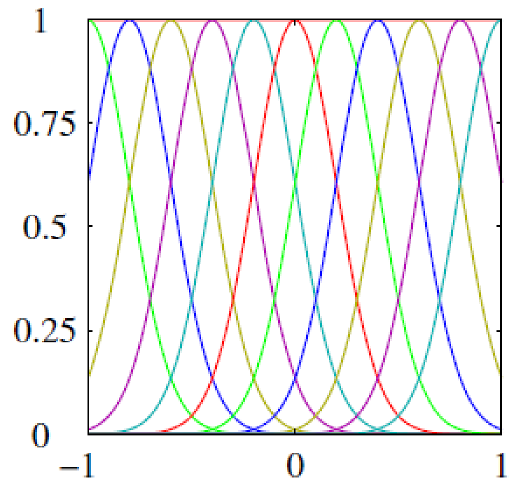
$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

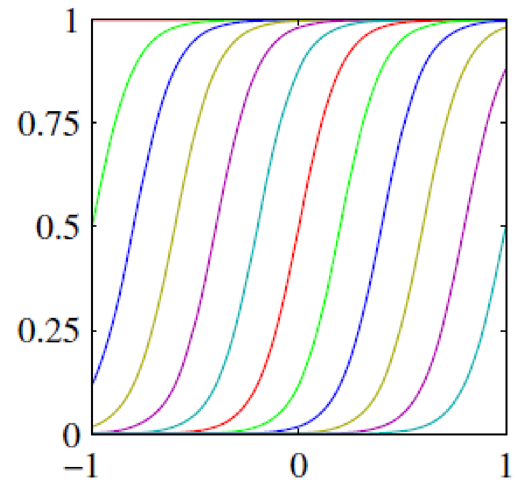
Typical basis functions

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}) \qquad \mathbf{w} = (w_0, \ldots, w_{M-1})^{\mathrm{T}} \qquad \begin{array}{c} \phi_0(\mathbf{x}) = 1 \\ \boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^{\mathrm{T}} \end{array}$$

- Polynomial basis function: $\phi_j(x) = x^j$
- '*Gaussian*' basis function: $\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$
- sigmoid basis function: $\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right)$ $\qquad \sigma(a) = \frac{1}{1+\exp(-a)}$
- Fourier basis / wavelets basis







Maximum likelihood and least squares

- Assume: $$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \qquad y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$
- Thus: $$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \longrightarrow \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x}) \, \mathrm{d}t = y(\mathbf{x}, \mathbf{w})$$
- For data set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and target vector $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$, the likelihood function:

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

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

SSE: sum-of-squares error function

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

Maximum likelihood and least squares

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

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

- Solving w by ML:

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

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

$$\Rightarrow \quad \mathbf{w}_{\mathrm{ML}} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}$$

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

$$\mathbf{\Phi}^{\dagger} \equiv \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}$$ *Moore-Penrose pseudo-inverse*

Maximum likelihood and least squares

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

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

- About w_0:

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - w_0 - \sum_{j=1}^{M-1} w_j\phi_j(\mathbf{x}_n)\}^2 \Rightarrow w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j\overline{\phi_j}$$

$$\bar{t} = \frac{1}{N}\sum_{n=1}^{N} t_n$$

$$\overline{\phi_j} = \frac{1}{N}\sum_{n=1}^{N}\phi_j(\mathbf{x}_n)$$

- Solving β by ML:

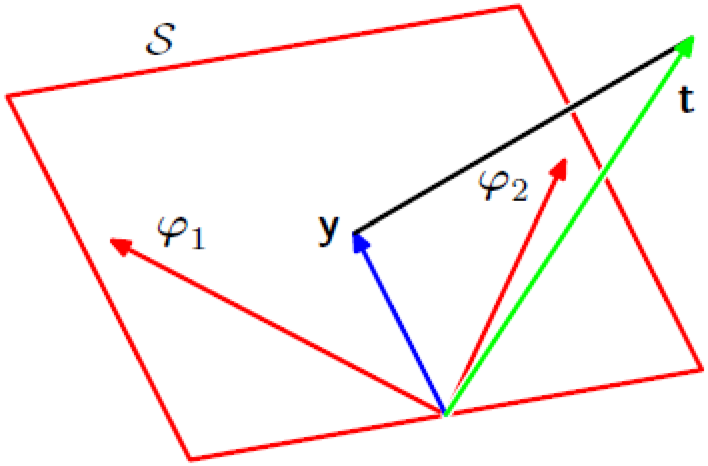
$$\frac{N}{2\beta} = E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 \Rightarrow \frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{t_n - \mathbf{w}_{\mathrm{ML}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

Geometry of least squares

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

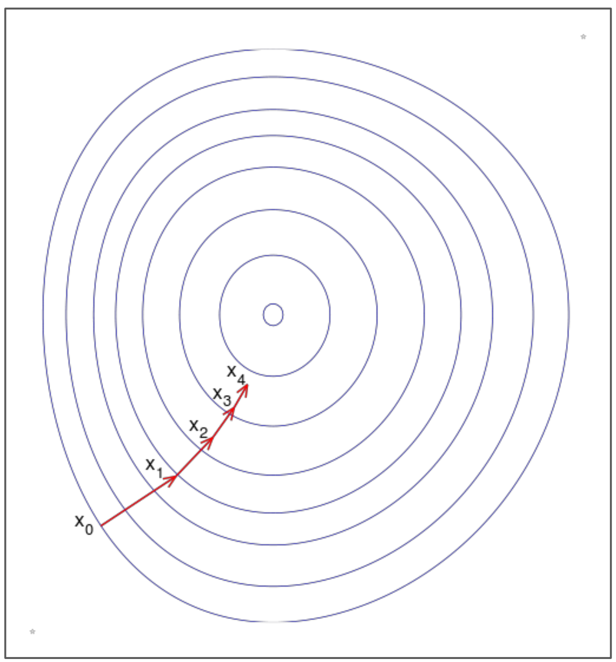
$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 \qquad \mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}$$

Geometrical interpretation of the least-squares solution, in an N-dimensional space whose axes are the values of $t_1, \ldots, t_N$. The least-squares regression function is obtained by finding the orthogonal projection of the data vector **t** onto the subspace spanned by the basis functions $\phi_j(\mathbf{x})$ in which each basis function is viewed as a vector φ_j of length N with elements $\phi_j(\mathbf{x}_n)$.



Sequential learning

- Gradient descent
  - Gradient descent is based on the observation that if the multivariable function J(**w**) is defined and differentiable in a neighborhood of a point $\mathbf{w}_0$, then J(**w**) decreases *fastest* if one goes from $\mathbf{w}_0$ in the direction of the negative gradient of J(.) at $\mathbf{w}_0$, $-J(\mathbf{w}_0)$.



Sequential learning

- Stochastic gradient descent (sequential gradient descent)

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n \qquad n = 1, 2, \ldots, N$$

Learning rate

Error function

  - least-mean-squares or the LMS algorithm

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

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

Sequential learning

- Batch gradient descent:

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

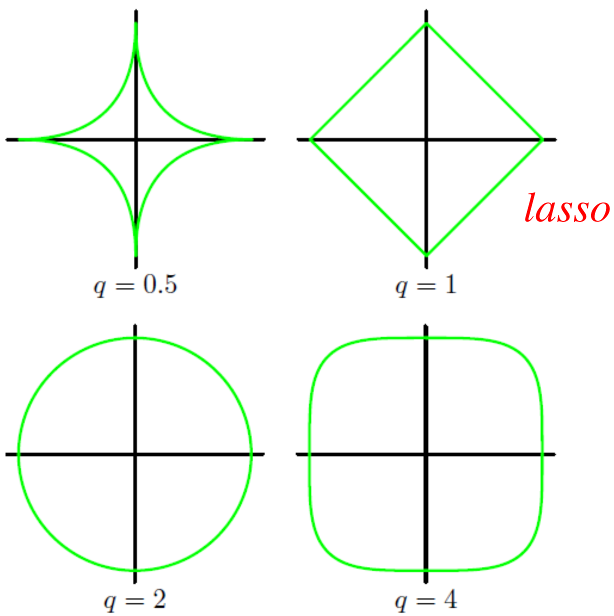
Regularized least squares

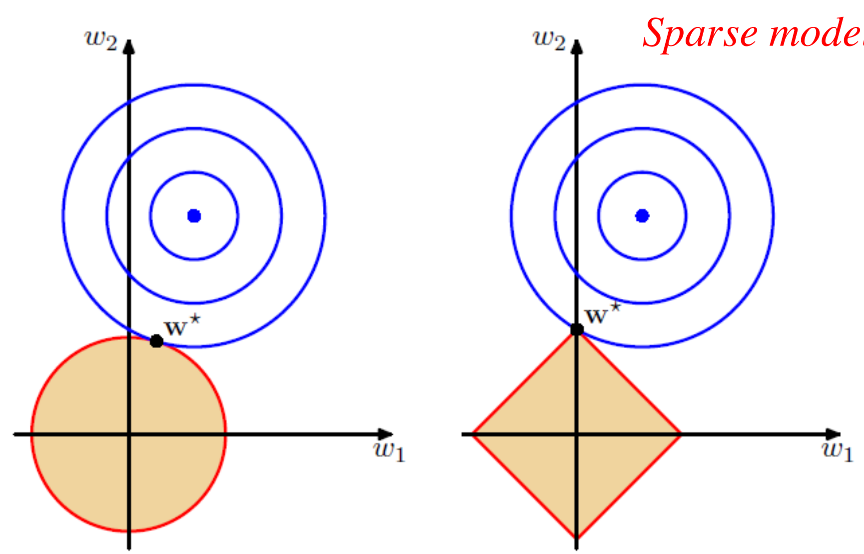
- Error function with regularization term:

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \quad \Rightarrow \quad \mathbf{w} = \left(\lambda\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}$$

- Weight decay:
  - parameter shrinkage method

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





Multiple outputs

M × K matrix of parameters

- Output K-dimensional target vector $\mathbf{y}$:

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

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

Multiple outputs

- Estimate $\mathbf{W}$ by ML:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n), \beta^{-1}) \qquad \mathbf{w}_{\mathrm{ML}} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}$$

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

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

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

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

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

Decision Theory

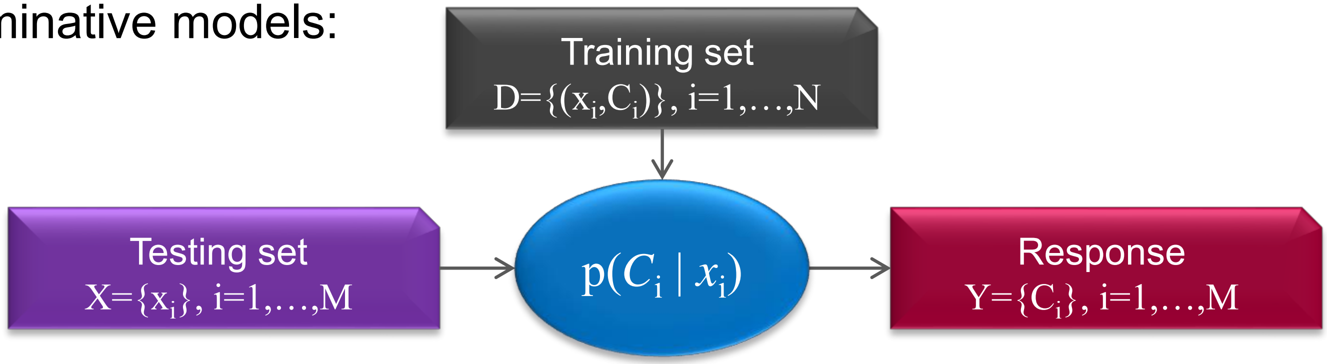
References:

1. *Bishop. "Pattern Recognition and Machine Learning", Chapter 1.5. 2006.*

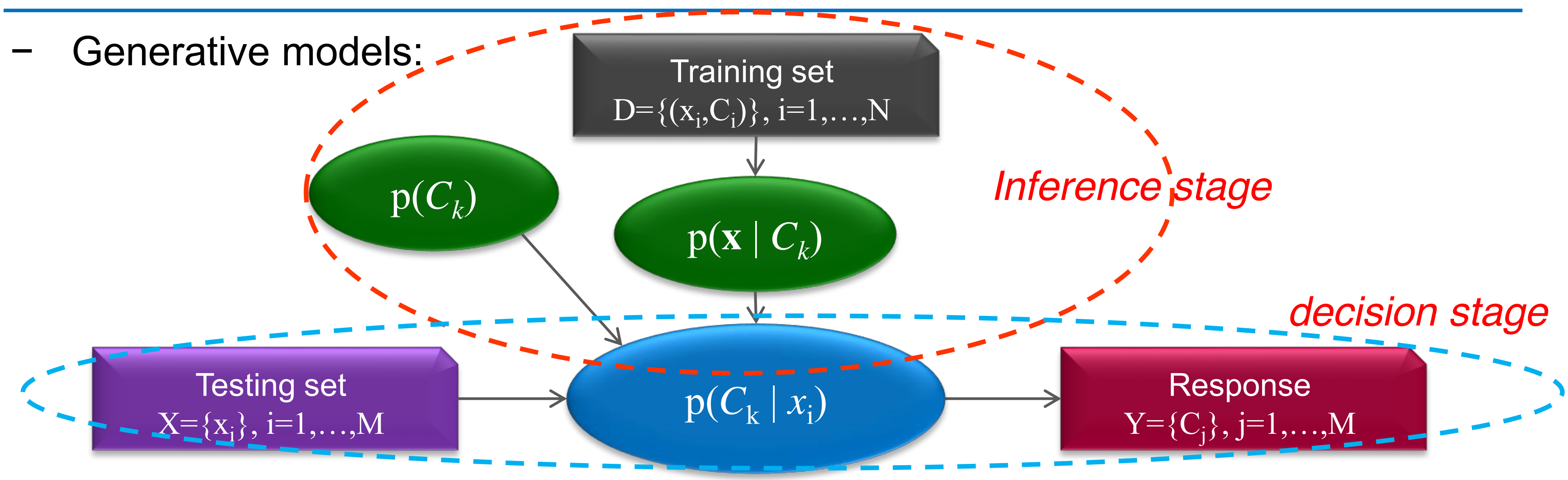
Decision Theory

- How to make optimal decisions in situations involving uncertainty?

– Discriminative models:



– Generative models:



Minimizing the misclassification rate

- Naïve Bayes classifier:

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}$$

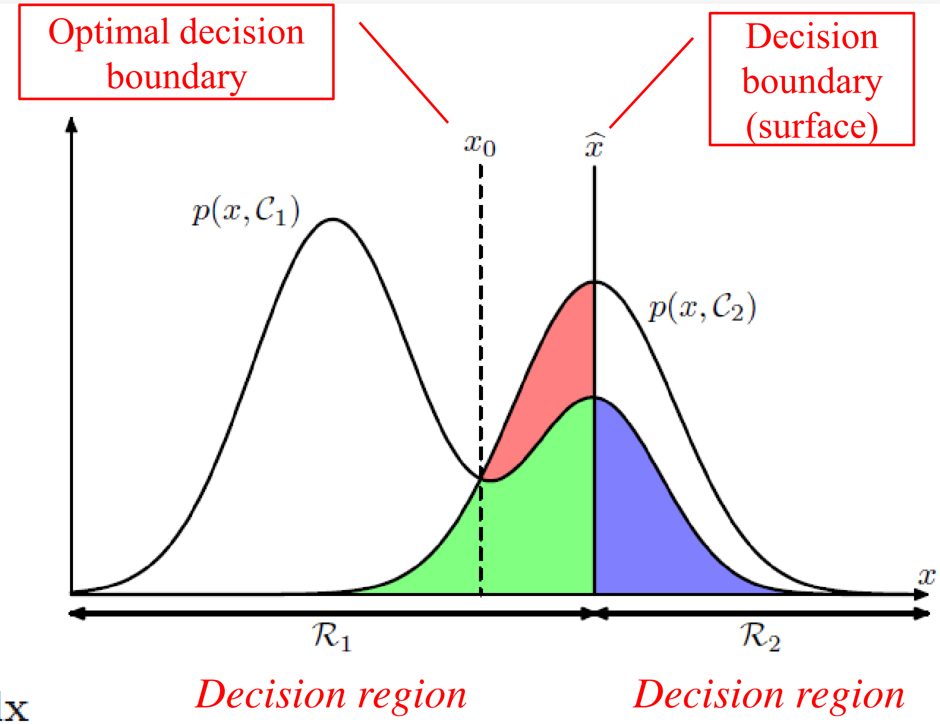
- Decision rule:

if $p(\mathrm{x}, C_1) > p(\mathrm{x}, C_2)$, assign x to class C_1

Or if $p(C_1|\mathrm{x}) > p(C_2|\mathrm{x})$, assign x to class C_1

- Misclassification rate $p(\text{mistake})$:

$$\begin{aligned} p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\ &= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\,\mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\,\mathrm{d}\mathbf{x} \end{aligned}$$



- Maximize correct classification rate:

$$p(\text{correct}) = \sum_{k=1}^{K} p(\mathbf{x} \in \mathcal{R}_k, \mathcal{C}_k) = \sum_{k=1}^{K} \int_{\mathcal{R}_k} p(\mathbf{x}, \mathcal{C}_k)\,\mathrm{d}\mathbf{x}$$

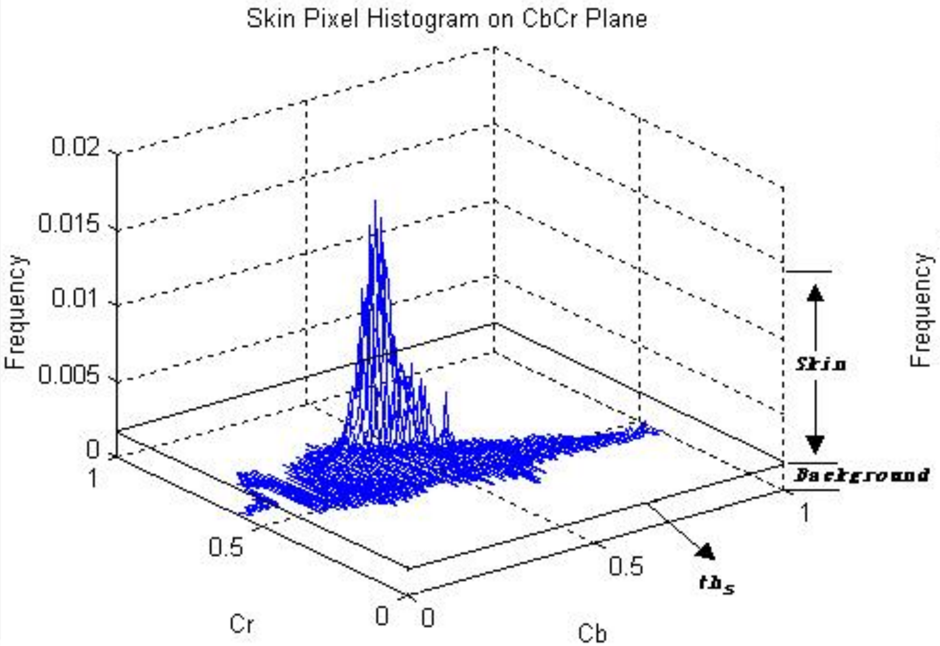
Naïve Bayes classifier

- Example: skin detection

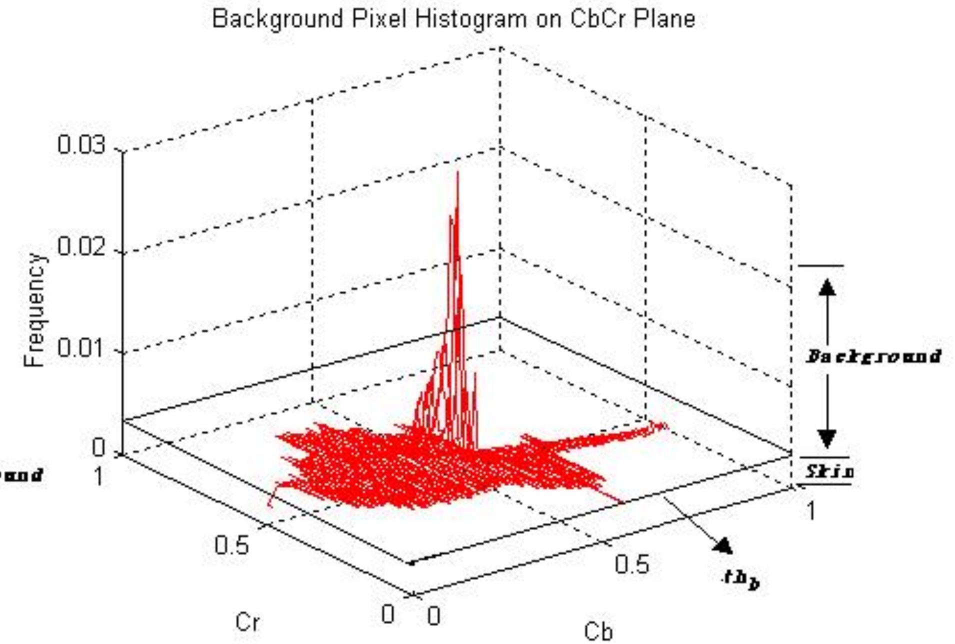
$$\frac{P(skin|c)}{P(\neg skin|c)} = \frac{P(c|skin)P(skin)}{P(c|\neg skin)P(\neg skin)}$$

$$\frac{P(c|skin)}{P(c|\neg skin)} > \Theta$$

$$\Theta = K \times \frac{1-P(skin)}{P(skin)}$$



$P(c|skin)$



$P(c|\neg skin)$

Minimizing the expected loss

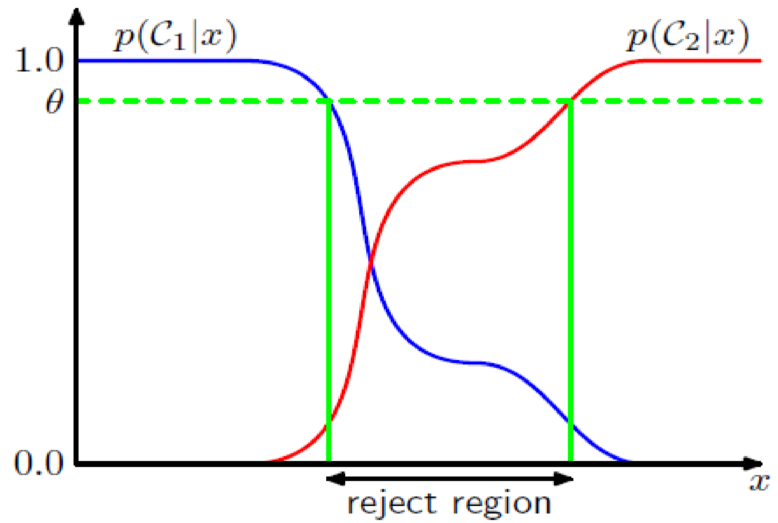
- Loss function (cost function) / utility function
  - Loss matrix: $\boldsymbol{L}$

$$\begin{array}{c} \\ \text{cancer} \\ \text{normal} \end{array} \begin{array}{c} \begin{array}{cc} \text{cancer} & \text{normal} \end{array} \\ \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} \end{array}$$

  - Minimize the average loss (expected loss): $\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) \, \mathrm{d}\mathbf{x}.$
  - Decision rule:

$$\sum_k L_{kj} p(\mathbf{x}, \mathcal{C}_k) \Rightarrow \sum_k L_{kj} p(\mathcal{C}_k | \mathbf{x})$$

- Reject option
  - Threshold θ
  - $\theta = 1$: reject all
  - For K classes, $\theta < 1/K$:
    no examples rejected



Loss function for regression

- Choice the squared loss as loss function, the average, or expected, loss is then given by:

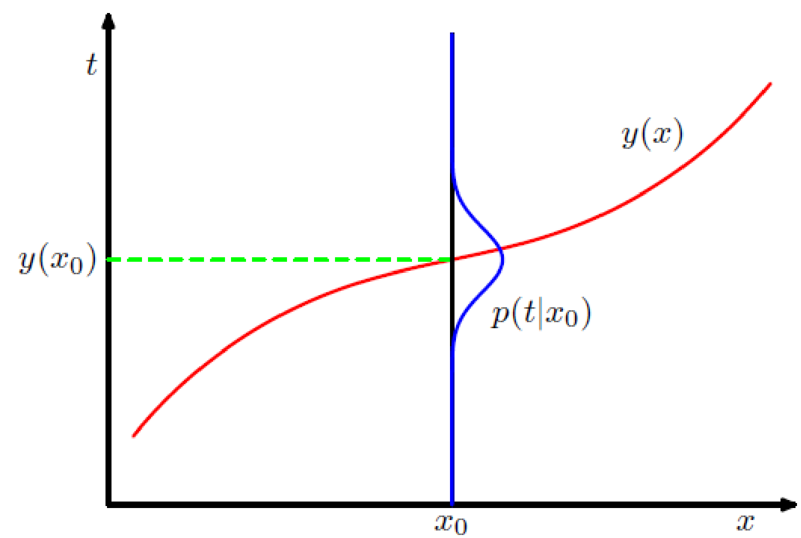
$$\mathbb{E}[L] = \iint L(t, y(\mathbf{x}))p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t = \iint \{y(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

$$\frac{\delta \mathbb{E}[L]}{\delta y(\mathbf{x})} = 2\int \{y(\mathbf{x}) - t\}p(\mathbf{x}, t)\,\mathrm{d}t = 0$$

$$\Rightarrow \quad y(\mathbf{x}) = \frac{\int t p(\mathbf{x}, t)\,\mathrm{d}t}{p(\mathbf{x})} = \int t p(t|\mathbf{x})\,\mathrm{d}t = \mathbb{E}_t[t|\mathbf{x}]$$

$$\mathbf{y}(\mathbf{x}) = \mathbb{E}_t[\mathbf{t}|\mathbf{x}]$$

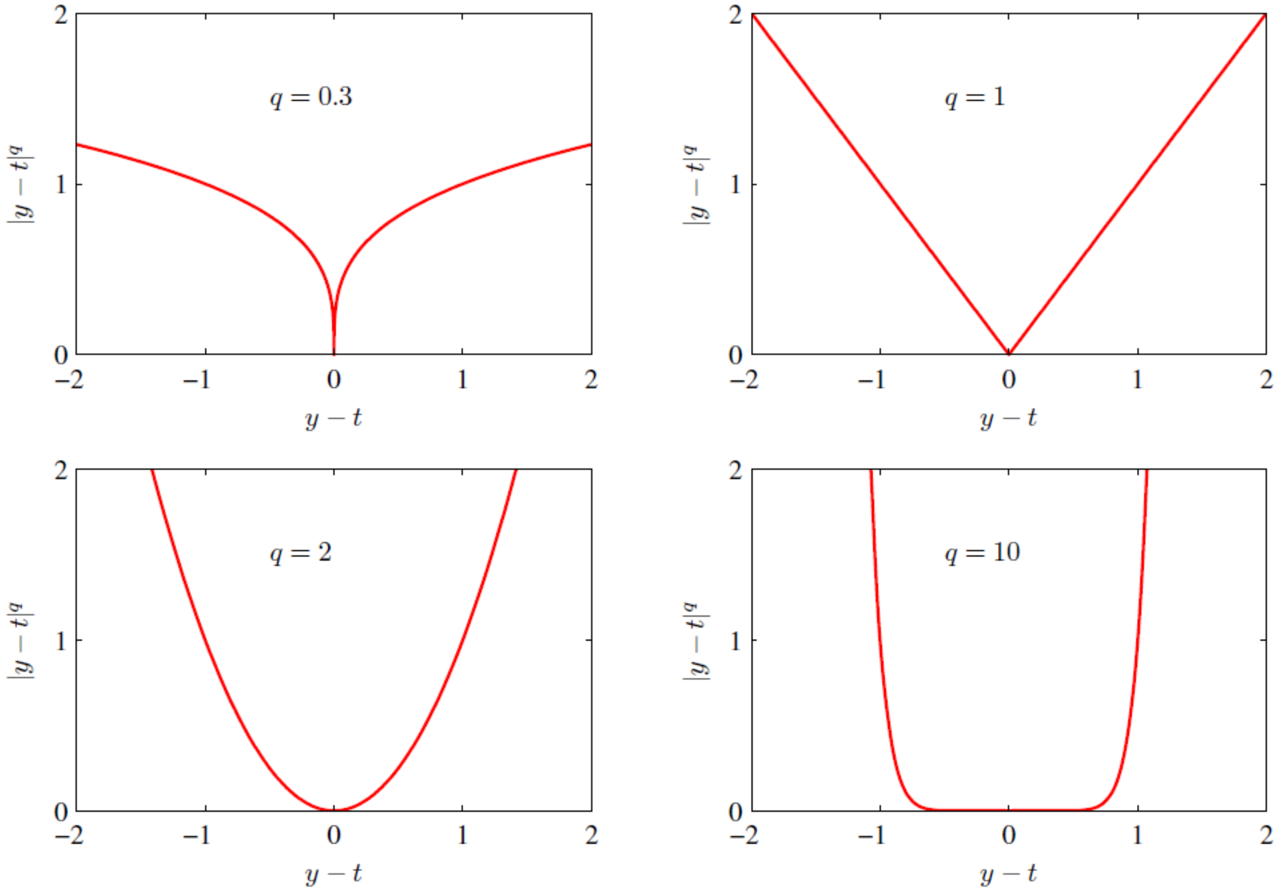
Regression function



Loss function for regression

- *Minkowski* loss :

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



The Bias-Variance Decomposition

References:

1. *Bishop. "Pattern Recognition and Machine Learning", Chapter 3. 2006.*

The Bias-Variance Decomposition

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

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

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

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

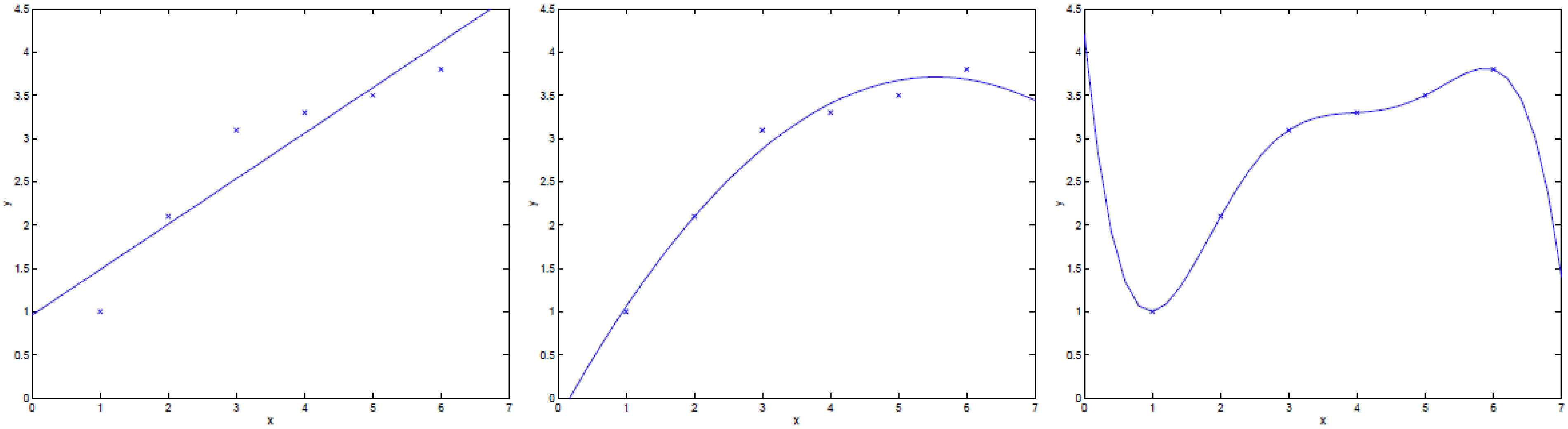
$$\Rightarrow \quad \mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$$

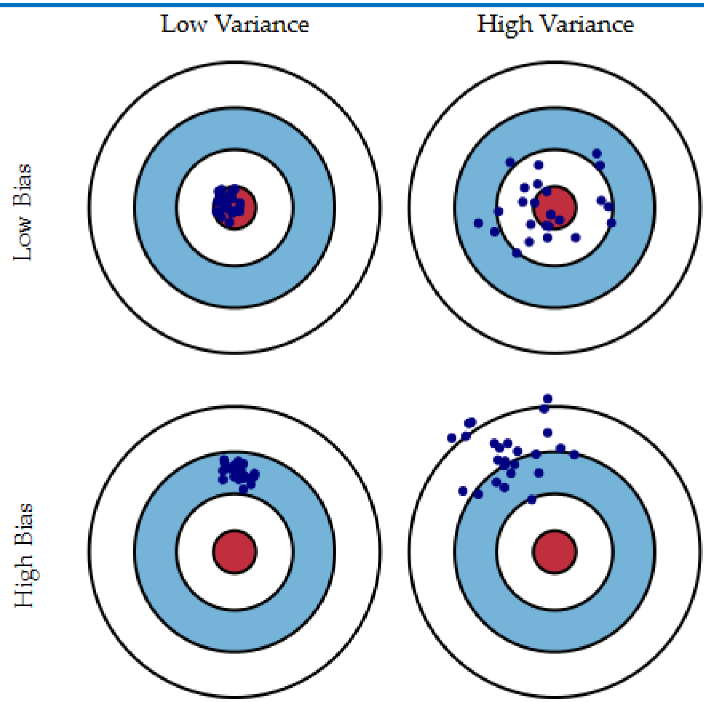
- For data set $\mathcal{D}$: $$\begin{aligned}&\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\ &= \{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]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

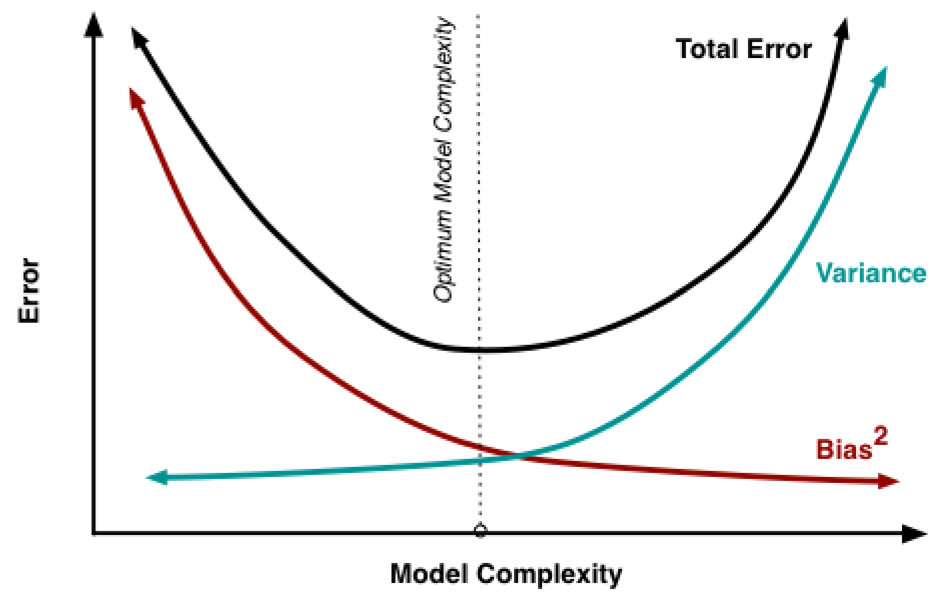
Prediction function

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

The Bias-Variance Trade-off







Bayesian linear regression

Parameter distribution

- Bayesian treatment of linear regression: (*β as a known constant*)

$$p(t|\mathbf{x},\mathbf{w},\beta) = \mathcal{N}(t|\,\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}),\beta^{-1}) \quad + \quad p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0,\mathbf{S}_0)$$

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

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

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

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

$$\boxed{p(\mathbf{w}|\alpha) = \left[\frac{q}{2}\left(\frac{\alpha}{2}\right)^{1/q}\frac{1}{\Gamma(1/q)}\right]^M \exp\left(-\frac{\alpha}{2}\sum_{j=1}^{M}|w_j|^q\right)}$$

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

Bayesian inference of parameter distribution

- True parameter values: (w0, w1) = (-0.3, 0.5), set $\beta = (1/0.2)^2 = 25$, α=2.0

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

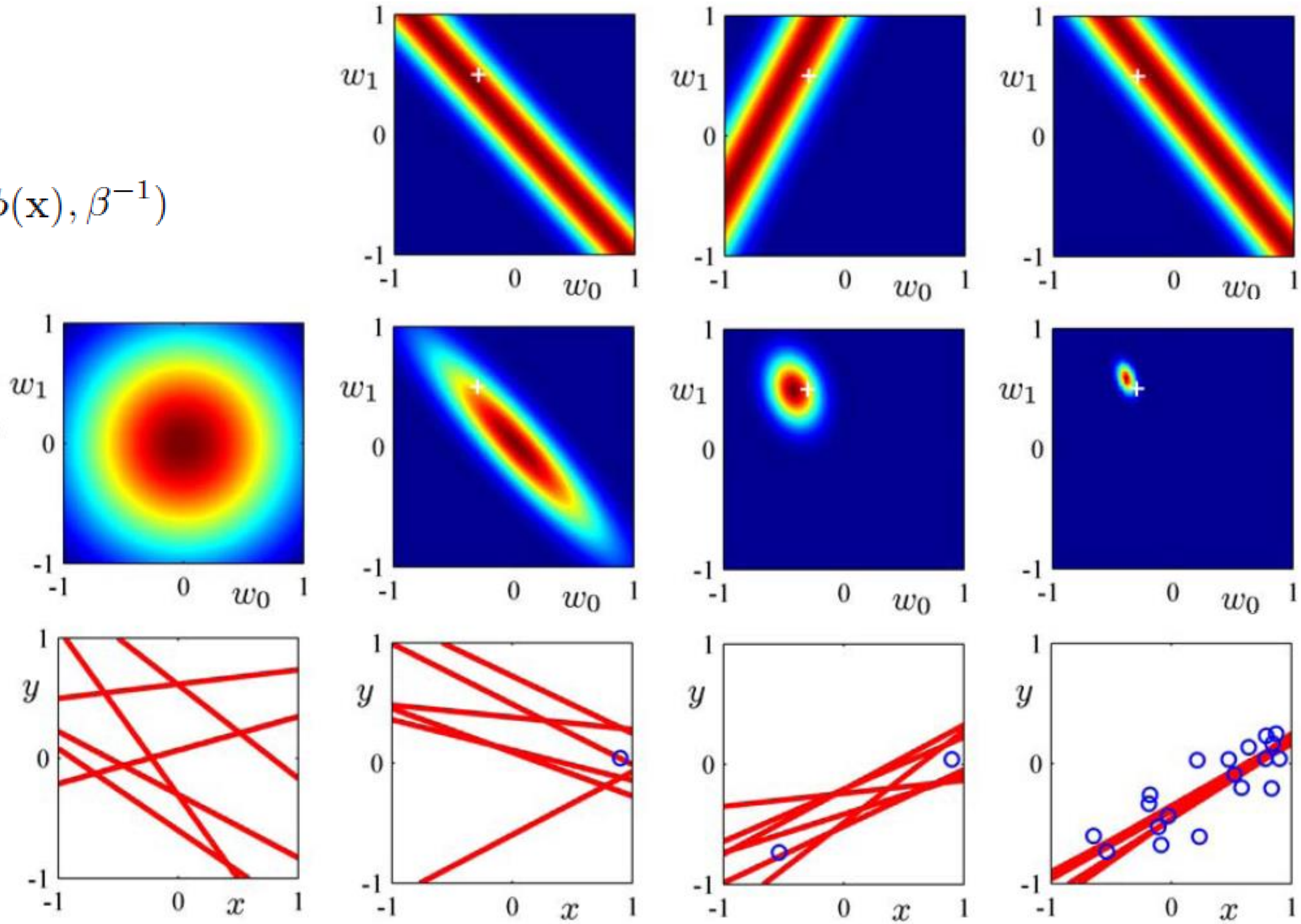
likelihood

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

prior/posterior

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

data space



Predictive distribution

- Definition:

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

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

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

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

$$\Rightarrow \quad p(t|\mathbf{x},\mathbf{t},\alpha,\beta) = \int \mathcal{N}(t|\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{w},\beta^{-1})\mathcal{N}(\mathbf{w}|\mathbf{m}_N,\mathbf{S}_N)\,\mathrm{d}\mathbf{w}$$

$$\boxed{\begin{aligned}p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\mathbf{\Lambda}^{-1})\\ p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b},\mathbf{L}^{-1})\\ p(\mathbf{y}) &= \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu}+\mathbf{b},\mathbf{L}^{-1}+\mathbf{A}\mathbf{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}})\end{aligned}} \quad \Rightarrow \quad \boxed{\begin{aligned}p(t|\mathbf{x},\mathbf{t},\alpha,\beta) &= \mathcal{N}(t|\mathbf{m}_N^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}),\sigma_N^2(\mathbf{x}))\\ \sigma_N^2(\mathbf{x}) &= \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x})\end{aligned}}$$

$$\sigma_{N+1}^2(\mathbf{x}) \leqslant \sigma_N^2(\mathbf{x}) \qquad N\to\infty, \quad \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}) \to \text{zero}$$

Predictive distribution: Examples

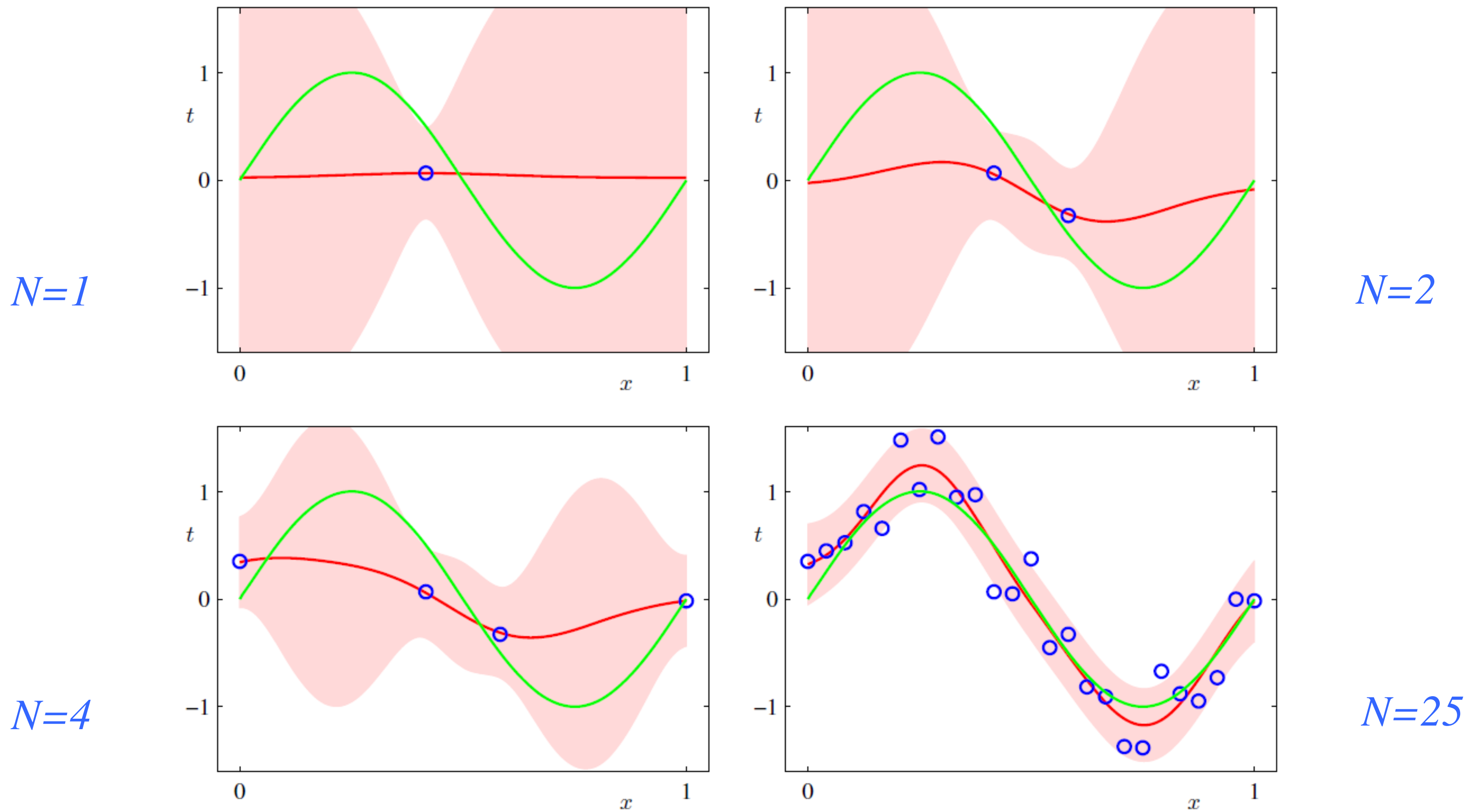
- *A model consisting of 9 'Gaussian' basis functions*

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

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

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

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



Predictive distribution: Examples

- *A model consisting of 9 'Gaussian' basis functions*

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

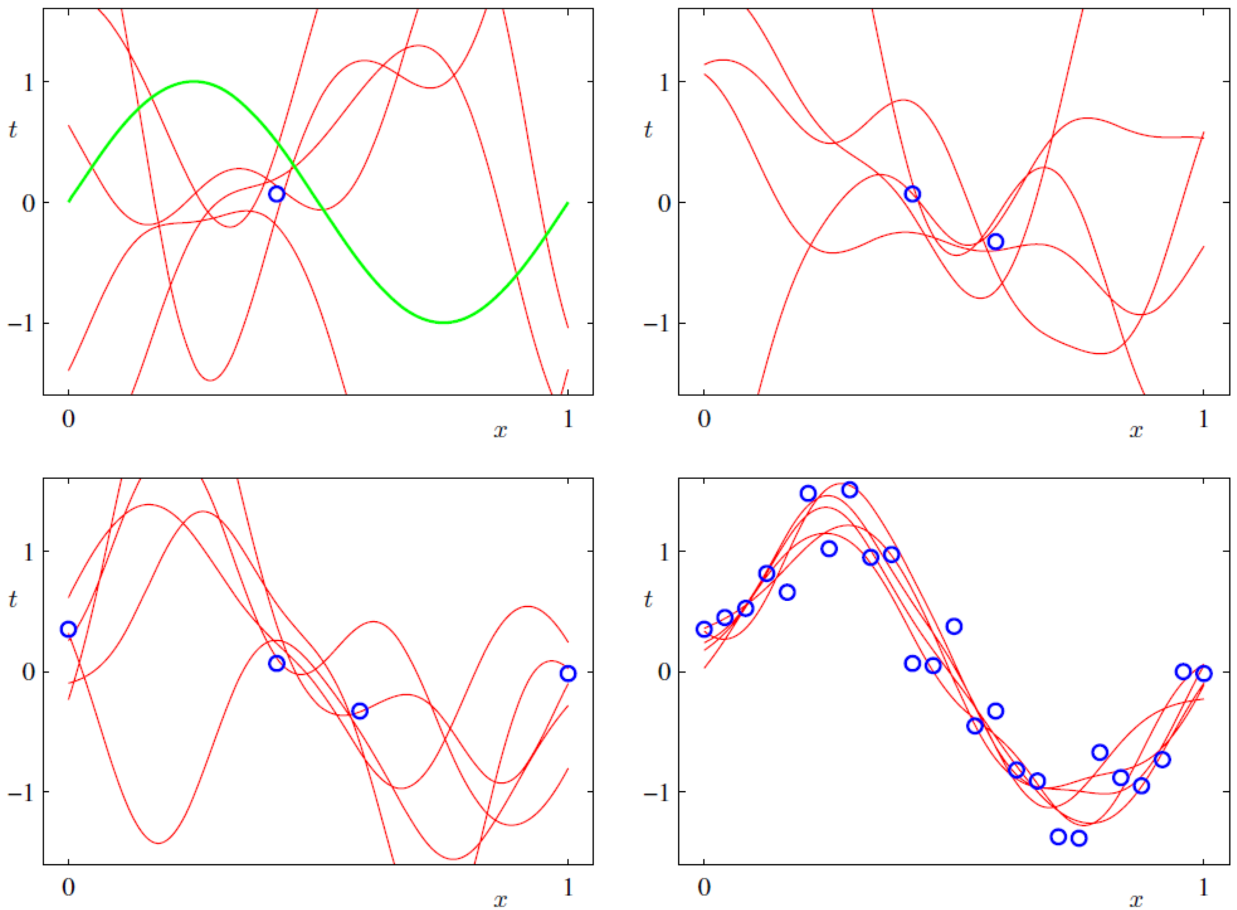
$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$

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

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

N=1

N=2



N=4

N=25

Next: Linear Models for Classification

- HW3:
  - 3.6, 3.7, 3.12, 3.13
  - Repair a damaged image by using the regression method:
    - see website for details.