Deep Learning: Theory and Practice

Linear and Logistic Models for Classification

01-02-2018

deeplearning.cce2018@gmail.com

Established 1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

Matrix Derivatives

- Derivative of a scalar with a vector

$$\left(\frac{\partial x}{\partial \mathbf{a}}\right)_i = \frac{\partial x}{\partial a_i}$$

- Derivative of a vector with a scalar

$$\left(\frac{\partial \mathbf{a}}{\partial x}\right)_i = \frac{\partial a_i}{\partial x}$$

- Derivative of a vector with a vector

$$\left(\frac{\partial \mathbf{a}}{\partial \mathbf{b}}\right)_{ij} = \frac{\partial a_i}{\partial b_j}.$$

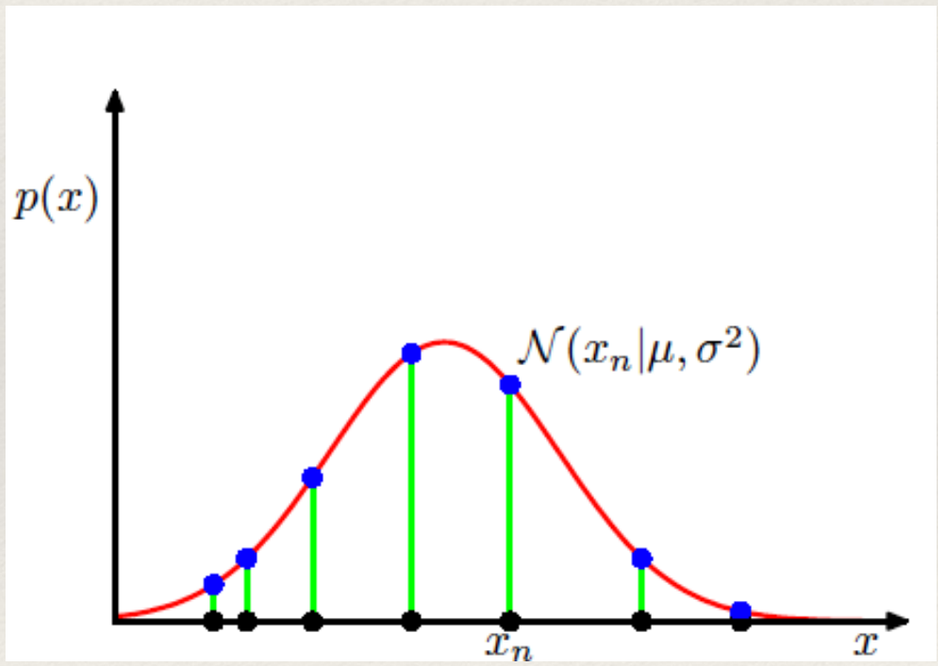
- Derivative of trace of a matrix ?

Maximum Likelihood

- Gaussian Distribution - multivariate

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

$$p(\mathbf{x}|\mu,\sigma^2) = \prod_{n=1}^{N}\mathcal{N}\left(x_n|\mu,\sigma^2\right)$$



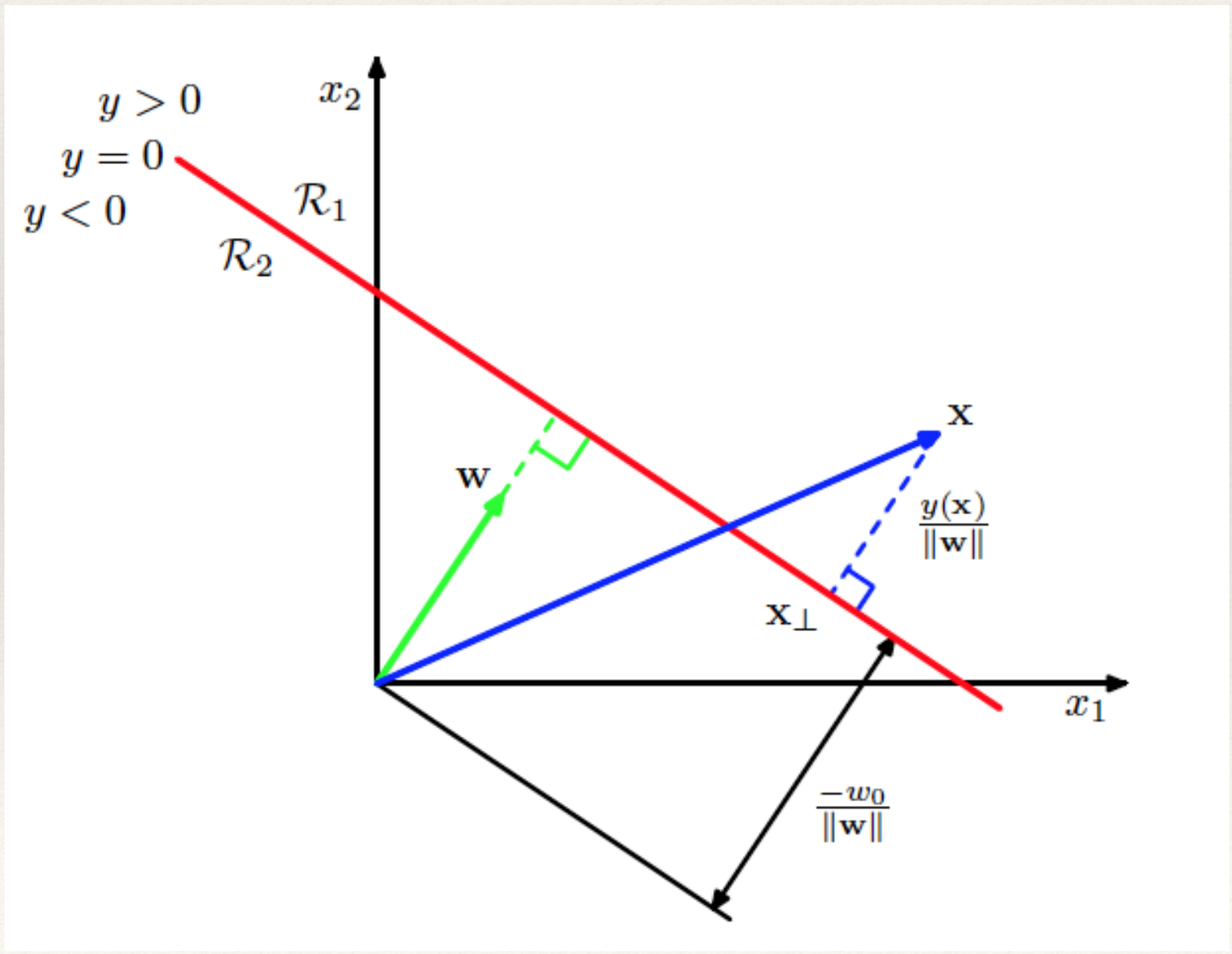
$$\ln p\left(\mathbf{x}|\mu,\sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}x_n$$

Linear Models for Classification

- Optimize a modified cost function

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$



Least Squares for Classification

- K-class classification problem

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}}$$

- With 1-of-K hot encoding, and least squares regression

$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\}$$

Bishop - PRML book (Chap 3)

Logistic Regression

- 2- class logistic regression

$$p(\mathcal{C}_1|\phi) = y(\phi) = \sigma\left(\mathbf{w}^{\mathrm{T}}\phi\right)$$

- Maximum likelihood solution

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N}(y_n - t_n)\phi_n$$

- K-class logistic regression

$$p(\mathcal{C}_k|\phi) = y_k(\phi) = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

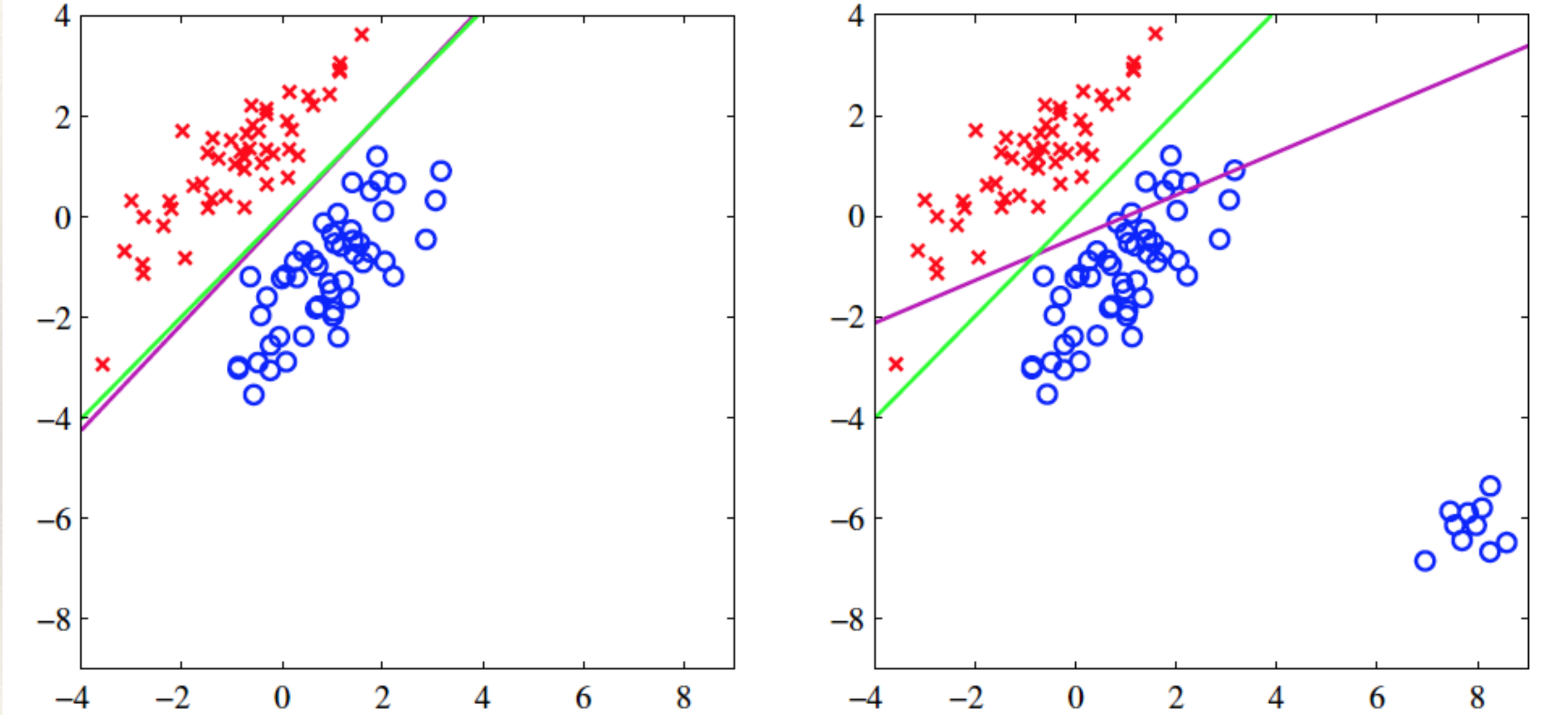
$$a_k = \mathbf{w}_k^{\mathrm{T}}\phi.$$

- Maximum likelihood solution

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj})\,\phi_n$$

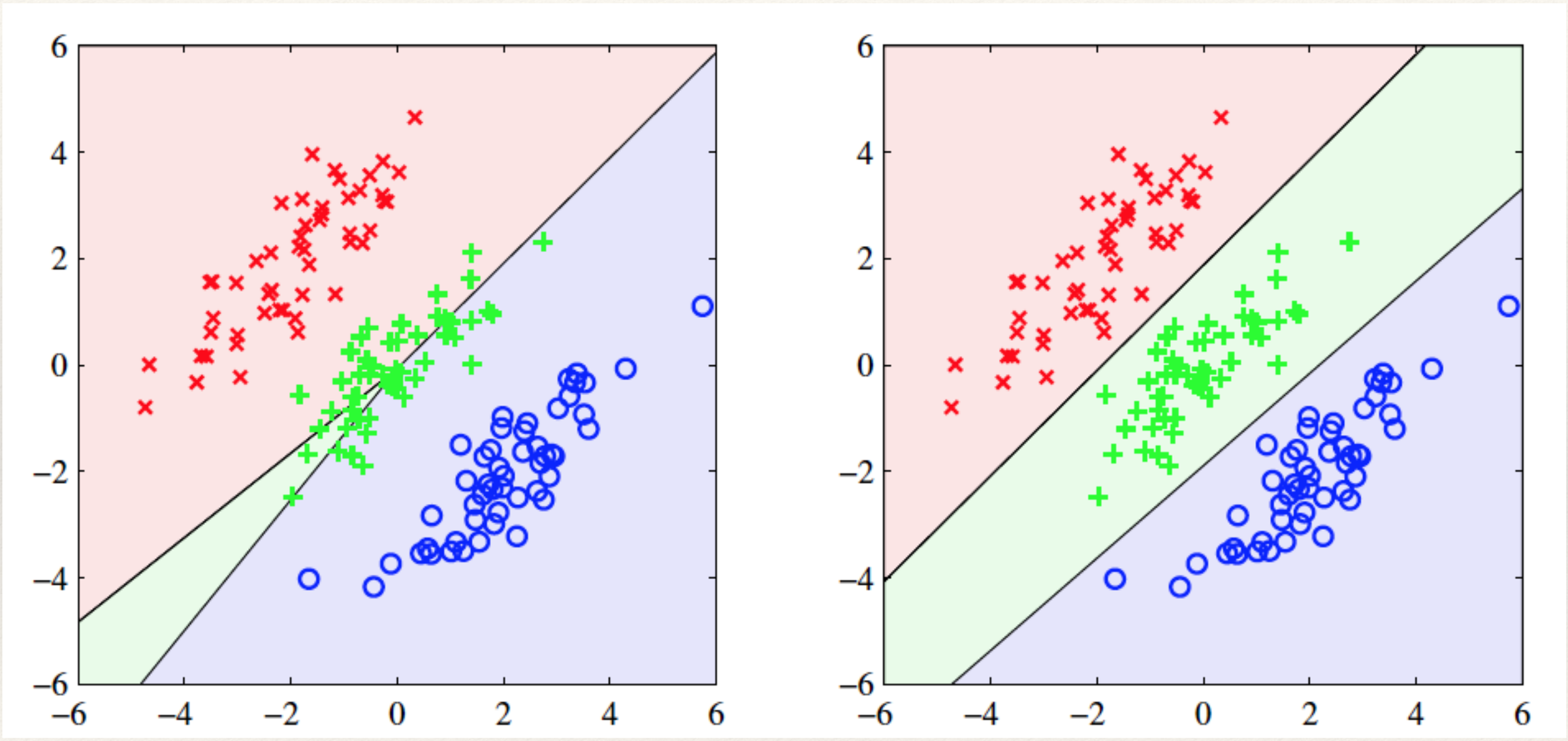
Bishop - PRML book (Chap 3)

Least Squares versus Logistic Regression



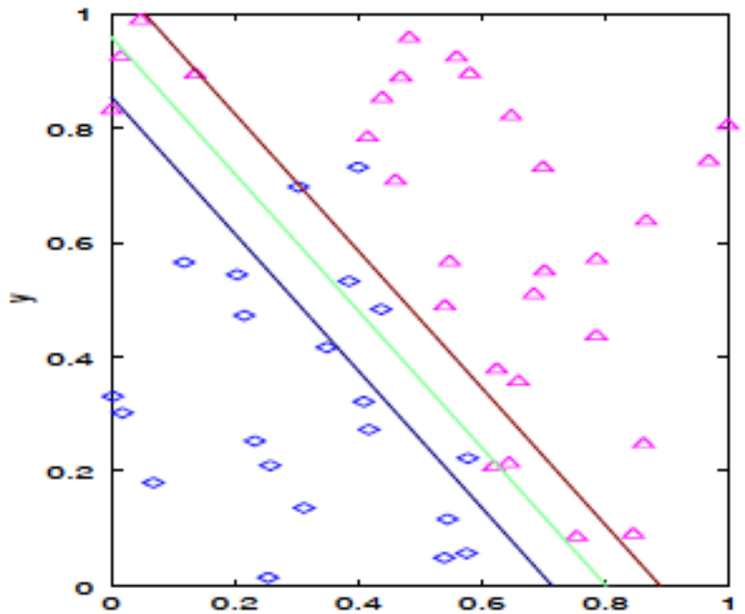
Bishop - PRML book (Chap 4)

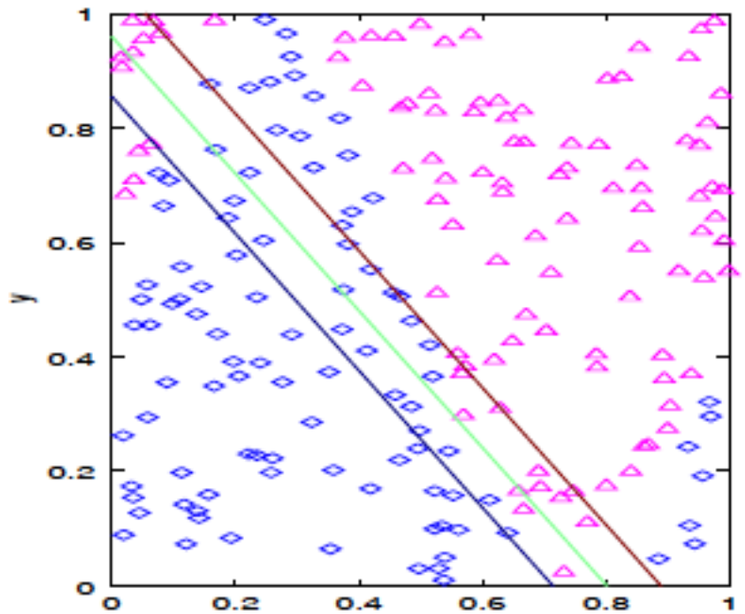
Least Squares versus Logistic Regression



Bishop - PRML book (Chap 4)

Underfit

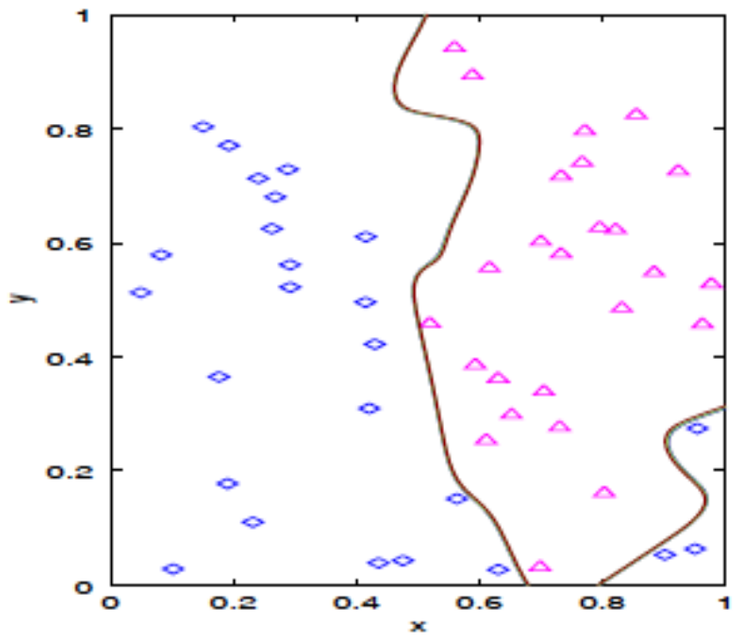


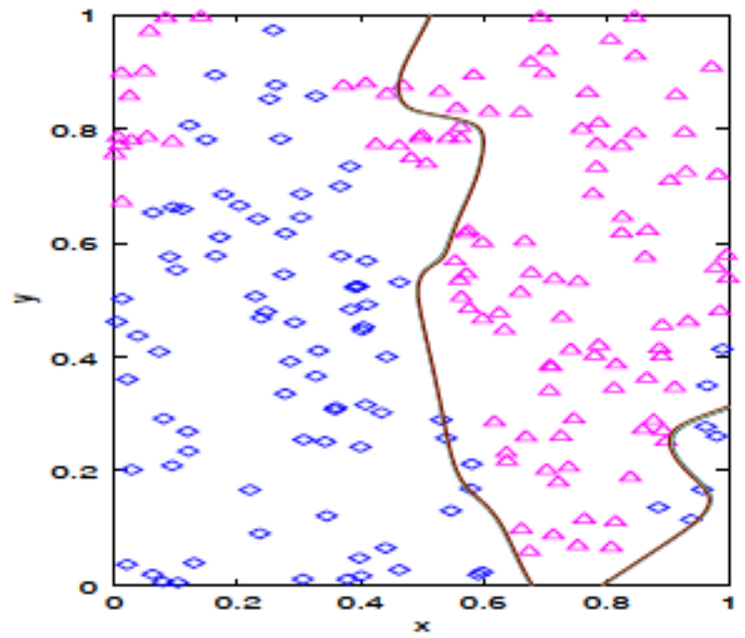


- The model is not able to capture the variability in the data (Linear Model)
- Both the training and testing error are high (15%,20%)
- Try to learn a more complex model – more features, more hidden neurons, decrease regularization
- More data would not help

(Courtesy - Dr D. Vijayasenan, NITK)

Overfit

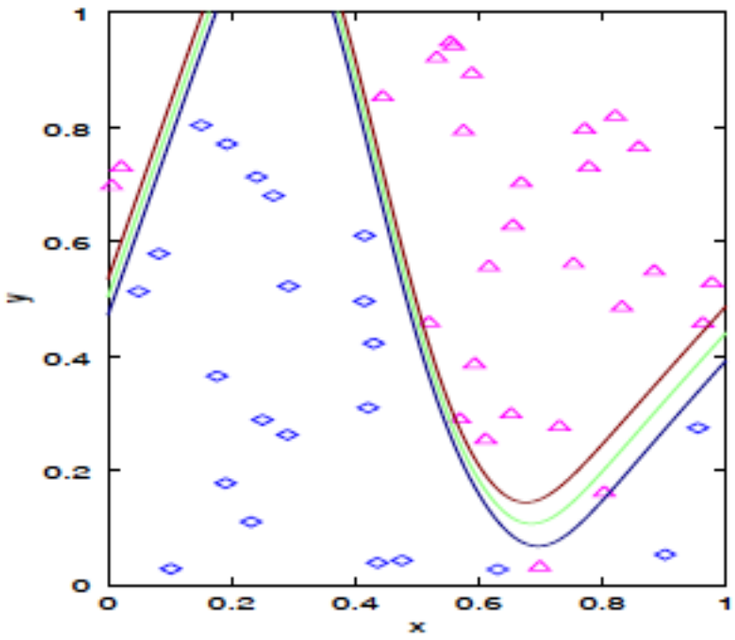


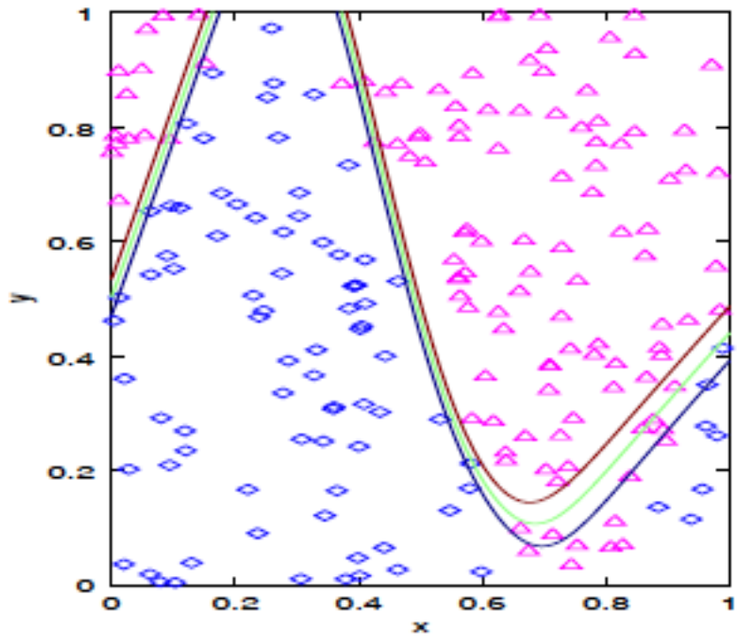


- The model is capturing data as well as accidental variations (100 hidden neurons)
- Training error is too low and testing error is too high (0%, and 16%)
- Try to learn a simpler model – less features, less hidden neurons, increase regularization
- More data would help

(Courtesy - Dr D. Vijayasenan, NITK)

Compromise





- Reasonable training and test errors – $(4\%, 8\%)$
- Appropriate model – capturing only the global characteristics not details

(Courtesy - Dr D. Vijayasenan, NITK)