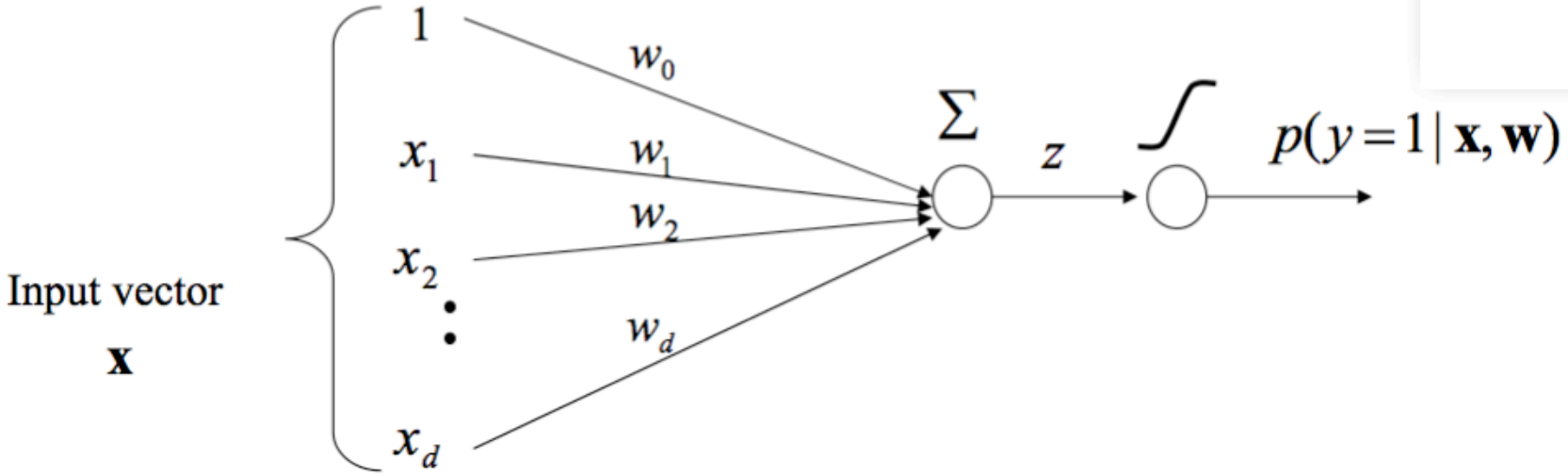
Logistic Regression

- **Discriminant functions:**

$$g_1(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x}) \qquad g_0(\mathbf{x}) = 1 - g(\mathbf{w}^T\mathbf{x})$$

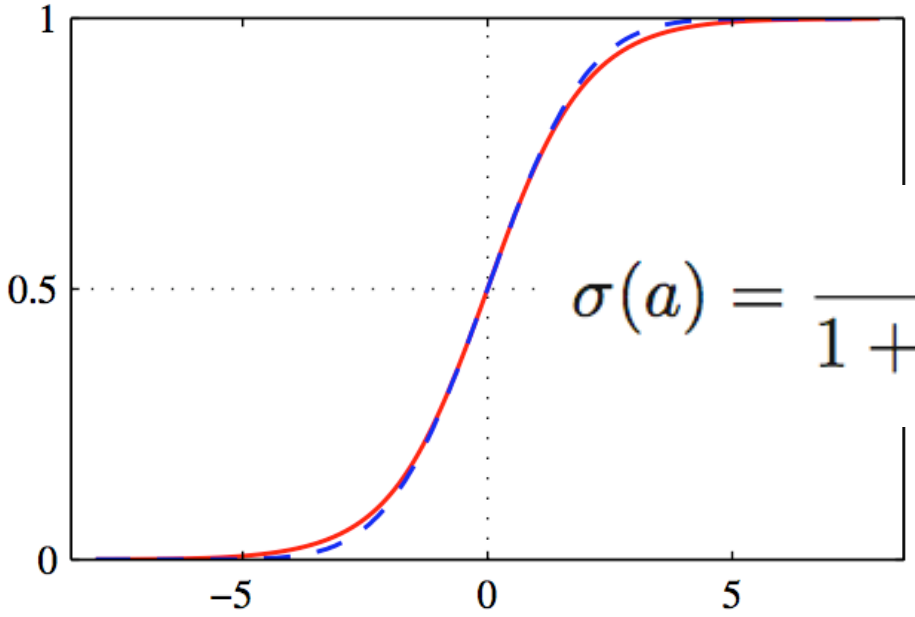
- **Where** $g(z) = 1/(1+e^{-z})$ - is a logistic function
- **Values of discriminant functions vary in [0,1]**
  - **Probabilistic interpretation**

$$f(\mathbf{x},\mathbf{w}) = p(y=1 \mid \mathbf{w},\mathbf{x}) = g_1(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x})$$



Logistic Trick

Plot of the logistic sigmoid function $\sigma(a)$ defined by (4.59), shown in red, together with the scaled probit function $\Phi(\lambda a)$, for $\lambda^2 = \pi/8$, shown in dashed blue, where $\Phi(a)$ is defined by (4.114). The scaling factor $\pi/8$ is chosen so that the derivatives of the two curves are equal for $a = 0$.



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

Posterior Probability For Two Classes

$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{1}{1 + \exp(-a)} = \sigma(a) \end{aligned}$$

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$

Logistic Regression

- Discriminative learning of Posterior probabilities
  - we learn **a probabilistic function**

    f: $X \rightarrow [0,1]$

  where *f is* the probability of class 1 given **x:**

$$f(\mathbf{x}, \mathbf{w}) = p(y = 1 \mid \mathbf{x}, \mathbf{w})$$

Two class case gives the decision rule:

If $p(y = 1 \mid \mathbf{x}) \geq 1/2$ then choose **1**
Else choose **0**

Logistic decision boundary

- Logistic regression model defines a **linear decision boundary**
- **Why?**
- **Answer:** Compare two **discriminant functions**.
- **Decision boundary:** $g_1(\mathbf{x}) = g_0(\mathbf{x})$
- For the boundary it must hold:

$$\log \frac{g_o(\mathbf{x})}{g_1(\mathbf{x})} = \log \frac{1 - g(\mathbf{w}^T\mathbf{x})}{g(\mathbf{w}^T\mathbf{x})} = 0$$

$$\log \frac{g_o(\mathbf{x})}{g_1(\mathbf{x})} = \log \frac{\frac{\exp -(\mathbf{w}^T\mathbf{x})}{1+\exp -(\mathbf{w}^T\mathbf{x})}}{\frac{1}{1+\exp -(\mathbf{w}^T\mathbf{x})}} = \log \exp -(\mathbf{w}^T\mathbf{x}) = \mathbf{w}^T\mathbf{x} = 0$$

Likelihood of data

Likelihood of data

- **Let**

$$D_i =< \mathbf{x}_i, y_i > \quad \mu_i = p(y_i = 1 \mid \mathbf{x}_i, \mathbf{w}) = g(z_i) = g(\mathbf{w}^T \mathbf{x})$$

- **Then**

$$L(D, \mathbf{w}) = \prod_{i=1}^{n} P(y = y_i \mid \mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{n} \mu_i^{\,y_i} (1 - \mu_i)^{1 - y_i}$$

- **Find weights w that maximize the likelihood of outputs**
  - Apply the log-likelihood trick The optimal weights are the same for both the likelihood and the log-likelihood

$$l(D, \mathbf{w}) = \log \prod_{i=1}^{n} \mu_i^{\,y_i} (1 - \mu_i)^{1 - y_i} = \sum_{i=1}^{n} \log \mu_i^{\,y_i} (1 - \mu_i)^{1 - y_i} =$$

$$= \sum_{i=1}^{n} y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i)$$

Learning

- **Log likelihood**

$$l(D,\mathbf{w}) = \sum_{i=1}^{n} y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i)$$

$$l(D,\mathbf{w}) = \sum_{i=1}^{n} y_i \log f(\mathbf{x}_i,\mathbf{w}) + (1 - y_i) \log\left(1 - f(\mathbf{x}_i,\mathbf{w})\right)$$

- **Gradient**

$$\nabla_{\mathbf{w}} l(D,\mathbf{w}) = -\sum_{i=1}^{n} \mathbf{x}_i \left(y_i - f(\mathbf{x}_i,\mathbf{w})\right)$$

- **Gradient descent:**

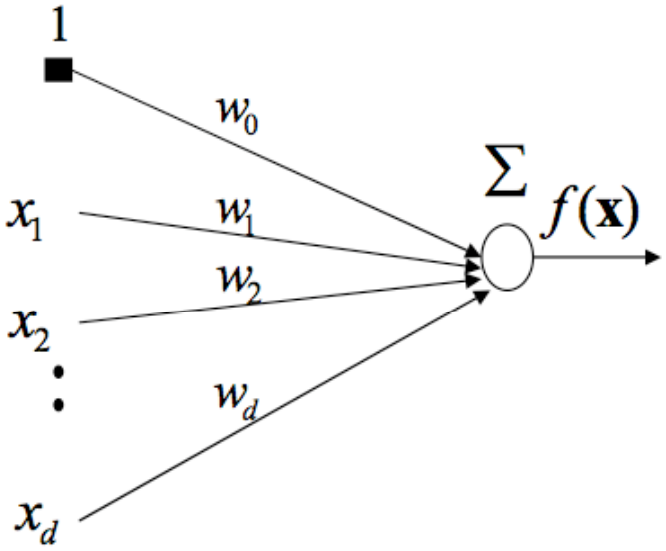
$$\mathbf{w}^{(k)} \leftarrow \mathbf{w}^{(k-1)} - \alpha(k) \nabla_{\mathbf{w}} [-l(D,\mathbf{w})] \big|_{\mathbf{w}^{(k-1)}}$$

$$\mathbf{w}^{(k)} \leftarrow \mathbf{w}^{(k-1)} + \alpha(k) \sum_{i=1}^{n} [y_i - f(\mathbf{w}^{(k-1)}, \mathbf{x}_i)] \mathbf{x}_i$$

Linear Regression

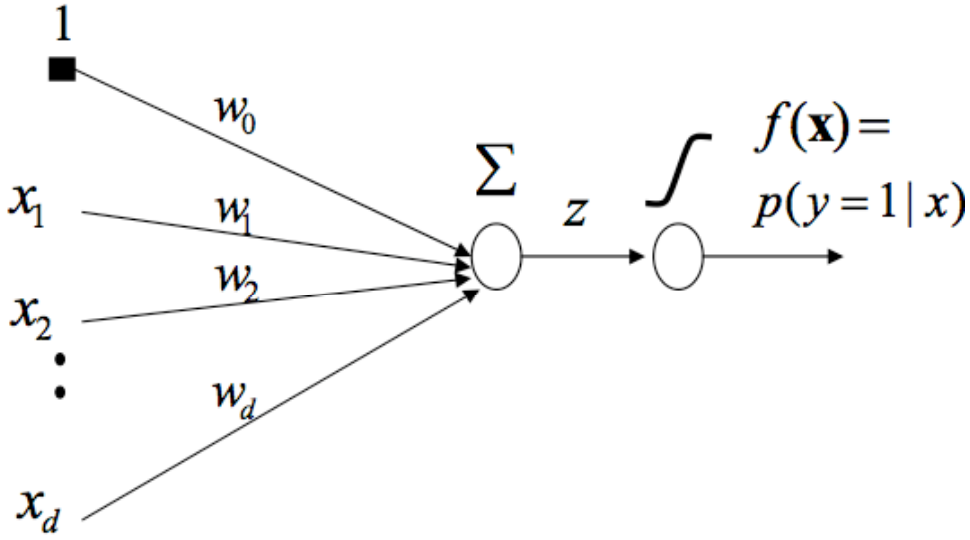
Linear regression

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$



Logistic regression

$$f(\mathbf{x}) = p(y=1 \mid \mathbf{x}, \mathbf{w}) = g(\mathbf{w}^T \mathbf{x})$$



Gradient update:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))\mathbf{x}_i$$

Online: $\mathbf{w} \leftarrow \mathbf{w} + \alpha (y - f(\mathbf{x}))\mathbf{x}$

The same

Gradient update:

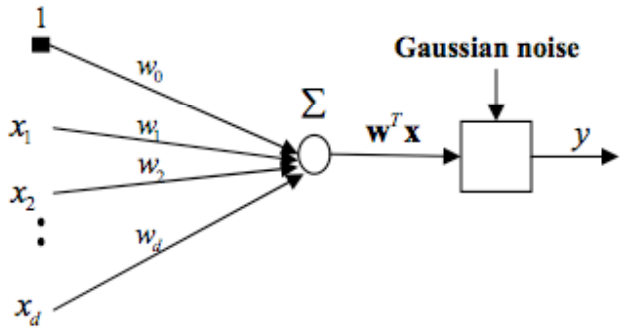
$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))\mathbf{x}_i$$

Online: $\mathbf{w} \leftarrow \mathbf{w} + \alpha (y - f(\mathbf{x}))\mathbf{x}$

Simple Gradient

- The **same simple gradient update rule** derived for both the linear and logistic regression models
- Where the magic comes from?
- Under the **log-likelihood** measure the function models and the models for the output selection fit together:
  - **Linear model + Gaussian noise**

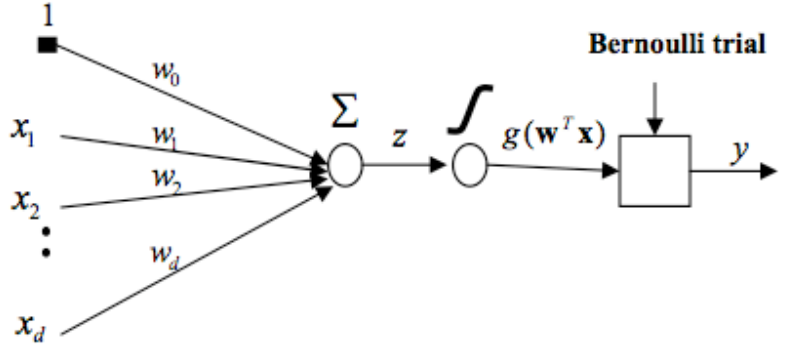
$$y = \mathbf{w}^T \mathbf{x} + \varepsilon \qquad \varepsilon \sim N(0, \sigma^2)$$



  - **Logistic + Bernoulli**

$$y = \text{Bernoulli}(\theta)$$

$$\theta = p(y = 1 \mid \mathbf{x}) = g(\mathbf{w}^T \mathbf{x})$$



An algorithm

Online-logistic-regression (*D, number of iterations*)

initialize weights $\mathbf{w} = (w_0, w_1, w_2 \ldots w_d)$

for i=1:1: *number of iterations*

do **select** a data point $D_i =< \mathbf{x}_i, y_i >$ from D

set $\alpha = 1/i$

update weights (in parallel)

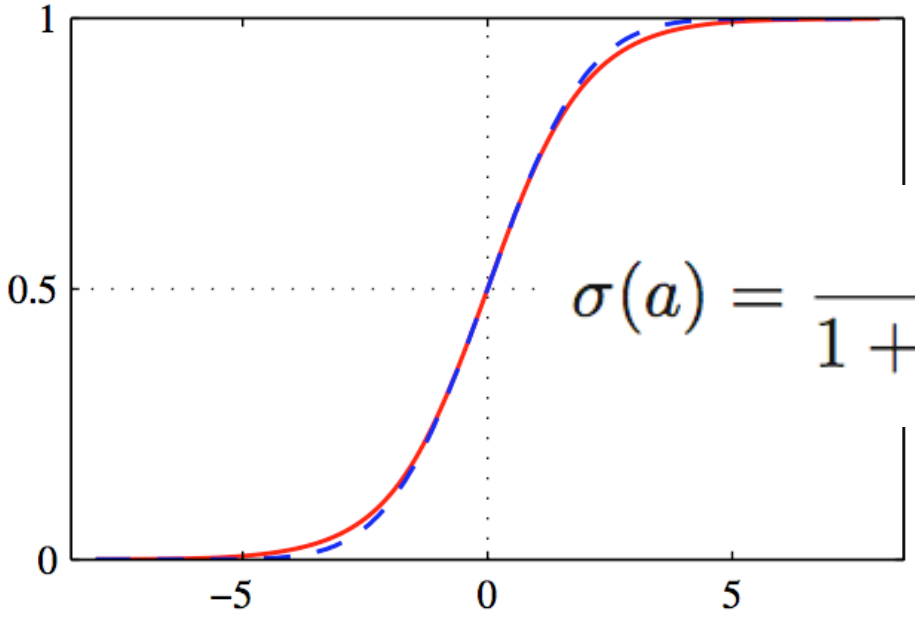
$$\mathbf{w} \leftarrow \mathbf{w} + \alpha(i)[y_i - f(\mathbf{w}, \mathbf{x}_i)]\mathbf{x}_i$$

end for

return weights **w**

Logistic Trick

Plot of the logistic sigmoid function $\sigma(a)$ defined by (4.59), shown in red, together with the scaled probit function $\Phi(\lambda a)$, for $\lambda^2 = \pi/8$, shown in dashed blue, where $\Phi(a)$ is defined by (4.114). The scaling factor $\pi/8$ is chosen so that the derivatives of the two curves are equal for $a = 0$.



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

Posterior Probability For Two Classes

$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{1}{1 + \exp(-a)} = \sigma(a) \end{aligned}$$

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$

MultiClass

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

$$= \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

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

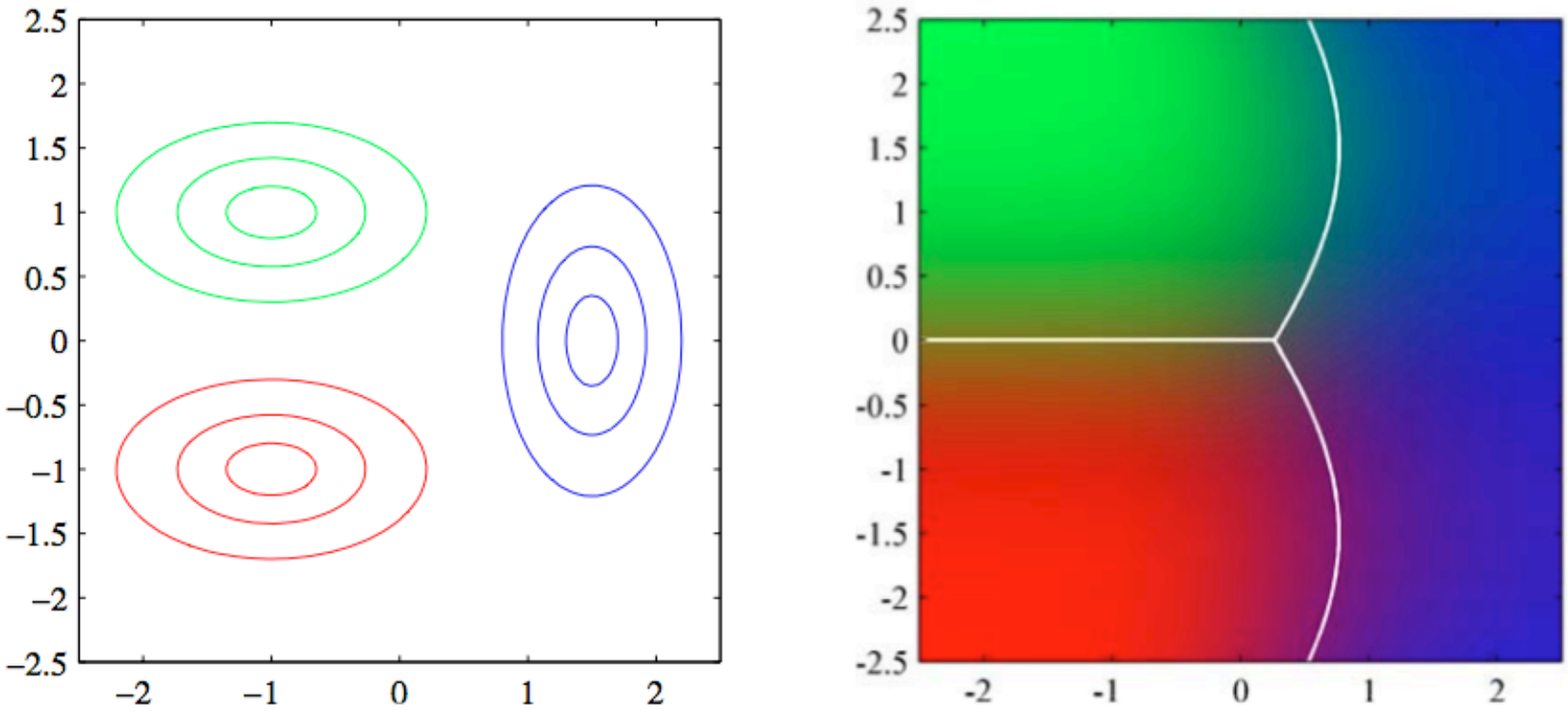


Figure 4.11 The left-hand plot shows the class-conditional densities for three classes each having a Gaussian distribution, coloured red, green, and blue, in which the red and green classes have the same covariance matrix. The right-hand plot shows the corresponding posterior probabilities, in which the RGB colour vector represents the posterior probabilities for the respective three classes. The decision boundaries are also shown. Notice that the boundary between the red and green classes, which have the same covariance matrix, is linear, whereas those between the other pairs of classes are quadratic.

Example: Gaussians

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

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

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

$$\mathbf{w}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$$

$$w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(\mathcal{C}_k)$$

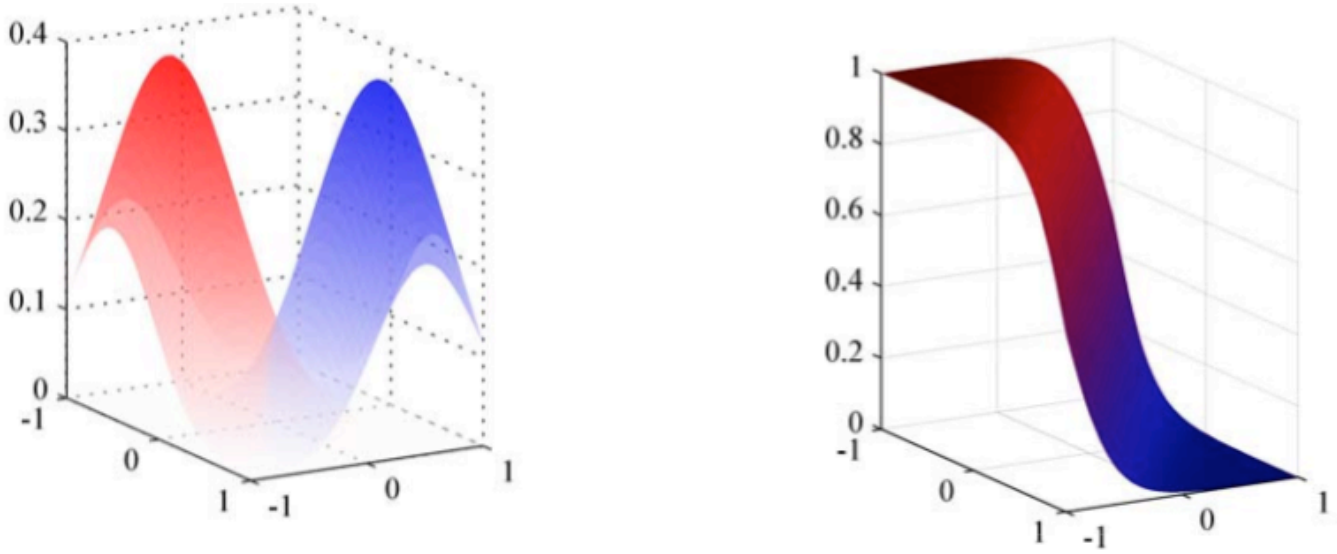
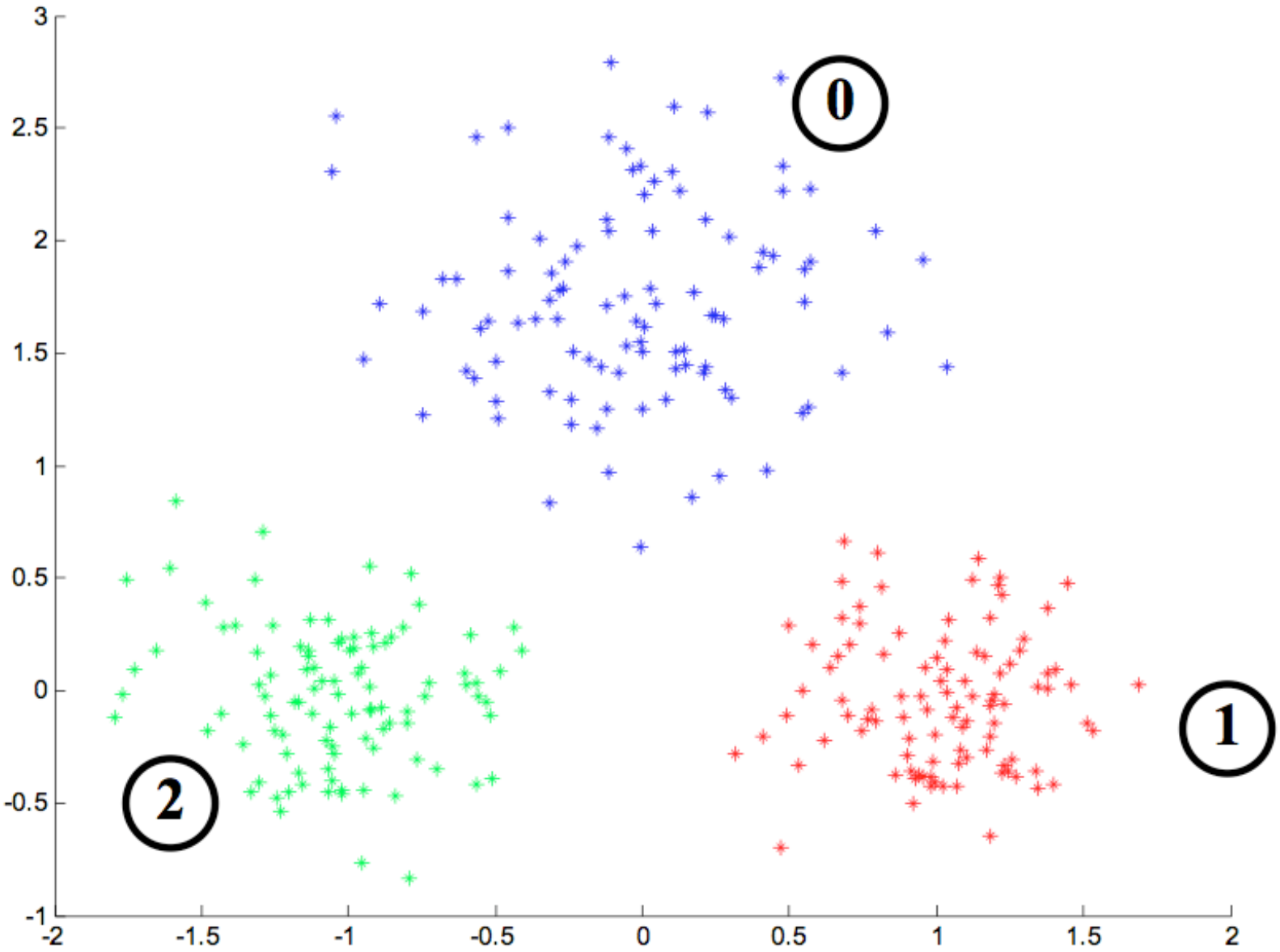
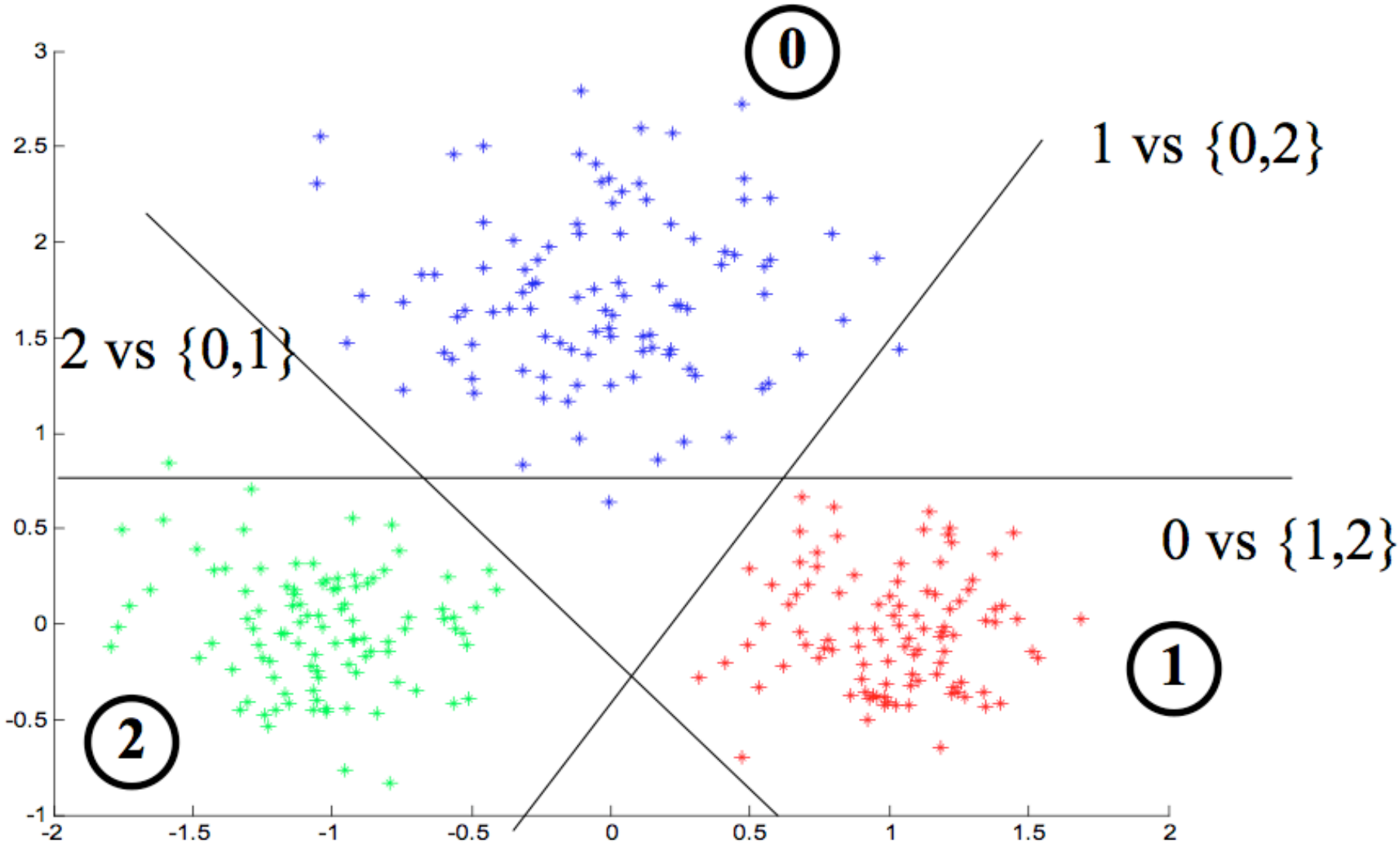


Figure 4.10 The left-hand plot shows the class-conditional densities for two classes, denoted red and blue. On the right is the corresponding posterior probability $p(\mathcal{C}_1|\mathbf{x})$, which is given by a logistic sigmoid of a linear function of $\mathbf{x}$. The surface in the right-hand plot is coloured using a proportion of red ink given by $p(\mathcal{C}_1|\mathbf{x})$ and a proportion of blue ink given by $p(\mathcal{C}_2|\mathbf{x}) = 1 - p(\mathcal{C}_1|\mathbf{x})$.

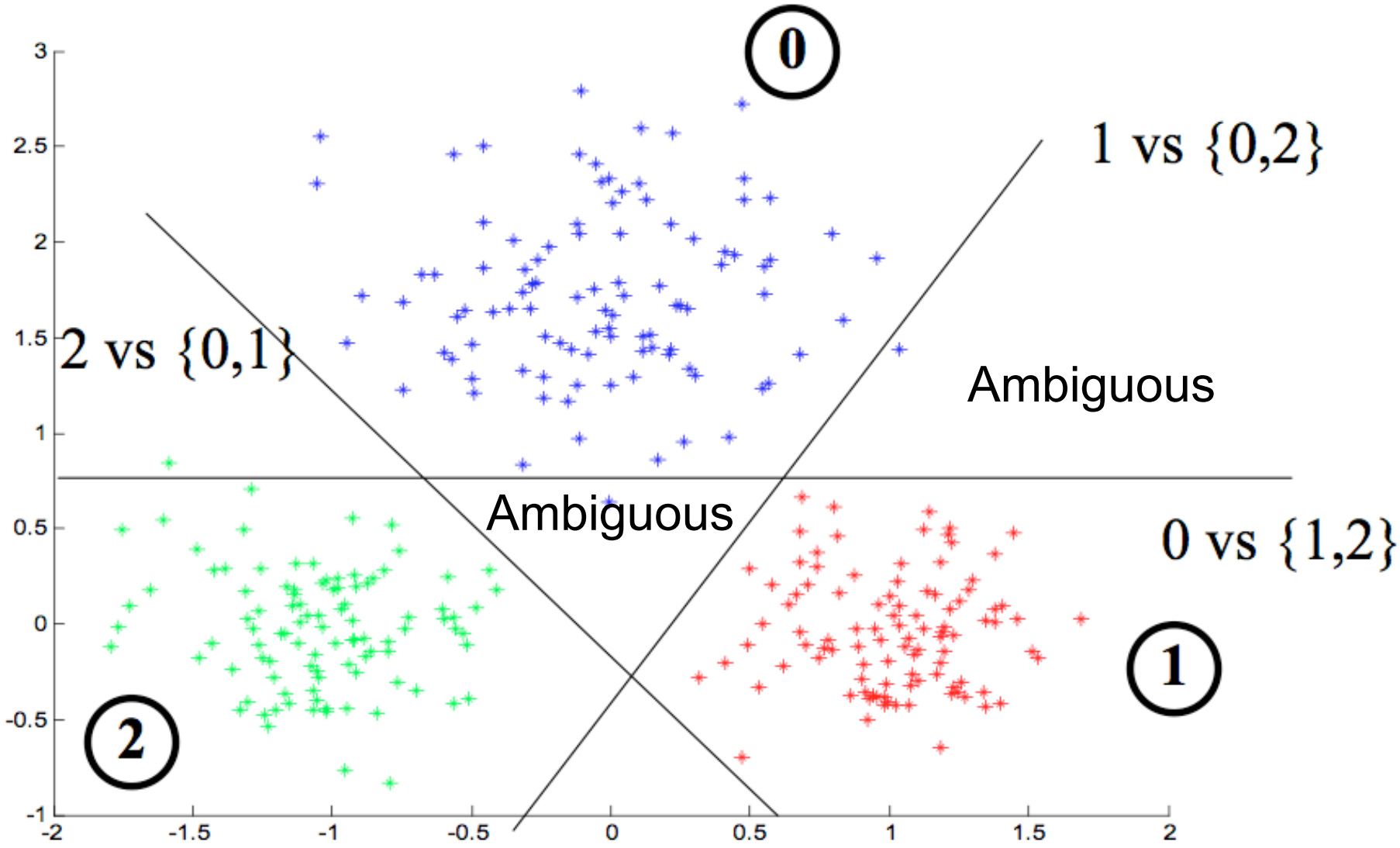
Multiway Classification



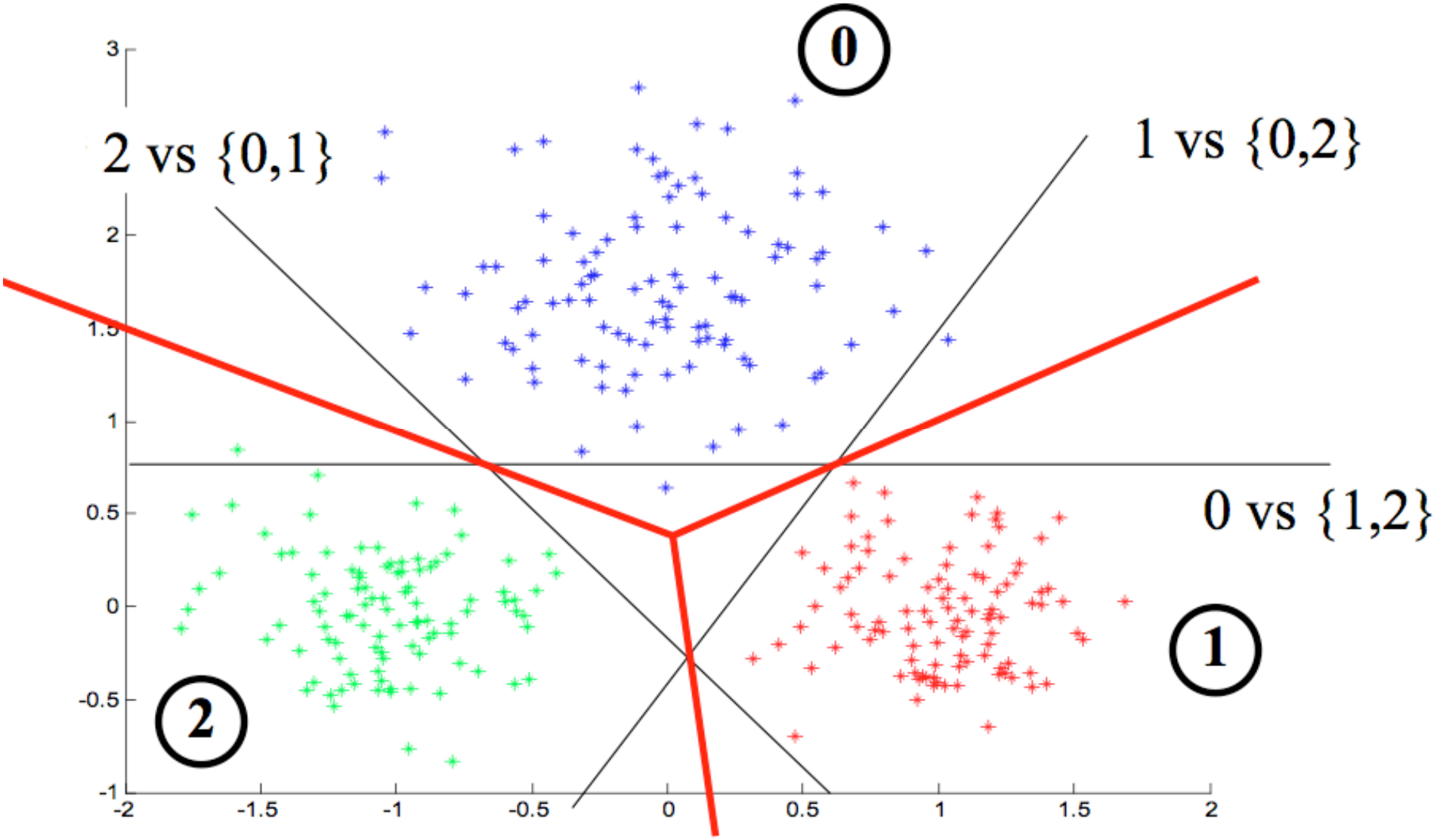
One approach



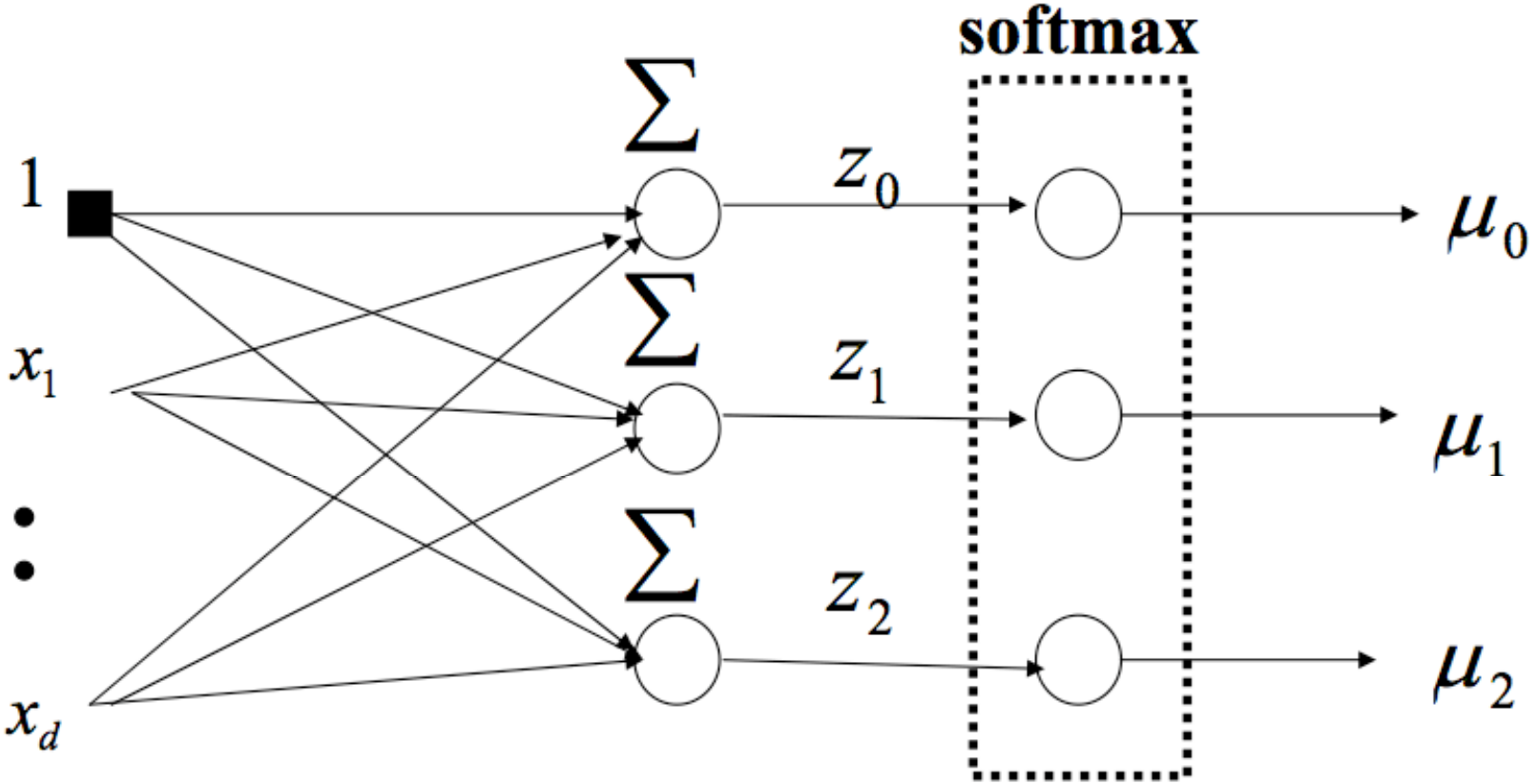
One approach



Correct Approach



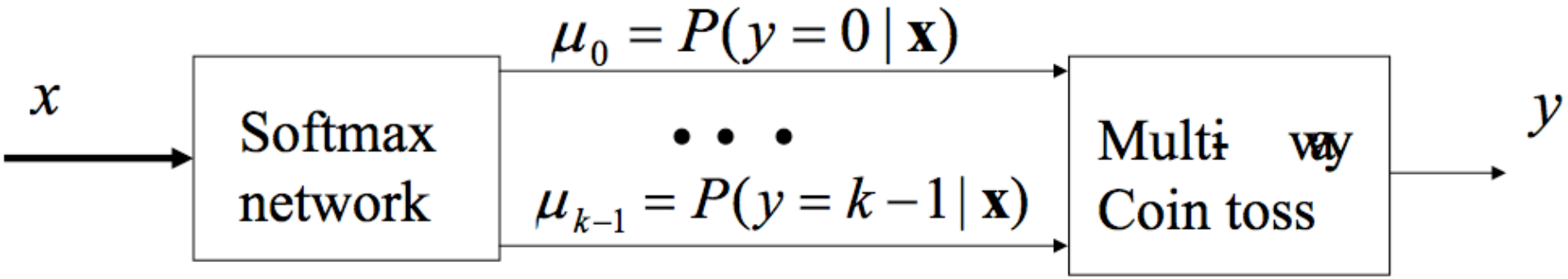
Softmax



$$p(y = i \mid \mathbf{x}) = \mu_i = \frac{\exp(\mathbf{w}_i^T \mathbf{x})}{\sum_i \exp(\mathbf{w}_i^T \mathbf{x})} \qquad \sum_i \mu_i = 1$$

Learning a Softmax Network

- Learning of parameters **w**: statistical view



Assume outputs y are transformed as follows

$$y \in \{0 \quad 1 \quad .. \quad k-1\} \quad \Rightarrow \quad y \in \left\{ \begin{pmatrix} 1 \\ 0 \\ .. \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ .. \\ 0 \end{pmatrix} .. \begin{pmatrix} 0 \\ 0 \\ .. \\ 1 \end{pmatrix} \right\}$$

Learning multi-class

- Learning of the parameters **w**: statistical view
- **Likelihood of outputs**

$$L(D,\mathbf{w}) = p(\mathbf{Y} \mid \mathbf{X}, w) = \prod_{i=1,..n} p(y_i \mid \mathbf{x}_i, \mathbf{w})$$

- We want parameters **w** that maximize the likelihood
- **Log-likelihood trick**
  - Optimize log likelihood of outputs instead:

$$l(D,\mathbf{w}) = \log \prod_{i=1,..n} p(y_i \mid \mathbf{x},\mathbf{w}) = \sum_{i=1,..n} \log p(y_i \mid \mathbf{x},\mathbf{w})$$

$$= \sum_{i=1,..n} \sum_{q=0}^{k-1} \log \mu_i^{\,y_{i,q}} = \sum_{i=1,..n} \sum_{q=0}^{k-1} y_{i,q} \log \mu_{i,q}$$

- **Objective to optimize**

$$J(D_i, \mathbf{w}) = -\sum_{i=1}^{n} \sum_{q=0}^{k-1} y_{i,q} \log \mu_{i,q}$$

Learning multi-class

- **Error to optimize:**

$$J(D_i, \mathbf{w}) = -\sum_{i=1}^{n} \sum_{q=0}^{k-1} y_{i,q} \log \mu_{i,q}$$

- **Gradient**

$$\frac{\partial}{\partial w_{jk}} J(D_i, \mathbf{w}) = \sum_{i=1}^{n} -x_{i,j}(y_{i,j} - \mu_{i,j})$$

- The same very easy **gradient update** as used for the binary logistic regression

$$\mathbf{w}_j \leftarrow \mathbf{w}_j + \alpha \sum_{i=1}^{n} (y_{i,j} - \mu_{i,j})\mathbf{x}_i$$

- But now we have to update the weights of k networks

Exponential Family

$$f(\mathbf{x} \mid \boldsymbol{\theta}, \boldsymbol{\varphi}) = h(x, \boldsymbol{\varphi}) \exp\left\{ \frac{\boldsymbol{\theta}^T \mathbf{x} - A(\boldsymbol{\theta})}{a(\boldsymbol{\varphi})} \right\}$$

$\boldsymbol{\theta}$ - A location parameter $\boldsymbol{\varphi}$ - A scale parameter

Claim: A logistic regression is a correct model when class conditional densities are from the same distribution in the exponential family and have **the same scale factor** $\boldsymbol{\varphi}$

All but the linear part cancels in the posterior probabilities

Logistic and Exponential Family

- **Class conditional:**

$$p(\mathbf{x} \mid y = i) = h(\mathbf{x}, \boldsymbol{\varphi}) \exp\left\{ \frac{(\boldsymbol{\theta}_i^T \mathbf{x} - A(\boldsymbol{\theta}_i))}{a(\boldsymbol{\varphi})} \right\}$$

- **Class posterior:**

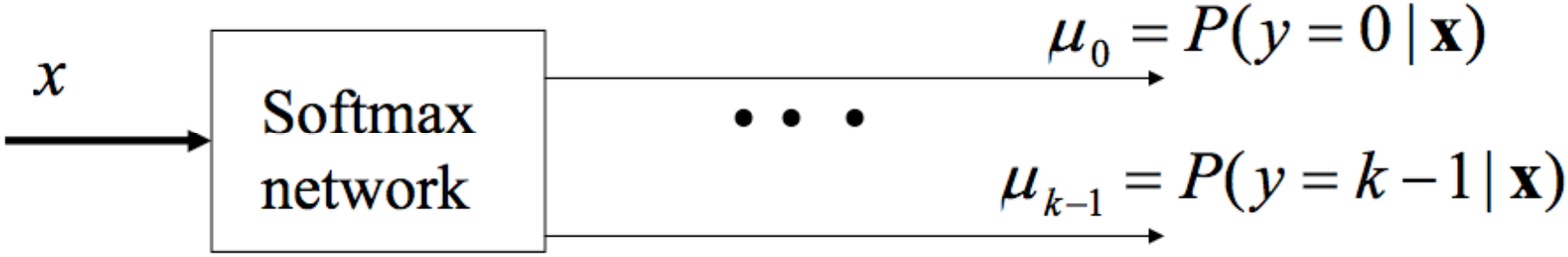
$$p(y = i \mid x) = \frac{p(\mathbf{x} \mid y = i) p(y = i)}{\sum_j p(\mathbf{x} \mid y = j) p(y = j)}$$

$$= \frac{h(\mathbf{x}, \boldsymbol{\varphi}) \exp\left\{ \frac{(\boldsymbol{\theta}_i^T \mathbf{x} - A(\boldsymbol{\theta}_i))}{a(\boldsymbol{\varphi})} \right\} p(y = i)}{\sum_j h(\mathbf{x}, \boldsymbol{\varphi}) \exp\left\{ \frac{(\boldsymbol{\theta}_j^T \mathbf{x} - A(\boldsymbol{\theta}_j))}{a(\boldsymbol{\varphi})} \right\} p(y = j)} = \frac{\exp(\mathbf{w}_i^T \mathbf{x} + b_i)}{\sum_j \exp(\mathbf{w}_j^T \mathbf{x} + b_j)}$$

$$\mathbf{w}_i = \frac{\boldsymbol{\theta}_i}{a(\boldsymbol{\varphi})} \qquad b_i = \frac{A(\boldsymbol{\theta}_i)}{a(\boldsymbol{\varphi})} + \ln p(y = i)$$

When is it right?

- **Softmax model is an accurate model** when class conditional densities are represented with densities from the exponential family with the same scaling parameter
- For **two classes** it reduces to the **logistic regression model**



$$p(\mathbf{x} \mid y = i) = \exp\left\{ \frac{(\boldsymbol{\theta}_i^T \mathbf{x} - b(\boldsymbol{\theta}_i))}{a(\boldsymbol{\varphi})} + c(\mathbf{x}, \boldsymbol{\varphi}) \right\}$$

$\boldsymbol{\theta}_i$ - location parameter for class conditional i

$\boldsymbol{\varphi}$ - scaling parameter (the same for all classes)

Bayesian Treatment

Need posterior on weights-

Combine likelihood with prior

Then approximate the predictive distribution:

$$p(c_\cdot|D,\mathbf{x}) = \int p(c_\cdot|\mathbf{w},\mathbf{x})\,p(\mathbf{w}|D)\,d\mathbf{w} \approx \int \frac{1}{1+\exp(\mathbf{w}^T\mathbf{x})}\,q(\mathbf{w})\,d\mathbf{w}$$