Introduction to Machine Learning

Linear Models for Classification

SHENG-JYH WANG

NATIONAL YANG MING CHIAO TUNG UNIVERSITY, TAIWAN

SPRING, 2024

Prerequisite Knowledge

Commonly Used Distributions

- Regression Problem
  - ✓ **Gaussian** Distribution
  - ✓ Conjugate Prior: **Gaussian** Distribution, **Wishart** Distribution, **Gaussian-Wishart** Distribution, **Gamma** Distribution
  - ✓ Related Distribution: **Student's t-distribution**
- Binary Classification Problem
  - ✓ **Bernoulli Distribution**, **Binomial Distribution**
  - ✓ Conjugate Prior: **Beta Distribution**
- Multi-class Classification Problem
  - ✓ **Multinomial Distribution**
  - ✓ Conjugate Prior: **Dirichlet Distribution**

Bernoulli Distribution (1/2)

$$p(x=1|\mu)=\mu \qquad \text{where } 0 \leqslant \mu \leqslant 1$$

$$\mathrm{Bern}(x|\mu)=\mu^x(1-\mu)^{1-x}$$

$$\begin{aligned}\mathbb{E}[x] &= \mu \\ \mathrm{var}[x] &= \mu(1-\mu)\end{aligned}$$

Example: Flipping a coin $x \in \{0, 1\}$

Bernoulli Distribution (2/2)

For a data set $D = \{x_1, \ldots, x_N\}$,

$$p(\mathcal{D}|\mu) = \prod_{n=1}^{N} p(x_n|\mu) = \prod_{n=1}^{N} \mu^{x_n}(1-\mu)^{1-x_n}$$

$$\ln p(\mathcal{D}|\mu) = \sum_{n=1}^{N} \ln p(x_n|\mu) = \sum_{n=1}^{N} \{x_n \ln \mu + (1-x_n)\ln(1-\mu)\}$$

The maximum likelihood (ML) estimator of μ is $\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n$

Binomial Distribution (1/2)

$$\mathrm{Bin}(m|N,\mu) = \binom{N}{m} \mu^m (1-\mu)^{N-m}$$

where

$$\binom{N}{m} \equiv \frac{N!}{(N-m)!m!}$$

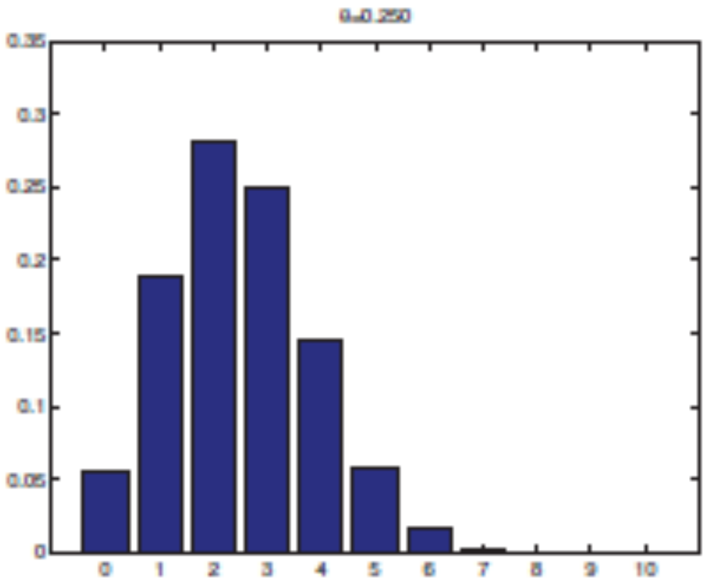
Example: Get m heads in N coin flips.

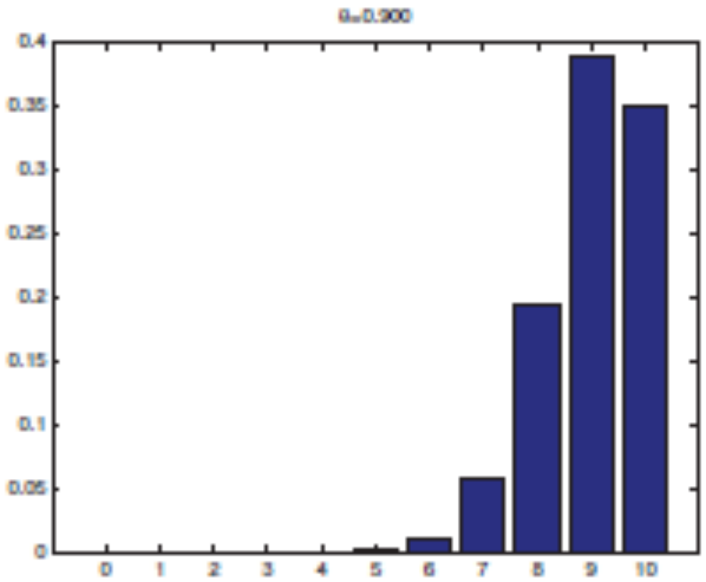
$$\mathbb{E}[m] \equiv \sum_{m=0}^{N} m\mathrm{Bin}(m|N,\mu) = N\mu$$

$$\mathrm{var}[m] \equiv \sum_{m=0}^{N} (m - \mathbb{E}[m])^2 \, \mathrm{Bin}(m|N,\mu) = N\mu(1-\mu)$$

Binomial Distribution (2/2)

N = 10
μ = 0.25





N = 10
μ = 0.9

(Ref: Murphy, “Machine Learning: A Probabilistic Perspective”)

Multinomial Distribution (1/3)

A generalization of Bernoulli distribution and Binomial distribution to more than two outcomes.

1-of-K coding scheme: $x = (0, 0, 1, 0, 0, 0)^T$

$$\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)^{\mathrm{T}}$$

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k} \qquad \sum_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\mu}) = \sum_{k=1}^{K} \mu_k = 1$$

$\forall k$: $\mu_k \geq 0$ and $\sum_{k=1}^{K} \mu_k = 1$

$$\mathbb{E}[\mathbf{x}|\boldsymbol{\mu}] = \sum_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\mu})\mathbf{x} = (\mu_1, \ldots, \mu_M)^{\mathrm{T}} = \boldsymbol{\mu}$$

Multinomial Distribution (2/3)

Consider a data set D of N independent observations $\mathbf{x}_1, \ldots, \mathbf{x}_N$.

$$p(\mathcal{D}|\boldsymbol{\mu}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\mu_k^{x_{nk}} = \prod_{k=1}^{K}\mu_k^{\left(\sum_n x_{nk}\right)} = \prod_{k=1}^{K}\mu_k^{m_k}$$

To find the maximum likelihood estimation of μ,

$\Rightarrow$ Maximize $$\sum_{k=1}^{K} m_k \ln \mu_k + \lambda\left(\sum_{k=1}^{K}\mu_k - 1\right)$$

$\Rightarrow$ $\mu_k = -m_k/\lambda.$ $\qquad \mu_k^{\mathrm{ML}} = \dfrac{m_k}{N}$

Multinomial Distribution (3/3)

$$\begin{aligned}
\mathrm{Mult}(m_1, m_2, \ldots, m_K|\boldsymbol{\mu}, N) &= \binom{N}{m_1 m_2 \ldots m_M} \prod_{k=1}^{M} \mu_k^{m_k} \\
\mathbb{E}[m_k] &= N\mu_k \\
\mathrm{var}[m_k] &= N\mu_k(1-\mu_k) \\
\mathrm{cov}[m_j m_k] &= -N\mu_j\mu_k
\end{aligned}$$

Multinomial distribution

Beta Distribution (1/2)

$$\mathrm{Beta}(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1}$$

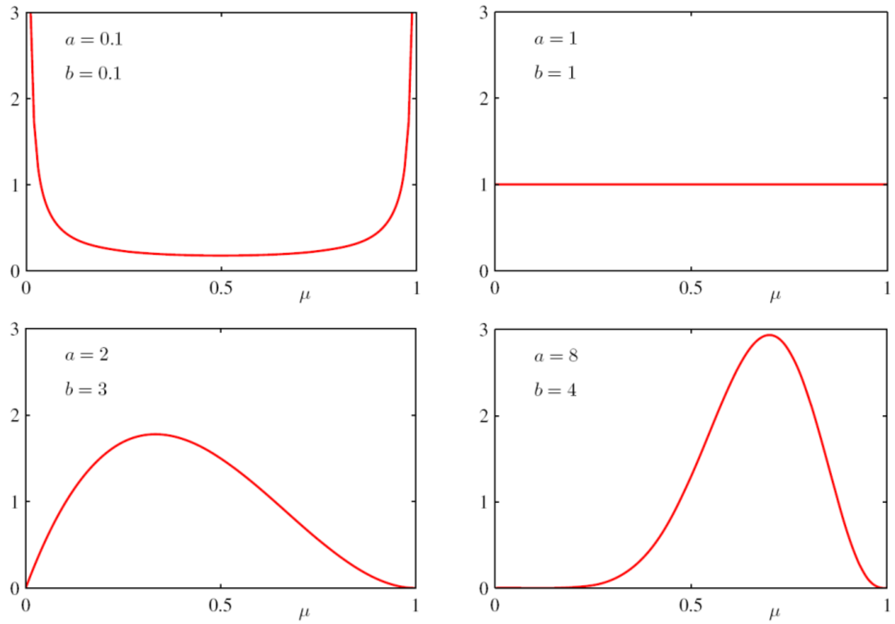
a, b: hyperparameters

Γ(x): Gamma function

Distribution over $\mu \in [0,1]$

$$\mathbb{E}[\mu] = \frac{a}{a+b}$$

$$\mathrm{var}[\mu] = \frac{ab}{(a+b)^2(a+b+1)}$$



Beta Distribution (2/2)

- Provide the conjugate prior for the Bernoulli distribution and Binomial distribution

Beta Posterior = Binomial (or Bernoulli) Likelihood × Beta Prior

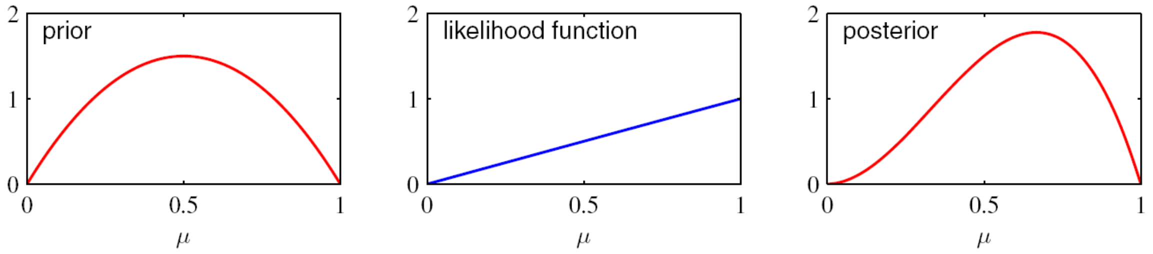


Figure 2.3 Illustration of one step of sequential Bayesian inference. The prior is given by a beta distribution with parameters $a = 2$, $b = 2$, and the likelihood function, given by (2.9) with $N = m = 1$, corresponds to a single observation of $x = 1$, so that the posterior is given by a beta distribution with parameters $a = 3$, $b = 2$.

$$p(\mu|m, l, a, b) = \frac{\Gamma(m + a + l + b)}{\Gamma(m + a)\Gamma(l + b)} \mu^{m+a-1}(1 - \mu)^{l+b-1}$$

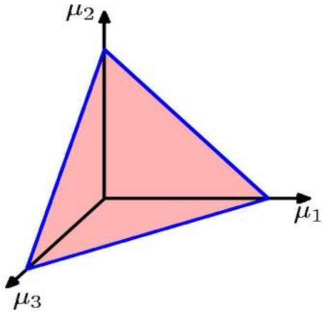
where *l* = *N* – *m*.

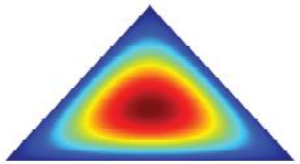
Remark: a and b can be interpreted as the effective number of observations of x = 1 and x = 0.

Dirichlet Distribution (1/2)

Conjugate prior for the multinomial distribution.

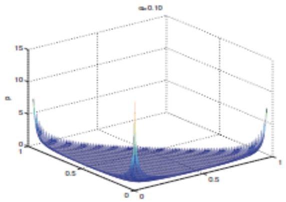
$$\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1} \qquad \alpha_0 = \sum_{k=1}^{K} \alpha_k$$





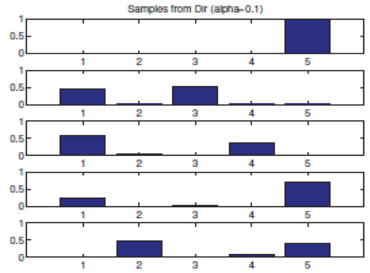
α = (2, 2, 2)

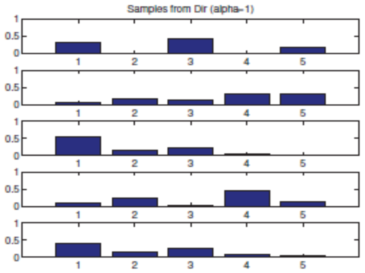
α = (20, 2, 2)



α = (0.1, 0.1, 0.1)

α = (0.1, 0.1, 0.1, 0.1, 0.1)





α = (1, 1, 1, 1, 1)

(Ref: Murphy, "Machine Learning: A Probabilistic Perspective")

Dirichlet Distribution (2/2)

Dirichlet Posterior = Multinomial Likelihood $\times$ *Dirichlet Prior*

$$p(\boldsymbol{\mu}|\mathcal{D}, \boldsymbol{\alpha}) \propto p(\mathcal{D}|\boldsymbol{\mu})p(\boldsymbol{\mu}|\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} \mu_k^{\alpha_k + m_k - 1}$$

$$\begin{aligned} p(\boldsymbol{\mu}|\mathcal{D}, \boldsymbol{\alpha}) &= \mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha} + \mathbf{m}) \\ &= \frac{\Gamma(\alpha_0 + N)}{\Gamma(\alpha_1 + m_1) \cdots \Gamma(\alpha_K + m_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k + m_k - 1} \end{aligned}$$

Exponential Family (1/5)

$$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta}) \exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$

$$g(\boldsymbol{\eta}) \int h(\mathbf{x}) \exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\} \mathrm{d}\mathbf{x} = 1$$

η: natural parameters

Exponential Family (2/5)

Example: Bernoulli distribution

$$p(x|\mu) = \mathrm{Bern}(x|\mu) = \mu^x(1-\mu)^{1-x}$$

$$\begin{aligned} p(x|\mu) &= \exp\{x \ln \mu + (1-x)\ln(1-\mu)\} \\ &= (1-\mu)\exp\left\{\ln\left(\frac{\mu}{1-\mu}\right)x\right\}. \end{aligned}$$

$$\eta = \ln\left(\frac{\mu}{1-\mu}\right) \quad \Rightarrow \quad \sigma(\eta) = \frac{1}{1+\exp(-\eta)}$$

logistic sigmoid function

$$p(x|\eta) = \sigma(-\eta)\exp(\eta x)$$

$$\begin{aligned} u(x) &= x \\ h(x) &= 1 \\ g(\eta) &= \sigma(-\eta) \end{aligned}$$

Exponential Family (3/5)

Example: Multinomial distribution

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{M} \mu_k^{x_k} = \exp\left\{\sum_{k=1}^{M} x_k \ln \mu_k\right\}$$

$$p(\mathbf{x}|\boldsymbol{\eta}) = \exp(\boldsymbol{\eta}^{\mathrm{T}}\mathbf{x})$$

where $\eta_k = \ln \mu_k$

$$\begin{aligned} \mathbf{u}(\mathbf{x}) &= \mathbf{x} \\ h(\mathbf{x}) &= 1 \\ g(\boldsymbol{\eta}) &= 1. \end{aligned}$$

$$\sum_{k=1}^{M} \mu_k = 1$$

the parameters η_k are not independent.

$$\begin{aligned} &\exp\left\{\sum_{k=1}^{M} x_k \ln \mu_k\right\} \\ &= \exp\left\{\sum_{k=1}^{M-1} x_k \ln \mu_k + \left(1 - \sum_{k=1}^{M-1} x_k\right) \ln\left(1 - \sum_{k=1}^{M-1} \mu_k\right)\right\} \\ &= \exp\left\{\sum_{k=1}^{M-1} x_k \ln\left(\frac{\mu_k}{1 - \sum_{j=1}^{M-1} \mu_j}\right) + \ln\left(1 - \sum_{k=1}^{M-1} \mu_k\right)\right\}. \end{aligned}$$

Exponential Family (4/5)

We now identify

$$\ln\left(\frac{\mu_k}{1-\sum_j \mu_j}\right) = \eta_k \qquad \Rightarrow \qquad \mu_k = \frac{\exp(\eta_k)}{1+\sum_j \exp(\eta_j)}.$$

softmax function (normalized exponential)

$$p(\mathbf{x}|\boldsymbol{\eta}) = \left(1+\sum_{k=1}^{M-1}\exp(\eta_k)\right)^{-1} \exp(\boldsymbol{\eta}^{\mathrm{T}}\mathbf{x})$$

$$\begin{aligned}\mathbf{u}(\mathbf{x}) &= \mathbf{x}\\ h(\mathbf{x}) &= 1\\ g(\boldsymbol{\eta}) &= \left(1+\sum_{k=1}^{M-1}\exp(\eta_k)\right)^{-1}\end{aligned}$$

Exponential Family (5/5)

Example: Gaussian distribution

$$\begin{aligned}
p(x|\mu,\sigma^2) &= \frac{1}{(2\pi\sigma^2)^{1/2}}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} \\
&= \frac{1}{(2\pi\sigma^2)^{1/2}}\exp\left\{-\frac{1}{2\sigma^2}x^2+\frac{\mu}{\sigma^2}x-\frac{1}{2\sigma^2}\mu^2\right\}
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\eta} &= \begin{pmatrix} \mu/\sigma^2 \\ -1/2\sigma^2 \end{pmatrix} \\
\mathbf{u}(x) &= \begin{pmatrix} x \\ x^2 \end{pmatrix} \\
h(\mathbf{x}) &= (2\pi)^{-1/2} \\
g(\boldsymbol{\eta}) &= (-2\eta_2)^{1/2}\exp\left(\frac{\eta_1^2}{4\eta_2}\right)
\end{aligned}$$

Classification Problem

Introduction

Classification Problem:

The goal in classification is to take an input vector **x** and to assign it to one of K discrete classes C_k where $k = 1, \ldots, K$.

✓ ***Linear models for classification:***

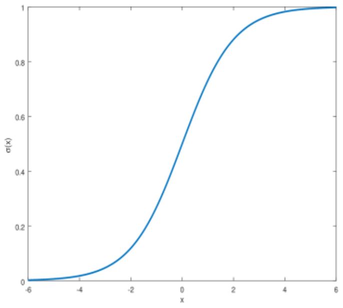
$$y(\mathbf{x}) = f\left(\mathbf{w}^T \phi(\mathbf{x})\right)$$

$$\mathbf{w} = \begin{bmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_{M-1} \end{bmatrix} \quad \phi(\mathbf{x}) = \begin{bmatrix} 1 \\ \phi_1(\mathbf{x}) \\ \vdots \\ \phi_{M-1}(\mathbf{x}) \end{bmatrix}$$

Generalized Linear Model

$f(\cdot)$: ***activation function***

decision surface: $\mathbf{w}^T \phi(\mathbf{x})$ = constant



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Logistic Sigmoid Function

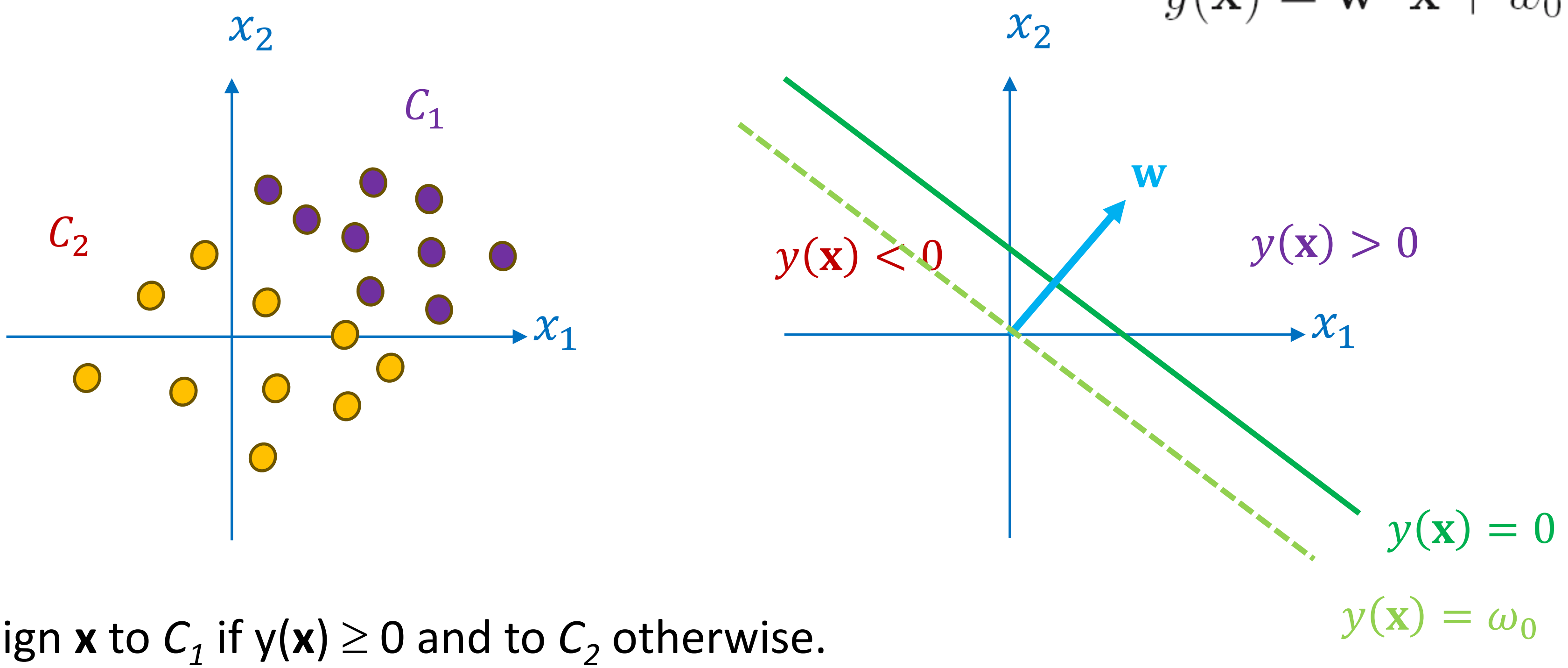
Major Approaches

- **Discriminant Function**: A function that takes an input vector x and assigns it to one of K classes.
  - Linear discriminant, the perceptron algorithm
- **Probabilistic Generative Model**:
  - $P(\mathbf{x},C_k)$
- **Probabilistic Discriminative Model**:
  - $P(C_k|\mathbf{x})$
  - Logistic regression

Linear Discriminant (1/7)

Two Classes

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

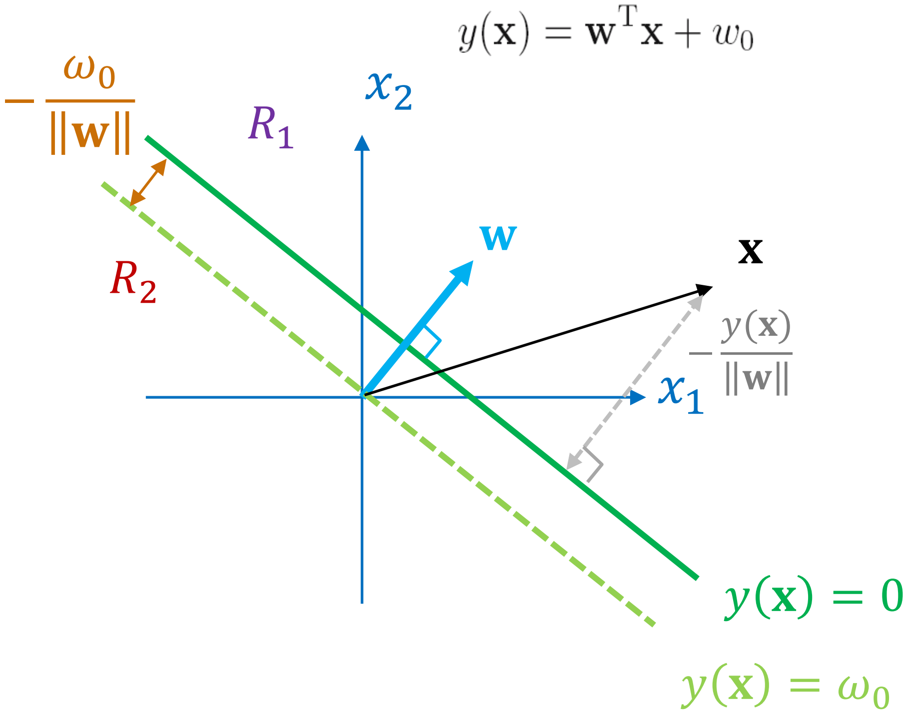


Assign $\mathbf{x}$ to C_1 if $y(\mathbf{x}) \geq 0$ and to C_2 otherwise.

w: weight vector, determine the orientation of the decision surface

w_0: bias   (- w_0: threshold)

Linear Discriminant (2/7)



Remarks:

1. The decision surface y(**x**) = 0 is a (D-1)-dimensional hyperplane in the D-dimensional input space.
2. The distance from the origin to the decision surface is $\frac{\mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|} = -\frac{\omega_0}{\|\mathbf{w}\|}$
3. The signed perpendicular distance *r* of a point **x** from the decision surface is $r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}$.

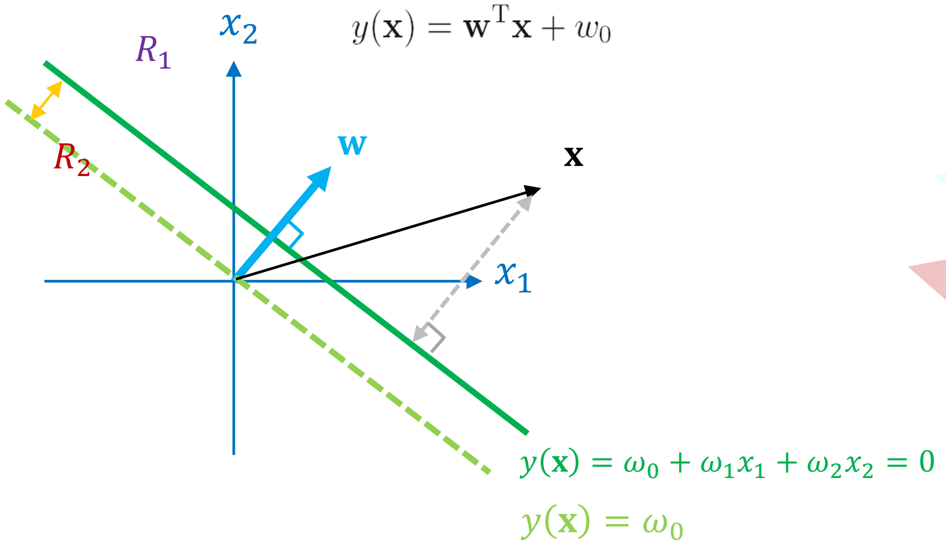
Linear Discriminant (3/7)

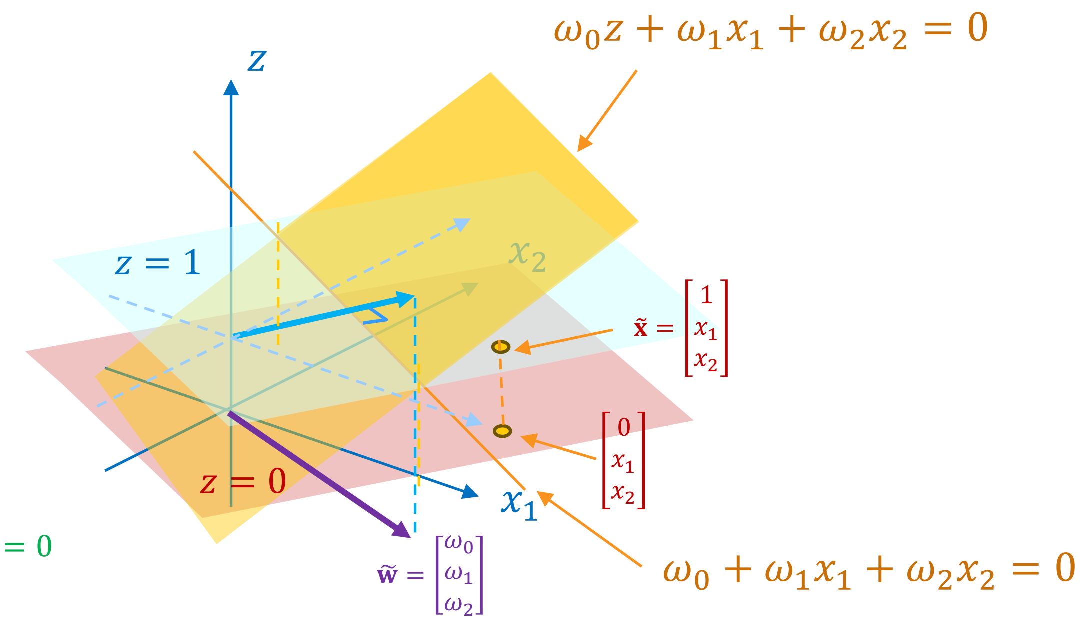
With the introduction of $\tilde{\mathbf{w}} = (\omega_0, \mathbf{w})$ and $\tilde{\mathbf{x}} = (1, \mathbf{x})$, we have $y(\mathbf{x}) = \tilde{\mathbf{w}}^T\tilde{\mathbf{x}}$.

- The decision surface is a D-dimensional hyperplane passing through the origin of the (D+1)-dimensional input space.

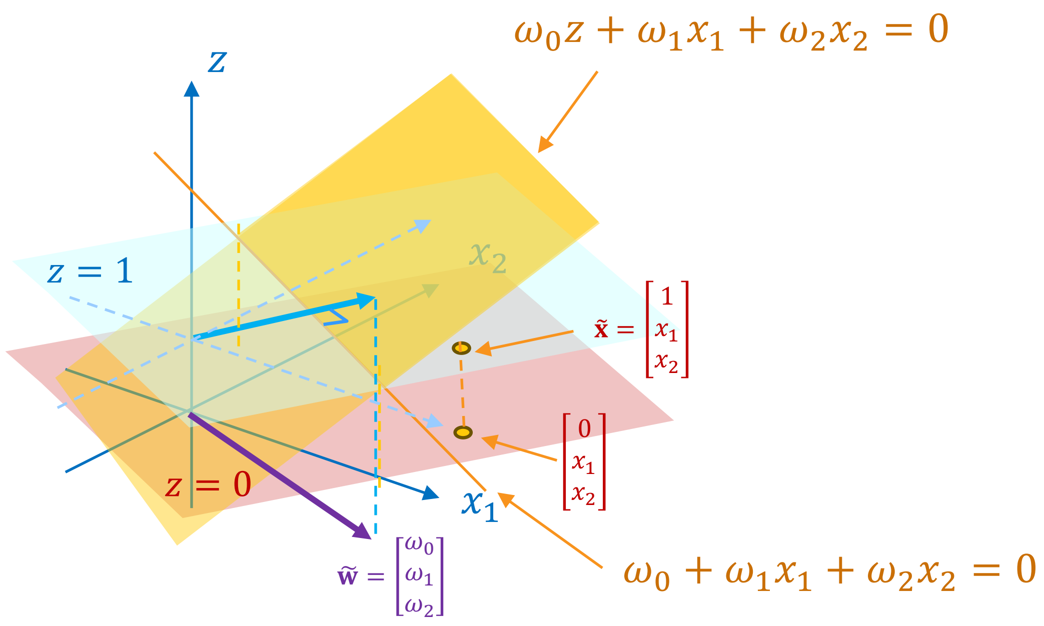
Example: $y(\mathbf{x}) = \omega_0 + \omega_1 x_1 + \omega_2 x_2$

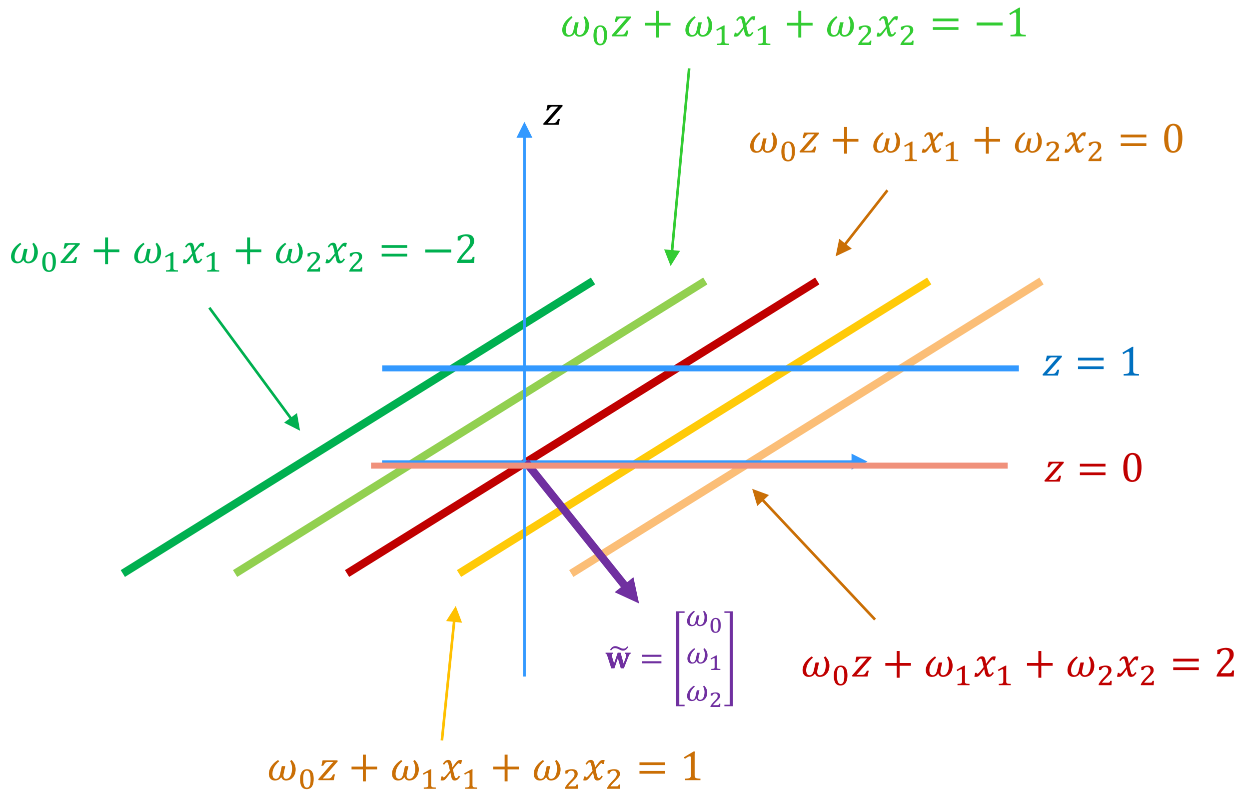
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



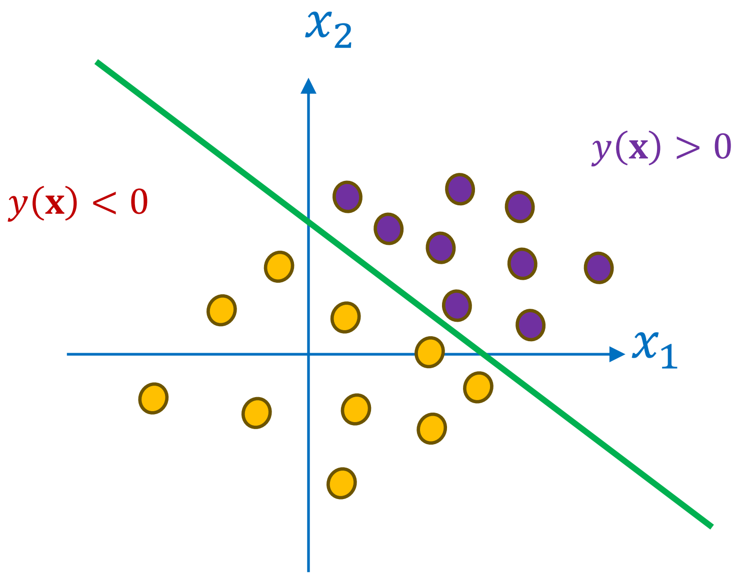


Linear Discriminant (4/7)

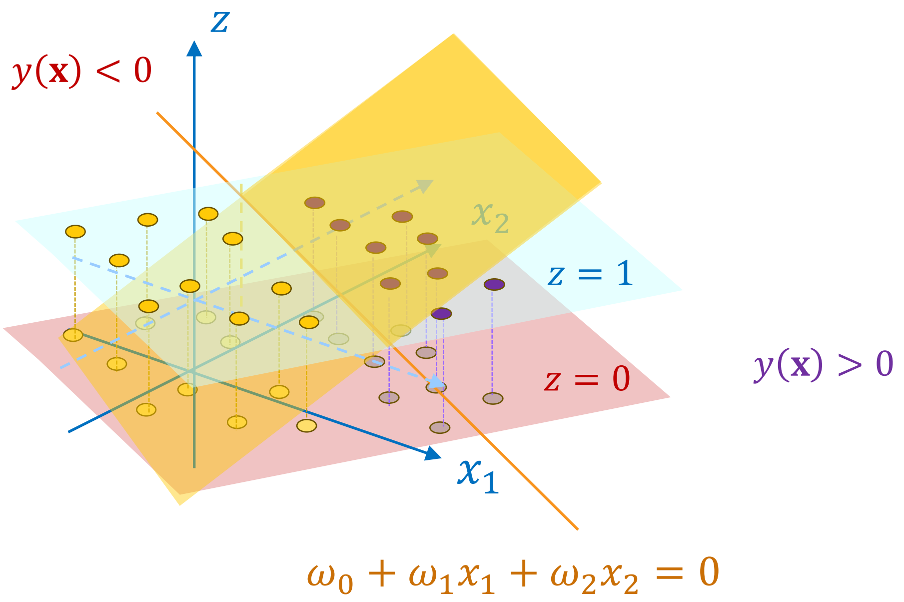




Linear Discriminant (5/7)



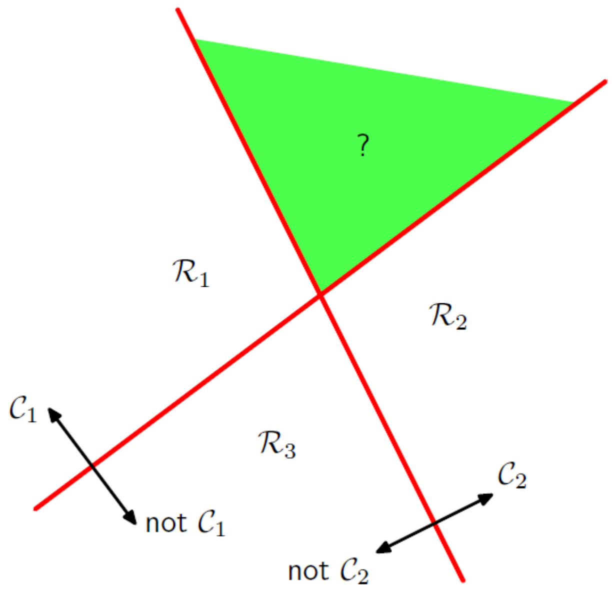
$\mathbf{w}^T\mathbf{x} + \omega_0$



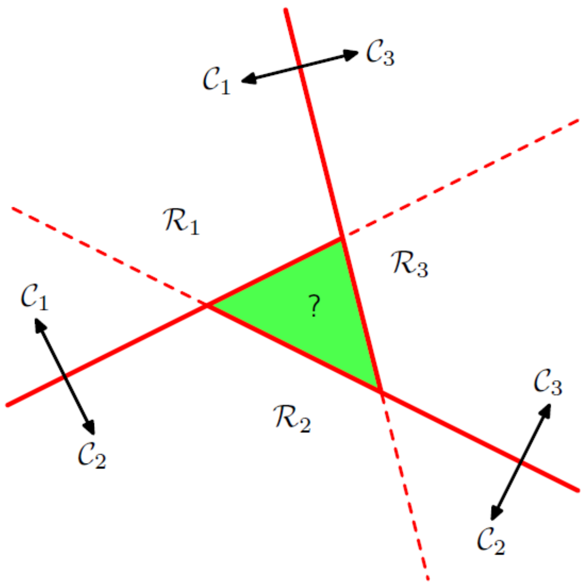
$y(\mathbf{x}) = \tilde{\mathbf{w}}^T\tilde{\mathbf{x}}$

Linear Discriminant (6/7)

Multiple Classes



one-versus-the-rest classifier

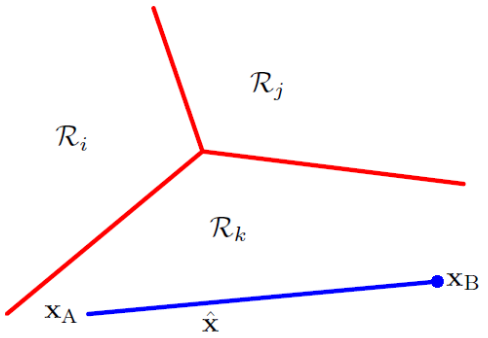


one-versus-one classifier

Linear Discriminant (7/7)

K-class discriminant: K linear functions

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



Assign a point $\mathbf{x}$ to class C_k if $y_k(\mathbf{x}) > y_j(\mathbf{x})$ for all $j \neq k$.

The decision boundary between C_k and C_j corresponds to a (D-1)-dimensional hyperplane defined by

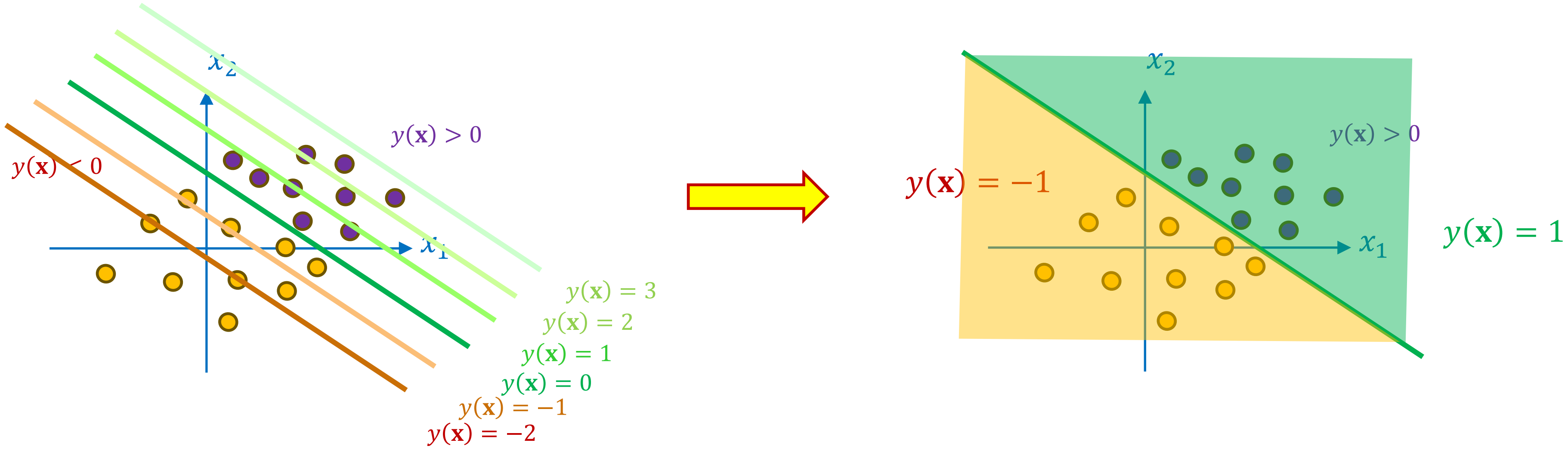
$$(\mathbf{w}_k - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{k0} - w_{j0}) = 0.$$

Remark: The decision regions are always singly connected and convex.

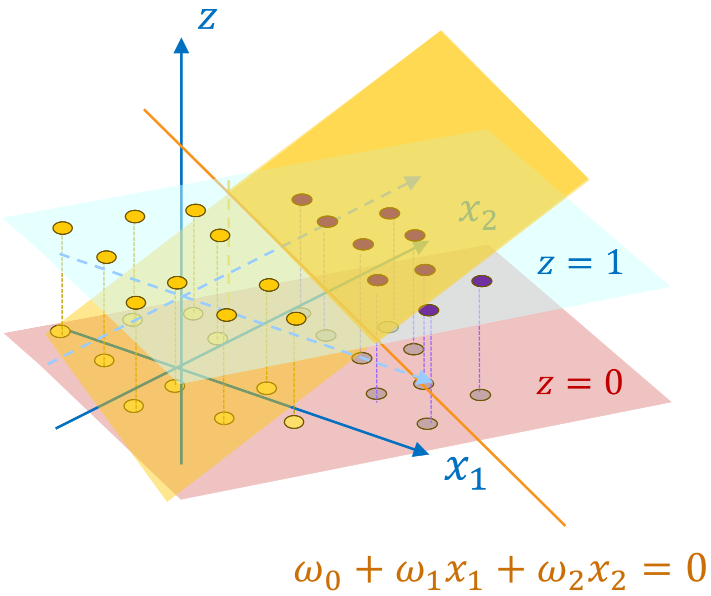
Perceptron Algorithm (1/5)

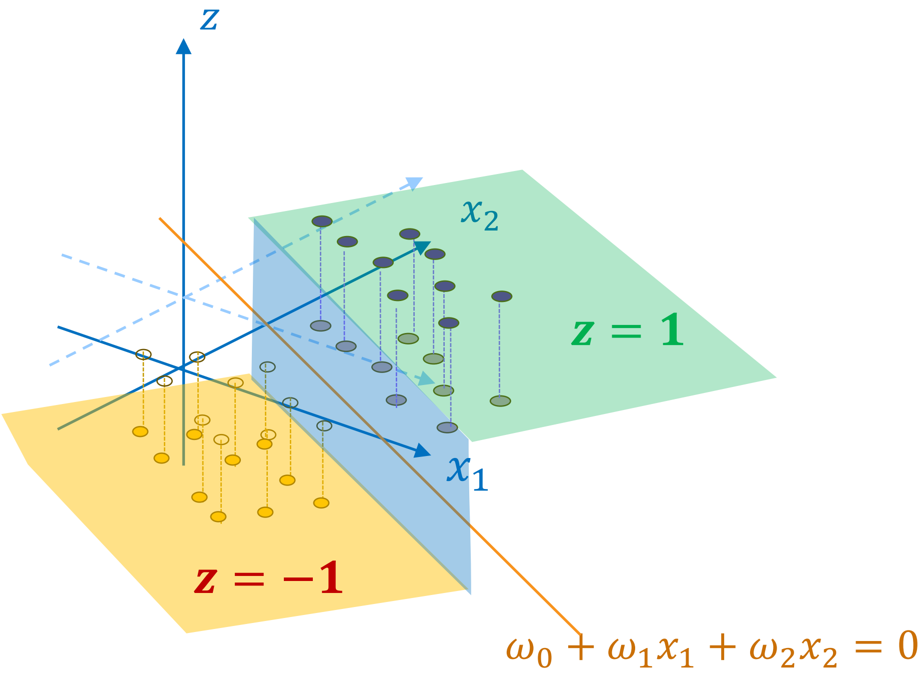
$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})\right)$$

where $$f(a) = \begin{cases} +1, & a \geqslant 0 \\ -1, & a < 0. \end{cases}$$



Perceptron Algorithm (2/5)





Perceptron Algorithm (3/5)

To determine the parameters w of the perceptron, we adopt the ***perceptron criterion***

$$E_P(\mathbf{w}) = -\sum_{n\in\mathcal{M}} \mathbf{w}^T \phi_n t_n$$

where $$\phi_n = \phi(\mathbf{x}_n)$$

M: all misclassified patterns

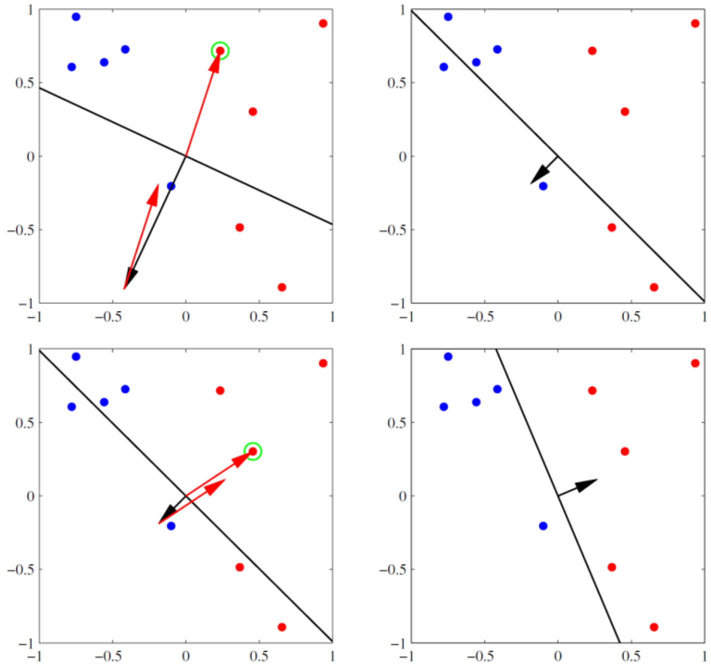
Remark: The contribution to the error associated with a particular misclassified pattern is a linear function of **w** if the pattern is misclassified and is zero if the pattern is correctly classified.

Perceptron Algorithm (4/5)

The Perceptron Learning Algorithm

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_{\mathrm{P}}(\mathbf{w}) = \mathbf{w}^{(\tau)} + \eta \boldsymbol{\phi}_n t_n$$

η: learning rate parameter



Red: C_1
Blue: C_2

Perceptron Algorithm (5/5)

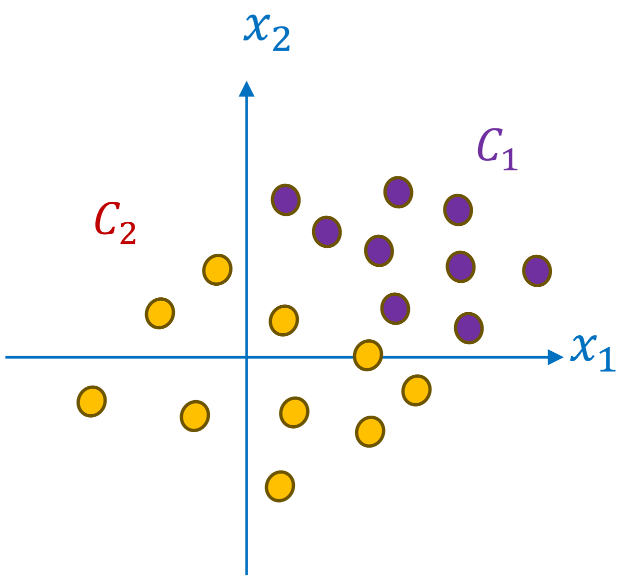
Remarks:

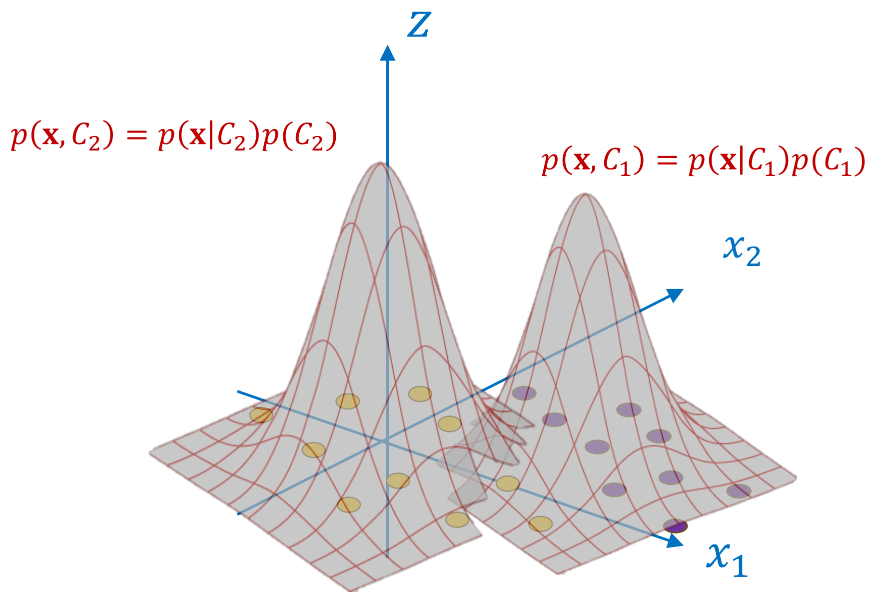
1. If the training data set is linearly separable, then the perceptron learning algorithm is guaranteed to find an exact solution in a finite number of steps.
2. For data sets that are not linearly separable, the perceptron learning algorithm will never converge.

$$
\begin{aligned}
-w^{(\tau+1)^T}\phi_n t_n &= -w^{(\tau)^T}\phi_n t_n - (\phi_n t_n)^T \phi_n t_n \\
&< -w^{(\tau)^T}\phi_n t_n
\end{aligned}
$$

Probabilistic Generative Models (1/14)

For two-class problems,





Probabilistic Generative Models (2/14)

Two-class case

Assume each class has a Gaussian class-conditional density with a shared covariance matrix.

Training data: $\{\mathbf{x}_n, t_n\}$, $n = 1, \ldots, N$. C_1: $t_n = 1$ C_2: $t_n = 0$

Prior class probability: $p(C_1) = \pi$, $p(C_2) = 1 - \pi$.

$$p(\mathbf{x}_n, C_1) = p(C_1)p(\mathbf{x}_n|\mathrm{C}_1) = \pi N(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$$

$$p(\mathbf{x}_n, C_2) = p(C_2)p(\mathbf{x}_n|\mathrm{C}_2) = (1-\pi) N(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$$

$$p(\boldsymbol{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} [\pi N(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} [(1-\pi) N(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n}$$

$$\ln p(\boldsymbol{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \{t_n \ln \pi + t_n \ln N(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + (1-t_n)\ln(1-\pi) + (1-t_n)\ln N(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})\}$$

Probabilistic Generative Models (3/14)

Maximization with respect to $\pi \Rightarrow$ $$\pi = \frac{1}{N}\sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2}$$

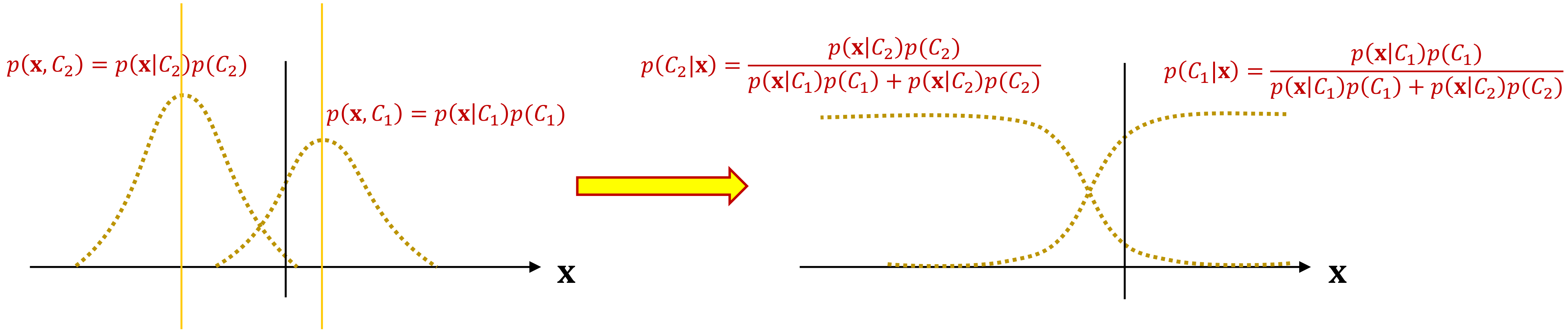
Maximization with respect to $\mu_1 \Rightarrow$ $$\boldsymbol{\mu}_1 = \frac{1}{N_1}\sum_{n=1}^{N} t_n \mathbf{x}_n$$

Maximization with respect to $\mu_2 \Rightarrow$ $$\boldsymbol{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N} (1 - t_n)\mathbf{x}_n$$

Maximization with respect to $\Sigma \Rightarrow$ $$\boldsymbol{\Sigma} = \frac{N_1}{N}\mathbf{S}_1 + \frac{N_2}{N}\mathbf{S}_2$$

$$\mathbf{S}_1 = \frac{1}{N_1}\sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \boldsymbol{\mu}_1)(\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}} \qquad \mathbf{S}_2 = \frac{1}{N_2}\sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \boldsymbol{\mu}_2)(\mathbf{x}_n - \boldsymbol{\mu}_2)^{\mathrm{T}}.$$

Probabilistic Generative Models (4/14)



$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}, C_1)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2)} = \frac{1}{1 + e^{-a(\mathbf{x})}} \equiv \sigma(a(\mathbf{x}))$$

where $\sigma(a)$ is the *logistic sigmoid* function defined by $\boldsymbol{\sigma}(\boldsymbol{a}) = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-\boldsymbol{a}}}$ and $a(x) = \ln \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)}$

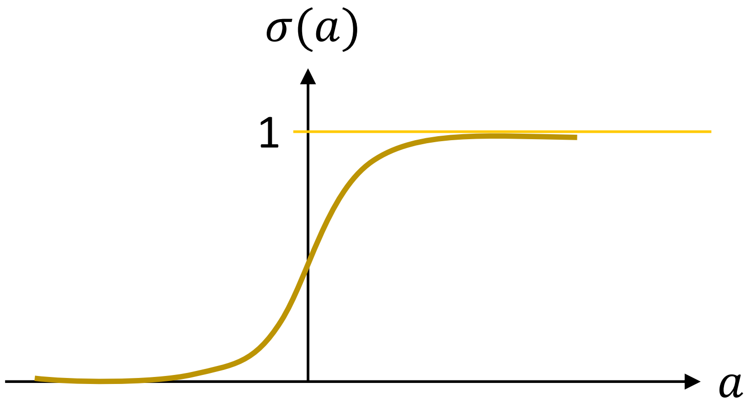
Remarks: 1. $\sigma(-a) = 1 - \sigma(a)$

2. $a = \ln(\frac{\sigma}{1-\sigma})$ *is called the logit function.*

Probabilistic Generative Models (5/14)

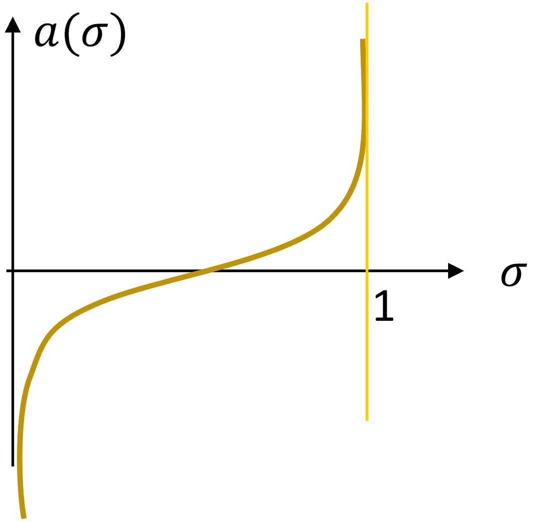
Logistic Sigmoid Function

$$\boldsymbol{\sigma(a) = \frac{1}{1 + e^{-a}}}$$



Logit Function

$$\boldsymbol{a = \ln(\frac{\sigma}{1 - \sigma})}$$



Probabilistic Generative Models (6/14)

Continuous Inputs

Assume the class-conditional densities are Gaussian with the same covariance matrix.

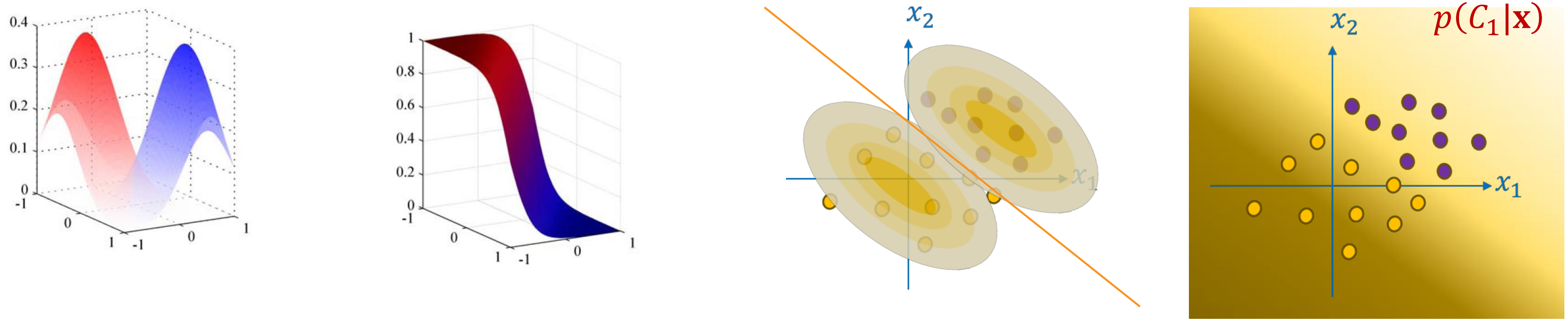
$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

For the two-class case,

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)$$

$$\begin{aligned} \mathbf{w} &= \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \\ w_0 &= -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \end{aligned}$$

Probabilistic Generative Models (7/14)

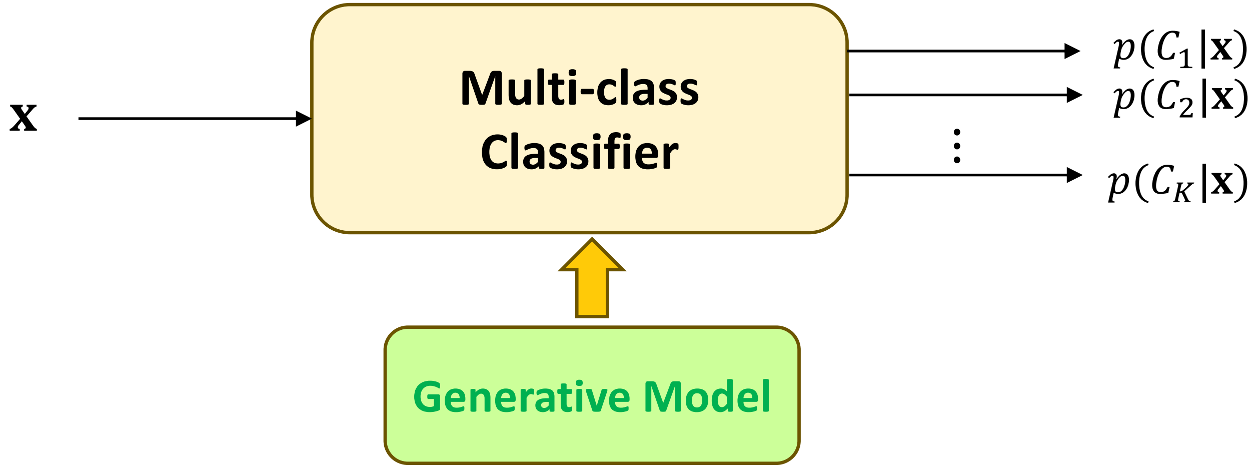


Remarks:

1. The decision boundaries are linear in input space.
2. The prior probabilities $p(C_k)$ only affects the bias parameter w_0.
3. Changes in the priors cause parallel shifts of the decision boundary.

Probabilistic Generative Models (8/14)

For multi-class problems,



$$
\begin{aligned}
p(\mathcal{C}_k|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)} \\
&= \frac{\exp(a_k)}{\sum_j \exp(a_j)}
\end{aligned}
$$

softmax function

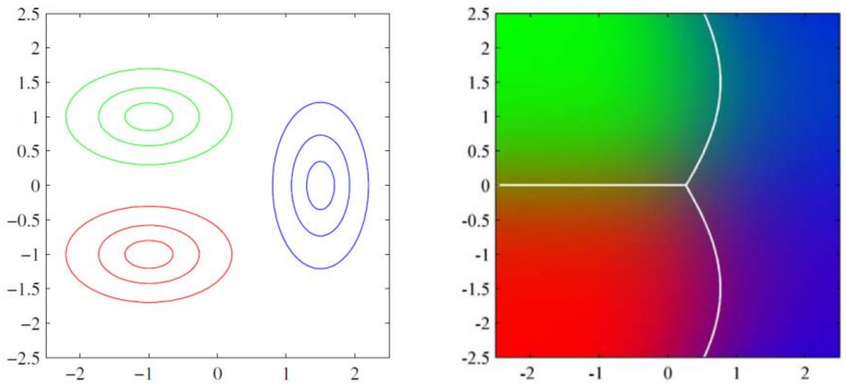
where $a_k = \ln p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$

Probabilistic Generative Models (9/14)

For the multiple-class case with the same covariance matrix

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

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



Remarks:

1. With the same covariance matrix, $a_k(\mathbf{x})$ are linear functions of $\mathbf{x}$.
2. If we allow each class-conditional density $p(\mathbf{x}|C_k)$ to have its own covariance matrix $\boldsymbol{\Sigma}_k$, then we will obtain quadratic functions of $\mathbf{x}$, giving rise to a *quadratic discriminant*.

Probabilistic Generative Models (10/14)

- If there are K classes, we first construct the K models $p(\mathbf{x}, C_i)$, for i = 1, 2, …, K. Based on $p(\mathbf{x}, C_i)$, we compute the desired posterior probabilities $p(C_i|\mathbf{x})$.
- Construction of $p(\mathbf{x}, C_i)$
  - ✓ For each model , we assume it follows a certain parametric probabilistic model.

    For example, we may assume

    $$p(\mathbf{x}, C_i; \boldsymbol{\theta}_i) = p(C_i; \boldsymbol{\theta}_i)p(\mathbf{x}|C_i; \boldsymbol{\theta}_i) = \pi_i N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \text{ where } \boldsymbol{\theta}_i = \{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$$

  - ✓ We collect the set of training data $\mathbf{D} \equiv \{(\mathbf{x}_1, \boldsymbol{t}_1), (\mathbf{x}_2, \boldsymbol{t}_2), \dots, (\mathbf{x}_N, \boldsymbol{t}_N)\}$, where $\boldsymbol{t}_n$’s are represented in the 1-of-K format.

Probabilistic Generative Models (11/14)

- ✓ Based on the training data D and the model $p(\mathbf{x}, C_i)$ for each class, we form the likelihood function

$$p(D|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K) = \prod_{n=1}^{N} p(\mathbf{x}_n, C_1; \boldsymbol{\theta}_1)^{t_{n1}}\, p(\mathbf{x}_n, C_2; \boldsymbol{\theta}_2)^{t_{n2}} \ldots p(\mathbf{x}_n, C_\mathrm{K}; \boldsymbol{\theta}_\mathrm{K})^{t_{nK}}$$

$$= \prod_{n=1}^{N} \prod_{k=1}^{K} p(\mathbf{x}_n, C_k; \boldsymbol{\theta}_k)^{t_{nk}}$$

$$\Rightarrow \quad E(\boldsymbol{\Theta}) = -\ln p(D|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K)$$

$$= -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln p(\mathbf{x}_n, C_k; \boldsymbol{\theta}_k)$$

Probabilistic Generative Models (12/14)

- ✓ Based on $E(\mathbf{\Theta})$, we compute $\nabla E(\mathbf{\Theta}) = \mathbf{0}$ to find the optimal set of the model parameters $\mathbf{\Theta}^{ML}$.
- ✓ Based on $\mathbf{\Theta}^{ML}$, we construct the generative models $p(\mathbf{x}, C_i)$ for i = 1, 2, …, K.
- ✓ Based on $p(\mathbf{x}, C_i)$, we deduce $p(C_i|\mathbf{x})$.

$$p(C_i|\mathbf{x}) = \frac{p(\mathbf{x}, C_i)}{\sum_{\boldsymbol{k}=\mathbf{1}}^{\boldsymbol{K}} p(\mathbf{x}, C_k)}$$

Probabilistic Generative Models (13/14)

Exponential Family

$$p(\mathbf{x}|\boldsymbol{\lambda}_k) = h(\mathbf{x})g(\boldsymbol{\lambda}_k)\exp\left\{\boldsymbol{\lambda}_k^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$

For the subclass with **u**(**x**) = **x**

$$p(\mathbf{x}|\boldsymbol{\lambda}_k, s) = \frac{1}{s}h\left(\frac{1}{s}\mathbf{x}\right)g(\boldsymbol{\lambda}_k)\exp\left\{\frac{1}{s}\boldsymbol{\lambda}_k^{\mathrm{T}}\mathbf{x}\right\}$$

For the two-class problem

$$a(\mathbf{x}) = \frac{1}{s}(\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2)^T\mathbf{x} + \ln g(\boldsymbol{\lambda}_1) - \ln g(\boldsymbol{\lambda}_2) + \ln p(C_1) - \ln p(C_2)$$

For the multiclass problem

$$a_k(\mathbf{x}) = \frac{1}{s}\boldsymbol{\lambda}_k{}^T\mathbf{x} + \ln g(\boldsymbol{\lambda}_k) + \ln p(C_k)$$

Both are linear functions of x.

Probabilistic Generative Models (14/14)

Summary:

For a wide choice of class-conditional distributions $p(x|C_k)$, $p(C_k|x)$ is a logistic sigmoid function of a linear function of x for the two-class classification problem, and is the softmax transformation of a linear function of x for the multiclass case.

Probabilistic Discriminative Models (1/17)

Simpler forms

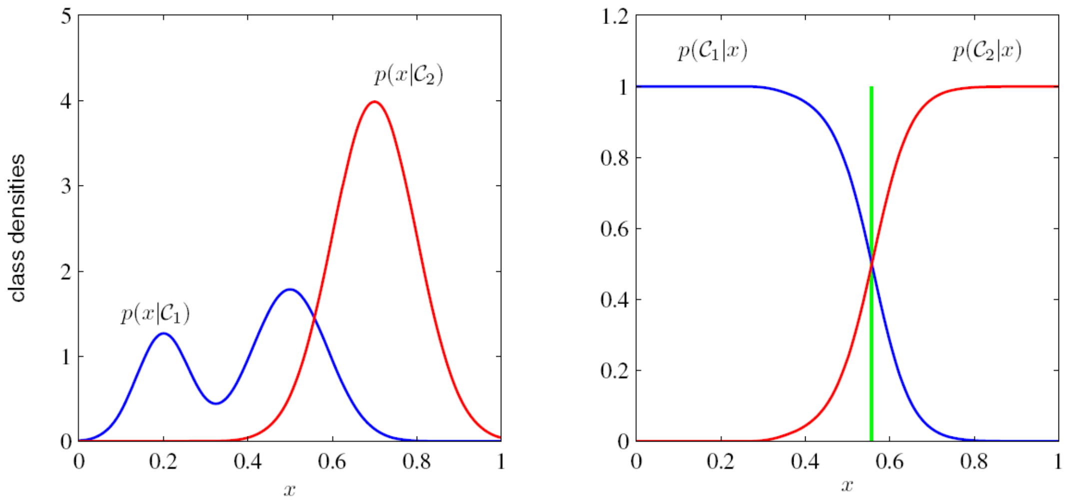
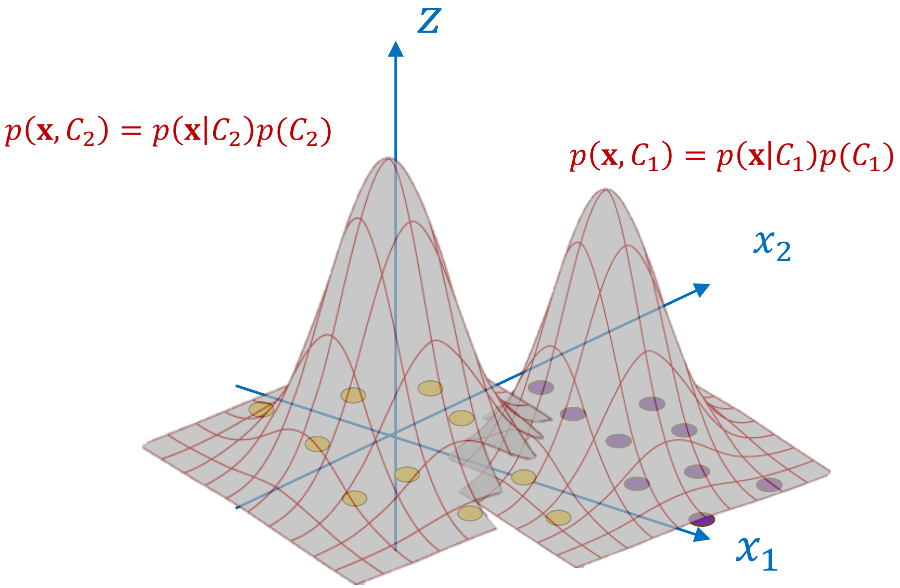


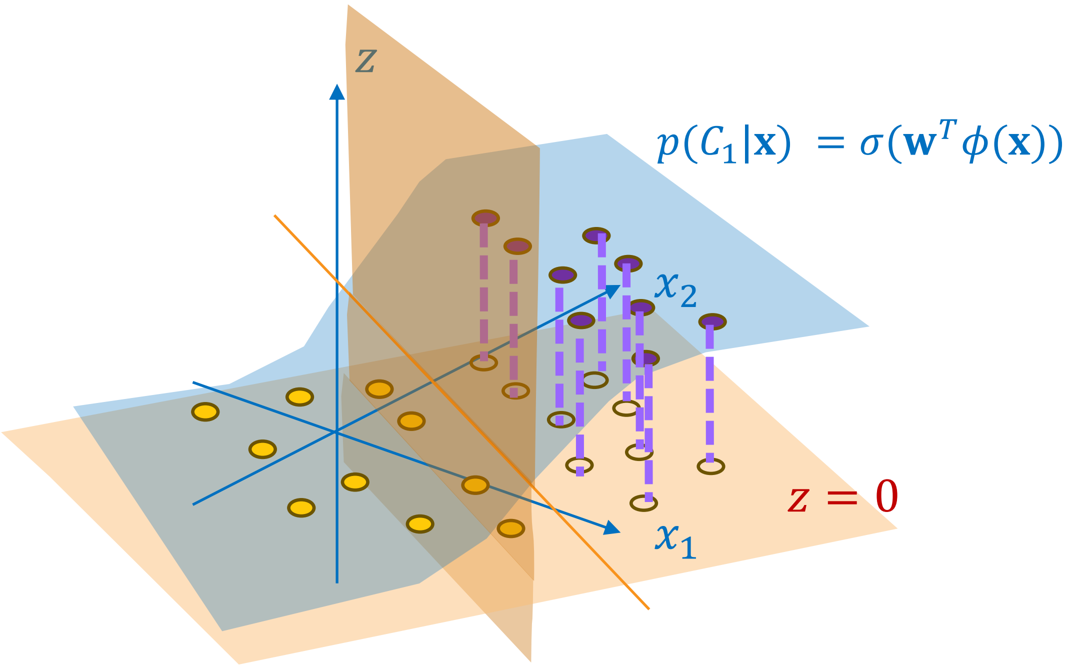
Figure 1.27 Example of the class-conditional densities for two classes having a single input variable x (left plot) together with the corresponding posterior probabilities (right plot). Note that the left-hand mode of the class-conditional density $p(\mathbf{x}|\mathcal{C}_1)$, shown in blue on the left plot, has no effect on the posterior probabilities. The vertical green line in the right plot shows the decision boundary in x that gives the minimum misclassification rate.

We can trivially revise the minimum risk decision criterion when the elements of the loss matrix are subject to revision from time to time.

Probabilistic Discriminative Models (2/17)



Generative Model



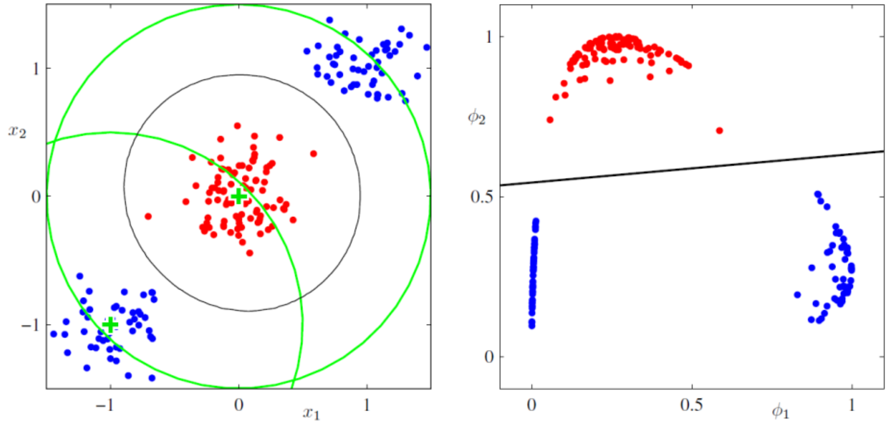
Discriminative Model

Probabilistic Discriminative Models (3/17)

Use the functional form of the generalized linear model explicitly and determine its parameters directly.

Logistic Regression

$$p(\mathcal{C}_1|\phi) = y(\phi) = \sigma\left(\mathbf{w}^{\mathrm{T}}\phi\right) \qquad p(\mathcal{C}_2|\phi) = 1 - p(\mathcal{C}_1|\phi)$$



Remark: The use of nonlinear basis functions can help in dealing with classes that are not linearly separable.

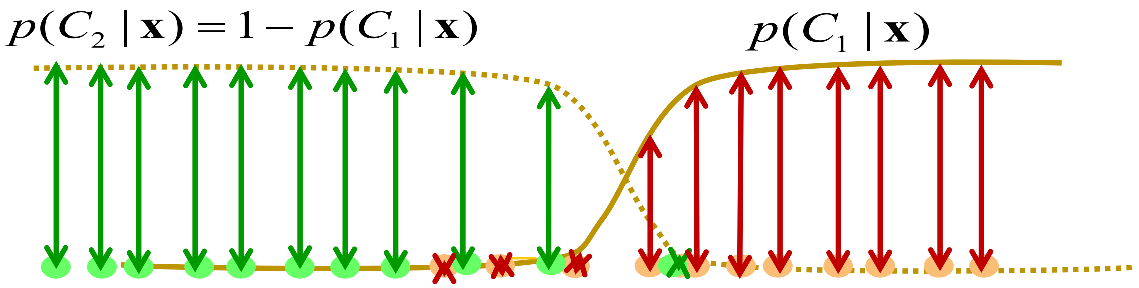
Probabilistic Discriminative Models (4/17)

$$\frac{d\sigma}{da} = \sigma(1-\sigma)$$

For a data set $\{\phi_n, t_n\}$, where $t_n \in \{0, 1\}$ and $\phi_n = \phi(\mathrm{x}_n)$, with $n = 1, \ldots, N$.

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{\,t_n} (1-y_n)^{1-t_n}$$

where $\mathrm{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$ and $y_n = p(C_1 | \phi_n) = \sigma(a_n)$ and $a_n = \mathbf{w}^{\mathrm{T}}\phi_n$.

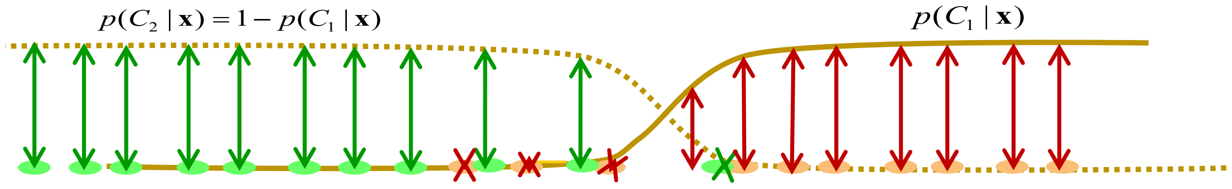


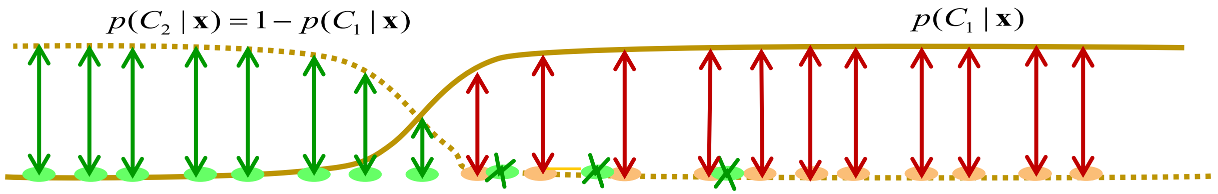
Probabilistic Discriminative Models (5/17)

Find the **w** that minimizes

$$E(\mathbf{w}) = -\ln p(\mathbf{t} \mid \mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}$$

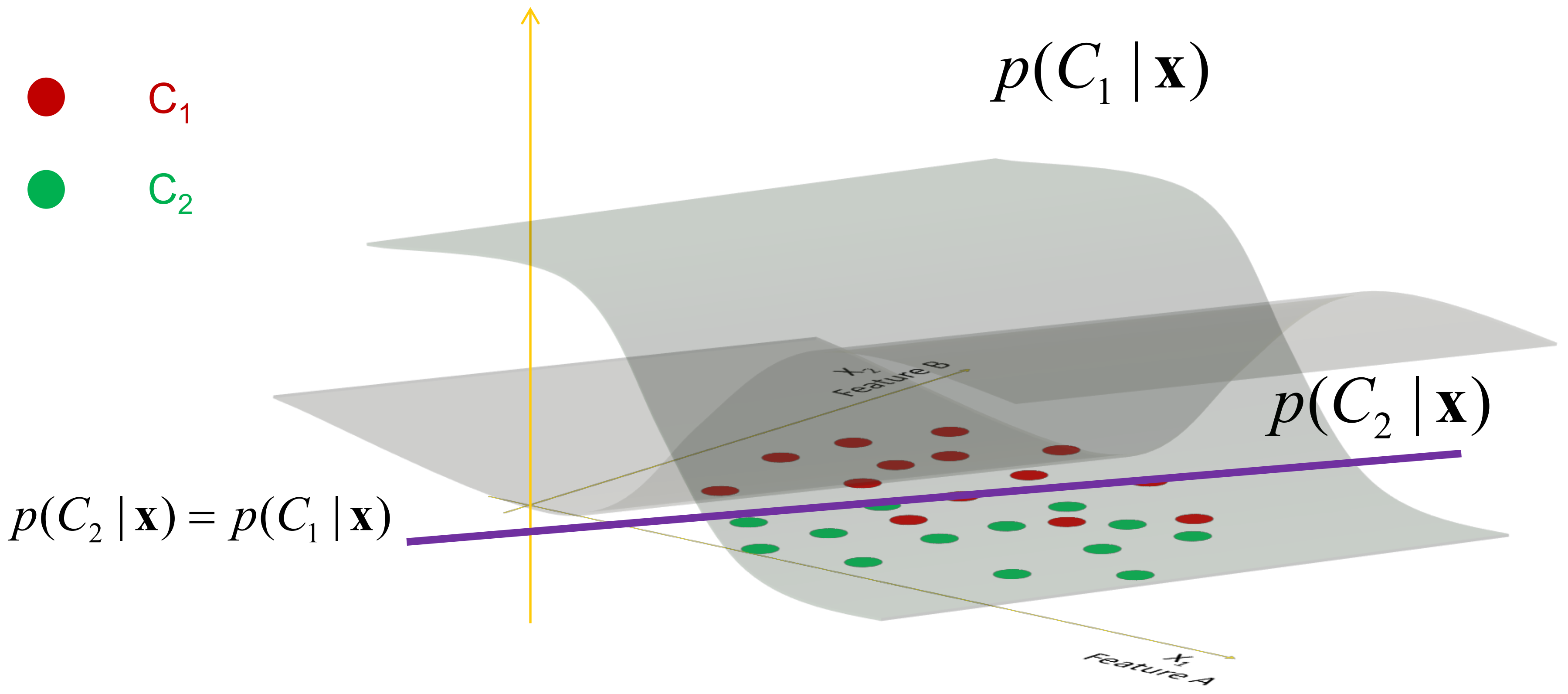
cross-entropy error function *for the binary classification problem*





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

Probabilistic Discriminative Models (6/17)



Probabilistic Discriminative Models (7/17)

Due to the nonlinearity of the logistic sigmoid function, there is no longer a closed-form solution for the optimal **w**.

Sequential Learning:

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

Newton-Raphson **method**

$$g(\theta) \approx g(\theta_0) + \frac{dg(\theta)}{d\theta}(\theta - \theta_0) \quad \Rightarrow \quad \theta_1 = \theta_0 - \frac{g(\theta_0)}{\left.\frac{dg(\theta)}{d\theta}\right|_{\theta=\theta_0}}$$

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \mathbf{H}^{-1}\nabla E(\mathbf{w})$$

$\mathbf{H} = \nabla\nabla E(\mathbf{w})$ Hessian matrix

Probabilistic Discriminative Models (8/17)

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

$$\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^{\mathrm{T}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi}$$

R: diagonal matrix with $R_{nn} = y_n(1- y_n)$.

Since $0 < y_n < 1$, it follows that $\mathbf{u}^{\mathrm{T}}\mathbf{H}\mathbf{u} > 0$ for an arbitrary vector **u**.

$\Rightarrow$ The error function is a convex function of **w** and there is a unique minimum.

Probabilistic Discriminative Models (9/17)

$$
\begin{aligned}
\mathbf{w}^{(\text{new})} &= \mathbf{w}^{(\text{old})} - (\Phi^{\mathrm{T}}\mathbf{R}\Phi)^{-1}\Phi^{\mathrm{T}}(\mathbf{y}-\mathbf{t}) \\
&= (\Phi^{\mathrm{T}}\mathbf{R}\Phi)^{-1}\left\{\Phi^{\mathrm{T}}\mathbf{R}\Phi\mathbf{w}^{(\text{old})} - \Phi^{\mathrm{T}}(\mathbf{y}-\mathbf{t})\right\} \\
&= (\Phi^{\mathrm{T}}\mathbf{R}\Phi)^{-1}\Phi^{\mathrm{T}}\mathbf{R}\mathbf{z}
\end{aligned}
$$

where $\mathbf{z} = \mathbf{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{R}^{-1}(\mathbf{y} - \mathbf{t})$.

Iterative reweighted least squares algorithm (***IRLS***)

The element of **R** can be interpreted as variances

$$E[t] = \sigma(\mathbf{x}) = y$$

$$var[t] = E[t^2] - E[t]^2 = \sigma(\mathbf{x}) - \sigma(\mathbf{x})^2 = y(1-y)$$

$$E[t] = 1 \times p(C_1 \mid \mathbf{x}) + 0 \times p(C_2 \mid \mathbf{x}) = p(C_1 \mid \mathbf{x}) = y$$

Probabilistic Discriminative Models (10/17)

- If there are K classes, we direct model $p(C_i|\mathbf{x}), i = 1, 2, . . , \mathrm{K},$ in terms of

$$p(C_i|\mathbf{x}) = y_k(\mathbf{x}) = \frac{\exp(a_k(\mathbf{x}))}{\sum_{j=1}^{K} \exp(a_j(\mathbf{x}))} \quad \text{where } a_k(\mathbf{x}) = \mathbf{w}_k^T \boldsymbol{\phi}(\mathbf{x})$$

where $\boldsymbol{\phi}(\mathbf{x}) = [\phi_0(\mathbf{x})\ \phi_1(\mathbf{x})\ \ldots\ \phi_{M-1}(\mathbf{x})]^T$

and $\mathbf{w}_k = \left[w_{k,0}\ \ w_{k,1}\ \ldots w_{k,M-1}\right]^T$

✓ We collect the set of training data $\mathbf{D} \equiv \{(\mathbf{x}_1, \boldsymbol{t}_1), (\mathbf{x}_2, \boldsymbol{t}_2), \ldots, (\mathbf{x}_N, \boldsymbol{t}_N)\}$, where $\boldsymbol{t}_n$'s are represented in the 1-of-K format.

Probabilistic Discriminative Models (11/17)

✓ Based on the training data D and the model $p(C_i|\mathbf{x})$ for each class, we form the likelihood function

$$p(D|\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_\mathrm{K}) = \prod_{n=1}^{N} p(C_1|\mathbf{x}_n; \mathbf{w}_1)^{t_{n1}}\, p(C_2|\mathbf{x}_n; \mathbf{w}_2)^{t_{n2}} \ldots p(C_\mathrm{K}|\mathbf{x}_n, ; \mathbf{w}_\mathrm{K})^{t_{nK}}$$

$$= \prod_{n=1}^{N} \prod_{k=1}^{K} y_k(\mathbf{x}_n; \mathbf{w}_k)^{t_{nk}}$$

$$\Rightarrow \quad E(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_\mathrm{K}) = -\ln p(D|\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_\mathrm{K})$$

$$= -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_k(\mathbf{x}_n; \mathbf{w}_k)$$

Probabilistic Discriminative Models (12/17)

- ✓ Based on $E(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{\mathrm{K}})$, we compute $\nabla E(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{\mathrm{K}}) = \mathbf{0}$ to find the optimal set of the model parameters $\{\mathbf{w}_1^{ML}, \mathbf{w}_2^{ML}, \dots, \mathbf{w}_K^{ML}\}$.
- ✓ Based on $\{\mathbf{w}_1^{ML}, \mathbf{w}_2^{ML}, \dots, \mathbf{w}_K^{ML}\}$, we construct the discriminant models $p(C_i|\mathbf{x})$

  for i = 1, 2, …, K.

Probabilistic Discriminative Models (13/17)

- For multi-class logistic regression, we don't have a closed form for the solution of $\nabla E(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_\mathrm{K}) = 0$.
  $\Rightarrow$ Use Gradient Descent method or Newton-Raphson method to find the solution.

If we define $\boldsymbol{W} = [\, \boldsymbol{w}_1^T \quad \boldsymbol{w}_2^T \quad \dots \quad \boldsymbol{w}_K^T]^T$

$$\boldsymbol{W}^{(new)} = \boldsymbol{W}^{(old)} - \eta \nabla E(\boldsymbol{W}^{(old)})$$ **Gradient Descent**

$$\boldsymbol{W}^{(new)} = \boldsymbol{W}^{(old)} - \mathbf{H}^{-1} \nabla E(\boldsymbol{W}^{(old)})$$ **Newton-Raphson**

where $\mathbf{H} = \nabla\nabla E(\boldsymbol{W})$

Probabilistic Discriminative Models (14/17)

- If there are K classes and we choose M basis functions, then $\boldsymbol{w}_k's$ are $M \times 1$ vectors and $\boldsymbol{W}$ is a $KM \times 1$ vector.
- $\boldsymbol{\nabla}E(\boldsymbol{W})$ is a $KM \times 1$ vector defined as following

$$\boldsymbol{\nabla}E(\boldsymbol{W}) = \left[(\boldsymbol{\nabla}_{\boldsymbol{w}_1}E(\boldsymbol{W}))^T \quad (\boldsymbol{\nabla}_{\boldsymbol{w}_2}E(\boldsymbol{W}))^T \quad \ldots \quad (\boldsymbol{\nabla}_{\boldsymbol{w}_K}E(\boldsymbol{W}))^T \right]^T$$

where $\boldsymbol{\nabla}_{\boldsymbol{w}_j}E(\boldsymbol{W}) = \sum_{n=1}^{N}(y_j(\mathbf{x}_n; \mathbf{w}_j) - t_{nj})\boldsymbol{\phi}(\mathbf{x}_n)$

- $\mathbf{H} = \nabla\nabla E(\boldsymbol{W})$ is a $KM \times KM$ matrix which can be decomposed into $K \times K$ blocks, with each block being an $M \times M$ matrix defined as

$$\boldsymbol{\nabla}_{\boldsymbol{w}_k}\boldsymbol{\nabla}_{\boldsymbol{w}_j}E(\boldsymbol{W}) = \sum_{n=1}^{N} y_k(\mathbf{x}_n; \mathbf{w}_k)(\mathbf{I}_{kj} - y_j(\mathbf{x}_n; \mathbf{w}_j))\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T$$

Probabilistic Discriminative Models (15/17)

$$\nabla E(\boldsymbol{W}) = \begin{bmatrix} \nabla_{\boldsymbol{w}_1} E(\boldsymbol{W}) \\ \nabla_{\boldsymbol{w}_2} E(\boldsymbol{W}) \\ \vdots \\ \nabla_{\boldsymbol{w}_K} E(\boldsymbol{W}) \end{bmatrix} \begin{array}{l} \}\ \text{M} \\ \}\ \text{M} \\ \\ \}\ \text{M} \end{array} \Bigg\}\ \text{KM}$$

$M \times M$ matrix

$KM \times KM$ matrix

$$\nabla\nabla E(\boldsymbol{W}) = \begin{bmatrix} \nabla_{\boldsymbol{w}_1}\nabla_{\boldsymbol{w}_1} E(\boldsymbol{W}) & \nabla_{\boldsymbol{w}_1}\nabla_{\boldsymbol{w}_2} E(\boldsymbol{W}) & \cdots & \nabla_{\boldsymbol{w}_1}\nabla_{\boldsymbol{w}_K} E(\boldsymbol{W}) \\ \nabla_{\boldsymbol{w}_2}\nabla_{\boldsymbol{w}_1} E(\boldsymbol{W}) & \nabla_{\boldsymbol{w}_2}\nabla_{\boldsymbol{w}_2} E(\boldsymbol{W}) & \cdots & \nabla_{\boldsymbol{w}_2}\nabla_{\boldsymbol{w}_K} E(\boldsymbol{W}) \\ \vdots & \vdots & \ddots & \vdots \\ \nabla_{\boldsymbol{w}_K}\nabla_{\boldsymbol{w}_1} E(\boldsymbol{W}) & \nabla_{\boldsymbol{w}_K}\nabla_{\boldsymbol{w}_2} E(\boldsymbol{W}) & \cdots & \nabla_{\boldsymbol{w}_K}\nabla_{\boldsymbol{w}_K} E(\boldsymbol{W}) \end{bmatrix}$$

Probabilistic Discriminative Models (16/17)

- Since $y_k(\mathbf{x}) = \frac{\exp(a_k(\mathbf{x}))}{\sum_{j=1}^{K} \exp(a_j(\mathbf{x}))} = \frac{\exp(a_k(\mathbf{x})+c)}{\sum_{j=1}^{K} \exp(a_j(\mathbf{x})+c)}$ for an arbitrary constant *c*, there are infinite solutions of the optimal $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_\mathrm{K}\}$.

  $\Rightarrow$ The iterative method may never converge!

  $\Rightarrow$ You may check the value of $E(\boldsymbol{W})$ and terminate the iterations when $E(\boldsymbol{W})$ has reached a stable value.

Probabilistic Discriminative Models (17/17)

- Alternative Way:

$$y_k(\mathbf{x}) = \frac{\exp(a_k(\mathbf{x}))}{\sum_{j=1}^{K} \exp(a_j(\mathbf{x}))} = \frac{\exp(a_k(\mathbf{x}) - a_1(\mathbf{x}))}{\sum_{j=1}^{K} \exp(a_j(\mathbf{x}) - a_1(\mathbf{x}))}$$

$$= \begin{cases} \dfrac{1}{1 + \sum_{j=2}^{K} \exp(\tilde{a}_j(\mathbf{x}))} & \text{if } k = 1 \\ \dfrac{\exp(\tilde{a}_j(\mathbf{x}))}{1 + \sum_{j=2}^{K} \exp(\tilde{a}_j(\mathbf{x}))} & \text{if } k = 2, \ldots, K \end{cases}$$

$\Rightarrow$ We force $\tilde{a}_1(\mathbf{x}) = 0$ and only define (K-1) logit functions

Regression versus Logistic Regression (1/5)

Regression

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

Gaussian Distribution

$$\Longrightarrow \quad E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = \tfrac{1}{2}\textstyle\sum_{n=1}^{N}\{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x_n})\}^2 + \mathbf{w}\text{-independent terms}$$

$$\Longrightarrow \quad \nabla E(\mathbf{w}) = -\sum_{n=1}^{N}\{t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x_n})\}\boldsymbol{\phi}(\mathbf{x_n}) = \sum_{n=1}^{N}\{\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x_n}) - t_n\}\boldsymbol{\phi}(\mathbf{x_n})$$

$$= \sum_{n=1}^{N}\{y_n - t_n\}\boldsymbol{\phi}(\mathbf{x_n})$$

Regression versus Logistic Regression (2/5)

Binary Logistic Regression

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} {y_n}^{t_n} (1 - y_n)^{1-t_n}$$

Bernoulli Distribution

$$\Longrightarrow \quad E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

$$\Longrightarrow \quad \nabla E(\mathbf{w}) = \sum_{n=1}^{N} \{\mathbf{w}^T \phi(\mathbf{x_n}) - t_n\} \boldsymbol{\phi}(\mathbf{x_n}) = \sum_{n=1}^{N} \{y_n - t_n\} \boldsymbol{\phi}(\mathbf{x_n})$$

Regression versus Logistic Regression (3/5)

Multi-class Logistic Regression

$$p(D|\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_\mathrm{K}) = \prod_{n=1}^{N}\prod_{k=1}^{K} y_k(\mathbf{x}_n; \mathbf{w}_k)^{t_{nk}}$$

Multinomial Distribution

$$\Rightarrow \quad E(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_\mathrm{K}) = -\ln p(D|\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_\mathrm{K}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_k(\mathbf{x}_n; \mathbf{w}_k)$$

$$\Rightarrow \quad \nabla E(\boldsymbol{W}) = \left[\, (\nabla_{\boldsymbol{w}_1} E(\boldsymbol{W}))^T \quad (\nabla_{\boldsymbol{w}_2} E(\boldsymbol{W}))^T \quad \dots \quad (\nabla_{\boldsymbol{w}_K} E(\boldsymbol{W}))^T \right]^T$$

where $\nabla_{\boldsymbol{w}_j} E(\boldsymbol{W}) = \sum_{n=1}^{N} \left(y_j\left(\mathbf{x}_n; \mathbf{w}_j\right) - t_{nj}\right)\boldsymbol{\phi}(\mathbf{x}_n)$

Regression versus Logistic Regression (4/5)

Exponential Family $$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$

Gaussian $$p(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\{-\frac{1}{2\sigma^2}(x-\mu)^2\} = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\{-\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}\mu^2\}$$

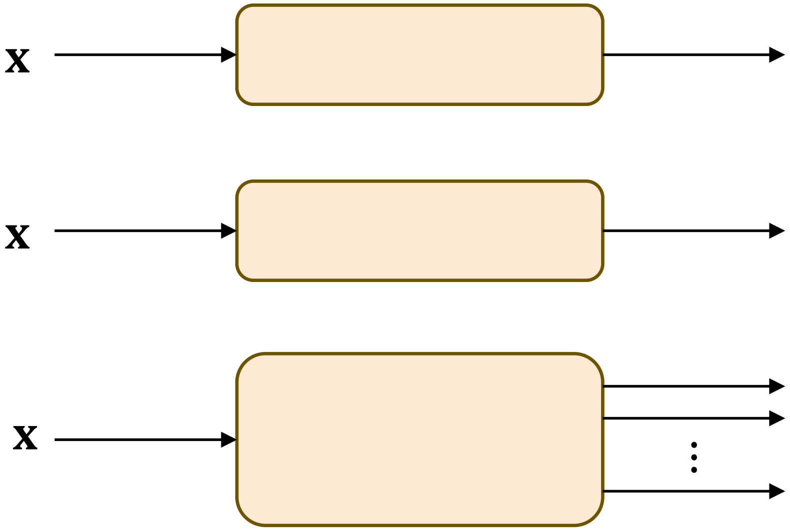
Bernoulli $$p(x|\mu) = \mu^x(1-\mu)^{1-x}$$

If we define $\eta = \ln(\frac{\mu}{1-\mu})$ or $\mu = \sigma(\eta) = \frac{1}{1+e^{-\eta}}$, we have $p(x|\eta) = \sigma(-\eta)\exp(\eta x)$

Multinomial $$\boldsymbol{p}(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{M}{\mu_k}^{x_k}$$

If we define $\eta_k = \ln(\frac{\mu_k}{1-\sum_j \mu_j})$ or $\mu_k = \frac{e^{\eta_k}}{1+\sum_j e^{\eta_j}}$, we have $p(\mathbf{x}|\boldsymbol{\eta}) = (1+\sum_{k=1}^{M-1} e^{\eta_k})^{-1}\exp(\boldsymbol{\eta}^T\mathbf{x})$

Regression versus Logistic Regression (5/5)



Regression

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

Binary Classification

$$y(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})}}$$

Multi-class Classification

$$y_k(\mathbf{x}, \mathbf{W}) = p(C_i|\mathbf{x}) = \frac{\exp(a_k(\mathbf{x}))}{\sum_{j=1}^{K} \exp(a_j(\mathbf{x}))}$$

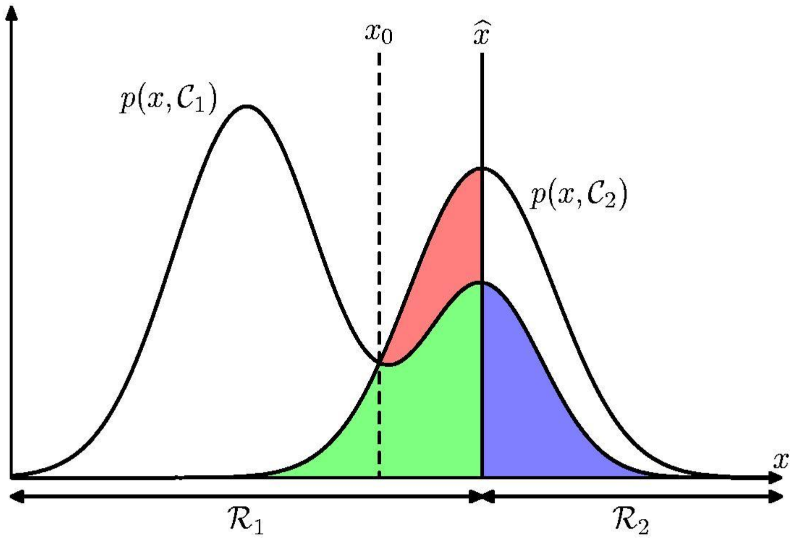
where $a_k(\mathbf{x}) = \mathbf{w}_k^T \boldsymbol{\phi}(\mathbf{x})$

$$E_n(\mathbf{w}) = \begin{cases} \frac{1}{2}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2 & \text{regression} \\ -\{t_n \ln y\,(\mathbf{x}_n, \mathbf{w}) + (1 - t_n) \ln(1 - y(\mathbf{x}_n, \mathbf{w}))\} & \text{binary classification} \\ -\sum_{k=1}^{K} t_{kn} \ln y_k\,(\mathbf{x}_n, \mathbf{W}) & \text{multi}-\text{calss classification} \end{cases}$$

Decision Theory

- Inference Step: Determine either $p(t|\mathbf{x})$ or $p(\mathbf{x},t)$
- Decision Step: For given $\mathbf{x}$, determine optimal t.

Minimizing the Misclassification Rate (Classification)



R_k: decision region of Class k.

$$p(\mathbf{x}, C_k) = p(C_k \mid \mathbf{x}) p(\mathbf{x})$$

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

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

Minimizing the Expected Loss (Classification)

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k)\, \mathrm{d}\mathbf{x} = \sum_j \int_{R_j} \sum_k L_{kj}\, p(C_k \mid \mathbf{x})\, p(\mathbf{x}) d\mathbf{x}$$

e.g.

$$\begin{array}{c c} & \begin{array}{cc} \text{cancer} & \text{normal} \end{array} \\ \begin{array}{c} \text{cancer} \\ \text{normal} \end{array} & \left(\begin{array}{cc} 0 & 1000 \\ 1 & 0 \end{array} \right) \end{array}$$

Regions R_j are chosen to minimize

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

Rejection Option

Reject the input **x** when the largest of the posterior probabilities $p(C_k|\mathbf{x}) \leq \theta$.