1

Machine Learning

Chapter 4: Linear Models for Classification

林嘉文 (Chia-Wen Lin)
清華大學電機系
cwlin@ee.nthu.edu.tw

2

Discriminant Functions

- Linear discriminant function is a linear combination of the components of $\mathbf{x}$
- $y(\mathbf{x}) = \mathbf{w}^{\top}\mathbf{x} + w_0$
- The two-category case

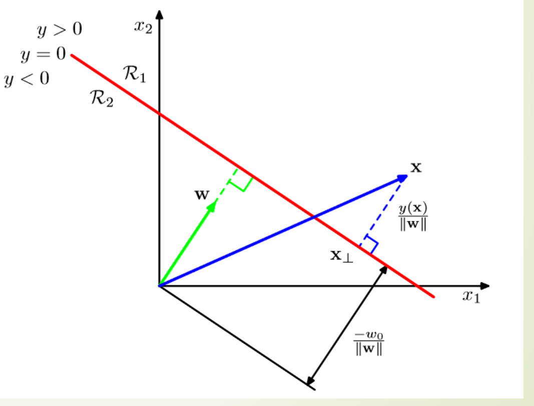
Decide C_1 if $y(\mathbf{x}) > 0$ or $y(\mathbf{x}) = \mathbf{w}^{\top}\mathbf{x} > -w_0$
C_2 if $y(\mathbf{x}) < 0$ or $y(\mathbf{x}) = \mathbf{w}^{\top}\mathbf{x} < -w_0$

3/28/2021

3

Discriminant Functions

- $\mathbf{w}$ is normal to any vector lying in the hyperplane
- Each $\mathbf{x}$ in the space can be expressed as
  - $\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$



3/28/2021

4

Discriminant Functions

- $\mathbf{w}$ is normal to any vector lying in the hyperplane
- Each $\mathbf{x}$ in the space can be expressed as
  - $\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$
  - r is positive if $\mathbf{x}$ is on the positive side, and is negative if $\mathbf{x}$ is on the negative side
  - $y(\mathbf{x}) = \mathbf{w}^\top\mathbf{x} + w_0 = r\|\mathbf{w}\|$ or $r = y(\mathbf{x})/\|\mathbf{w}\|$ ($y(\mathbf{x}_\perp) = 0$)
  - The distance from the origin to H is given by $\frac{w_0}{\|\mathbf{w}\|}$
    - If $w_0 > 0$, the original is on the positive side of H
    - If $w_0 < 0$, it is on the negative side of H
- Linear discriminant function divides the feature space by a hyperplane decision surface. The orientation is determined by $\mathbf{w}$ and location determined by w_0

3/28/2021

5

Multi-Class Discriminant Functions

- Solve it by K two-class problems
  - Separate points assigned to C_i from those not assigned to C_i.
- Solve it by $K(K-1)/2$ two-class problems
  - Separate every pair of classes.
- Define K linear discriminant functions
  - Assign $\mathbf{x}$ to ω_i if $y_i(\mathbf{x}) > y_j(\mathbf{x})$ for all $j \neq i$

$$y_i(\mathbf{x}) = \mathbf{w}_i^{\mathsf{T}}\mathbf{x} + w_{i0} \quad i = 1, \cdots, K$$

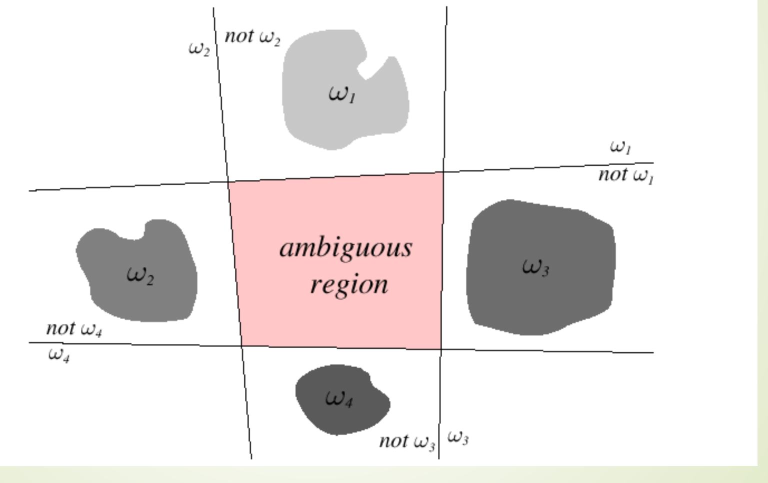
$$y_i(\mathbf{x}) = y_j(\mathbf{x}) \Rightarrow \left(\boldsymbol{w}_i - \boldsymbol{w}_j\right)^{\mathsf{T}}\mathbf{x} + \left(w_{i0} - w_{j0}\right) = \mathbf{0}$$

  - $\mathbf{w}_i - \mathbf{w}_j$ is normal to H_{ij} and the signed distance from $\mathbf{x}$ to H_{ij} is given by $\left(y_i - y_j\right)/\left\|\mathbf{w}_i - \mathbf{w}_j\right\|$

6

Multi-Class Discriminant Functions

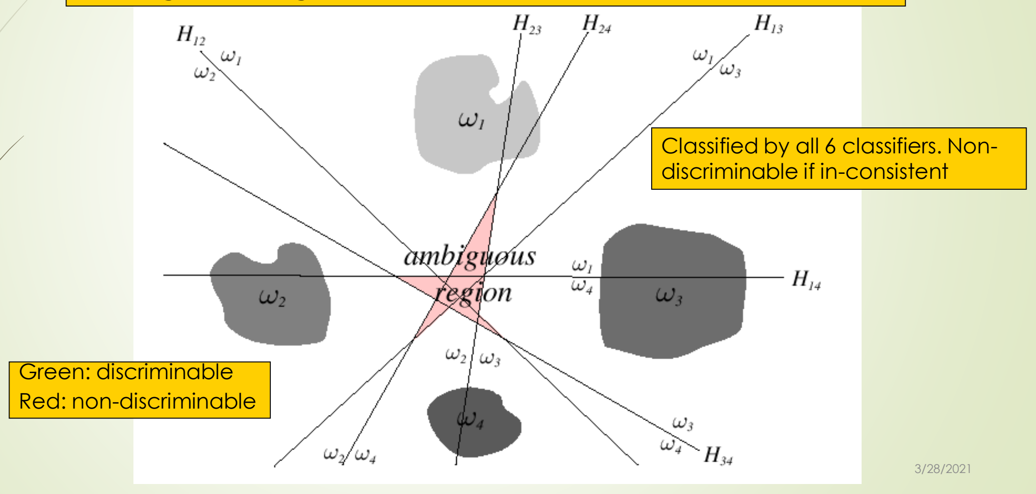
Ambigious region in K two-class problems



7

Multi-Class Discriminant Functions

Ambiguous region in $K(K-1)/2$ two-class problems

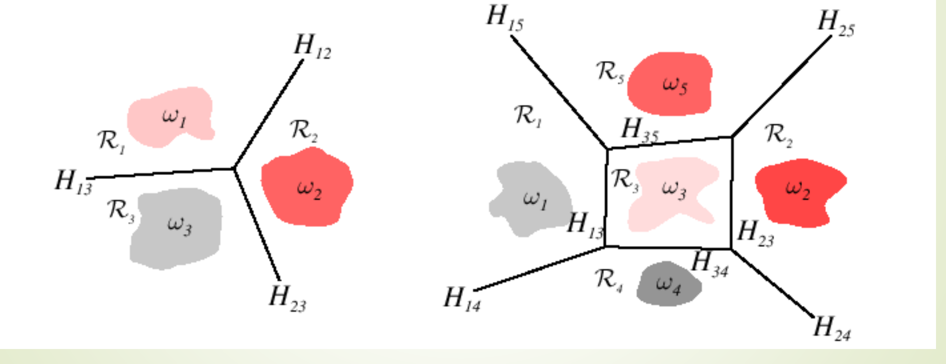


3/28/2021

8

Multi-Class Discriminant Functions

Use K linear discriminant functions



3/28/2021

9

Least Squares for Classification

- For a K-class classification problems, each class C_k is described by its own linear model:

$$y_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} + w_{k0} \quad i = 1, \cdots, K$$

- We can group these together:

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^\top \tilde{\mathbf{x}}, \quad \widetilde{\mathbf{w}}_k = \left(w_{k0}, \mathbf{w}_k^\top\right)^\top, \;\; \tilde{\mathbf{x}} = (1, \mathbf{x}^\top)^\top$$

- Let $\mathbf{T} = [\mathbf{t}_1 \mathbf{t}_2 \cdots \mathbf{t}_N]^\top$, $\widetilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1 \tilde{\mathbf{x}}_2 \cdots \tilde{\mathbf{x}}_N]^\top$, define the following sum-of-squares error function

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

- Setting the derivative with respect to $\widetilde{\mathbf{W}}$ to zero, we obtain

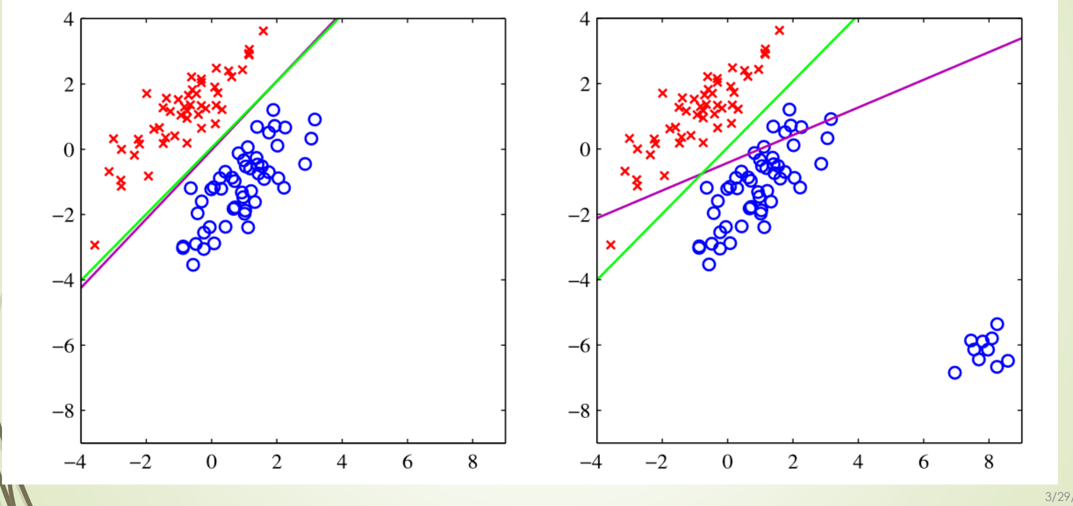
$$\widetilde{\mathbf{W}} = \left(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}}\right)^{-1} \widetilde{\mathbf{X}}^\top \mathbf{T} = \widetilde{\mathbf{X}}^\dagger \mathbf{T}$$

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^\top \tilde{\mathbf{x}} = \mathbf{T}^\top \left(\widetilde{\mathbf{X}}^\dagger\right)^\top \widetilde{\mathbf{X}}^\dagger \tilde{\mathbf{x}}$$

3/29/2021

10

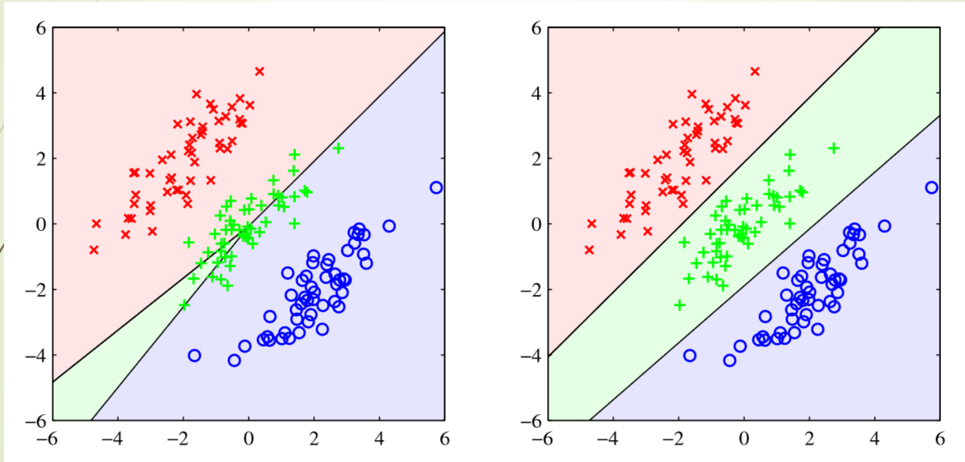
Least Squares for Classification



3/29/2021

11

Least Squares for Classification



Least Squares

Logistic Regression

3/29/2021

12

Component Analysis and Discriminants

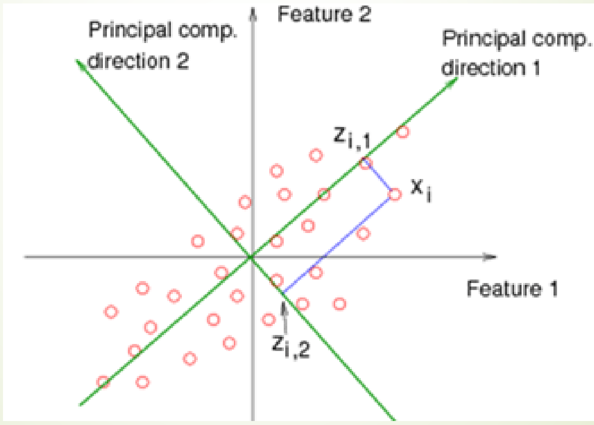
- Reduce dimensionality by combining features
- Project high-dimensional data onto a lower dimensional space.
  - Principal component analysis (PCA)
    - Seek projection to best **represent** the data in least-squares error sense.
  - Fisher discriminant analysis
    - Seek projection to best **separate** the data in least-squares error sense.

3/28/2021

13

Component Analysis and Discriminants

Principal axis of a figure

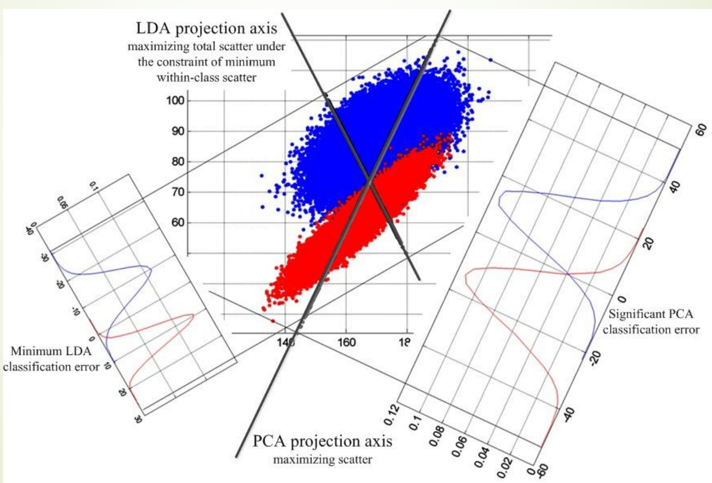


3/28/2021

14

Component Analysis and Discriminants

1-D projection of two overlapping classes



3/28/2021

16

Zero-dimensional Representation

$$J_0(\mathbf{x}_0) = \sum_{k=1}^{N} \|(\mathbf{x}_0 - \mathbf{m}) - (\mathbf{x}_k - \mathbf{m})\|^2$$
$$= \sum_{k=1}^{N} \|\mathbf{x}_0 - \mathbf{m}\|^2 - 2\sum_{k=1}^{N} (\mathbf{x}_0 - \mathbf{m})^\top (\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^{N} \|\mathbf{x}_k - \mathbf{m}\|^2$$
$$= \sum_{k=1}^{N} \|\mathbf{x}_0 - \mathbf{m}\|^2 - 2(\mathbf{x}_0 - \mathbf{m})^\top \sum_{k=1}^{N} (\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^{N} \|\mathbf{x}_k - \mathbf{m}\|^2$$
$$= \sum_{k=1}^{N} \|\mathbf{x}_0 - \mathbf{m}\|^2 + \sum_{k=1}^{N} \|\mathbf{x}_k - \mathbf{m}\|^2$$

$\Rightarrow \mathbf{x}_0 = \mathbf{m}$ ($\because$ 2nd term is independent of $\mathbf{x}_0$)

17

One-Dimensional Representation

- Project a set of N d-D samples $\mathbf{x}_1, \cdots, \mathbf{x}_N$ onto a line running through $\mathbf{m}$.
  - $\mathbf{x} = \mathbf{m} + a\mathbf{e}$
    - $\mathbf{e}$ is a unit vector in the direction of the line.
  - Squared-error criterion function
$$J_1(a_1, \cdots, a_N, \mathbf{e}) = \sum_{k=1}^{N} \|(\mathbf{m} + a_k\mathbf{e}) - \mathbf{x}_k\|^2$$
  - Solution of $\mathbf{y}_k$
$$a_k = \mathbf{e}^\top (\mathbf{x}_k - \mathbf{m})$$

18

One-Dimensional Representation

$$J_1(a_1, \cdots, a_N, \mathbf{e}) = \sum_{k=1}^{N} \|(\mathbf{m} + a_k \mathbf{e}) - \mathbf{x}_k\|^2 = \sum_{k=1}^{N} \|a_k \mathbf{e} - (\mathbf{x}_k - \mathbf{m})\|^2$$

$$= \sum_{k=1}^{N} a_k^2 \|\mathbf{e}\|^2 - 2 \sum_{k=1}^{N} a_k \mathbf{e}^\top (\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^{\top} \|\mathbf{x}_k - \mathbf{m}\|^2$$

$$\frac{\partial J_1(a_1, \cdots, a_N, \mathbf{e})}{\partial a_k} = 2a_k \|\mathbf{e}\|^2 - 2\mathbf{e}^\top (\mathbf{x}_k - \mathbf{m}) = 0$$

$$\Rightarrow a_k = \mathbf{e}^\top (\mathbf{x}_k - \mathbf{m}) \qquad (\because \|\mathbf{e}\|^2 = 1)$$

3/28/2021

19

Scatter Matrix & The Projection Line

- Scatter matrix of a data set

$$\mathbf{S} = \sum_{k=1}^{N} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^\top$$

- Finding the best direction $\mathbf{e}$ for projection in the minimum squared-error sense
  - $\mathbf{e}$ is the eigenvector of $\mathbf{S}$ with the largest eigenvalue.

3/28/2021

20

Derivation of Principal Axis $\mathbf{e}$

- Squared-error criterion function

$$J_1(\mathbf{e}) = \sum_{k=1}^{N} a_k^2 - 2\sum_{k=1}^{N} a_k^2 + \sum_{k=1}^{N} \|\mathbf{x}_k - \mathbf{m}\|^2 = -\sum_{k=1}^{N} [\mathbf{e}^\top(\mathbf{x}_k - \mathbf{m})]^2 + \sum_{k=1}^{N} \|\mathbf{x}_k - \mathbf{m}\|^2$$

$$= -\sum_{k=1}^{N} \mathbf{e}^\top(\mathbf{x}_k - \mathbf{m})\mathbf{e}^\top(\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^{N} \|\mathbf{x}_k - \mathbf{m}\|^2$$

$$= -\sum_{k=1}^{N} \mathbf{e}^\top(\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^\top\mathbf{e} + \sum_{k=1}^{N} \|\mathbf{x}_k - \mathbf{m}\|^2 \qquad a_k = \mathbf{e}^t(\mathbf{x}_k - \mathbf{m})$$

$$= -\mathbf{e}^\top\mathbf{S}\mathbf{e} + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2$$

$\Rightarrow$ Maximize $\mathbf{e}^\top\mathbf{S}\mathbf{e}$

3/28/2021

21

Maximize $\mathbf{e}^\top\mathbf{S}\mathbf{e}$ by Lagrange Multipliers

- Maximize $\mathbf{e}^\top\mathbf{S}\mathbf{e}$
  subject to the constraint $\|\mathbf{e}\| = 1$
- λ: Lagrange multiplier to be determined
- Formula : Maximize $u = \mathbf{e}^\top\mathbf{S}\mathbf{e} - \lambda(\boldsymbol{e}^\top\boldsymbol{e} - 1)$

$$\Rightarrow \frac{\partial u}{\partial \mathbf{e}} = 2\mathbf{S}\mathbf{e} - 2\lambda\mathbf{e} = 0$$

$$\Rightarrow \mathbf{S}\mathbf{e} = \lambda\mathbf{e} \qquad \Rightarrow \mathbf{e}^\top\mathbf{S}\mathbf{e} = \lambda\mathbf{e}^\top\mathbf{e} = \lambda$$

- $\mathbf{e}$ must be an eigenvector of $\mathbf{S}$
- To maximize $\mathbf{e}^\top\mathbf{S}\mathbf{e}$, the eigenvector with **the largest eigenvalue** should be selected.

3/28/2021

22

Maximize $\mathbf{e}^t\mathbf{S}\mathbf{e}$ by Lagrange Multipliers

$$\frac{\partial}{\partial \mathbf{e}}\{\mathbf{e}^\top\mathbf{S}\mathbf{e}-\lambda\mathbf{e}^\top\mathbf{e}\}=\begin{pmatrix}\frac{\partial}{\partial e_1}\\ \frac{\partial}{\partial e_2}\end{pmatrix}\left\{(e_1\quad e_2)\begin{pmatrix}s_{11} & s_{12}\\ s_{12} & s_{22}\end{pmatrix}\begin{pmatrix}e_1\\ e_2\end{pmatrix}-\lambda(e_1\quad e_2)\begin{pmatrix}e_1\\ e_2\end{pmatrix}\right\}$$

$$=\begin{pmatrix}\frac{\partial}{\partial e_1}\\ \frac{\partial}{\partial e_2}\end{pmatrix}\left\{\left(s_{11}e_1^2+2s_{12}e_1e_2+s_{22}e_2^2\right)-\lambda\left(e_1^2+e_2^2\right)\right\}$$

$$=\begin{pmatrix}2s_{11}e_1+2s_{12}e_2\\ 2s_{12}e_1+2s_{22}e_2\end{pmatrix}-2\lambda\begin{pmatrix}e_1\\ e_2\end{pmatrix}=2\begin{pmatrix}s_{11} & s_{12}\\ s_{12} & s_{22}\end{pmatrix}\begin{pmatrix}e_1\\ e_2\end{pmatrix}-2\lambda\begin{pmatrix}e_1\\ e_2\end{pmatrix}=2\mathbf{S}\mathbf{e}-2\lambda\mathbf{e}$$

23

Lagrange Optimization

- Seek the position $\mathbf{x}_0$ of an extremum of $f(\mathbf{x})$, subject to the constraint $g(\mathbf{x})=0$.
- Form the Lagrange function:

$$L(\mathbf{x},\lambda)=f(\mathbf{x})+\lambda g(\mathbf{x})$$

  - λ is the Lagrange multiplier to be determined
- Convert the constrained optimization into an unconstrained problem.
- $\frac{\partial L(\mathbf{x},\lambda)}{\partial \mathbf{x}}=\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}+\lambda\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}=0$ and $\frac{\partial L}{\partial \lambda}=g(\mathbf{x})=0$
  - Solve the resulting equation for λ and $\mathbf{x}_0$.

24

d'-Dimensional Representation

- Project a set of N d-D samples $\mathbf{x}_1, \cdots, \mathbf{x}_N$ to d'-D space, where $d' \leq d$

$$\mathbf{x} = \mathbf{m} + \sum_{i=1}^{d'} a_i \mathbf{e}_i$$

- Squared-error criterion function

$$J_{d'} = \sum_{k=1}^{N} \left\| \left(\mathbf{m} + \sum_{i=1}^{d'} a_{ki} \mathbf{e}_i \right) - \mathbf{x}_k \right\|^2$$

- Solution of a_{ki}: $a_{ki} = \mathbf{e}_i^\top (\mathbf{x}_k - \mathbf{m})$

$$\mathbf{a}_k = (\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{d'})^\top (\mathbf{x}_k - \mathbf{m})$$

$$a_k = \mathbf{E}^\top (\mathbf{x}_k - \mathbf{m}) \qquad \mathbf{E} : d \times d'$$

3/28/2021

25

d'-Dimensional Representation

- Best $\mathbf{e}_i$
  - $\boldsymbol{e}_1, \cdots, \mathbf{e}_{d'}$ are the d' eigenvectors of $\mathbf{S}$ having the largest d' eigenvalues.

$$(\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{d'})^\top \mathbf{S} (\mathbf{e}_1 \quad \cdots \quad \mathbf{e}_{d'}) = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_{d'} \end{pmatrix} = \boldsymbol{\Lambda}_{d'}$$

  - All the d' eigenvectors, forming an eigen-matrix $\mathbf{E}$, make diagonalization of $\mathbf{S}$ matrix $\Rightarrow \mathbf{E}^\top \mathbf{S} \mathbf{E} = \boldsymbol{\Lambda}$ (decorrelation of $\mathbf{S}$)

3/28/2021

26

Fisher Linear Discriminant

- PCA seeks directions that are useful for representation, while discriminant analysis seeks directions that are most discriminative.
- One-dimensional projection
  - A set of N d-D samples $\mathbf{x}_1, \cdots, \mathbf{x}_N$
    - N_1 d-D samples in the subset D_1 labeled C_1
    - N_2 d-D samples in the subset D_2 labeled C_2
  - A linear combination of the components of $\mathbf{x}$

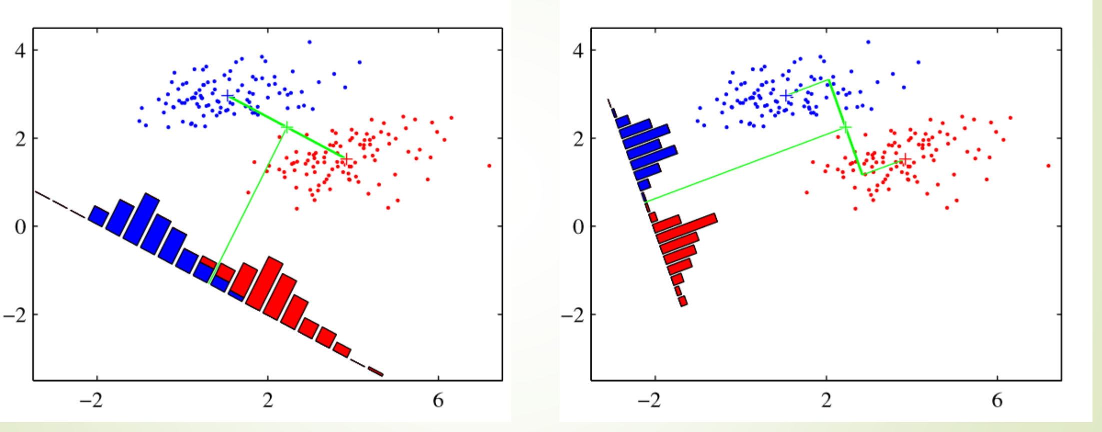
$$y_i = \mathbf{w}^\top \mathbf{x}_i \qquad \|\mathbf{w}\| = 1$$

  - A corresponding set of N projected samples $y_1, \cdots, y_N$ are divided into two subsets $\mathcal{Y}_1$ and $\mathcal{Y}_2$

3/28/2021

27

Fisher Linear Discriminant



3/28/2021

28

Fisher Linear Discriminant

- The Fisher linear discriminant finds the linear function $y = \mathbf{w}^\top \mathbf{x}$ to maximize the criterion function

$$\boxed{J(\mathbf{w}) = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2} = \frac{\text{between-class scatter}}{\text{within-class scatter}}}$$

$$s_i^2 = \sum_{y \in \mathcal{Y}_i} (y - m_i)^2 \text{ (Scatter)} \qquad \mathbf{m}_i = \frac{1}{N_i} \sum_{\mathbf{x} \in D_i} \mathbf{x} \quad \text{(Sample mean)}$$

$$m_i = \frac{1}{n_i} \sum_{y \in \mathcal{Y}_i} y = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{w}^\top \mathbf{x} = \mathbf{w}^\top \mathbf{m}_i \qquad |m_1 - m_2| = \left|\mathbf{w}^\top (\mathbf{m}_1 - \mathbf{m}_2)\right|$$

3/28/2021

29

Another Form of Separation Criterion

- Scatter matrices: $\mathbf{S}_i = \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^\top$
- Within-class scatter matrix: $\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$
- Between-class scatter matrix: $\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\top$

$$\Rightarrow \boxed{J(\mathbf{w}) = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^\top \mathbf{S}_B \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_W \mathbf{w}}}$$

$$s_i^2 = \sum_{\mathbf{x} \in D_i} (\mathbf{w}^\top \mathbf{x} - \mathbf{w}^\top \mathbf{m}_i)^2 = \sum_{\mathbf{x} \in D_i} \mathbf{w}^\top (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t \mathbf{w} = \mathbf{w}^\top \mathbf{S}_i \mathbf{w}$$

$$\Rightarrow s_1^2 + s_2^2 = \mathbf{w}^\top \mathbf{S}_1 \mathbf{w} + \mathbf{w}^\top \mathbf{S}_2 \mathbf{w} = \mathbf{w}^\top (\mathbf{S}_1 + \mathbf{S}_2) \mathbf{w} = \mathbf{w}^\top \mathbf{S}_W \mathbf{w}$$

$$|m_1 - m_2|^2 = (\mathbf{w}^\top \mathbf{m}_1 - \mathbf{w}^\top \mathbf{m}_2)^2 = \mathbf{w}^\top (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\top \mathbf{w} = \mathbf{w}^\top \mathbf{S}_B \mathbf{w}$$

3/28/2021

30

$\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w}$ for Maximizing $\frac{\mathbf{w}^\top\mathbf{S}_B\mathbf{w}}{\mathbf{w}^\top\mathbf{S}_W\mathbf{w}}$

- Derive $\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}}\left(\frac{\mathbf{w}^\top\mathbf{S}_B\mathbf{w}}{\mathbf{w}^\top\mathbf{S}_W\mathbf{w}}\right) = 0$

$$\Rightarrow \quad (2\mathbf{S}_B\mathbf{w})\cdot(\mathbf{w}^\top\mathbf{S}_W\mathbf{w}) - (2\mathbf{S}_W\mathbf{w})\cdot(\mathbf{w}^\top\mathbf{S}_B\mathbf{w}) = 0$$

$$\Rightarrow \quad \mathbf{S}_W\mathbf{w}\cdot(\mathbf{w}^\top\mathbf{S}_B\mathbf{w})(\mathbf{w}^\top\mathbf{S}_W\mathbf{w})^{-1} = \mathbf{S}_B\mathbf{w}$$

$$\Rightarrow \quad \lambda\mathbf{S}_W\mathbf{w} = \mathbf{S}_B\mathbf{w}$$

(Let $(\mathbf{w}^\top\mathbf{S}_B\mathbf{w})(\mathbf{w}^\top\mathbf{S}_B\mathbf{w})^{-1} = \lambda$, a scalar)

3/28/2021

31

Solution to Maximize $\frac{\mathbf{w}^\top\mathbf{S}_B\mathbf{w}}{\mathbf{w}^\top\mathbf{S}_W\mathbf{w}}$

- $\mathbf{w}$ must satisfy the condition of $\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w}$
- derivation of $\mathbf{x}$ (two methods)
  - $\mathbf{S}_W^{-1}$ is nonsingular $\Rightarrow$ $\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w} = \lambda\mathbf{w}$ (Eigenvalue problem)
  - $\mathbf{S}_B\mathbf{w} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\top\mathbf{w} = (\mathbf{m}_1 - \mathbf{m}_2)\cdot\underbrace{(m_1 - m_2)}_{\text{scalar}}$

$$\Rightarrow \mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2)\cdot\frac{m_1 - m_2}{\lambda}$$

$$\Rightarrow \mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \quad \text{(ignoring scale factor } \frac{m_1 - m_2}{\lambda}\text{)}$$

- Find the threshold, a point along the 1-D subspace separating the projected points, for classification

$\boxed{\mathbf{w}^t\mathbf{x} + w_0 = 0}$ $\quad$ $\boxed{\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2)}$

3/28/2021

32

Multiple Linear Discriminant

- d'-dimensional projection
  - A set of N d-D samples $\mathbf{x}_1, \cdots, \mathbf{x}_N$
    - N_1 d-D samples in the subset D_1 labeled C_1
    - …
    - N_K d-D samples in the subset D_K labeled C_K
  - A projection from a d-D space to d'-D space

$$y_i = \mathbf{w}_i^\top \mathbf{x}, \qquad i = 1, \cdots, d'$$
$$\mathbf{y} = \mathbf{W}^\top \mathbf{x}, \qquad \mathbf{W} = \left(\mathbf{w}_1 \cdots \mathbf{w}_{d'}\right)$$

  - A corresponding set of N d'-D samples $\mathbf{y}_1, \cdots, \mathbf{y}_N$ are divided into K subsets $\mathcal{Y}_1, \cdots, \mathcal{Y}_K$

3/28/2021

33

Scatter Matrices

- Within-class scatter matrix

$$\mathbf{S}_W = \sum_{i=1}^{K} \mathbf{S}_i, \qquad \mathbf{S}_i = \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^\top, \qquad \mathbf{m}_i = \frac{1}{N_i} \sum_{\mathbf{x} \in D_i} \mathbf{x}$$

- Total scatter matrix (as unclassified samples)

$$\mathbf{S}_T = \sum_{\mathbf{x}} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^\top, \ \mathbf{m} = \frac{1}{N} \sum_{\mathbf{x}} \mathbf{x} = \frac{1}{N} \sum_{i=1}^{K} N_i \mathbf{m}_i$$

- Between-class scatter matrix

$$\mathbf{S}_B = \sum_{i=1}^{K} N_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^\top$$

$$\boxed{\text{cf. } \mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\top}$$

3/28/2021

34

Scatter Matrices

$$\mathbf{S}_T = \sum_{\mathbf{x}} (\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^\top$$
$$= \sum_{i=1}^{K} \sum_{\mathbf{x} \in D_i} (\mathbf{x}-\mathbf{m}_i+\mathbf{m}_i-\mathbf{m})(\mathbf{x}-\mathbf{m}_i+\mathbf{m}_i-\mathbf{m})^\top$$
$$= \sum_{i=1}^{K} \sum_{\mathbf{x} \in D_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^\top + \sum_{i=1}^{K} \sum_{\mathbf{x} \in D_i} (\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^\top$$
$$= \mathbf{S}_W + \sum_{i=1}^{K} N_i(\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^\top = \mathbf{S}_W + \mathbf{S}_B$$

(this is why we change the definition of $\mathbf{S}_B$)

two−class case : $\mathbf{S}_B = \sum_{i=1}^{2} N_i\,(\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^\top = \frac{N_1 \times N_2}{N}(\mathbf{m}_1-\mathbf{m}_2)(\mathbf{m}_1-\mathbf{m}_2)^\top$

35

Multiple Linear Discriminant

- Find the linear function $\mathbf{y} = \mathbf{W}^\top\mathbf{x}$ to maximize the ratio between the between-class scatter and within-class scatter
- Using the determinant of the scatter matrix as the measure of scatter, we maximize

$$J(\mathbf{W}) = \frac{|\tilde{\mathbf{S}}_B|}{|\tilde{\mathbf{S}}_W|} = \frac{|\mathbf{W}^\top\mathbf{S}_B\mathbf{W}|}{|\mathbf{W}^\top\mathbf{S}_W\mathbf{W}|}$$

Determinant of matrix represents the product of eigenvalues or variances, measuring the square of the hyperellipsoidal volume

$$\tilde{\mathbf{S}}_B = \sum_{i=1}^{K} N_i\,(\tilde{\mathbf{m}}_i-\tilde{\mathbf{m}})(\tilde{\mathbf{m}}_i-\tilde{\mathbf{m}})^\top \Rightarrow \quad \tilde{\mathbf{S}}_B = \mathbf{W}^\top\mathbf{S}_B\mathbf{W}$$
$$\tilde{\mathbf{S}}_W = \sum_{i=1}^{K} \sum_{\mathbf{y} \in \mathcal{Y}_i} (\mathbf{y}-\tilde{\mathbf{m}}_i)(\mathbf{y}-\tilde{\mathbf{m}}_i)^\top \Rightarrow \quad \tilde{\mathbf{S}}_W = \mathbf{W}^\top\mathbf{S}_W\mathbf{W}$$
$$\tilde{\mathbf{m}} = \frac{1}{N}\sum_{i=1}^{K} N_i\tilde{\mathbf{m}}_i\,, \qquad \tilde{\mathbf{m}}_i = \frac{1}{N_i}\sum_{\mathbf{y} \in \mathcal{Y}_i} \mathbf{y}$$

3/28/2021

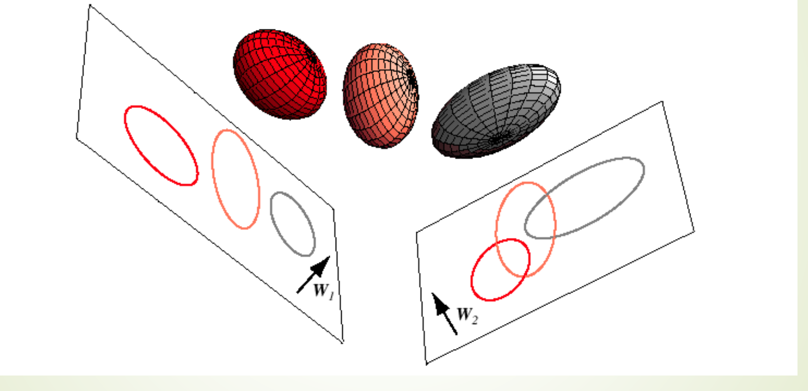
36

Optimizing $J(\mathbf{W})$

- $\mathbf{w}_i$ is the generalized eigenvectors corresponding to the largest eigenvalues in $\mathbf{S}_B\mathbf{w}_i = \lambda_i \mathbf{S}_W \mathbf{w}_i$
- If $\mathbf{S}_W$ is nonsingular, $\mathbf{w}_i$ is the eigenvectors corresponding to the largest eigenvalues in
$$\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w}_i = \lambda_i \mathbf{w}_i$$
- Instead, find eigenvalues as the roots of characteristic polynomial of $|\mathbf{S}_B - \lambda_i \mathbf{S}_W| = 0$, and solve $(\mathbf{S}_B - \lambda_i \mathbf{S}_W)\mathbf{w}_i = \mathbf{0}$
- Generally, the solution for $\mathbf{W}$ is not unique. The allowable transformations include rotating and scaling the axes in various ways (e.g., the planes in the figure shown in the next page have many possible axes) and leave $J(\mathbf{W})$ and classifier unchanged.

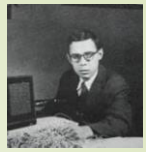
37

Optimizing $J(\mathbf{W})$



38

The Perceptron Algorithm



Frank Rosenblatt
(1928-1969)

- An example of a two-class linear discriminant model invented by Rosenblatt (1962). The input vector $\mathbf{x}$ is first transformed using a fixed nonlinear transformation to give a feature vector $\boldsymbol{\phi}(\mathbf{x})$, which is then used to construct a generalized linear model

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

where the **non-linear activation function** $f(\cdot)$ is

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

3/28/2021

39

The Perceptron Algorithm

- Error function for determining $\mathbf{w}$:
  - the total number of misclassified patterns: not appropriate for gradient based optimization
  - The perceptron criterion: to minimize

$$E_p(\mathbf{w}) = -\sum_{n \in \mathcal{M}} \mathbf{w}^\top \boldsymbol{\phi}_n t_n$$

where $\mathcal{M}$ denotes the set of all misclassified patterns, $t_n \in \{-1,1\}$ to make $\mathbf{w}^\top \boldsymbol{\phi}_n t_n > 0$

3/28/2021

40

The Perceptron Algorithm

- Applying the SGD algorithm to $E_p(\mathbf{w})$. The iterative update of $\mathbf{w}$ is given by :

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

where η is the learning rate parameter.

- The contribution to the error from a misclassified pattern will be reduced because

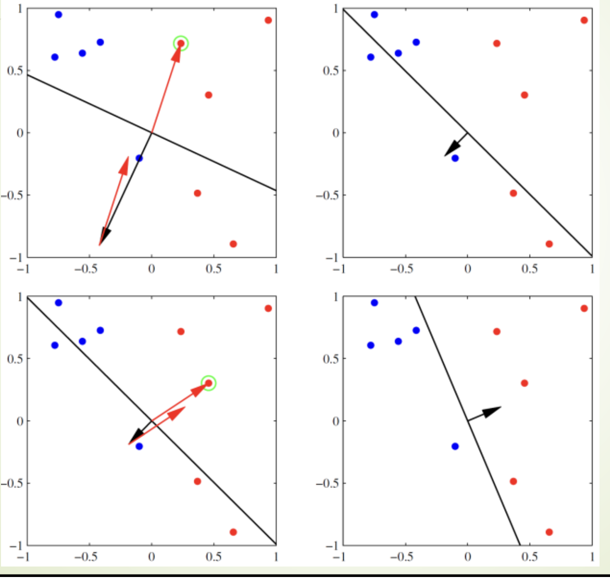
$$-\mathbf{w}^{(\tau+1)\mathrm{T}} \boldsymbol{\phi}_n t_n = -\mathbf{w}^{(\tau)\mathrm{T}} \boldsymbol{\phi}_n t_n - (\boldsymbol{\phi}_n t_n)^{\mathrm{T}} \boldsymbol{\phi}_n t_n < -\mathbf{w}^{(\tau)\mathrm{T}} \boldsymbol{\phi}_n t_n$$

where we set $\eta = 1$

3/28/2021

41

The Perceptron Algorithm



3/28/2021

42

The Perceptron Algorithm

- If there exists an exact solution (i.e., if the data set is linearly separable), then the perceptron learning algorithm is guaranteed to find an exact solution in a finite number of steps.

Mark 1 perceptron hardware for processing 20x20 image

3/28/2021

43

Probabilistic Generative Models

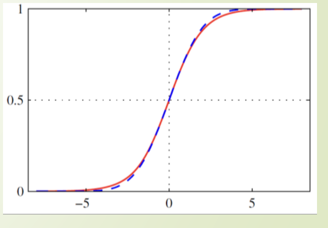
Consider first of all the case of two classes. The posterior probability for class C_1 can be written as:

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2)}$$

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

where

$$a = \ln \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)}$$

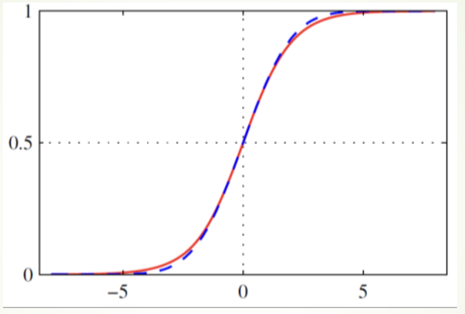


and $\sigma(a)$ is the logistic sigmoid function:

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

3/28/2021

Logistic Sigmoid Function



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

$$\sigma(-a) = 1 - \sigma(a)$$

The inverse of the logistic sigmoid is given by

$$a = \ln\left(\frac{\sigma}{1-\sigma}\right) = \ln \frac{p(C_1|\mathbf{x})}{p(C_2|\mathbf{x})}$$ **(logit function)**

Softmax Function

For the case of $K > 2$ classes, we have

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_j p(\mathbf{x}|C_j)p(C_j)} = \boxed{\frac{\exp(a_k)}{\sum_j \exp(a_j)}}$$ **(softmax function)**

$$a_k = \ln p(\mathbf{x}|C_k)p(C_k)$$

which is known as the **normalized exponential (softmax function)**, as it represents a smoothed version of the "max" function because if $a_k \geq a_j$ for all $j \neq k$, then $p(C_k|\mathbf{x}) \approx 1$, and $p(C_j|\mathbf{x}) \approx 0$.

46

Continuous Inputs

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

Consider first the case of two classes, we have

$$p(C_1|\mathbf{x}) = \sigma(\mathbf{w}^\top\mathbf{x} + w_0)$$

where we have defined

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$\mathbf{x}_0 = -\frac{1}{2}\boldsymbol{\mu}_1^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(C_1)}{p(C_2)}$$

3/28/2021

47

Likelihood vs. Posterior

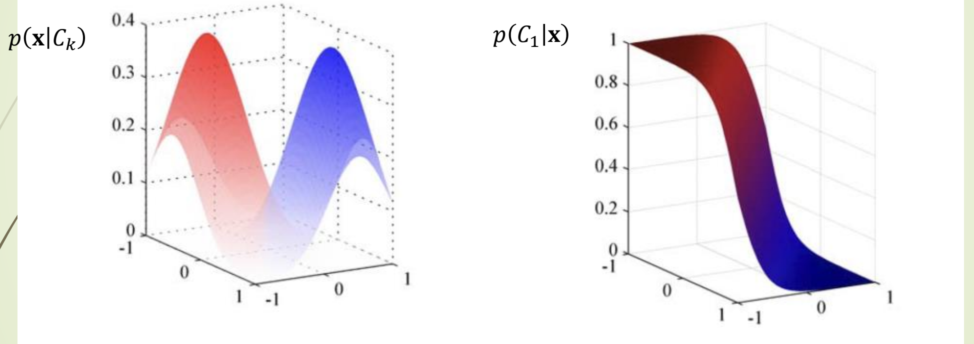


Figure 4.10 The left-hand plot shows the class-conditional densities for two classes, denoted red and blue. On the right is the corresponding posterior probability $p(\mathcal{C}_1|\mathbf{x})$, which is given by a logistic sigmoid of a linear function of $\mathbf{x}$. The surface in the right-hand plot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

3/28/2021

48

Decision Surface

$$p(C_k|\mathbf{x}) = \sigma(a_k(\mathbf{x})) = \sigma(\mathbf{w}_k^\top \mathbf{x} + w_{k0})$$

- $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$
  - Formulations (ignoring $|\boldsymbol{\Sigma}_k|$ and $(d/2)\ln 2\pi$)

$$a_k(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) + \ln p(C_k)$$
$$= \boxed{-\frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x}} + \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln P(C_k)$$

$$\Rightarrow a_k(\mathbf{x}) = \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(C_k)$$

$$\Rightarrow \quad a_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} + w_{k0}$$

$$\mathbf{w}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k, \qquad w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(C_k)$$

3/28/2021

49

Decision Surface

- For any two categories C_i and C_j
  - The decision surface is determined by $p(C_i|\mathbf{x}) = p(C_j|\mathbf{x})$

$$a(\mathbf{x}) = a_i(\mathbf{x}) - a_j(\mathbf{x}) = 0$$

$$\Rightarrow \quad \boldsymbol{\mu}_i^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_i^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln p(C_i) - \boldsymbol{\mu}_j^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} + \frac{1}{2}\boldsymbol{\mu}_j^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j - \ln p(C_j) = 0$$

$$\Rightarrow (\boldsymbol{\mu}_i^\top - \boldsymbol{\mu}_j^\top)\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}(\boldsymbol{\mu}_i^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i - \boldsymbol{\mu}_j^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j) + \frac{(\boldsymbol{\mu}_i^\top - \boldsymbol{\mu}_j^\top)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i^\top - \boldsymbol{\mu}_j^\top)}{(\boldsymbol{\mu}_i^\top - \boldsymbol{\mu}_j^\top)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i^\top - \boldsymbol{\mu}_j^\top)}\ln\frac{p(C_i)}{p(C_j)} = 0$$

$$\Rightarrow (\boldsymbol{\mu}_i^\top - \boldsymbol{\mu}_j^\top)\boldsymbol{\Sigma}^{-1}\left[\mathbf{x} - \frac{1}{2}(\boldsymbol{\mu}_i + \boldsymbol{\mu}_j) + \frac{(\boldsymbol{\mu}_i^\top - \boldsymbol{\mu}_j^\top)}{(\boldsymbol{\mu}_i^\top - \boldsymbol{\mu}_j^\top)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i^\top - \boldsymbol{\mu}_j^\top)}\ln\frac{p(C_i)}{p(C_j)}\right] = 0$$

Note: $\boldsymbol{\mu}_i^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i - \boldsymbol{\mu}_j^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j = (\boldsymbol{\mu}_i^\top - \boldsymbol{\mu}_j^\top)\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i + \boldsymbol{\mu}_j)$

3/28/2021

50

Decision Surface

$$a(\mathbf{x}) = \mathbf{w}^{\top}(\mathbf{x} - \mathbf{x}_0) = 0$$

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j),$$

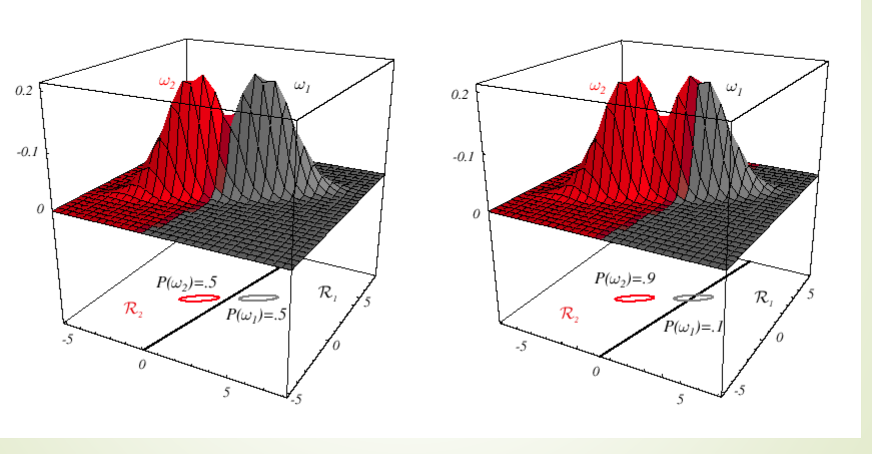
$$\mathbf{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_i + \boldsymbol{\mu}_j) - \frac{\ln[p(C_i)/p(C_j)]}{(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$$

- The hyperplane is generally not orthogonal to the line between two mean vectors
- For equal prior probabilities, $\mathbf{x}_0$ is halfway between two means.

3/28/2021

51

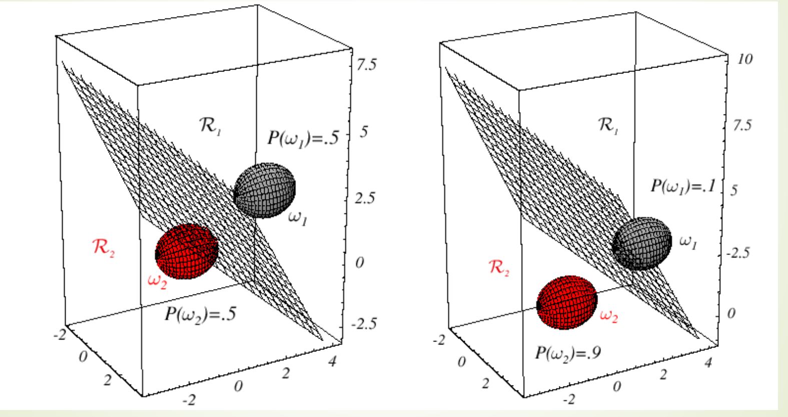
Decision Surface



3/28/2021

52

Decision Surface



3/28/2021

53

Decision Surface

- $\boldsymbol{\Sigma}_k$ = arbitrary
  - Formulations

$$a_k(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{\Sigma}_k| + \ln p(C_k)$$

$$\Rightarrow \quad a_k(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_k| + \ln p(C_k)$$

$$= -\frac{1}{2}\mathbf{x}^\top\boldsymbol{\Sigma}_k^{-1}\mathbf{x} + \boldsymbol{\mu}_k^\top\boldsymbol{\Sigma}_k^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_k^\top\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\ln|\boldsymbol{\Sigma}_k| + \ln p(C_k)$$

$$= \mathbf{x}^\top\mathbf{W}_k\mathbf{x} + \mathbf{w}_k^\top\mathbf{x} + w_{k0}$$

$$\boxed{\begin{aligned}\mathbf{W}_k &= -\frac{1}{2}\boldsymbol{\Sigma}_k^{-1}, \qquad \mathbf{w}_k = \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k,\\ w_{k0} &= -\frac{1}{2}\boldsymbol{\mu}_k^\top\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\ln|\boldsymbol{\Sigma}_k| + \ln p(C_k)\end{aligned}}$$

3/28/2021

54

Decision Surface (Hyperquadrics)

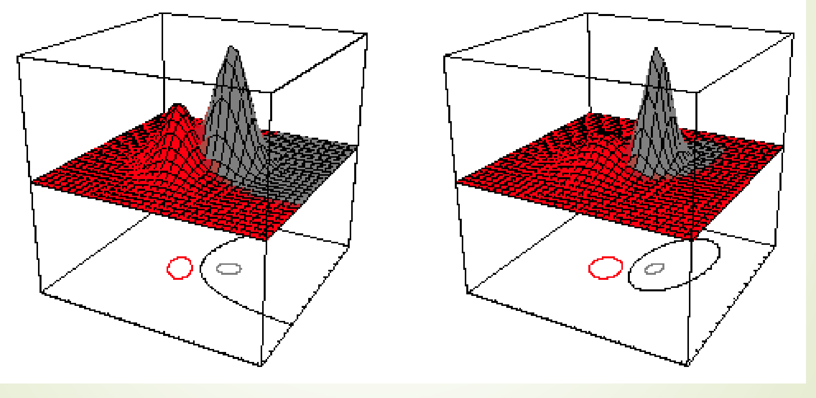
- Possible hyperquadrics
  - Hyperplanes
  - Pairs of hyperplanes
  - Hyperspheres
  - Hyperellipsoids
  - Hyperparaboloids

For arbitrary variance, the decision regions need not be simply connected

3/28/2021

55

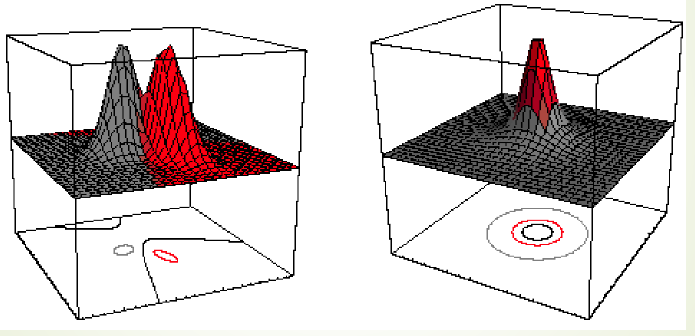
Decision Surface



3/28/2021

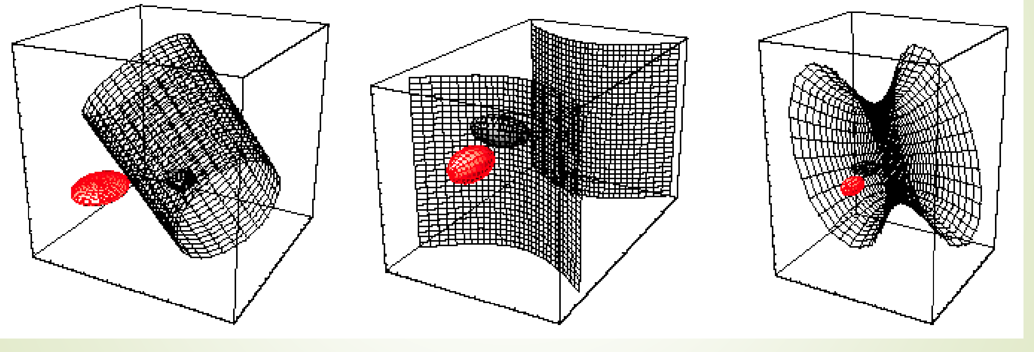
56

Decision Surface



57

Decision Surface



58

Decision Surface

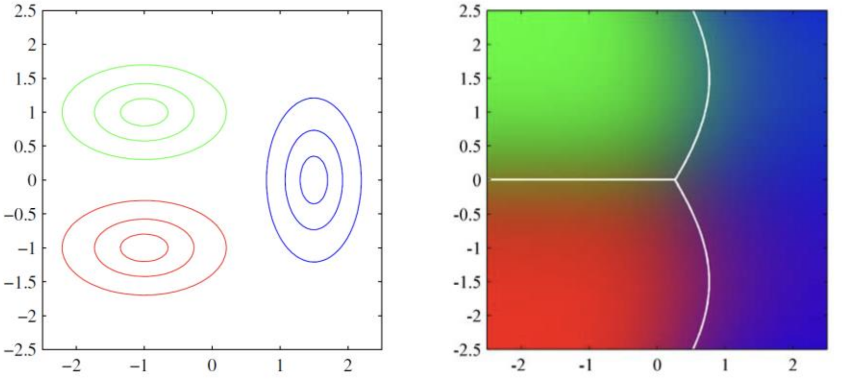


Figure 4.11 The left-hand plot shows the class-conditional densities for three classes each having a Gaussian distribution, coloured red, green, and blue, in which the red and green classes have the same covariance matrix. The right-hand plot shows the corresponding posterior probabilities, in which 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.

3/28/2021

59

Maximum Likelihood Solution

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

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

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

3/28/2021

60

Maximum Likelihood Solution

the log likelihood function that depend on π are

$$\sum_{n-1}^{N}\{t_n \ln \pi + (1-t_n)\ln(1-\pi)\}$$

Setting the derivative with respect to π equal to zero and rearranging, we obtain

$$\pi = \frac{1}{N}\sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1+N_2}$$

3/28/2021

61

Maximum Likelihood Solution

Now consider the maximization with respect to $\boldsymbol{\mu}_1$. Again we can pick out of the log likelihood function those terms that depend on $\boldsymbol{\mu}_1$

$$\sum_{n=1}^{N} t_n \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \Sigma)$$

Setting the derivative w.r.t. $\boldsymbol{\mu}_1$ equal to zero, we obtain

$$\boldsymbol{\mu}_1 = \frac{1}{N_1}\sum_{n=1}^{N} t_n \mathbf{x}_n$$

Similarly

$$\boldsymbol{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N} (1-t_n)\mathbf{x}_n$$

3/28/2021

62

Maximum Likelihood Solution

Now consider the maximization with respect to $\boldsymbol{\mu}_1$. Again we can pick out of the log likelihood function those terms that depend on $\boldsymbol{\mu}_1$

$$\sum_{n=1}^{N} t_n \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = -\frac{1}{2}\sum_{n=1}^{N} t_n(\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_1) + \text{const}$$

Setting the derivative w.r.t. $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ equal to zero, we obtain

$$\boldsymbol{\mu}_1 = \frac{1}{N_1}\sum_{n=1}^{N} t_n \mathbf{x}_n \qquad \boldsymbol{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N} (1 - t_n)\mathbf{x}_n$$

$$\begin{aligned}
&-\frac{1}{2}\sum_{n=1}^{N} t_n \ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N} t_n(\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_1) \\
&-\frac{1}{2}\sum_{n=1}^{N} (1-t_n) \ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N} (1-t_n)(\mathbf{x}_n - \boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_2) \\
&= -\frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{N}{2}\mathrm{Tr}\left\{\boldsymbol{\Sigma}^{-1}\mathbf{S}\right\} \qquad (4.77)
\end{aligned}$$

3/28/2021

Two principles for estimating parameters

- **Maximum likelihood estimation (MLE)**
  Choose $\boldsymbol{\theta}$ that maximizes the probability (likelihood) of observed data

$$\hat{\boldsymbol{\theta}}^{\mathrm{MLE}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, P(D|\boldsymbol{\theta})$$

- **Maximum a posteriori estimation (MAP)**
  Choose $\boldsymbol{\theta}$ that is most probable given prior probability and data

$$\hat{\boldsymbol{\theta}}^{\mathrm{MAP}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, P(\boldsymbol{\theta}|D) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \frac{P(D|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(D)}$$

64

Generative vs. Discriminative

Generative Approach

Ex: **Naïve Bayes**

Estimate $P(Y)$ and $P(X|Y)$

Prediction
$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(Y=y)P(X=x|Y=y)$

Discriminative Approach

Ex: **Logistic Regression**

Estimate $P(Y|X)$ directly
(Or a discriminant function: e.g., SVM)

Prediction
$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(Y=y|X=x)$

3/28/2021

Naïve Bayes classifier

- Want to learn $P(Y|X_1,\cdots,X_N)$
  - But require 2^N parameters…
- How about applying Bayes rule?
  - $P(Y|X_1,\cdots,X_N) = \frac{P(X_1,\cdots,X_N|Y)P(Y)}{P(X_1,\cdots,X_N)} \propto P(X_1,\cdots,X_N|Y)P(Y)$
  - $P(X_1,\cdots,X_N|Y)$: Need $(2^N-1)\times 2$ parameters
  - $P(Y)$: Need 1 parameter
- Apply conditional independence assumption
  - $P(X_1,\cdots,X_N|Y) = \prod_{i=1}^N P(X_i|Y)$: Need $N\times 2$ parameters

Naïve Bayes classifier

- Bayes rule:

$$P(Y = y_k | X_1, \cdots, X_N) = \frac{P(Y = y_k) P(X_1, \cdots, X_N | Y = y_k)}{\sum_j P(Y = y_j) P(X_1, \cdots, X_N | Y = y_j)}$$

- Assume conditional independence among X_i's:

$$P(Y = y_k | X_1, \cdots, X_n) = \frac{P(Y = y_k) \Pi_i P(X_i | Y = y_k)}{\sum_j P(Y = y_j) \Pi_i P(X_i | Y = y_j)}$$

- Pick the most probable Y

$$\hat{Y} \leftarrow \underset{y_k}{\operatorname{argmax}} P(Y = y_k) \Pi_i P(X_i | Y = y_k)$$

Example

- $P(Y | X_1, X_2) \propto P(Y) P(X_1, X_2 | Y) = P(Y) P(X_1 | Y) P(X_2 | Y)$

  Bayes rule Conditional indep.

- Estimating parameters

$P(Y = 1) = 0.4$	$P(Y = 0) = 0.6$
$P(X_1 = 1 \| Y = 1) = 0.2$	$P(X_1 = 0 \| Y = 1) = 0.8$
$P(X_1 = 1 \| Y = 0) = 0.7$	$P(X_1 = 0 \| Y = 0) = 0.3$
$P(X_2 = 1 \| Y = 1) = 0.3$	$P(X_2 = 0 \| Y = 1) = 0.7$
$P(X_2 = 1 \| Y = 0) = 0.9$	$P(X_2 = 0 \| Y = 0) = 0.1$

- Test example: $X_1 = 1, X_2 = 0$
  - $Y = 1$: $P(Y = 1) P(X_1 = 1 | Y = 1) P(X_2 = 0 | Y = 1) = 0.4 \times 0.2 \times 0.7 = 0.056$
  - $Y = 0$: $P(Y = 0) P(X_1 = 1 | Y = 0) P(X_2 = 0 | Y = 0) = 0.6 \times 0.7 \times 0.1 = 0.042$

Naïve Bayes Algorithm – Discrete $\mathrm{X_i}$

- For each value y_k

  Estimate $\pi_k = P(Y = y_k)$

  For each value x_{ij} of each attribute X_i

  Estimate $\theta_{ijk} = P(X_i = x_{ij} | Y = y_k)$

- Classify $\mathrm{X}^{\mathrm{test}}$

$$\hat{Y} \leftarrow \underset{y_k}{\operatorname{argmax}}\, P(Y = y_k) \Pi_i P\left(X_i^{\mathrm{test}} \middle| Y = y_k\right)$$

$$\hat{Y} \leftarrow \underset{y_k}{\operatorname{argmax}}\, \pi_k \, \Pi_i \theta_{ijk}$$

Estimating Parameters: Discrete Y, X_i

- Maximum likelihood estimates (MLE)

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|}$$

$$\hat{\theta}_{ijk} = \hat{P}\left(X_i = x_{ij} \middle| Y = y_k\right) = \frac{\#D\left\{X_i = x_{ij} \wedge Y = y_k\right\}}{\#D\{Y = y_k\}}$$

Estimating Parameters: Discrete Y, X_i

- **Maximum likelihood estimates (MLE)**

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y=y_k\}}{|D|}$$

$$\hat{\theta}_{ijk} = \hat{P}\left(X_i = x_{ij} \middle| Y = y_k\right) = \frac{\#D\{X_i=x_{ij}, Y=y_k\}}{\#D\{Y=y_k\}}$$

- **MAP estimates (Dirichlet priors):**

$$\hat{\pi}_k = \hat{P}(Y = y_k) = \frac{\#D\{Y=y_k\}+(\beta_k-1)}{|D|+\sum_m(\beta_m-1)}$$

$$\hat{\theta}_{ijk} = \hat{P}\left(X_i = x_{ij} \middle| Y = y_k\right) = \frac{\#D\{X_i=x_{ij}, Y=y_k\}+(\beta_k-1)}{\#D\{Y=y_k\}+\sum_m(\beta_m-1)}$$

What If We Have Continuous X_i

- Gaussian Naïve Bayes (GNB): assume

$$P(X_i = x \mid Y = y_k) = \frac{1}{\sqrt{2\pi}\sigma_{ik}} \exp\left(-\frac{(x-\mu_{ik})^2}{2\sigma_{ik}^2}\right)$$

- Additional assumption on σ_{ik}:
  - Is independent of Y (σ_i)
  - Is independent of X_i (σ_k)
  - Is independent of X_i and Y (σ)

Naïve Bayes Algorithm – Continuous X_i

- For each value $\mathrm{y_k}$
  - Estimate $\pi_k = P(Y = y_k)$
  - For each attribute X_i estimate
    - Class conditional mean μ_{ik}, variance σ_{ik}
- Classify X^{test}

$$\hat{Y} \leftarrow \underset{y_k}{\operatorname{argmax}}\, P(Y = y_k)\Pi_i P\left(X_i^{\text{test}} \middle| Y = y_k\right)$$

$$\hat{Y} \leftarrow \underset{y_k}{\operatorname{argmax}}\, \pi_k\ \Pi_i\ \mathcal{N}\left(X_i^{\text{test}}, \mu_{ik}, \sigma_{ik}\right)$$

Things to Remember

- Probability basics
  - Conditional probability, joint probability, Bayes rule
- Estimating parameters from data
  - Maximum likelihood (ML) maximize $P(\text{Data}|\theta)$
  - Maximum a posteriori estimation (MAP) maximize $P(\theta|\text{Data})$
- Naive Bayes

$$P(Y = y_k | X_1, \cdots, X_n) \propto P(Y = y_k)\Pi_i P(X_i | Y = y_k)$$

74

Generative vs. Discriminative

Generative Approach

Ex: **Naïve Bayes**

Estimate $P(Y)$ and $P(X|Y)$

Prediction
$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(Y = y)P(X = x|Y = y)$

Discriminative Approach

Ex: **Logistic Regression**

Estimate $P(Y|X)$ directly
(Or a discriminant function: e.g., SVM)

Prediction
$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(Y = y|X = x)$

3/28/2021

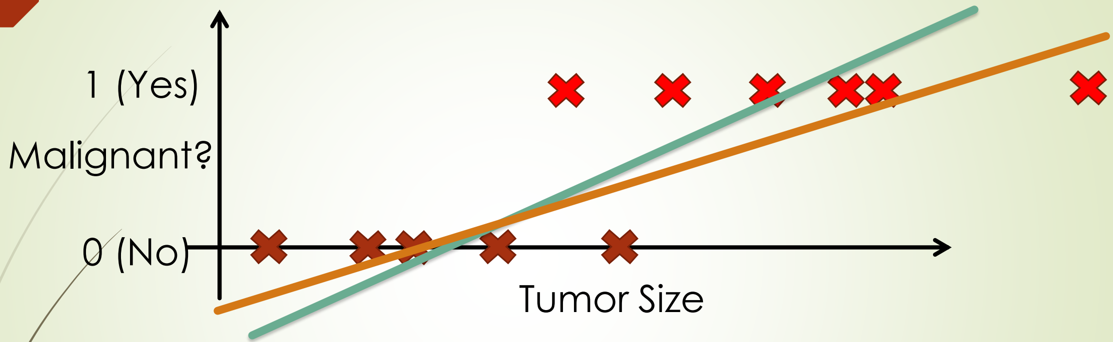
75

Logistic Regression

- **Hypothesis representation**
- Cost function
- Logistic regression with gradient descent
- Regularization
- Multi-class classification

3/28/2021

76



$h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^{\top}\mathbf{x}$

- Threshold classifier output $h_{\boldsymbol{\theta}}(\mathbf{x})$ at 0.5
  - If $h_{\boldsymbol{\theta}}(\mathbf{x}) \geq 0.5$, predict "$y = 1$"
  - If $h_{\boldsymbol{\theta}}(\mathbf{x}) < 0.5$, predict "$y = 0$"

3/28/2021

77

Logistic Regression

Classification: $y = 1$ or $y = 0$

$h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^{\top}\mathbf{x}$ (from linear regression)
can be > 1 or < 0

Logistic regression:$0 \leq h_{\boldsymbol{\theta}}(\mathbf{x}) \leq 1$

Logistic regression is actually for **classification**

3/28/2021

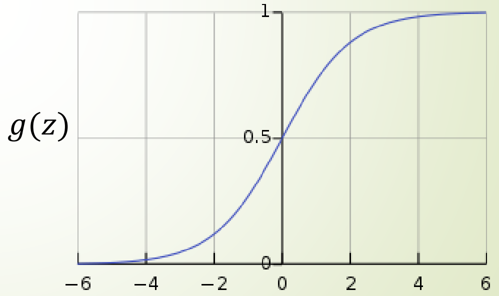
78

Hypothesis Representation

- Want $0 \le h_{\boldsymbol{\theta}}(\mathbf{x}) \le 1$
- $h_{\boldsymbol{\theta}}(\mathbf{x}) = g(\boldsymbol{\theta}^\top \mathbf{x})$,

  where $g(z) = \frac{1}{1+e^{-z}}$
- Sigmoid function
- Logistic function

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \boxed{\frac{1}{1+e^{-\boldsymbol{\theta}^\top \mathbf{x}}}}$$



3/28/2021

79

Interpretation of Hypothesis Output

- $h_{\boldsymbol{\theta}}(\mathbf{x})$ = estimated probability that $y = 1$ on input $\mathbf{x}$
- Example: If $\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$
- $h_{\boldsymbol{\theta}}(\mathbf{x}) = 0.7$
- Tell patient that 70% chance of tumor being malignant

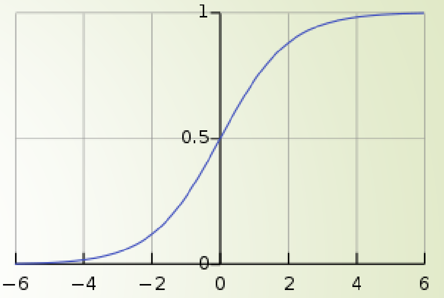
3/28/2021

80

Logistic Regression

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = g(\boldsymbol{\theta}^\top \mathbf{x})$$

$$g(z) = \frac{1}{1+e^{-z}}$$



Suppose predict "y = 1" if $h_{\boldsymbol{\theta}}(\mathbf{x}) \geq 0.5$

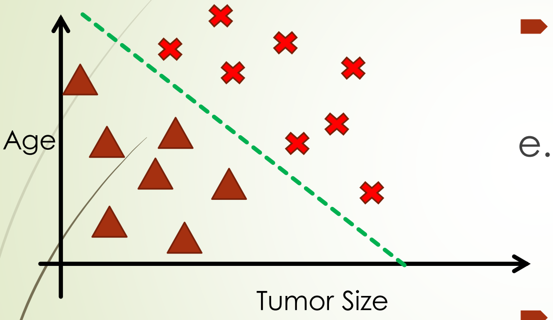
$z = \boldsymbol{\theta}^\top \mathbf{x} \geq 0$

predict "y = 0" if $h_{\boldsymbol{\theta}}(\mathbf{x}) < 0.5$

$z = \boldsymbol{\theta}^\top \mathbf{x} < 0$

81

Decision Boundary



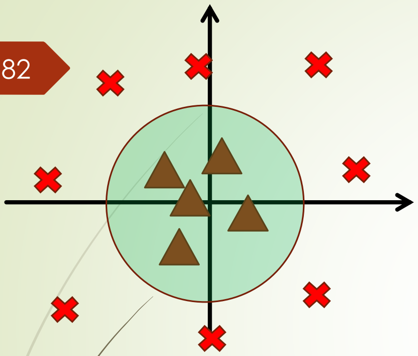
- $h_{\boldsymbol{\theta}}(\mathbf{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

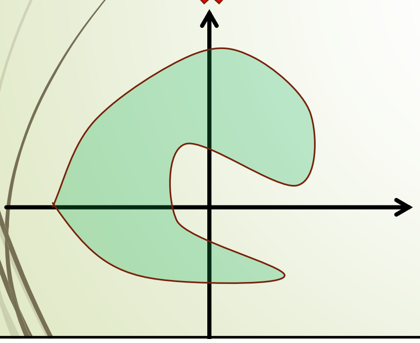
e.g., $\theta_0 = -3, \theta_1 = 1, \ \theta_2 = 1$

- Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

3/28/2021

82





- $h_{\boldsymbol{\theta}}(\mathbf{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$

e.g., $\theta_0 = -1, \theta_1 = 0, \theta_2 = 0, \theta_3 = 1,$
$\theta_4 = 1$

- Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$

- $h_{\boldsymbol{\theta}}(\mathbf{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \cdots)$

3/28/2021

83

Where Does the Form Come from?

- Logistic regression hypothesis representation

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^\top \mathbf{x}}} = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n)}}$$

- Consider learning $f\colon X \to Y$, where
  - X is a vector of real-valued features $[X_1, \cdots, X_d]^\top$
  - Y is Boolean
  - Assume all X_i are conditionally independent given Y
  - Model $P(X_i | Y = y_k)$ as Gaussian $\mathcal{N}(\mu_{ik}, \sigma_i)$
  - Model $P(Y)$ as Bernoulli π

What is $P(Y | X_1, X_2, \cdots, X_d)$?

3/28/2021

84

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$ Applying Bayes rule

$$= \frac{1}{1+\frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$ Divide by $P(Y=1)P(X|Y=1)$

$$= \frac{1}{1+\exp(\ln(\frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}))}$$ Apply $\exp(\ln(\cdot))$

$$= \frac{1}{1+\exp(\ln(\frac{1-\pi}{\pi}) + \boxed{\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)}})}$$ Plug in $P(X_i|Y)$

$$\boxed{P(x|y_k) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-\mu_{ik})^2}{2\sigma_i^2}}}$$

$$\boxed{\sum_i (\boxed{\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}} X_i + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2})}$$

$$P(Y=1|X_1, X_2, \cdots, X_n) = \frac{1}{1+\exp(\theta_0 + \sum_i \boxed{\theta_i} X_i)}$$

85

Logistic Regression

- Hypothesis representation
- **Cost function**
- Logistic regression with gradient descent
- Regularization
- Multi-class classification

86

Cost Function

Training set with N examples

$$\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})$$

$$\mathbf{x} \in \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \qquad x_0 = 1,\ y \in \{0, 1\}$$

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^\top \mathbf{x}}}$$

How to choose parameters $\boldsymbol{\theta}$?

3/28/2021

87

Cost Function for Linear Regression

$$J(\boldsymbol{\theta}) = \frac{1}{2N} \sum_{n=1}^{N} \left(h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right) - y^{(n)}\right)^2 = \frac{1}{N} \sum_{n=1}^{N} \mathrm{Cost}(h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right), y))$$

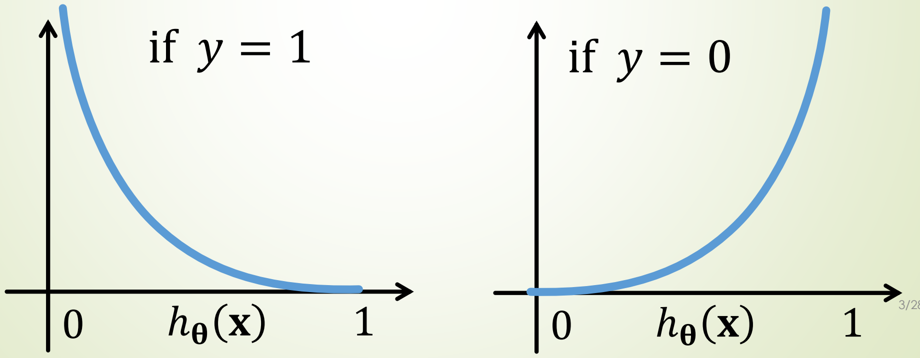
$$\boxed{\mathrm{Cost}(h_{\boldsymbol{\theta}}(\mathbf{x}), y) = \frac{1}{2}(h_{\boldsymbol{\theta}}(\mathbf{x}) - y)^2}$$

3/28/2021

88

Cost function for Logistic Regression

$$\boxed{\mathrm{Cost}(h_{\boldsymbol{\theta}}(\mathbf{x}), y) = \begin{cases} -\log\big(h_{\boldsymbol{\theta}}(\mathbf{x})\big) & \text{if } y = 1 \\ -\log\big(1 - h_{\boldsymbol{\theta}}(\mathbf{x})\big) & \text{if } y = 0 \end{cases}}$$



89

Logistic Regression Cost Function

- $\mathrm{Cost}(h_{\boldsymbol{\theta}}(\mathbf{x}), y) = \begin{cases} -\log\big(h_{\boldsymbol{\theta}}(\mathbf{x})\big) & \text{if } y = 1 \\ -\log\big(1 - h_{\boldsymbol{\theta}}(\mathbf{x})\big) & \text{if } y = 0 \end{cases}$

$$\Downarrow$$

- $\mathrm{Cost}(h_{\boldsymbol{\theta}}(\mathbf{x}), y) = -y\log\big(h_{\boldsymbol{\theta}}(\mathbf{x})\big) - (1-y)\log\big(1 - h_{\boldsymbol{\theta}}(\mathbf{x})\big)$
- If $y = 1$: $\mathrm{Cost}(h_{\boldsymbol{\theta}}(\mathbf{x}), y) = -\log\big(h_{\boldsymbol{\theta}}(\mathbf{x})\big)$
- If $y = 0$: $\mathrm{Cost}(h_{\boldsymbol{\theta}}(\mathbf{x}), y) = -\log\big(1 - h_{\boldsymbol{\theta}}(\mathbf{x})\big)$

Logistic Regression

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N} \mathrm{Cost}(h_{\boldsymbol{\theta}}(\mathbf{x}^{(n)}), y^{(n)}))$$

$$= -\frac{1}{N}\left[\sum_{n=1}^{N} y^{(n)} \log\left(h_{\boldsymbol{\theta}}(\mathbf{x}^{(n)})\right) + (1 - y^{(n)}) \log\left(1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(n)})\right)\right]$$

Learning: fit parameter $\boldsymbol{\theta}$

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

Prediction: given new $\mathbf{x}$

Output $h_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^\top \mathbf{x}}}$

Where Does the **Cost** Come from?

- Training set with m examples

$$\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}$$

- Maximum likelihood estimate for parameter $\boldsymbol{\theta}$

$$\boldsymbol{\theta}_{\mathrm{MLE}} = \arg\max_{\boldsymbol{\theta}} P_{\boldsymbol{\theta}}\left((\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\right)$$

$$= \arg\max_{\boldsymbol{\theta}} \prod_{n=1}^{N} P_{\boldsymbol{\theta}}\left((\mathbf{x}^{(n)}, y^{(n)})\right)$$

- Maximum <u>conditional</u> likelihood estimate for parameter $\boldsymbol{\theta}$

92

Maximum Conditional Likelihood Estimation

- **Goal**: choose $\boldsymbol{\theta}$ to maximize conditional likelihood of training data
  - $P_{\boldsymbol{\theta}}(Y=1|X=x) = h_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^{\top}\mathbf{x}}}$
  - $P_{\boldsymbol{\theta}}(Y=0|X=x) = 1 - h_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{e^{-\boldsymbol{\theta}^{\top}\mathbf{x}}}{1+e^{-\boldsymbol{\theta}^{\top}\mathbf{x}}}$
- **Training data** $\mathrm{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}$
- **Data likelihood** $= \prod_{n=1}^{N} P_{\boldsymbol{\theta}}\left((\mathbf{x}^{(n)}, y^{(n)})\right)$
- **Data conditional likelihood** $= \prod_{n=1}^{N} P_{\boldsymbol{\theta}}(y^{(n)}|\mathbf{x}^{(n)})$

$$\boxed{\boldsymbol{\theta}_{\mathrm{MCLE}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{n=1}^{N} P_{\boldsymbol{\theta}}(y^{(n)}|\mathbf{x}^{(n)})}$$

3/28/2021

93

Expressing Conditional log-Likelihood

$$\mathcal{L}(\boldsymbol{\theta}) = \log \prod_{n=1}^{N} P_{\boldsymbol{\theta}}(y^{(n)}|\mathbf{x}^{(n)}) = \sum_{n=1}^{N} \log P_{\boldsymbol{\theta}}(y^{(n)}|\mathbf{x}^{(n)})$$

$$= \sum_{n=1}^{N} y^{(n)} \log P_{\boldsymbol{\theta}}(y^{(n)} = 1|\mathbf{x}^{(n)}) + (1 - y^{(n)}) \log P_{\boldsymbol{\theta}}(y^{(n)} = 0|\mathbf{x}^{(n)})$$

$$= \sum_{n=1}^{N} y^{(n)} \log\left(h_{\boldsymbol{\theta}}(\mathbf{x}^{(n)})\right) + (1 - y^{(n)}) \log\left(1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(n)})\right)$$

$$\boxed{\mathrm{Cost}(h_{\boldsymbol{\theta}}(\mathbf{x}), y) = \begin{cases} -\log\left(h_{\boldsymbol{\theta}}(\mathbf{x})\right) & \text{if } y = 1 \\ -\log\left(1 - h_{\boldsymbol{\theta}}(\mathbf{x})\right) & \text{if } y = 0 \end{cases}}$$

3/28/2021

94

Logistic Regression

- Hypothesis representation
- Cost function
- **Logistic regression with gradient descent**
- Regularization
- Multi-class classification

3/28/2021

95

Gradient Descent

$$J(\boldsymbol{\theta}) = -\frac{1}{N}\left[\sum_{n=1}^{N} y^{(n)} \log\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right)\right) + (1 - y^{(n)}) \log\left(1 - h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right)\right)\right]$$

Goal: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

Good news: Convex function!
Bad news: No analytical solution

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \boldsymbol{\theta}_j} J(\boldsymbol{\theta})$$

}

(Simultaneously update all θ_j)

$$\boxed{\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right) - y^{(n)}\right)\mathbf{x}_j^{(n)}}$$

Slide credit: Andrew Ng

96

Gradient Descent

$$J(\boldsymbol{\theta}) = -\frac{1}{N}\left[\sum_{n=1}^{N} y^{(n)} \log\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right)\right) + (1-y^{(n)}) \log\left(1 - h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right)\right)\right]$$

Goal: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

Repeat { (Simultaneously update all θ_j)

$$\theta_j := \theta_j - \alpha \frac{1}{N}\sum_{n=1}^{N}\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right) - y^{(n)}\right) x_j^{(n)}$$

}

3/28/2021

97

Gradient Descent: Linear vs Logistic

Gradient descent for **Linear Regression**

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{N}\sum_{n=1}^{N}\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right) - y^{(n)}\right) x_j^{(n)} \qquad \boxed{h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^{\top}\mathbf{x}}$$

}

Gradient descent for **Logistic Regression**

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{N}\sum_{n=1}^{N}\left(h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right) - y^{(n)}\right) x_j^{(n)} \qquad \boxed{h_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^{\top}\mathbf{x}}}}$$

}

3/28/2021

98

Logistic Regression

- Hypothesis representation
- Cost function
- Logistic regression with gradient descent
- **Regularization**
- Multi-class classification

3/28/2021

99

How about MAP?

- Maximum conditional likelihood estimate (MCLE)

$$\boxed{\boldsymbol{\theta}_{\text{MCLE}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{n=1}^{N} P_{\boldsymbol{\theta}}\left(y^{(n)} | \mathbf{x}^{(n)}\right)}$$

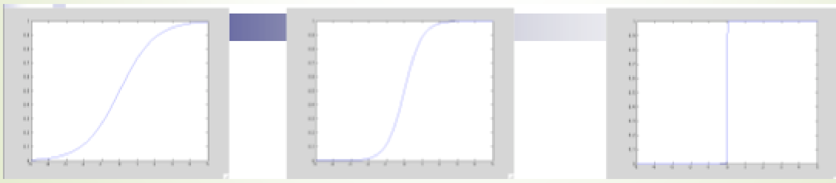
- Maximum conditional a posterior estimate (MCAP)

$$\boxed{\boldsymbol{\theta}_{\text{MCAP}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} P_{\boldsymbol{\theta}}\left(y^{(n)} | \mathbf{x}^{(n)}\right) P(\boldsymbol{\theta})}$$

100

Prior $P(\boldsymbol{\theta})$

- Common choice of $P(\boldsymbol{\theta})$:
  - Normal distribution, zero mean, identity covariance
  - “Pushes” parameters towards zeros
- Corresponds to **Regularization**
  - Helps avoid very large weights and overfitting



$$\frac{1}{1+e^{-x}} \qquad \frac{1}{1+e^{-2x}} \qquad \frac{1}{1+e^{-100x}}$$

3/28/2021

101

MLE vs. MAP

- **Maximum conditional likelihood estimate (MCLE)**

$$\theta_j := \theta_j - \alpha \frac{1}{N} \sum_{n=1}^{N} \left(h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right) - y^{(n)}\right) x_j^{(n)}$$

- **Maximum conditional a posterior estimate (MCAP)**

$$\theta_j := \theta_j - \alpha\lambda\theta_j - \alpha \frac{1}{N} \sum_{n=1}^{N} \left(h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right) - y^{(n)}\right) x_j^{(n)}$$

3/28/2021

102

Logistic Regression

- Hypothesis representation
- Cost function
- Logistic regression with gradient descent
- Regularization
- **Multi-class classification**

3/28/2021

103

Multi-Class Classification

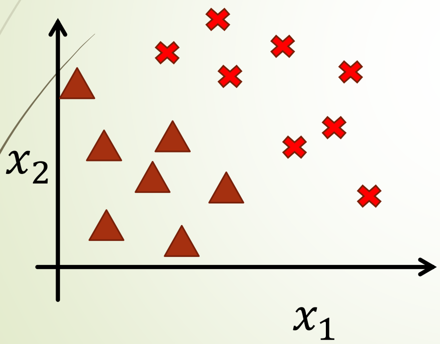
- Email foldering/tagging: Work, Friends, Family, Hobby
- Medical diagrams: Not ill, Cold, Flu
- Weather: Sunny, Cloudy, Rain, Snow

Slide credit: Andrew Ng

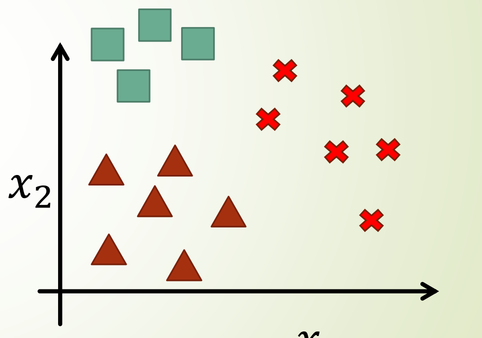
104

Multi-Class Classification

Binary classification



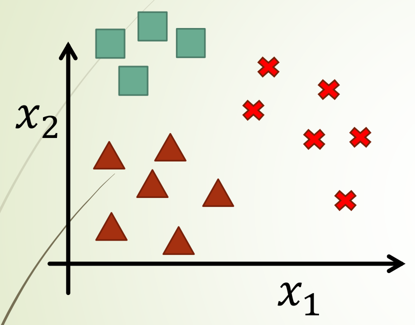
Multiclass classification



3/28/2021

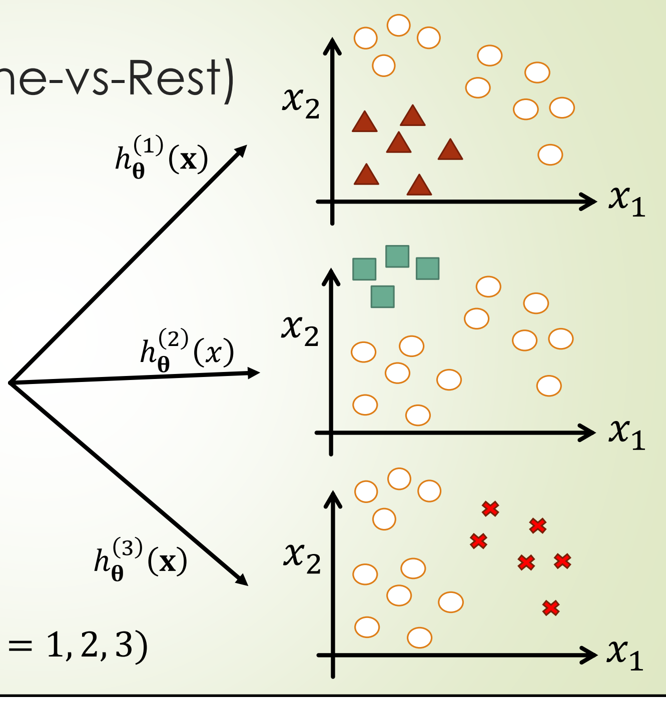
105

One-vs-All (One-vs-Rest)



Class 1:
Class 2:
Class 3:

$$h_{\boldsymbol{\theta}}^{(i)}(\mathbf{x}) = P(y = i|\mathbf{x}; \boldsymbol{\theta}) \quad (i = 1, 2, 3)$$



106

One-vs-All

- Train a logistic regression classifier $h_{\boldsymbol{\theta}}^{(i)}(\mathbf{x})$ for each class i to predict the probability that $y = i$
- Given a new input $\mathbf{x}$, pick the class i that maximizes

$$\max_{i} h_{\boldsymbol{\theta}}^{(i)}(\mathbf{x})$$

3/28/2021

107

Generative vs. Discriminative

Generative Approach

Ex: **Naïve Bayes**

Estimate $P(Y)$ and $P(X|Y)$

Prediction
$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(Y = y)P(X = x|Y = y)$

Discriminative Approach

Ex: **Logistic Regression**

Estimate $P(Y|X)$ directly
(Or a discriminant function: e.g., SVM)

Prediction
$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(Y = y|X = x)$

3/28/2021