FUNDAMENTALS OF MACHINE LEARNING

LINEAR DISCRIMINANTION

CSCI3320

Prof. John C.S. Lui, CSE Department, CUHK
Introduction to Machine Learning

INTRODUCTION

- Discuss *linear discrimination* methods
- We assume instances of a class are *linearly separable* from instances of other classes
  - *Give example in class of linearly separable classes*
  - *Give example in class of linearly non-separable classes*
- A discriminant-based method that ***estimates*** parameters of the linear discriminant directly from a given label samples

Likelihood- vs. Discriminant-based Classification

- In previous chapters, we define a set of discriminant functions $g_j(\boldsymbol{x}), j = 1, \ldots, K$, then we

$$\text{choose } C_i \text{ if } g_i(\boldsymbol{x}) = \max_{j=1,\ldots,K} g_j(\boldsymbol{x})$$

- **Likelihood-based method:**
  - estimate prior probabilities for each class: $\hat{P}(C_i)$
  - estimate the class likelihoods: $\hat{p}(\boldsymbol{x}|C_i)$
  - use Bayes' rule to calculate the posterior densities $P(C_i|\boldsymbol{x})$
- Then define the discriminant functions as

$$g_i(\boldsymbol{x}) = \log \hat{P}(C_i|\boldsymbol{x})$$

- Used this method in parametric, semi-parametric and nonparametric approaches.

Likelihood- vs. Discriminant-based Classification

- **Discriminant-based: Assume** a model for

$$g_i(\boldsymbol{x}|\Phi_i)$$

  no need for density estimation

- Estimating the ***boundaries is enough***; no need to accurately estimate the densities inside the boundaries
- Instead of making assumption on the form of the class densities (e.g., Gaussian), it makes assumption on the ***form of the boundaries*** separating classes
- In discriminant-based classification, it is about **optimization of the model parameters Φ_i** to maximize the **quality of the separation** of classes

Linear Discriminant

- **Linear discriminant for class i :**

$$g_i(\boldsymbol{x}|\boldsymbol{w}_i, w_{i0}) = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0} = \sum_{j=1}^{d} w_{ij}\, x_j + w_{i0}$$

- **Advantages:**
  - **Simple**: $O(d)$ space/computation
  - **Knowledge extraction**:
    - Weighted sum of attributes;
    - Magnitude of w_{ij} shows the importance of x_j for the class i discriminant
    - positive (enforcing) or negative (inhibiting) weights (e.g., credit score)
  - Linear discriminant is optimal when the likelihood $p(\boldsymbol{x}|C_i)$ are Gaussian with shared covariance matrix
  - Useful when classes are (almost) linearly separable

Generalized Linear Model

- **Quadratic discriminant:** $g_i(\boldsymbol{x}|\boldsymbol{W}_i, \boldsymbol{w}_i, w_{i0}) = \boldsymbol{x}^T\boldsymbol{W}_i\boldsymbol{x} + \boldsymbol{w}_i^T\boldsymbol{x} + w_{i0}$
  - Complexity is $O(d^2)$, and we have bias/variance dilemma
  - Require much larger training sets and may over-fit on small samples

For example, let $d = 2$, $\boldsymbol{W}_i = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, we have

$$\begin{aligned} g_i() &= \boldsymbol{x}^T\boldsymbol{W}_i\boldsymbol{x} + \boldsymbol{w}_i^T\boldsymbol{x} + w_{i0} \\ &= [x_1, x_2]\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}^T\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + w_{i0} \\ &= a_{11}x_1^2 + (a_{12} + a_{21})x_1x_2 + a_{22}x_2^2 + w_1x_1 + w_2x_2 + w_{i0} \end{aligned}$$

- Adding higher-order (product) terms. E.g., two inputs x_1 and x_2

$$z_1 = x_1, z_2 = x_2, z_3 = x_1^2, z_4 = x_2^2, z_5 = x_1x_2$$

take $\boldsymbol{z} = [z_1, z_2, z_3, z_4, z_5]^T$ as the input

Map from **x** to **z** using nonlinear basis functions and use a linear discriminant in **z**-space $g_i(\boldsymbol{x}) = \sum_{j=1}^{k} w_j\phi_{ij}(\boldsymbol{x})$

where $\phi_{ij}(\boldsymbol{x})$ are *basis functions*

Generalized Linear Model

- Higher-order terms are only one set of possible basis functions
- Other possible basis functions:
  - $\sin(x_1)$
  - $\exp(-(x_1 - m)^2/c)$
  - $\exp(-\|\boldsymbol{x} - \boldsymbol{m}\|^2/c)$
  - $\log(x_2)$
  - $1(x_1 > c)$
  - $1(ax_1 + bx_2 > c)$

where *m*, *a*, *b*, *c* are scalars and ***m*** is a *d*-dimensional vector and *1(b)* is an indicating function

Geometry of Linear Discriminant: Two Classes

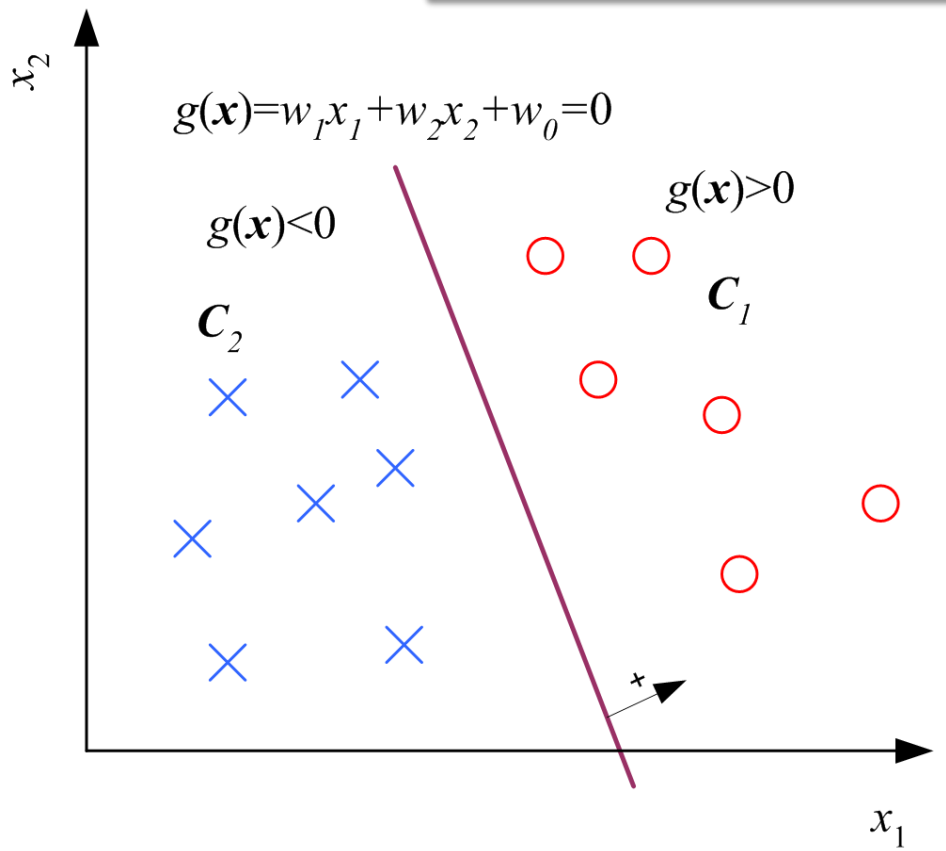
When there are 2 classes, one discriminant is enough:

$$\begin{aligned} g(\boldsymbol{x}) &= g_1(\boldsymbol{x}) - g_2(\boldsymbol{x}) \\ &= (\boldsymbol{w}_1^T\boldsymbol{x} + w_{10}) - (\boldsymbol{w}_2^T\boldsymbol{x} + w_{20}) \\ &= (\boldsymbol{w}_1 - \boldsymbol{w}_2)^T\boldsymbol{x} + (w_{10} - w_{20}) \\ &= \boldsymbol{w}^T\boldsymbol{x} + w_0 \end{aligned}$$

$$\text{choose} \begin{cases} C_1 & \text{if } g(\boldsymbol{x}) > 0 \\ C_2 & \text{otherwise} \end{cases}$$

$\boldsymbol{w}$ is the *weight vector*

w_0 is the *threshold*.



Geometrical Interpretation for 2 Classes

Take two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ both on the decision surface

$$g(\boldsymbol{x}_1) = g(\boldsymbol{x}_2) = 0$$

$$\boldsymbol{w}^T\boldsymbol{x}_1 + w_0 = \boldsymbol{w}^T\boldsymbol{x}_2 + w_0$$

$$\boldsymbol{w}^T(\boldsymbol{x}_1 - \boldsymbol{x}_2) = 0$$

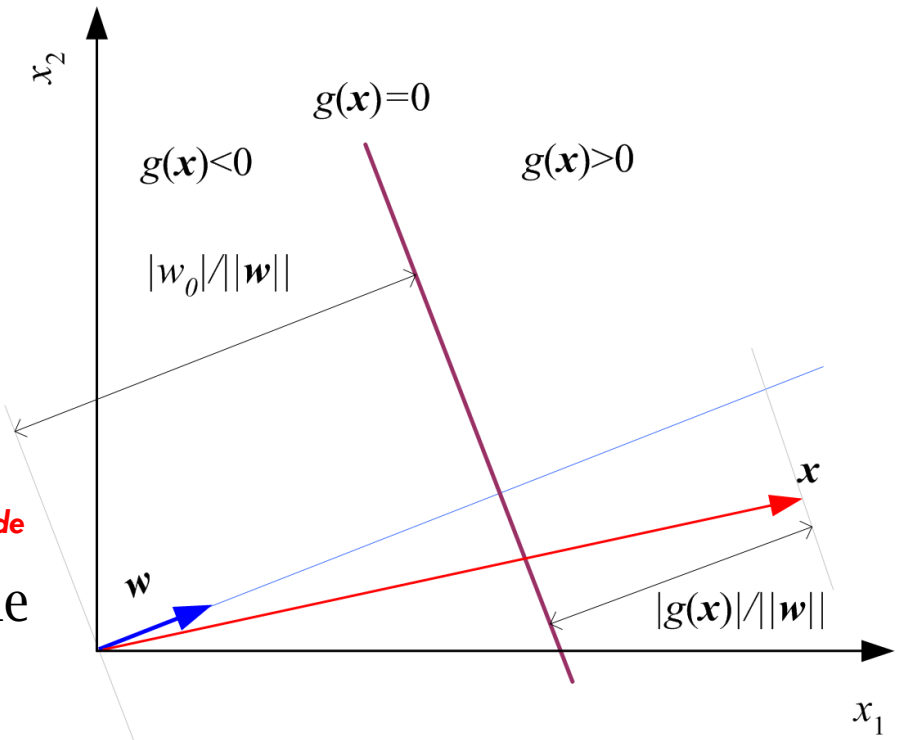
$\boldsymbol{w}$ is normal to any vector on the hyperplane

$\boldsymbol{x}_p$ is the normal projection of $\boldsymbol{x}$ onto the hyperplane

r gives the distance from $\boldsymbol{x}$ to the hyperplane since $g(\boldsymbol{x}_p) = 0$

$$\boldsymbol{x} = \boldsymbol{x}_p + r\frac{\boldsymbol{w}}{||\boldsymbol{w}||}$$

Vector magnitude



See *derivation* in the next page:

$$r = \frac{g(\boldsymbol{x})}{||\boldsymbol{w}||}$$

Apply it to the origin $\boldsymbol{x} = 0$ and following similar derivation:

$$r_0 = \frac{w_0}{||\boldsymbol{w}||}$$

Geometrical Interpretation for 2 Classes (derivation)

We have $g(\boldsymbol{x}_p) = \boldsymbol{w}^T \boldsymbol{x}_p + w_0 = 0$.

Since $\boldsymbol{x} = \boldsymbol{x}_p + \frac{r}{||\boldsymbol{w}||}\boldsymbol{w}$, or $\boldsymbol{x}_p = \boldsymbol{x} - \frac{r}{||\boldsymbol{w}||}\boldsymbol{w}$.

Apply $\boldsymbol{x}_p$ onto the $g()$, we have:

$$\boldsymbol{w}^T \left(\boldsymbol{x} - \frac{r}{||\boldsymbol{w}||}\boldsymbol{w} \right) + w_0 = 0$$

$$\boldsymbol{w}^T \boldsymbol{x} - \frac{r}{||\boldsymbol{w}||}\boldsymbol{w}^T \boldsymbol{w} + w_0 = 0$$

$$\boldsymbol{w}^T \boldsymbol{x} - \frac{r}{||\boldsymbol{w}||}||\boldsymbol{w}||^2 + w_0 = 0$$

$$\boldsymbol{w}^T \boldsymbol{x} + w_0 = r||\boldsymbol{w}||$$

$$g(\boldsymbol{x}) = r||\boldsymbol{w}||$$

$$r = \frac{g(\boldsymbol{x})}{||\boldsymbol{w}||}$$

Multiple Classes ($K > 2$)

there are K discriminant functions

$$g_i(\boldsymbol{x}|\boldsymbol{w}_i, w_{i0}) = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}$$

where

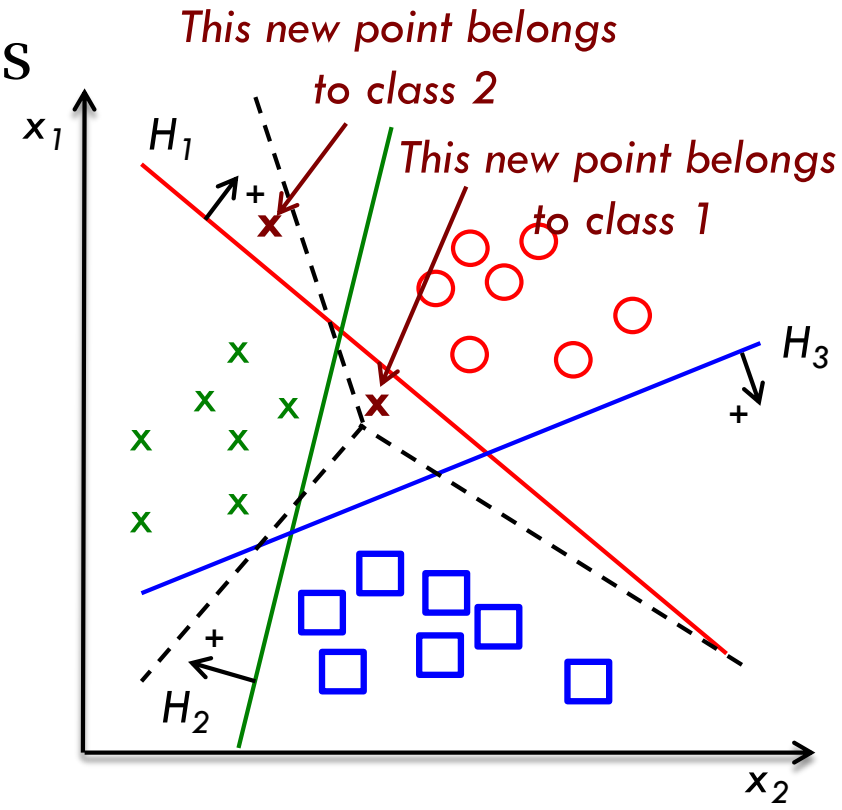
$$g_i(\boldsymbol{x}|\boldsymbol{w}_i, w_{i0}) = \begin{cases} > 0 & \text{if } \boldsymbol{x} \in C_i \\ \leq 0 & \text{otherwise} \end{cases}$$

Classes are

linearly separable

To relax "linear separability"

Choose C_i if $g_i(\boldsymbol{x}) = \max_{j=1}^{K} g_j(\boldsymbol{x})$



$|g_i(\boldsymbol{x})| / \|\boldsymbol{w}_i\|$ is the distance from the input point to the hyperplane i. Assume $\boldsymbol{w}_i$ has the same length. **It assigns the point to the class with the largest distance.**

Pairwise Separation

- If the classes are not linearly separable, we divide it into a set of linear problems, e.g., *pairwise separation*
- Let K be # of classes. We use $K(K-1)/2$ linear discriminants, $g_{ij}(\boldsymbol{x})$, one for every pair of distinct classes, where

$$g_{ij}(\boldsymbol{x}|\boldsymbol{w}_{ij}, w_{ij0}) = \boldsymbol{w}_{ij}^T\boldsymbol{x} + w_{ij0}$$

- We compute the parameters, $\boldsymbol{w}_{ij}$, during the training

$$g_{ij}(\boldsymbol{x}) = \begin{cases} > 0 & \text{if } \boldsymbol{x} \in C_i \\ \leq 0 & \text{if } \boldsymbol{x} \in C_j \\ \text{don't care} & \text{otherwise} \end{cases} \quad i, j = 1, \ldots, K \text{ and } i \neq j$$

If $\boldsymbol{x}^t \in C_k$ and $k \neq i, k \neq j$, then $\boldsymbol{x}^t$ is not used in training of $g_{ij}(\boldsymbol{x})$

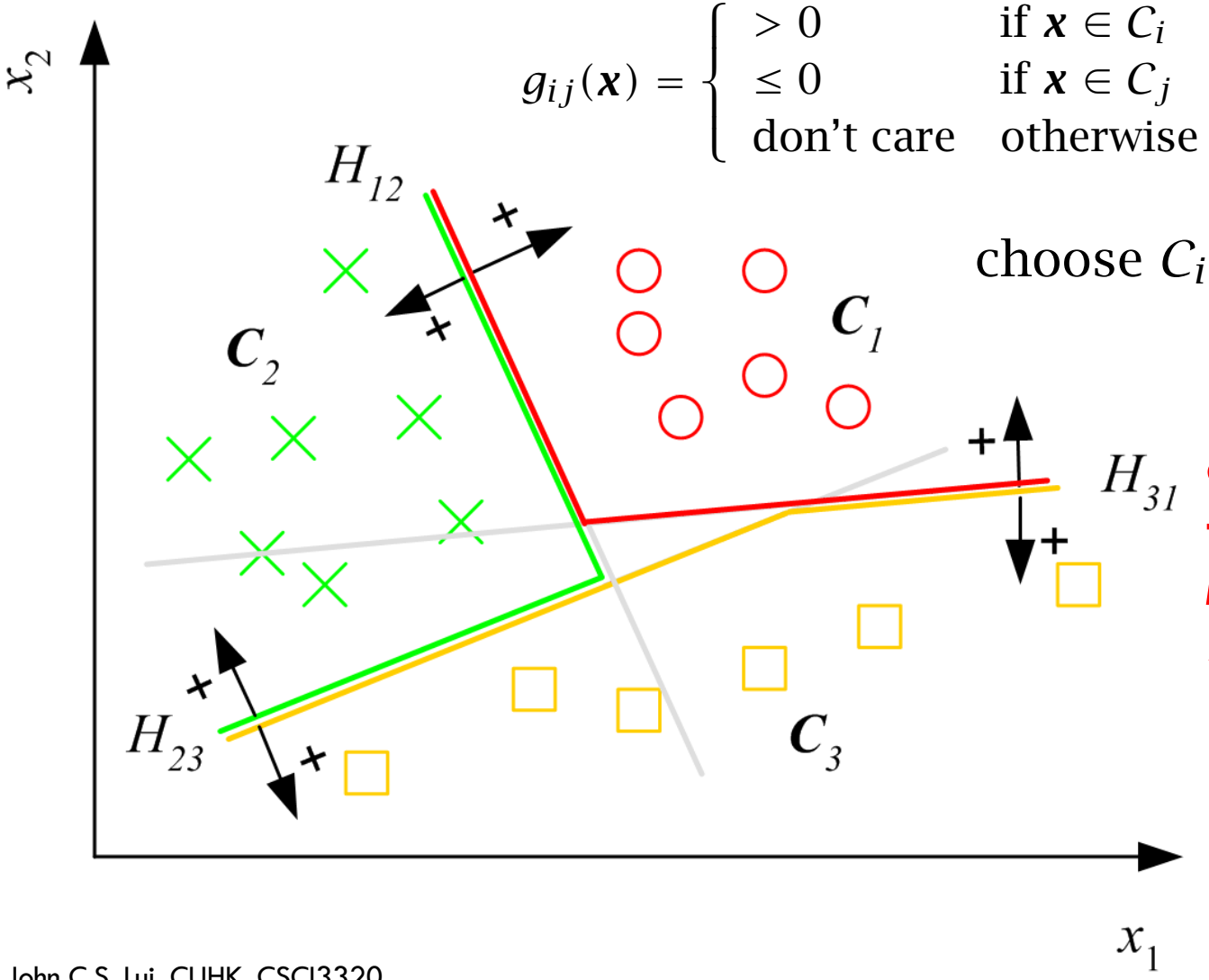
- During test, we choose C_i if $\forall\, j \neq i, g_{ij}(\boldsymbol{x}) > 0$

Pairwise Separation

$$g_{ij}(\boldsymbol{x}|\boldsymbol{w}_{ij}, w_{ij0}) = \boldsymbol{w}_{ij}^T\boldsymbol{x} + w_{ij0}$$

$$g_{ij}(\boldsymbol{x}) = \begin{cases} > 0 & \text{if } \boldsymbol{x} \in C_i \\ \leq 0 & \text{if } \boldsymbol{x} \in C_j \\ \text{don't care} & \text{otherwise} \end{cases} \quad i, j = 1, \ldots, K \text{ and } i \neq j$$

choose C_i if $\forall j \neq i, g_{ij}(\boldsymbol{x}) > 0$



For an input to be assigned in C_1, it has to be positive side of H_{12} and H_{13} (or negative side of H_{31})

Parametric Discrimination Revisited

In parametric ML, if the class densities, $p(\boldsymbol{x}|C_i)$, are Gaussian and share a common covariance matrix or $p(\boldsymbol{x}|C_i) \sim N(\boldsymbol{\mu}_i, \sum)$, we have:

$$g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}$$

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

$$w_{i0} = -\frac{1}{2}\boldsymbol{\mu}_i^T \Sigma^{-1}\boldsymbol{\mu}_i + \log P(C_i)$$

Given the data, we "estimate" $\boldsymbol{\mu}_i$ and Σ via $\boldsymbol{m}_i$ and $\boldsymbol{S}$, then use the last two equations to get $\boldsymbol{w}_i$ and w_{i0}

Parametric Discrimination Revisited

Consider a special case of $K=2$ classes. We define

$$y \equiv P(C_1|\boldsymbol{x}) \text{ and } P(C_2|\boldsymbol{x}) = 1 - y$$

In classification, we have

$$\text{choose } C_1 \text{ if } \begin{cases} y > 0.5 \\ \frac{y}{1-y} > 1 \\ \log \frac{y}{1-y} > 0 \end{cases} \quad \text{and } C_2 \text{ otherwise}$$

$\log y/(1-y)$ is known as the *logit* transformation or *log odds* of y

Note that $\log \frac{y}{1-y}$ goes from $-\infty$ to $+\infty$.

Assume two normal classes share a common covariance matrix

$$\text{logit}(P(C_1|\boldsymbol{x})) = \log \frac{P(C_1|\boldsymbol{x})}{1-P(C_1|\boldsymbol{x})} = \log \frac{P(C_1|\boldsymbol{x})}{P(C_2|\boldsymbol{x})}$$

$$= \log \frac{p(\boldsymbol{x}|C_1)}{p(\boldsymbol{x}|C_2)} + \log \frac{P(C_1)}{P(C_2)}$$

$$= \log \frac{(2\pi)^{-d/2}|\boldsymbol{\Sigma}|^{-1/2}\exp[-(1/2)(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)]}{(2\pi)^{-d/2}|\boldsymbol{\Sigma}|^{-1/2}\exp[-(1/2)(\boldsymbol{x}-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2)]} + \log \frac{P(C_1)}{P(C_2)}$$

$$= \boldsymbol{w}^T\boldsymbol{x} + w_0$$

1. estimate $\boldsymbol{m}_1, \boldsymbol{m}_2, \mathbf{S}$
2. calculate $g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$ and choose C_1 if $g(\boldsymbol{x}) > 0$

where

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

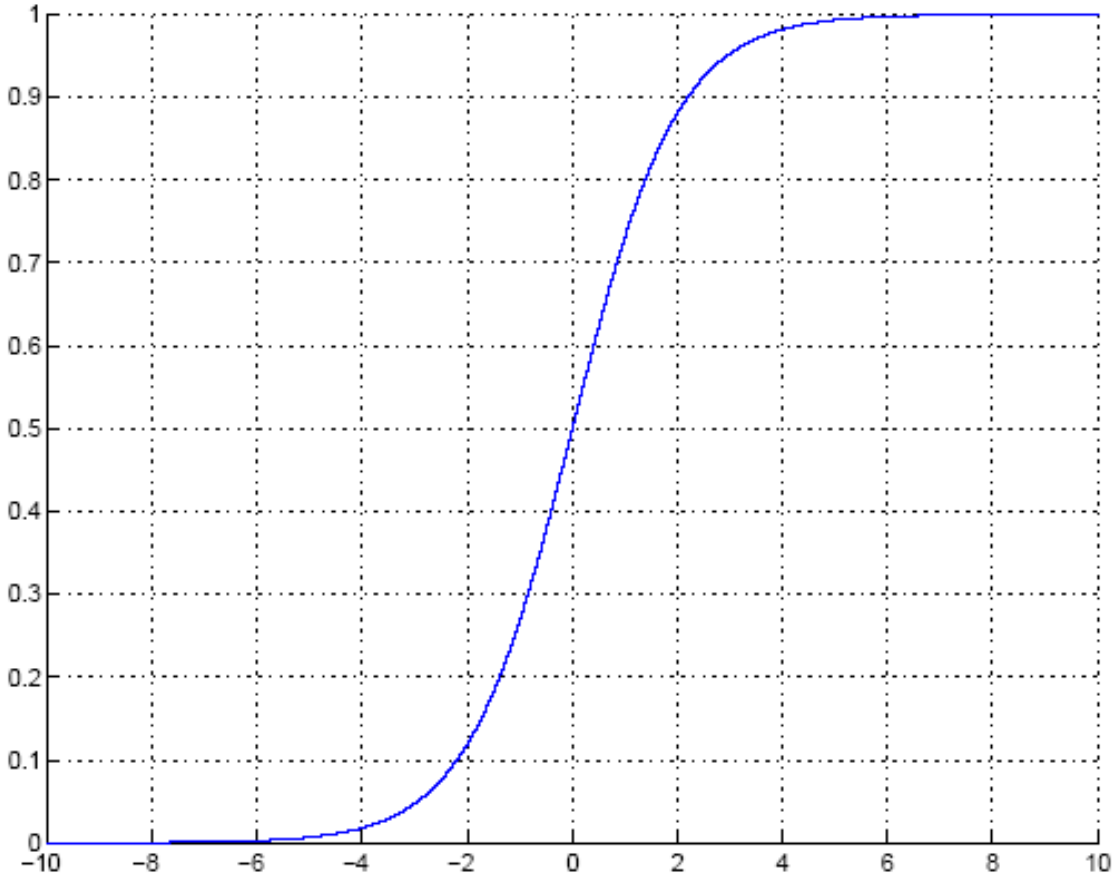
$$w_0 = -\frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \log \frac{P(C_1)}{P(C_2)}$$

$$P(C_1|\boldsymbol{x}) = \text{sigmoid}(\boldsymbol{w}^T\boldsymbol{x} + w_0) = \frac{1}{1+\exp[-(\boldsymbol{w}^T\boldsymbol{x} + w_0)]}$$

Show in page 21 how we derive this from logit !!!!

Let us study when $\boldsymbol{w}^T\boldsymbol{x} + w_0$ is very positive (very negative).

Sigmoid (Logistic) Function



1. calculate $g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$ and choose C_1 if $g(\boldsymbol{x}) > 0$, or
2. calculate $y = \text{sigmoid}(\boldsymbol{w}^T\boldsymbol{x} + w_0)$ and choose C_1 if $y > 0.5$,

Will show later how to handle $K>2$

Gradient-Descent

- In discriminant-based approach, we **optimized** the parameters of the discriminants to reduce classification error
- Define $E(\boldsymbol{w}|\boldsymbol{X})$ as error with parameters $\boldsymbol{w}$ on sample $\boldsymbol{X}$

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}}\, E(\boldsymbol{w}|\boldsymbol{X})$$

- No analytical solution, resort to *iterative method*
- **Gradient descent**

$$\nabla_w E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_d}\right]^T$$

- Gradient-descent: Starts from random $\boldsymbol{w}$, and updates $\boldsymbol{w}$ iteratively in the negative direction of gradient

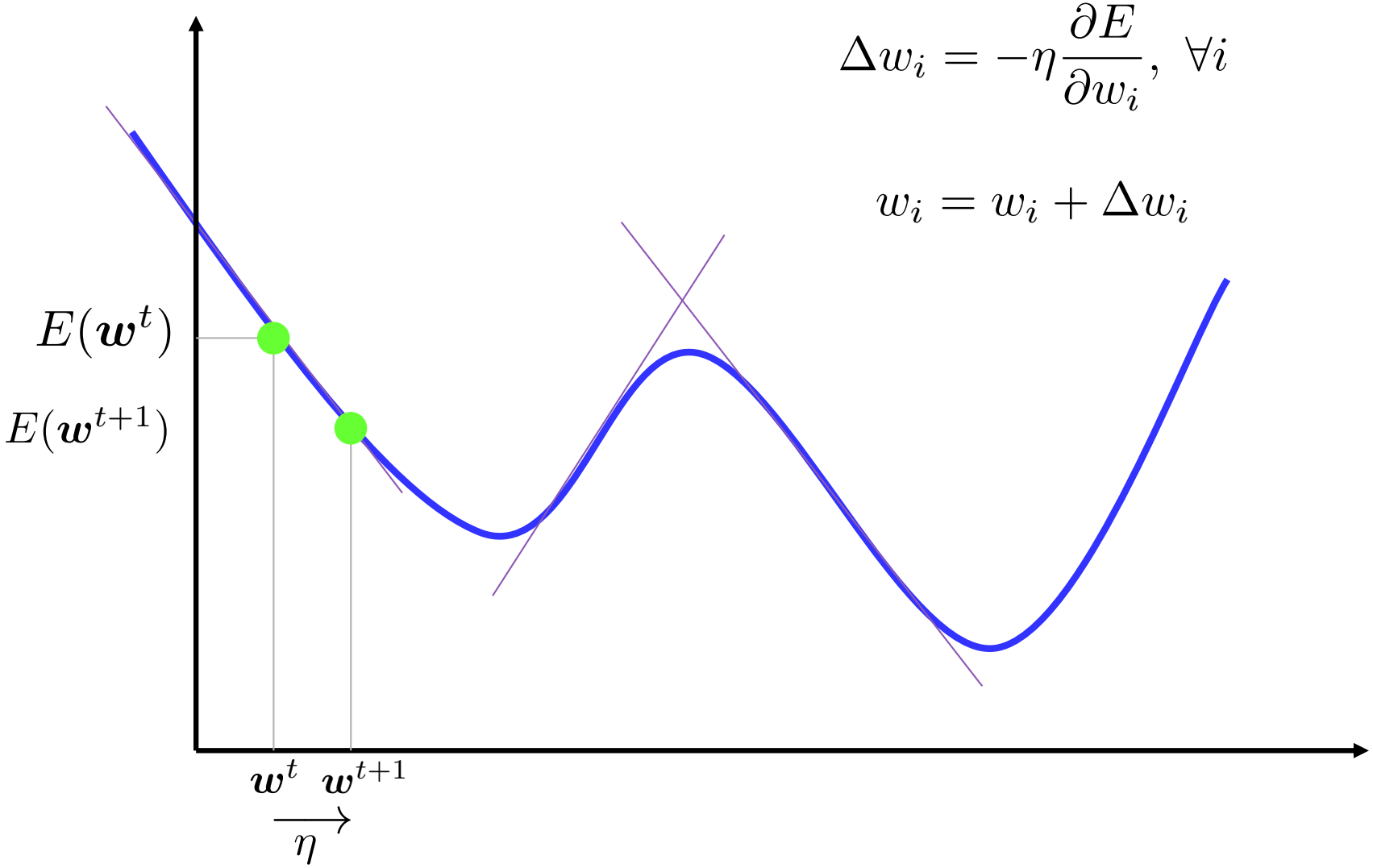
$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}, \forall i \qquad w_i = w_i + \Delta w_i$$

η **is *step-size*, or *learning factor***

Gradient-Descent

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}, \ \forall i$$

$$w_i = w_i + \Delta w_i$$



Logistic Discrimination

- In logistic discrimination, we don't model class-conditional densities $p(\boldsymbol{x}|C_i)$, but instead their ***ratio***.
- Assume 2 classes and assume **linear** log likelihood ratio

$$\log \frac{p(\boldsymbol{x}|C_1)}{p(\boldsymbol{x}|C_2)} = \boldsymbol{w}^T\boldsymbol{x} + w_0^o$$

(true when class-conditional densities are normal)

- Use Bayes' rule, we have

$$\begin{aligned}
\text{logit}(P(C_1|\boldsymbol{x})) &= \log \frac{P(C_1|\boldsymbol{x})}{1 - P(C_1|\boldsymbol{x})} \\
&= \log \frac{p(\boldsymbol{x}|C_1)}{p(\boldsymbol{x}|C_2)} + \log \frac{P(C_1)}{P(C_2)} \\
&= \boldsymbol{w}^T\boldsymbol{x} + w_0
\end{aligned}$$

where

$$w_0 = w_0^o + \log \frac{P(C_1)}{P(C_2)}$$

- Rearranging terms, our estimator of $P(C_1|\boldsymbol{x})$

$$y = \hat{P}(C_1|\boldsymbol{x}) = \frac{1}{1 + \exp[-(\boldsymbol{w}^T\boldsymbol{x} + w_0)]}$$

See next page for derivation

Derivation

We have:

$$\begin{aligned}\text{logit}(P(C_1|\boldsymbol{x})) &= \log\frac{P(C_1|\boldsymbol{x})}{P(C_2|\boldsymbol{x})} = \log\frac{P(C_1|\boldsymbol{x})}{1-P(C_1|\boldsymbol{x})} \\ &= \boldsymbol{w}^T\boldsymbol{x} + w_0\end{aligned}$$

$$\begin{aligned}\log\frac{P(C_1|\boldsymbol{x})}{1-P(C_1|\boldsymbol{x})} &= \boldsymbol{w}^T\boldsymbol{x} + w_0 \\ \log\frac{1-P(C_1|\boldsymbol{x})}{P(C_1|\boldsymbol{x})} &= -(\boldsymbol{w}^T\boldsymbol{x} + w_0) \\ \frac{1-P(C_1|\boldsymbol{x})}{P(C_1|\boldsymbol{x})} &= e^{-(\boldsymbol{w}^T\boldsymbol{x}+w_0)} \\ P(C_1|\boldsymbol{x}) &= \frac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{x}+w_0)}}\end{aligned}$$

How to learn $\boldsymbol{w}$ and w_0 for $K = 2$ Classes ?

- Input: sample of two classes:

$$\boldsymbol{X} = \{\boldsymbol{x}^t, r^t\}, \text{ where } r^t = 1 \text{ if } \boldsymbol{x}^t \in C_1 \text{ and } r^t = 0 \text{ if } \boldsymbol{x}^t \in C_2$$

- Assume r^t, given $\boldsymbol{x}^t$, is Bernoulli, or $r^t | \boldsymbol{x}^t \sim \text{Bernoulli}\ (y^t)$

$$y^t \equiv P(C_1|\boldsymbol{x}^t) = \frac{1}{1 + \exp[-(\boldsymbol{w}^T\boldsymbol{x} + w_0)]}$$

- The sample likelihood is

Use sequence of coin toss as example

$$l(\boldsymbol{w}, w_0|\mathcal{X}) = \prod_t (y^t)^{(r^t)}(1 - y^t)^{(1-r^t)}$$

- Maximize likelihood can be turned into minimize error function as $E = -\log l$
- In this case, we have *cross-entropy*:

$$E(\boldsymbol{w}, w_0|\mathcal{X}) = -\left[\sum_t r^t \log(y^t) + (1 - r^t)\log(1 - y^t)\right]$$

Training: Gradient-Descent

- We use gradient descent to minimize cross-entropy (or maximize likelihood or log likelihood)
- Note that if

$$y = \text{sigmoid}(a) = 1/(1+\exp(-a))$$

$$\frac{dy}{da} = y(1-y)$$

and we get the following update equations:

$$\Delta w_j = -\eta \frac{\partial E}{\partial w_j} = \eta \sum_t \left(\frac{r^t}{y^t} - \frac{1-r^t}{1-y^t} \right) y^t (1-y^t) x_j^t$$

$$= \eta \sum_t (r^t - y^t) x_j^t, j = 1, \ldots, d$$

See derivation

$$\Delta w_0 = -\eta \frac{\partial E}{\partial w_0} = \eta \sum_t (r^t - y^t)$$

Derivation

$$y^t = \frac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{x}+w_0)}} \quad \text{and} \quad (1-y^t) = \frac{e^{-(\boldsymbol{w}^T\boldsymbol{x}+w_0)}}{1+e^{-(\boldsymbol{w}^T\boldsymbol{x}+w_0)}}$$

$$E = -\log l = -\sum_{t=1}^{N}\left[r^t \log y^t + (1-r^t)\log(1-y^t)\right]$$

Remember the following rules: $\frac{d\log(x)}{dx} = \frac{1}{x}dx$ $\frac{de^{ax}}{dx} = ae^{ax}dx$

$$\begin{aligned}
\frac{\partial \log(l)}{\partial w_j} &= \sum_{t=1}^{N}\left[\left(\frac{r^t}{y^t}\right)\left(\frac{x_j e^{-(\boldsymbol{w}^t\boldsymbol{x}+w_0)}}{(1+e^{-(\boldsymbol{w}^t\boldsymbol{x}+w_0)})^2}\right) - \left(\frac{1-r^t}{1-y^t}\right)\left(\frac{x_j e^{-(\boldsymbol{w}^t\boldsymbol{x}+w_0)}}{(1+e^{-(\boldsymbol{w}^t\boldsymbol{x}+w_0)})^2}\right)\right] \\
&= \sum_{t=1}^{N}\left[\left(\frac{r^t}{y^t}\right)x_j y^t(1-y^t) - \left(\frac{1-r^t}{1-y^t}\right)x_j y^t(1-y^t)\right] \\
&= \sum_{t=1}^{N}\left[\left(\frac{r^t}{y^t} - \frac{1-r^t}{1-y^t}\right)x_j y^t(1-y)^t\right] \\
&= \sum_{t=1}^{N}\left[\frac{(r^t-y^t)}{y^t(1-y^t)}y^t(1-y^t)x_j\right] = \sum_{i=1}^{N}\left[(r^t-y^t)x_j\right]
\end{aligned}$$

What about "-"?

Because E is defined with "-", so the two "negatives" cancel out.

For $j = 0, \ldots, d$

$w_j \leftarrow \text{rand}(-0.01, 0.01)$

Repeat

For $j = 0, \ldots, d$

$\Delta w_j \leftarrow 0$

For $t = 1, \ldots, N$

$o \leftarrow 0$

For $j = 0, \ldots, d$

$o \leftarrow o + w_j x_j^t$

$y \leftarrow \text{sigmoid}(o)$

For $j = 0, \ldots, d$

$\Delta w_j \leftarrow \Delta w_j + (r^t - y) x_j^t$

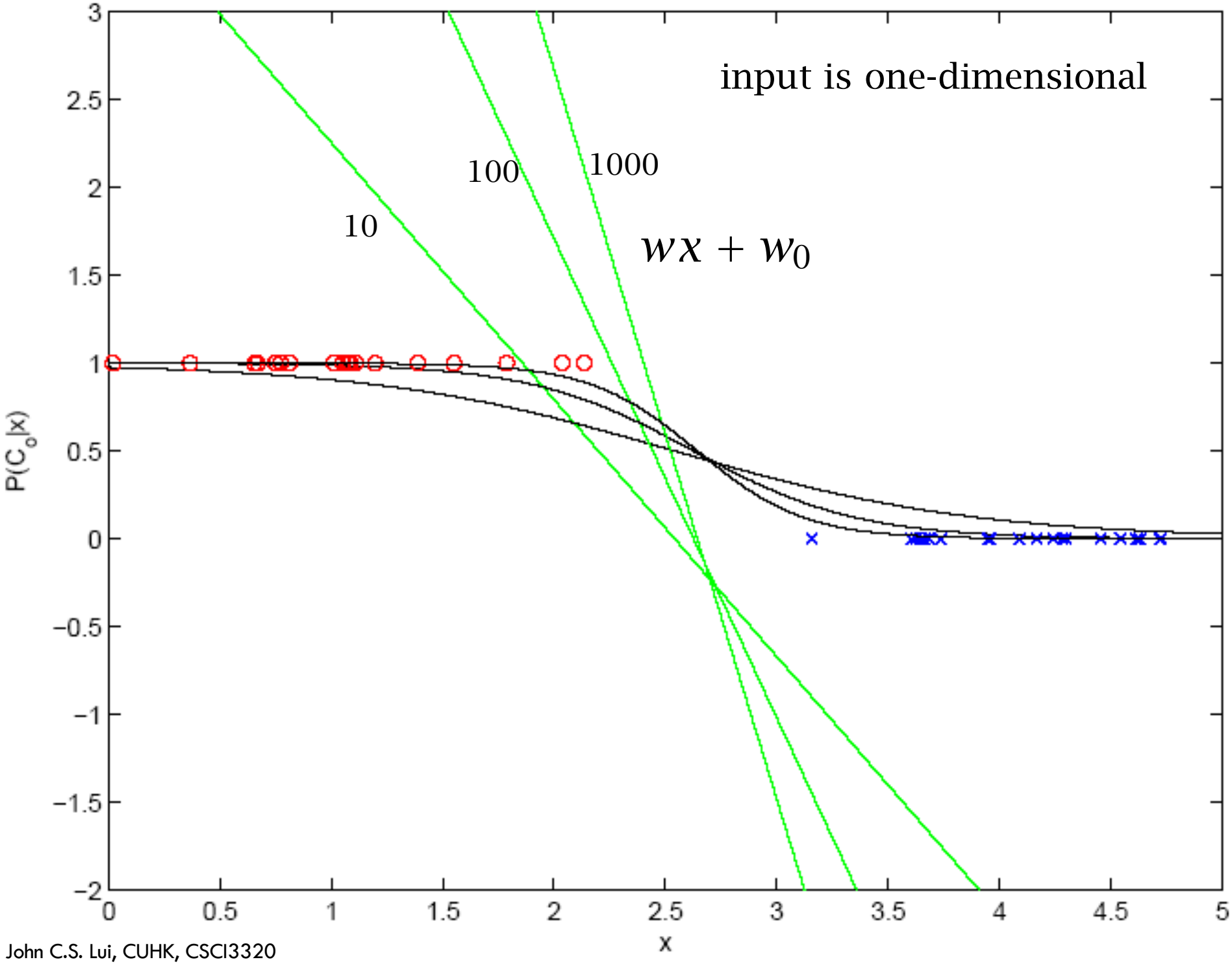
For $j = 0, \ldots, d$

$w_j \leftarrow w_j + \eta \Delta w_j$

Until convergence

Why?

Note that we are looping over N training points


input is one-dimensional
10
100
1000
$wx + w_0$
$P(C_o|x)$
x
3
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

$K > 2$ Classes

Input: $\boldsymbol{X} = \{\boldsymbol{x}^t, \boldsymbol{r}^t\}_t \quad \boldsymbol{r}^t \Big| \boldsymbol{x}^t \sim \text{Multi}_k(1, \boldsymbol{y}^t)$, where $y_i^t = P(C_i | \boldsymbol{x}^t)$

Take class C_K as the reference class and assume

$$\log \frac{p(\boldsymbol{x}|C_i)}{p(\boldsymbol{x}|C_K)} = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}^o$$

Then we have

$$\frac{P(C_i|\boldsymbol{x})}{P(C_K|\boldsymbol{x})} = \exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}]$$

$$\text{with } w_{i0} = w_{i0}^o + \log P(C_i)/P(C_K)$$

$$\sum_{i=1}^{K-1} \frac{P(C_i|\boldsymbol{x})}{P(C_K|\boldsymbol{x})} = \frac{1 - P(C_K|\boldsymbol{x})}{P(C_K|\boldsymbol{x})} = \sum_{i=1}^{K-1} \exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}]$$

$$\Rightarrow \quad P(C_K|\boldsymbol{x}) = \frac{1}{1 + \sum_{i=1}^{K-1} \exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}]}$$

$K > 2$ classes

- Since $\frac{P(C_i|\boldsymbol{x})}{P(C_K|\boldsymbol{x})} = \exp[\boldsymbol{w}_i^T x + w_{i0}]$, and so

$$P(C_i|\boldsymbol{x}) = \frac{\exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}]}{1 + \sum_{j=1}^{K-1} \exp[\boldsymbol{w}_j^T \boldsymbol{x} + w_{j0}]}, \quad i = 1,2, \ldots, K-1$$

- Treat all classes uniformly, we have the "***softmax***"

$$y_i = \hat{P}(C_i|\boldsymbol{x}) = \frac{\exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}]}{\sum_{j=1}^{K} \exp[\boldsymbol{w}_j^T \boldsymbol{x} + w_{j0}]}, \quad i = 1,2, \ldots, K$$

$K > 2$ classes: Softmax to Sigmoid

- **Softmax** can be viewed as an extension of the Sigmoid function for $K=2$ classes

$$y_i = \hat{P}(C_i|\boldsymbol{x}) = \frac{\exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}]}{\sum_{j=1}^{K} \exp[\boldsymbol{w}_j^T \boldsymbol{x} + w_{j0}]}, \quad i = 1,2,\ldots,K$$

- Consider $K=2$, define $a_i = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}, \quad i = 1,2.$
- $y_1 = \frac{e^{a_1}}{e^{a_1}+e^{a_2}} = \frac{1}{1+e^{a_2-a_1}} = \frac{1}{1+e^{-(a_1-a_2)}} = \frac{1}{1+e^{-a}}$
- $y_2 = \frac{e^{a_2}}{e^{a_1}+e^{a_2}} = \frac{e^{a_2-a_1}}{1+e^{a_2-a_1}} = \frac{e^{-(a_1-a_2)}}{1+e^{-(a_1-a_2)}} = \frac{e^{-a}}{1+e^{-a}}$
- Where $a = \boldsymbol{w}^T \boldsymbol{x} + w_0$
- This is exactly the result we have for $K=2$
- How to learn the parameters via the sample likelihood?
- Learn the parameters via the sample likelihood:

$$l(\{\boldsymbol{w}_i, w_{i0}\}_i | \boldsymbol{X}) = \Pi_t \Pi_i \left(y_i^t\right)^{r_i^t}$$

$K > 2$ Classes: how to learn the parameters

The sample likelihood is

$$l(\{\boldsymbol{w}_i, w_{i0}\}_i | \mathcal{X}) = \prod_t \prod_i (y_i^t)^{r_i^t}$$

the error function is again cross-entropy:

$$E(\{\boldsymbol{w}_i, w_{i0}\}_i | \mathcal{X}) = -\sum_t \sum_i r_i^t \log y_i^t$$

use gradient descent. If $y_i = \exp(a_i) / \sum_j \exp(a_j)$

$$\frac{\partial y_i}{\partial a_j} = y_i(\delta_{ij} - y_j)$$

δ_{ij} is the Kronecker delta

$\delta_{ij} = 1$ if $i = j$ and 0 otherwise

$$\Delta \boldsymbol{w}_j = \eta \sum_t (r_j^t - y_j^t) \boldsymbol{x}^t$$

$$\Delta w_{j0} = \eta \sum_t (r_j^t - y_j^t)$$

For $i = 1, \ldots, K$

  For $j = 0, \ldots, d$

    $w_{ij} \leftarrow \text{rand}(-0.01, 0.01)$

Repeat

  For $i = 1, \ldots, K$

    For $j = 0, \ldots, d$

      $\Delta w_{ij} \leftarrow 0$

  For $t = 1, \ldots, N$

    For $i = 1, \ldots, K$

      $o_i \leftarrow 0$

      For $j = 0, \ldots, d$

        $o_i \leftarrow o_i + w_{ij} x_j^t$

    For $i = 1, \ldots, K$

      $y_i \leftarrow \exp(o_i) / \sum_k \exp(o_k)$

    For $i = 1, \ldots, K$

      For $j = 0, \ldots, d$

        $\Delta w_{ij} \leftarrow \Delta w_{ij} + (r_i^t - y_i) x_j^t$

  For $i = 1, \ldots, K$

    For $j = 0, \ldots, d$

      $w_{ij} \leftarrow w_{ij} + \eta \Delta w_{ij}$

Until convergence

Example

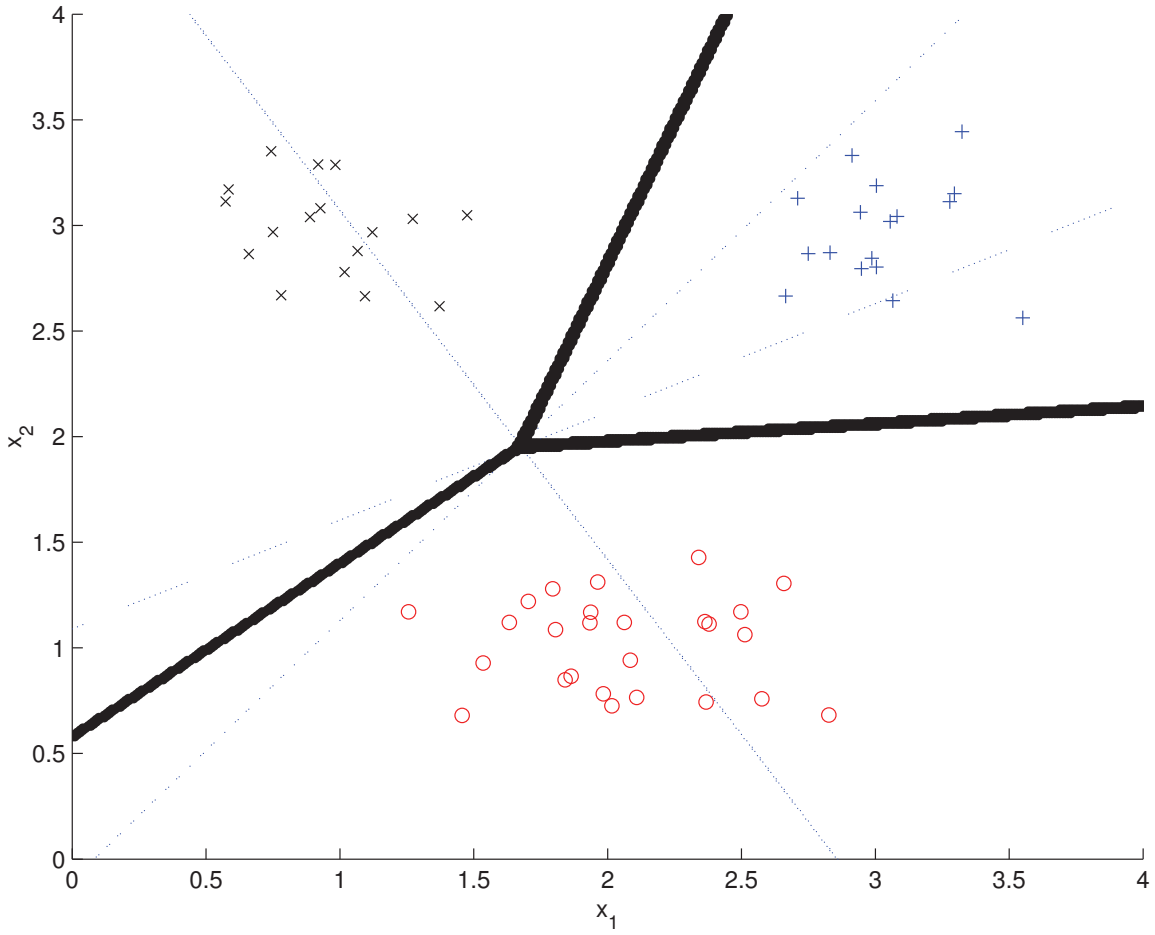
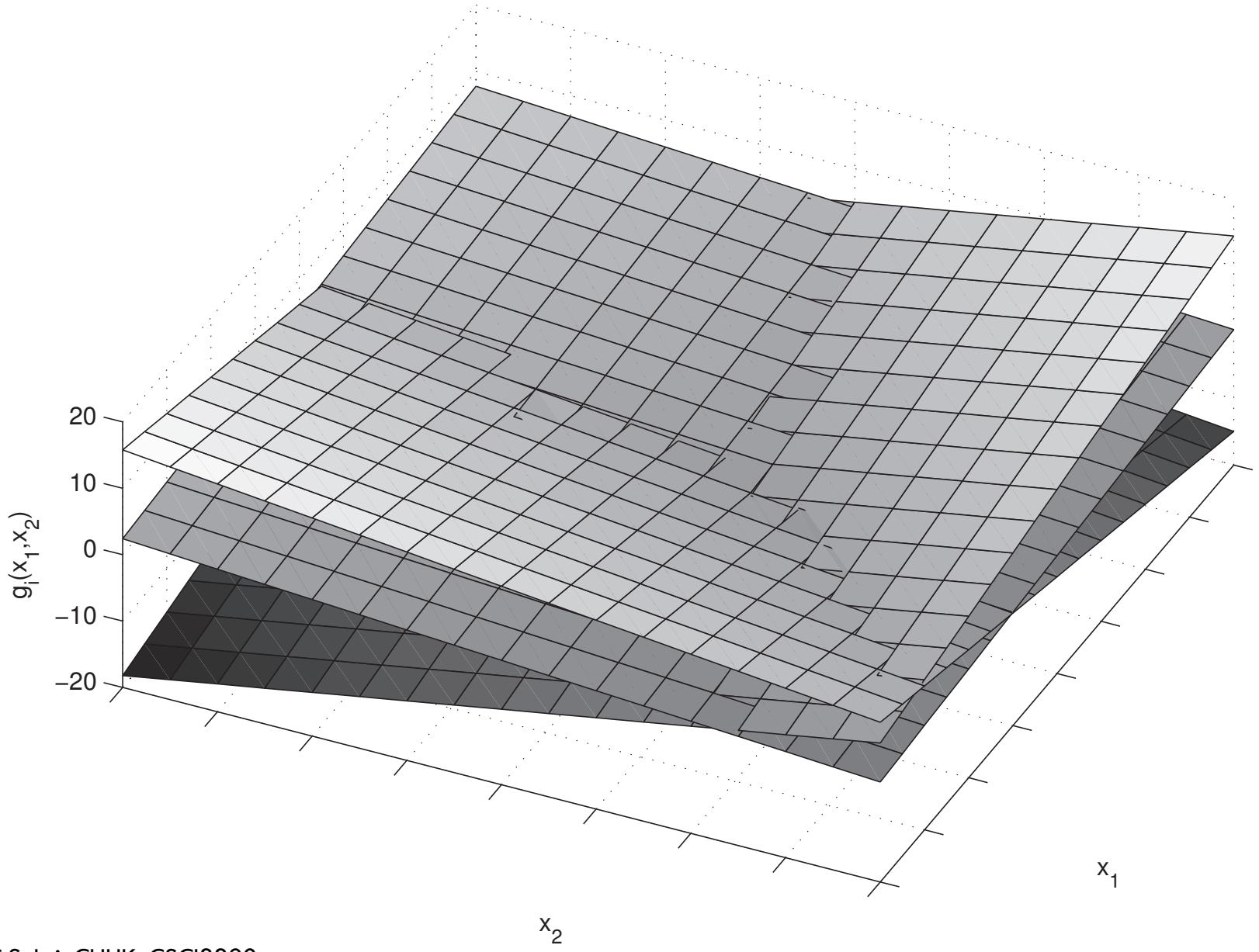


Figure 10.9 For a two-dimensional problem with three classes, the solution found by logistic discrimination. Thin lines are where $g_i(\boldsymbol{x}) = 0$, and the thick line is the boundary induced by the linear classifier choosing the maximum.

Example



Example

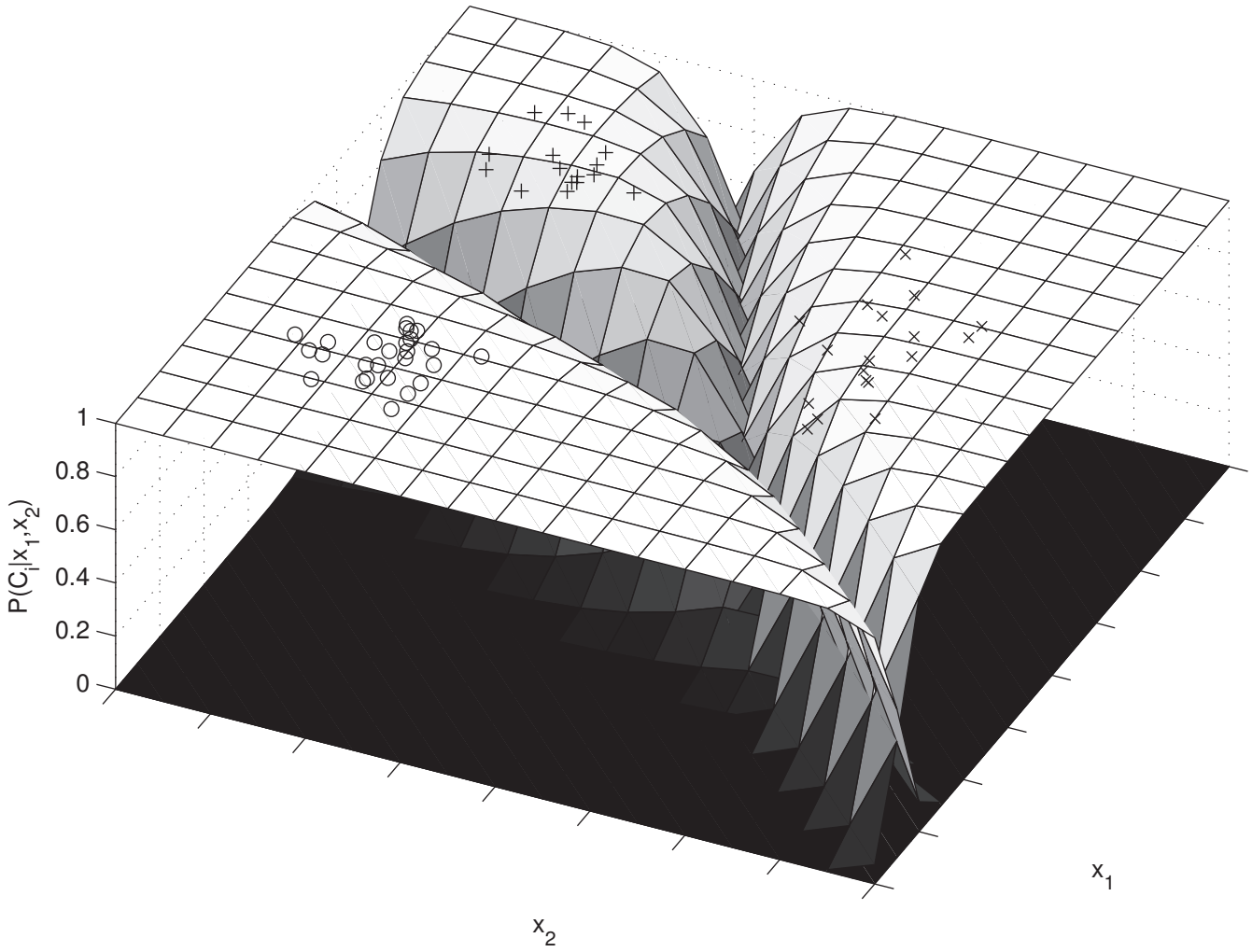


Figure 10.10 For the same example in figure 10.9, the linear discriminants (top), and the posterior probabilities after the softmax (bottom).

Generalizing the Linear Model

- Quadratic:

$$\log \frac{p(\boldsymbol{x}|C_i)}{p(\boldsymbol{x}|C_K)} = \boldsymbol{x}^T \mathbf{W}_i \boldsymbol{x} + \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}$$

- Sum of basis functions:

$$\log \frac{p(\boldsymbol{x}|C_i)}{p(\boldsymbol{x}|C_K)} = \boldsymbol{w}_i^T \boldsymbol{\phi}(\boldsymbol{x}) + w_{i0}$$

where $\boldsymbol{\phi}(\cdot)$ are basis functions. E.g., *sigmoid function*.

Discrimination by Regression

- Classes are NOT mutually exclusive and exhaustive
- In regression, the probabilistic model is

$$r^t = y^t + \epsilon \qquad \text{where } \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- If $r^t \in \{0, 1\}$, then y^t can be constrained to lie in the range of a sigmoid function

$$y^t = \text{sigmoid}(\boldsymbol{w}^T \boldsymbol{x}^t + w_0) = \frac{1}{1 + \exp[-(\boldsymbol{w}^T \boldsymbol{x}^t + w_0)]}$$

- The sample *likelihood* in regression, assume $r|\boldsymbol{x} \sim \mathcal{N}(y, \sigma^2)$

$$l(\boldsymbol{w}, w_0 | \mathcal{X}) = \prod_t \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(r^t - y^t)^2}{2\sigma^2}\right]$$

- Maximize *log likelihood* is to minimize sum of square errors $E(\boldsymbol{w}, w_0 | \mathcal{X}) = \frac{1}{2}\sum_t (r^t - y^t)^2$. Using gradient decent:

$$\Delta \boldsymbol{w} = \eta \sum_t (r^t - y^t) y^t (1 - y^t) \boldsymbol{x}^t$$

$$\Delta w_0 = \eta \sum_t (r^t - y^t) y^t (1 - y^t)$$

Learning to Rank

- Ranking: A different problem than classification or regression
- Let us say $\boldsymbol{x}^u$ and $\boldsymbol{x}^v$ are two instances, e.g., two movies

  We prefer u to v implies that $g(\boldsymbol{x}^u) > g(\boldsymbol{x}^v)$ where $g(\boldsymbol{x})$ is a score function, here linear:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

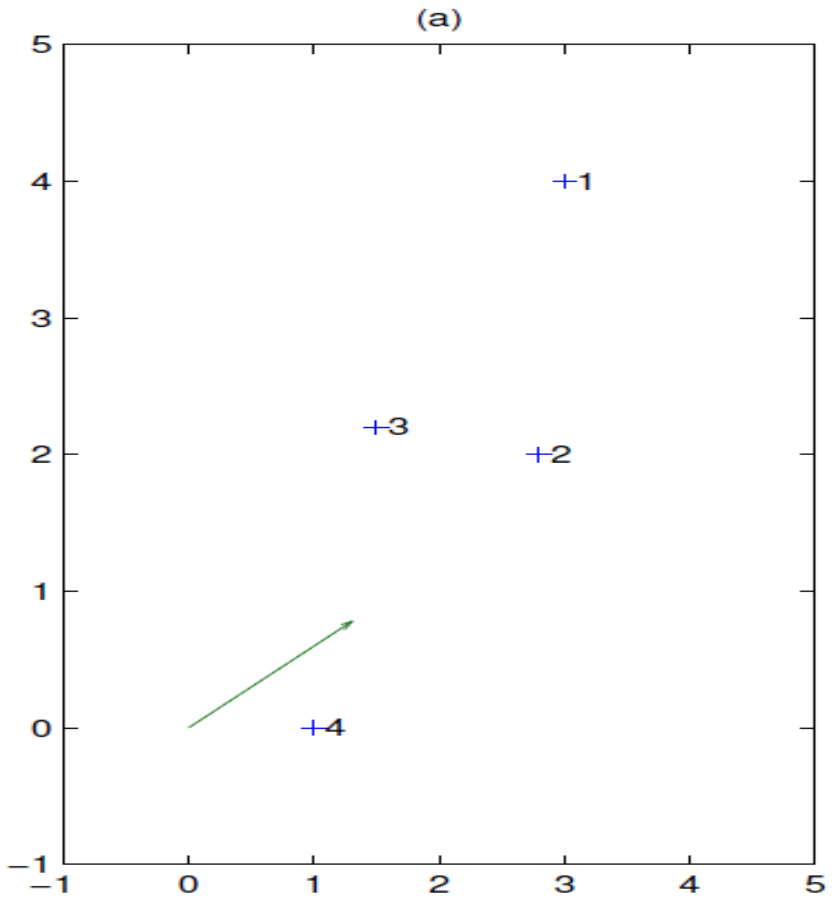
- Find a direction $\boldsymbol{w}$ such that we get the desired ranks when instances *are projected* along $\boldsymbol{w}$

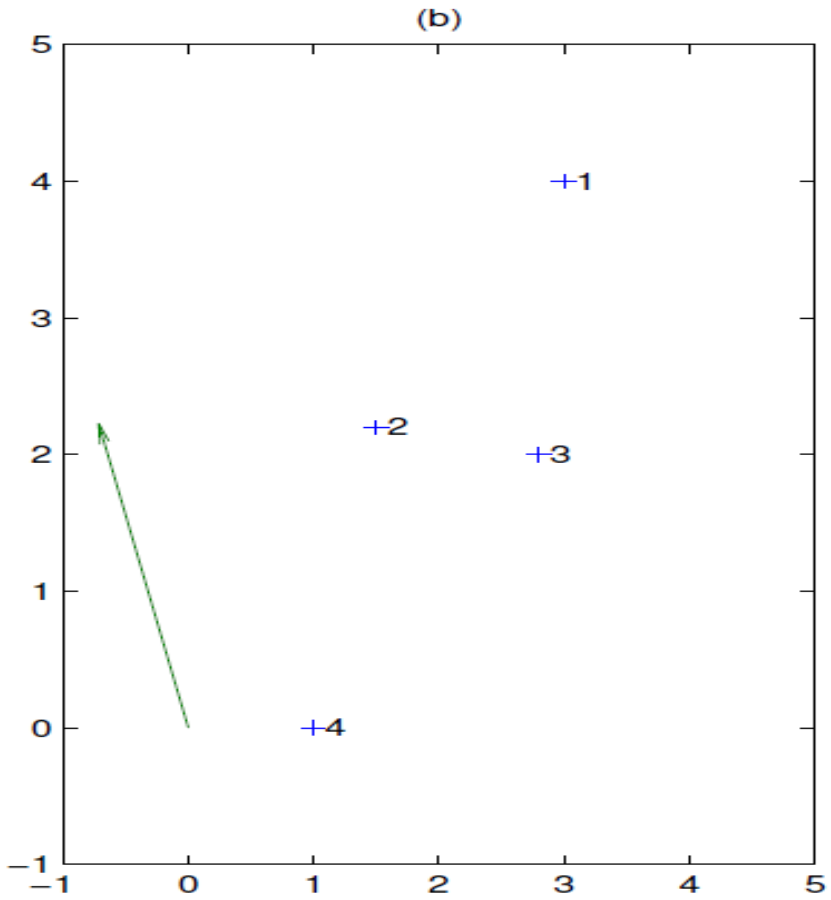
Ranking Error

- We prefer u to v implies that $g(\boldsymbol{x}^u) > g(\boldsymbol{x}^v)$, so error is $g(\boldsymbol{x}^v) - g(\boldsymbol{x}^u)$, if $g(\boldsymbol{x}^u) < g(\boldsymbol{x}^v)$

$$E(\boldsymbol{w}|\{r^u, r^v\}) = \sum_{r^u \prec r^v} \left[g(\boldsymbol{x}^v|\theta) - g(\boldsymbol{x}^u|\theta)\right]_+$$

where a_+ is equal to a if $a \geq 0$ and 0 otherwise.


(a)
5
4
3
2
1
0
−1
−1
0
1
2
3
4
5
+1
+2
+3
+4


(b)
5
4
3
2
1
0
−1
−1
0
1
2
3
4
5
+1
+2
+3
+4

Conclusion

- Geometric interpretation
- Sigmoid and softmax
- Likelihood and error functions
- Gradient decent method
- Depending on our error function, we have different update rules for $\boldsymbol{w}$
- We can change variables, to consider quadratic (or higher order) discriminants
- We consider logistic regression.