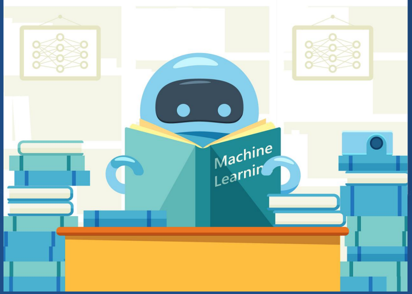
山东大学软件学院
SCHOOL OF SOFTWARE, SHANDONG UNIVERSITY

Machine Learning

机器学习



Linear Discriminant Functions

软件学院 罗昕

luoxin@sdu.edu.cn

软件学院办公楼-425

单选题 1分

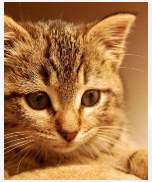
设置

如果用判别函数去分类这些数据的话，
How many functions do we need?

A Just one

B Two

C Three



提交

Contents

- Introduction
- Linear Discriminant Functions and Decision Surface
- Linear Separability
- Learning
  - Gradient Decent Algorithm
  - Newton’ s Method

Decision-Making Approaches

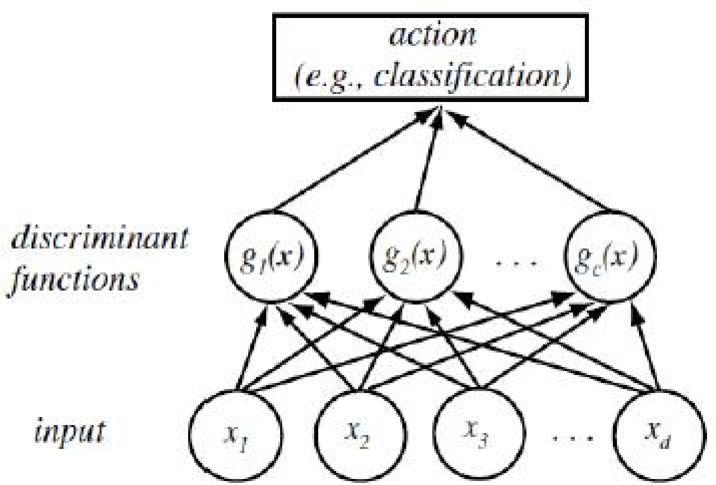
- Probabilistic Approaches
  - Based on the underlying *probability densities* of training sets.
  - For example, *Bayesian decision rule* assumes that the underlying probability densities were available.
- Discriminating Approaches
  - Assume we know the proper *forms for the discriminant functions*.
  - Use the samples to estimate the values of *parameters of the classifier*.

Discriminant Function

$$g_i : R^d \rightarrow R(\mathbf{x}) \qquad (1 \le i \le c)$$

- Useful way to represent classifier
- One function per category (**c functions**)
- Decide ω_i, if

$$g_i(x) > g_j(x) \quad \textit{for all } j \neq i$$



Linear Discriminant Functions

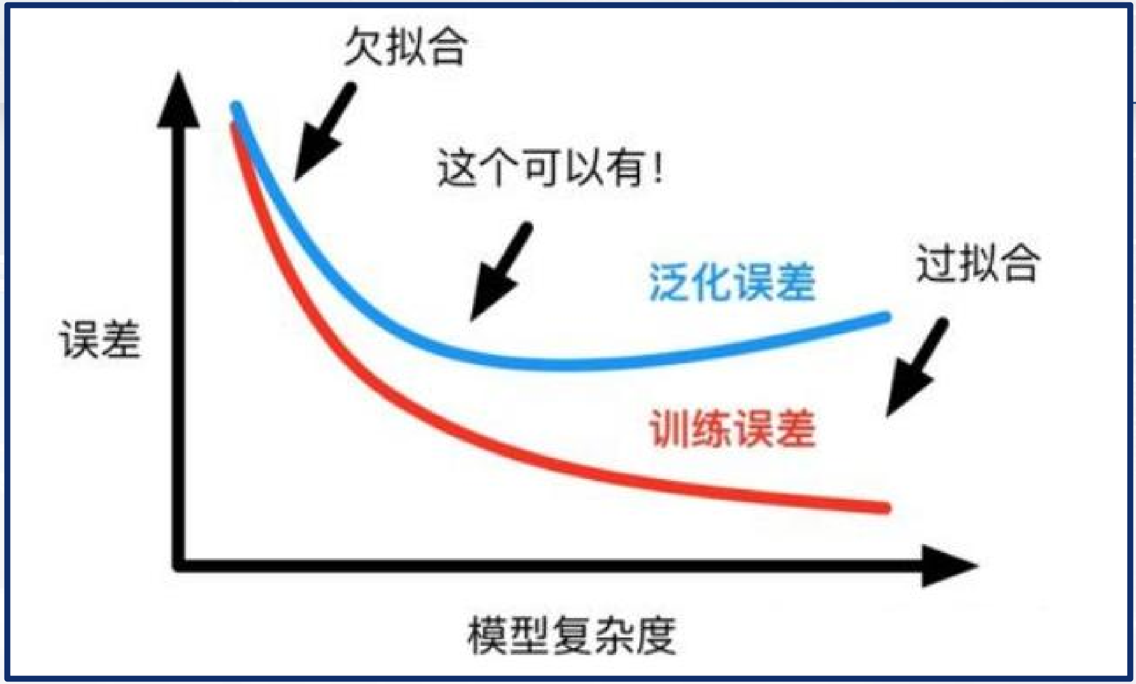
$$g_i(\mathbf{x}) = \underline{\mathbf{w}_i^T}\mathbf{x} + \underline{w_{i0}}$$

Weight vector

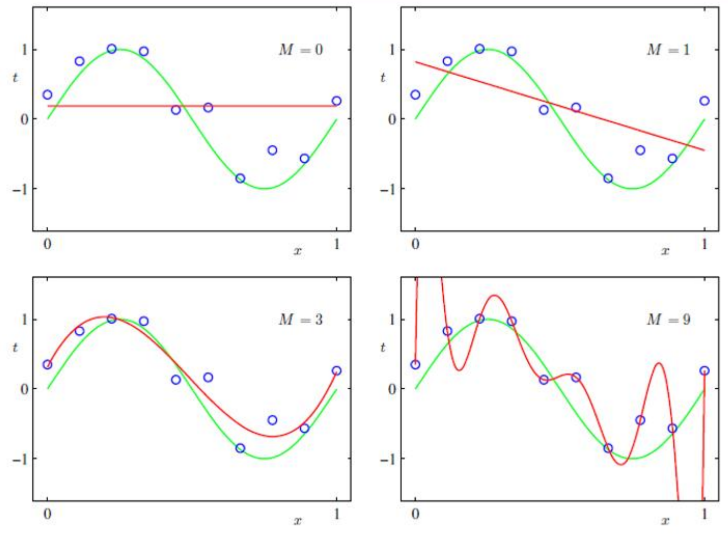
Bias/threshold 偏置/阈值

- Easy for *computing*, *analysis* and *learning*.
- Linear classifiers are attractive candidates for *initial, trial classifier*.
- Learning by minimizing a *criterion function*, e.g., training error.

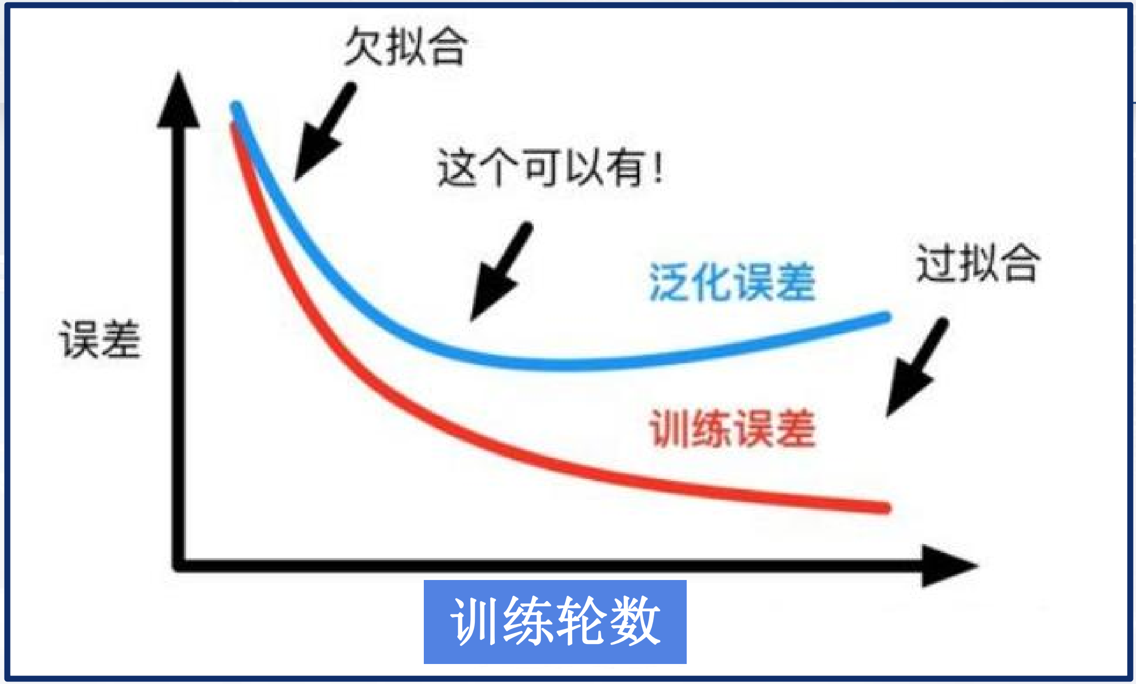
Difficulty: a small training error does not guarantee a small test error.



Example of complexity



Difficulty: a small training error does not guarantee a small test error.



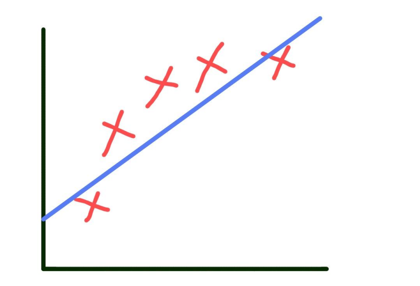
Difficulty: a small training error does not guarantee a small test error.

单选题 1分

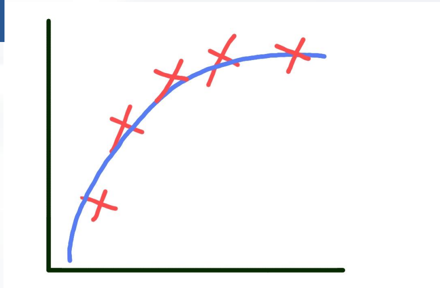
设置

Which fit is better?

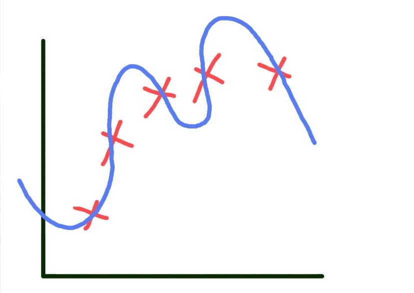
A



B



C



提交

Linear Discriminant Functions

- Two-category case

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 \Rightarrow \begin{cases} \text{Decide } \omega_1 \text{ if } g(\mathbf{x}) > 0 \\ \text{Decide } \omega_2 \text{ if } g(\mathbf{x}) < 0 \end{cases}$$

$$\Updownarrow$$

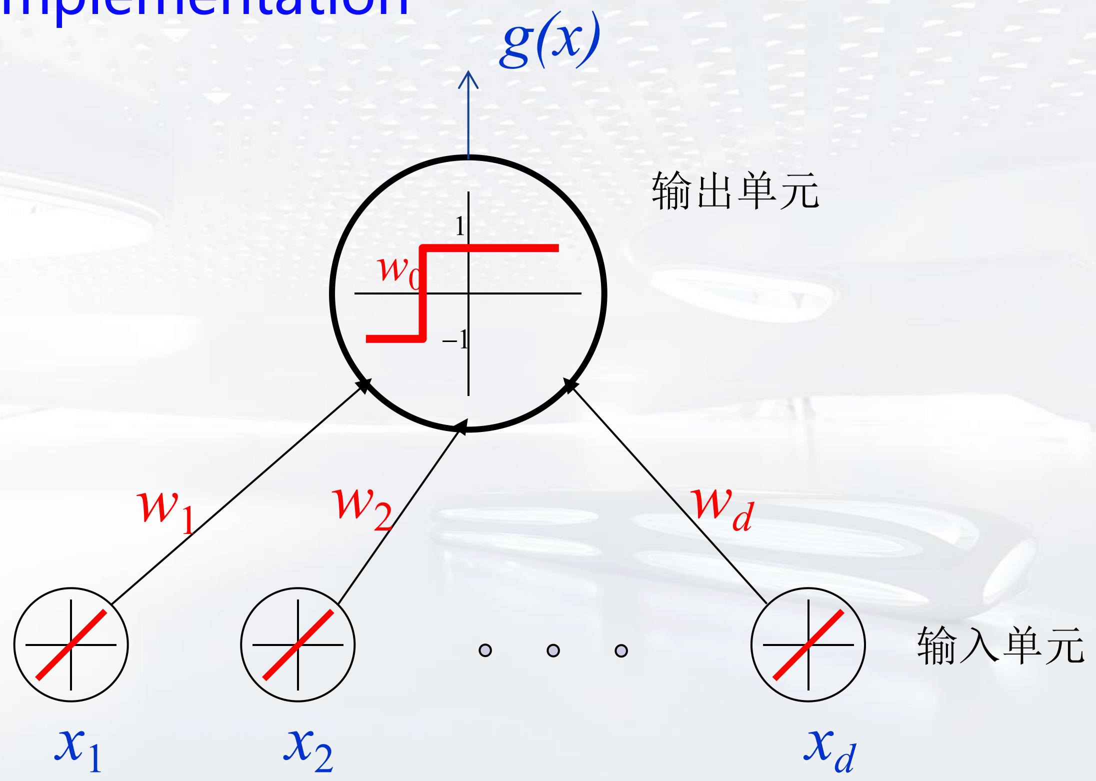
c=2 functions $\begin{cases} g_1(\mathbf{x}) = \mathbf{w}_1^T \mathbf{x} + w_{10} \\ g_2(\mathbf{x}) = \mathbf{w}_2^T \mathbf{x} + w_{20} \end{cases}$

Thus, it is suffices to consider only d+1 parameters (w and d) instead of 2(d+1) parameters under two-category case.

Linear Discriminant Functions

- Two-category case: Implementation

$$\begin{cases} \text{Decide } \omega_1 \text{ if } g(\mathbf{x}) > 0 \\ \text{Decide } \omega_2 \text{ if } g(\mathbf{x}) < 0 \end{cases}$$

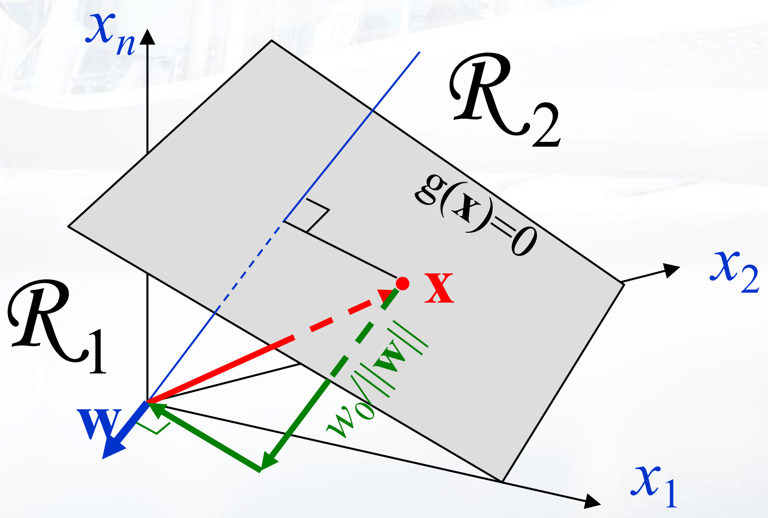


Decision Surface

判定面

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

$$g(\mathbf{x}) = 0$$



$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \frac{\mathbf{w}^T\mathbf{w}}{\|\mathbf{w}\|^2} w_0 = 0$$

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

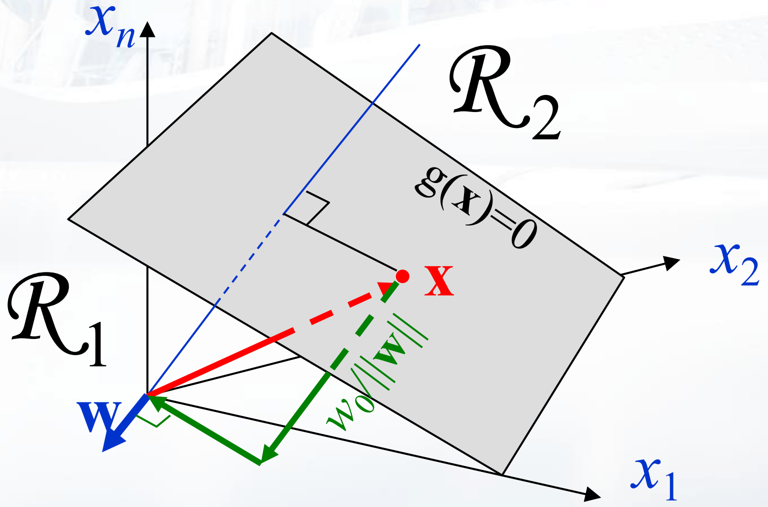
Decide ω_1 if $g(\mathbf{x}) > 0$

Decide ω_2 if $g(\mathbf{x}) < 0$

Decision Surface

判定面

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

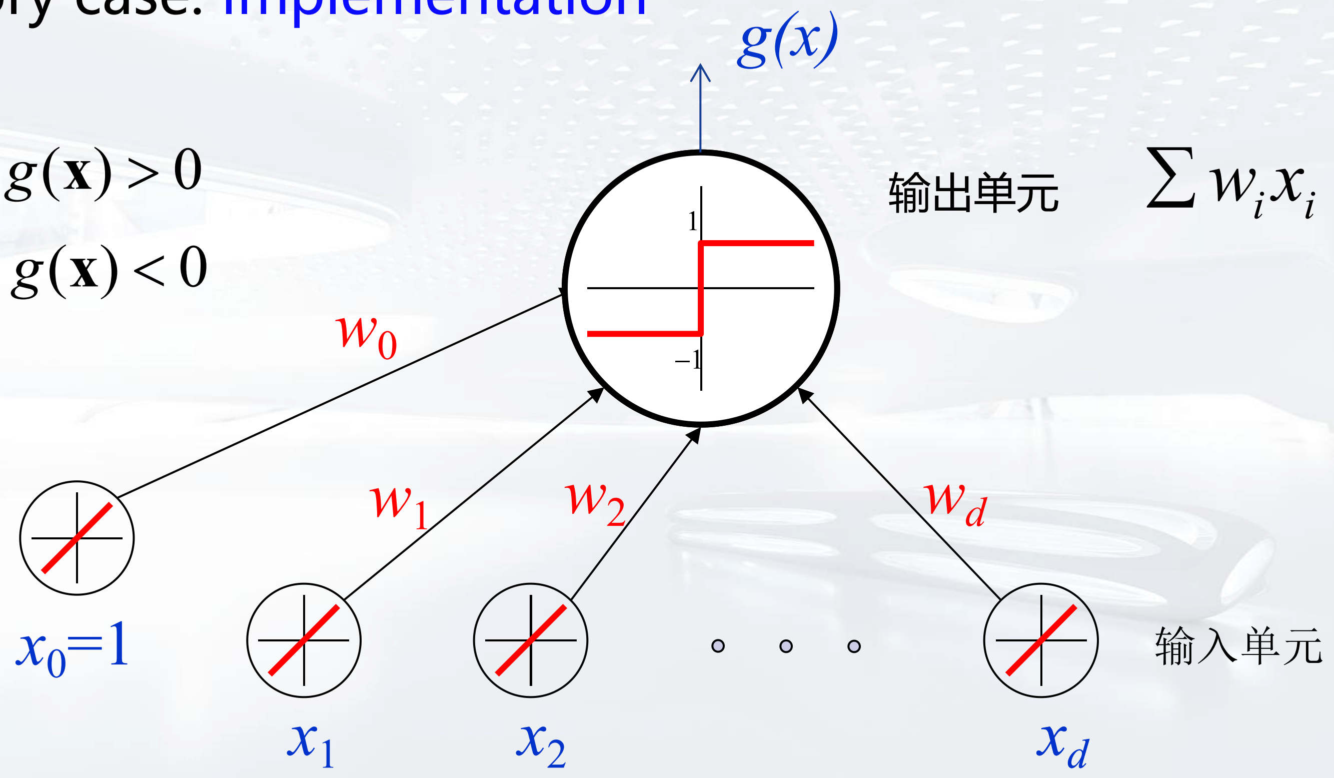


1. A linear discriminant function divides the feature space by a *hyperplane*.
2. The orientation of the surface is determined by the normal vector $\mathbf{w}$.
3. The location of the surface is determined by the bias w_0.

Linear Discriminant Functions

- Two-category case: Implementation

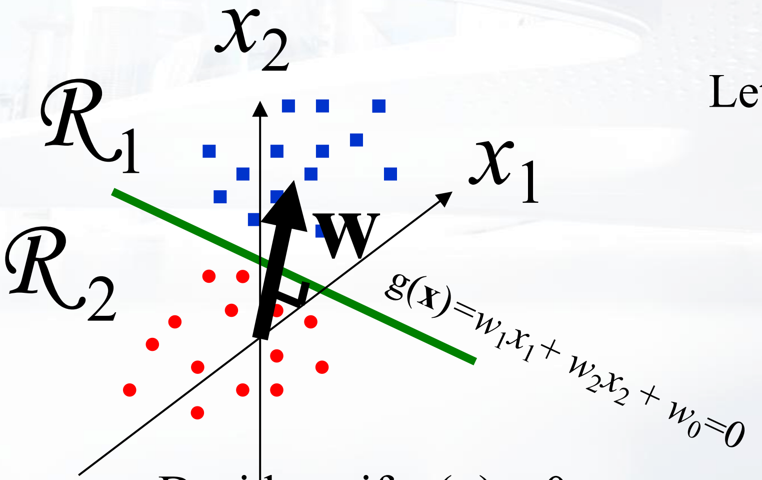
$$\begin{cases} \text{Decide } \omega_1 \text{ if } g(\mathbf{x}) > 0 \\ \text{Decide } \omega_2 \text{ if } g(\mathbf{x}) < 0 \end{cases}$$



Augmented Space

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

d-dimension



Decide ω_1 if $g(\mathbf{x}) > 0$

Decide ω_2 if $g(\mathbf{x}) < 0$

$$\underbrace{\text{Let } \mathbf{a} = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_d \end{bmatrix}}_{(d+1)\text{-dimension}}$$

增广权向量

增广特征向量

Decision surface: $\mathbf{a}^T\mathbf{y} = 0$

Decide ω_1 if $\mathbf{a}^T\mathbf{y} > 0$

Decide ω_2 if $\mathbf{a}^T\mathbf{y} < 0$

Augmented Space

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

Augmented weight vector

Augmented feature vector

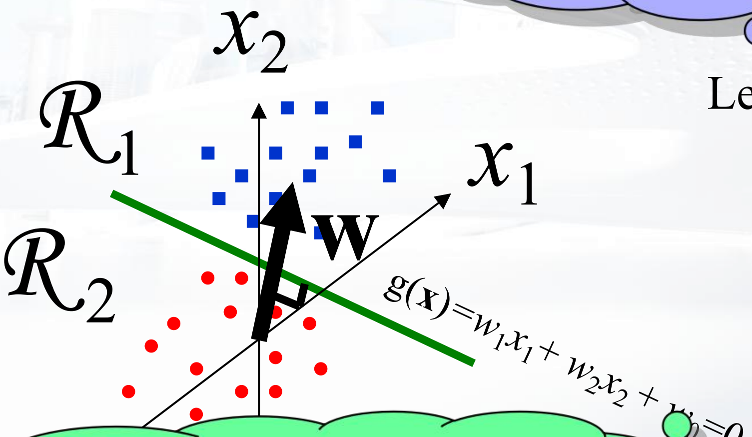
dimension

Let $\mathbf{a} = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{bmatrix}$ $\quad \mathbf{y} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_d \end{bmatrix}$

$(d+1)$-dimension

增广权向量

增广特征向量



Pass through the origin of augmented space

Decision surface: $\mathbf{a}^T\mathbf{y} = 0$

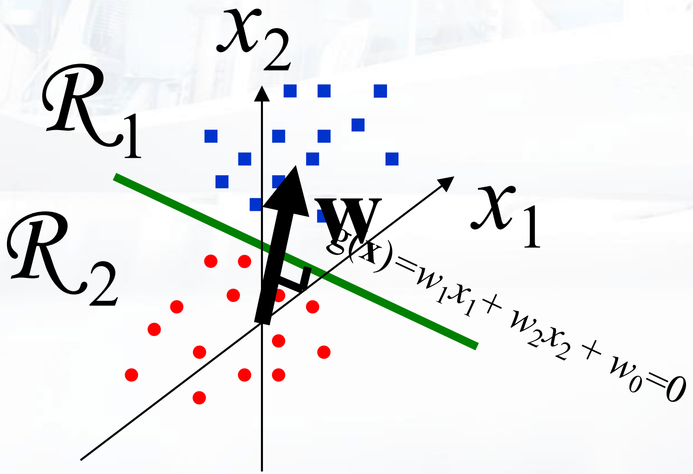
Decide ω_1 if $\mathbf{a}^T\mathbf{y} > 0$

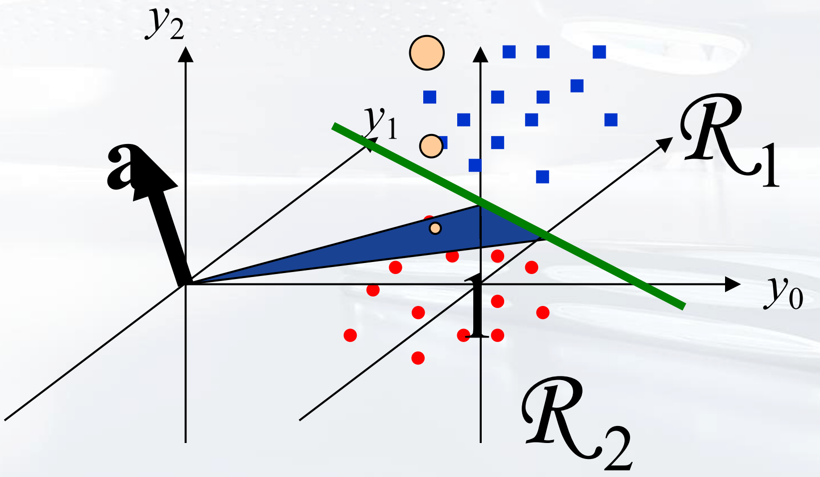
Decide ω_2 if $\mathbf{a}^T\mathbf{y} < 0$

Augmented Space

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

$$\mathbf{ay} = 0$$





Augmented Space

- Decision surface in feature space:

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = 0 \quad \Rightarrow$$

Pass through the origin only when w_0=0.

- Decision surface in augmented space:

$$g(\mathbf{x}) = \mathbf{a}^T\mathbf{y} = 0 \quad \Rightarrow$$

Always pass through the origin.

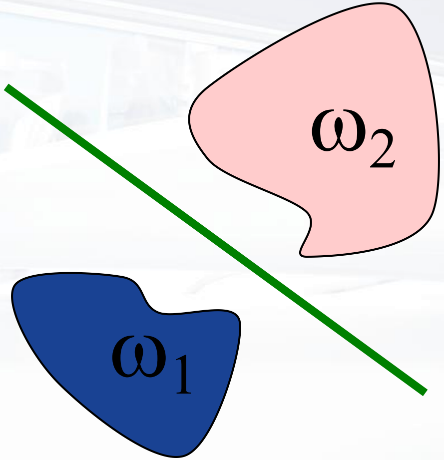
$$\mathbf{a} = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}$$

By using this mapping, the problem of finding weight wector w and threshold w0 is reduced to finding a single vector a.

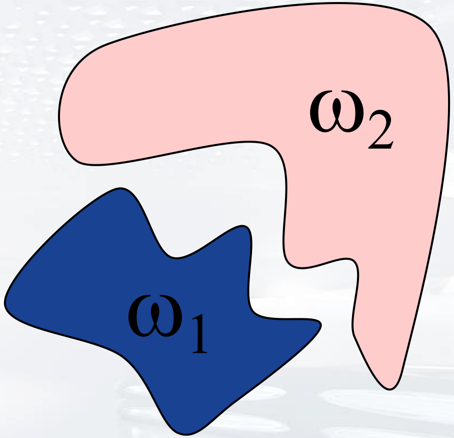
Linear Separability

- Two-Category Case

线性可分/不可分



Linearly Separable

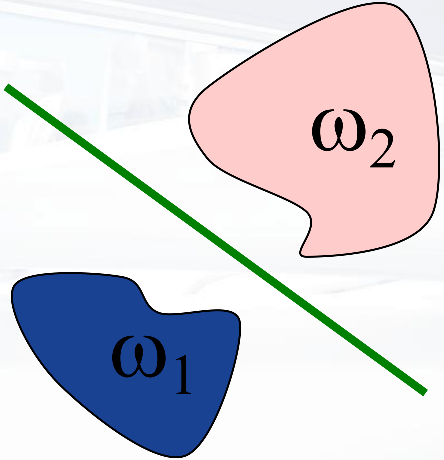


Not Linearly Separable

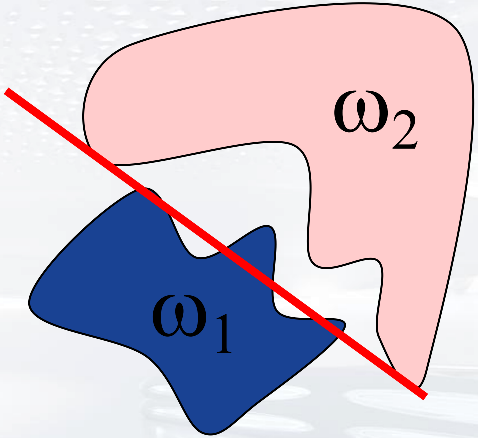
Linear Separability

- Two-Category Case

线性可分/不可分



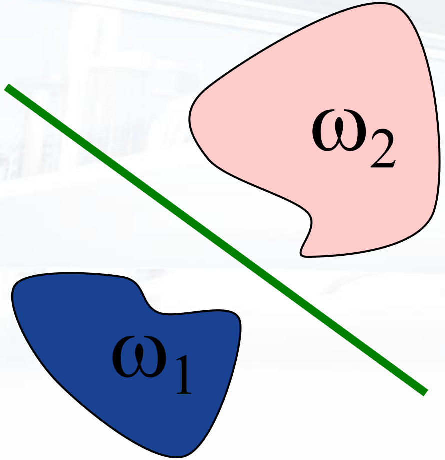
Linearly Separable



Not Linearly Separable

Linear Separability

- Two-Category Case



Linearly Separable

Given a set of samples $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$, some labeled ω_1 and some labeled ω_2,

if there exists a vector $\mathbf{a}$ such that

$$\mathbf{a}^T\mathbf{y}_i > 0 \quad \text{if } \mathbf{y}_i \text{ is labeled } \omega_1$$

$$\mathbf{a}^T\mathbf{y}_i < 0 \quad \text{if } \mathbf{y}_i \text{ is labeled } \omega_2$$

then the samples are said to be

Linearly Separable

Linear Separability

- Two-Category Case

Given a set of samples $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$, some labeled ω_1 and some labeled ω_2,

if there exists a vector $\mathbf{a}$ such that

$$\mathbf{a}^T\mathbf{y}_i > 0 \qquad \text{if } \mathbf{y}_i \text{ is labeled } \omega_1$$

$$\mathbf{a}^T\mathbf{y}_i < 0 \qquad \text{if } \mathbf{y}_i \text{ is labeled } \omega_2$$

then the samples are said to be

Linearly Separable

Withdrawing all labels of samples and replacing the ones labeled ω_2 by their *negatives*, it is equivalent to find a vector $\mathbf{a}$ such that

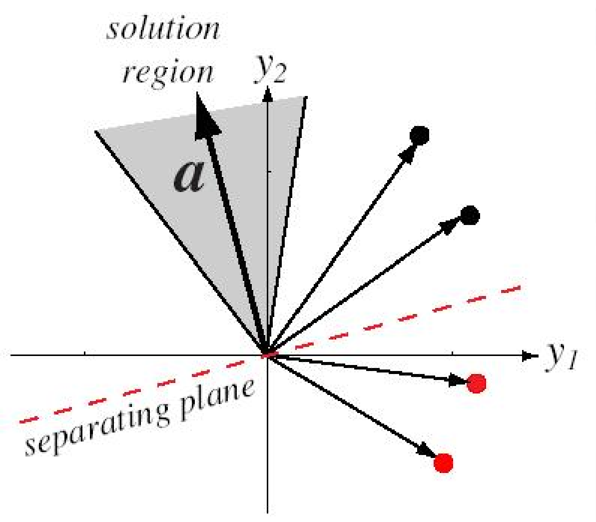
$$\boxed{\mathbf{a}^T\mathbf{y}_i > 0 \quad \forall i}$$

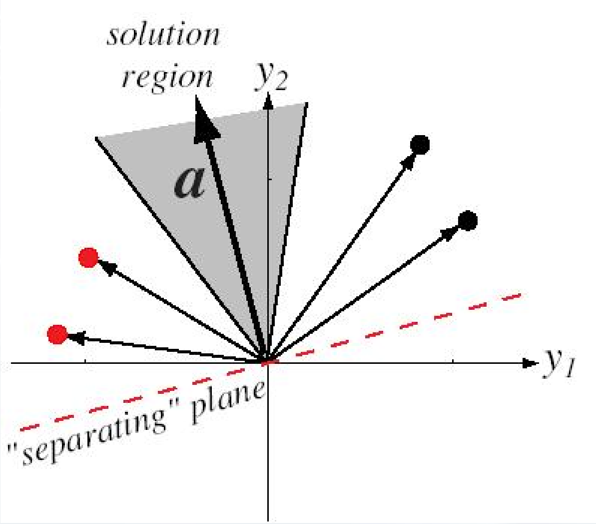
normalization

Solution Region in **Feature** Space

Separating Plane:

$$a_1 y_1 + a_2 y_2 + \cdots + a_n y_n = 0$$

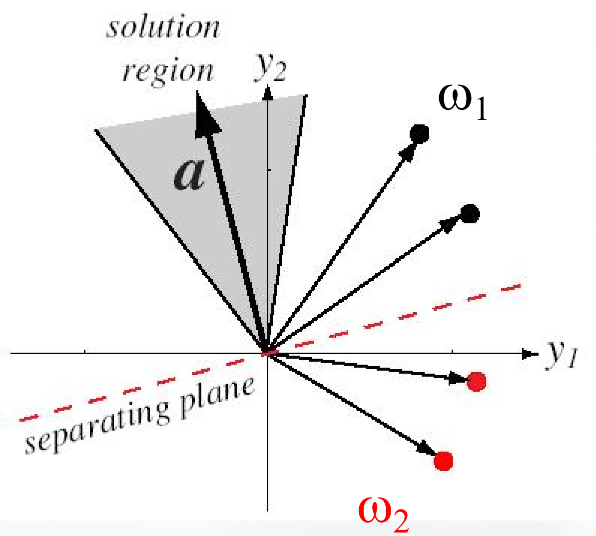




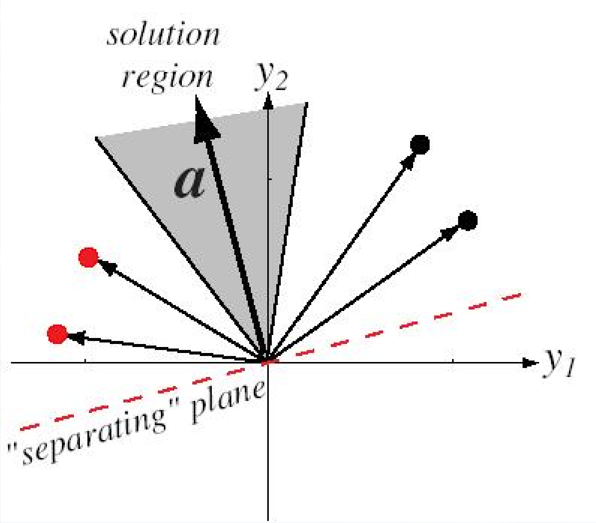
Normalized Case

Solution Region in Feature Space

Separating Plane: $\boxed{a_1 y_1 + a_2 y_2 + \cdots + a_n y_n = 0}$



normalization

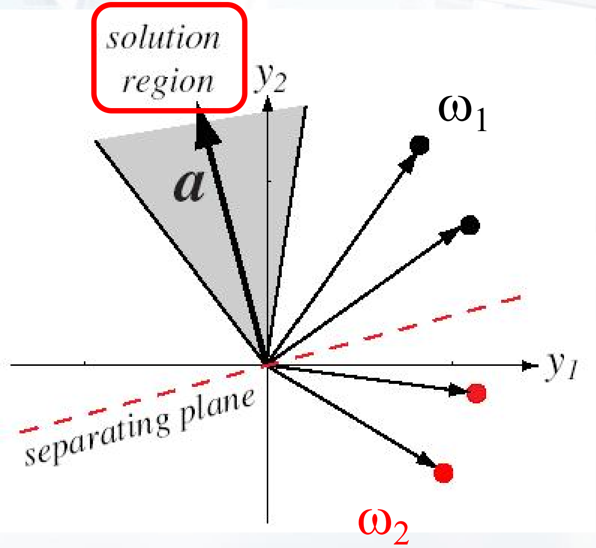


Normalized Case

Solution Region in Feature Space

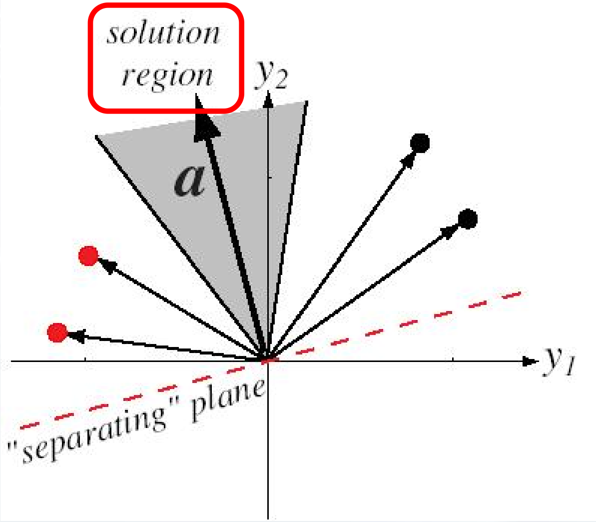
分离平面 *Separating Plane:*

$$a_1 y_1 + a_2 y_2 + \cdots + a_n y_n = 0$$



normalization

规范化

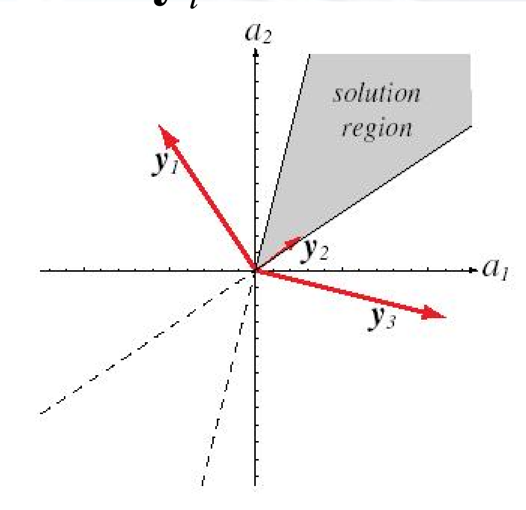


Normalized Case

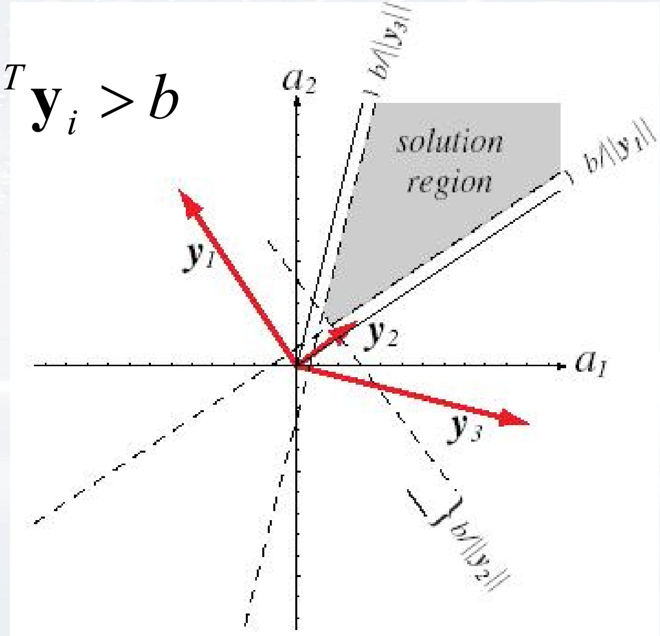
Solution Region in Weight Space

- Solution Region in Weight Space

$$\mathbf{a}^T\mathbf{y}_i > 0$$



$$\mathbf{a}^T\mathbf{y}_i > b$$



Shrink solution region by margin

$$b/\|\mathbf{y}_i\|, \quad b > 0$$

边沿裕量/间隔

Linear Discriminant Functions

How to learn the weights?

Criterion Function

- To facilitate learning, we usually define a scalar *criterion function*.
- It usually represents the *penalty* or *cost* of a solution.
- Our goal is to *minimize* its value.

$$J(\mathbf{w}, b) = -\sum_{i=1}^{n} \mathtt{sign}[\omega_i \cdot g(\mathbf{x}_i)]$$

$$J(\mathbf{w}, b) = -\sum_{i=1}^{n} \omega_i \cdot g(\mathbf{x}_i)$$

$$J(\mathbf{w}, b) = \sum_{i=1}^{n} (g(\mathbf{x}_i) - \omega_i)^2$$

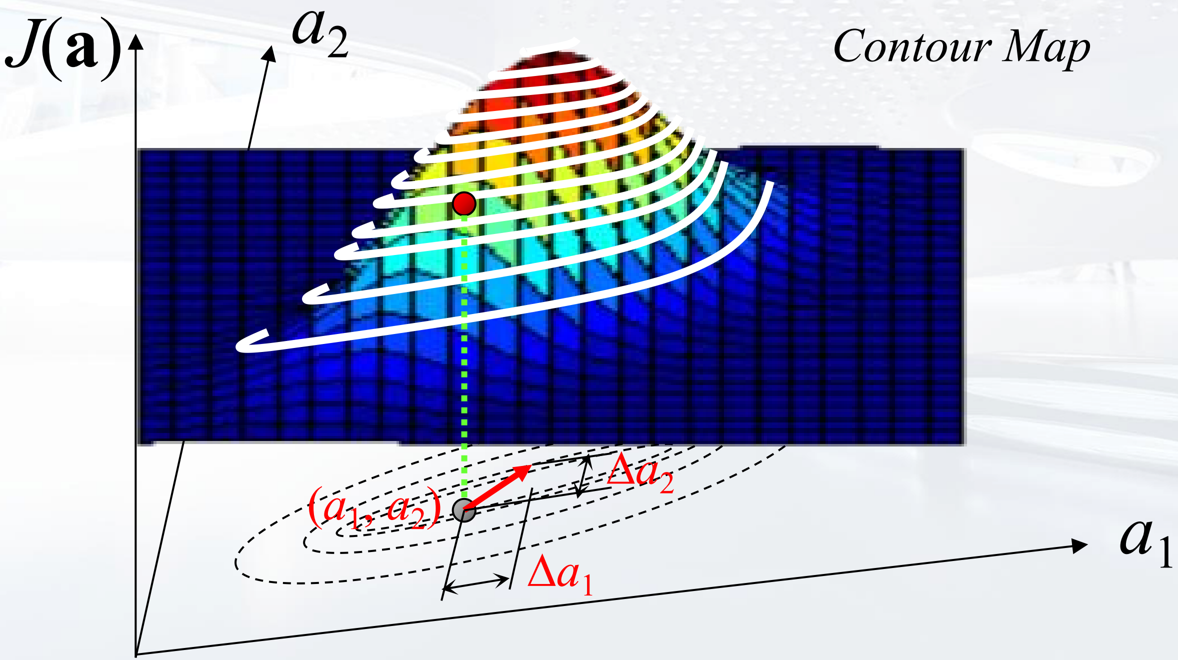
......

- *Function optimization*.

How to minimize the criterion function?

Gradient Decent Algorithm

- Our goal is to go *downhill*

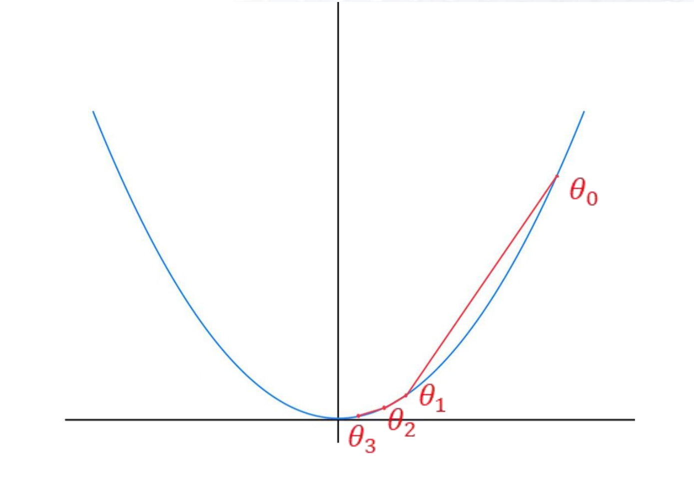


Example

$$J(\theta) = \theta^2$$

$$\nabla J(\theta) = 2\theta$$

$$\alpha = 0.4 \;\; \theta_0 = 1$$



$$\theta_1 = \theta_0 - \alpha \nabla J(\theta)$$

$$\theta_1 = 0.2$$
$$\theta_2 = 0.04$$
$$\theta_3 = 0.008$$

Gradient Decent Algorithm

- Taylor Expansion

$$f(x+\Delta x) = f(x) + \nabla f(x)^{\mathrm{T}} \cdot \Delta x + O(\Delta x^{\mathrm{T}} \cdot \Delta x)$$

$f: R^d \rightarrow R:$	A real-valued *d*-variate function
$x \in R^d:$	A point in the *d*-dimensional Euclidean space
$\Delta x \in R^d:$	A small shift in the *d*-dimensional Euclidean space
$\nabla f(x):$	gradient of *f*(.) at *x*
$O(\Delta x^T \cdot \Delta x):$	The big oh order of $\Delta x^T \cdot \Delta x$

Gradient Decent Algorithm

- Taylor Expansion

$$f(x+\Delta x) = f(x) + \nabla f(x)^{\mathrm{T}} \cdot \Delta x + O(\Delta x^{\mathrm{T}} \cdot \Delta x)$$

What happens if we set Δx to be negatively proportional to the gradient at x, i.e.,

$$\Delta x = -\eta \cdot \nabla f(x) \quad (\eta \text{ being a small positive scalar})$$

$$f(x+\Delta x) = f(x) - \eta \underbrace{\nabla f(x)^{t} \cdot \nabla f(x)}_{\text{being non-negative}} + \underbrace{O(\Delta x^{t} \cdot \Delta x)}_{\text{ignored when it is small}}$$

There, we have $f(x+\Delta x) <= f(x)$

Gradient Decent Algorithm

- Basic strategy
  - To minimize some function f(.), the general gradient descent techniques work in the following iterative way:

1. Set learning rate >0, and a small threshold >0.
2. Randomly initialize $x_0 \in R^d$ as the starting point; set k=0.
3. do $k=k+1$
4. $$x_k = x_{k-1} - \eta \cdot \nabla f(x_{k-1})$$
5. until
6. Return x_k and $f(x_k)$

Gradient Decent Algorithm

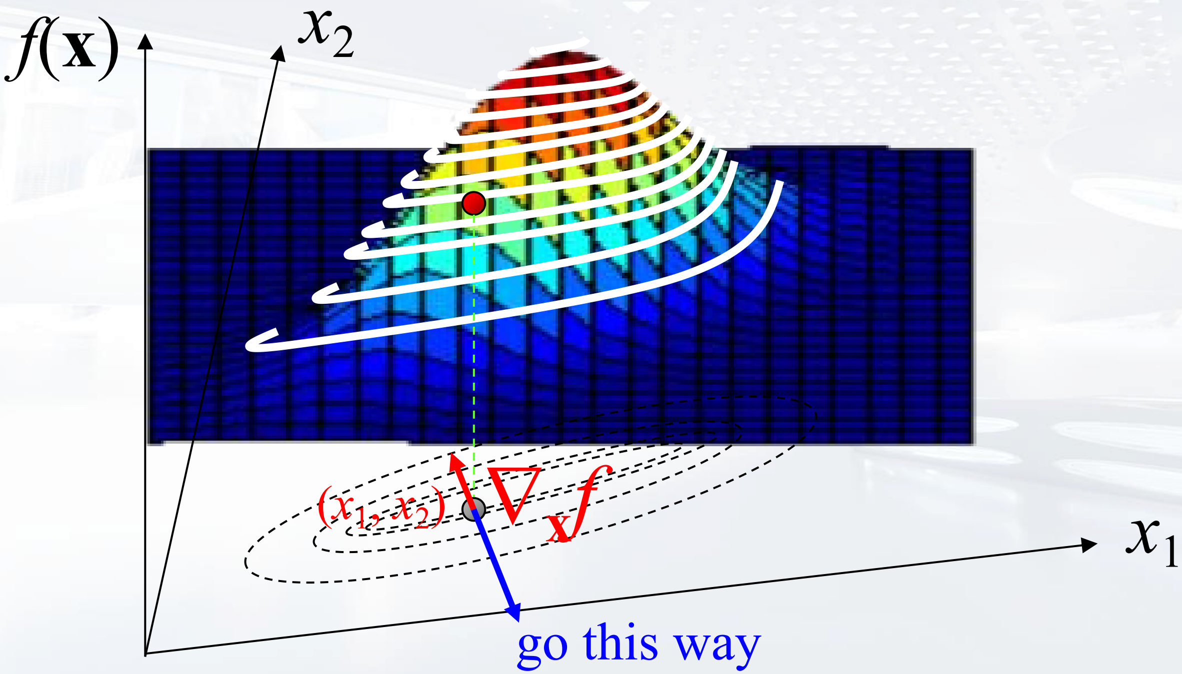
Why the negative gradient direction?

$$\nabla_x = \left(\frac{\partial}{\partial x_1} \quad \frac{\partial}{\partial x_2} \quad \cdots \quad \frac{\partial}{\partial x_d} \right)^T$$

$$df = (\nabla_x f)^T dx \begin{cases} \text{steepest if } \overrightarrow{dx} = \overrightarrow{\nabla_x f} \\ = 0 \text{ if } \overrightarrow{dx} \perp \overrightarrow{\nabla_x f} \\ \text{steepest decent if } \overrightarrow{dx} = -\overrightarrow{\nabla_x f} \end{cases}$$

Gradient Decent Algorithm

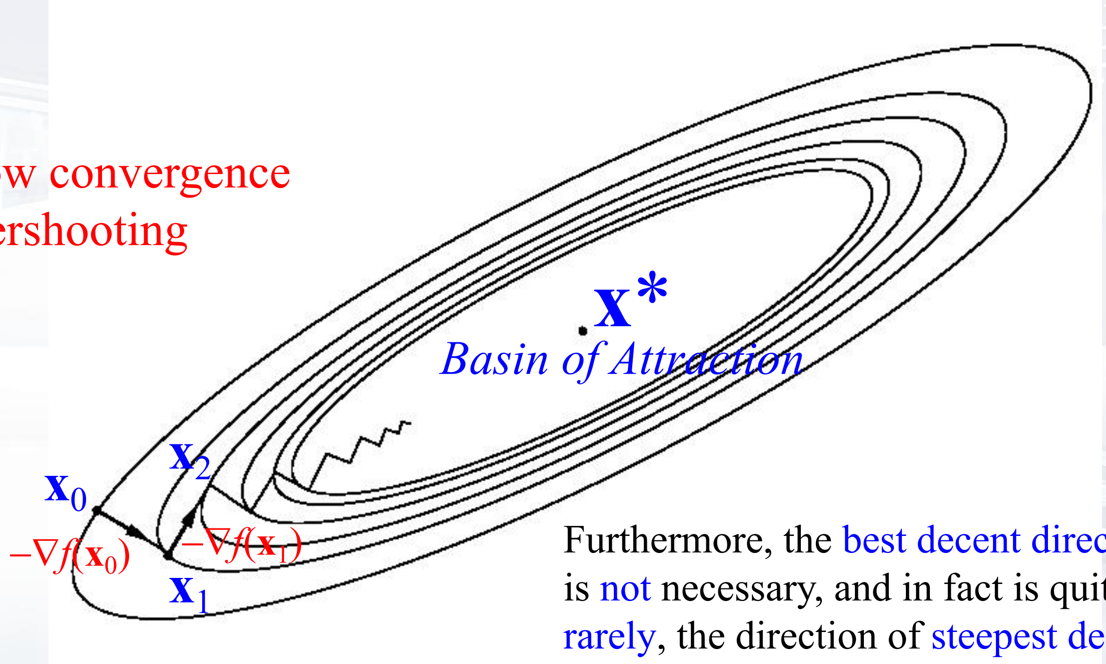
How long a step shall we take?



Gradient Decent Algorithm

If improper *learning rate* (η_k) is used, the convergence rate may be poor.

1. Too small: slow convergence
2. Too large: overshooting



Furthermore, the best decent direction is not necessary, and in fact is quite rarely, the direction of steepest decent.

Newton's Method

- Global minimum of a Paraboloid

Paraboloid $f(\mathbf{x}) = c + \mathbf{a}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x}$

We can find the global minimum of a paraboloid by setting its gradient to zero.

$$\nabla_{\mathbf{x}} f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k} = \mathbf{a} + \mathbf{Q}\mathbf{x}_k = 0$$

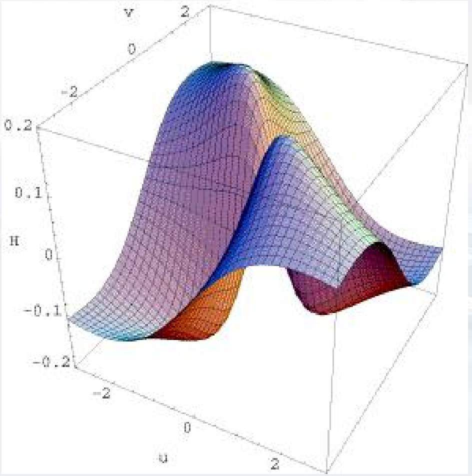
$$\mathbf{x}^* = -\mathbf{Q}^{-1}\mathbf{a}$$

Newton's Method

$$f(\mathbf{x}_k + \Delta\mathbf{x}) \approx f(\mathbf{x}_k) + \mathbf{g}_k^T \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \mathbf{Q}_k \Delta\mathbf{x}$$

Taylor Series Expansion

All smooth functions can be approximated by paraboloids in a sufficiently small neighborhood of any point.



Newton's Method

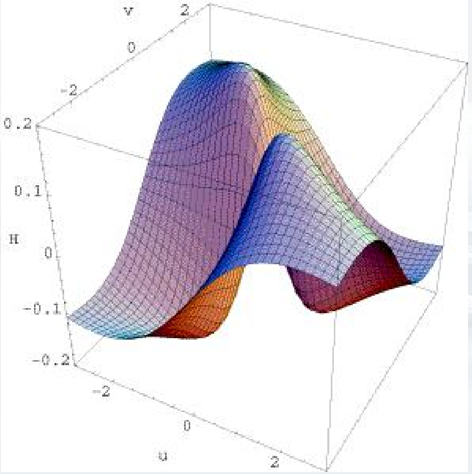
$$f(\mathbf{x}_k + \Delta\mathbf{x}) \approx f(\mathbf{x}_k) + \mathbf{g}_k^T \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \mathbf{Q}_k \Delta\mathbf{x}$$

$$\mathbf{g}_k = \nabla_{\mathbf{x}} f(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_k}$$

黑塞矩阵
海森矩阵
海瑟矩阵
海塞矩阵

Hessian Matrix

$$\mathbf{Q}_k = \frac{\partial^2 f}{\partial\mathbf{x}\partial\mathbf{x}^T}\Big|_{\mathbf{x}=\mathbf{x}_k} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_d x_1} & \cdots & \frac{\partial^2 f}{\partial x_d^2} \end{bmatrix}_{\mathbf{x}=\mathbf{x}_k}$$

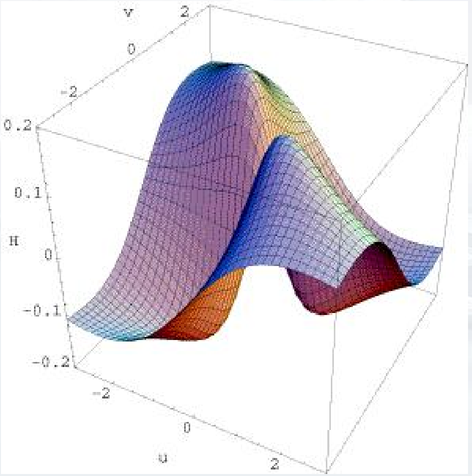


Newton's Method

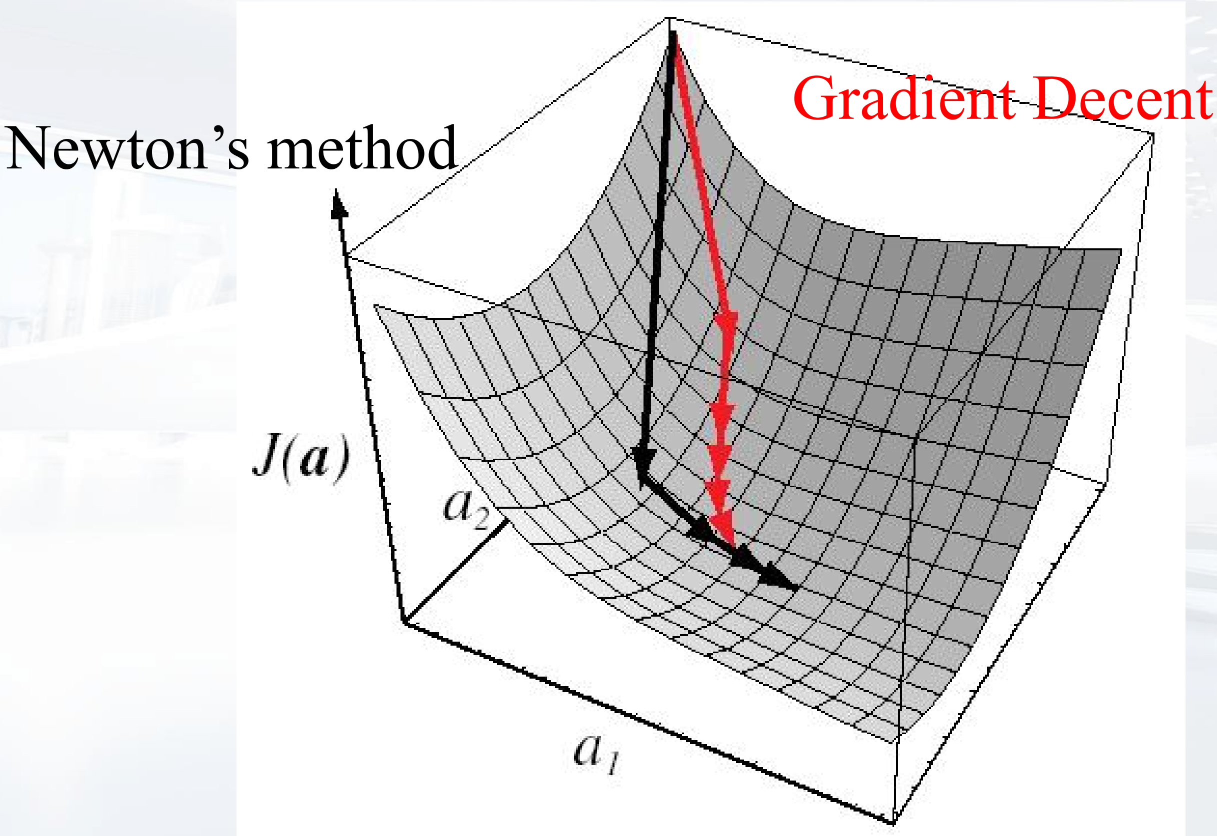
$$f(\mathbf{x}_k+\Delta\mathbf{x}) \approx f(\mathbf{x}_k)+\mathbf{g}_k^T\Delta\mathbf{x}+\frac{1}{2}\Delta\mathbf{x}^T\mathbf{Q}_k\Delta\mathbf{x}$$

Set $\nabla_{\Delta\mathbf{x}} f = \mathbf{g}_k + \mathbf{Q}_k\Delta\mathbf{x} = 0$

$$\Rightarrow \mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{Q}_k^{-1}\mathbf{g}_k = 0$$



Comparison



Comparison

- *Newton's Method* will usually give a greater improvement *per step* than the *simple gradient decent algorithm*, even with optimal value of η_k.
- However, Newton's Method is not applicable if the Hessian matrix **Q** is *singular*.
- Even when **Q** is nonsingular, compute **Q** is time consuming $O(d^3)$.
- It often takes less time to set η_k to a constant (small than necessary) than it is to compute the optimum η_k at each step.

Summary

- Discriminant functions
- Linear Discriminant Functions and Decision Surface
  - The general setting, one function for each class
  - The two-category case
  - Minimization of criterion/objection function
- Linear Separability

Summary

- Learning
  - Gradient descent

$$x_k = x_{k-1} - \eta \cdot \nabla f(x_{k-1})$$

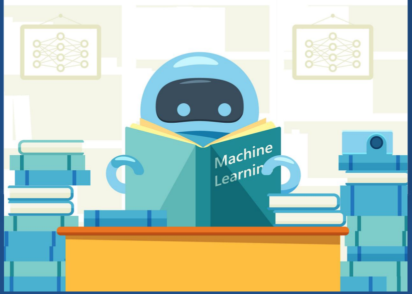
  - Newton’ s method

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{Q}_k^{-1}\mathbf{g}_k = 0$$

山东大学软件学院
SCHOOL OF SOFTWARE, SHANDONG UNIVERSITY

Machine Learning

机器学习



感谢同学们！

luoxin@sdu.edu.cn

软件学院办公楼-425