



模式识别 (6-7)

线性分类器

左旺孟

哈尔滨工业大学计算机学院
机器学习研究中心 综合楼712

cswmzuo@gmail.com

13134506692

两类判别分类器

- 基于最小错误率判别准则的贝叶斯分类器
 - 以**两类**问题为例

$$\begin{cases} \text{如果 } P(\omega_1|\mathbf{x}) > P(\omega_2|\mathbf{x}), & \mathbf{x} \in \omega_1 \\ \text{如果 } P(\omega_1|\mathbf{x}) < P(\omega_2|\mathbf{x}), & \mathbf{x} \in \omega_2 \end{cases} \longleftrightarrow F(\mathbf{x}) = \log \left(\frac{P(\omega_1|\mathbf{x})}{P(\omega_2|\mathbf{x})} \right) \begin{cases} \geq 0, & \mathbf{x} \in \omega_1 \\ < 0, & \mathbf{x} \in \omega_2 \end{cases}$$



2. 判别学习：根据训练样本对**类别后验概率**进行建模

判别函数： $g_i(\mathbf{x}) = P(\omega_i|\mathbf{x})$



1. 判别函数：根据训练样本学习直接学习**函数** $F(\mathbf{x})$

判别函数： $g(\mathbf{x}) = F(\mathbf{x})$

判别分类器

- 线性判别分类器
 - 判别函数：感知器、线性支持向量机
 - 判别学习：Logistic回归
- 非线性判别分类器
 - 判别学习：kNN
 - 判别函数：神经网络
 - 核方法
 - 推广能力
- 两类问题 and 多类问题

SVM: 判别函数 -> 判别学习

- J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, 1999.
- H.T. Lin, C.J. Lin, R.C. Weng. A note on Platt's probabilistic outputs for support vector machines, 2007.
- 名词解释：频率学派、贝叶斯学派、Probabilistic
 - **K. Murphy, Machine Learning: A Probabilistic Perspective. MIT Press, 2012.**
 - **K. Murphy, Probabilistic Machine Learning: An Introduction. MIT Press, 2022.**

1. 线性判别函数

- 问题

- 基本概念：线性判别函数、线性判别分类器
- 两类问题
- 多类问题

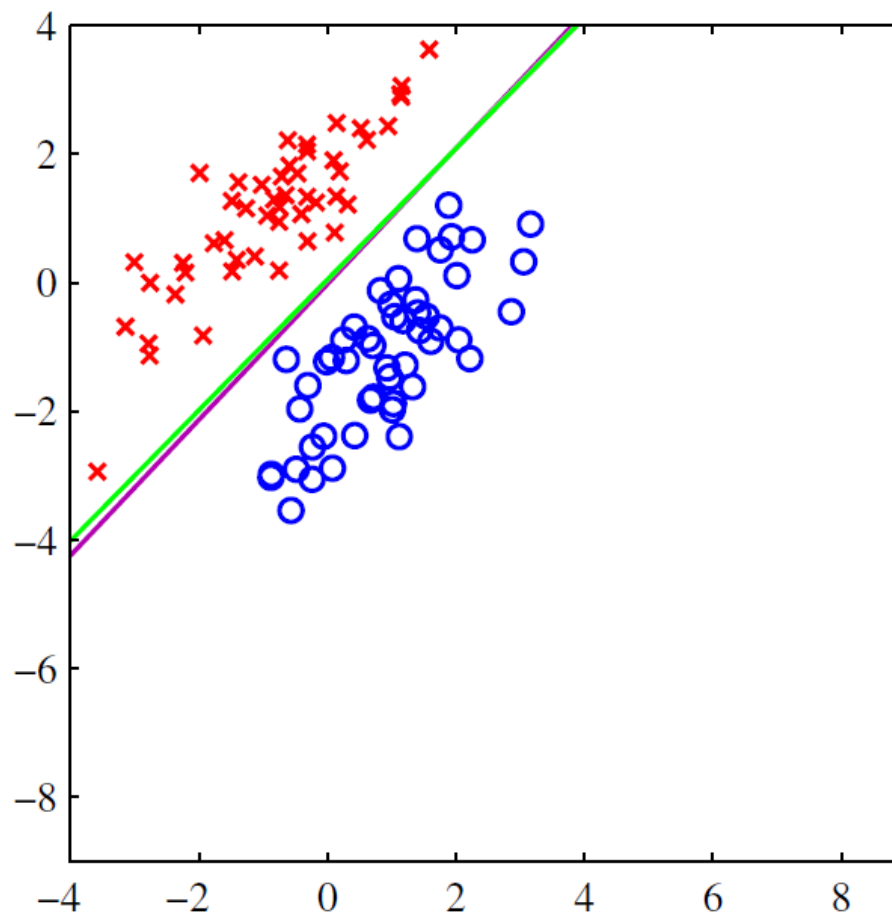
- 训练方法

- 感知器准则
- 线性支持向量机
- Logistic回归方法

线性判别分类器、判别函数、决策平面

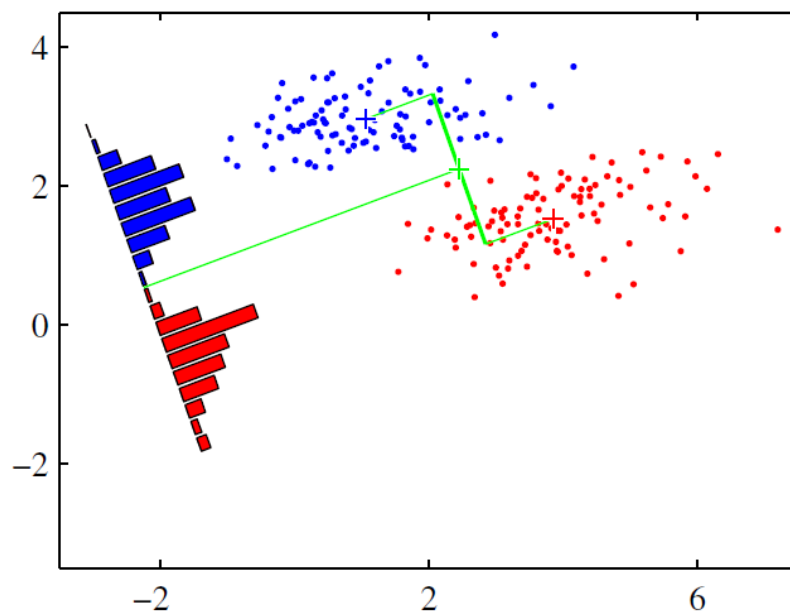
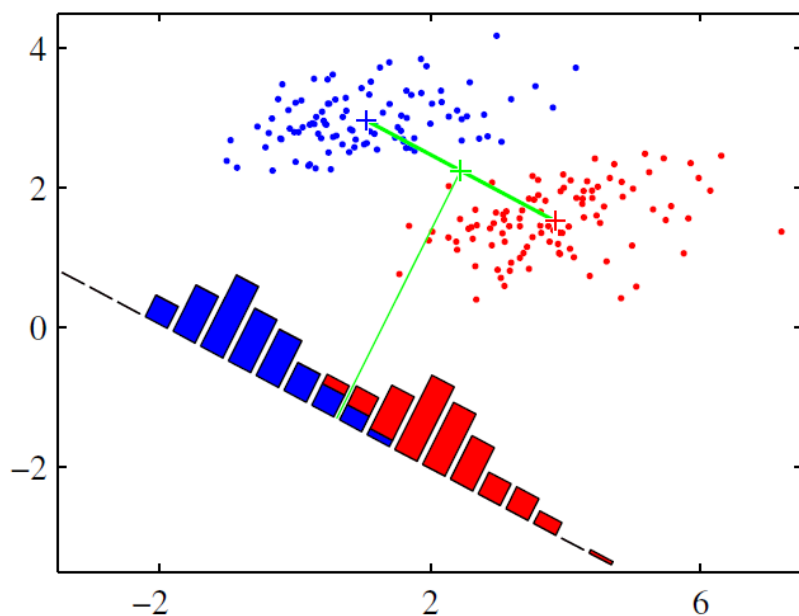
线性判别分类器

- 决策面是平面或超平面
- 判别函数是一个线性函数



线性判别分类器

- 判别函数是一个线性函数

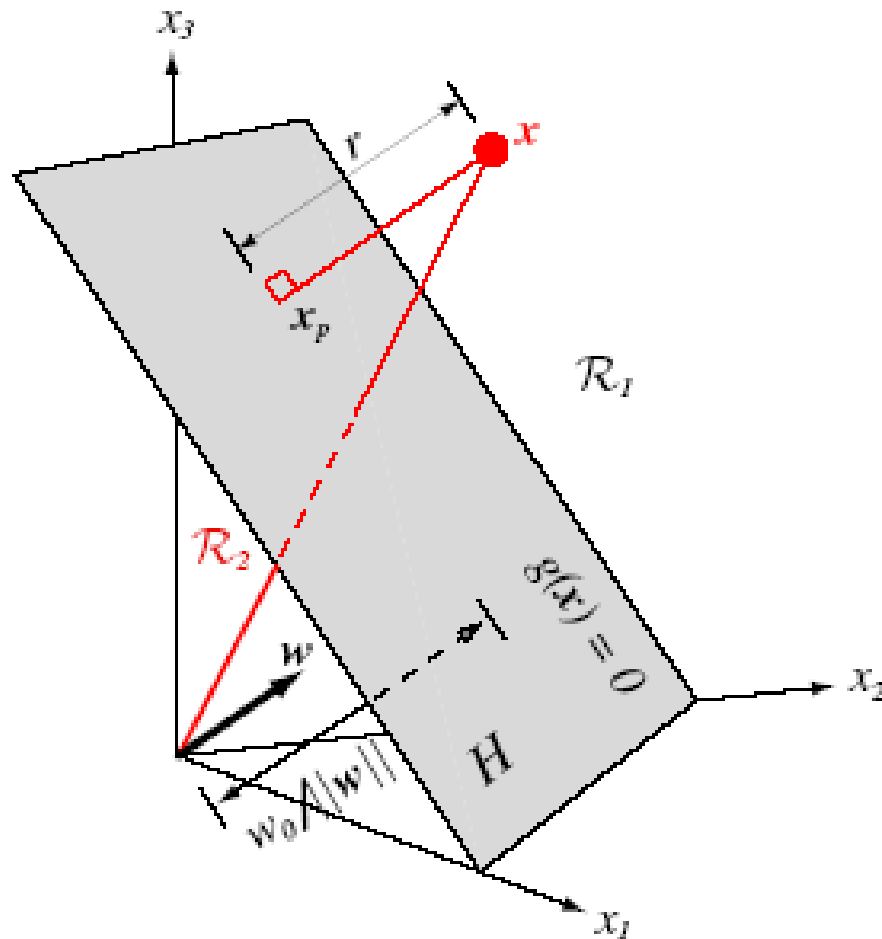


两类问题的线性判别函数

$$g(\mathbf{x}) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + w_0 = \mathbf{w}^t\mathbf{x} + w_0$$

- $\mathbf{x} = (x_1, x_2, \dots, x_n)^t$ 为待识别模式的特征向量;
- $\mathbf{w} = (w_1, w_2, \dots, w_n)^t$ 称为权向量。

分类界面（决策面）与权值



线性判别函数的增广形式

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

- $\mathbf{x} = (x_1, x_2, \dots, x_n, 1)^t$ 称为增广的特征向量(齐次);
- $\mathbf{w} = (w_1, w_2, \dots, w_n, w_0)^t$ 称为增广的权向量。

两类问题线性判别准则 ——线性判别分类器

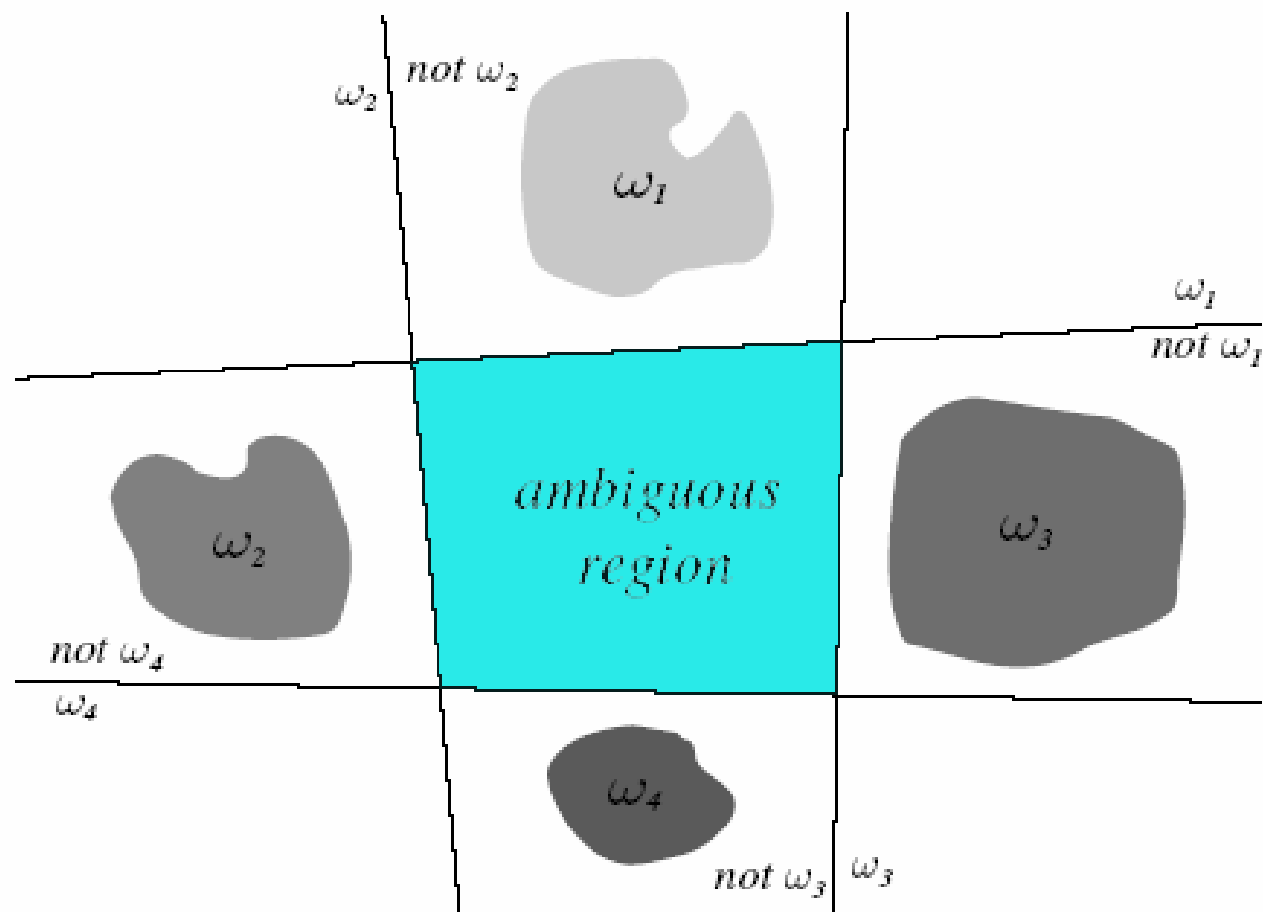
$$g(\mathbf{x}) = \mathbf{w}^t \mathbf{x} \begin{cases} > 0, & \mathbf{x} \in \omega_1 \\ < 0, & \mathbf{x} \in \omega_2 \\ = 0, & \text{拒识} \end{cases}$$

从两类到多类

多类问题（情况一）

- 每一类模式可以用一个超平面与其它类别分开；
- 这种情况可以把 C 个类别的多类问题分解为 C 个两类问题解决；

多类问题（情况一）



多类问题（情况一）判别规则

- 判别规则1

- 若存在 i ，使得 $g_i(x) > 0$ ， $g_j(x) < 0$ ， $j \neq i$ ，则判别 x 属于 ω_i 类；
- 其它情况，拒识。

- 判别规则2

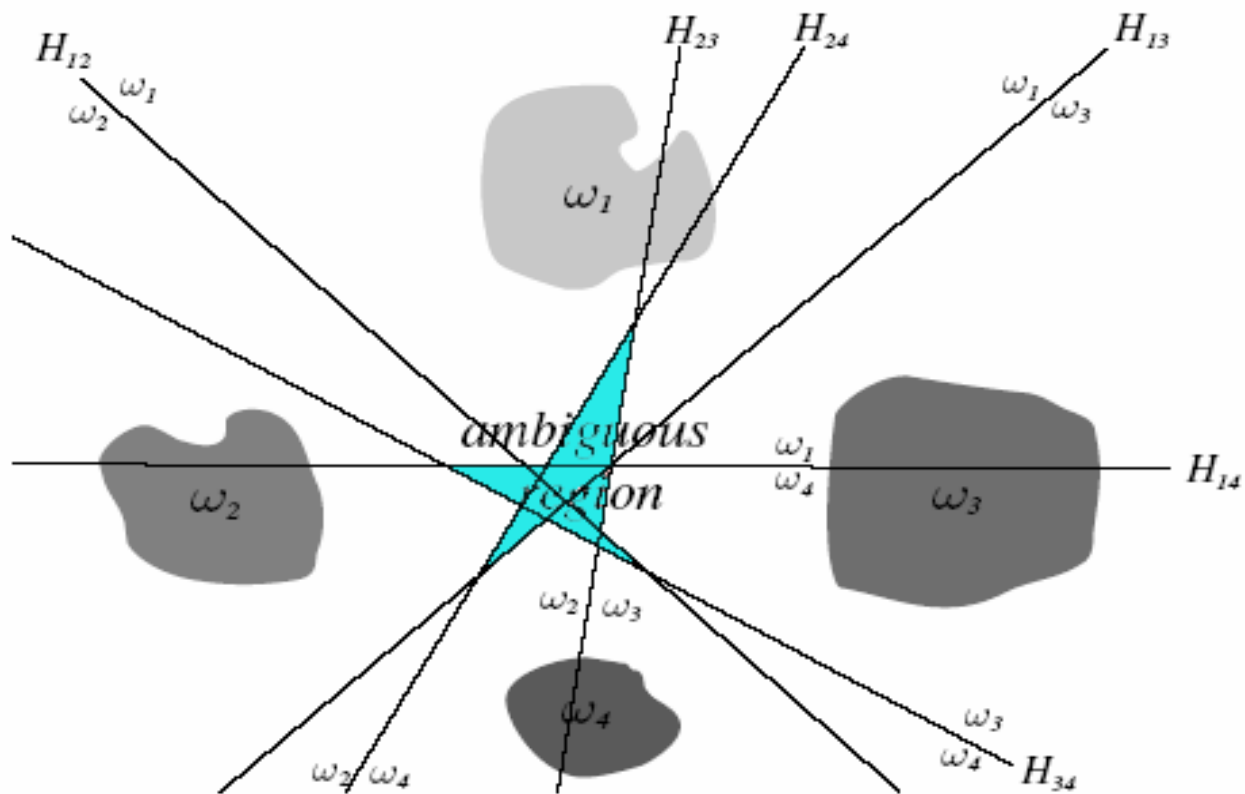
- $i = \arg \max g_j(x)$ ，如果 $g_i(x)$ 大于某一预先设定的值 T ，则判别 x 属于 ω_i 类；
- 如果 $g_i(x)$ 小于等于某一预先设定的值 T ，拒识

多类问题（情况二）

- 每两类之间可以用一个超平面分开，但是不能用来把其余类别分开；
- 需要将C个类别的多类问题转化为 $C(C-1)/2$ 个两类问题。
- 第i类与第j类之间的判别函数的为：

$$g_{ij}(\mathbf{x}) = \mathbf{w}_{ij}^t \mathbf{x} \quad i \neq j$$

多类问题（情况二）



多类问题（情况二）判别准则

- 判别准则1

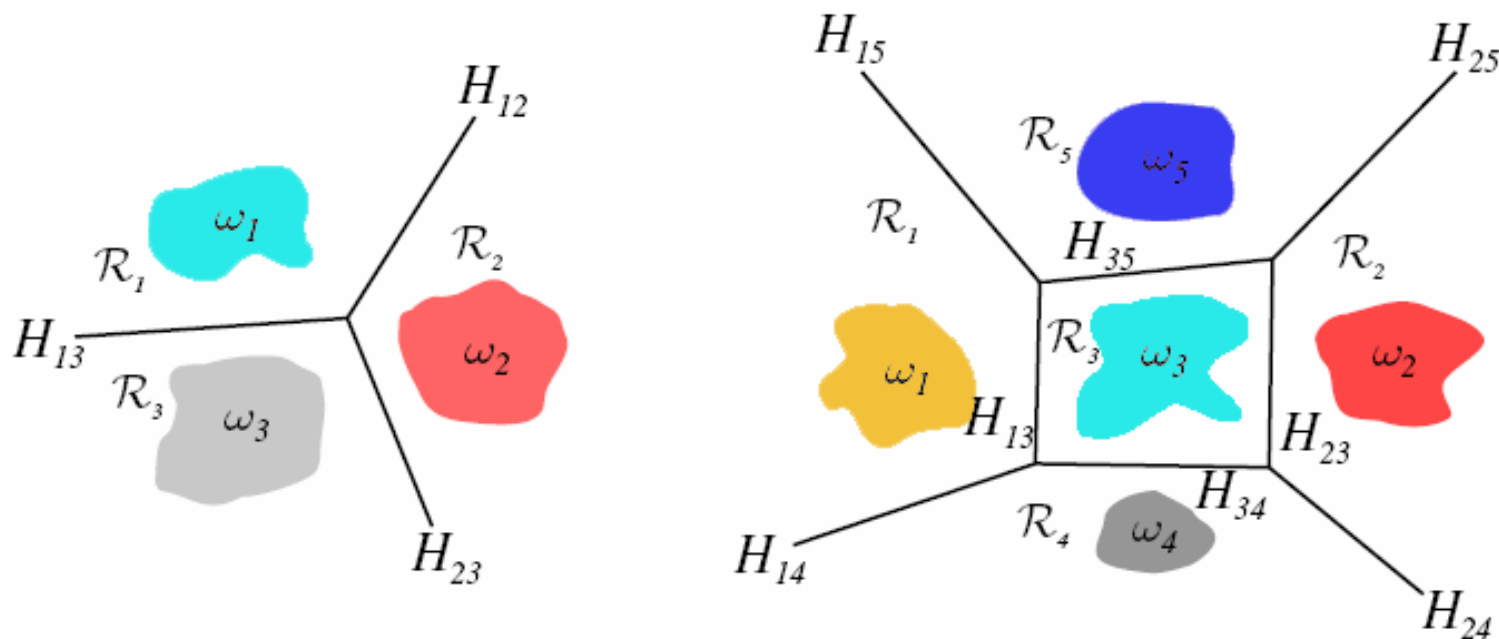
- 如果对任意 $j \neq i$ ，有 $g_{ij}(\mathbf{x}) \geq 0$ ，则决策 \mathbf{x} 属于 ω_i 。
- 其它情况，则拒识。

- 判别准则2

- $i = \arg \max_j \sum_{k=1, k \neq j}^C \text{sgn}(g_{jk}(\mathbf{x}))$
- 如果 $\sum_{k=1, k \neq j}^C \text{sgn}(g_{jk}(\mathbf{x}))$ 大于某一阈值 T ，则决策 \mathbf{x} 属于 ω_i 。
- 其它情况，则拒识。

多类问题（情况三）

- 情况三是情况二的特例，不存在拒识区域。



多类问题（情况三）判别函数

- C个类别需要C个线性函数：

$$g_i(\mathbf{x}) = \mathbf{w}_i^t \mathbf{x} = w_{i1}x_1 + w_{i2}x_2 + \cdots + w_{in}x_n + w_{i0}$$

∨ 判别准则：

$$g_i(\mathbf{x}) = \max_{1 \leq j \leq C} \{g_j(\mathbf{x})\} \quad \mathbf{x} \in \omega_i$$

课外阅读：

- Ryan Rifkin, Aldebaro Klautau. In Defense of One-Vs-All Classification, JMLR, 2004.
 - A simple “one-vs-all” scheme is **as accurate as** any other approach, assuming that the underlying binary classifiers are well-tuned regularized classifiers.

启示

- 基本问题：两类问题
- 多类问题
 - 可以转化为若干个两类问题
 - Reduction（约简）
 - 模式识别和机器学习中的一个重要概念
 - 针对某种学习算法，我们可能只会结合两类问题加以介绍，大家可以在实际应用中方便地推广到多类问题和复杂应用。
 - <http://hunch.net/~jl/projects/reductions/reductions.html>

两类线性判别函数的学习

- 判别分类器
 - 感知器算法
 - 支持向量机
- 判别学习
 - Logistic回归

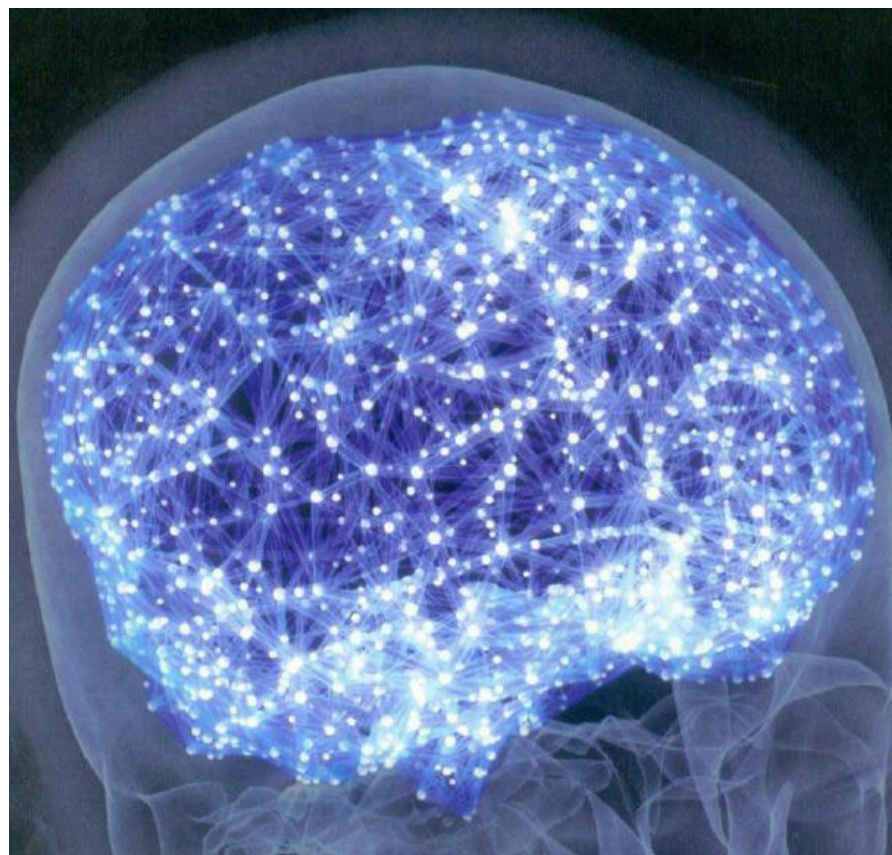
感知器算法

- 一、问题的表达
- 二、感知器算法
- 三、最小均方误差算法 (LMSE)

神经网络发展的三个阶段

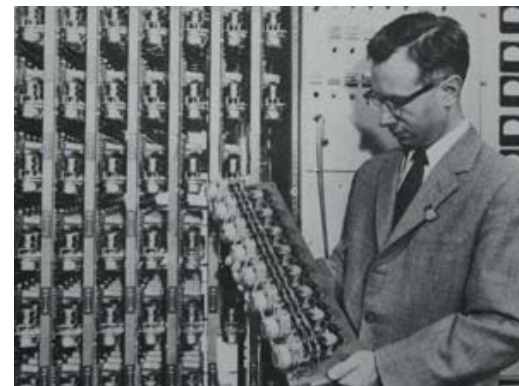
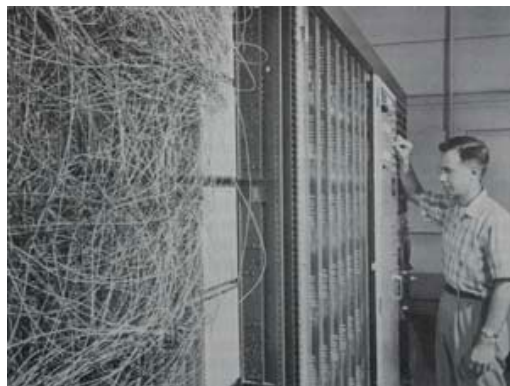
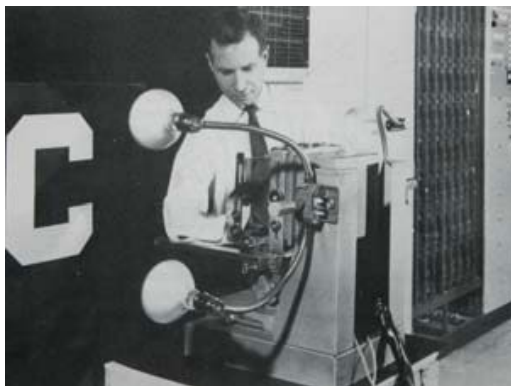
- 一、感知器
 - **Rosenblatt**
 - **Minski**
- 二、多层感知器感知器
 - **Hinton / LeCun**
 - **Minski**
- 三、深度学习（**Hinton、LeCun**）

人脑：神经元及其连接



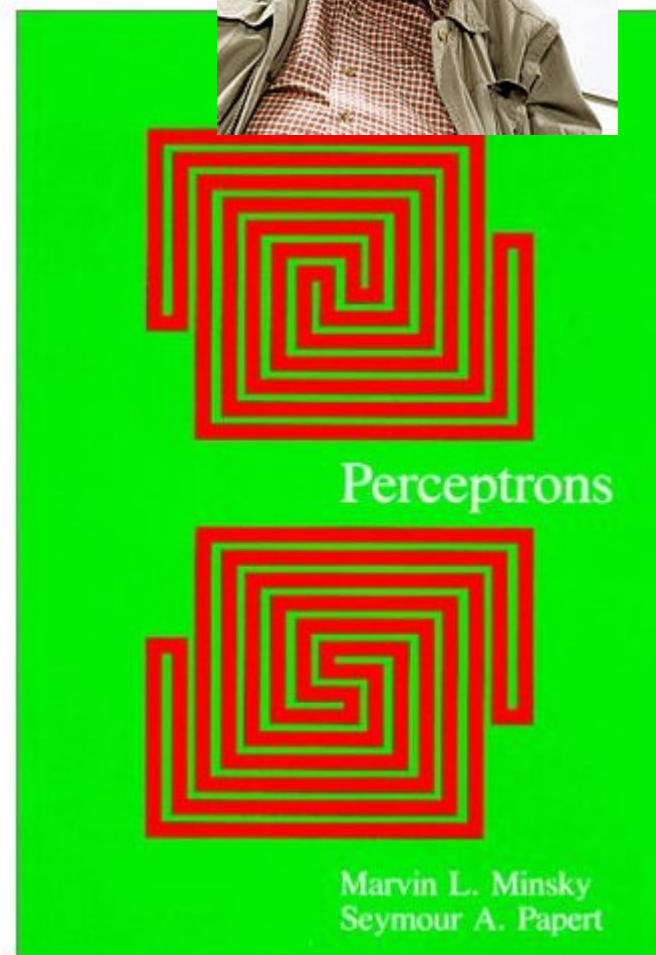
感知器算法: 历史

- 1957年, Rosenblatt提出感知器算法
- 线性 vs. 非线性



Perceptrons: An Introduction to Computational Geometry

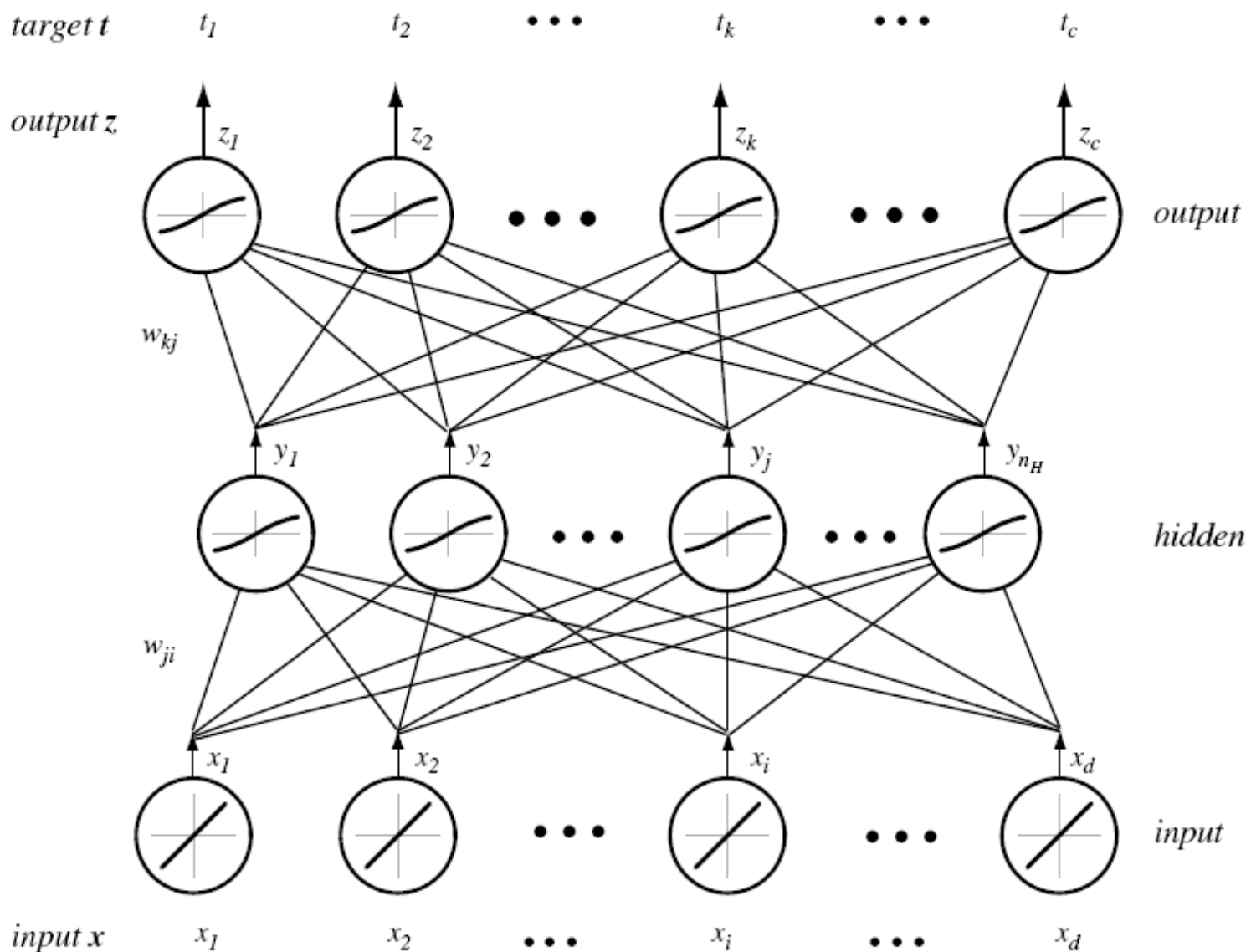
- 1969, 1987
- AI winter
- *Turing* Award, the *Japan* Prize, the *IJCAI* Award for Research Excellence, and the *Benjamin Franklin Medal, Fellow of the Computer History Museum*.
- An adviser on the movie *2001: A Space Odyssey* and is referred to in the movie.



AI Koan

- In the days when Sussman was a novice, Minsky once came to him as he sat hacking at the PDP-6.
- “What are you doing?” asked Minsky. “I am training a randomly wired neural net to play tic-tac-toe,” Sussman replied. “Why is the net wired randomly?” asked Minsky. Sussman replied, “I do not want it to have any preconceptions of how to play.”
- Minsky then shut his eyes. “Why do you close your eyes?” Sussman asked his teacher. “So that the room will be empty,” replied Minsky. At that moment, Sussman was enlightened.

多层感知器



Deep Learning

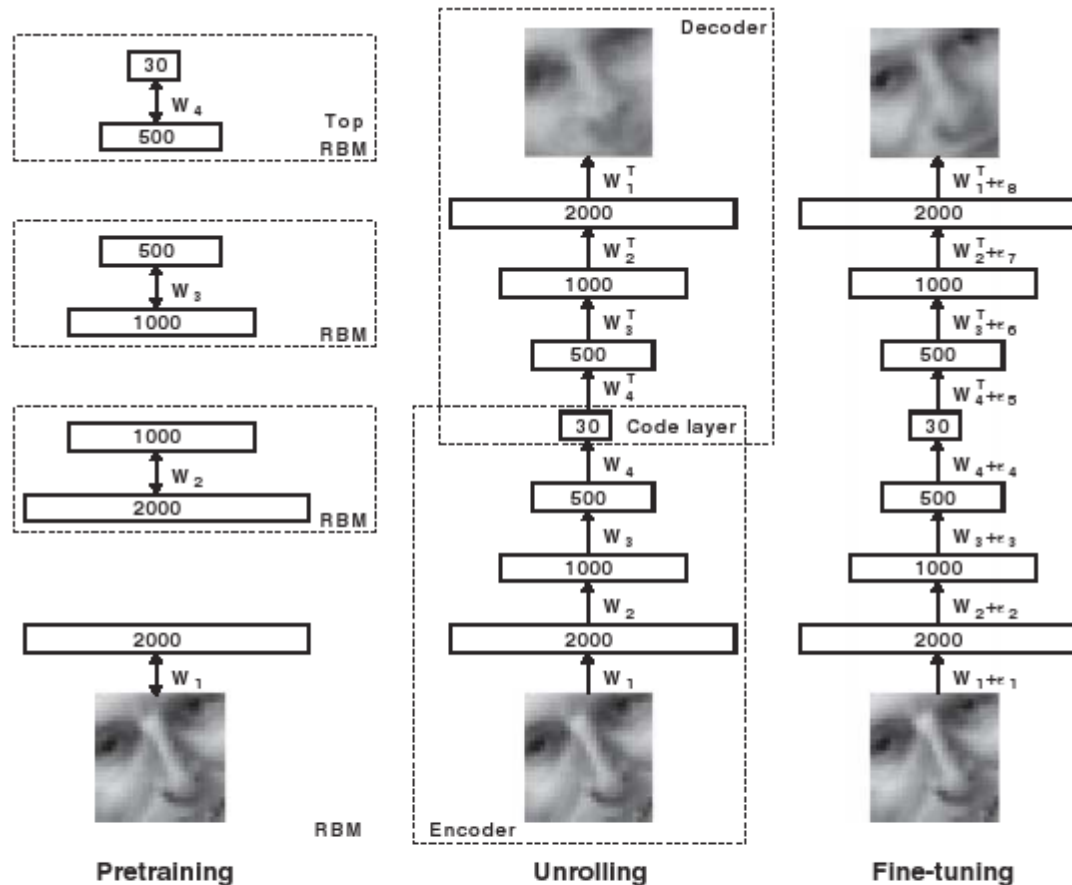
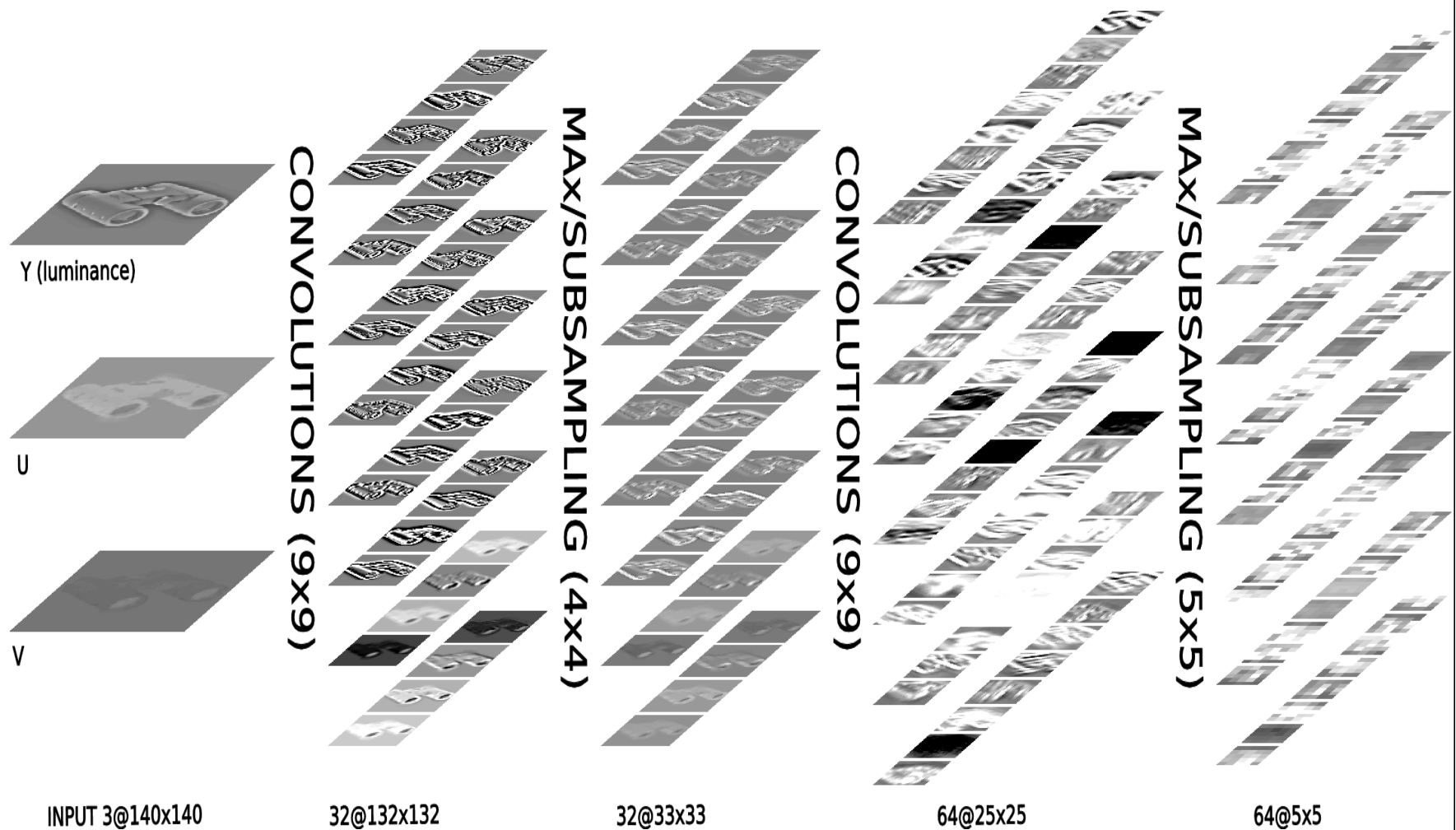


Fig. 1. Pretraining consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors. The learned feature activations of one RBM are used as the “data” for training the next RBM in the stack. After the pretraining, the RBMs are “unrolled” to create a deep autoencoder, which is then fine-tuned using backpropagation of error derivatives.

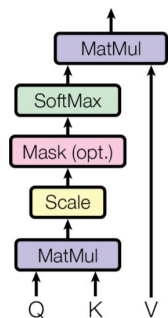
Convolutional Net



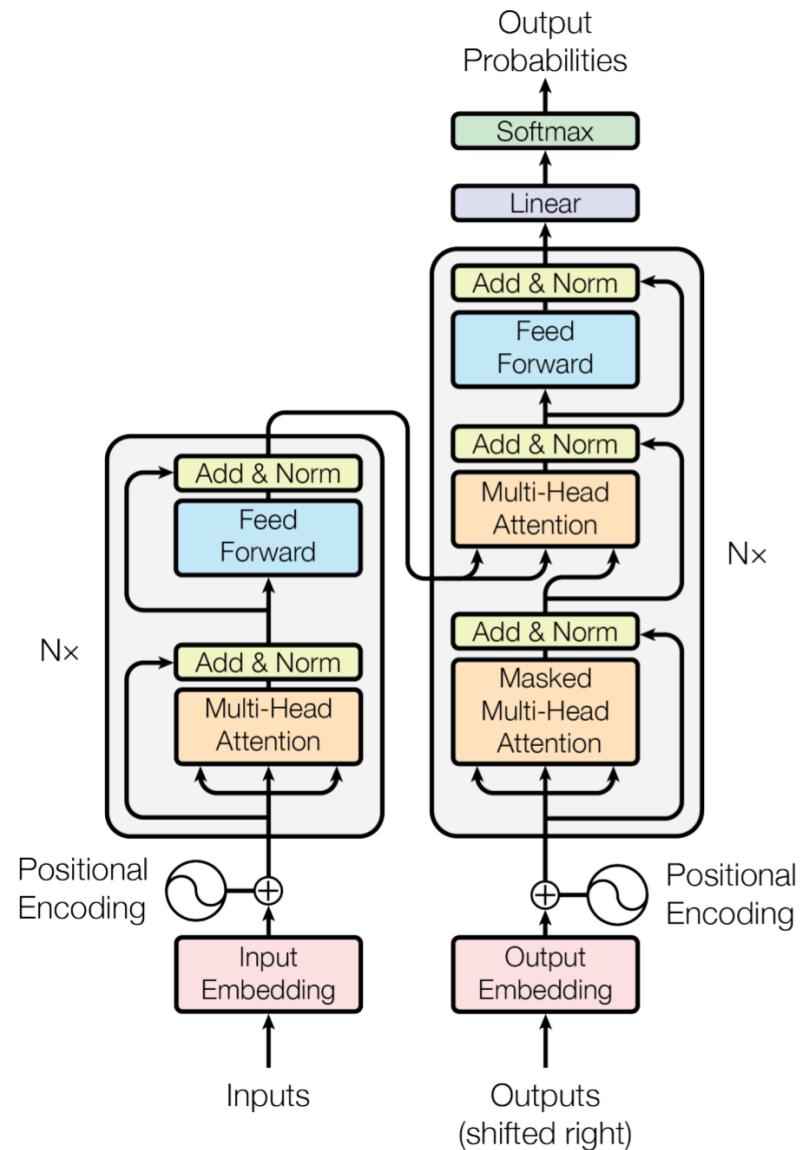
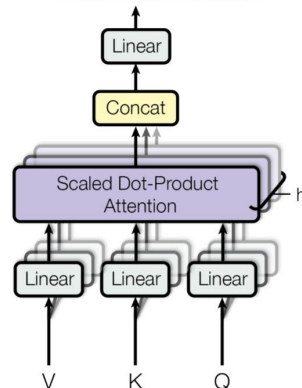
Transformer

- NLP
- Encoder-decoder
- Feed Forward Network

Scaled Dot-Product Attention



Multi-Head Attention



感知器算法

- 一、问题的表达
- 二、感知器算法
- 三、最小均方误差算法 (LMSE)

问题的表达

- 假设样本线性可分
- 已知两个类别的训练样本集合：

$$\omega_1 : \{ \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L \}$$

$$\omega_2 : \{ \mathbf{x}_{L+1}, \mathbf{x}_{L+2}, \dots, \mathbf{x}_M \}$$

∨ 求向量 \mathbf{w} ，使得 $g(\mathbf{x}) = \mathbf{w}^t \mathbf{x}$ ，能够区分 ω_1 类和 ω_2 类。

问题的表达

$$\mathbf{x}_1^t \mathbf{w} > 0, \mathbf{x}_2^t \mathbf{w} > 0, \dots, \mathbf{x}_L^t \mathbf{w} > 0$$

$$-\mathbf{x}_{L+1}^t \mathbf{w} > 0, -\mathbf{x}_{L+2}^t \mathbf{w} > 0, \dots, -\mathbf{x}_M^t \mathbf{w} > 0$$

矩阵形式描述

$$\begin{bmatrix} \mathbf{x}_1^t \\ \vdots \\ \mathbf{x}_L^t \\ -\mathbf{x}_{L+1}^t \\ \vdots \\ -\mathbf{x}_M^t \end{bmatrix} \mathbf{w} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{L1} & x_{L2} & \cdots & x_{Ln} & 1 \\ -x_{(L+1)1} & -x_{(L+1)2} & \cdots & -x_{(L+1)n} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -x_{M1} & -x_{M2} & \cdots & -x_{Mn} & -1 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_i \\ w_{i+1} \\ \vdots \\ w_0 \end{bmatrix} > \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\mathbf{X}\mathbf{w} > \mathbf{0}$$

\mathbf{X} 称为增广矩阵。

权向量的解

- 存在性：只有当**样本集线性可分**的条件下，解才存在；
- 唯一性：线性不等式组的**解是不唯一**；

一般求解方法 — 梯度下降法

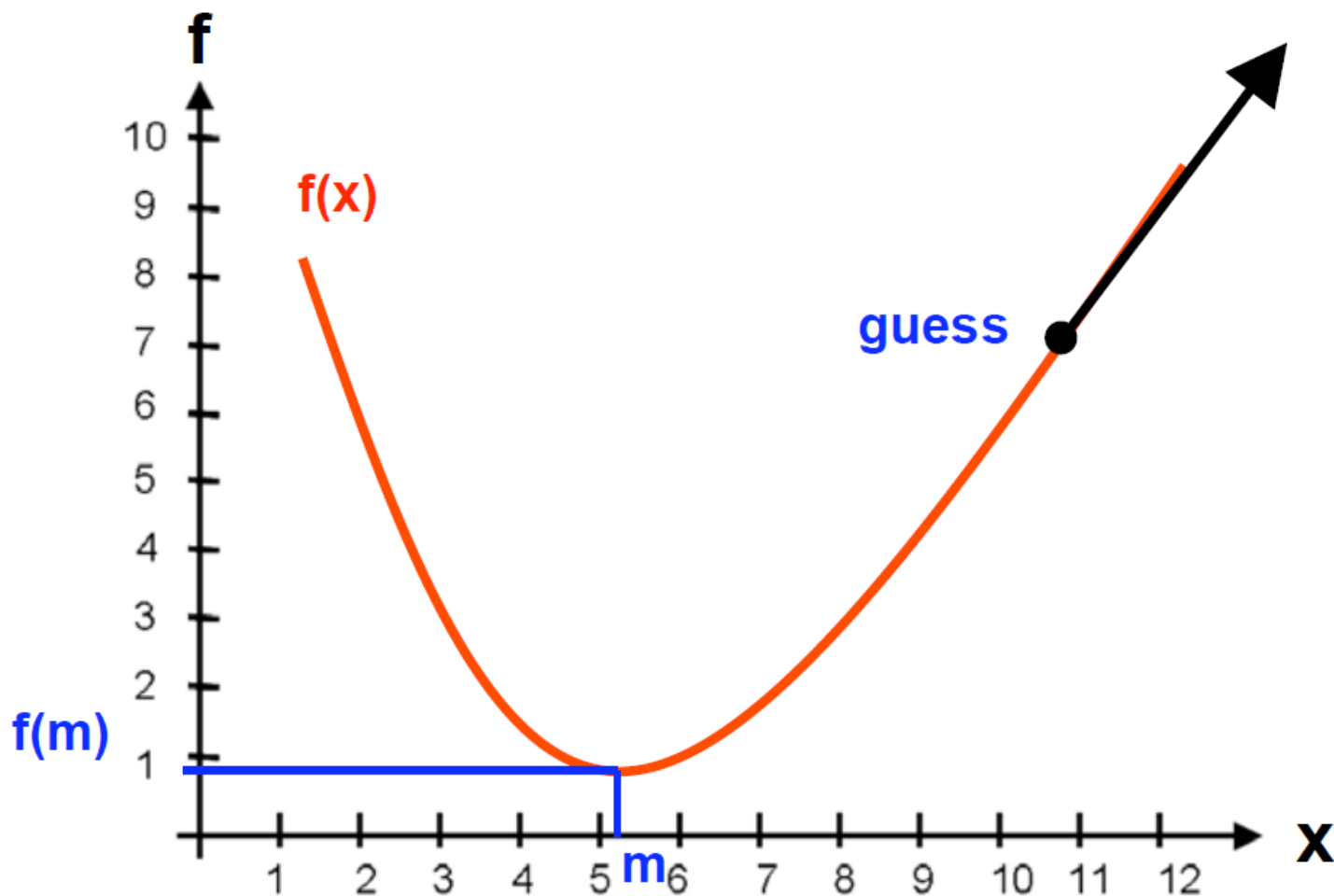
- 求解不等式组采用的最优化方法：
 1. 定义一个**准则函数** $J(\mathbf{w})$ ，当 \mathbf{w} 是解向量时， $J(\mathbf{w})$ 为最小；
 2. 采用最优化方法求解标量函数 $J(\mathbf{w})$ 的极小值。
- 最优化方法采用最多的是**梯度下降法**，设定初始权值向量 $\mathbf{w}(1)$ ，然后沿梯度的负方向迭代计算：

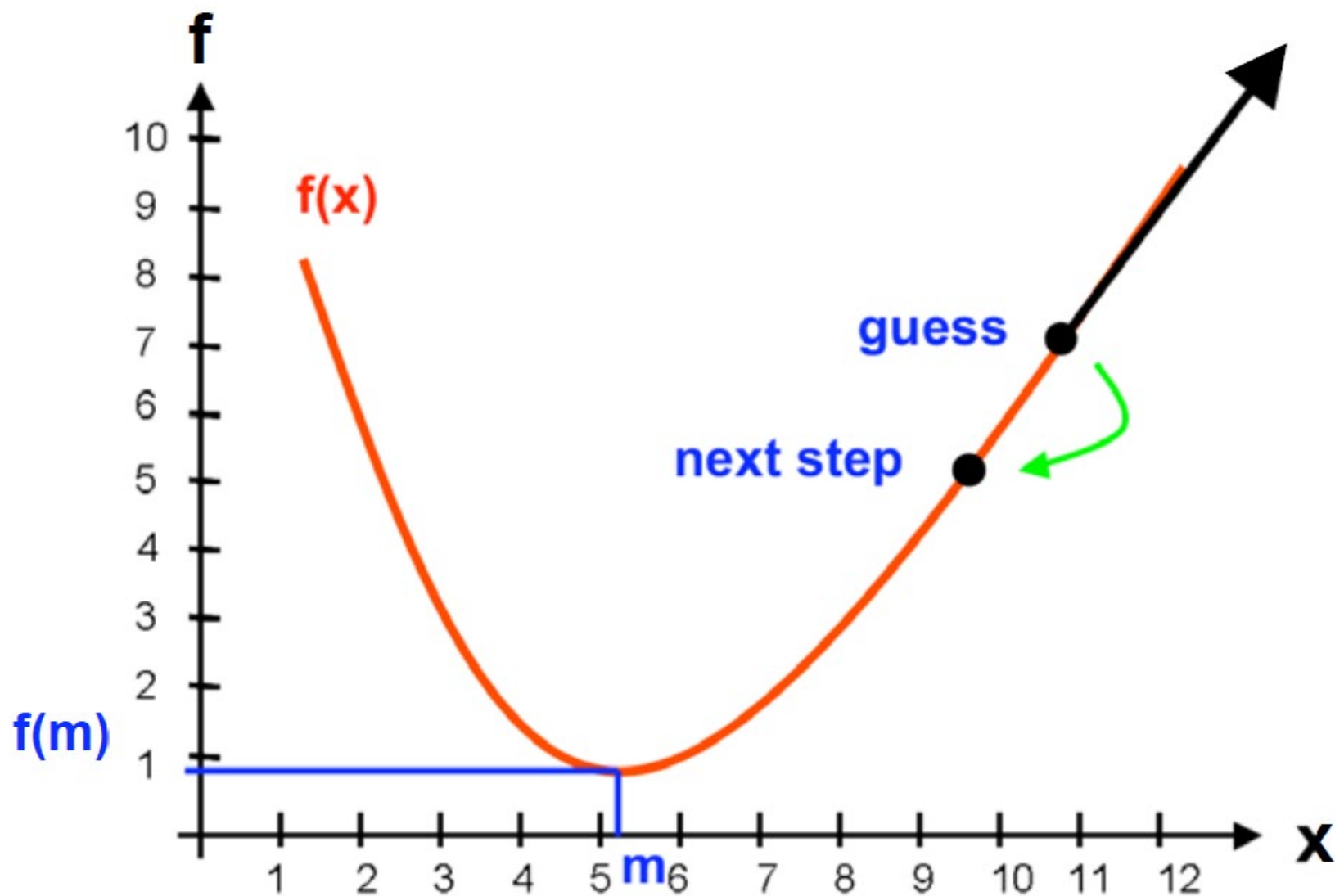
$$\mathbf{w}(k+1) = \mathbf{w}(k) - \eta(k) \nabla J(\mathbf{w}(k))$$

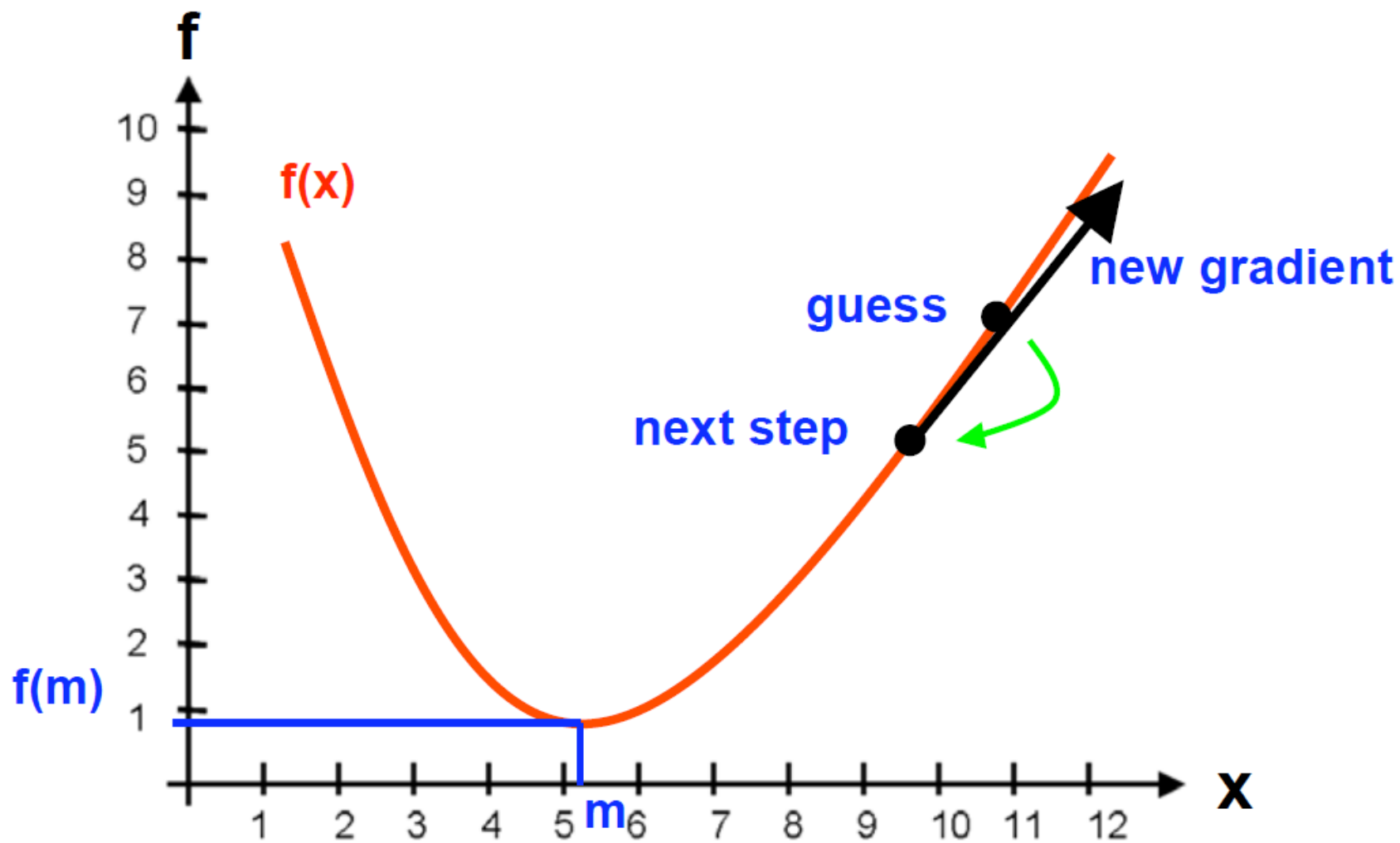
其中 $\eta(k)$ 称为**学习率**，或称步长。

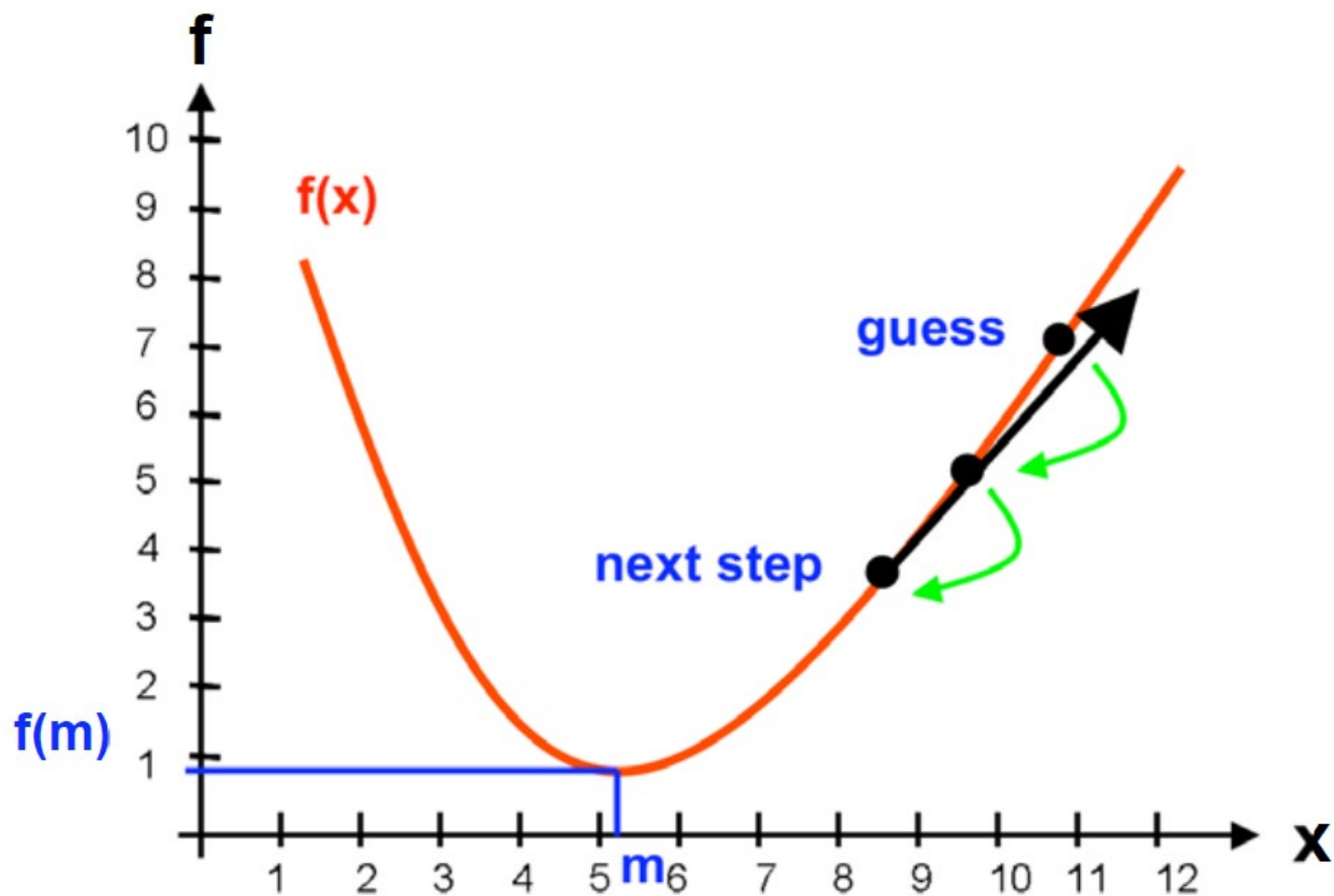
梯度下降法

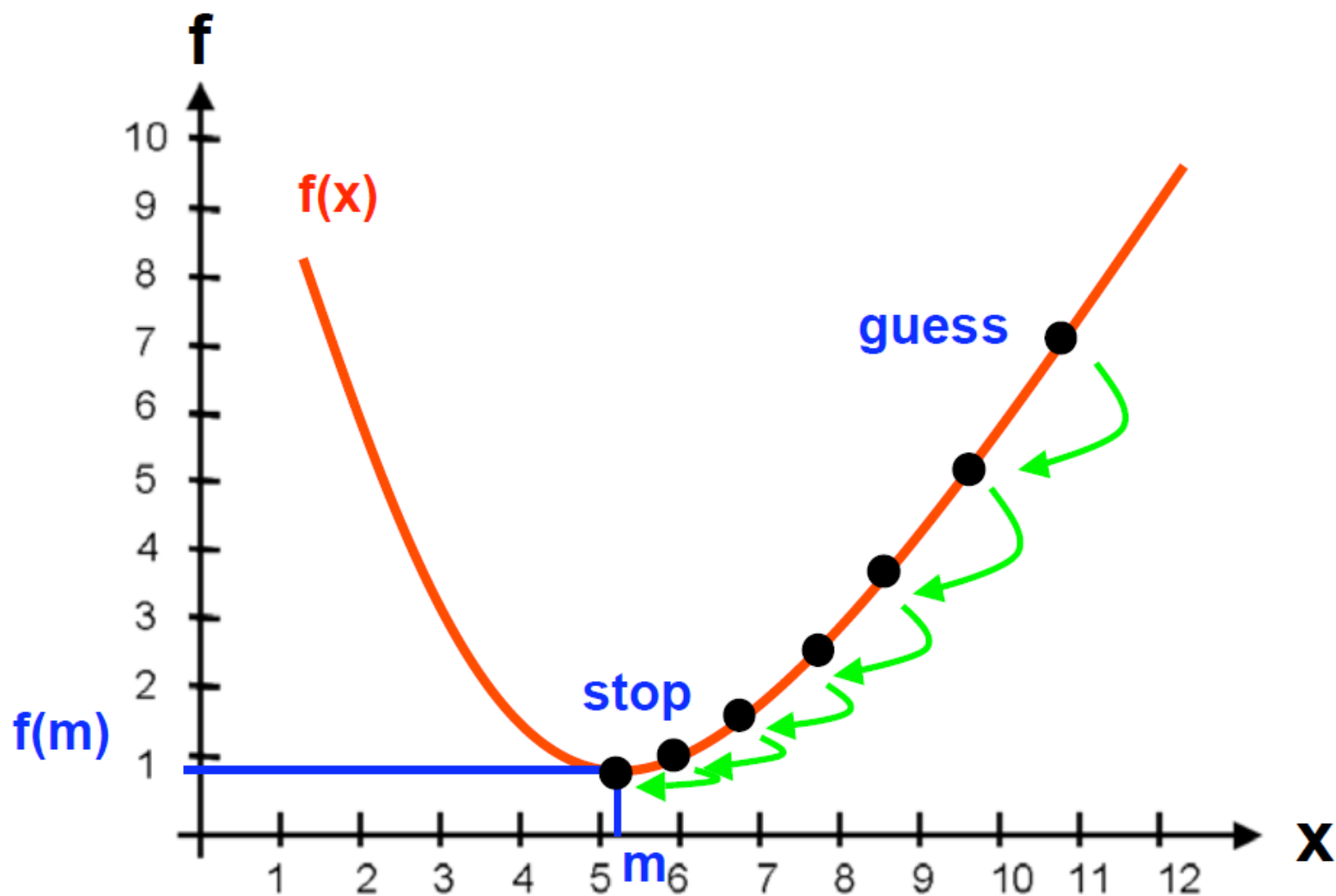
Minimum of a function is found by following the slope of the function











Start with a point (guess)

Repeat

Determine a descent direction

Choose a step

Update

Until stopping criterion is satisfied

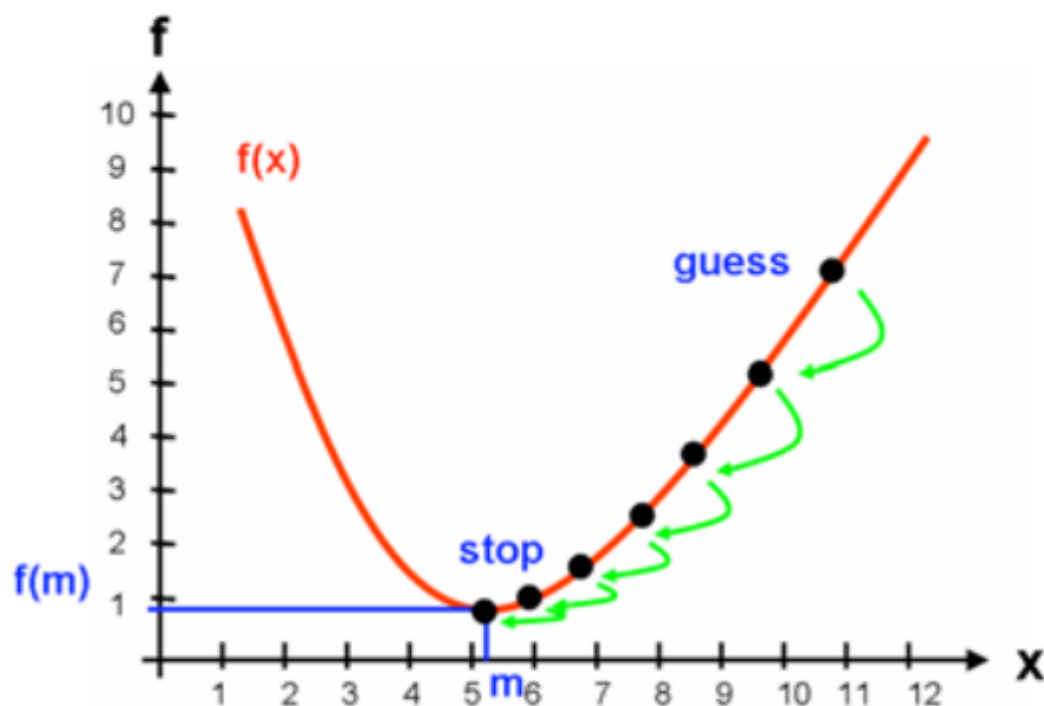
$\text{guess} = x$

$\text{direction} = -f'(x)$

$\text{step} = h > 0$

$x := x - hf'(x)$

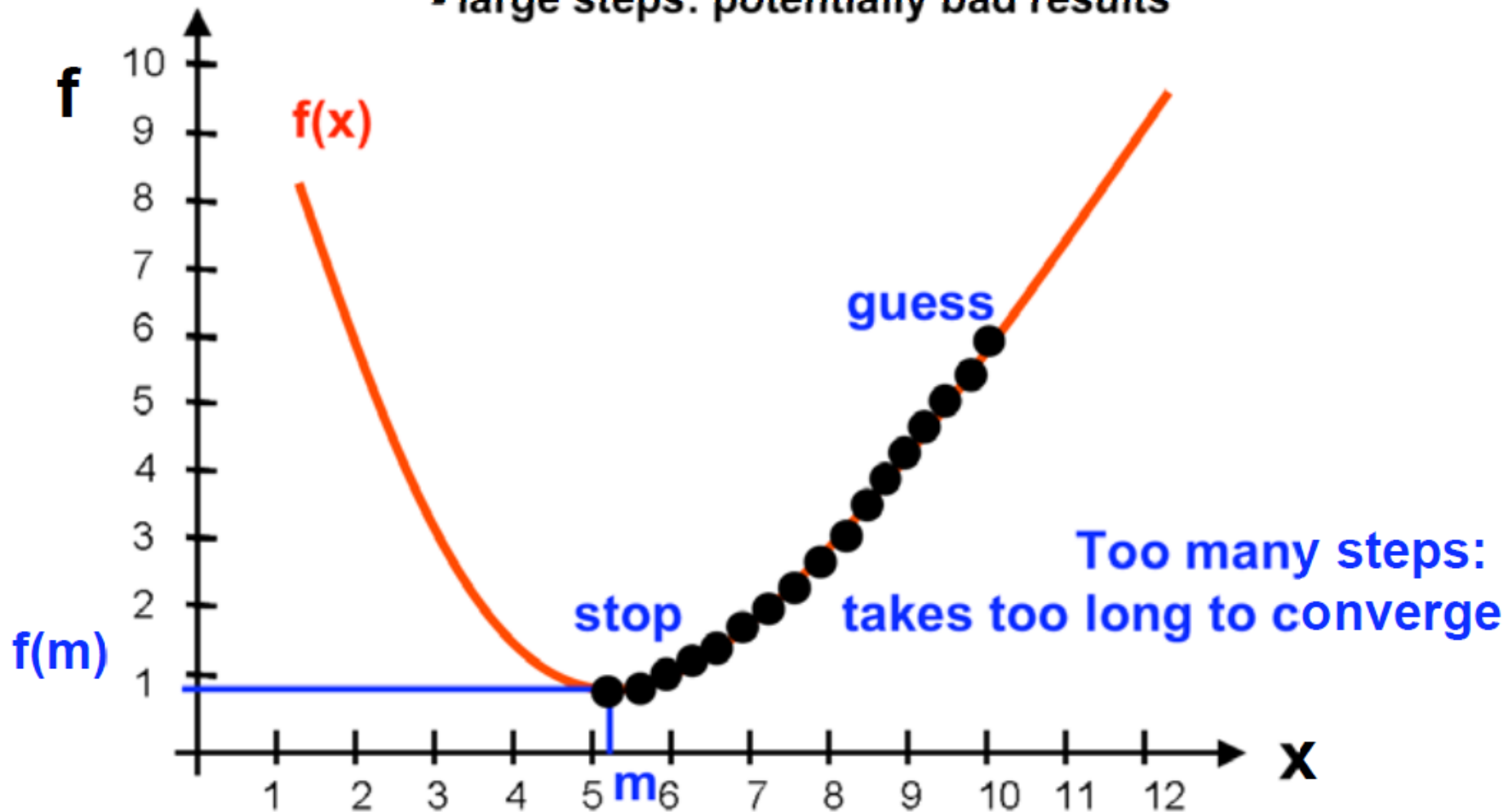
$f'(x) \sim 0$



Problem 1: choice of the step

When updating the current computation:

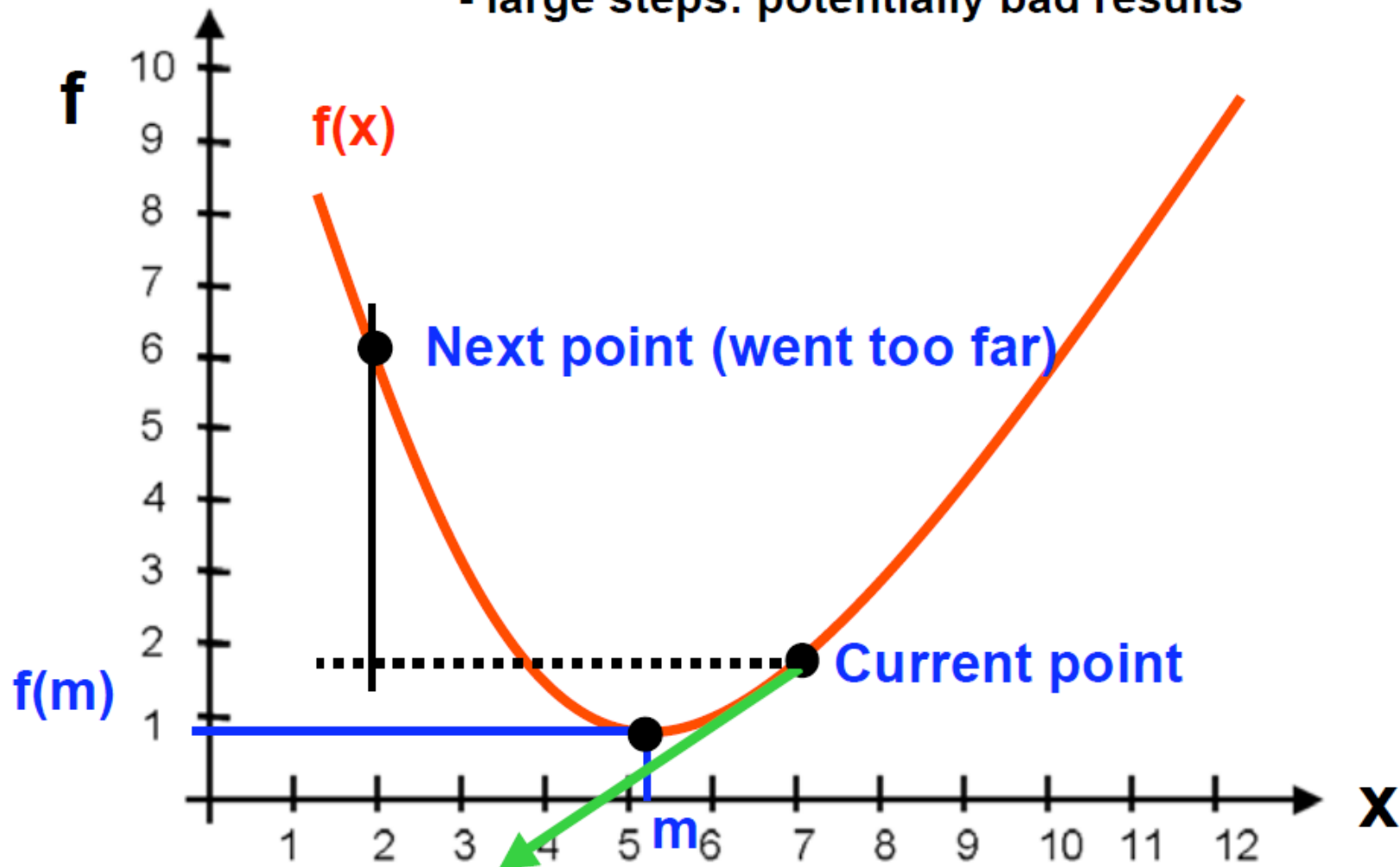
- small steps: inefficient
- large steps: potentially bad results



Problem 1: choice of the step

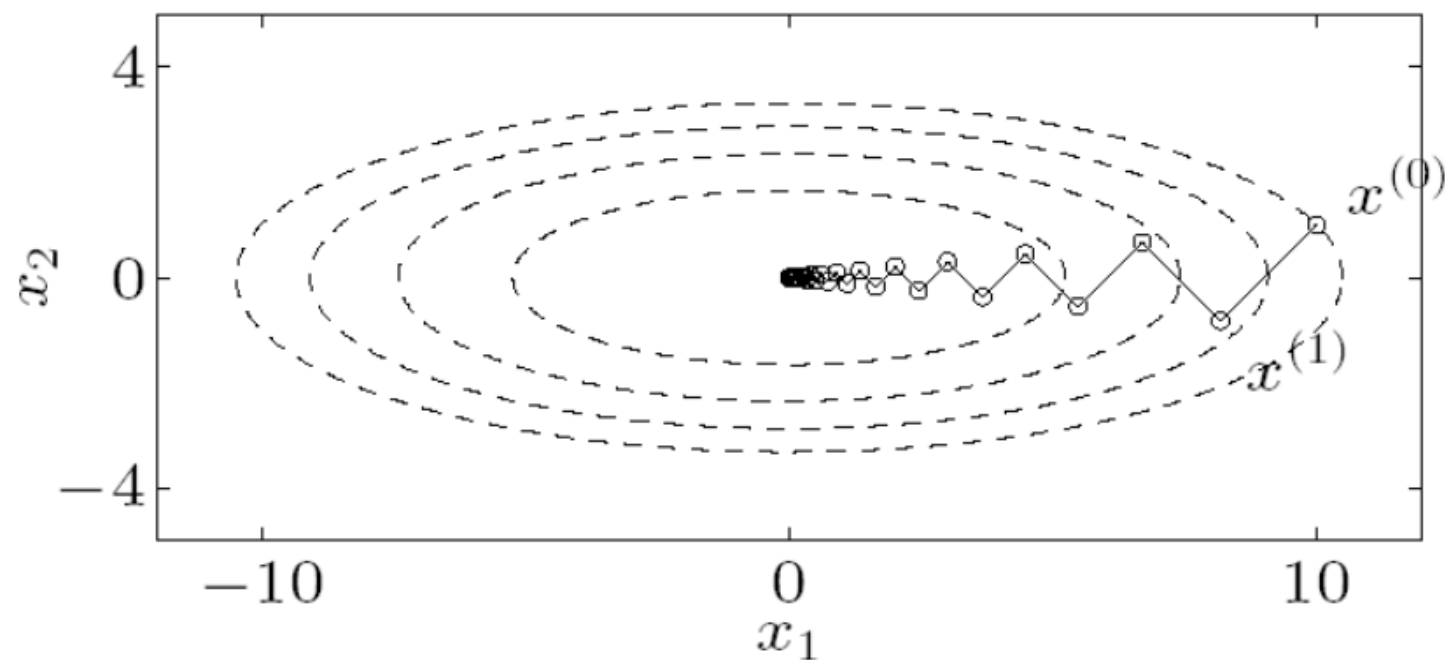
When updating the current computation:

- small steps: inefficient
- large steps: potentially bad results



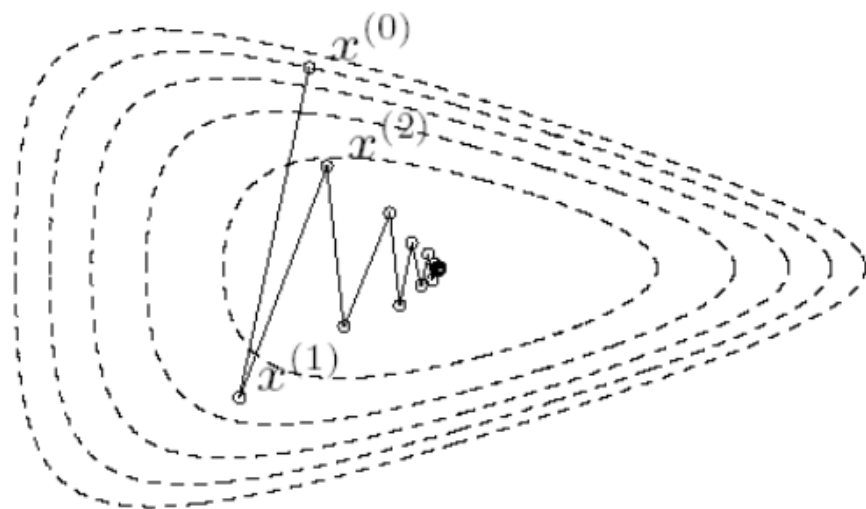
Problem 2: « ping pong effect »

$$f(x) = (1/2)(x_1^2 + \gamma x_2^2) \quad (\gamma > 0)$$

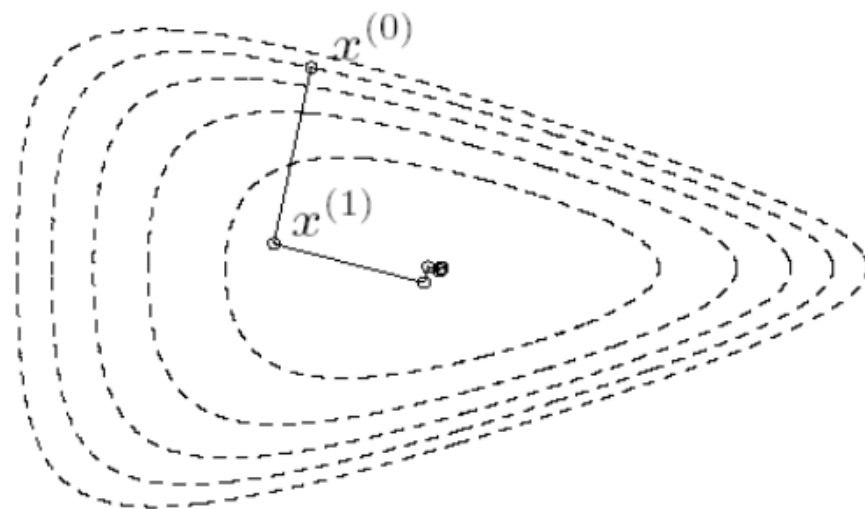


Problem 2: « ping pong effect »

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$$



backtracking line search



exact line search

Problem 3: stopping criterion

Intuitive criterion:

$$|f'(x)| \leq \varepsilon$$

In multiple dimensions:

$$\|\nabla f\| \leq \varepsilon$$

Or equivalently

$$\|\nabla f\| = \sqrt{\sum_{i=1}^N \left(\frac{\partial f}{\partial x_i}\right)^2} = \sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2 + \left(\frac{\partial f}{\partial x_2}\right)^2 + \cdots + \left(\frac{\partial f}{\partial x_N}\right)^2} \leq \varepsilon$$

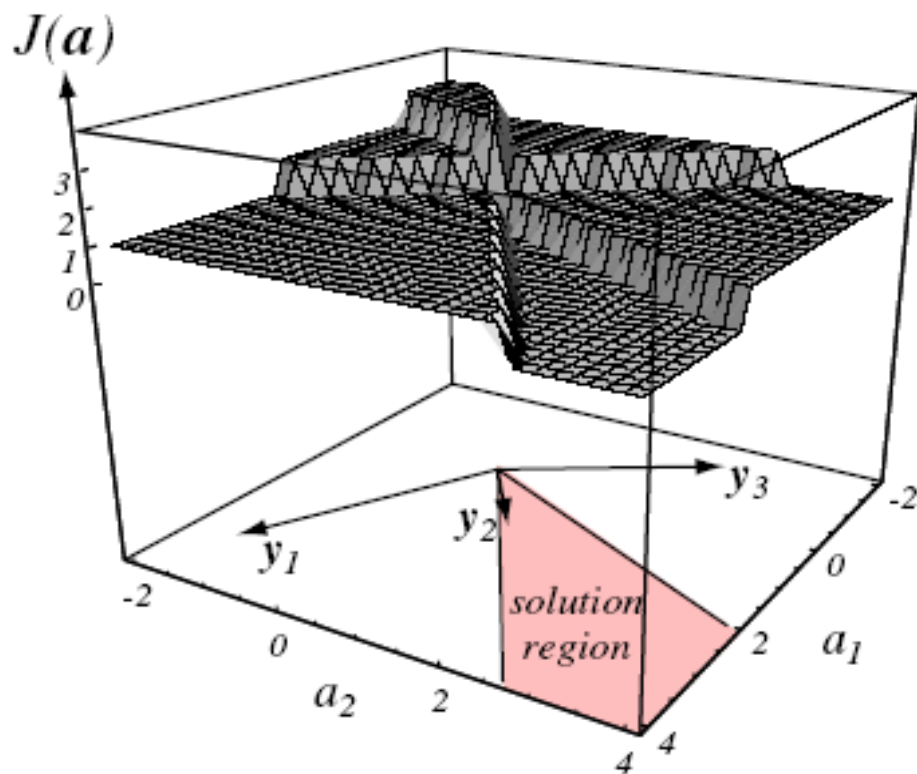
Rarely used in practice.

More about this in EE227A (convex optimization, Prof. L. El Ghaoui).

感知器算法(Perceptron)

- 最直观的准则函数定义是**最少错分样本数准则**:

$J_N(\mathbf{w})$ = 样本集合中被错误分类的样本数;



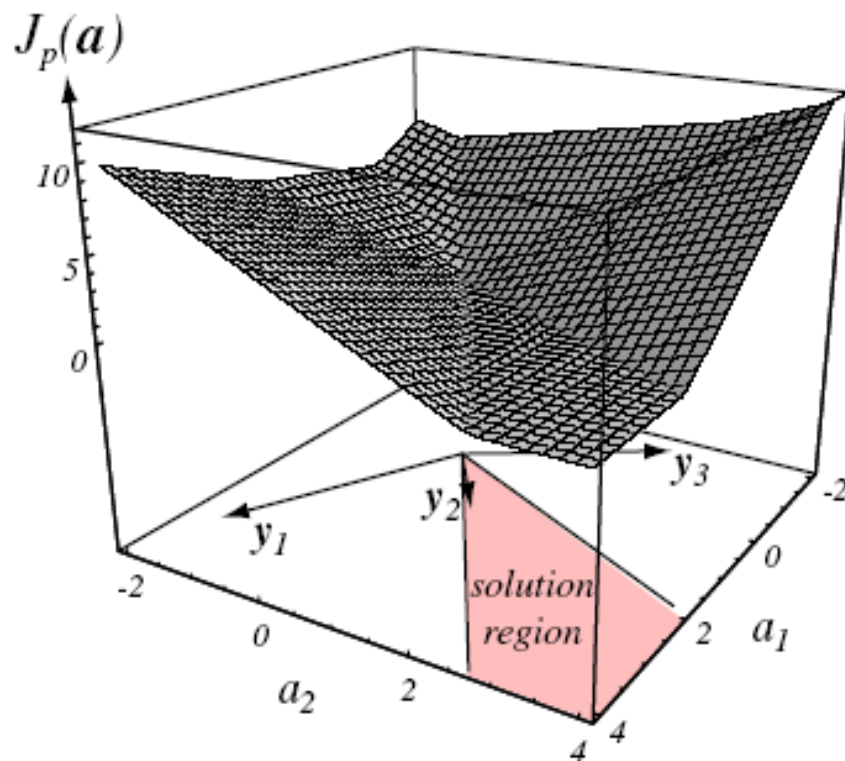
$J_N(\mathbf{w})$ 不可导!

感知器准则

- 以错分样本到判别界面距离之和作为准则：

$$J_P(\mathbf{w}) = \sum_{\mathbf{x} \in X} (-\mathbf{w}^t \mathbf{x})$$

$$\nabla J_P = \sum_{\mathbf{x} \in X} (-\mathbf{x})$$



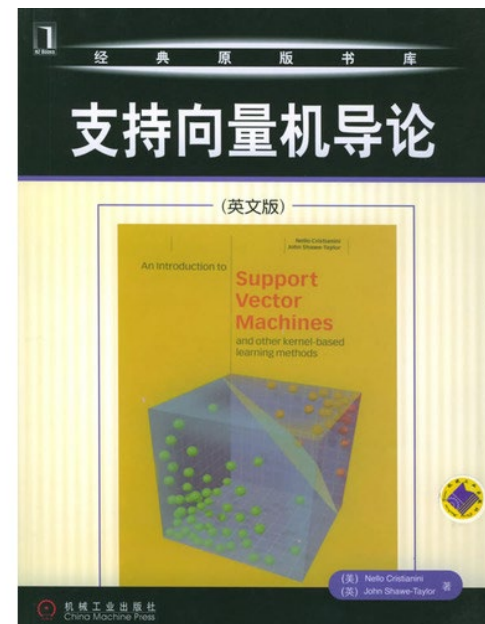
感知器算法(单样本调整版本)

1. **begin initialize** $\mathbf{w}(0)$, $k = 0$
2. **do** $k = (k+1) \bmod n$
3. **if** \mathbf{x}_k is misclassified by $\mathbf{w}(k)$ **then**
 $\mathbf{w}(k+1) \leftarrow \mathbf{w}(k) + \mathbf{x}_k$
4. **until** all patterns properly classified
5. **return** \mathbf{w}
6. **end**

Stochastic Optimization

收敛性定理 (Rosenblatt, 1962)

- Let the subsets of training vectors X_1 and X_2 be linearly separable. Let the inputs presented to the perceptron originate from these subsets. The perceptron converges after some n_0 iterations.
- 自己考虑可否证明上述定理？
- 支持向量机导论 (英文版)
 - 克里斯蒂亚尼尼 (Cristianini, N.) 等著
 - 机械工业出版社
 - 2005-7-1



课外作业：证明上述定理

例1

- 现有两个类别的训练样本：

$$\omega_1 = \{(0,0)^t, (0,1)^t\}, \omega_2 = \{(1,0)^t, (1,1)^t\}$$

用(单样本)感知器算法对线性分类器进行学习。

感知器算法(批量调整版本)

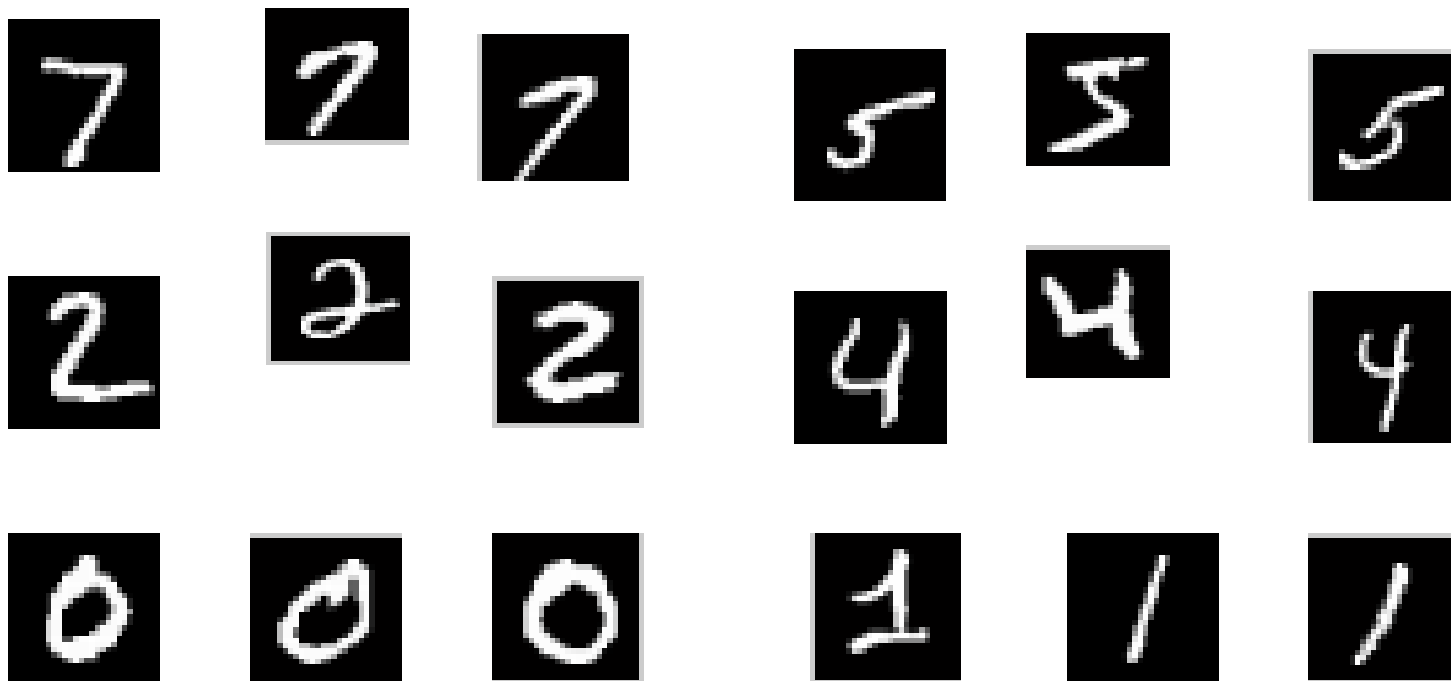
1. **begin initialize** $\mathbf{w}(0), \eta(0), \theta, k = 0$
2. **do** $k \leftarrow k+1$
3. $\mathbf{w}(k+1) \leftarrow \mathbf{w}(k) + \eta(k) \sum_{\mathbf{x} \in X_k} \mathbf{x}$
4. **until** $\left| \eta(k) \sum_{\mathbf{x} \in X_k} \mathbf{x} \right| < \theta$
5. **return** \mathbf{w}
6. **End**

- $\eta(k)$ 的取法: $\eta(0)=1, \eta(k)=1/k$

例2

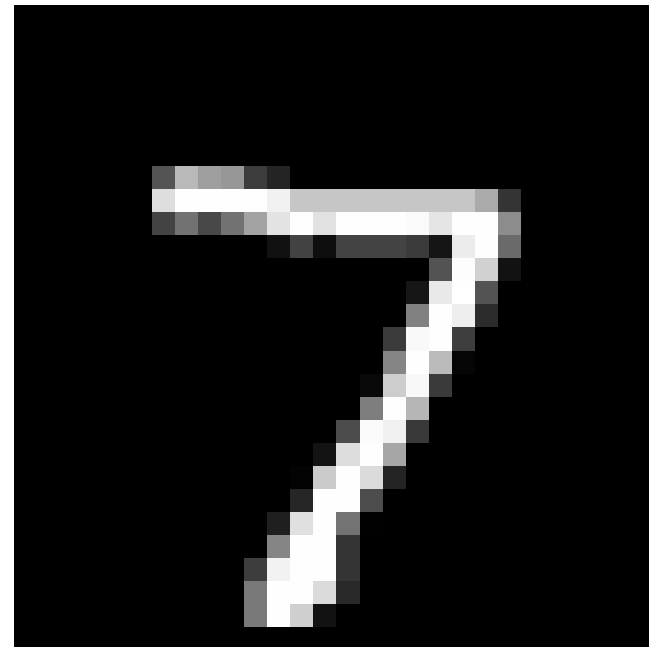
- 使用手写数字样本集(MNIST)训练线性分类器，多类问题的解决方案采用方案2：一对一方式，训练45个线性分类器；
- 用测试样本测试识别效果。

MNIST手写数字数据集



MNIST数据集

- 图像大小：28*28;
- 灰度图像：每点取值范围0~255;
- 样本数量：
训练样本：60,000
测试样本：10,000
- 目的：正确识别手写体数字0 ~ 9
- <http://yann.lecun.com/exdb/mnist/>



LMSE算法的思想

$$\begin{bmatrix} \mathbf{x}_1^t \\ \vdots \\ \mathbf{x}_L^t \\ -\mathbf{x}_{L+1}^T \\ \vdots \\ -\mathbf{x}_M^t \end{bmatrix} \mathbf{w} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{L1} & x_{L2} & \cdots & x_{Ln} & 1 \\ -x_{(L+1)1} & -x_{(L+1)2} & \cdots & -x_{(L+1)n} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -x_{M1} & -x_{M2} & \cdots & -x_{Mn} & -1 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_i \\ w_{i+1} \\ \vdots \\ w_0 \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_i \\ b_{i+1} \\ \vdots \\ b_M \end{bmatrix}$$

$$\mathbf{b} = (b_1, b_2, \dots, b_M)^t, \quad b_i > 0$$

$$\mathbf{X}\mathbf{w} = \mathbf{b}$$

问题求解

- 已知：增广矩阵 \mathbf{X} (可由训练样本集得到)和 \mathbf{b} (设定);
- 求： \mathbf{w} 。
- \mathbf{X} 一般不是方阵，所以问题实际上无解，只能求近似解。

优化的准则函数

- 定义误差向量 \mathbf{e} :

$$\mathbf{e} = \mathbf{X}\mathbf{w} - \mathbf{b}$$

- √ 定义准则函数 $J(\mathbf{w})$:

$$J(\mathbf{w}) = \frac{1}{2} \|\mathbf{e}\|^2 = \frac{1}{2} \|\mathbf{X}\mathbf{w} - \mathbf{b}\|^2 = \frac{1}{2} (\mathbf{X}\mathbf{w} - \mathbf{b})^t (\mathbf{X}\mathbf{w} - \mathbf{b})$$

梯度法求解

$$\frac{\partial J}{\partial \mathbf{w}} = \mathbf{X}^t (\mathbf{X}\mathbf{w} - \mathbf{b}) = \mathbf{0}$$

√ 定义伪逆矩阵 \mathbf{X}^* :

$$\begin{aligned}\mathbf{X}^+ &= (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \\ \mathbf{w} &= \mathbf{X}^+ \mathbf{b} = (\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t \mathbf{b}\end{aligned}$$

感知器网络的训练

- 训练样本 $\mathbf{x}_1, \dots, \mathbf{x}_n$ ，期望输出 $\mathbf{b}_1, \dots, \mathbf{b}_n$ ，

准则函数：
$$J(\mathbf{W}) = \frac{1}{2} \sum_{i=1}^n \|\mathbf{W}^t \mathbf{x}_i - \mathbf{b}_i\|^2$$

梯度：
$$\frac{\partial J}{\partial \mathbf{W}} = \sum_{i=1}^n (\mathbf{W}^t \mathbf{x}_i - \mathbf{b}_i) \mathbf{x}_i^t$$

梯度下降：
$$\begin{aligned} \mathbf{W}(k+1) &= \mathbf{W}(k) - \eta \frac{\partial J}{\partial \mathbf{W}} \\ &= \mathbf{W}(k) - \eta \sum_{i=1}^n (\mathbf{W}^t \mathbf{x}_i - \mathbf{b}_i) \mathbf{x}_i^t \end{aligned}$$

例3

- 现有两个类别的训练样本：

$$\omega_1 = \left\{ (0,0)^t, (0,1)^t \right\}, \omega_2 = \left\{ (1,0)^t, (1,1)^t \right\}$$

用LMSE算法训练线性分类器。

例4

- 使用手写数字样本集训练线性分类器，多类问题的解决方案采用方案1：一对多方式，训练10个线性分类器；
- 用测试样本测试识别效果；
- 使用经过PCA降维之后的样本；

多类别线性判别函数的学习

- 情况一：C类问题转化为C个两类问题： ω_i 样本作为一类，其它样本作为另一类进行训练；
- 情况二：C类问题问题转化为 $C(C-1)/2$ 个两类问题， ω_i 样本作为一类， ω_j 样本作为另一类，训练 w_{ij} ；

多类问题情况三

- 采用扩展的感知器算法
 1. 初始化C个权向量 $\mathbf{w}_i(1)$ ，选择常数 η ，置步数 $k=1$;
 2. 输入增广特征向量 \mathbf{x}_k ，计算C个判别函数的输出:

$$g_i(\mathbf{x}_k) = \mathbf{w}_i^t(k) \mathbf{x}_k$$

扩展的感知器算法

3. 修改权向量，规则为：

若 \mathbf{x}_k 属于 ω_i ，并且 $g_i(\mathbf{x}_k) > g_j(\mathbf{x}_k)$ ，对任意的 $j \neq i$ ，则：

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k), \quad i = 1, \dots, C$$

若 \mathbf{x}_k 属于 ω_i ，而 $g_i(\mathbf{x}_k) < g_l(\mathbf{x}_k)$ ，则：

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \eta \mathbf{x}_k;$$

$$\mathbf{w}_l(k+1) = \mathbf{w}_l(k) - \eta \mathbf{x}_k$$

$$\mathbf{w}_j(k+1) = \mathbf{w}_j(k), \quad j \neq i, l$$

扩展的感知器算法

4. 重复2, 3步, 当 $k=M$ 时, 检测 C 个判别函数是否能够对全部训练样本正确分类, 如正确分类, 则结束, 否则 $k=1$, 转2, 继续。

感知器算法小结

- 线性可分情形：一定能够学习到一个线性分类器将两类分开。
- 线性不可分情形：能够学习到一个具有一定分类能力的线性分类器并在有限时间内停止训练。
- 思考：修改扩展的感知器算法，应用于批量和不可分情形下的训练问题。

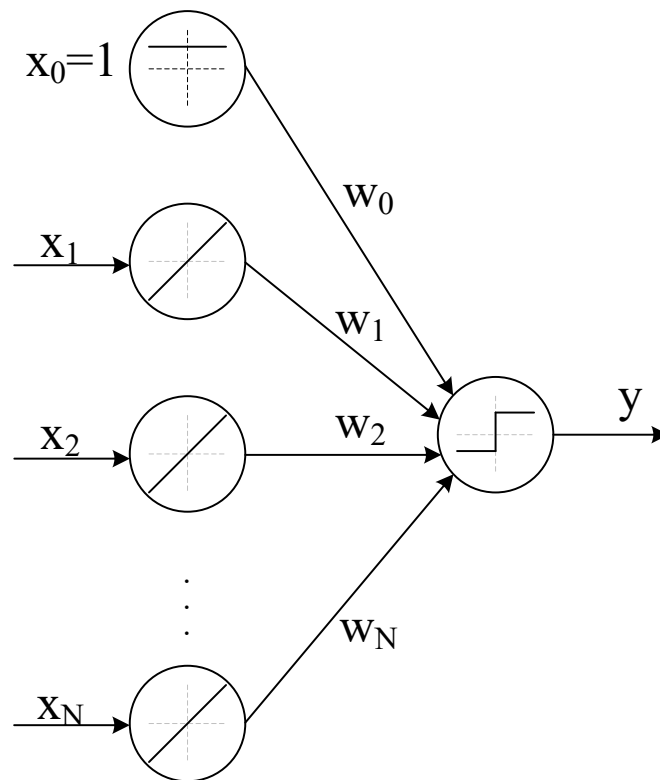
例5

- 现有三个类别的样本：

$$\omega_1 : \{(0, 0)^t\}, \omega_2 : \{(1, 1)^t\}, \omega_3 : \{(-1, 1)^t\}$$

用扩展的感知器算法训练线性分类器。

两类问题的感知器网络

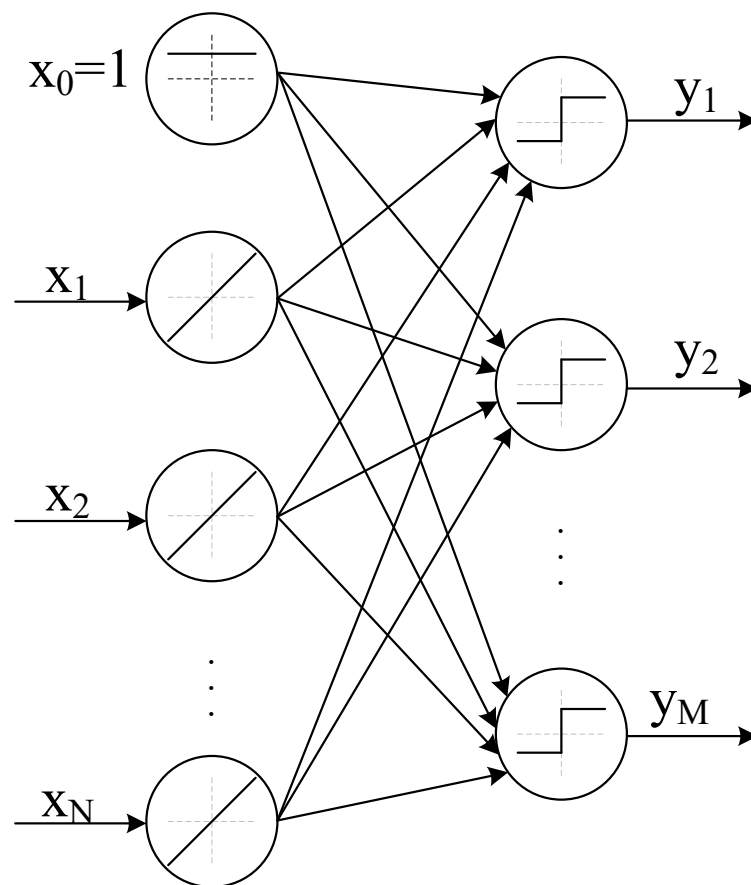


输入层的映射函数为线性函数，输出层为符号函数

$$\text{sgn}(\text{net}) = 1, \text{net} > 0$$

$$-1, \text{net} < 0$$

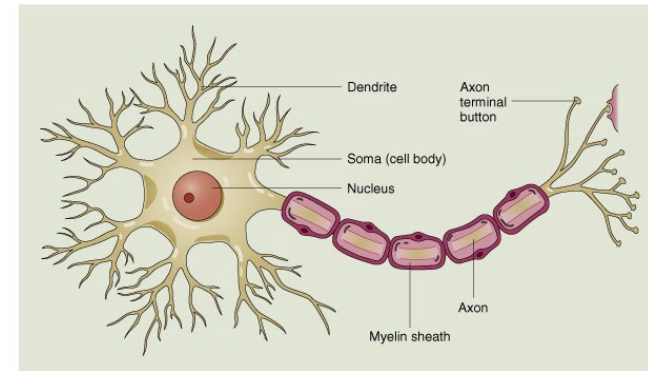
多类问题的感知器网络



输出层也可以采用编码输出的方式。输出层的映射函数可以为线性函数，阈值函数或S函数，但效果均为线性映射。

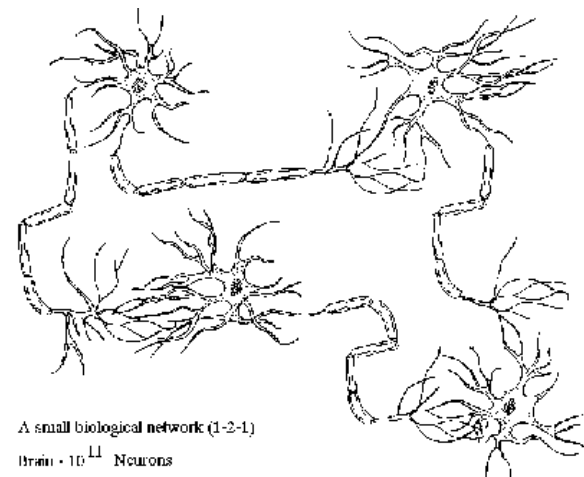
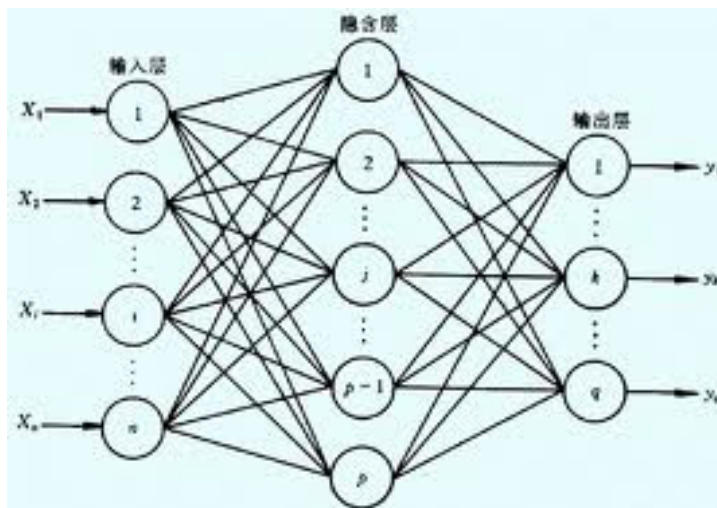
神经网络、感知器、人工神经网络

- 神经元、突触 (Hubel & Wiesel, 1959, 1962, Nobel Prize 1981)



© 2000 John Wiley & Sons, Inc.

- 神经网络



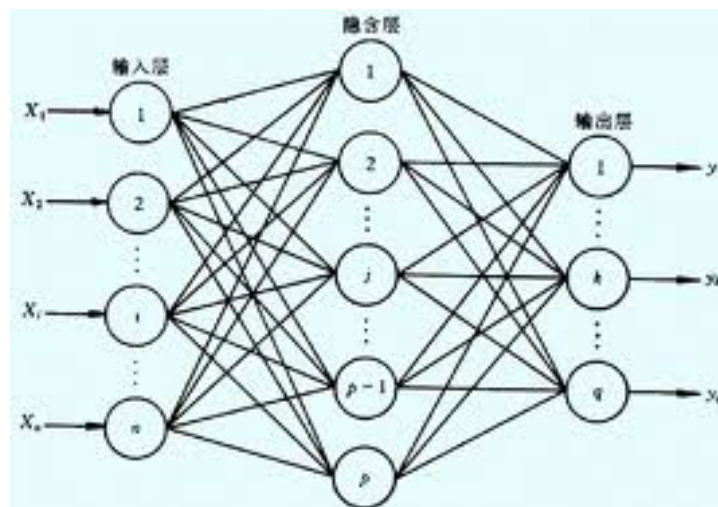
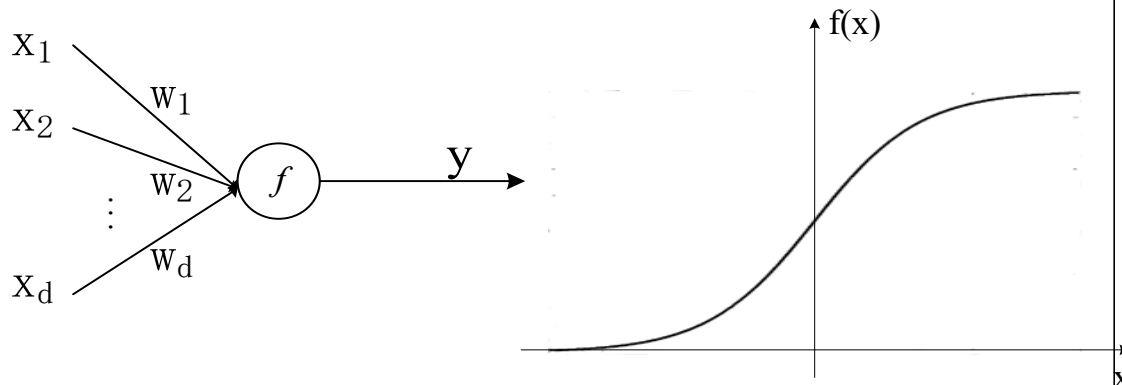
A small biological network (1-2-1)

Brain - 10^{11} Neurons

Nervous system - 10^{14} Synapses

人工神经网络

- 出发点
 - 模拟人类智能
- 建模方法
 - 神经元-突触
 - 网络结构
 - 模型参数?
- 学习方法
 - 结构学习
 - 参数学习

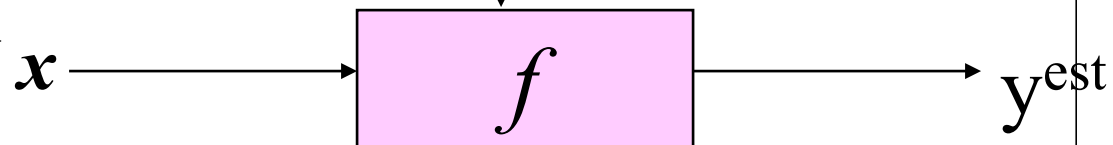


约·冯·诺依曼, 《计算机与人脑》, 2011, 商务出版社 (1958, 1965)

感知器算法的不足

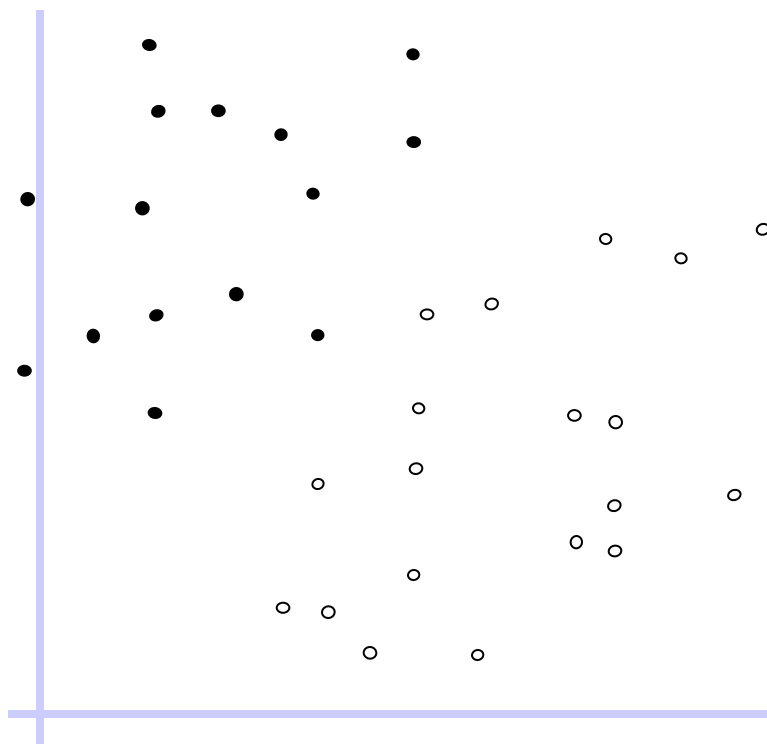
- 在样本线性可分的情况下, 感知器能够找到一个线性分类器将两类分开。
- 实际上当样本可分时, 会有无穷多种线性分类器
 - (P1): 哪一个才是最优线性分类器
 - (P2): 如何学习?
 - (P3): 如何推广到线性不可分情形?

P1: 最优线性分类器



$$f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$

- +1类
- -1类



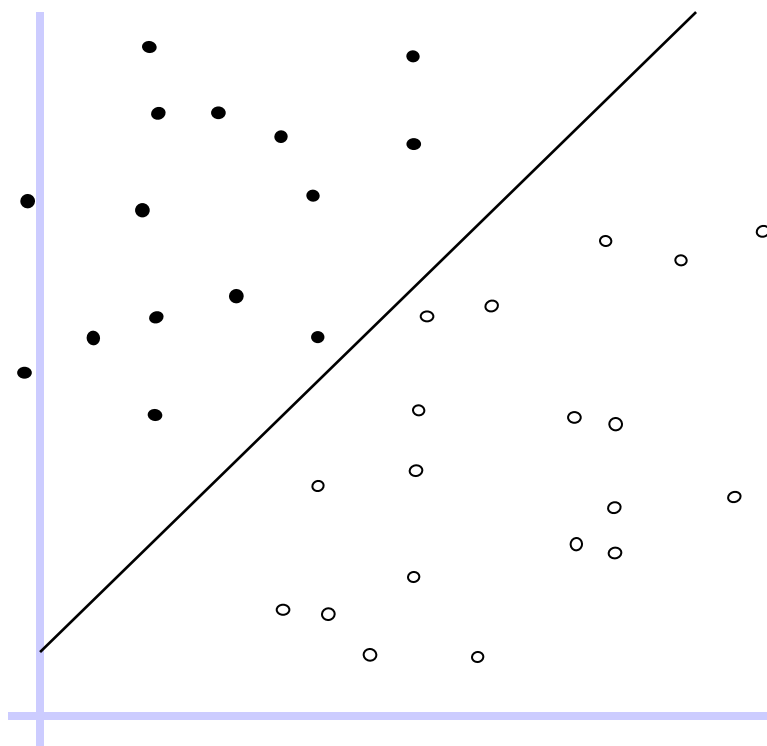
如何确定决策界面？

P1: 最优线性分类器



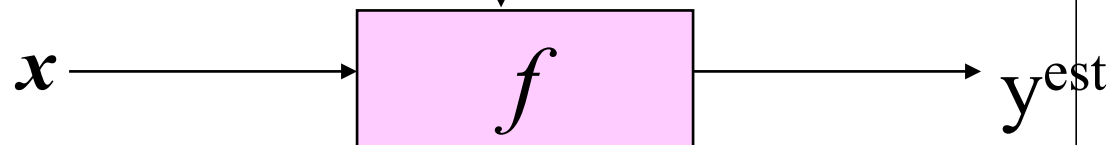
- +1类
- -1类

$$f(x, w, b) = \text{sign}(w \cdot x - b)$$



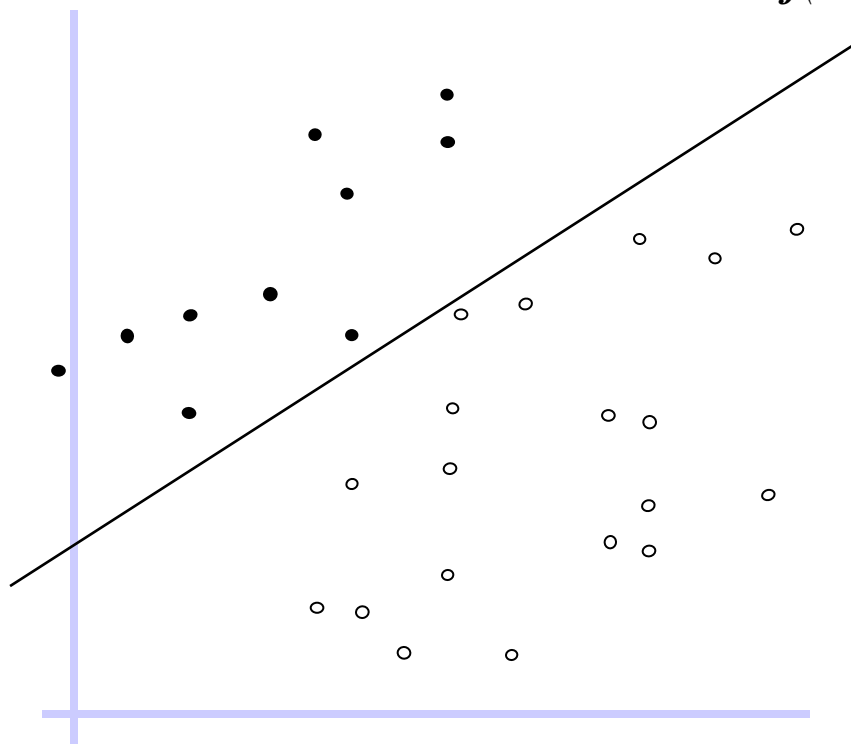
如何确定决策界面?

P1: 最优线性分类器



- +1类
- -1类

$$f(\mathbf{x}, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$



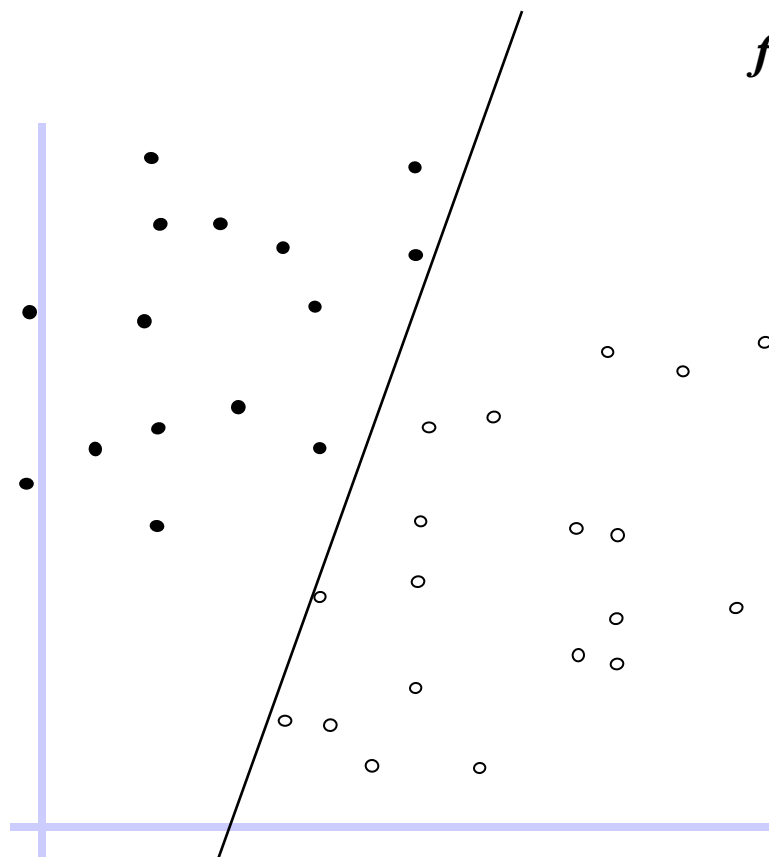
如何确定决策界面？

P1: 最优线性分类器



$$f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$

- +1类
- -1类

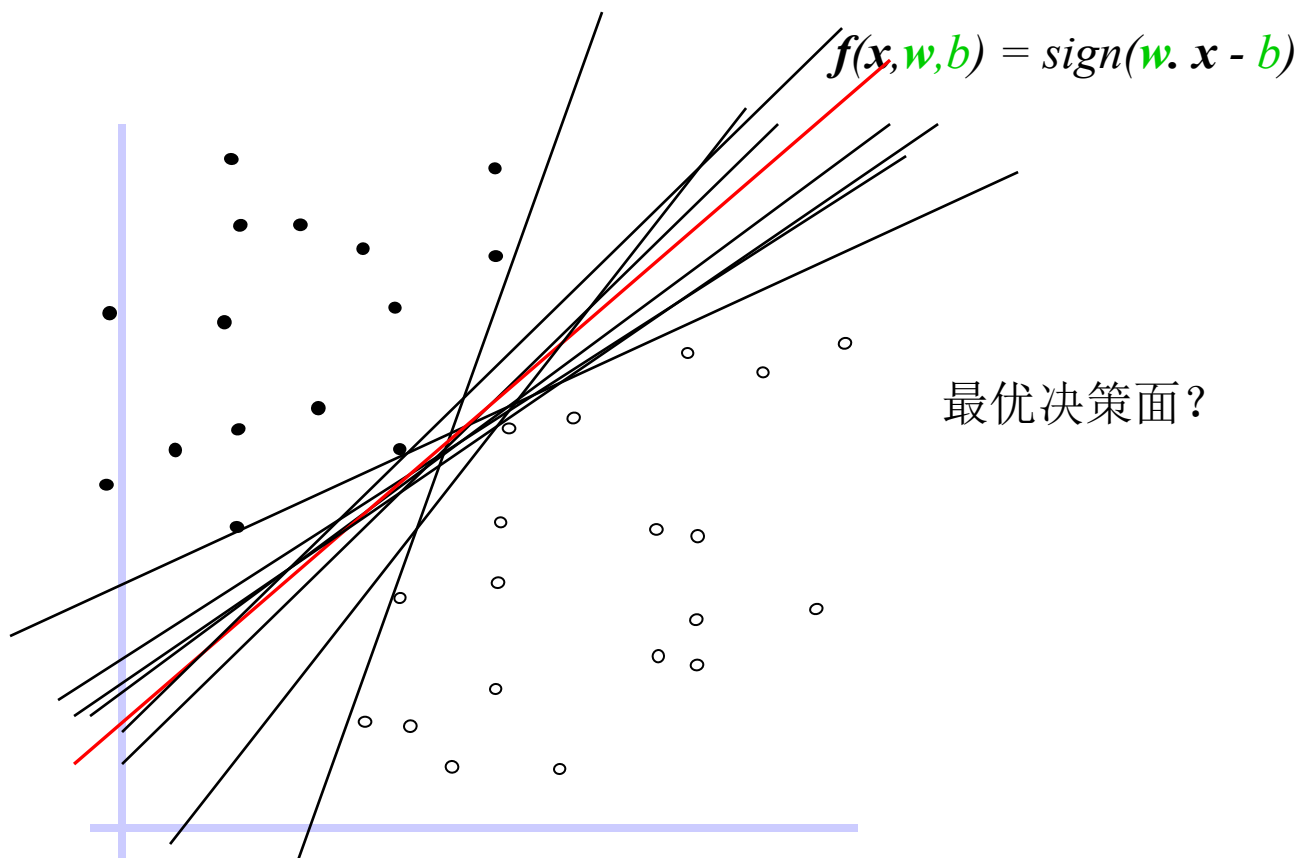


如何确定决策界面?

P1: 最优线性分类器



- +1类
- -1类

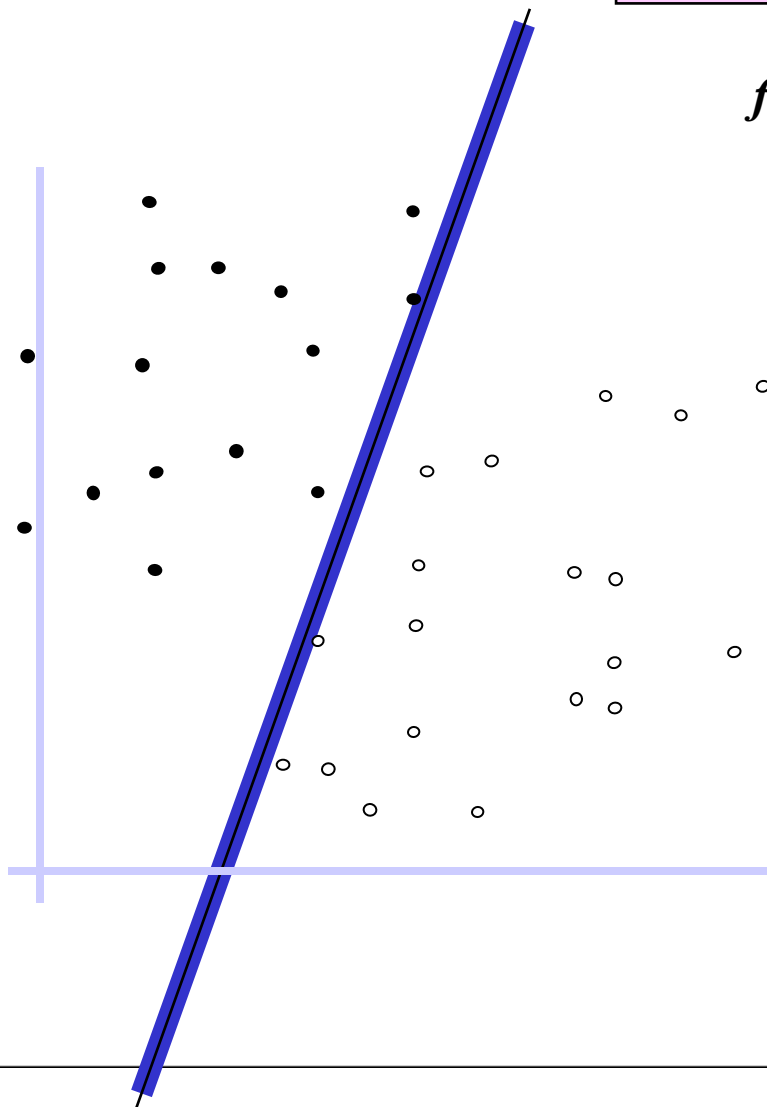


分类器间隔



$$f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$

- +1类
- -1类



定义线性分类器的
间隔为数据点到分
界面的距离。

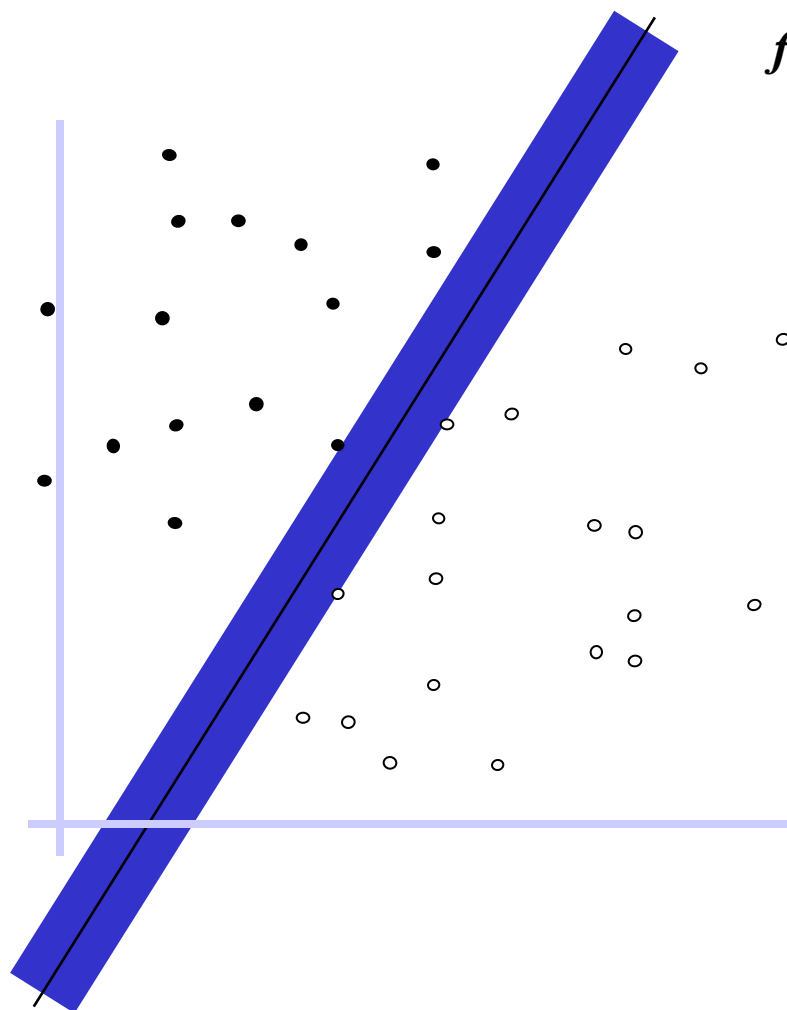
最大间隔



- +1类
- -1类

$$f(x, w, b) = \text{sign}(w \cdot x - b)$$

最大间隔线性分类器，即线性支持向量机 (LSVM)



LSVM

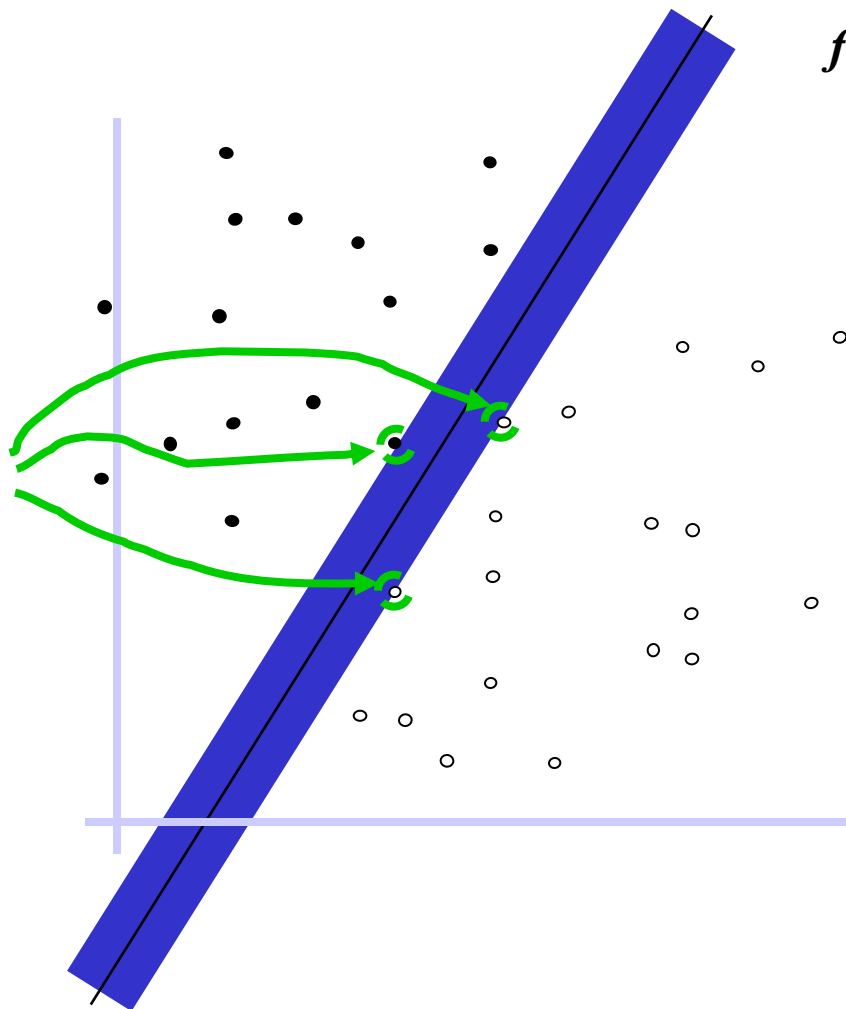
最大间隔



$$f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$

- +1类
- -1类

支持向量



最大间隔线性分类器，即线性支持向量机 (LSVM)

LSVM

最大间隔



$$f(x, w, b) = \text{sign}(w \cdot x - b)$$

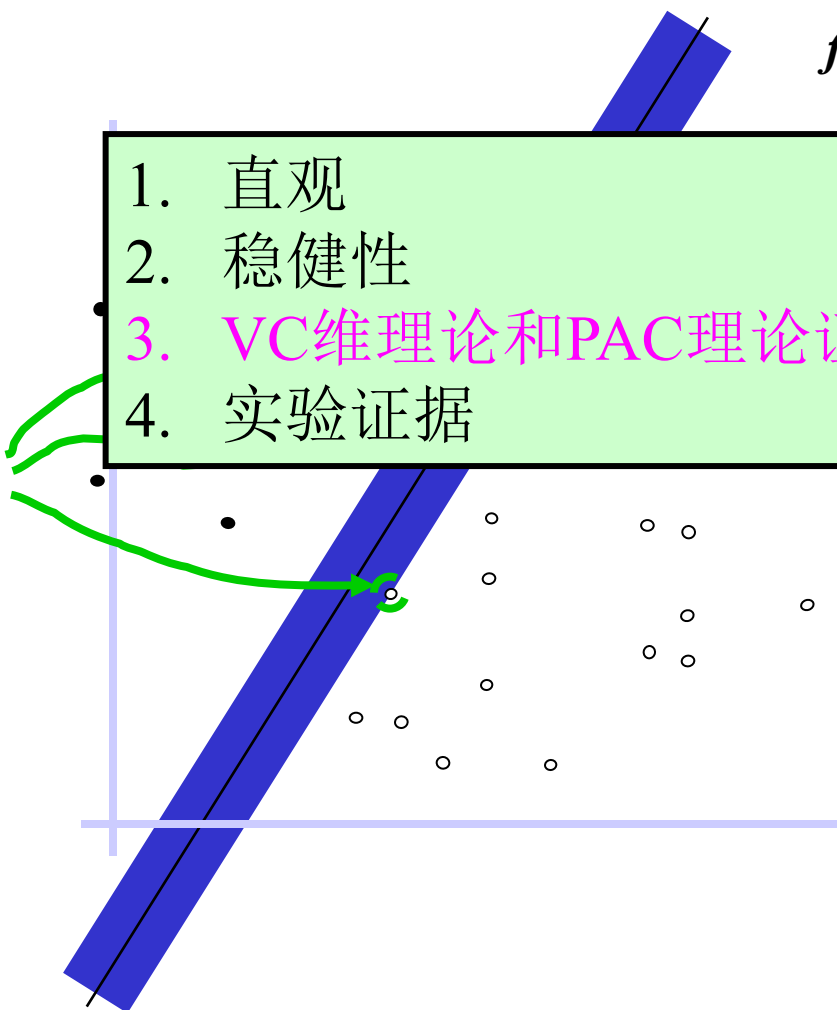
- +1类
- -1类

1. 直观
2. 稳健性
3. VC维理论和PAC理论证据
4. 实验证据

间隔线性分类
线性支持向量
(VM)

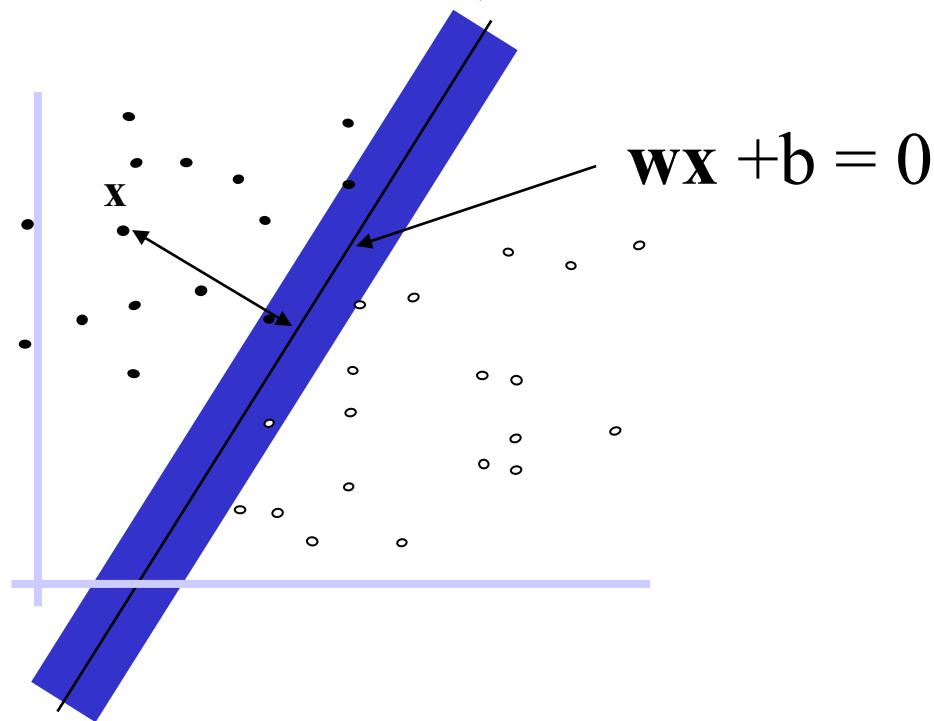
LSVM

支持向量



间隔的估计

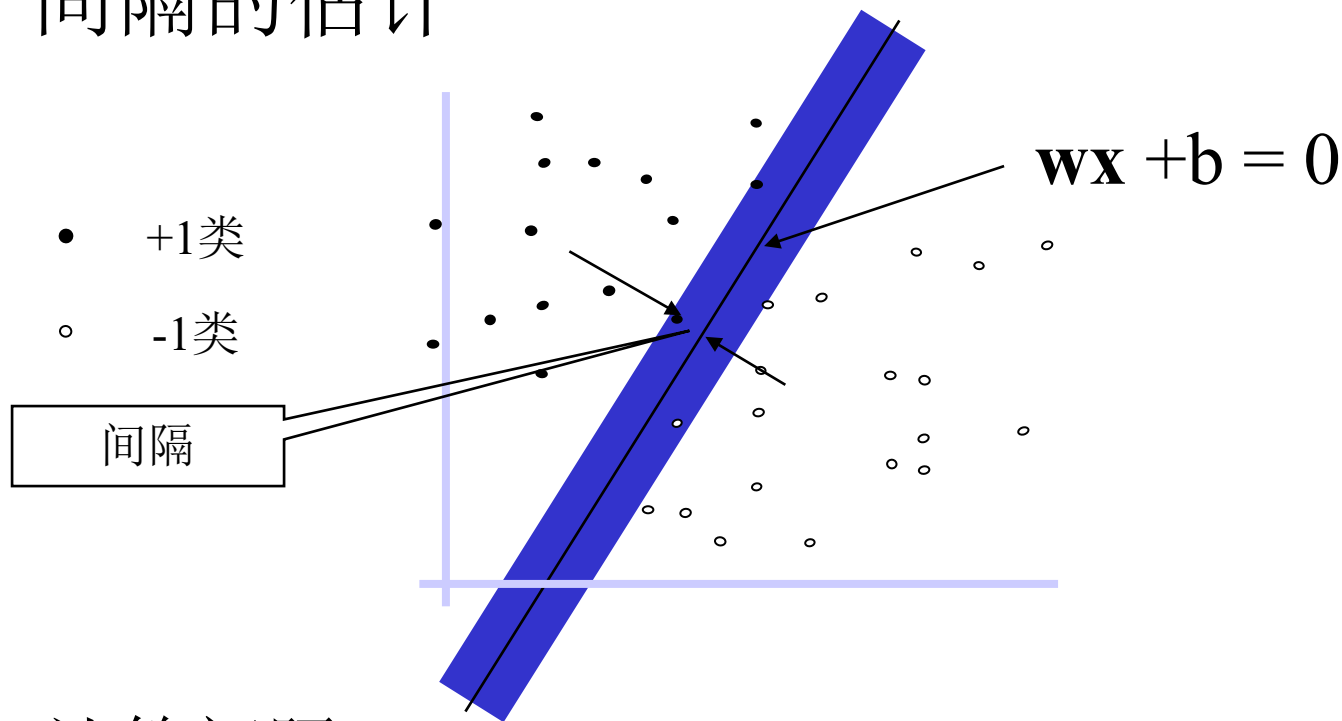
- +1类
- -1类



- 点 \mathbf{x} 到直线 $\mathbf{w}\mathbf{x} + b = 0$ 的距离?

$$d(\mathbf{x}) = \frac{|\mathbf{x} \cdot \mathbf{w} + b|}{\sqrt{\|\mathbf{w}\|_2^2}} = \frac{|\mathbf{x} \cdot \mathbf{w} + b|}{\sqrt{\sum_{i=1}^d w_i^2}}$$

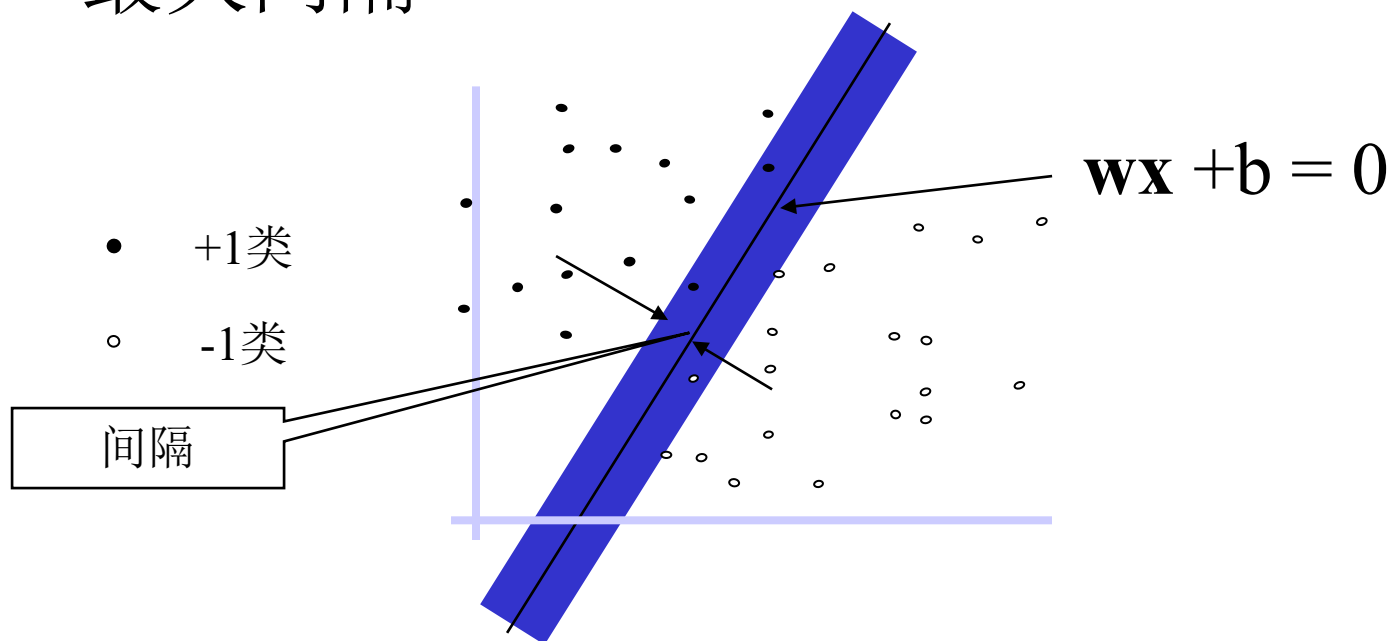
间隔的估计



- 计算间隔?

$$\text{margin} \equiv \min_{\mathbf{x} \in D} d(\mathbf{x}) = \min_{\mathbf{x} \in D} \frac{y(\mathbf{x} \cdot \mathbf{w} + b)}{\sqrt{\sum_{i=1}^d w_i^2}}$$

最大间隔

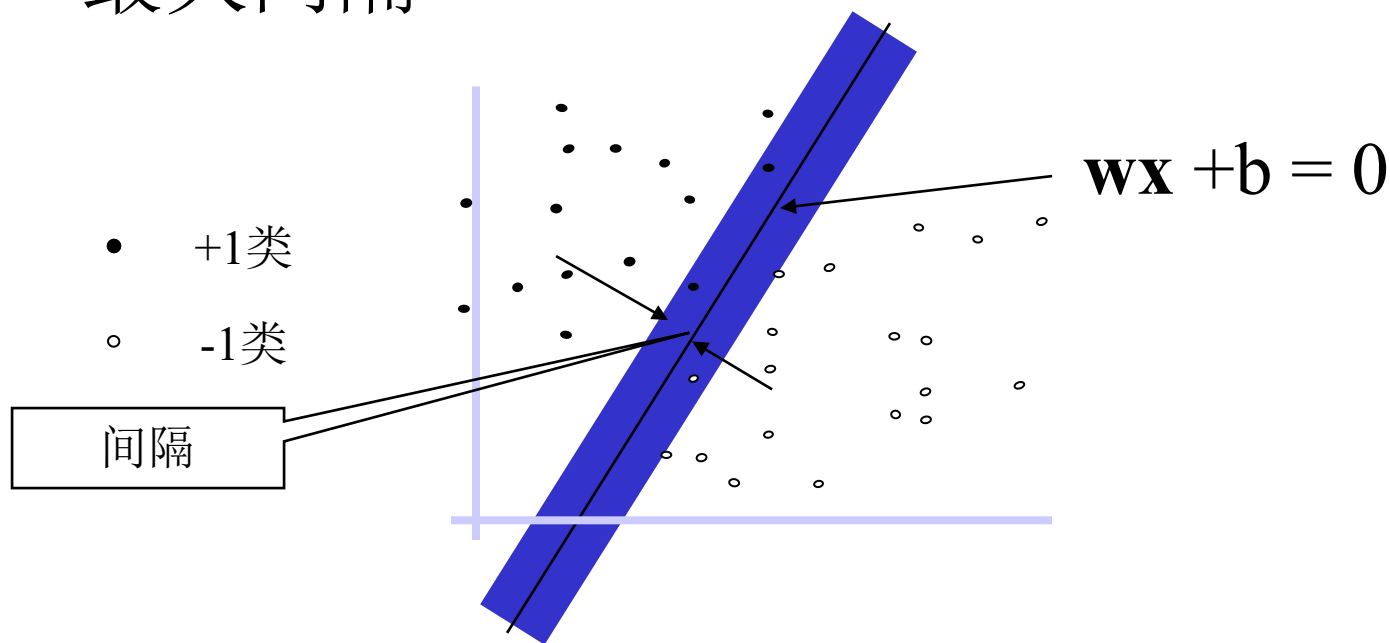


$$\operatorname{argmax}_{\mathbf{w}, b} \operatorname{margin}(\mathbf{w}, b, D)$$

$$= \operatorname{argmax}_{\mathbf{w}, b} \min_{\mathbf{x}_i \in D} d(\mathbf{x}_i)$$

$$= \operatorname{argmax}_{\mathbf{w}, b} \min_{\mathbf{x}_i \in D} \frac{y(\mathbf{x} \cdot \mathbf{w} + b)}{\sqrt{\sum_{i=1}^d w_i^2}}$$

最大间隔



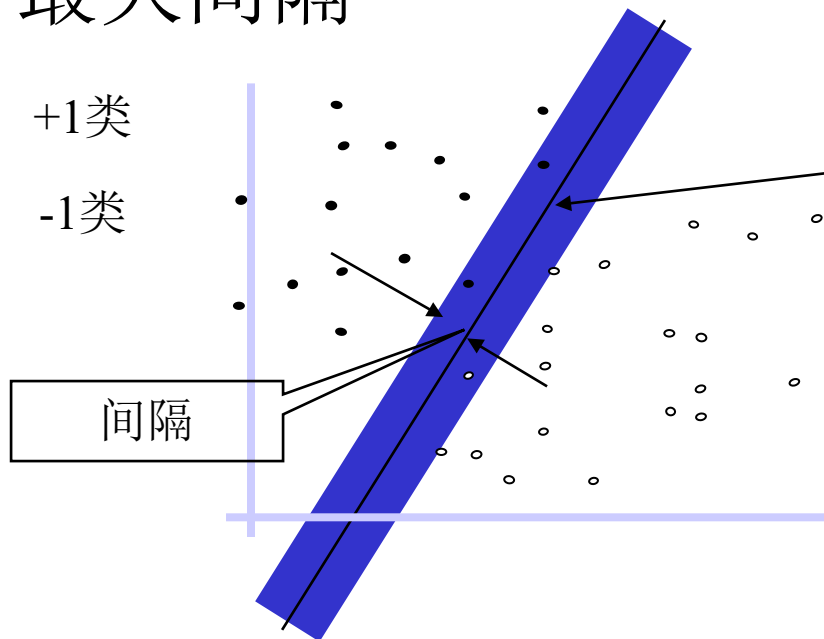
$$\operatorname{argmax}_{\mathbf{w}, b} \min_{\mathbf{x}_i \in D} \frac{y(\mathbf{x} \cdot \mathbf{w} + b)}{\sqrt{\sum_{i=1}^d w_i^2}}$$

$$\text{subject to } \forall \mathbf{x}_i \in D : y_i (\mathbf{x}_i \cdot \mathbf{w} + b) > 0$$

- 极小-极大问题

最大间隔

- +1类
- -1类



$$\mathbf{w}\mathbf{x} + b = 0$$

$$\operatorname{argmax}_{\mathbf{w}, b} \min_{\mathbf{x}_i \in D} \frac{y_i (\mathbf{x}_i \cdot \mathbf{w} + b)}{\sqrt{\sum_{i=1}^d w_i^2}}$$

$$\text{subject to } \forall \mathbf{x}_i \in D : y_i (\mathbf{x}_i \cdot \mathbf{w} + b) > 0$$



策略:

$$\forall \mathbf{x}_i \in D : |b + \mathbf{x}_i \cdot \mathbf{w}| \geq 1$$

$$\operatorname{argmin}_{\mathbf{w}, b} \sum_{i=1}^d w_i^2$$

$$\text{subject to } \forall \mathbf{x}_i \in D : y_i (\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$$

线性支持向量机：问题表示

$$\{\vec{w}^*, b^*\} = \operatorname{argmin}_{\vec{w}, b} \sum_{k=1}^d w_k^2$$

subject to

$$y_1 (\vec{w} \cdot \vec{x}_1 + b) \geq 1$$

$$y_2 (\vec{w} \cdot \vec{x}_2 + b) \geq 1$$

....

$$y_N (\vec{w} \cdot \vec{x}_N + b) \geq 1$$

二次规划问题

- (P2): 如何求解?

二次规划 (Quadratic Programming)

- QP: 目标函数是二次函数, 约束是线性等式或不等式。
- 是一种简单的凸优化问题, 较为简单, 便于求解:
 - Lagrange方法
 - 起作用集方法
 - Lemke方法
 - 路径跟踪法
- Matlab: quadprog, qpdpantz

二次规划

目标 $\arg \min_{\mathbf{u}} c + \mathbf{d}^T \mathbf{u} + \frac{\mathbf{u}^T R \mathbf{u}}{2}$ 二次函数

约束1

$$\begin{aligned} a_{11}u_1 + a_{12}u_2 + \dots + a_{1m}u_m &\leq b_1 \\ a_{21}u_1 + a_{22}u_2 + \dots + a_{2m}u_m &\leq b_2 \\ &\vdots \\ a_{n1}u_1 + a_{n2}u_2 + \dots + a_{nm}u_m &\leq b_n \end{aligned}$$

n 个线性不等式约束

约束2

$$\begin{aligned} a_{(n+1)1}u_1 + a_{(n+1)2}u_2 + \dots + a_{(n+1)m}u_m &= b_{(n+1)} \\ a_{(n+2)1}u_1 + a_{(n+2)2}u_2 + \dots + a_{(n+2)m}u_m &= b_{(n+2)} \\ &\vdots \\ a_{(n+e)1}u_1 + a_{(n+e)2}u_2 + \dots + a_{(n+e)m}u_m &= b_{(n+e)} \end{aligned}$$

e 个线性等式约束

二次规划

目标

$$\arg \min_{\mathbf{u}} c + \mathbf{d}^T \mathbf{u} + \frac{\mathbf{u}^T R \mathbf{u}}{2}$$

二次函数

约束1

$$a_{11}u_1 + a_{12}u_2 + \dots + a_{1m}u_m = b_1$$

约束2

$$a_{(n+2)1}u_1 + a_{(n+2)2}u_2 + \dots + a_{(n+2)m}u_m = b_{(n+2)}$$

:

$$a_{(n+e)1}u_1 + a_{(n+e)2}u_2 + \dots + a_{(n+e)m}u_m = b_{(n+e)}$$

线性不等式约束

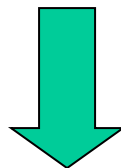
e 个线性等式约束

目前有各种各样的快速算法可以
被用来求解二次规划问题。
(但是这些方法都比较复杂和繁
琐可能大部分人都不会亲
自动手去实现一个具体的QP算
法)

(P2): 如何求解?

$$\operatorname{argmin}_{\mathbf{w}, b} \sum_{i=1}^d w_i^2$$

$$\text{subject to } \forall \mathbf{x}_i \in D : y_i (\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$$



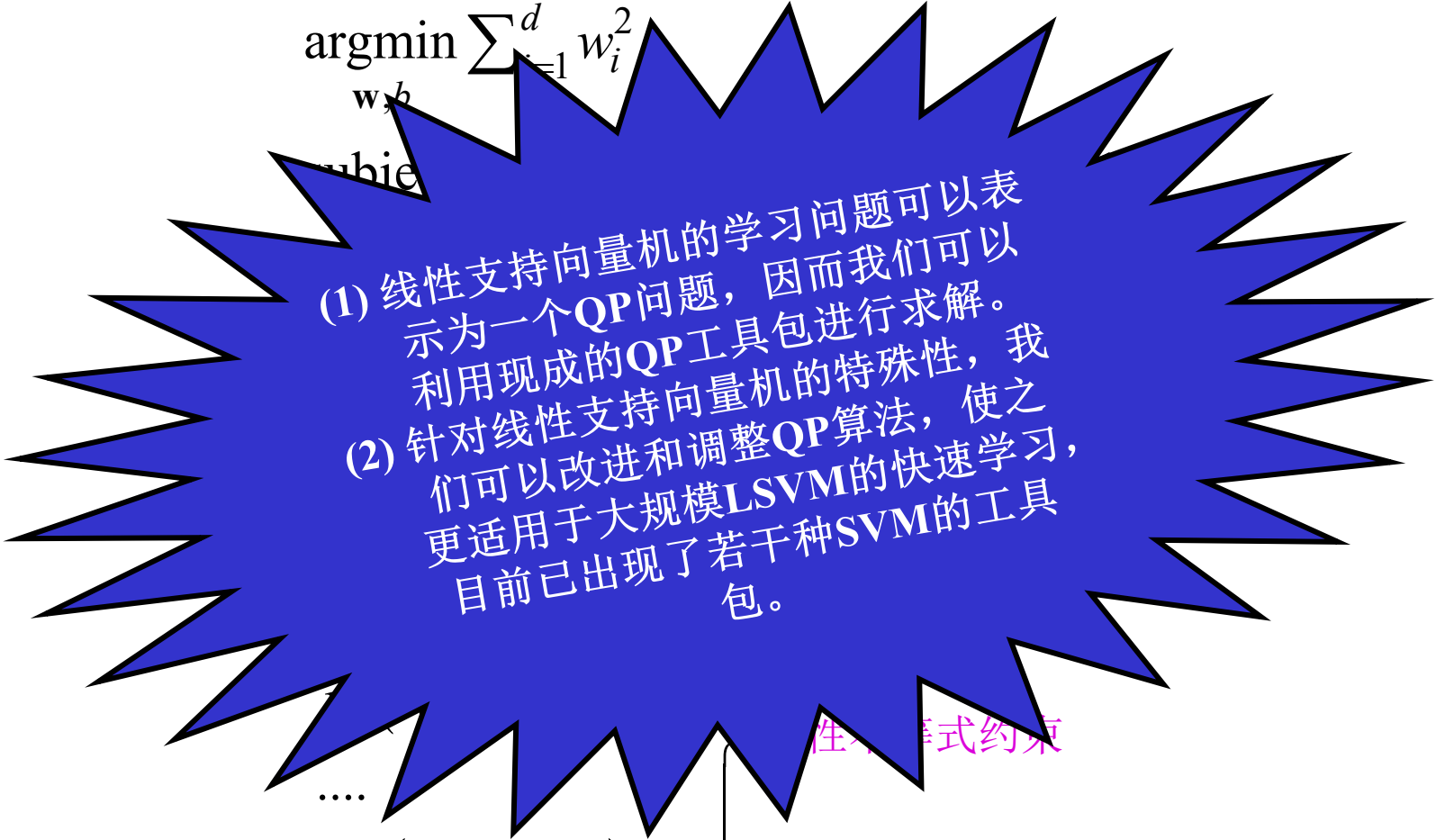
$$\{\vec{w}^*, b^*\} = \operatorname{argmin}_{\vec{w}, b} \left\{ 0 + \vec{0} \cdot \vec{w} + \vec{w}^T \mathbf{I}_n \vec{w} \right\}$$

$$\left. \begin{array}{l} y_1 (\vec{w} \cdot \vec{x}_1 + b) \geq 1 \\ y_2 (\vec{w} \cdot \vec{x}_2 + b) \geq 1 \\ \dots \\ y_N (\vec{w} \cdot \vec{x}_N + b) \geq 1 \end{array} \right\} \text{线性不等式约束}$$

(P2): 如何求解?

$$\operatorname{argmin}_{w, b} \sum_{i=1}^d w_i^2$$

subject to

- 
- (1) 线性支持向量机的学习问题可以表示为一个QP问题, 因而我们可以利用现成的QP工具包进行求解。
 - (2) 针对线性支持向量机的特殊性, 我们可以改进和调整QP算法, 使之更适用于大规模LSVM的快速学习, 目前已出现了若干种SVM的工具包。

线性不等式约束

....

$$y_N (\vec{w} \cdot \vec{x}_N + b) \geq 1$$

针对线性支持向量机的特殊性来改进和调整QP算法

- 支持向量机的性质
 - 支持向量
- 对偶问题
 - Lagrange函数
 - 推导过程
 - 最终形式

LSVM的对偶问题

$$\text{Max} \sum_{k=1}^R \alpha_k - \frac{1}{2} \sum_{k=1}^R \sum_{l=1}^R \alpha_k \alpha_l Q_{kl} \quad \text{其中} \quad Q_{kl} = y_k y_l (\mathbf{x}_k \cdot \mathbf{x}_l)$$

$$\text{Subject to:} \quad 0 \leq \alpha_k, \forall k \quad \sum_{k=1}^R \alpha_k y_k = 0$$

定义:

$$\mathbf{w} = \sum_{k=1}^R \alpha_k y_k \mathbf{x}_k$$

分类器:

$$f(\mathbf{x}, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$

Kuhn-Tucker构造法（以线性可分情况为例）

- 构造Lagrange函数

$$\max_{\boldsymbol{\alpha}} \min_{\mathbf{w}, b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i \left[y_i (\mathbf{w}^t \mathbf{x}_i + b) - 1 \right], \alpha_i \geq 0$$

- 分别对参数 \mathbf{w} 和 b 求导： $\alpha_i \left[y_i (\mathbf{w}^t \mathbf{x}_i + b) - 1 \right] = 0$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^n y_i \alpha_i \mathbf{x}_i = 0$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = \sum_{i=1}^n y_i \alpha_i = 0$$

Kuhn-Tucker构造法

- 因此有：
$$\mathbf{w} = \sum_{i=1}^n y_i \alpha_i \mathbf{x}_i$$

$$\sum_{i=1}^n y_i \alpha_i = 0$$

$$\alpha_i \left[y_i (\mathbf{w}^t \mathbf{x}_i + b) - 1 \right] = 0$$

- 代入Lagrange函数，有：

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$$

Kuhn-Tucker构造法

- 因此SVM的优化问题可以转化为一个经典的二次规划问题：

$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$$

约束条件：

$$\sum_{i=1}^n y_i \alpha_i = 0$$

$$\alpha_i \geq 0, \quad i = 1, \dots, n$$

SVM解的讨论

- 这是一个典型的不等式约束条件下的二次优化问题，其解法的基础是**Kuhn-Tucker定理**；
- 首先求解的是n个Lagrange乘子，n为训练样本数。但根据Kuhn-Tucker定理，有：

$$y_i (\mathbf{w}^t \mathbf{x}_i + b) > 1, \quad \alpha_i = 0$$

$$y_i (\mathbf{w}^t \mathbf{x}_i + b) = 1, \quad \alpha_i > 0$$

满足第2个条件的 y_i 称为**支持向量**。

SVM解的讨论

- 根据找到的支持向量 \mathbf{x}_i 以及相应的Lagrange乘子 α_i , 计算权向量 \mathbf{w} :

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$$

- √ 偏置 b 可以用支持向量满足的条件求得:

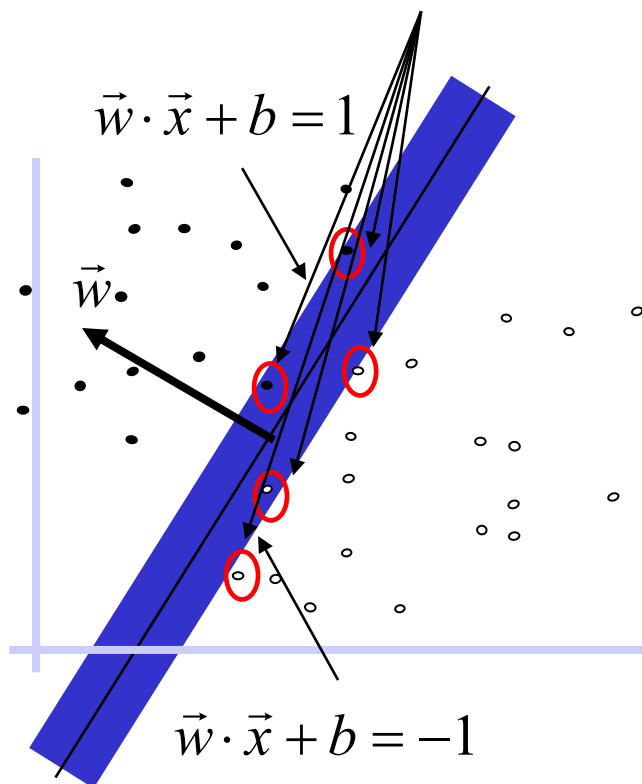
$$y_i (\mathbf{w}^t \mathbf{x}_i + b) = 1$$

支持向量

支持向量

$$\forall i: \alpha_i (y_i (\vec{w} \cdot \vec{x}_i + b) - 1) = 0$$

- denotes +1
- denotes -1



$\alpha_i = 0$ 非支持向量
 $\alpha_i \neq 0$ 支持向量

$$\mathbf{w} = \sum_{k=1}^R \alpha_k y_k \mathbf{x}_k$$

决策函数只有支持向量决定

LSVM的对偶问题

为什么要求解QP的对偶问题？

- 仍然是一个QP问题，但更容易求解；
- 结果可以表示为 α_i 和支持向量的函数，支持向量机的由来；
- 可以很方便的推广为非线性最大间隔支持向量机。

Max

$\sum_{i=1}^n$

α_i

C

then

\mathbf{w}

$=$

$\sum_{k=1}^n$

α_k

\mathbf{y}_k

\mathbf{x}_k

\mathbf{w}

$\cdot \mathbf{x} - \xi$

≤ 0

ξ

$\leq \epsilon$

ϵ

ϵ

$\Delta_i(\mathbf{x} \cdot \mathbf{x}_i)$

Δ

作业

- 推导下面问题的对偶问题

$$\min \mathbf{x}^T \mathbf{x}, \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}$$

- 对偶问题

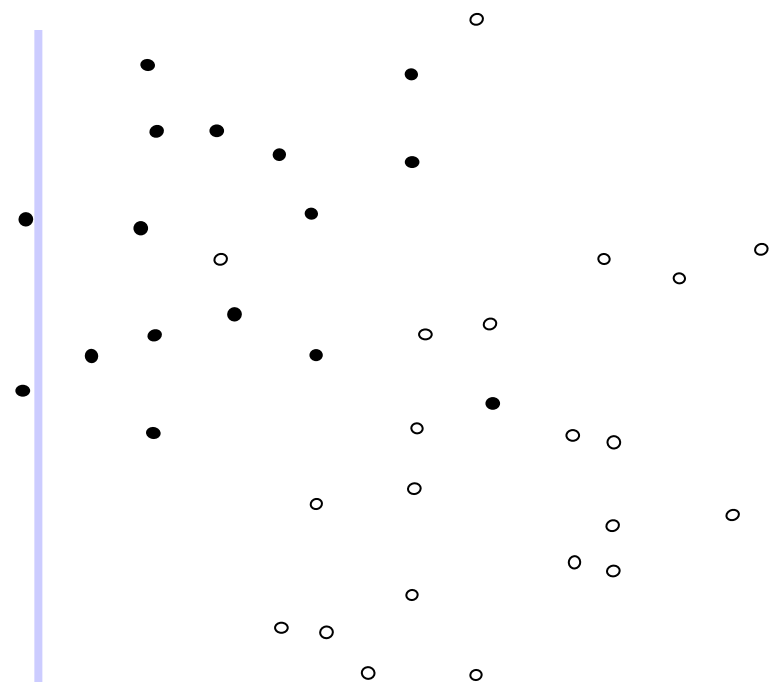
$$\max -\frac{1}{4} \boldsymbol{\alpha}^T \mathbf{A}\mathbf{A}^T \boldsymbol{\alpha} - \mathbf{b}^T \boldsymbol{\alpha}$$

线性不可分情形: An Interesting Comment

- +1类
- -1类

$$\operatorname{argmin}_{\mathbf{w}, b} \sum_{i=1}^d w_i^2$$

$$\text{subject to } \forall \mathbf{x}_i \in D : y_i (\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$$



$$\max L(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$$

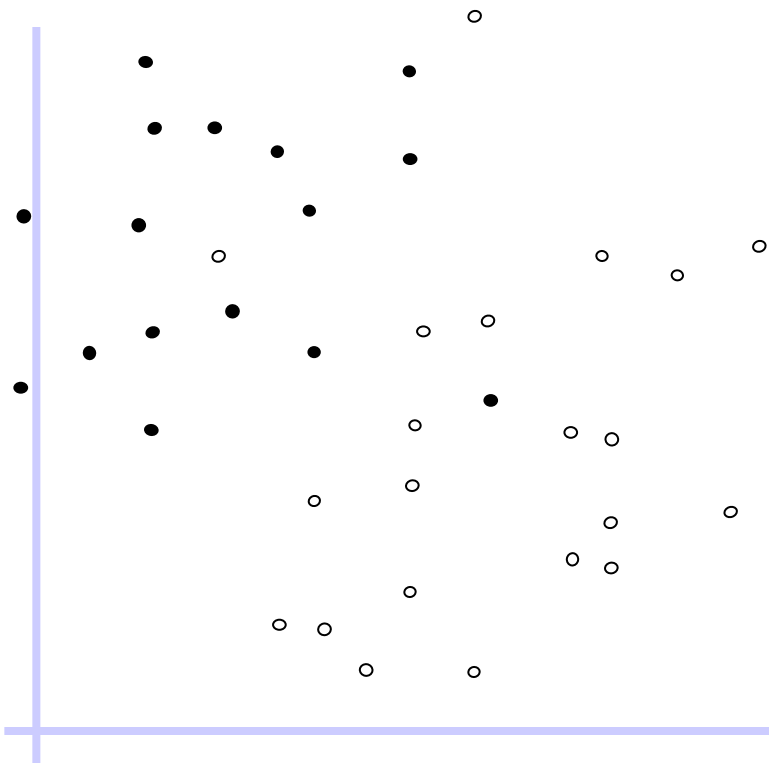
$$\sum_{i=1}^n y_i \alpha_i = 0$$

$$\alpha_i \geq 0, \quad i = 1, \dots, n$$

Kuhn-Tucker定理

(P3): 如何推广到线性不可分情形?

- +1类
- -1类



如何处理?

(P3): 如何推广到线性不可分情形?

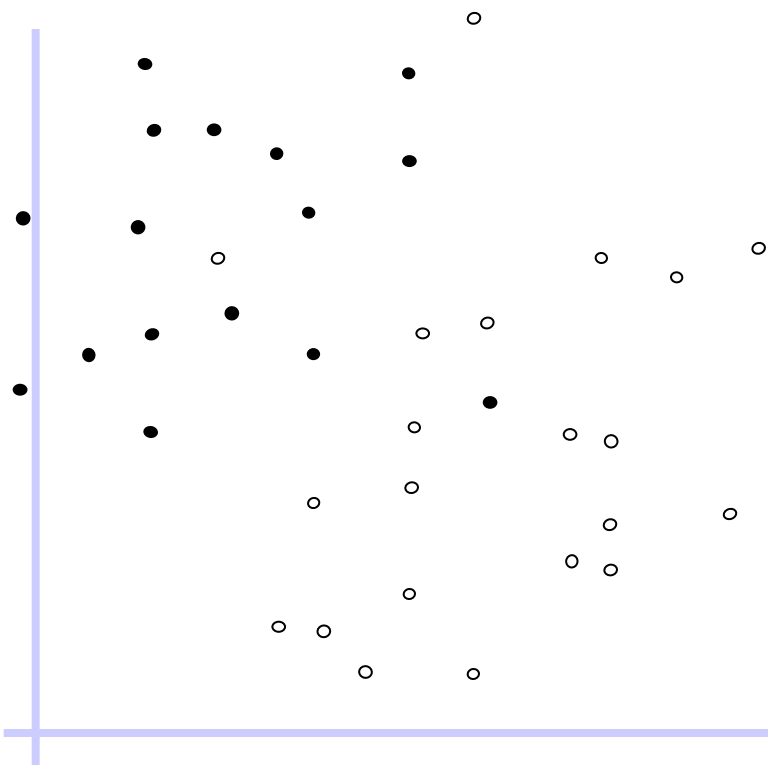
- +1类
- -1类

处理策略1:

多目标优化, 最小化 $w \cdot w$, 同时最小化在训练集上的错误样本数目。

问题1: 多目标优化, 难于求解;

问题2: 训练集上的错误样本数目不可导, 难于求解
(参考感知器算法部分的讨论)



(P3): 如何推广到线性不可分情形?

- +1类
- -1类

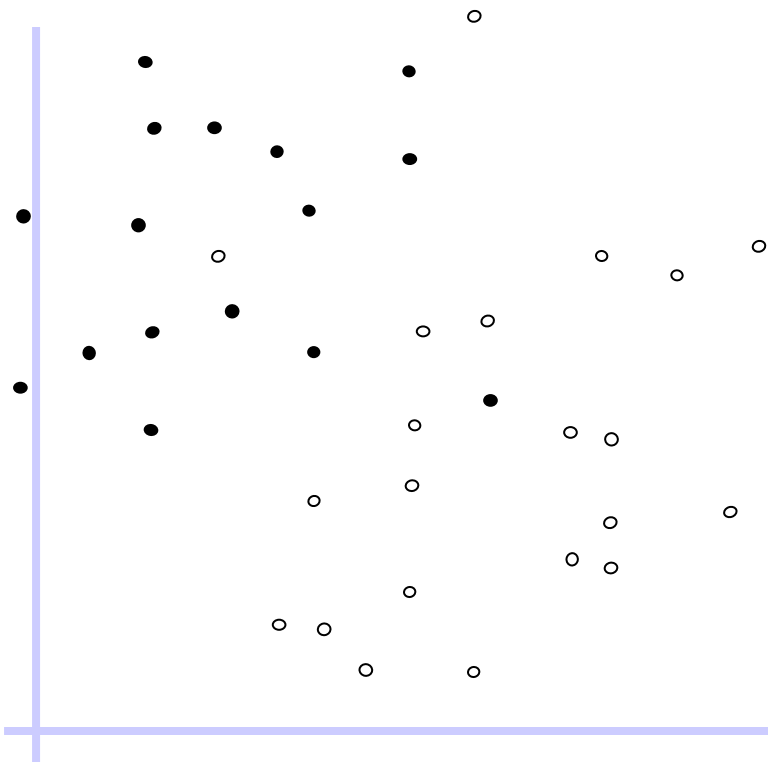
处理策略 1.1:

最小化

$$\mathbf{w} \cdot \mathbf{w} + C (\#train\ errors)$$

折中参数

问题: 虽然可以解决问题1,
但问题2依然存在



(P3): 如何推广到线性不可分情形?

- +1类
- -1类

处理策略 1.1:

最小化

$$w \cdot w + C (\#train\ errors)$$

Tradeoff parameter

目标函数不再是二次函数，问题也不再是二次规划问题。

(此外，错误分类样本和间隔大小的相互影响)

似问题！

施。
我们已经处理过类

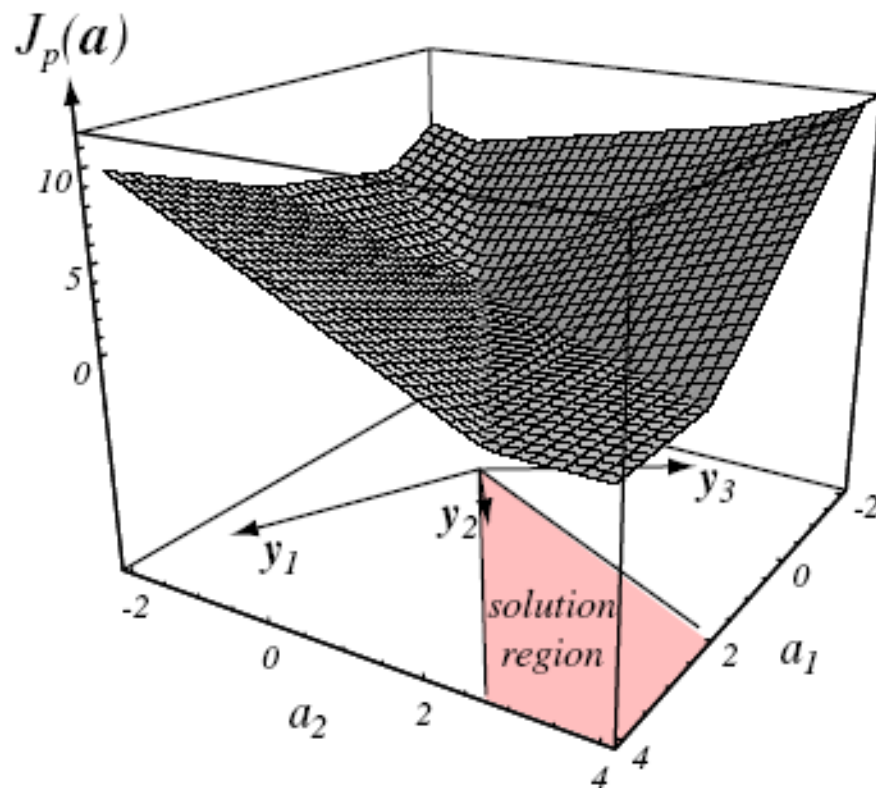
是否还有其
它方法？

感知器准则

- 以错分样本到判别界面距离之和作为准则：

$$J_P(\mathbf{w}) = \sum_{\mathbf{x} \in X} (-\mathbf{w}^t \mathbf{x})$$

$$\nabla J_P = \sum_{\mathbf{x} \in X} (-\mathbf{x})$$



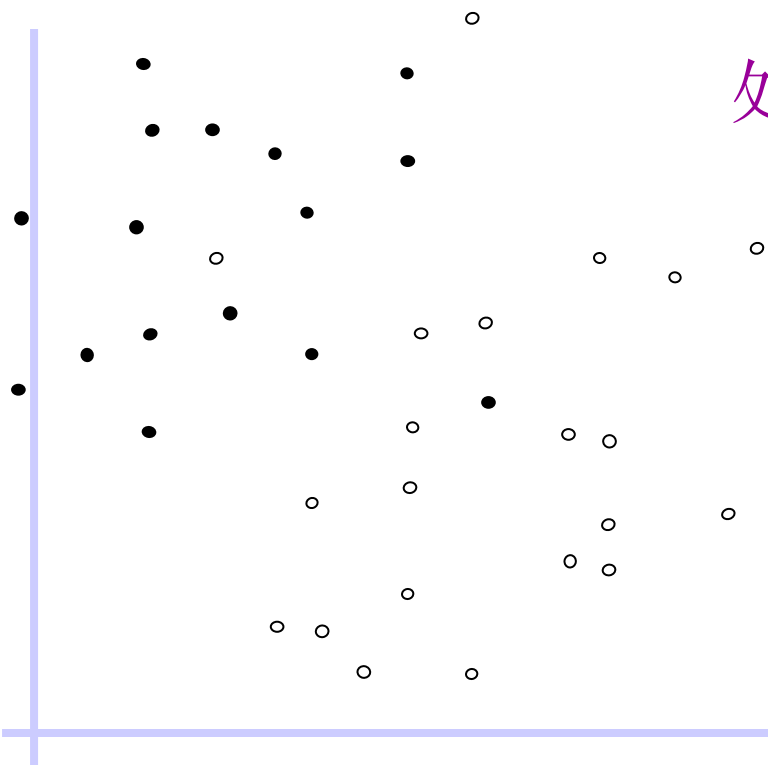
(P3): 如何推广到线性不可分情形?

- +1类
- -1类

处理策略 2.0:

最小化

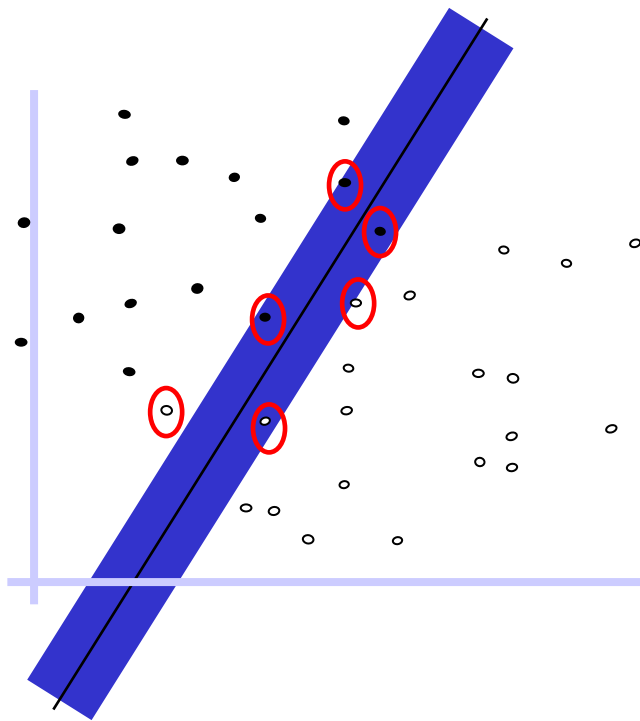
$w \cdot w + C$ (错误样本到正确的平面 (间隔面) 的距离)



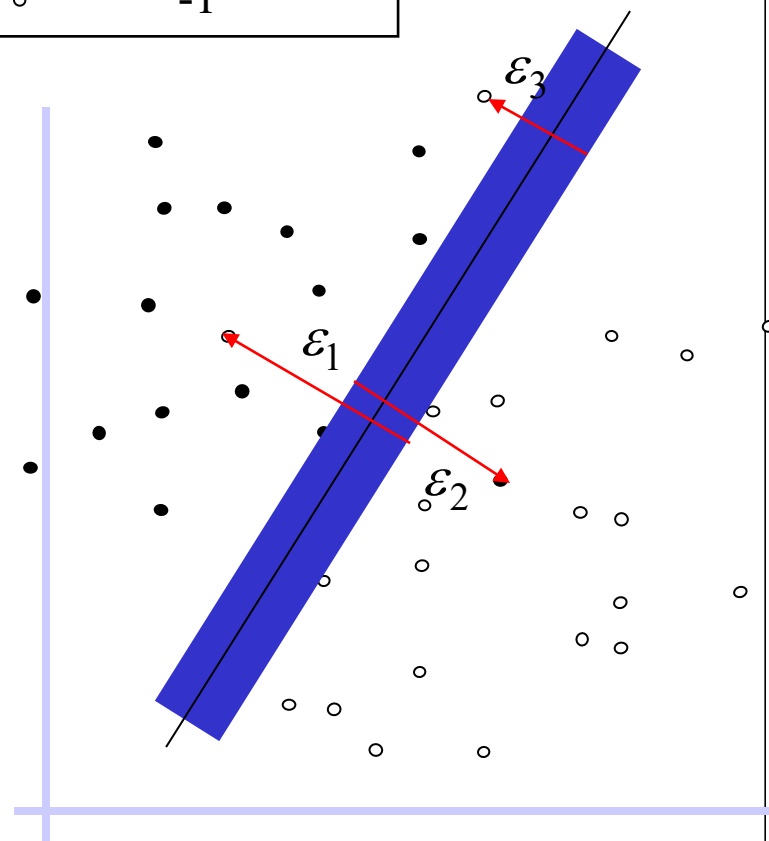
错误样本和正确的平面

•	+1
○	-1

• denotes +1
○ denotes -1



$$y_i (\vec{w} \cdot \vec{x}_i + b) \leq 1$$



$$\max \{1 - y_i (\vec{w} \cdot \vec{x}_i + b), 0\}$$

线性不可分情形下的LSVM问题

$$\{\vec{w}^*, b^*\} = \min_{\vec{w}, b, \vec{\varepsilon}} \sum_{i=1}^d w_i^2 + c \sum_{j=1}^N \varepsilon_j$$

$$y_1 (\vec{w} \cdot \vec{x}_1 + b) \geq 1 - \varepsilon_1, \varepsilon_1 \geq 0$$

$$y_2 (\vec{w} \cdot \vec{x}_2 + b) \geq 1 - \varepsilon_2, \varepsilon_2 \geq 0$$

...

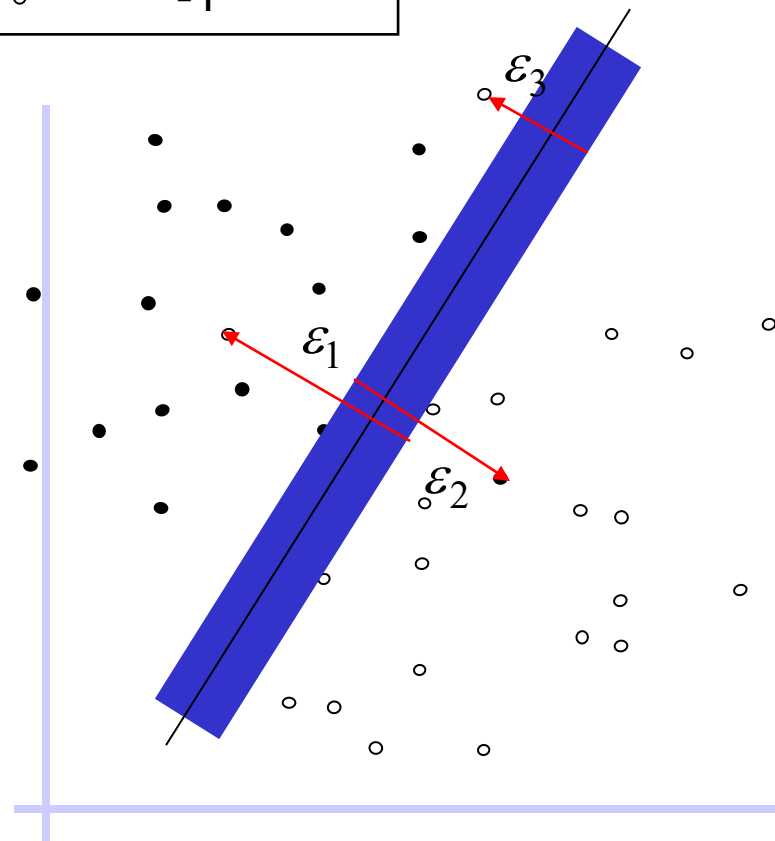
$$y_N (\vec{w} \cdot \vec{x}_N + b) \geq 1 - \varepsilon_N, \varepsilon_N \geq 0$$

❖ 仍然是一个QP问题！

❖ 问题：需要优化的变量？

❖ 辅助变量

•	+1
○	-1



线性不可分情形下的LSVM

$$\{\vec{w}^*, b^*\} = \operatorname{argmin}_{\vec{w}, b, \vec{\varepsilon}} \sum_i w_i^2 + c \sum_{j=1}^N \varepsilon_j$$

$$y_1 (\vec{w} \cdot \vec{x}_1 + b) \geq 1 - \varepsilon_1, \varepsilon_1 \geq 0$$

$$y_2 (\vec{w} \cdot \vec{x}_2 + b) \geq 1 - \varepsilon_2, \varepsilon_2 \geq 0$$

....

$$y_N (\vec{w} \cdot \vec{x}_N + b) \geq 1 - \varepsilon_N, \varepsilon_N \geq 0$$

线性不等式约束

求解算法与线性可分一样，同样也可以转化为求解对偶问题

线性不可分情形下的LSVM

$$\text{Max} \sum_{k=1}^R \alpha_k - \frac{1}{2} \sum_{k=1}^R \sum_{l=1}^R \alpha_k \alpha_l Q_{kl} \quad \text{其中} \quad Q_{kl} = y_k y_l (\mathbf{x}_k \cdot \mathbf{x}_l)$$

$$\text{Subject to:} \quad 0 \leq \alpha_k \leq c, \forall k \quad \sum_{k=1}^R \alpha_k y_k = 0$$

定义:

$$\mathbf{w} = \sum_{k=1}^R \alpha_k y_k \mathbf{x}_k$$

分类器:

$$f(\mathbf{x}, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$

支持向量

- KKT条件

$$y_i (\mathbf{w}^t \mathbf{x}_i + b) > 1, \quad \alpha_i = 0$$

$$y_i (\mathbf{w}^t \mathbf{x}_i + b) < 1, \quad c > \alpha_i > 0$$

$$y_i (\mathbf{w}^t \mathbf{x}_i + b) = 1, \quad \alpha_i = c$$

阅读材料

- Christopher J.C. Burges
 - A tutorial on support vector machines for pattern recognition
 - Data Mining and Knowledge Discovery, 2, 121–167 (1998)
while at present there exists no theory which shows that good generalization performance is guaranteed for SVMs
- 支持向量机导论（英文版）
 - 克里斯蒂亚尼尼（Cristianini, N.）等著
 - 机械工业出版社
 - 2005-7-1

解释：支持向量机

- 最大间隔分类器
- 支持向量机
- 支撑向量机
- **思考**：为什么不采用增广（齐次）坐标形式？

Logistic回归

- 两类问题
 - 基本原理
 - 学习算法
- 多类问题
 - 自己解决

线性判别学习

- 以两类问题为例
- 判别学习的目标：后验概率

$$P(y=i|\mathbf{x}), i=0,1$$

- 分类器：

$$F(\mathbf{x}) = \log \left(\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} \right) \begin{cases} \geq 0, & \mathbf{x} \in \omega_1 \\ < 0, & \mathbf{x} \in \omega_0 \end{cases}$$

- 决策面： $F(\mathbf{x})=0$
- 线性判别学习（要求）
 - 决策面是一个（超）平面
 - 即：判别函数 $F(\mathbf{x})$ 是一个线性函数

$$F(\mathbf{x}) = \mathbf{w}^t \mathbf{x}$$

Logistic回归：两类问题

- 数据采用齐次（增广）表示形式
- 假设样本的后验概率为

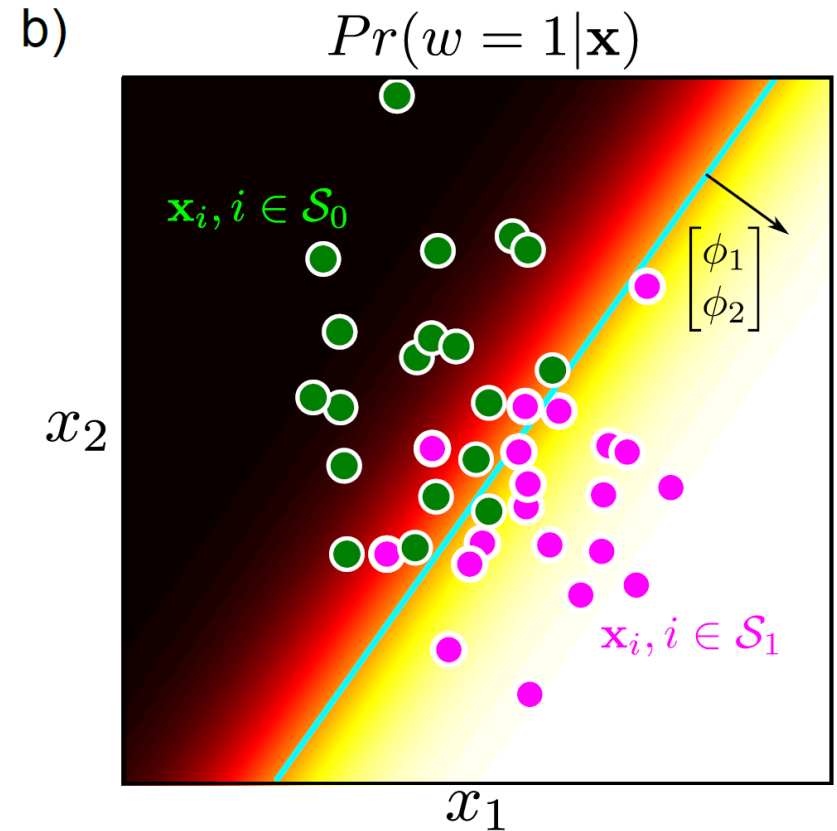
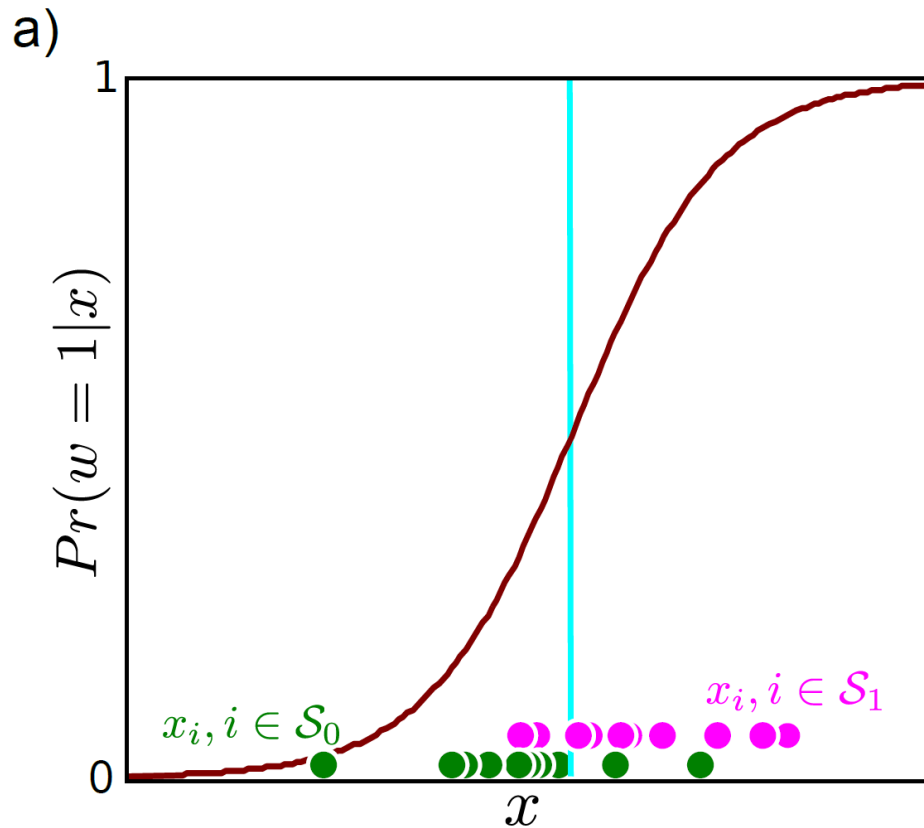
$$P(y=1|\mathbf{x}) = \frac{\exp(\mathbf{w}^t \mathbf{x})}{1 + \exp(\mathbf{w}^t \mathbf{x})}$$

$$P(y=0|\mathbf{x}) = \frac{1}{1 + \exp(\mathbf{w}^t \mathbf{x})}$$

- 则判别函数

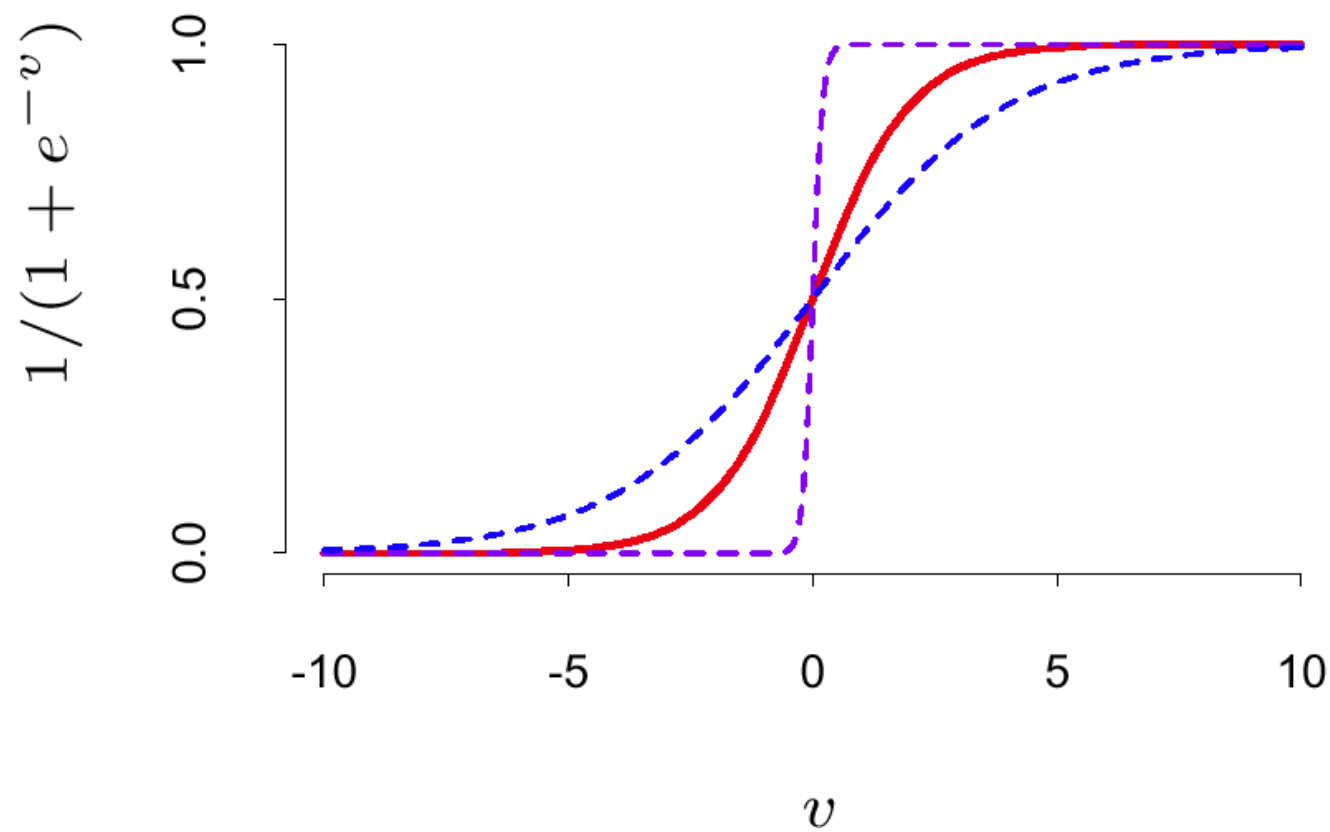
$$F(\mathbf{x}) = \log \left(\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} \right) = \mathbf{w}^t \mathbf{x}$$

Logistic回归



$$Pr(w|\phi, \mathbf{x}) = \text{Bern}_w \left[\frac{1}{1 + \exp[-\phi^T \mathbf{x}]} \right]$$

Logistic函数图示



目标函数

- 最大似然法 (ML)

- 训练样本

$$X : \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_N\}$$

$$Y : \{y_1, y_2, \cdots, y_i, \cdots, y_N\}, \quad y_i \in \{0, 1\}$$

- 似然函数

$$L(\mathbf{w}) = \prod_{i=1}^N P(y_i | \mathbf{x}_i, \mathbf{w})$$

目标函数

- 对数似然函数

$$\begin{aligned}\log(L(\mathbf{w})) &= \sum_{i=1}^N \log(P(y_i | \mathbf{x}_i, \mathbf{w})) \\ &= \sum_{i=1}^N \left\{ y_i \mathbf{w}^t \mathbf{x}_i - \log[1 + \exp(\mathbf{w}^t \mathbf{x}_i)] \right\}\end{aligned}$$

- 学习问题

$$\hat{\mathbf{w}} = \arg \max_{\mathbf{w}} \log(L(\mathbf{w}))$$

矩阵和向量导数

$$\frac{\partial(\mathbf{b}^T \mathbf{a})}{\partial \mathbf{a}} = \mathbf{b}$$

$$\frac{\partial(\mathbf{a}^T \mathbf{A} \mathbf{a})}{\partial \mathbf{a}} = (\mathbf{A} + \mathbf{A}^T) \mathbf{a}$$

$$\frac{\partial}{\partial \mathbf{A}} \text{tr}(\mathbf{B} \mathbf{A}) = \mathbf{B}^T$$

$$\frac{\partial}{\partial \mathbf{A}} \log |\mathbf{A}| = \mathbf{A}^{-T} \triangleq (\mathbf{A}^{-1})^T$$

$$\text{tr}(\mathbf{ABC}) = \text{tr}(\mathbf{CAB}) = \text{tr}(\mathbf{BCA})$$

学习算法

- 最优化条件

$$\begin{aligned}\frac{\partial \log(L(\mathbf{w}))}{\partial \mathbf{w}} &= \sum_{i=1}^N \mathbf{x}_i \left\{ y_i - \frac{\exp(\mathbf{w}^t \mathbf{x}_i)}{1 + \exp(\mathbf{w}^t \mathbf{x}_i)} \right\} \\ &= \sum_{i=1}^N \mathbf{x}_i \{ y_i - P(y = 1 | \mathbf{x}_i) \} = 0\end{aligned}$$

- 无法直接求解上述方程

学习算法：Newton–Raphson算法

- 一阶导数： $p_i = P(y = 1 | \mathbf{x}_i)$

$$\frac{\partial \log(L(\mathbf{w}))}{\partial \mathbf{w}} = \sum_{i=1}^N \mathbf{x}_i \{y_i - P(y = 1 | \mathbf{x}_i)\} = \sum_{i=1}^N \mathbf{x}_i (y_i - p_i),$$

- 二阶导数

$$\frac{\partial^2 \log(L(\mathbf{w}))}{\partial \mathbf{w} \partial \mathbf{w}^t} = - \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^t p_i (1 - p_i)$$

- 更新规则

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(k)} + \left(\frac{\partial^2 \log(L(\mathbf{w}))}{\partial \mathbf{w} \partial \mathbf{w}^t} \right)^{-1} \frac{\partial \log(L(\mathbf{w}))}{\partial \mathbf{w}}$$

Newton–Raphson算法：矩表示

- 一阶导数： \mathbf{X} ($N * (d+1)$ 矩阵), \mathbf{p} ($N*1$ 向量)

$$\frac{\partial \log(L(\mathbf{w}))}{\partial \mathbf{w}} = \mathbf{X}^t (\mathbf{y} - \mathbf{p})$$

- 二阶导数： \mathbf{W} (对角矩阵, $\mathbf{W}_{ii} = \mathbf{p}_i(1-\mathbf{p}_i)$)

$$\frac{\partial^2 \log(L(\mathbf{w}))}{\partial \mathbf{w} \partial \mathbf{w}^t} = -\mathbf{X}^t \mathbf{W} \mathbf{X}$$

- 更新规则

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(k)} + \left(\mathbf{X}^t \mathbf{W} \mathbf{X} \right)^{-1} \mathbf{X}^t (\mathbf{y} - \mathbf{p})$$

注释/思考

- 矩阵的上界？

$$-\frac{\partial^2 \log(L(\mathbf{w}))}{\partial \mathbf{w} \partial \mathbf{w}^t} = \mathbf{X}^t \mathbf{W} \mathbf{X}$$

- 意义何在？

多类问题

$$P(y=1|\mathbf{x}) = \frac{\exp(\mathbf{w}_1^t \mathbf{x})}{1 + \sum_{i=1}^{C-1} \exp(\mathbf{w}_i^t \mathbf{x})}$$

$$P(y=2|\mathbf{x}) = \frac{\exp(\mathbf{w}_1^t \mathbf{x})}{1 + \sum_{i=1}^{C-1} \exp(\mathbf{w}_i^t \mathbf{x})}$$

•
•
•

$$P(y=C|\mathbf{x}) = \frac{1}{1 + \sum_{i=1}^{C-1} \exp(\mathbf{w}_i^t \mathbf{x})}$$

多类问题

$$\log \frac{P(y=1|\mathbf{x})}{P(y=C|\mathbf{x})} = \mathbf{w}_1^t \mathbf{x}$$

$$\log \frac{P(y=2|\mathbf{x})}{P(y=C|\mathbf{x})} = \mathbf{w}_2^t \mathbf{x}$$

•
•
•

$$\log \frac{P(y=C-1|\mathbf{x})}{P(y=C|\mathbf{x})} = \mathbf{w}_{C-1}^t \mathbf{x}$$

作业

- 请推导并给出多类logistic回归的学习算法。
- 如果有什么问题，可以参考以下材料：
 - C. Bishop, Pattern Recognition and Machine Learning, Springer, 2008. (pp. 209 - 210)

进一步阅读

- Trevor Hastie, Robert Tibshirani, Jerome Friedman, 《统计学习基础——数据挖掘、推理和预测》，电子工业出版社，2004
- 阅读pp. **67-71**

应用：鸢尾属(Iris)植物分类

- (Anderson 1935)

150种鸢尾属植物 (每类50种):

u 类别(Species): setosa, versicolor, virginica;

特征:

u 萼片(sepals)的长度和宽度 (cm);

u 花瓣(petals)的长度和宽度 (cm);



iris setosa



iris versicolor

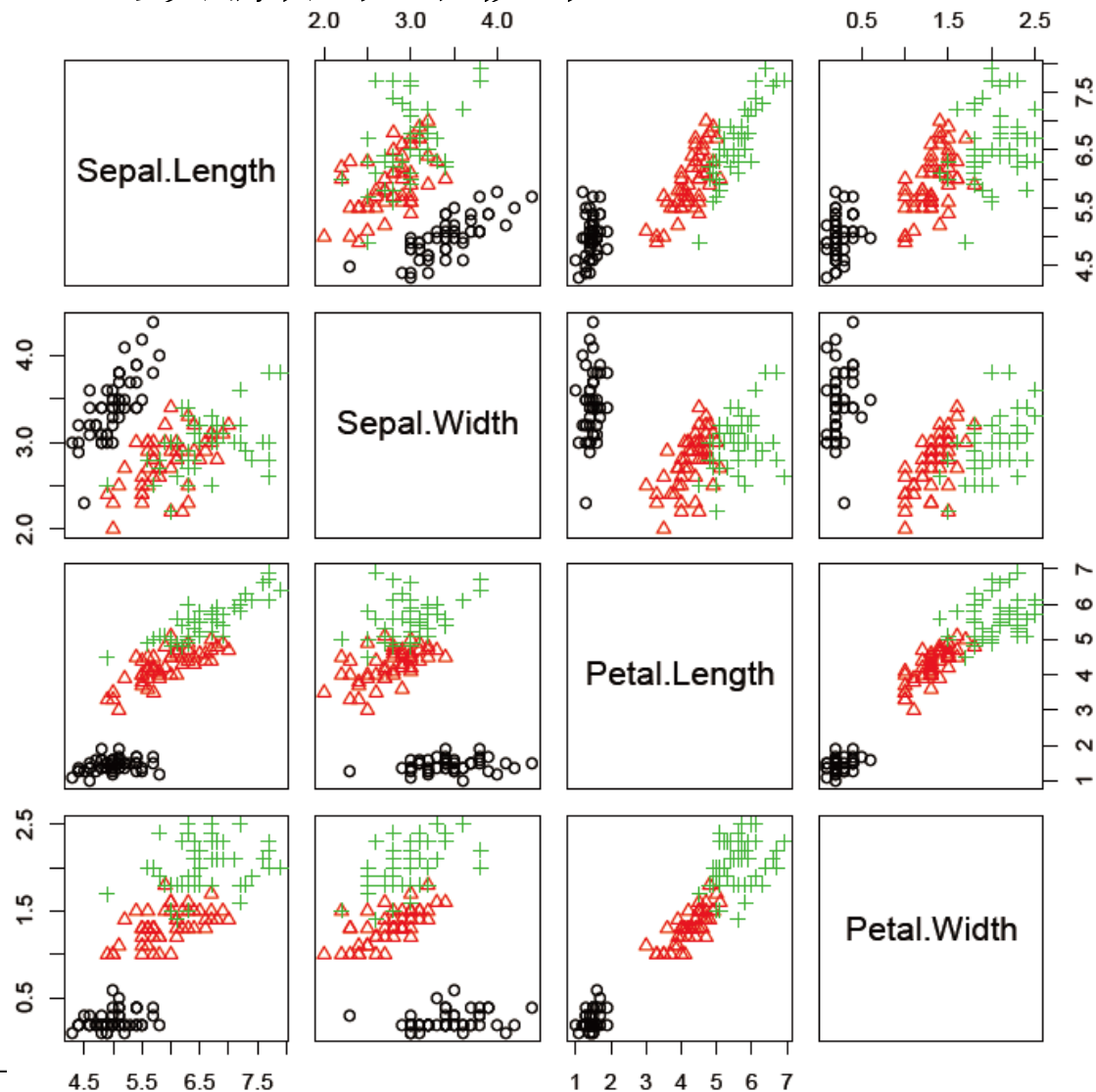
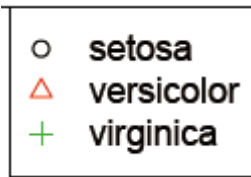


iris virginica

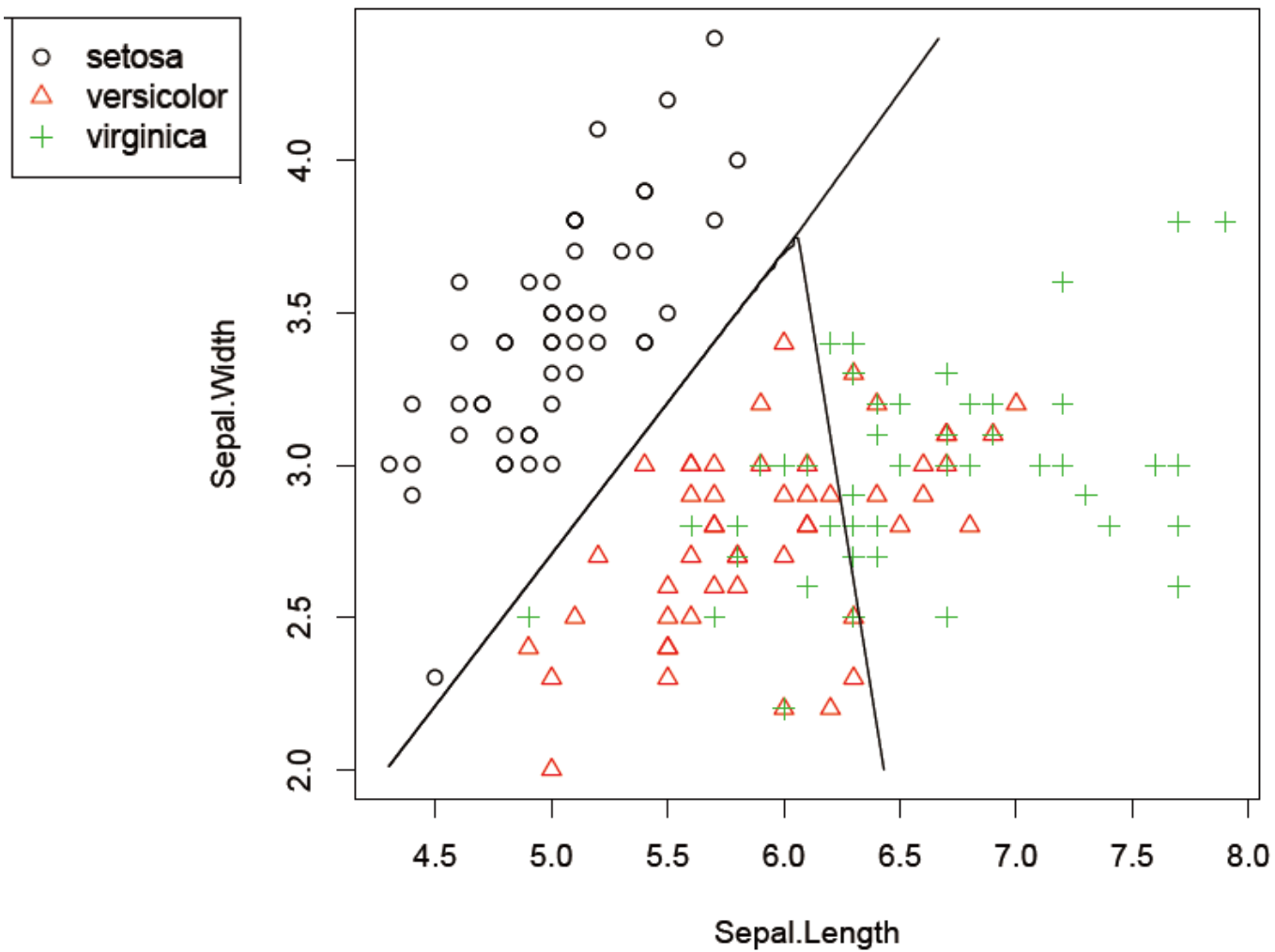
数据

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.10	3.50	1.40	0.20	setosa
2	4.90	3.00	1.40	0.20	setosa
3	4.70	3.20	1.30	0.20	setosa
4	4.60	3.10	1.50	0.20	setosa
5	5.00	3.60	1.40	0.20	setosa
:	:	:	:	:	
51	7.00	3.20	4.70	1.40	versicolor
52	6.40	3.20	4.50	1.50	versicolor
53	6.90	3.10	4.90	1.50	versicolor
54	5.50	2.30	4.00	1.30	versicolor
:	:	:	:	:	
101	6.30	3.30	6.00	2.50	virginica
102	5.80	2.70	5.10	1.90	virginica
103	7.10	3.00	5.90	2.10	virginica
104	6.30	2.90	5.60	1.80	virginica
105	6.50	3.00	5.80	2.20	virginica
:	:	:	:	:	
150	5.90	3.00	5.10	1.80	virginica

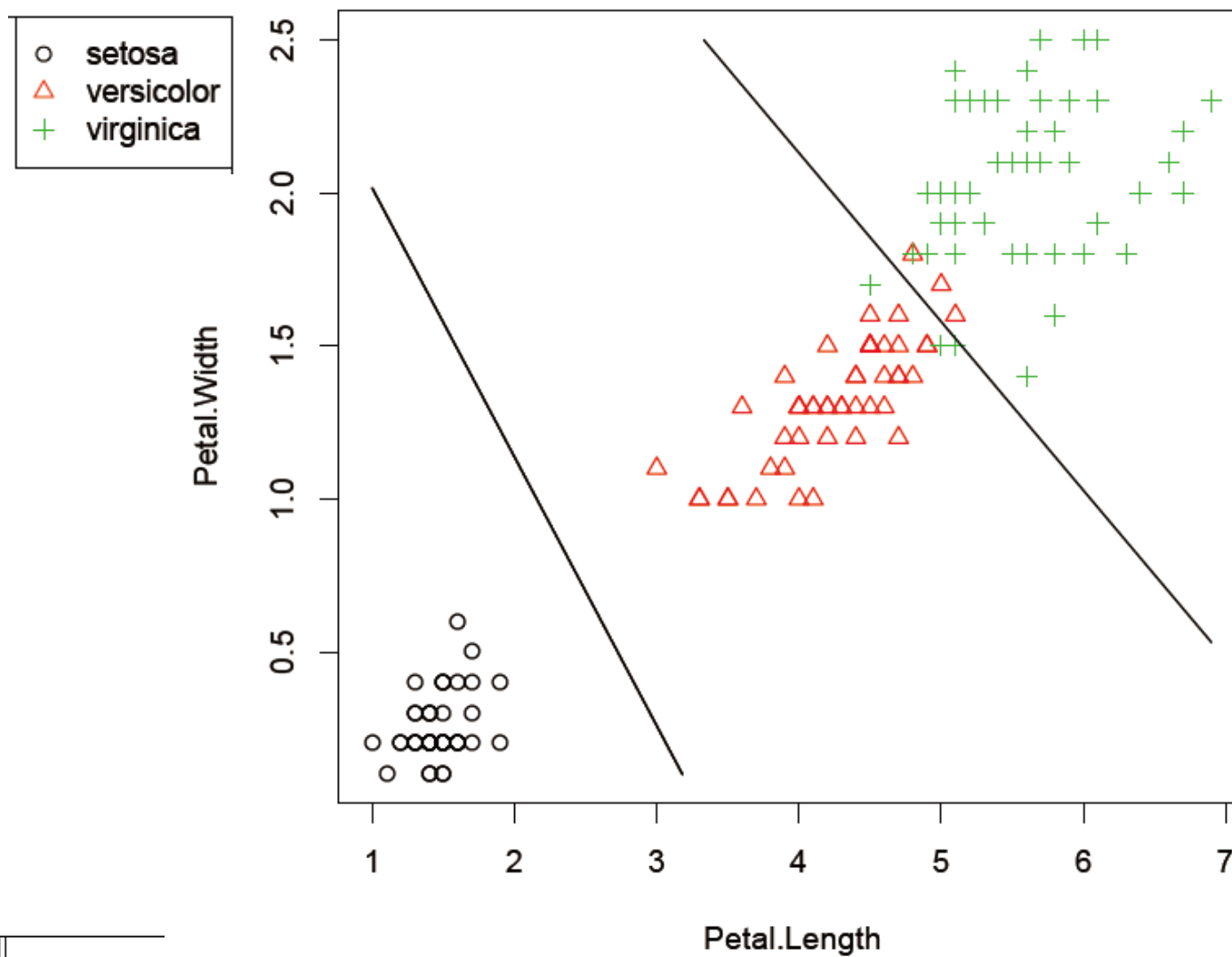
数据的可视化



分类结果

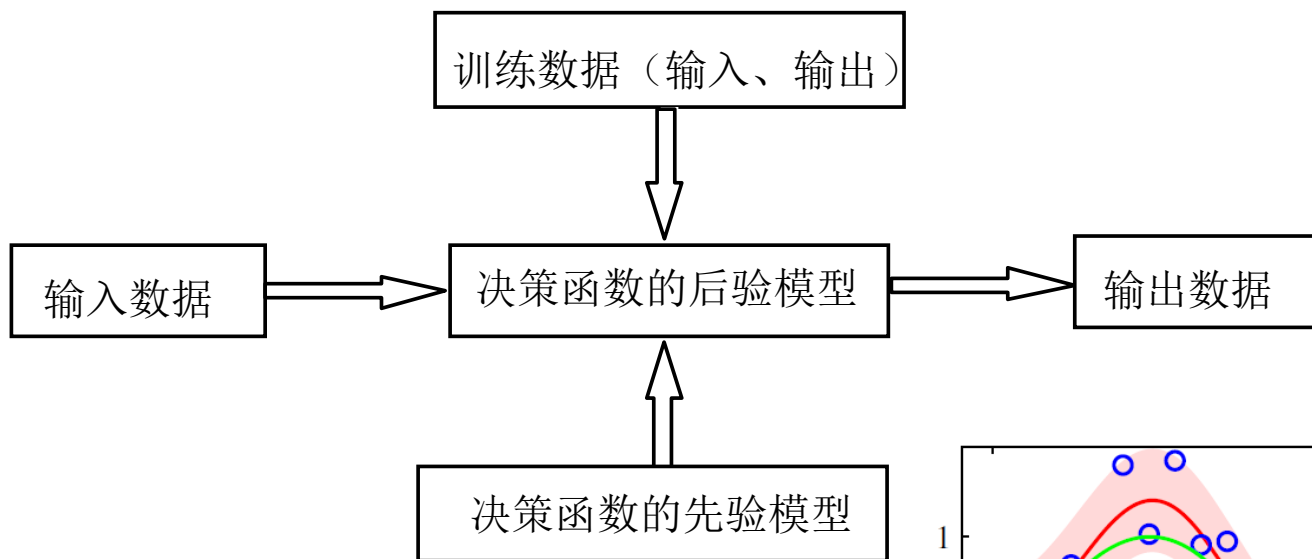


分类结果

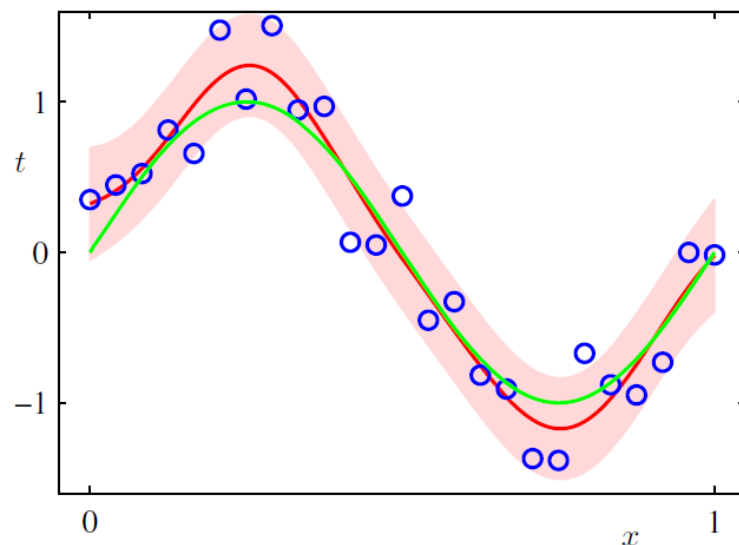


模式识别的贝叶斯观点 (1)

- 基于贝叶斯准则来训练决策函数



- 决策函数的概率解释



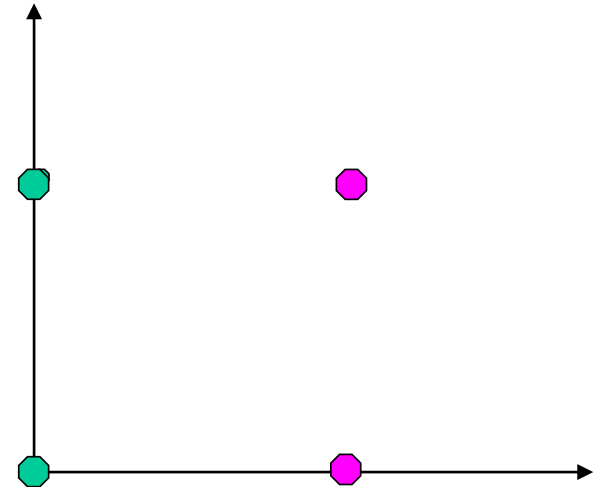
最大似然估计

- Logistic回归
 - 重点：参数 \mathbf{w} 的估计
- 最大似然估计

$$\begin{aligned}\hat{\mathbf{w}} &= \arg \max_{\mathbf{w}} \log(L(\mathbf{w})) = \sum_{i=1}^N \log(P(y_i | \mathbf{x}_i, \mathbf{w})) \\ &= \sum_{i=1}^N \left\{ y_i \mathbf{w}^t \mathbf{x} - \log[1 + \exp(\mathbf{w}^t \mathbf{x})] \right\}\end{aligned}$$

The Important of Prior

- Logistic regression (ML估计) fails on linear separable problems



贝叶斯(最大后验概率估计)

- 假设参数 \mathbf{w} 的先验服从正态分布

$$P(\mathbf{w}) = \left(\frac{\sqrt{\lambda}}{2\pi} \right)^n \exp \left(-\frac{\lambda \mathbf{w}^t \mathbf{w}}{2} \right)$$

- 则有

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} \sum_{i=1}^N \left\{ \log[1 + \exp(\mathbf{w}^t \mathbf{x})] - y_i \mathbf{w}^t \mathbf{x} \right\} + \frac{\lambda}{2} \mathbf{w}^t \mathbf{w}$$

Newton–Raphson算法：矩表示

- 一阶导数： \mathbf{X} ($N * (d+1)$ 矩阵)， \mathbf{p} ($N*1$ 向量)

$$\frac{\partial \log(L(\mathbf{w}))}{\partial \mathbf{w}} = -\mathbf{X}^t (\mathbf{y} - \mathbf{p}) + \lambda \mathbf{w}$$

- 二阶导数： \mathbf{W} (对角矩阵, $\mathbf{W}_{ii} = p_i(1-p_i)$)

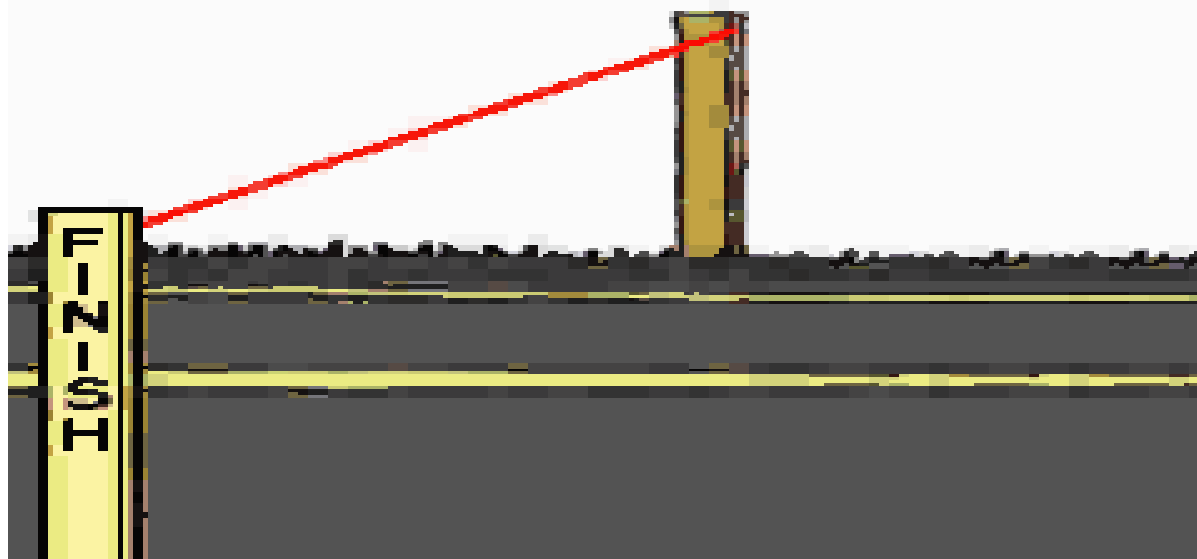
$$\frac{\partial^2 \log(L(\mathbf{w}))}{\partial \mathbf{w} \partial \mathbf{w}^t} = \mathbf{X}^t (\mathbf{W}) \mathbf{X} + \lambda \mathbf{I}$$

- 更新规则

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(k)} + \left(\mathbf{X}^t (\mathbf{W}) \mathbf{X} + \lambda \mathbf{I} \right)^{-1} \left[\mathbf{X}^t (\mathbf{y} - \mathbf{p}) - \lambda \mathbf{w} \right]$$

Project Two

- 利用Spectf Heart数据集（基于单光子发射型计算机断层仪数据的心脏疾病诊断）
 - 训练样本：SPECTF.train
 - 测试样本：SPECTF.test
- 利用Logistic回归方法，尝试训练一个线性分类器，并给出在测试集上的分类精度。
- 尝试引入 w 的先验并分析先验项的影响。
 - Kurgan, L.A., Cios, K.J., Tadeusiewicz, R., Ogiela, M. & Goodenday, L.S. "Knowledge Discovery Approach to Automated Cardiac SPECT Diagnosis" Artificial Intelligence in Medicine, vol. 23:2, pp 149-169, Oct 2001



END