

机器学习

Machine Learning

北京航空航天大学计算机学院

School of Computer Science and Engineering, Beihang University

黄迪 张永飞 陈佳鑫

2023年秋季学期

Fall 2023

第4章：线性模型

Chapter 4: Linear Models

机器学习算法

机器学习主要问题

	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Discrete</i>	Classification or Categorization	Clustering
<i>Continuous</i>	Regression	Dimensionality Reduction

“回归”起源

● Francis Galton

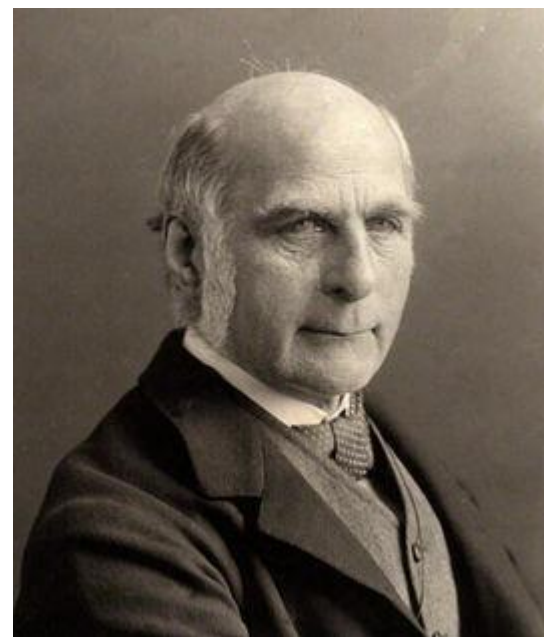
英国生物学家

研究父母身高与子女身高之间关系

若父母身高高于平均身高，则其子女身高倾向于倒退生长，即会比其父母身高矮一些而更接近于大众平均身高。

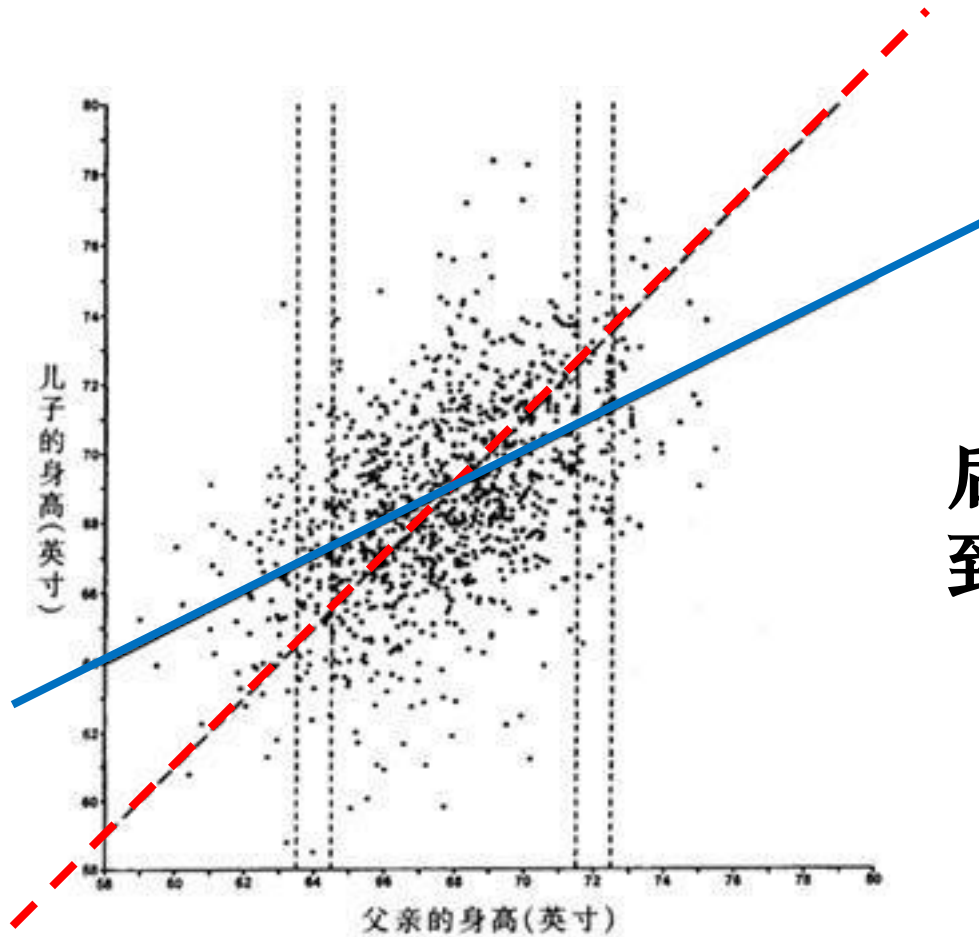
若父母身高小于平均身高，则其子女身高倾向于向上生长，即会更接近于大众平均身高。

此现象被Galton称之为回归现象，即
Regression.



Francis Galton
1822.2.16-1911.1.17

“回归”起源



后代的身高倾向于“回归”
到一个**平均值**。

回归问题举例

● 例：房价估计

Size (feet ²)	Number of bedrooms	Number of floors	Age of home (years)	Price (\$1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...

Size (feet ²)	Number of bedrooms	Number of floors	Age of home (years)	Price (\$1000)
1500	3	2	30	?

测试样本

回归问题举例

● 例：房价估计

训练样本

Size (feet ²)	Number of bedrooms	Number of floors	Age of home (years)	Price (\$1000)	
x_1	x_2	x_3	x_4	y	
2104	5	1	45	460	} N
1416	3	2	40	232	
1534	3	2	30	315	
852	2	1	36	178	
...	

监督学习

N : 训练样本个数

x : 输入变量/“特征”

y : 输出变量/目标变量

回归模型

训练集 (x_i, y_i)

学习算法

变量(特征) x \rightarrow f \rightarrow 计算 y

$$f(\mathbf{x}) = w_0 + w_1x_1 + \dots + w_mx_m$$

$$= w_0 + \sum_{i=1}^m w_ix_i$$

$$= \sum_{i=0}^m w_ix_i \quad [x_0 = 1]$$

如何表示 f ?

线性回归:
假设函数 f 为输入 x 的线性函数

$$\begin{aligned} f(\mathbf{x}) &= w_0 + w_1x_1 + \dots + w_mx_m \\ &= w_0 + \sum_{j=1}^m w_jx_j \end{aligned}$$

写成向量形式:
增加一维 $x_0=1$, 表示截距项

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

回归模型

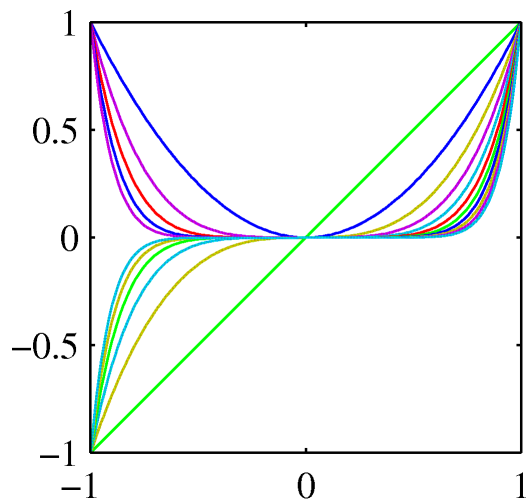
$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

基函数

最简单的情况下: $\phi_j(x) = x_j$

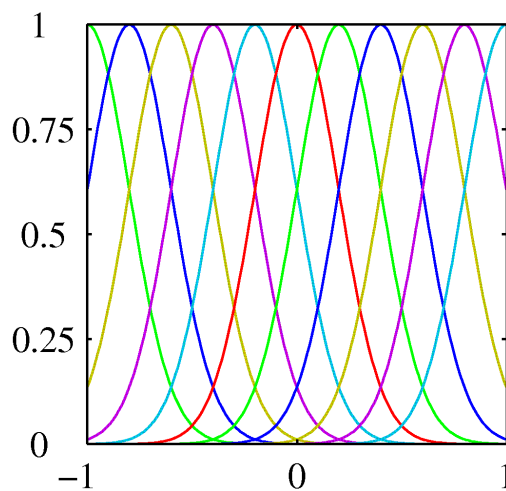
● 多项式基函数

$$\phi_j(x) = x^j$$



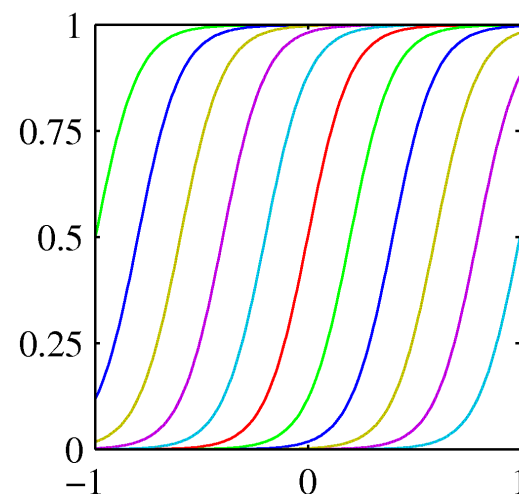
● 高斯基函数

$$\phi_j(x) = \exp\left\{-\frac{(x - \mu_j)^2}{2s^2}\right\}$$



● Sigmoid基函数

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right) \quad \sigma(a) = \frac{1}{1 + \exp(-a)}$$



问题求解

- 问题本质：确定模型中的参数 \mathbf{w}^T
- 基本思想：基于训练集最小化预测值 f 与真实输出值 y 的差异
- 定义目标函数(又叫代价函数Cost Function):

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (f(\mathbf{x}_i) - y_i)^2$$

- 进一步:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} J(\mathbf{w}) = \arg \min_{\mathbf{w}} \left[\frac{1}{2} \sum_{i=1}^N (f(\mathbf{x}_i) - y_i)^2 \right]$$

标准方程组 (Normal Equations)

- 目标函数：

$$\begin{aligned} J(\mathbf{w}) &= \sum_{i=1}^N (f(\mathbf{x}_i) - y_i)^2 \\ &= \sum_{i=1}^N (\mathbf{w}^T \mathbf{x}_i - y_i)^2 \end{aligned}$$

- 直接求导，并令其等于0，求得极值

标准方程组 (Normal Equations)

- 将目标函数写成矩阵形式：

$$\begin{aligned} J(\mathbf{w}) &= \sum_{i=1}^N (\mathbf{w}^T \mathbf{x}_i - y_i)^2 \\ &= (X\mathbf{w} - \mathbf{y})^T (X\mathbf{w} - \mathbf{y}) \end{aligned}$$

- 其中

$$X = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix} = \begin{pmatrix} x_{11}, x_{12}, \cdots, x_{1m} \\ x_{21}, x_{22}, \cdots, x_{2m} \\ \vdots & \ddots \\ x_{N1}, x_{N2}, \cdots, x_{Nm} \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}^T$$

标准方程组 (Normal Equations)

- 求导：

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) &= \frac{\partial}{\partial \mathbf{w}} (X\mathbf{w} - \mathbf{y})^T (X\mathbf{w} - \mathbf{y}) \\ &= 2X^T (X\mathbf{w} - \mathbf{y}) = 0\end{aligned}$$

$$\Rightarrow X^T X \mathbf{w} = X^T \mathbf{y}$$

- 得到模型的参数：

$$\hat{\mathbf{w}} = (X^T X)^{-1} X^T \mathbf{y}$$

梯度下降(Gradient Descent)

- 损失函数 $J(w)$

$$\min_w J(w)$$

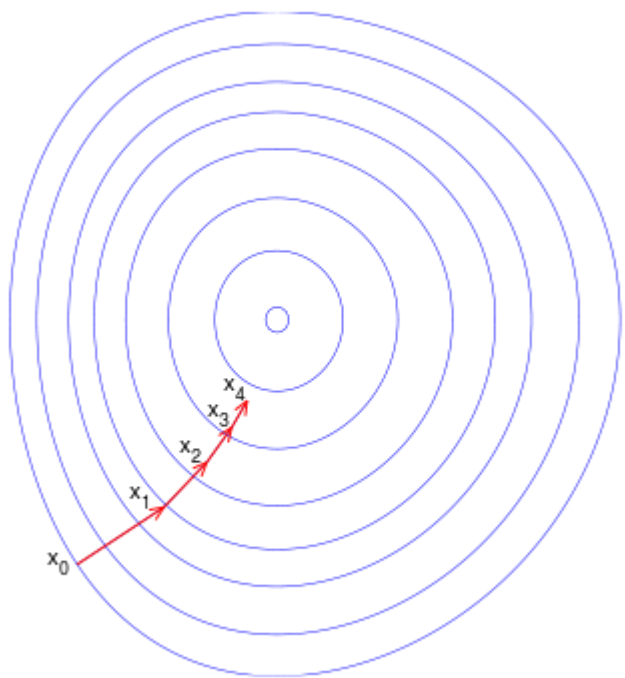
- 一般流程

- (1) 首先对 w 赋值，这个值可以是随机的，也可以是一个全零的向量。
- (2) 改变 w 的值，使得 $J(w)$ 按梯度下降的方向进行减少，直到收敛。

梯度下降(Gradient Descent)

- 梯度下降

梯度下降法是一个最优化算法，是求解无约束优化问题最简单和最基础的方法之一。



以负梯度方向为搜索方向
越接近目标值，步长越小，
前进越慢。

梯度下降(Gradient Descent)

- 梯度下降:

- 给定初始值 \mathbf{w}^0

- 更新 \mathbf{w} 使得 $J(\mathbf{w})$ 越来越小

$$w_j^t = w_j^{t-1} - \alpha \frac{\partial}{\partial w_j} J(\mathbf{w})$$

$$\frac{\partial}{\partial w_j} J(\mathbf{w}) = \sum_{i=1}^N (f(\mathbf{x}_i) - y_i) \cdot \mathbf{x}_{i,j}$$

- 同时更新 \mathbf{w} 的各维 $f(\mathbf{x}_i) = [\mathbf{w}^{t-1}]^T \mathbf{x}_i$

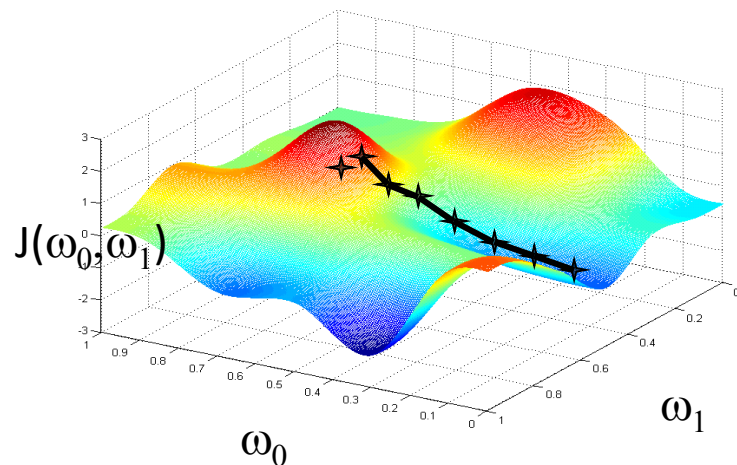
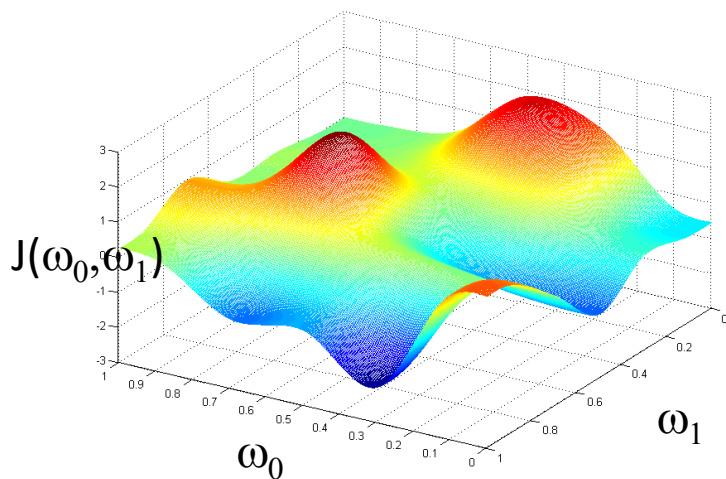
α 为学习率(Learning Rate)或更新步长

批处理梯度下降 (Batch Gradient Descent)

- 在梯度下降方法中，每次更新都利用所有数据，即：

$$w_j^t = w_j^{t-1} - \alpha \sum_{i=1}^N (f(\mathbf{x}_i) - y_i) \cdot x_j$$

在大样本条件下，批处理梯度下降的迭代速度很慢



随机梯度下降

(Stochastic Gradient Descent)

- 基本思想：如果条件对所有样本都成立，则对任一样本也成立

每次只用一个样本 (\mathbf{x}_r, y_r)

$$w_j^t = w_j^{t-1} - \alpha(f(\mathbf{x}_r) - y_r) \cdot \mathbf{x}_{r,j}$$

-收敛速度较快

-对大样本数据较有效

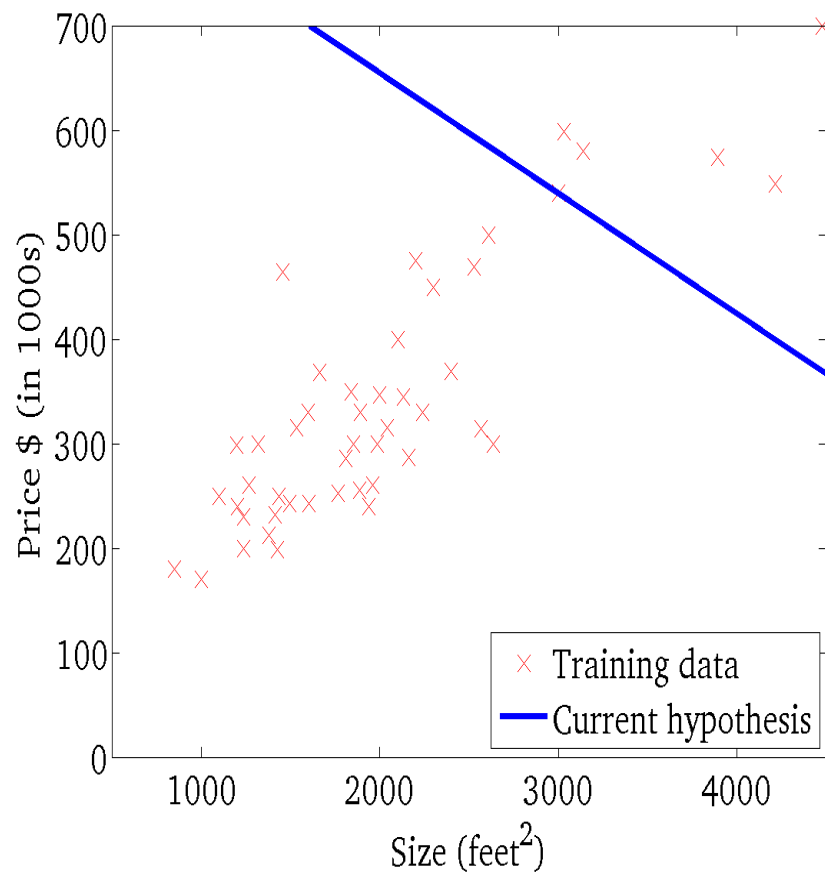
- 变化形式 又称在线学习(Online Learning)

可离线学习，每次“看”一个样本，对所有样本循环使用多次(一次循环称为一个Epoch)

每次可以看一些样本(Mini-Batch)

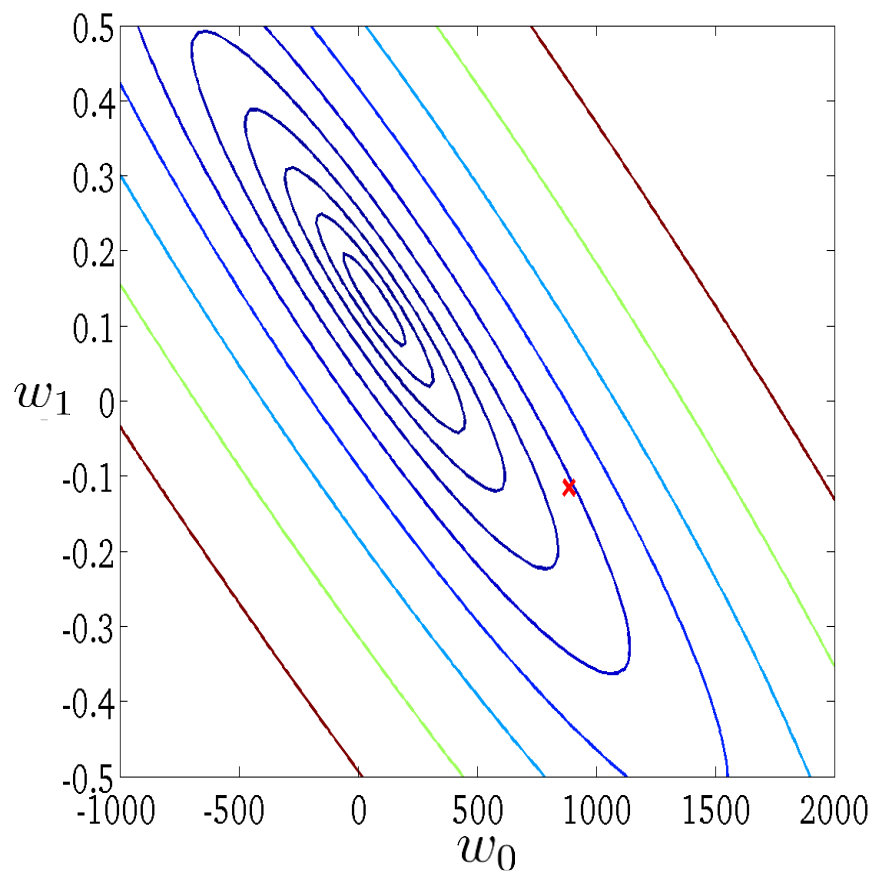
$$f_w(x)$$

(for fixed w_0, w_1 , this is a function of x)



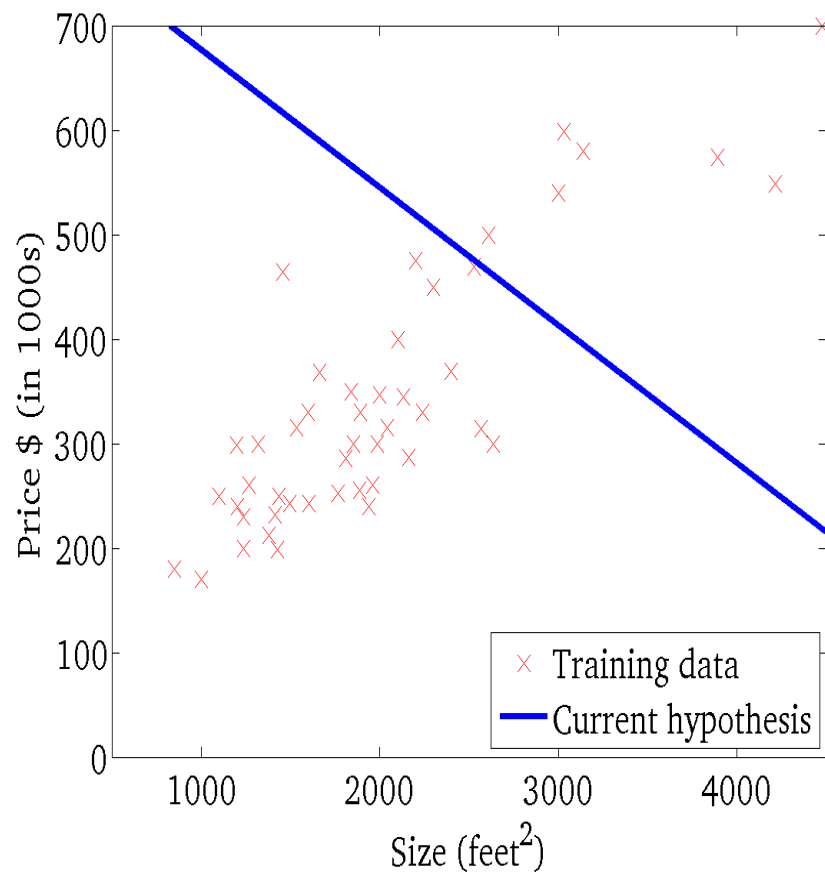
$$J(w)$$

(function of the parameters w_0, w_1)



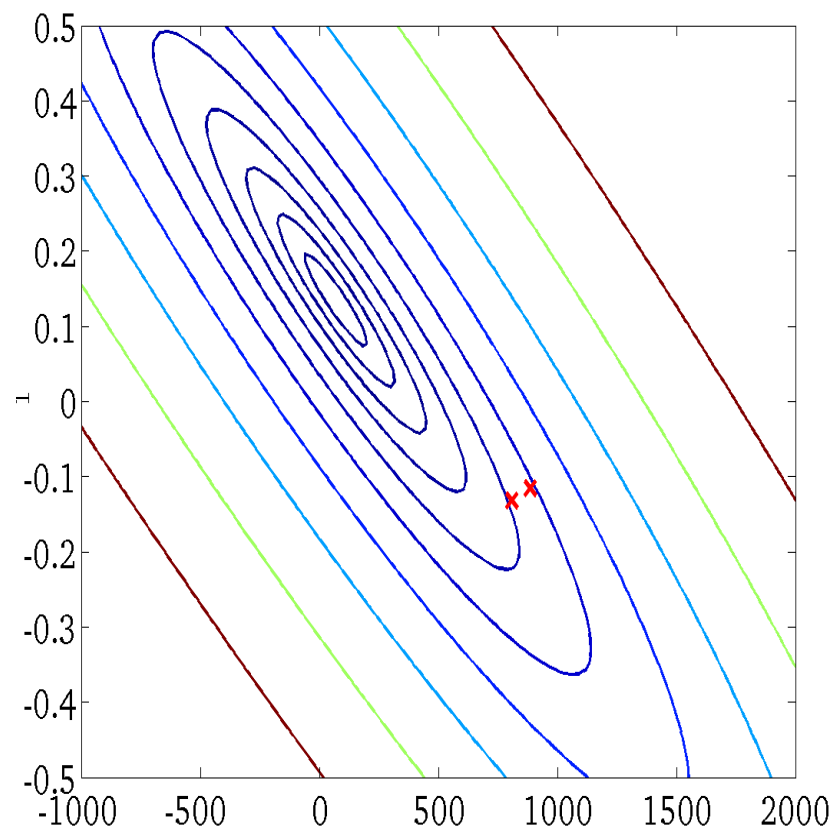
$$f_w(x)$$

(for fixed w_0, w_1 , this is a function of x)



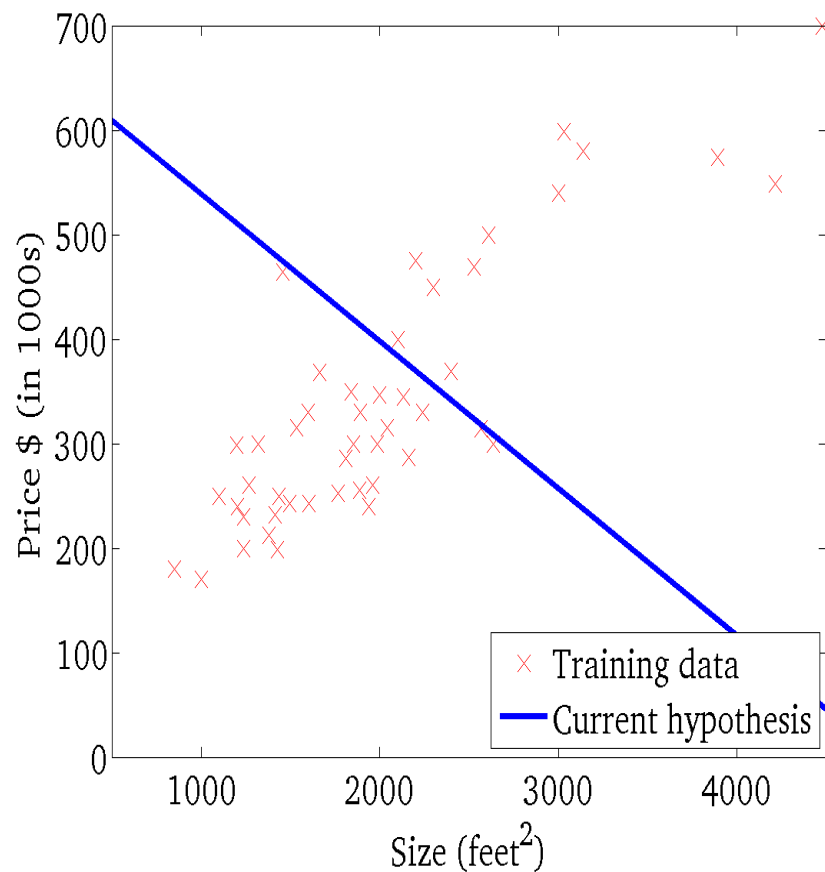
$$J(w)$$

(function of the parameters w_0, w_1)



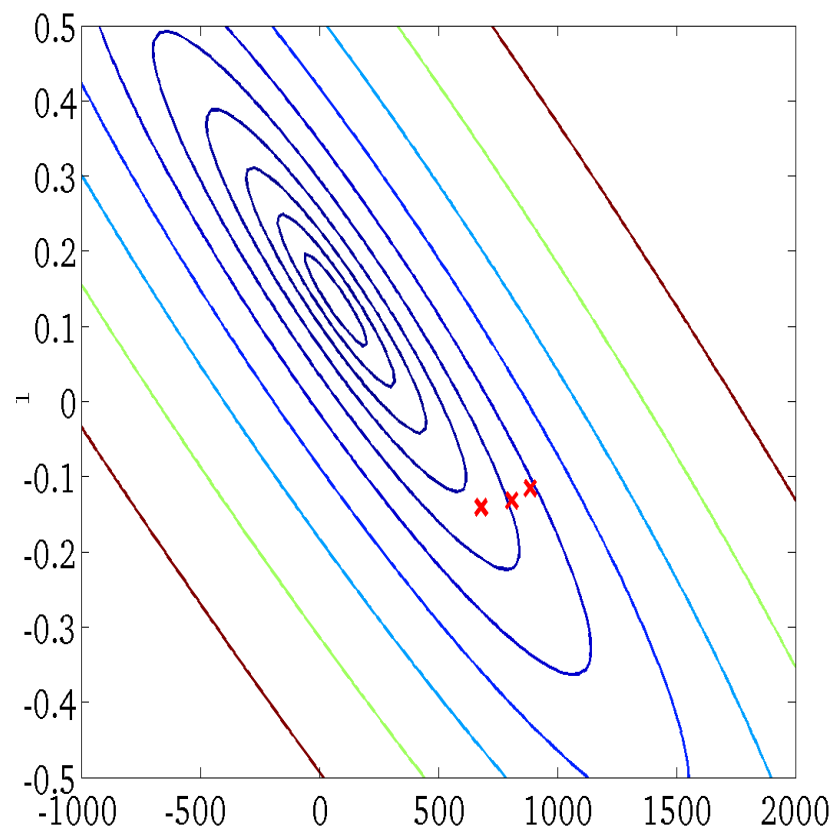
$$f_w(x)$$

(for fixed w_0, w_1 , this is a function of x)



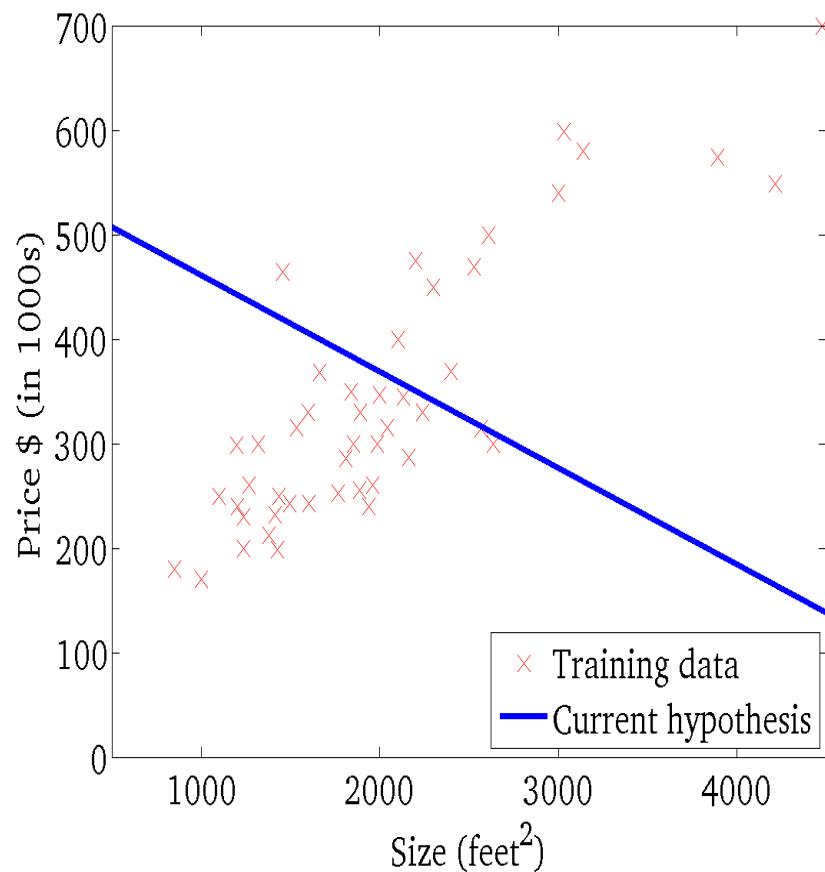
$$J(w)$$

(function of the parameters w_0, w_1)



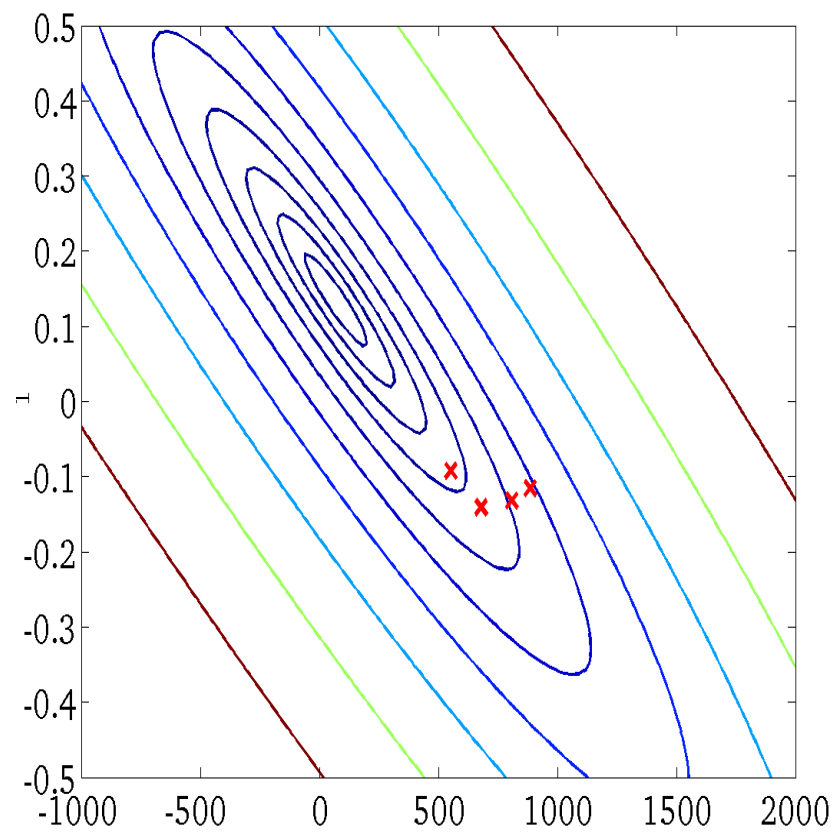
$$f_w(x)$$

(for fixed w_0, w_1 , this is a function of x)



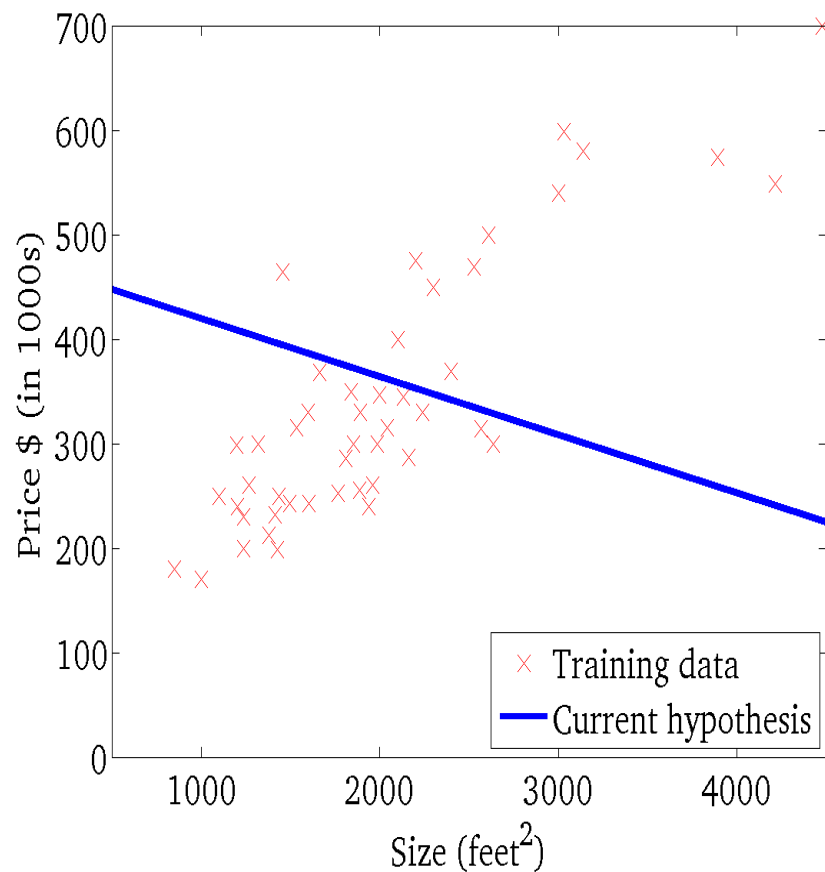
$$J(w)$$

(function of the parameters w_0, w_1)



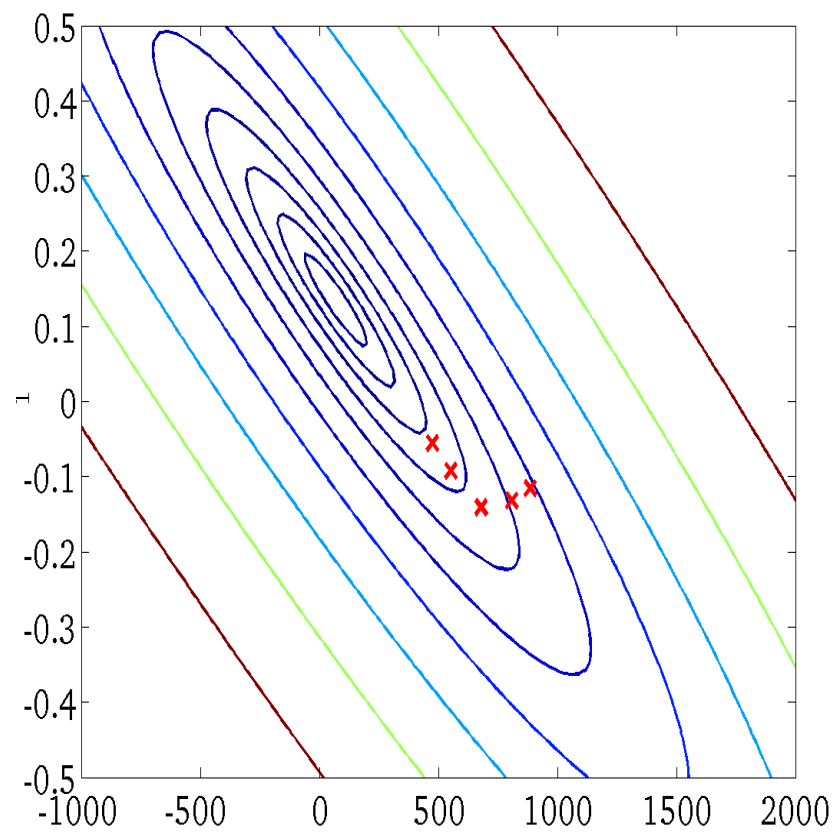
$$f_w(x)$$

(for fixed w_0, w_1 , this is a function of x)



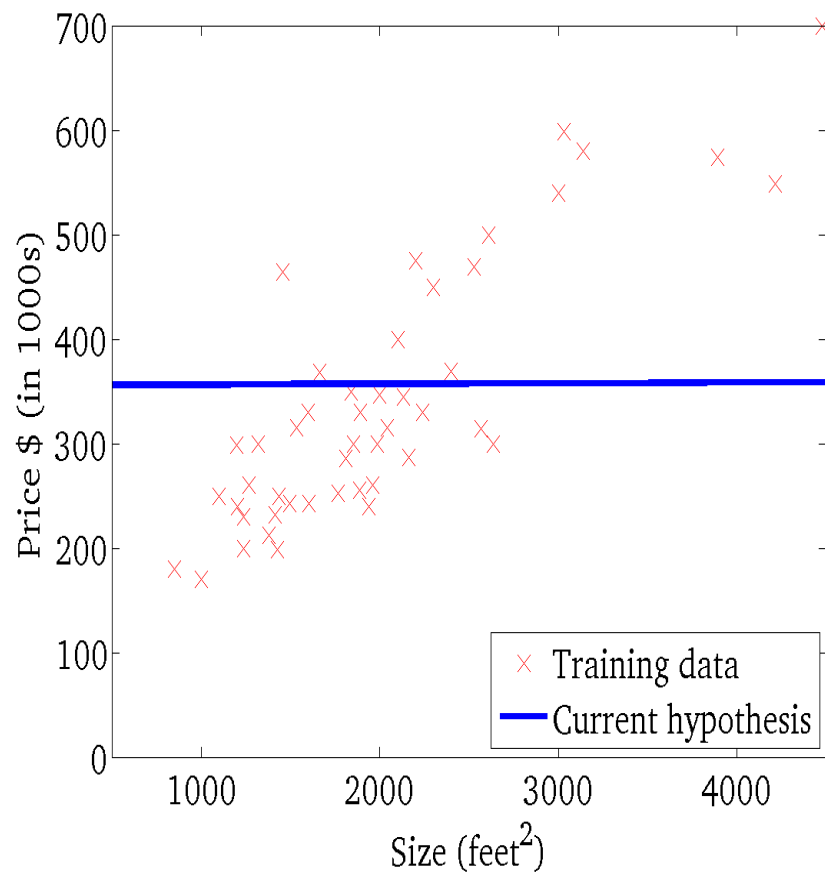
$$J(w)$$

(function of the parameters w_0, w_1)



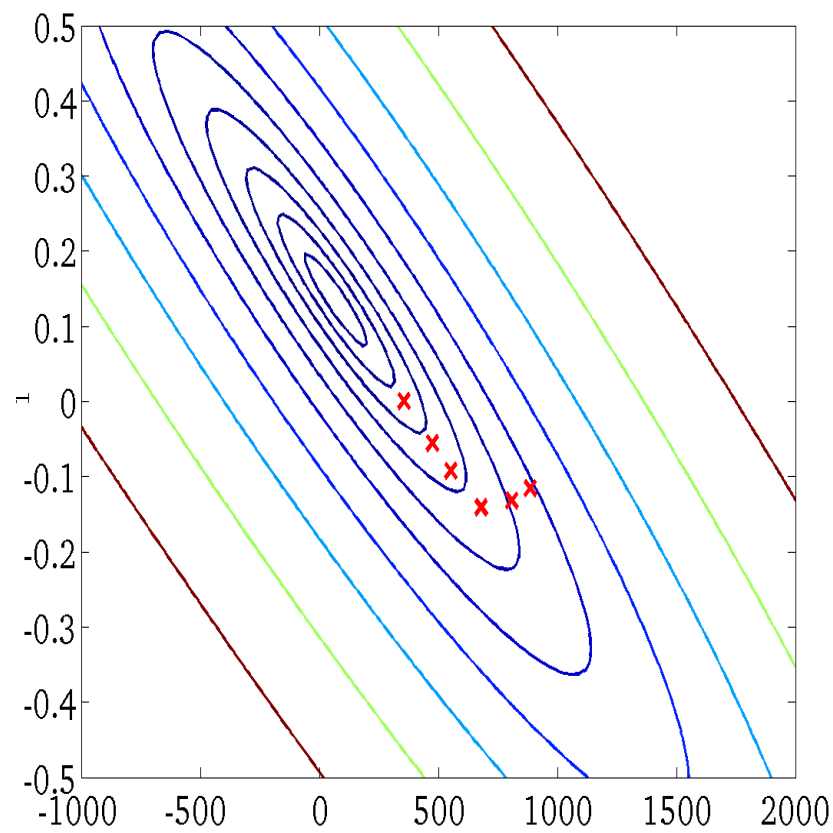
$$f_w(x)$$

(for fixed w_0, w_1 , this is a function of x)



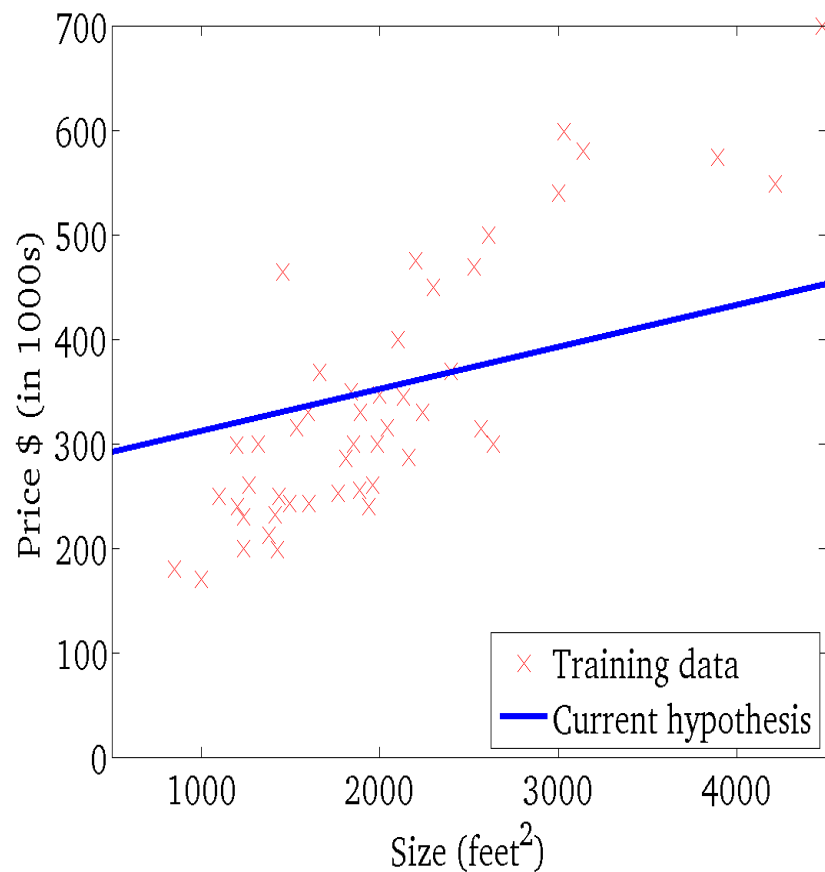
$$J(w)$$

(function of the parameters w_0, w_1)



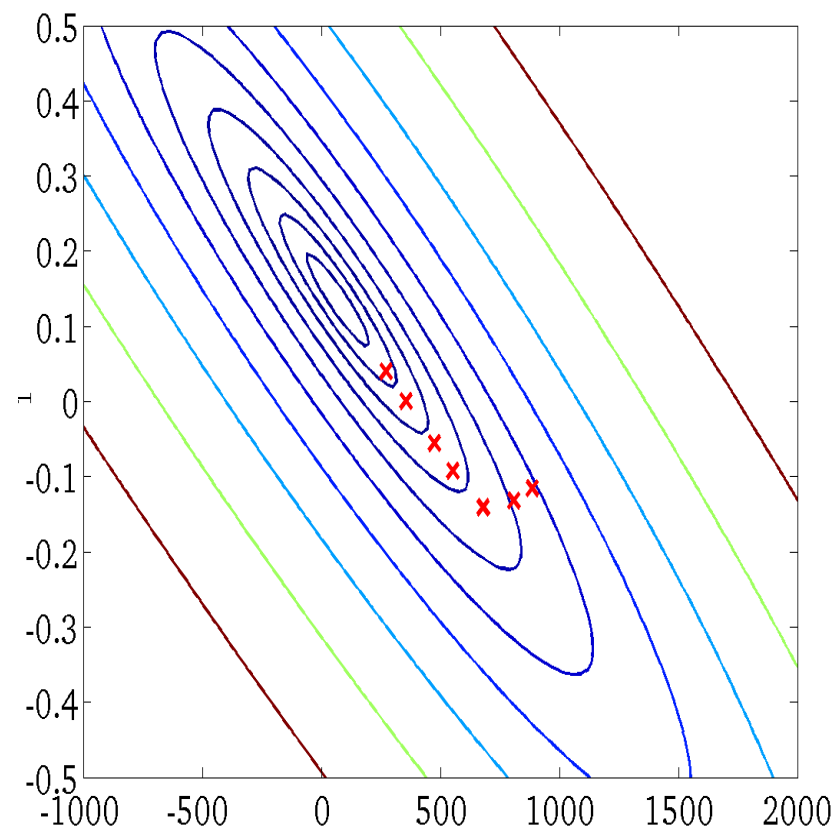
$$f_w(x)$$

(for fixed w_0, w_1 , this is a function of x)



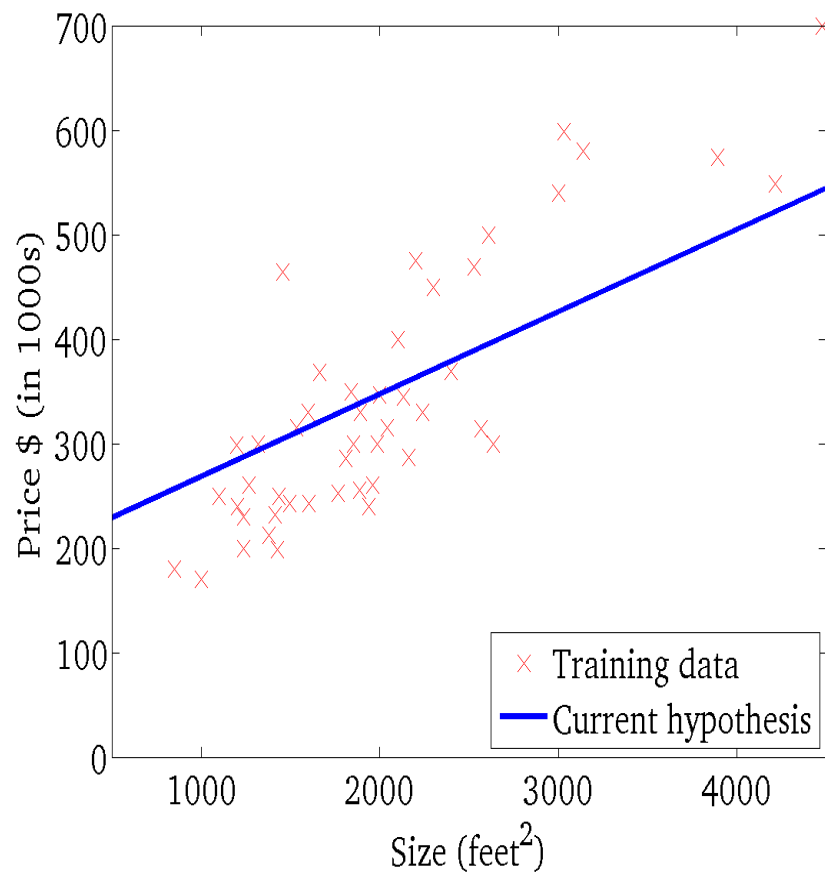
$$J(w)$$

(function of the parameters w_0, w_1)



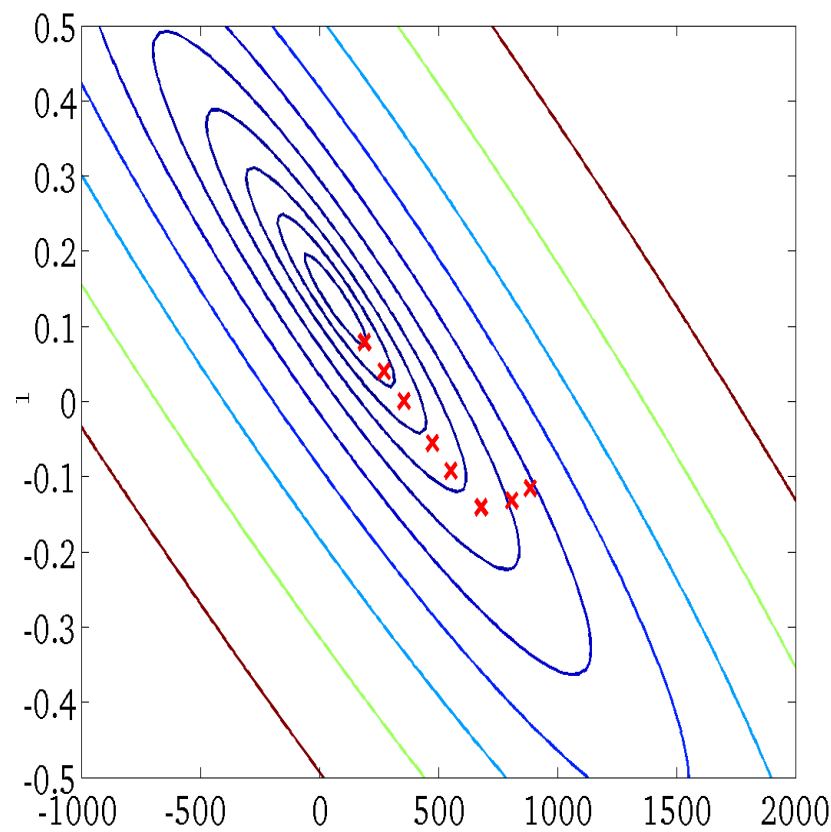
$$f_w(x)$$

(for fixed w_0, w_1 , this is a function of x)



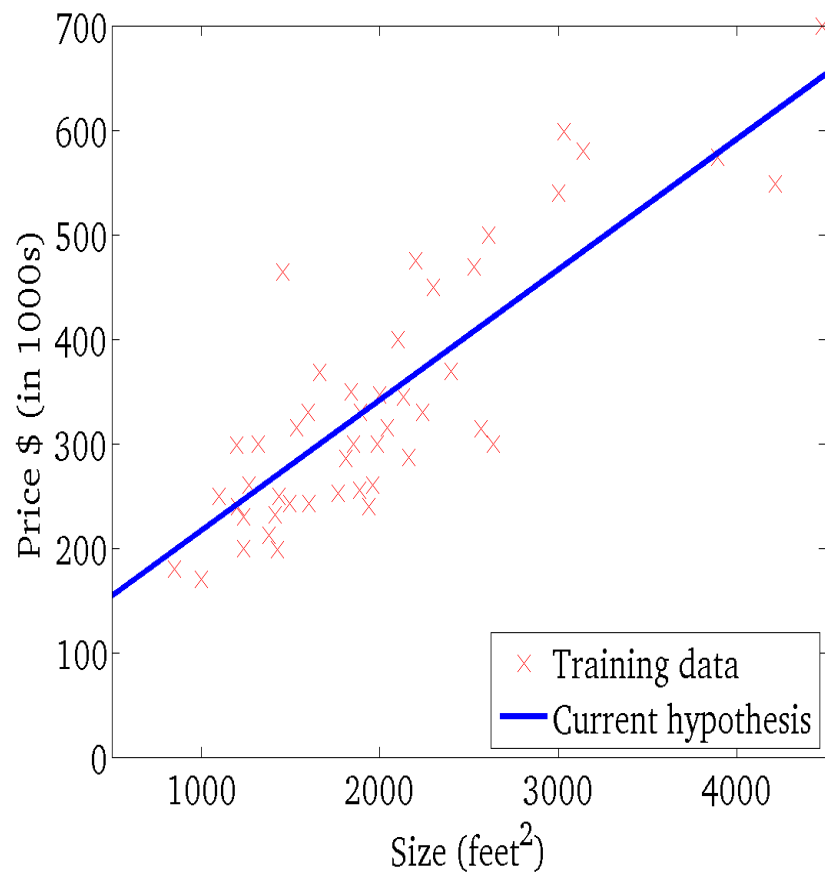
$$J(w)$$

(function of the parameters w_0, w_1)



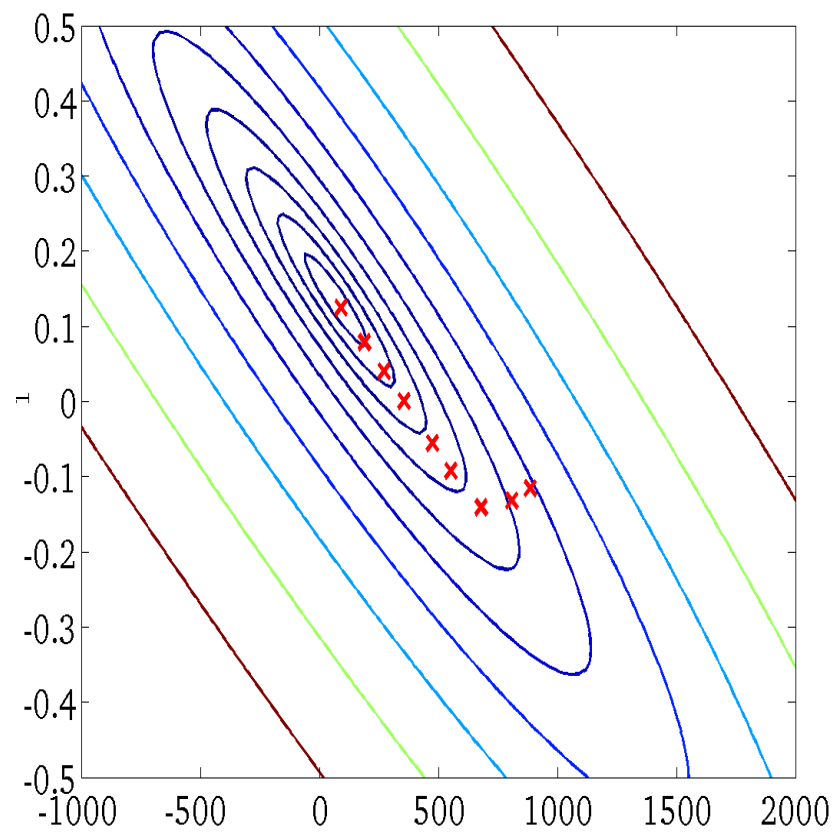
$$f_w(x)$$

(for fixed w_0, w_1 , this is a function of x)



$$J(w)$$

(function of the parameters w_0, w_1)



梯度下降 vs 标准方程组

● 梯度下降法：

- 需要选择 α
- 需要迭代多次
- 需要数据归一化
- 样本量非常大时也适用

● 标准方程组：

- 不需要选择 α
- 不需要迭代多次
- 无需数据归一化
- 样本量大时不适用
(需要计算 $(X^T X)^{-1}$)

样本量较小时选用标准方程组求解
样本量较大时选用梯度下降法求解

机器学习算法

机器学习主要问题

		<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Discrete</i>		Classification or Categorization	Clustering
	<i>Continuous</i>	Regression	Dimensionality Reduction

从贝叶斯分类说起

- 样本(Sample) $\mathbf{x} \in R^d$
- 状态(State) 第一类: $w = w_1$ 第二类: $w = w_2$
- 先验概率(A Priori Probability or Prior) $P(w_1)$ $P(w_2)$
- 样本分布密度(Sample Distribution Density) $p(\mathbf{x})$
- 类条件概率密度(Class-Conditional Probability Density)

$$p(\mathbf{x}|w_1) \quad p(\mathbf{x}|w_2)$$

从贝叶斯分类说起

- 后验概率(A Posteriori Probability or Posterior)

$$P(w_1|\mathbf{x}), P(w_2|\mathbf{x})$$

- 错误概率(Probability of Error)

$$P(e|\mathbf{x}) \begin{cases} P(w_2|\mathbf{x}) & \text{if } \mathbf{x} \text{ is assigned to } w_1 \\ P(w_1|\mathbf{x}) & \text{if } \mathbf{x} \text{ is assigned to } w_2 \end{cases}$$

- 平均错误率(Average Probability of Error)

$$P(e) = \int P(e|\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

- 正确率(Probability of Correctness) $P(c)$

基于最小错误率的决策

$$\min P(e) = \int P(e|x)p(x)dx$$

因为 $P(e|\mathbf{x}) \geq 0, p(x) \geq 0$

 $\min P(e|\mathbf{x})$ for all x

而 $P(e|\mathbf{x}) \begin{cases} P(w_2|\mathbf{x}), & \text{if } P(w_1|\mathbf{x}) > P(w_2|\mathbf{x}) \\ P(w_1|\mathbf{x}), & \text{if } P(w_2|\mathbf{x}) > P(w_1|\mathbf{x}) \end{cases}$

if $P(w_1|\mathbf{x}) \begin{matrix} > \\ < \end{matrix} P(w_2|\mathbf{x})$ assign $\begin{matrix} x \in w_1 \\ x \in w_2 \end{matrix}$

$$P(w|\mathbf{x}) = \max_{j=1,\dots,c} P(w_j|\mathbf{x})$$

基于最小风险的决策

条件期望损失：对于特定的 x 采取决策 α_i 的期望损失：

$$R(\alpha_i|x) = E[\lambda(\alpha_i, w_j)|x] = \sum_{j=1}^c \lambda(\alpha_i, w_j)P(w_j|x), \quad i = 1, 2, \dots, k$$

期望风险：对所有可能的 x 采取决策 $\alpha(x)$ 所造成的期望损失之和

$$R(\alpha) = \int R(\alpha(x)|x)p(x)dx$$

也称平均风险($R(\alpha)$ 表示 R 依赖于决策规则 $\alpha(\cdot)$)

对所有 x ，使 $R(\alpha(x)|x)$ 最小，则可以使 $R(\alpha)$ 最小

最小风险贝叶斯决策规则：

$$\text{if } R(\alpha_t|x) = \min_{j=1\dots k} R(\alpha_j|x), \text{ then } \alpha = \alpha_t$$

示例-基于最小错误率

- 假设在某个局部区域细胞中正常(w_1)和异常(w_2)两类的先验概率分别为：正常状态 $P(w_1)=0.9$ ，异常状态 $P(w_2)=0.1$ 。现有一待识别细胞，其观察值为 x ，从类条件概率密度曲线上查得 $p(x/w_1)=0.2$ ， $p(x/w_2)=0.4$ 。试对该细胞 x 进行分类。

解：利用贝叶斯公式计算 w_1 和 w_2 的后验概率

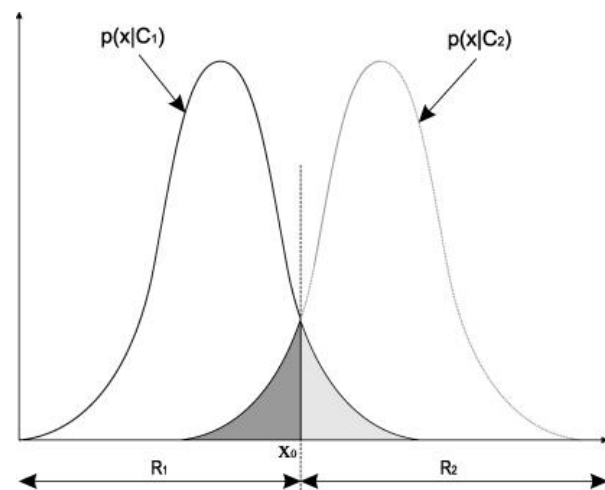
$$P(w_1|x) = \frac{p(x|w_1)P(w_1)}{\sum_{j=1}^2 p(x|w_j)P(w_j)} = \frac{0.2 \times 0.9}{0.2 \times 0.9 + 0.4 \times 0.1} = 0.818$$

$$P(w_2|x) = 1 - P(w_1|x) = 0.182$$

根据贝叶斯决策规则有

$$P(w_1|x) = 0.818 > P(w_2|x) = 0.182$$

把 x 归类于正常细胞。



示例-基于最小风险

● 决策表

决策 \ 损失 \ 状态	状态	
	w_1	w_2
α_1	0	6
α_2	1	0

解： $\lambda_{11}=0, \lambda_{12}=6, \lambda_{21}=1, \lambda_{22}=0$ $P(w_1|x) = 0.818$ $P(w_2|x) = 0.182$

计算条件风险 $R(\alpha_1|x) = \sum_{j=1}^2 \lambda_{1j}P(w_j|x) = \lambda_{12}P(w_2|x) = 1.092$

$$R(\alpha_2|x) = \lambda_{21}P(w_1|x) = 0.818$$

由于 $R(\alpha_1|x) > R(\alpha_2|x)$

把 x 归类于异常细胞。

小结

- 已知类条件概率密度 $p(\mathbf{x}|w_i)$ 和先验概率 $P(w_i)$ ，计算后验概率 $P(w_i|\mathbf{x})$ 进行决策

若类条件概率密度参数未知？

- 已知类条件概率密度 $p(\mathbf{x}|w_i)$ 的参数表达式，利用样本估计 $p(\mathbf{x}|w_i)$ 的未知参数，再利用贝叶斯定理将其转化成后验概率 $P(w_i|\mathbf{x})$ 进行决策

若类条件概率密度形式难以确定？

- 非参数方法估计

需要大量样本...

线性判别函数

- 利用样本集直接设计分类器

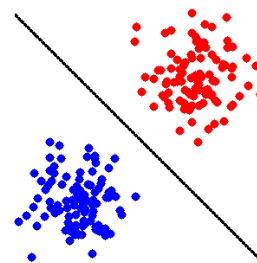
线性回归: $y(x) = w^T x + w_0$

利用样本估计 w , 对于给定 x , 计算 y 。

$$y(x) = f(w^T x + w_0) \quad f(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0 \end{cases}$$

$$w^T x + w_0 \geq 0 \quad \rightarrow \quad C_1$$

$$w^T x + w_0 \leq 0 \quad \rightarrow \quad C_2$$



将分类器设计问题转化为求准则函数极值的问题
准则函数: 分类器设计的某些要求的函数形式

线性判别函数

- 两类情况下线性判别函数 $g(x) = w^T x + w_0$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \text{ 特征向量/样本向量; } w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} \text{ 权向量; } w_0 \text{ 阈值权 (常数)}$$

$$\text{如果} \begin{cases} g(x) > 0, x \in w_1 \\ g(x) < 0, x \in w_2 \\ g(x) = 0, \text{可将} x \text{分到任意一类或拒绝} \end{cases}$$

$g(x)=0$ 定义了一个决策面，当 $g(x)$ 为线性函数时，决策面就是超平面。

线性判别函数

如果 x_1 和 x_2 都在决策面 H 上, 则有

$$w^T x_1 + w_0 = w^T x_2 + w_0$$

即

$$w^T (x_1 - x_2) = 0$$

说明 w 和超平面 H 上任一向量正交, 即 w 是 H 的法向量。

判别函数 $g(x)$ 可以看成是特征空间中某点 x 到超平面 H 距离的一种代数度量

$$x = x_p + r \frac{w}{\|w\|}$$

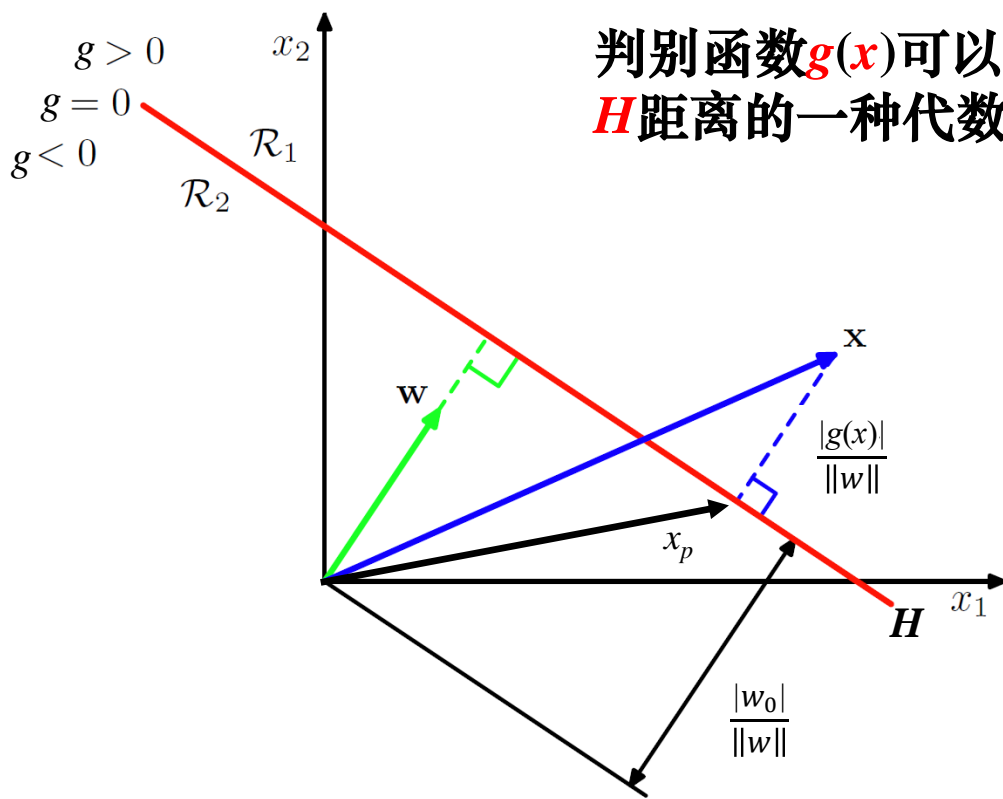
x_p : x 在 H 上的投影向量

r : x 到 H 上的垂直距离

$\frac{w}{\|w\|}$: 是 w 方向上的单位向量

$$\begin{aligned} g(x) &= w^T \left(x_p + r \frac{w}{\|w\|} \right) + w_0 \\ &= w^T x_p + r \frac{w^T w}{\|w\|} + w_0 = r \|w\| \end{aligned}$$

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



线性判别函数

若 x 为原点，则 $g(x) = w_0$

从原点到超平面 H 的距离 $r_0 = \frac{w_0}{\|w\|}$

如果 $w_0 > 0$ ，则原点在 H 的正侧

如果 $w_0 < 0$ ，则原点在 H 的负侧

如果 $w_0 = 0$ ，则 $g(x)$ 具有齐次形式，超平面 H 通过原点

超平面 H 的方向由**权向量** w 决定；其位置由**阈值权** w_0 确定。

判别函数 $g(x)$ 正比于 x 点到超平面的代数距离

当 x 在 H 的正侧时， $g(x) > 0$

当 x 在 H 的负侧时， $g(x) < 0$

广义线性判别函数

- 两类问题, X 是一维样本空间

若 $x < a$ 或 $x > b$, $x \in w_1$

若 $a < x < b$, $x \in w_2$

如果建立 $g(x) = (x - a)(x - b)$

决策规则 $\begin{cases} g(x) > 0, x \in w_1 \\ g(x) < 0, x \in w_2 \end{cases}$

一般形式 $g(x) = c_0 + c_1x + c_2x^2$

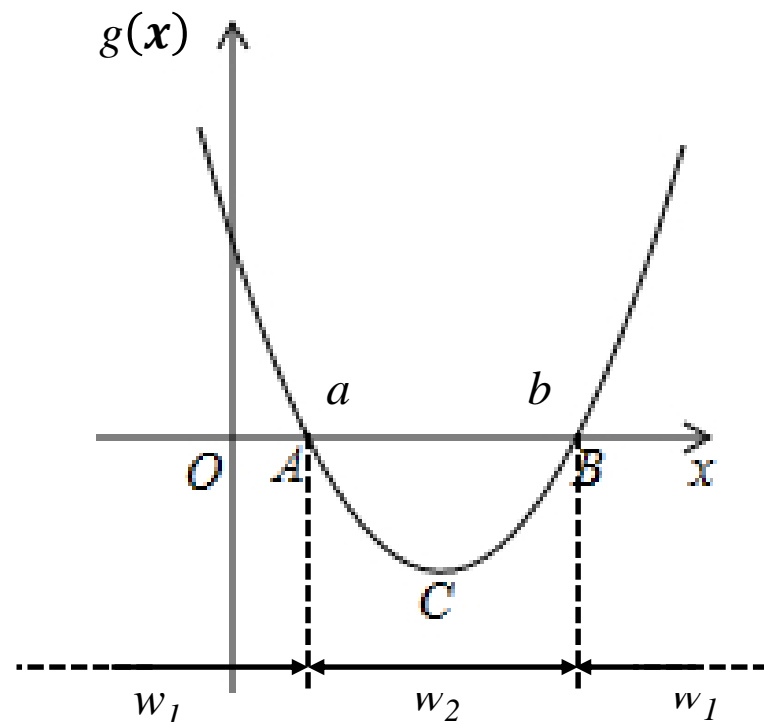
可转化为 $g(x) = a^T y = \sum_{i=1}^3 a_i y_i$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix} \quad a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}$$

称 $g(x) = a^T y$ 广义线性判别函数

a 广义权向量

不适用于非凸和多连通区域划分



利用线性函数的简单性解决复杂问题

维数大大增加 \rightarrow “维数灾难”

线性判别函数齐次化

- 线性判别函数 $g(x) = w^T x + w_0$

改写成 $g(x) = w_0 + \sum_{i=1}^d w_i x_i = \sum_{i=1}^d a_i y_i = a^T y$

称为**线性判别函数的齐次简化**

$$y = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} 1 \\ x \end{bmatrix} \text{增广样本向量}; \quad a = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} w_0 \\ w \end{bmatrix} \text{增广权向量}$$

$\hat{d} = d + 1$; **y**比**x**增加一维, 保持样本空间欧式距离不变, 变换后的样本仍全部位于**d**维子空间(原**X**空间)中。

方程 $a^T y = 0$ 在**Y**空间确定了一个通过原点的超平面 \hat{H} , 它对**d**维子空间的划分与原决策面 $w^T x + w_0 = 0$ 对原**X**空间的划分完全相同。**Y**空间中任意一点**y**到 \hat{H} 的距离:

$$r = \frac{g(x)}{\|a\|} = \frac{a^T y}{\|a\|}$$

线性分类器设计

- 利用训练样本建立线性判别函数

$$g(x) = w^T x + w_0$$

$$g(x) = w_0 + \sum_{i=1}^d w_i x_i = \sum_{i=1}^d a_i y_i = a^T y$$

最好的结果一般出现在准则函数的极值点上，所以将分类器设计问题转化为求准则函数极值 w^* , w_0^* 或 a^* 的问题。

步骤1： 具有类别标志的样本集 $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ 或其增广样本集 \mathcal{Y} 。

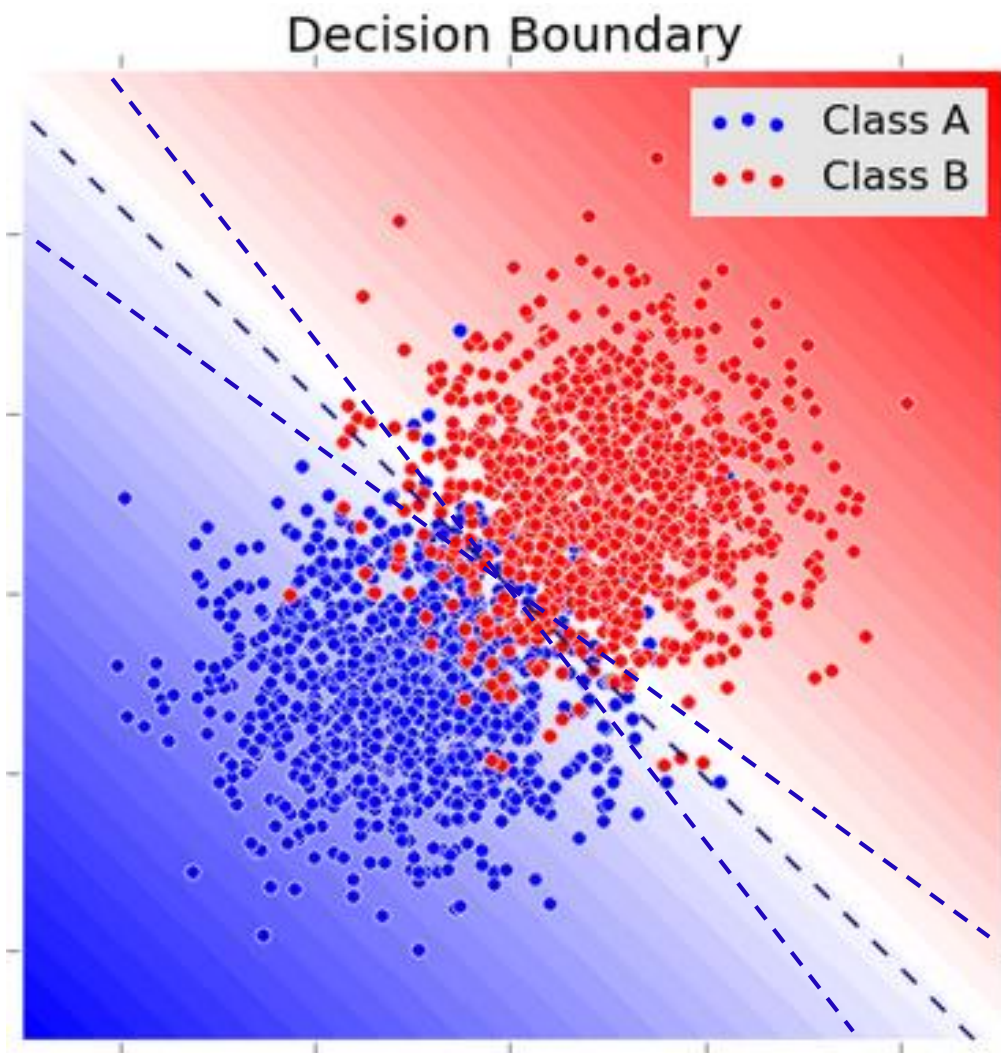
步骤2： 确定准则函数 \mathcal{J} ，满足① \mathcal{J} 是样本集和 w , w_0 或 a 的函数；② \mathcal{J} 的值反应分类器的性能，其极值对应“最好”的决策。

步骤3： 优化求解准则函数极值 w^* , w_0^* 或 a^* 。

最终得到线性判别函数： $g(x) = w^{*T} x + w_0^*$ 或 $g(x) = a^{*T} y$ ，对于未知类别样本 x_k ，计算 $g(x_k)$ 并通过决策规则判断其类别。

准则函数

- Fisher准则
- 感知机准则
- 最小平方误差准则
- 最小错分样本数准则
- ...

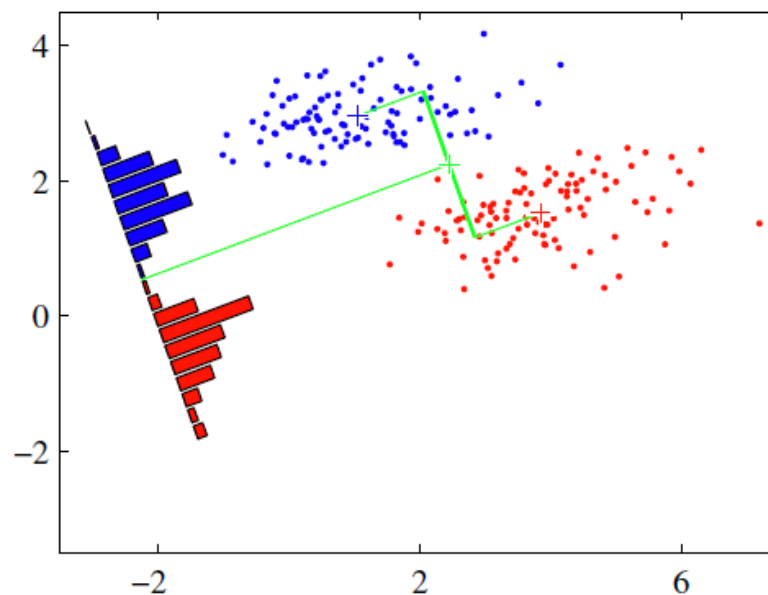
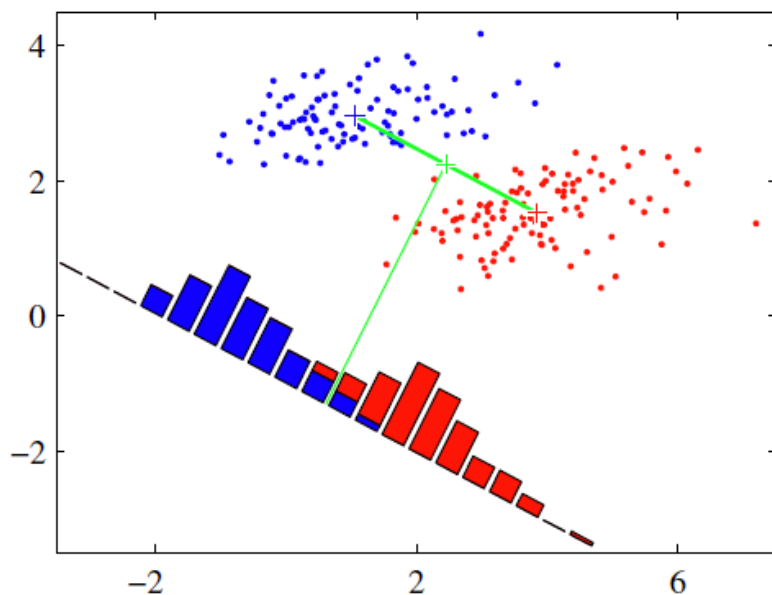


Fisher准则

- R. A. Fisher (1936年论文)

考虑把 d 维空间的样本投影到一条直线上形成一维空间。在一般情况下总可以找到某个方向，使样本在这个方向的直线上的投影分开得最好。

Fisher准则就是要解决
如何根据实际情况找到这条最好的、最易于分类的投影线的问题



Fisher准则

- 寻找最好投影方向 w^*

以二分类问题为例， d 维样本 x_1, x_2, \dots, x_N ，其中 N_1 个属于 w_1 类记为子集 X_1 ， N_2 个属于 w_2 类记为子集 X_2 。

在 d 维 X 空间

(1) 各类样本的均值向量 m_i $m_i = \frac{1}{N_i} \sum_{x \in X_i} x, i = 1, 2$

(2) 样本类内离散度矩阵 S_i 和总类内离散度矩阵 S_w

$$S_i = \sum_{x \in X_i} (x - m_i)(x - m_i)^T, i = 1, 2$$
$$S_w = S_1 + S_2$$

(3) 样本类间离散度矩阵 S_b

$$S_w = P(w_1)S_1 + P(w_2)S_2$$

$$S_b = (m_1 - m_2)(m_1 - m_2)^T$$

$$S_b = P(w_1)P(w_2)(m_1 - m_2)(m_1 - m_2)^T$$

Fisher准则

- 寻找最好投影方向 w^*

以二分类问题为例， d 维样本 x_1, x_2, \dots, x_N ，其中 N_1 个属于 w_1 类记为子集 X_1 ， N_2 个属于 w_2 类记为子集 X_2 。

在一维 Y 空间 $y_n = w^T x_n$

(1) 各类样本均值 \tilde{m}_i $\tilde{m}_i = \frac{1}{N_i} \sum_{y \in \eta_i} y, i = 1, 2$

(2) 样本类内离散度 \tilde{S}_i^2 和总类内离散度 \tilde{S}_w

$$\tilde{S}_i^2 = \sum_{y \in \eta_i} (y - \tilde{m}_i)^2, i = 1, 2$$

$$\tilde{S}_w = \tilde{S}_1^2 + \tilde{S}_2^2$$

Fisher准则

- 希望投影后在一维Y空间中各类样本尽可能分开，即两类均值之差 $(\tilde{m}_1 - \tilde{m}_2)$ 越大越好；同时希望各类样本内部尽量密集，即类内离散度越小越好。



Fisher准则函数: $J_F(w) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{S}_1^2 + \tilde{S}_2^2}$



Fisher准则

● 求 $J_F(w)$ 取得最大值的 w $J_F(w) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{S}_1^2 + \tilde{S}_2^2}$

$$\tilde{m}_i = \frac{1}{N_i} \sum_{y \in \eta_i} y = \frac{1}{N_i} \sum_{x \in \chi_i} w^T x = w^T \left(\frac{1}{N_i} \sum_{x \in \chi_i} x \right) = w^T m_i$$

$$(\tilde{m}_1 - \tilde{m}_2)^2 = (w^T m_1 - w^T m_2)^2 = w^T (m_1 - m_2)(m_1 - m_2)^T w = w^T S_b w$$

$$\tilde{S}_i^2 = \sum_{y \in \eta_i} (y - \tilde{m}_i)^2 = \sum_{x \in \chi_i} (w^T x - w^T m_i)^2 = w^T \left[\sum_{x \in \chi_i} (x - m_i)(x - m_i)^T \right] w = w^T S_i w$$

$$\tilde{S}_1^2 + \tilde{S}_2^2 = w^T (S_1 + S_2) w = w^T S_w w \quad \rightarrow \quad J_F(w) = \frac{w^T S_b w}{w^T S_w w}$$

Lagrange乘子法求解:

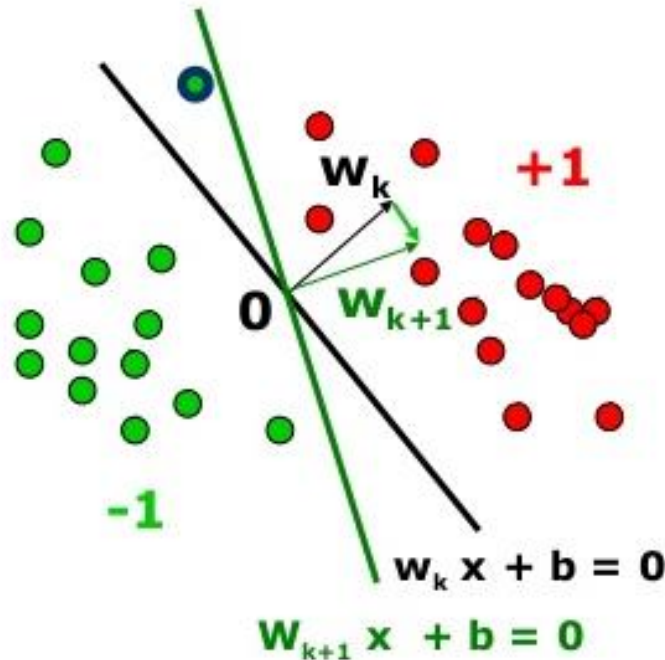
$$w^* = S_w^{-1} (m_1 - m_2)$$

分类时确定分界阈值 y_0 , 与 $y = w^{*T} x$ 比较进行决策。

感知机准则

- F. Rosenblatt(1950-1960年提出)

感知准则是一种自学习判别函数生成方法，由于Rosenblatt试图将其用于脑模型**感知器**，因此得名。该方法对随意给定的判别函数初始值，通过样本分类训练过程逐步对其修正直至最终确定。



几个基本概念

● 线性可分性

一组容量为 N 的样本集 y_1, y_2, \dots, y_N , 其中 y_n 为 \hat{d} 维增广样本向量, 分别来自 w_1 类和 w_2 类, 如果存在权向量 a , 使得对于任何 $y \in w_1$, 都有 $a^T y > 0$, 而对于任何 $y \in w_2$, 都有 $a^T y < 0$, 则称这组样本为线性可分的, 反之亦然成立。

● 样本的规范化

$$\begin{cases} a^T y_i > 0, y_i \in w_1 \\ a^T y_j < 0, y_j \in w_2 \end{cases}$$



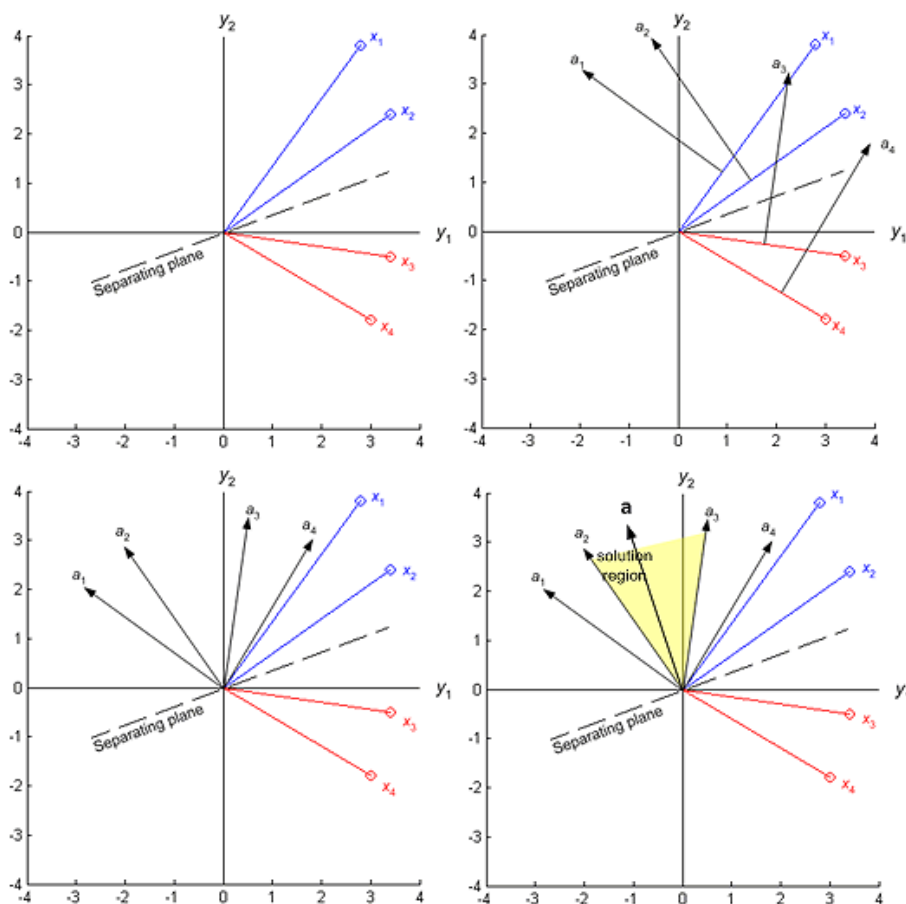
$$y'_n = \begin{cases} y_i > 0, y_i \in w_1 \\ -y_j < 0, y_j \in w_2 \end{cases} \quad \text{规范化增广样本向量}$$



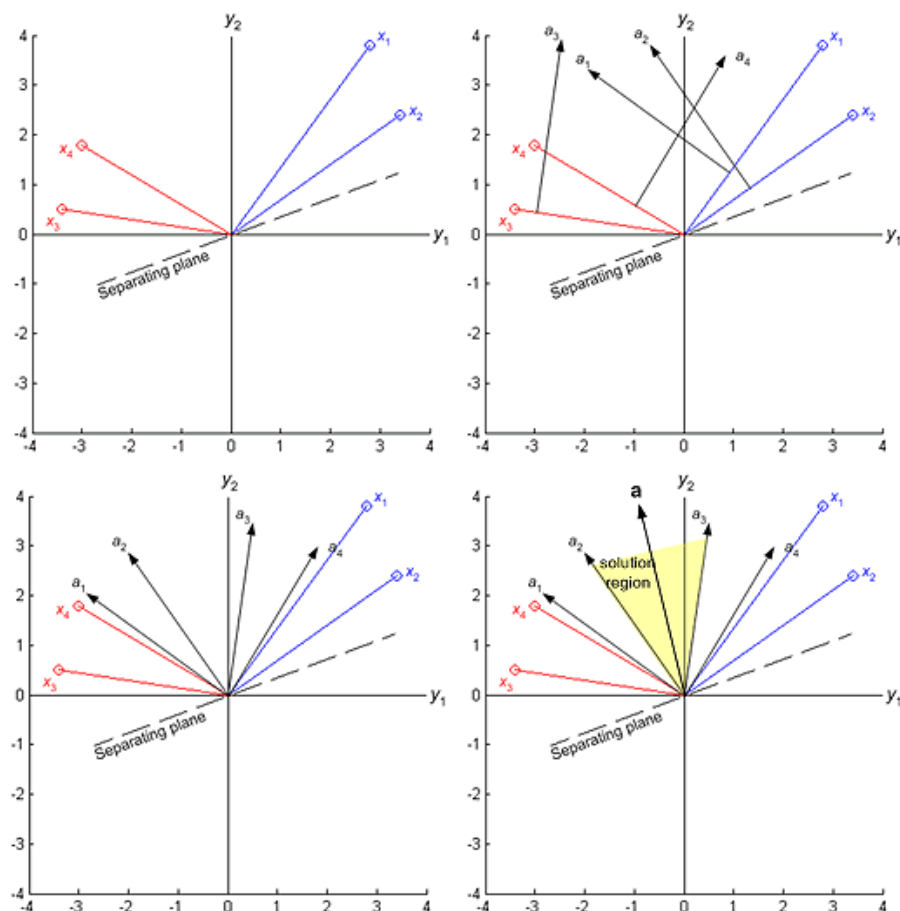
$$a^T y'_n > 0, n = 1, 2, \dots, N$$

几个基本概念

● 解向量和解区



未规范化

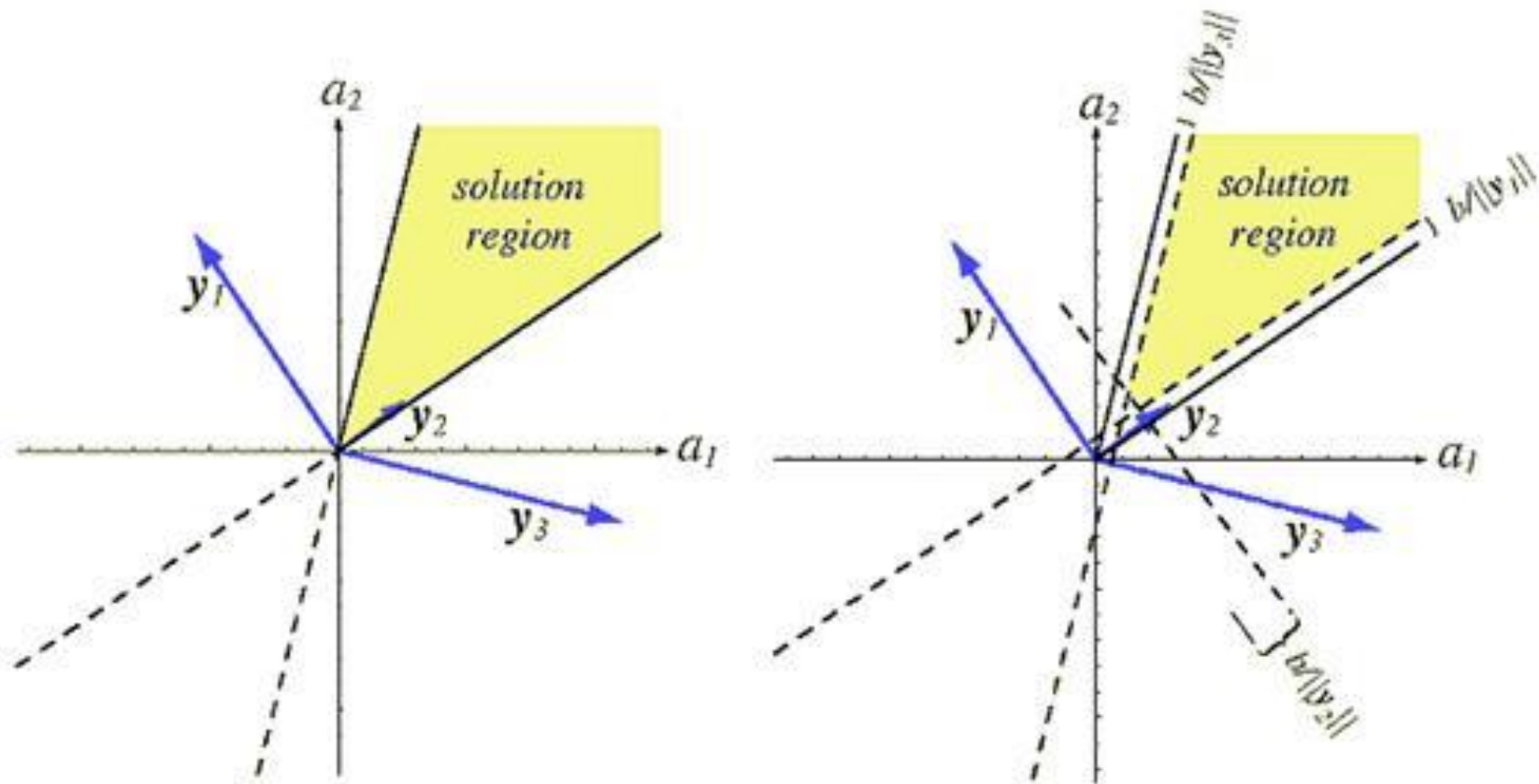


规范化

几个基本概念

● 对解区的限制

使解向量更可靠 $a^T y_n \geq b > 0$ ，避免解收敛到解区边界的某点上



感知机准则

● 寻找解向量 a^*

对于一组样本 y_1, y_2, \dots, y_N , 其中 y_n 是规范化增广样本向量, 使得:

$$a^T y_n > 0, n = 1, 2, \dots, N$$

对于线性可分问题, 构造准则函数 $J_P(a) = \sum_{y \in \eta^k} (-a^T y)$

其中 η^k 是被权向量 a 错分的样本集合, 即当 y 被错分时, 就有 $a^T y_n \leq 0$

因此 $J_P(a) \geq 0$, 仅当 a 为解向量或在解区边界时 $J_P(a) = 0$

也就是说, 当且仅当 η^k 为空集时 $J_P^*(a) = \min J_P(a) = 0$

此时无错分样本, 这时的 a 就是解向量 a^* 。

感知机准则

- 求使 $J_P(a)$ 达到最小值的 a^*

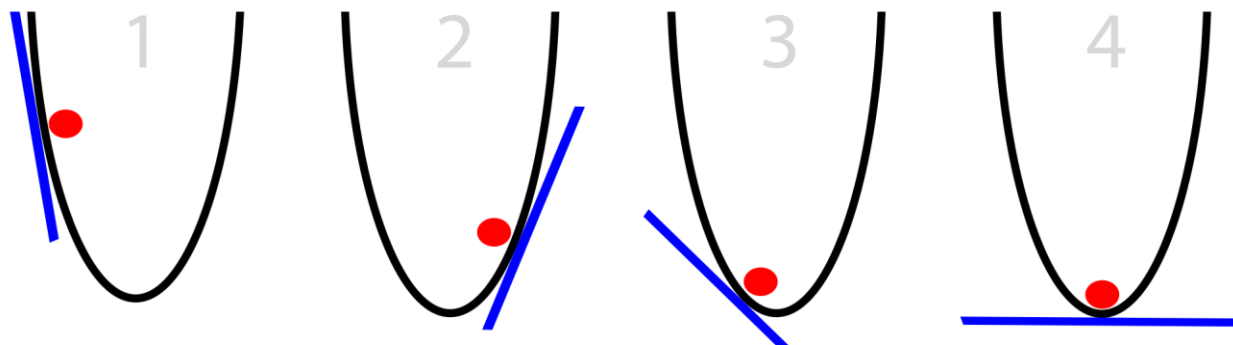
采用梯度下降法求解 $J_P(a) = \sum_{y \in \eta^k} (-a^T y)$

$$\nabla J_P(a) = \frac{\partial J_P(a)}{\partial a} = \sum_{y \in \eta^k} (-y)$$

$$a(k+1) = a(k) - \rho_k \nabla J$$

梯度下降法迭代公式

$$a(k+1) = a(k) + \rho_k \nabla \sum_{y \in \eta^k} y$$



感知机准则-示例

□ Sample set for two-class case

■ Class 1

$$\mathbf{x}_1 = \begin{pmatrix} -2 \\ 2 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} -2 \\ -2 \end{pmatrix} \quad \mathbf{y}_1 = \begin{pmatrix} 1 \\ -2 \\ 2 \end{pmatrix}, \mathbf{y}_2 = \begin{pmatrix} 1 \\ -2 \\ -2 \end{pmatrix}$$

■ Class 2

$$\mathbf{x}_3 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \quad \mathbf{y}_3 = \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix}, \mathbf{y}_4 = \begin{pmatrix} -1 \\ -2 \\ 1 \end{pmatrix}$$

■ Initial weight vector

$$\mathbf{a}(1) = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} \quad \mathbf{a}(1)^t = (0 \quad 2 \quad 1)$$

感知机准则-示例

□ Example (Cont.)

■ Iterative procedure

$$\square(1) \quad \mathbf{y}^{(1)t} = (1 \quad -2 \quad 2)$$

$$\mathbf{a}(1)^t \mathbf{y}^{(1)} = (0 \quad 2 \quad 1) \begin{pmatrix} 1 \\ -2 \\ 2 \end{pmatrix} = -2 < 0$$

$$\mathbf{a}(2)^t = (0 \quad 2 \quad 1) + (1 \quad -2 \quad 2) = (1 \quad 0 \quad 3)$$

感知机准则-示例

□ Example (Cont.)

■ Iterative procedure

$$\square(2) \quad \mathbf{y}^{(2)t} = (1 \quad -2 \quad -2)$$

$$\mathbf{a}(2)^t \mathbf{y}^{(2)} = (1 \quad 0 \quad 3) \begin{pmatrix} 1 \\ -2 \\ -2 \end{pmatrix} = -5 < 0$$

$$\mathbf{a}(3)^t = (1 \quad 0 \quad 3) + (1 \quad -2 \quad -2) = (2 \quad -2 \quad 1)$$

感知机准则-示例

□ Example (Cont.)

■ Iterative procedure

$$\square(3) \quad \mathbf{y}^{(3)t} = (-1 \quad -2 \quad -1)$$

$$\mathbf{a}(3)^t \mathbf{y}^{(3)} = (2 \quad -2 \quad 1) \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix} = 1 > 0$$

$$\mathbf{a}(4)^t = (2 \quad -2 \quad 1) \text{ (no change)}$$

感知机准则-示例

□ Example (Cont.)

■ Iterative procedure

$$\square (4) \quad \mathbf{y}^{(4)f} = (-1 \quad -2 \quad 1)$$

$$\mathbf{a}(4)^t \mathbf{y}^{(3)} = (2 \quad -2 \quad 1) \begin{pmatrix} -1 \\ -2 \\ 1 \end{pmatrix} = 3 > 0$$

$$\mathbf{a}(5)^t = (2 \quad -2 \quad 1) \quad (\text{no change})$$

感知机准则-示例

□ Example (Cont.)

■ Iterative procedure

$$\square(5) \quad \mathbf{y}^{(5)t} = (1 \quad -2 \quad 2)$$

$$\mathbf{a}(5)^t \mathbf{y}^{(1)} = (2 \quad -2 \quad 1) \begin{pmatrix} 1 \\ -2 \\ 2 \end{pmatrix} = 8 > 0$$

$$\mathbf{a}(6)^t = (2 \quad -2 \quad 1) \quad (\text{no change})$$

感知机准则-示例

□ Example (Cont.)

■ Iterative procedure

$$\square(6) \quad \mathbf{y}^{(6)t} = (1 \quad -2 \quad -2)$$

$$\mathbf{a}(6)^t \mathbf{y}^{(2)} = (2 \quad -2 \quad 1) \begin{pmatrix} 1 \\ -2 \\ -2 \end{pmatrix} = 4 > 0$$

$$\mathbf{a}(7)^t = (2 \quad -2 \quad 1)$$

感知机准则-示例

□ Example (Cont.)

■ Iterative procedure

□(7)

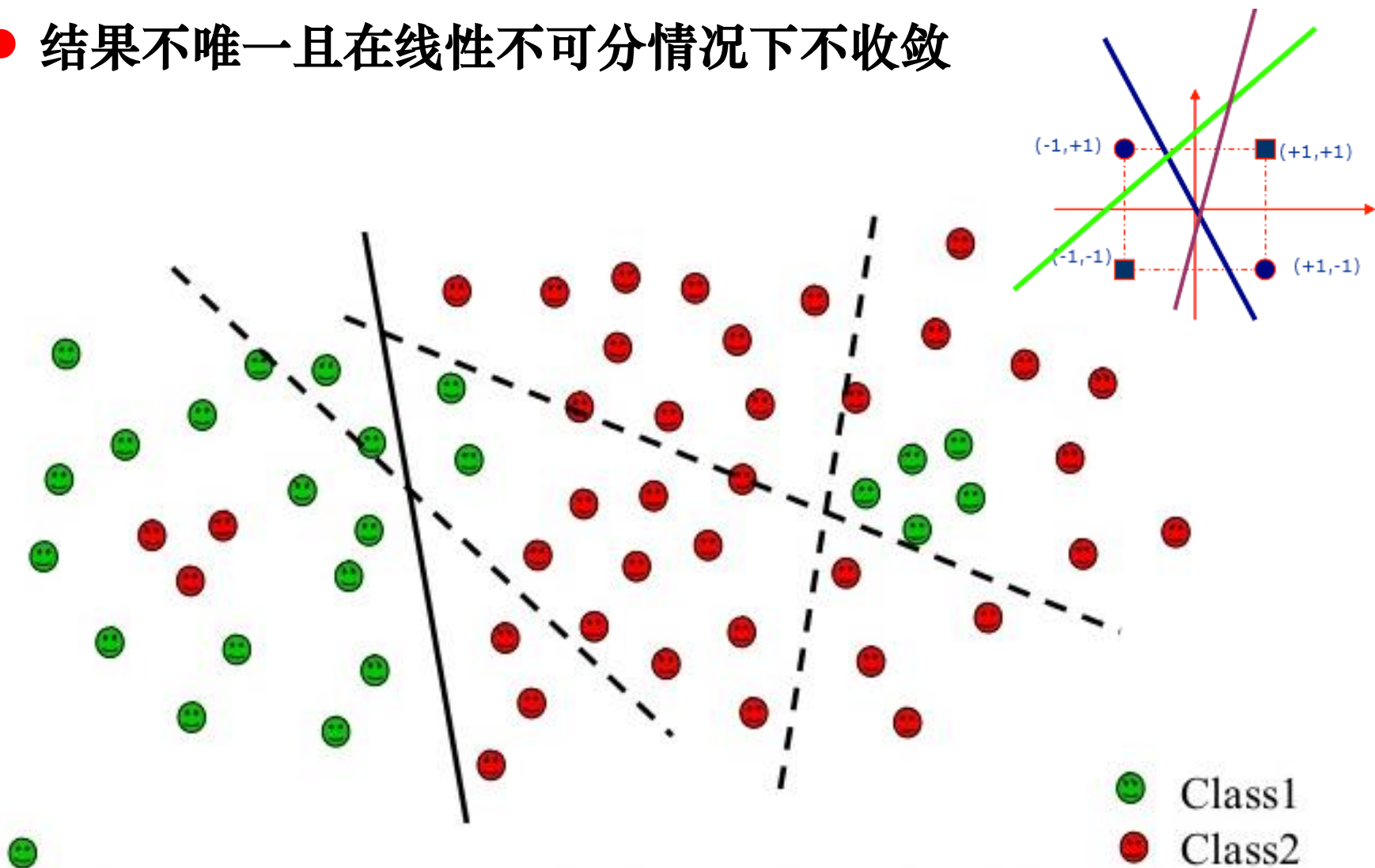
$$\mathbf{y}^{(7)t} = (-1 \quad -2 \quad -1)$$

$$\mathbf{a}(7)^t \mathbf{y}^{(7)} = (2 \quad -2 \quad 1) \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix} = 1 > 0$$

$$\mathbf{a}(8)^t = (2 \quad -2 \quad 1) \quad (\text{no change})$$

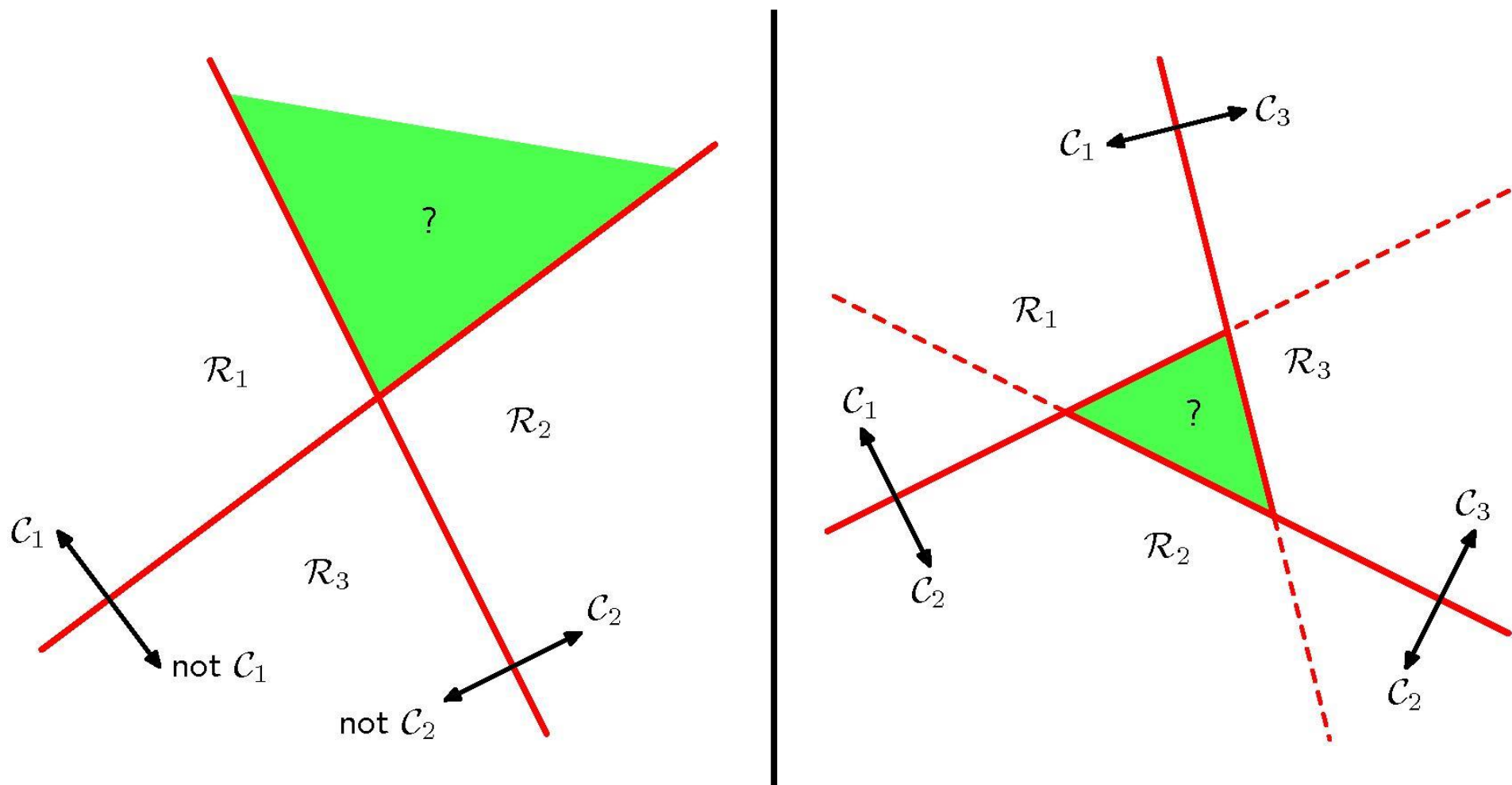
感知机准则

- 结果不唯一且在线性不可分情况下不收敛



多分类问题

- 1 vs. (N-1) or 1 vs. 1



生成式模型和判别式模型

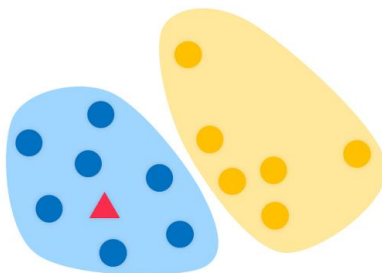
● 生成式模型

优点:

- 信息丰富
- 单类问题灵活性强
- 增量学习
- 合成缺失数据

缺点:

- 学习过程复杂
- 为分布牺牲分类性能



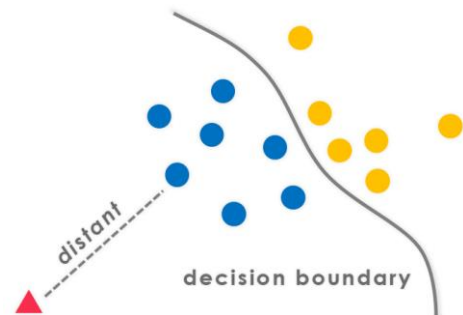
● 判别式模型

优点:

- 类间差异清晰
- 分类边界灵活
- 学习简单
- 性能较好

缺点:

- 不能反应数据特性
- 需要全部数据进行学习



由生成模型可以得到判别模型，
但由判别模型得不到生成模型。