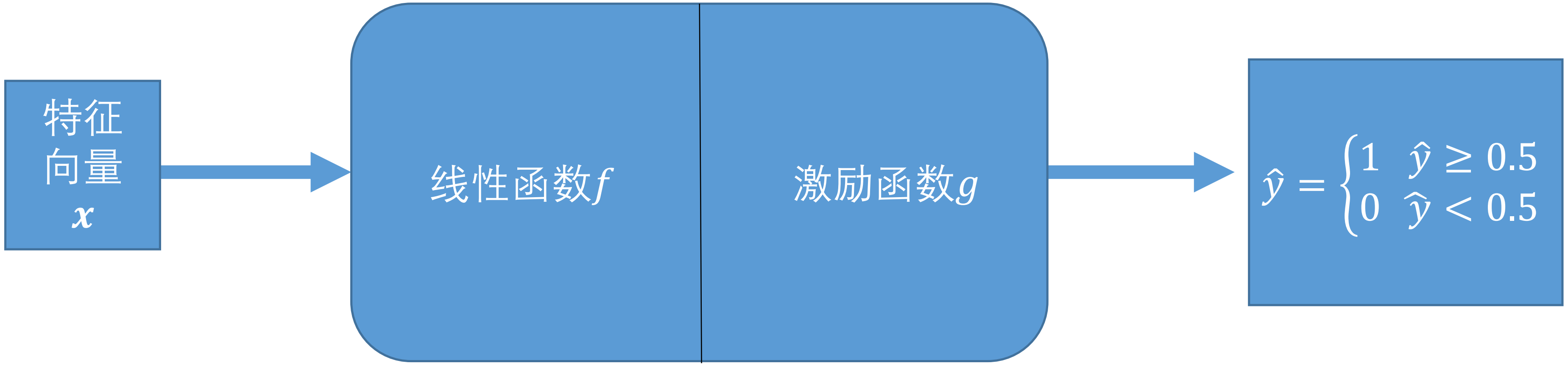
Preview

- 逻辑回归
- 支持向量机
- 朴素贝叶斯分类器

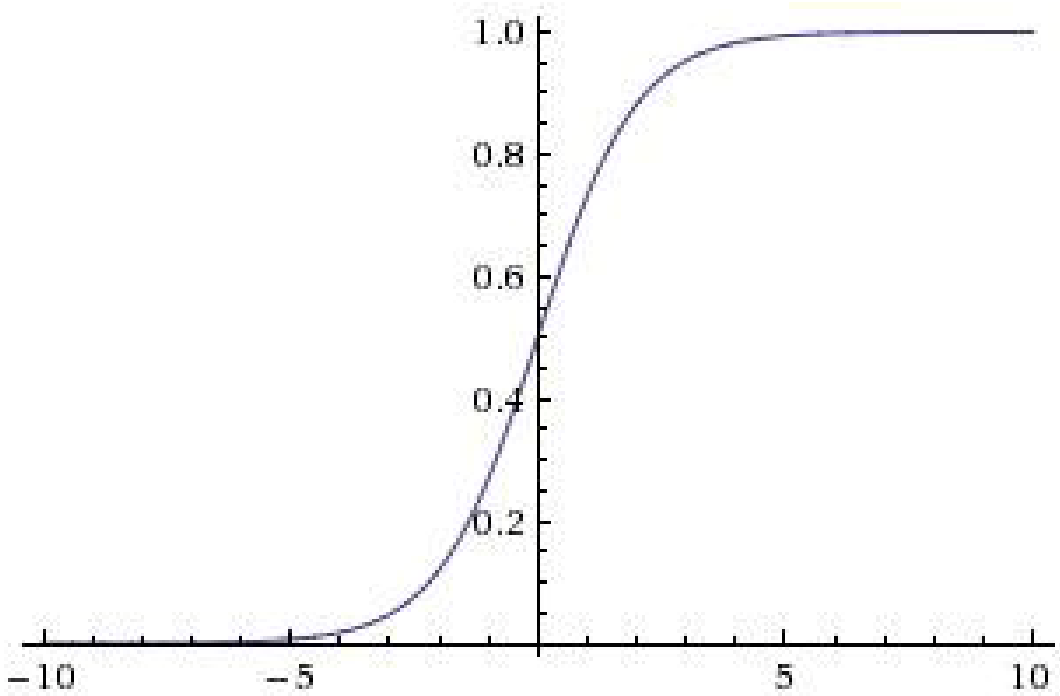
逻辑回归



-
- $f(\boldsymbol{x}) = \boldsymbol{\theta}^T\boldsymbol{x} + \theta_0$
- $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \ldots\ldots], \boldsymbol{x} = [x_1, x_2, x_3, x_4, \ldots\ldots]$
- $g(z) = \sigma(z) ------sigmoid\,函数$

Sigmoid函数

设置阈值

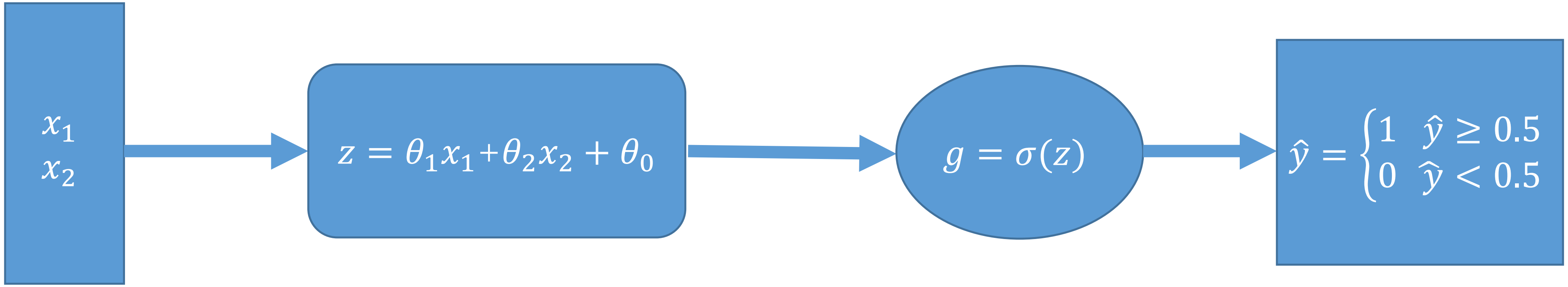


$$g(z) = \frac{1}{1+e^{-z}}$$

场景

- 选课：已有标注数据集---------$\{(\boldsymbol{x}, y)^{(i)}\}$, m个样本
- $\boldsymbol{x}=[x_1, x_2]$，$x_1$------评教分，$x_2$------平均分
- $y=\begin{cases}1\\0\end{cases}$，1------可选，0------不可选

问题抽象化


x_1
x_2
$z = \theta_1 x_1 + \theta_2 x_2 + \theta_0$
$g = \sigma(z)$
$\hat{y} = \begin{cases} 1 & \hat{y} \geq 0.5 \\ 0 & \hat{y} < 0.5 \end{cases}$

假设函数

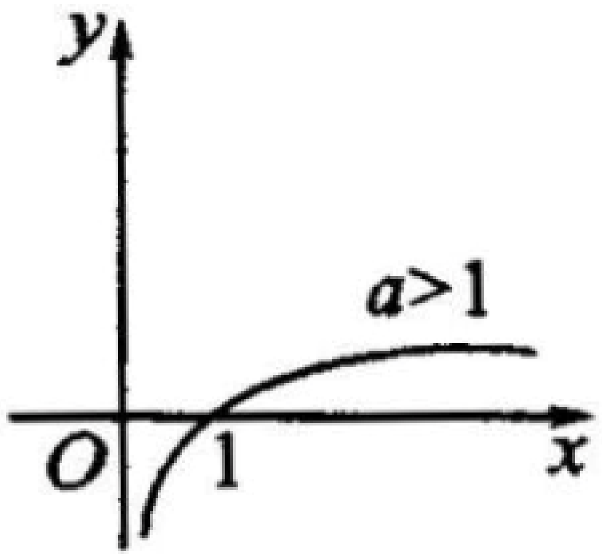
- $h_\theta(\boldsymbol{x}) = \sigma(\theta_1 x_1 + \theta_2 x_2 + \theta_0)$

$h_\theta(x)$ 的值有特殊的含义，它表示结果取1的概率，因此对于输入x分类结果为类别1和类别0的概率分别为：

$$
\begin{aligned}
P(y=1 \mid x;\theta) &= h_\theta(x) \\
P(y=0 \mid x;\theta) &= 1-h_\theta(x)
\end{aligned}
\qquad (1)
$$

损失函数

- $\hat{y} = h_\theta(\boldsymbol{x}) = \sigma(\theta_1 x_1 + \theta_2 x_2 + \theta_0)$
- $L(\hat{y}, y) = -y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))$



代价函数（交叉熵损失函数）

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))$$

梯度下降算法

$$\theta_j := \theta_j - \alpha \frac{\delta}{\delta_{\theta_j}} J(\theta)$$

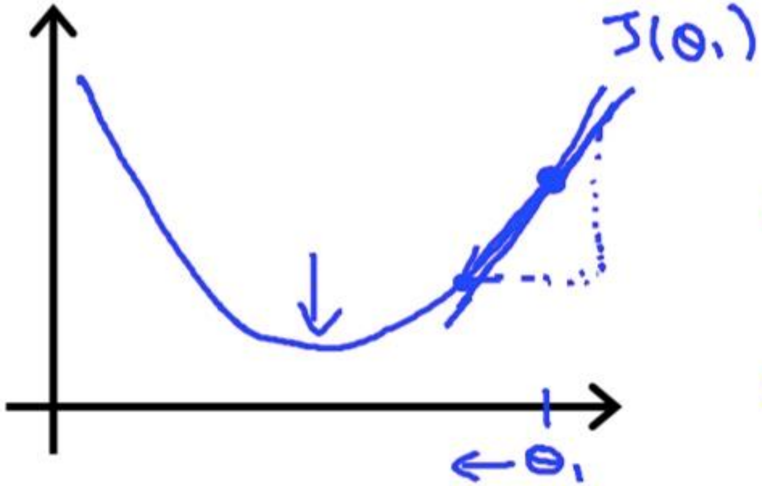
$$
\begin{aligned}
J(\theta) &= -\frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}(\log(1+e^{-\theta^T x^{(i)}})) + (1-y^{(i)})(-\theta^T x^{(i)} - \log(1+e^{-\theta^T x^{(i)}}))\right] \\
&= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\theta^T x^{(i)} - \theta^T x^{(i)} - \log(1+e^{-\theta^T x^{(i)}})\right] \\
&= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\theta^T x^{(i)} - \log e^{\theta^T x^{(i)}} - \log(1+e^{-\theta^T x^{(i)}})\right]_{③} \\
&= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\theta^T x^{(i)} - \left(\log e^{\theta^T x^{(i)}} + \log(1+e^{-\theta^T x^{(i)}})\right)\right]_{②} \\
&= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\theta^T x^{(i)} - \log(1+e^{\theta^T x^{(i)}})\right]
\end{aligned}
$$

这次再计算$J(\theta)$对第j个参数分量θ_j求偏导:

$$
\begin{aligned}
\frac{\partial}{\partial\theta_j}J(\theta) &= \frac{\partial}{\partial\theta_j}\left(\frac{1}{m}\sum_{i=1}^{m}\left[\log(1+e^{\theta^T x^{(i)}}) - y^{(i)}\theta^T x^{(i)}\right]\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}\left[\frac{\partial}{\partial\theta_j}\log(1+e^{\theta^T x^{(i)}}) - \frac{\partial}{\partial\theta_j}\left(y^{(i)}\theta^T x^{(i)}\right)\right] \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(\frac{x_j^{(i)}e^{\theta^T x^{(i)}}}{1+e^{\theta^T x^{(i)}}} - y^{(i)}x_j^{(i)}\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}
\end{aligned}
$$

梯度下降算法的正确性

- 以线性回归为例
- $h_\theta(x) = \theta_0 \; x$
- 方差代价函数（均方误差MSE）
- $J(\theta_0) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$

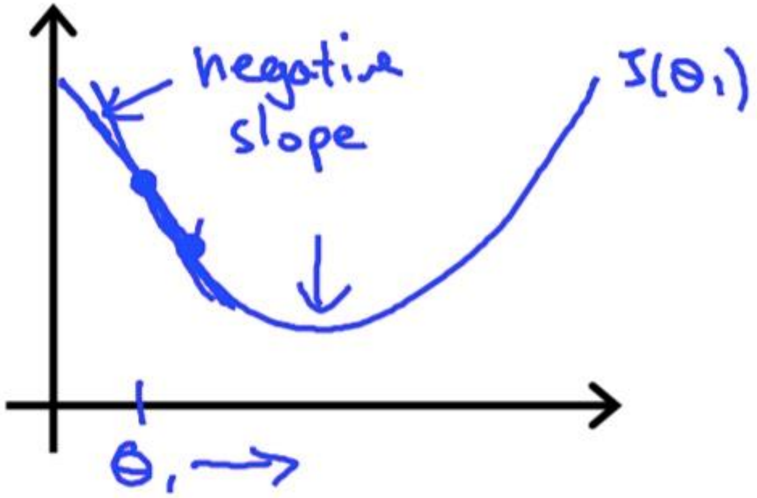


$(\theta_1 \in \mathbb{R})$

$\frac{\partial}{\partial \theta_1}$

$$\theta_1 := \theta_1 - \alpha \underbrace{\boxed{\frac{d}{d\theta_1} J(\theta_1)}}_{\geq 0}$$

$$\theta_1 := \theta_1 - \underline{\alpha} \cdot (\text{positive number})$$

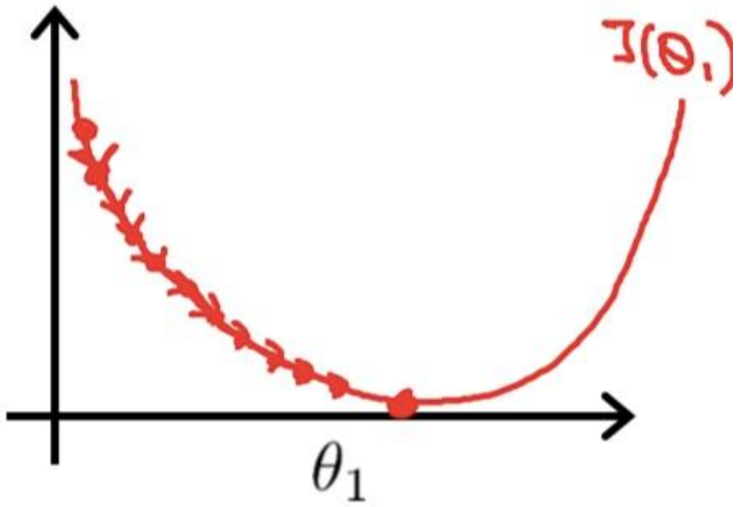


$$\underbrace{\frac{d}{d\theta_1} J(\theta_1)}_{\leq 0}$$

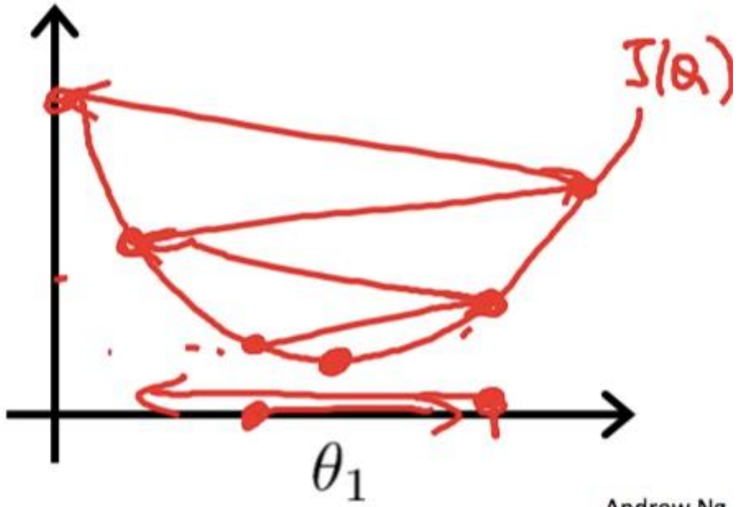
$$\theta_1 := \underline{\theta_1} - \alpha \, (\text{negative number})$$

$$\boxed{\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)}$$

If α is too small, gradient descent can be slow.



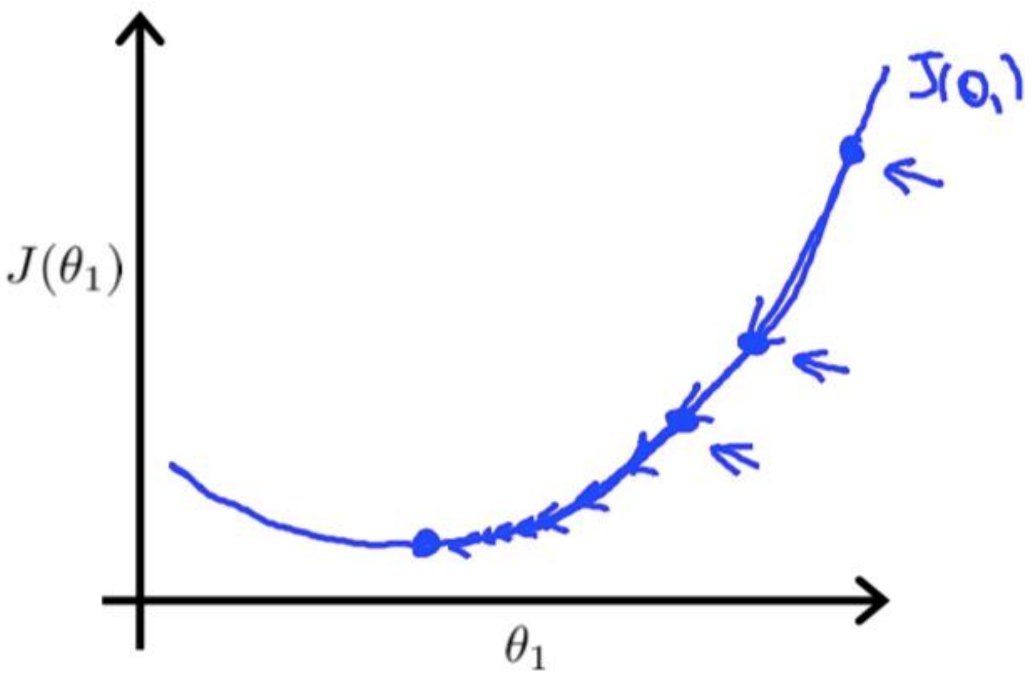
If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



Gradient descent can converge to a local minimum, even with the learning rate α fixed.

$$\theta_1 := \theta_1 - \alpha \boxed{\frac{d}{d\theta_1} J(\theta_1)}$$

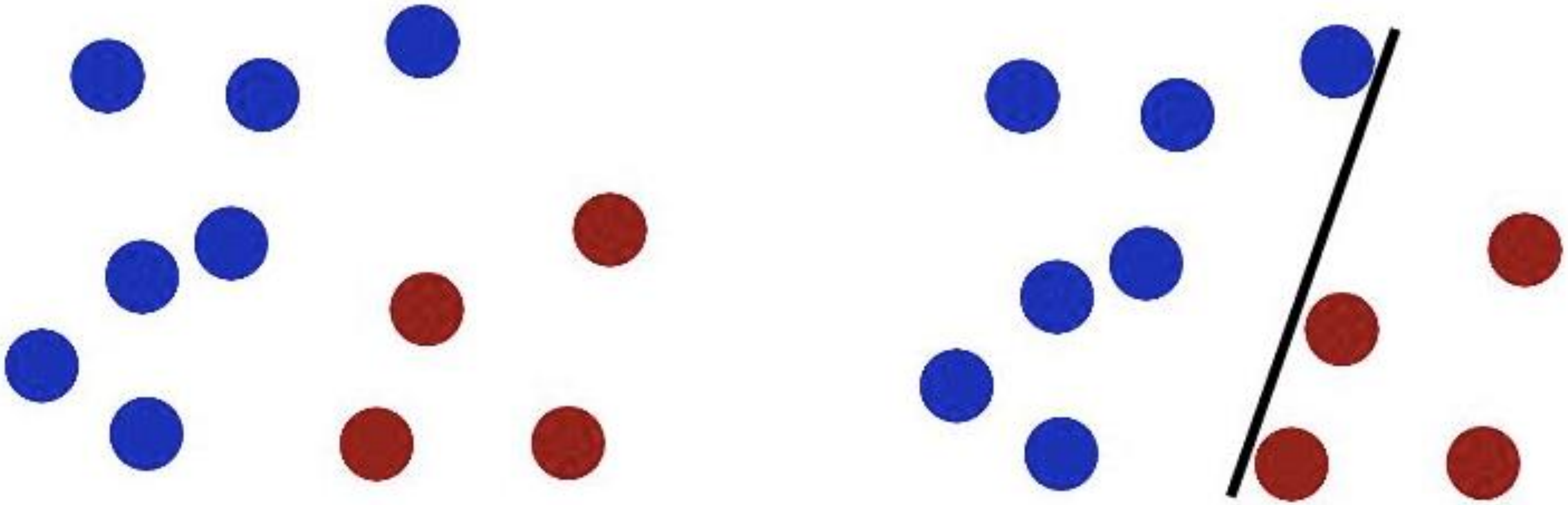
As we approach a local minimum, gradient descent will automatically take smaller steps. So, no need to decrease α over time.



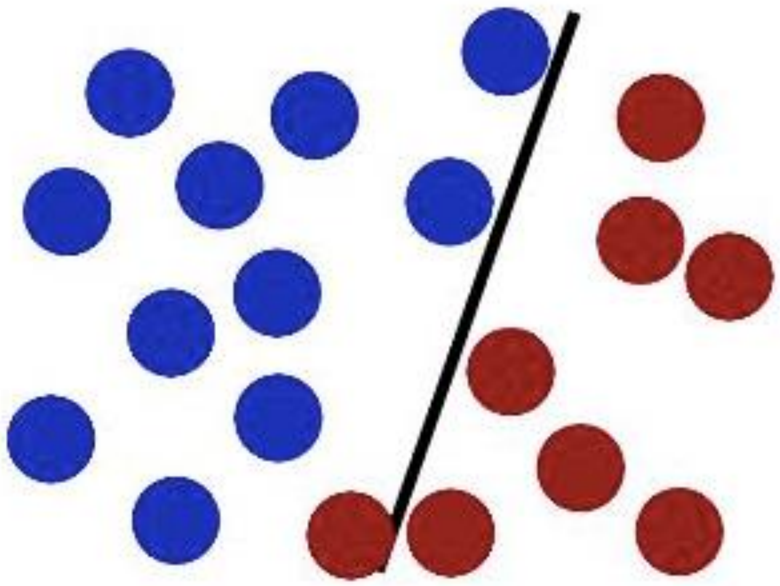
支持向量机

场景：在很久以前的情人节，大侠要去救他的爱人，但魔鬼和他玩了一个游戏。

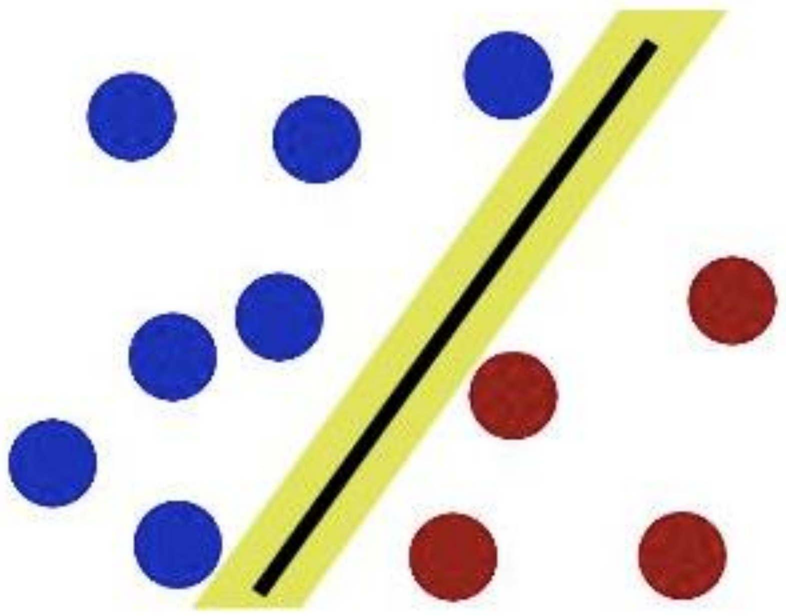
魔鬼在桌子上似乎有规律放了两种颜色的球，说：“你用一根棍分开它们？要求：尽量在放更多球之后，仍然适用。”



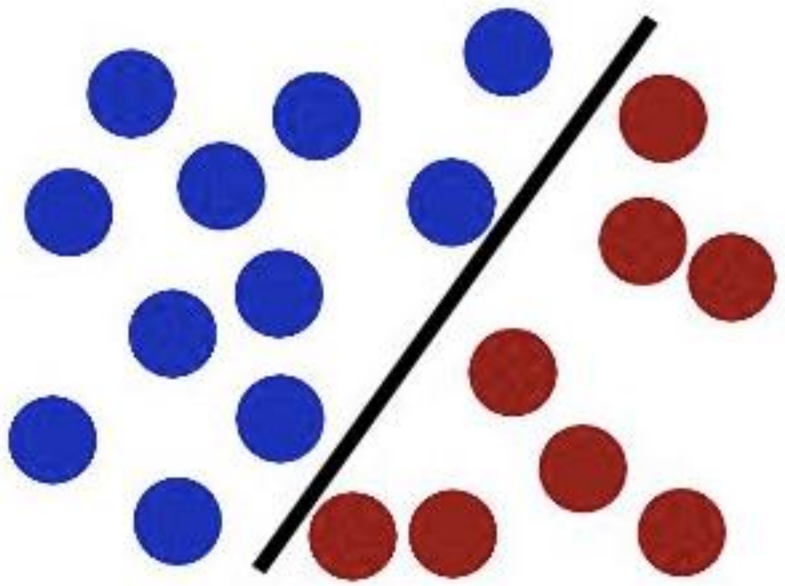
然后魔鬼，又在桌上放了更多的球，似乎有一个球站错了阵营。



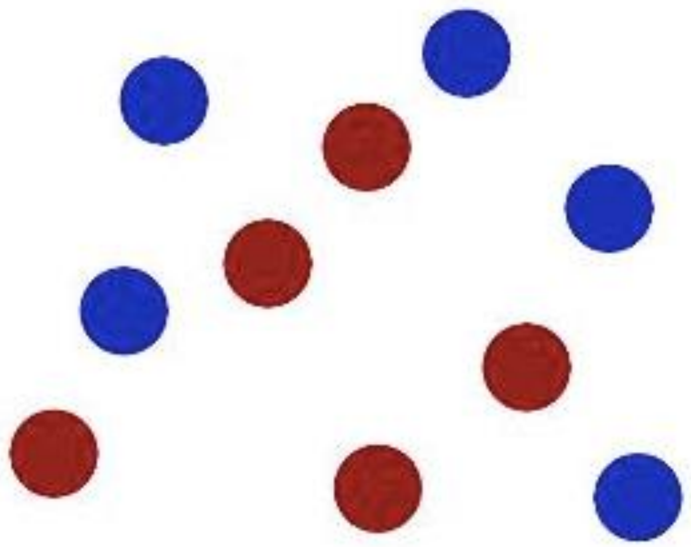
SVM就是试图把棍放在最佳位置，好让在棍的两边有尽可能大的间隙。



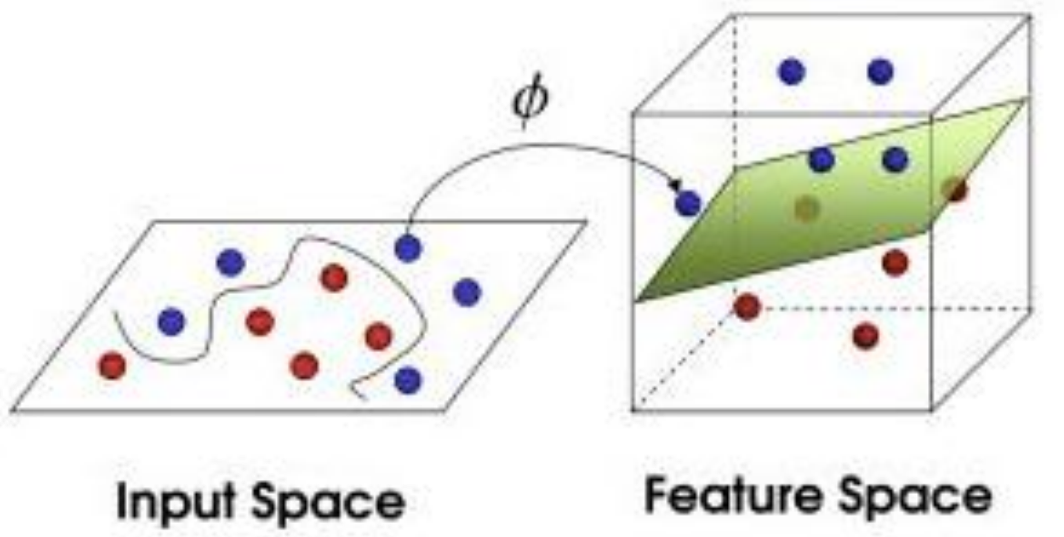
现在即使魔鬼放了更多的球，棍仍然是一个好的分界线



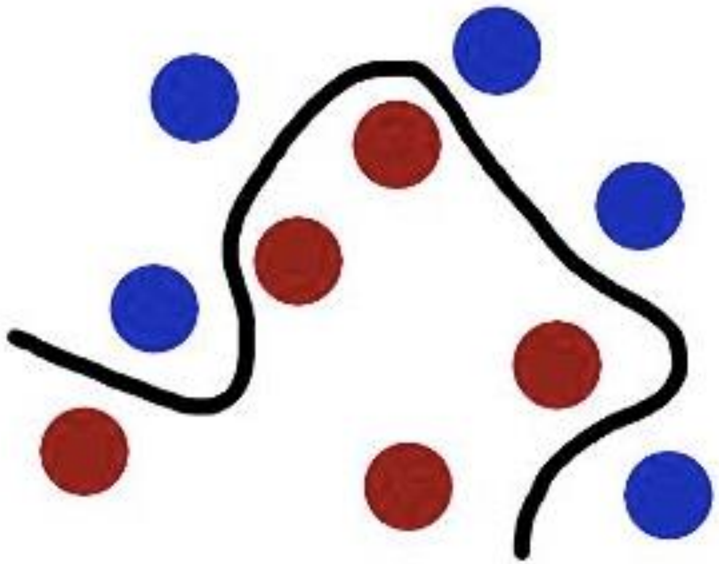
魔鬼看到大侠已经学会了一个trick，于是魔鬼给了大侠一个新的挑战。



现在，大侠没有棍可以很好帮他分开两种球了，现在怎么办呢？当然像所有武侠片中一样大侠桌子一拍，球飞到空中。然后，凭借大侠的轻功，大侠抓起一张纸，插到了两种球的中间。



现在，从魔鬼的角度看这些球，这些球看起来像是被一条曲线分开了。



再之后，无聊的大人们，把这些球叫做「data」，把棍子叫做「classifier」，最大间隙trick 叫做「optimization」，拍桌子叫做「kernelling」，那张纸叫做「hyperplane」。

在样本空间中，划分超平面可通过如下线性方程来描述:

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=0\ , \tag{6.1}$$

其中 $\boldsymbol{w}=(w_1;w_2;\ldots;w_d)$ 为法向量，决定了超平面的方向; b 为位移项，决定了超平面与原点之间的距离. 显然，划分超平面可被法向量 $\boldsymbol{w}$ 和位移 b 确定，

下面我们将其记为 $(\boldsymbol{w},b)$. 样本空间中任意点 $\boldsymbol{x}$ 到超平面 $(\boldsymbol{w},b)$ 的距离可写为

$$r=\frac{|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b|}{||\boldsymbol{w}||}\ . \tag{6.2}$$

假设超平面 $(\boldsymbol{w}, b)$ 能将训练样本正确分类, 即对于 $(\boldsymbol{x}_i, y_i) \in D$, 若 $y_i = +1$, 则有 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b > 0$; 若 $y_i = -1$, 则有 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b < 0$. 令

$$\begin{cases} \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b \geqslant +1, & y_i = +1\,; \\ \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b \leqslant -1, & y_i = -1\,. \end{cases} \tag{6.3}$$

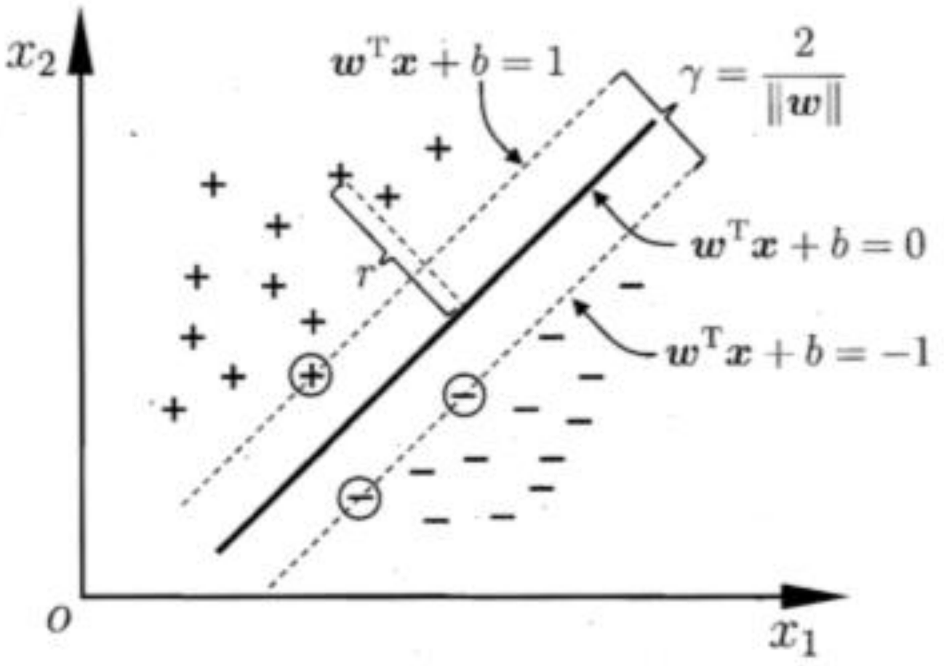


图 6.2 支持向量与间隔

如图 6.2 所示, 距离超平面最近的这几个训练样本点使式(6.3)的等号成立, 它们被称为“支持向量”(support vector), 两个异类支持向量到超平面的距离之和为

$$\gamma = \frac{2}{\|\boldsymbol{w}\|}\,, \tag{6.4}$$

它被称为“间隔”(margin).

欲找到具有“最大间隔”(maximum margin)的划分超平面, 也就是要找到能满足式(6.3)中约束的参数 $\boldsymbol{w}$ 和 b, 使得 γ 最大, 即

$$\begin{aligned}
&\max_{\boldsymbol{w},b} \ \frac{2}{||\boldsymbol{w}||} \\
&\text{s.t.}\ \ y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geqslant 1, \quad i = 1,2,\ldots,m.
\end{aligned} \tag{6.5}$$

$$\arg\max_{w,b}\left\{\frac{1}{||w||}\min_{n}\left[y_i(w^T x_i + b)\right]\right\}$$

显然, 为了最大化间隔, 仅需最大化 $\|\boldsymbol{w}\|^{-1}$, 这等价于最小化 $\|\boldsymbol{w}\|^2$. 于是, 式(6.5)可重写为

$$\begin{aligned} \min_{\boldsymbol{w},b} \ & \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{s.t.} \ & y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geqslant 1, \quad i = 1, 2, \ldots, m. \end{aligned} \tag{6.6}$$

这就是支持向量机(Support Vector Machine, 简称 SVM)的基本型.

朴素贝叶斯分类器

条件概率公式

- 公式中P(AB)为事件AB的联合概率，P(A|B)为条件概率，表示在B条件下A的概率，P(B)为事件B的概率。

$$P(A|B) = \frac{P(AB)}{P(B)}$$

全概率公式

- 完备事件集：如下的一组事件称为一个“完备事件群”。简而言之，就是事件之间两两互斥，所有事件的并集是整个样本空间（必然事件）。

$$B_iB_j = \varnothing \quad (i \neq j)$$

$$B_1 + B_2 + \cdots = \Omega$$

$$P(A) = P(B_1)P(A|B_1) + P(B_2)P(A|B_2) + P(B_3)P(A|B_3) + \cdots$$

贝叶斯公式

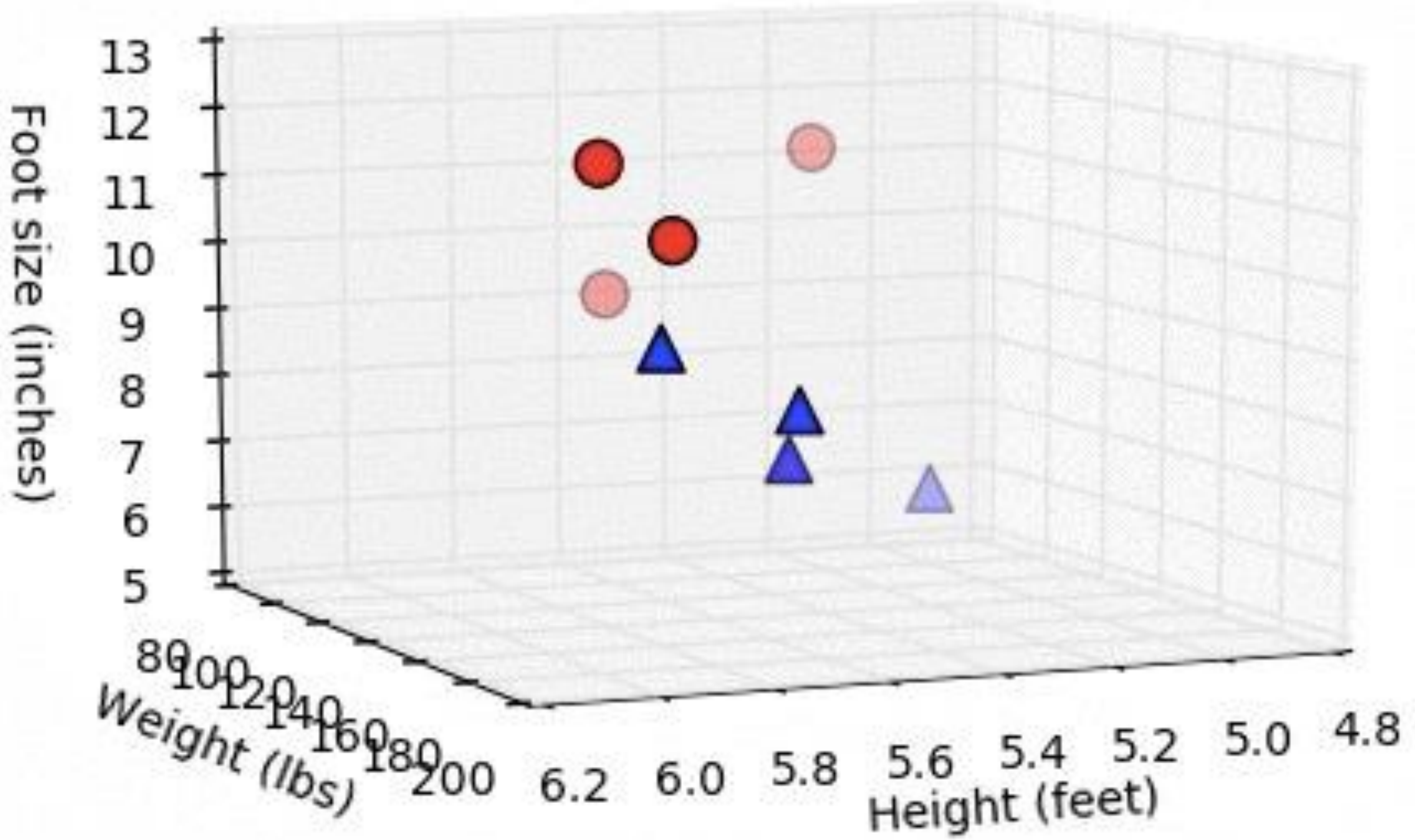
$$P(B_i|A)=\frac{P(AB_i)}{P(A)}=\frac{P(B_i)P(A|B_i)}{\sum_j P(B_j)P(A|B_j)}$$

- 在全概率公式中，如果将A看成是“结果”，Bi看成是导致结果发生的诸多“原因”之一，那么全概率公式就是一个“**原因推结果**”的过程。但贝叶斯公式却恰恰相反。贝叶斯公式中，我们是知道结果A已经发生了，所要做的是反过来研究造成结果发生的原因，是XX原因造成的可能性有多大，即“**结果推原因**”。

问题：如何通过知道一个人的身高、体重以及脚的尺寸，去判断这个人是男是女？

性别	身高（英尺）	体重（磅）	脚的尺寸（英寸）
男	6	180	12
男	5.92	190	11
男	5.58	170	12
男	5.92	165	10
女	5	100	6
女	5.5	150	8
女	5.42	130	7
女	5.75	150	9

训练数据



Foot size (inches)
13
12
11
10
9
8
7
6
5
80
100
120
140
160
180
200
Weight (lbs)
6.2
6.0
5.8
5.6
5.4
5.2
5.0
4.8
Height (feet)

问题抽象化：

- 我们记F1、F2、F3分别为身高、体重、脚尺寸的随机变量，取值当然是各自坐标轴上的值。再记C为分类结果的随机变量，取值为“男”或“女”。
- 我们所要解决的问题的本质，就是在已知F1、F2、F3的时候，判断p(男|F1,F2,F3)与p(女|F1,F2,F3)究竟哪个更加大，换言之，这个人是更像男还是更像女，写成数学语言就是：

$$\hat{c} = \arg\max_{c} p(C|F_1, F_2, F_3)$$

- 根据贝叶斯公式，得

$$p(C|F_1,F_2,F_3)=\frac{p(F_1,F_2,F_3|C)P(C)}{p(F_1,F_2,F_3)}$$

- 我们的任务只是比较大小，而上式右边的分母是一个常数，不妨将其忽略掉以简化计算。这时候我们的问题就剩下如何求

$p(F_1,F_2,F_3|C)P(C)$ 了。

- 我们认定F1、F2、F3是彼此独立的特征，那么有：

$$p(F_1, F_2, F_3|C) = p(F_1|C)p(F_2|C)p(F_3|C)$$

- 于是我们的问题就化简为了

$$\hat{c} = \arg\max_c p(F_1|C)p(F_2|C)p(F_3|C)P(C)$$

我们还有一个严重的问题没有解决——连续随机变量。我们不能想离散随机变量那样计算 $p(F_i|C)$ 。

- 然而我们可以假设，身高、体重、脚尺寸都是**正态分布**。
  我们分析一下样本数据的数字特征：

性别	均值(身高)	方差(身高)	均值(体重)	方差(体重)	均值(脚的尺寸)	方差(脚的尺寸)
男性	5.855	3.5033e-02	176.25	1.2292e+02	11.25	9.1667e-01
女性	5.4175	9.7225e-02	132.5	5.5833e+02	7.5	1.6667e+00

得到了均值与方差，也就得到了正态分布的μ与σ^2参数。
如此，$p(F_1|C)p(F_2|C)p(F_3|C)$就能顺利求出了。
比如，

$$p(F_1 = 6|男) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-(6-\mu)^2}{2\sigma^2}\right) \approx 1.5789$$

值得注意的是，这里求的是连续随机变量的概率密度，所以求出比1大的值也是正常的[5]。
剩下的$P(C)$可以用样本中男女的出现频率来估计，估算出都是0.5。

综上，我们计算可得：

$$p(F_1=6|\text{男})p(F_2=130|\text{男})p(F_3=8|\text{男})P(\text{男}) = 6.1984\times 10^{-9}$$
$$p(F_1=6|\text{女})p(F_2=130|\text{女})p(F_3=8|\text{女})P(\text{女}) = 5.3778\times 10^{-4}$$

从计算结果可以看出，这个人是女性的可能性远大于是男性的可能性。

如果要通过编程实现这一过程，还要考虑平滑处理，这里不再赘述。

参考资料

- 《机器学习》，周志华
- Machine Learning, Andrew Ng, Coursera