《机器学习》课件

# 3.1贝叶斯学习

# 贝叶斯学习

- 贝叶斯 **Thomas Bayes**， 1702-1763年，出生于伦敦，英国数学家，做过神甫。1742年成为英国皇家学会会员。
- 首先将归纳推理法用于概率论基础理论，创立了贝叶斯统计理论。
- 《机会的学说概论》，1758年
- 《论机会学说问题的求解》，1763年

# 贝叶斯学习

- 贝叶斯理论
- MAP，ML假设
- MAP学习器
- 朴素贝叶斯分类器（Naive Beyas Learner）
- HMM

# 贝叶斯方法的两个主要作用

- 提供实用学习算法

  朴素贝叶斯学习

  贝叶斯置信网学习

  先验知识与概率结合

  后验概率计算

- 提供了有用的概念框架

  提供了评估其他算法的“黄金标准” （Gold Standard）

  额外地提供了对“奥坎姆剃刀” 意义的深刻洞察

# 贝叶斯理论

$$P(h/D)=\frac{P(D/h)P(h)}{P(D)}$$

- 癌症诊察

  已知：1）某项指标的化验数据，输出 +，-

  2）先验知识

  所有人口中患该病的概率0.08%;

  患者中98%呈阳性反应（+），正确;

  患者中2%呈阴性反应（-），漏警;

  非患者97%呈阴性反应（-），正确;

  非患者3%呈阳性反应（+），虚警。

# 贝叶斯理论例

$$P(cancer) = 0.008, P(\neg cancer) = 0.992$$

$$P(+/cancer) = 0.98, P(-/cancer) = 0.02$$

$$P(+/\neg cancer) = 0.03, P(-/\neg cancer) = 0.97$$

如果某人化验结果为+，该病人患癌症的可能性：

$$P(+/cancer)P(cancer) = 0.98 \times 0.008 = 0.0078$$

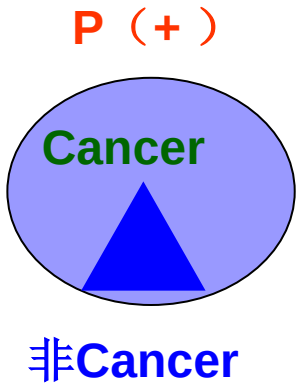
$$P(+/\neg cancer)P(\neg cancer) = 0.03 \times 0.992 = 0.0298$$

# 贝叶斯理论例

- 后验概率：

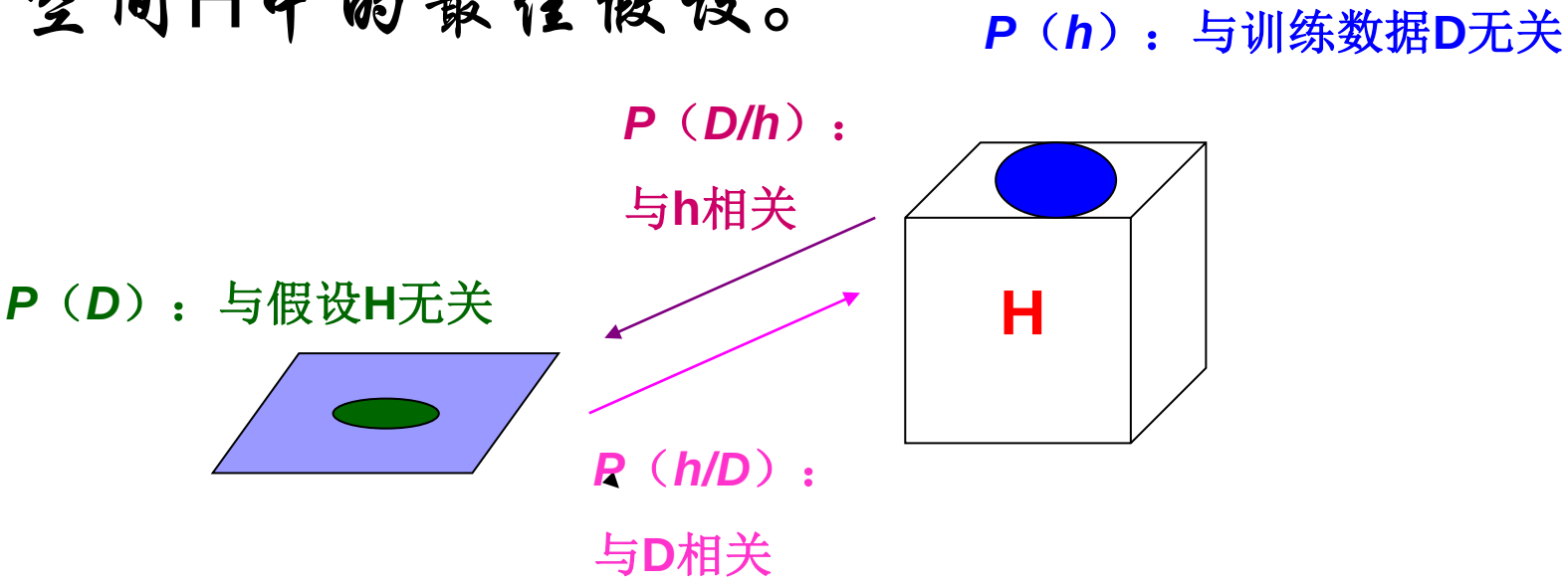
$$P(cancer/+)=\frac{0.0078}{0.0078+0.0298}=0.21$$

$$P(\neg cancer/+)=\frac{0.0298}{0.0078+0.0298}=0.79$$



# 贝叶斯理论

- 机器学习目标：给定训练数据D，确定假设空间H中的最佳假设。



# 贝叶斯理论

- 贝叶斯公式

给定假设h下，训练数据D的条件概率

假设h的先验概率

$$P(h/D) = \frac{P(D/h)P(h)}{P(D)}$$

给定训练数据D下，假设h的条件概率

训练数据D的先验概率

# 贝叶斯理论

- MAP---极大后验假设：

$$h_{MAP} \equiv \arg\max_{h \in H} P(h/D)$$

$$= \arg\max_{h \in H} \frac{P(D/h)P(h)}{P(D)}$$

$$= \arg\max_{h \in H} P(D/h)P(h)$$

# 贝叶斯理论

- ML---极大似然假设（$h$的先验概率未知）

$$h_{ML} = \arg\max_{h \in H} P(D/h)$$

$$P(h_i) = P(h_j), i \neq j$$

# 最大似然估计

- Maximum Likelihood (ML)估计
  - 估计的参数 $\theta$ 是确定而未知的，Bayes估计方法则视 $\theta$ 为随机变量。
  - 概率密度函数的形式已知，参数未知，为了描述概率密度函数 $p(x|\omega_i)$ 与参数 $\theta$ 的依赖关系，用 $p(x|\omega_i,\theta)$ 表示。
- 独立地按概率密度 $p(x|\theta)$ 抽取样本集 $K=\{x_1, x_2, \ldots, x_N\}$，用 $K$ 估计未知参数 $\theta$

# 似然函数

- 似然函数，存在就是有道理的

$$l(\boldsymbol{\theta}) = p(K \mid \boldsymbol{\theta}) = p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \mid \boldsymbol{\theta})$$

$$= \prod_{k=1}^{N} p(\mathbf{x}_k \mid \boldsymbol{\theta})$$

- 对数(loglarized)似然函数

$$H(\boldsymbol{\theta}) = \sum_{k=1}^{N} \ln p(\mathbf{x}_k \mid \boldsymbol{\theta})$$

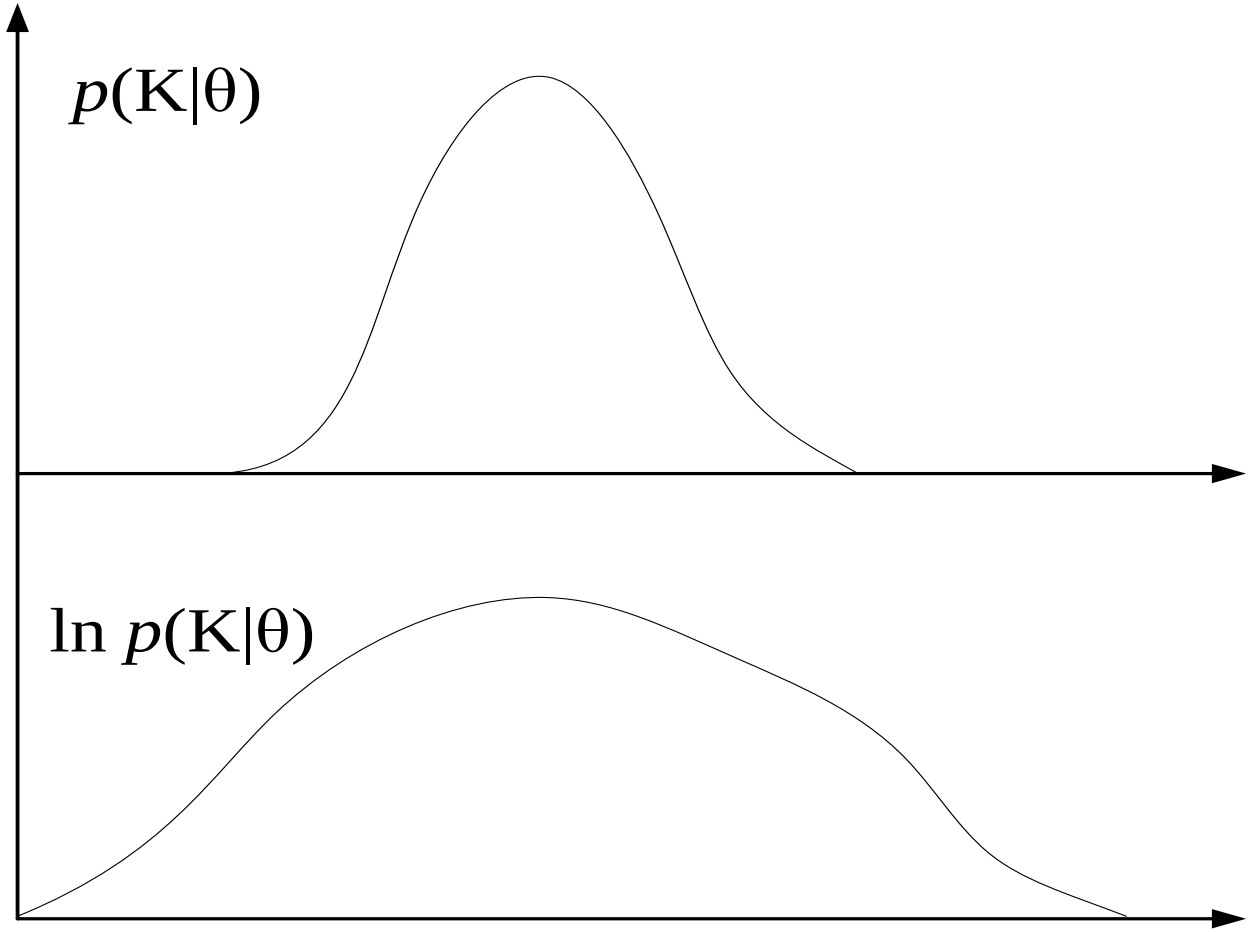
- 乘法-> 加法

# 最大似然估计

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta})$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{k=1}^{n} \ln p(\mathbf{x}_k \mid \boldsymbol{\theta})$$

# 最大似然估计示意图



# 计算方法

- 最大似然估计量使似然函数梯度为0

$$\nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta})\big|_{\hat{\theta}_{ML}} = \sum_{k=1}^{N} \nabla_{\boldsymbol{\theta}} \ln p(x_k \mid \boldsymbol{\theta})\big|_{\hat{\theta}_{ML}} = 0$$

$$\nabla_{\boldsymbol{\theta}} = \left[ \frac{\partial}{\partial \theta_1} \quad \cdots \quad \frac{\partial}{\partial \theta_s} \right]^T$$

# 贝叶斯估计-最大后验概率（理解层面就可以）

- 用一组样本集$K=\{x_1, x_2, \ldots, x_N\}$估计未知参数 $\theta$
- 未知参数 $\theta$视为随机变量，先验分布为 $p(\theta)$，而在已知样本集$K$出现的条件下的后验概率为$p(\theta|K)$
- 最大后验概率估计-Maximum a posteriori (MAP)

$$\hat{\theta}_{\mathrm{MAP}} = \arg\max_{\theta} p(\theta \mid K)$$

$$= \arg\max_{\theta} \frac{p(K \mid \theta)\, p(\theta)}{p(K)}$$

$$= \arg\max_{\theta} p(K \mid \theta)\, p(\theta)$$

正则，即常数的时候，极大似然是其的一个特例，或者工程实现

# 一元正态分布例解

$$p(x_k \mid \theta_1 = \mu, \theta_2 = \sigma^2) = \frac{1}{\sqrt{2\pi\theta_2}} \exp\left(-\frac{(x_k - \theta_1)^2}{2\theta_2}\right)$$

$$\ln p(x_k \mid \theta_1, \theta_2) = -\frac{1}{2}\ln(2\pi\theta_2) - \frac{1}{2\theta_2}(x_k - \theta_1)^2$$

# 一元正态分布均值的估计

$$\nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta})\big|_{\hat{\theta}_{ML}} = \sum_{k=1}^{N} \nabla_{\boldsymbol{\theta}} \ln p(x_k \mid \boldsymbol{\theta})\big|_{\hat{\theta}_{ML}} = 0$$

$$\frac{\partial}{\partial \theta_1} \ln p(x_k \mid \theta_1, \theta_2) = \frac{1}{\theta_2}(x_k - \theta_1)$$

代入前式, 得

$$\hat{\mu}_{ML} = \frac{1}{N} \sum_{k=1}^{N} x_k$$

# 一元正态分布方差的估计

$$\frac{\partial}{\partial \theta_2} \ln p(x_k \mid \theta_1, \theta_2) = -\frac{1}{2\theta_2} + \frac{(x_k - \theta_1)^2}{2\theta_2^2}$$

代入前式得

$$\hat{\sigma}^2_{ML} = \frac{1}{N} \sum_{k=1}^{N} (x_k - \hat{\mu})^2$$

# 多元正态分布参数最大似然估计

$$\hat{\boldsymbol{\mu}}_{ML} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k \qquad \hat{\Sigma} = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^T$$

- 均值估计是无偏的，协方差矩阵估计是有偏的。
- 协方差矩阵的无偏估计是：

$$\hat{\Sigma} = \frac{1}{N-1}\sum_{k=1}^{N}(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^T$$

总体均值向量和协方差矩阵

$$\boldsymbol{\mu} = E(\mathbf{x}) = (\mu_1, \mu_2, \ldots, \mu_n)^T,\ \mu_i = E(x_i)$$

$$\Sigma = E\left\{(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T\right\} = (\sigma_{ij}^2)_{n*n}, \sigma_{ij}^2 = E\left[(x_i-\mu_i)(x_j-\mu_j)\right]$$

# Brute-Force MAP学习器

蛮力

- 对H中每个h，计算

学习任务的先验知识，任意概率

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- 输出

$$h_{MAP} \equiv \arg\max_{h \in H} P(h|D)$$

提供了判断一个学习算法学习性能的标准,针对不同的假设空间

# Brute-Force MAP学习器

- 给定前提：

  训练数据D无噪声

  C∈H

  假设概率未知

$$P(h) = \frac{1}{|H|}, h \in H$$

$$P(D|h) = \begin{cases} 1, d_i = h(x_i) \\ 0, d_i \neq h(x_i) \end{cases}$$

# Brute-Force MAP学习器

- 考察后验概率

h与D不一致

$$P(h|D)=\frac{P(D|h)P(h)}{P(D)}=\frac{0\times P(h)}{P(D)}=0$$

h与D一致

$$P(h|D)=\frac{1\times\frac{1}{|H|}}{P(D)}=\frac{1\times\frac{1}{|H|}}{\frac{|VS_{H,D}|}{|H|}}=\frac{1}{|VS_{H,D}|}$$

H中与D一致的假设空间

# Brute-Force MAP学习器

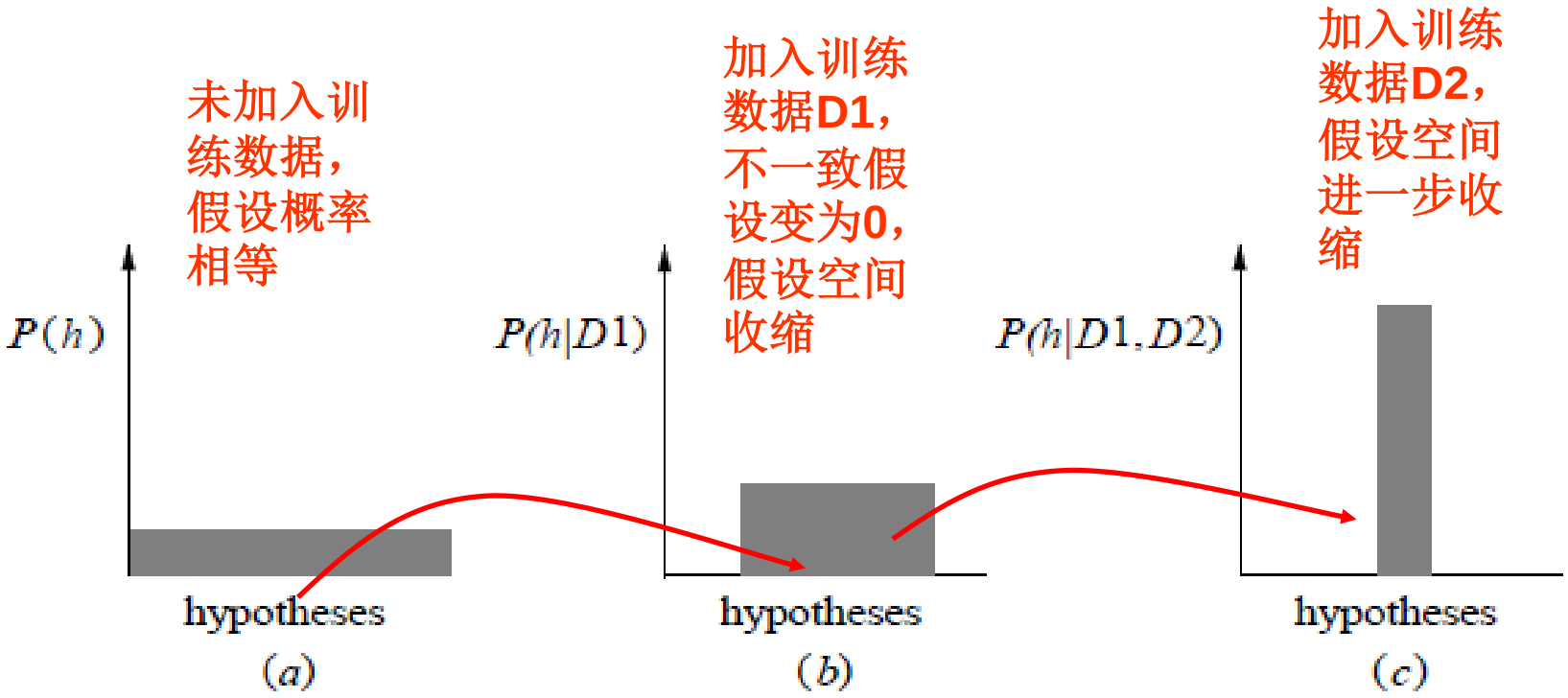
$$P(D)=\sum_{h_i\in H}P(D\mid h_i)P(h_i)$$

$$=\sum_{h_i\in VS_{H,S}}1\times\frac{1}{|H|}+\sum_{h_i\notin VS_{H,S}}0\times\frac{1}{|H|}$$

$$=\frac{|VS_{H,S}|}{|H|}$$

$$P(h\mid D)=\begin{cases}\dfrac{1}{|VS_{H,S}|}, h\text{与D一致}\\ 0,\ \text{其他}\end{cases}$$

# Brute-Force MAP学习器



# 极大似然和最小平方误差假设

- 连续值目标函数学习：神经网络、线性回归、多项式曲线拟合
- 证明：**特定**前提下，**任一**学习算法，如果使得输出假设的**预测**与**训练数据**之间的**误差平方最小化**，此时，该学习算法的输出就是一个建立在该训练数据上的**极大似然假设**。

# 朴素贝叶斯分类器

- 决策树、神经网络、最近邻方法之外，最实用的学习方法
- 适用性：

  中等或大规模数据学习

  实例属性有条件地独立于给定分类
- 成功应用

  故障诊断

  文本分类

# 朴素贝叶斯分类器

- 假定目标函数: $f : X \rightarrow V$
- 样例属性: $< a_1, a_2, a_3, ..., a_n >$
- 最可能的分类值为:

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2 \ldots a_n)$$

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2 \ldots a_n | v_j) P(v_j)$$

# 朴素贝叶斯分类器

- 朴素贝叶斯假定：

各属性分类结果独立

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

- 得到贝叶斯分类器：

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

# 朴素贝叶斯分类算法

- 朴素贝叶斯学习

  对于每个目标值Vj：

$$\hat{P}(v_j) \leftarrow \text{estimate } P(v_j)$$

  对于每个属性值：

$$\hat{P}(a_i|v_j) \leftarrow \text{estimate } P(a_i|v_j)$$

  分类新样例：

$$v_{NB} = \underset{v_j \in V}{\text{argmax}} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

# 朴素贝叶斯分类例

- 考虑Playtennis

$$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong$$

计算:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i|v_j)$$

$$P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y) = .005$$

$$P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n) = .021$$

$$\rightarrow v_{NB} = n$$

## 样本集合

| 示例 | 天气 | 温度 | 湿度 | 风力 | 打网球 |
|---|---|---|---|---|---|
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

# 朴素贝叶斯：细节

- 条件独立假设约束强

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j) \quad \hat{P}(v_j | \bar{x})$$

不必估计后验概率，仅需要计算：

$$\underset{v_j \in V}{\operatorname{argmax}} \hat{P}(v_j) \prod_i \hat{P}(a_i | v_j) = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) P(a_1 \ldots, a_n | v_j)$$

# EM算法

- 在许多实际的学习问题框架中，相关实例特征中只有一部分可观察到

- EM算法是存在隐含变量时广泛使用的一种学习方法，可用于变量的值从来没有被直接观察到的情形，只要这些变量所遵循的概率分布的一般形式已知
  - 用于贝叶斯网的训练
  - 用于马尔可夫模型的训练

# 估计k个高斯分布的均值

- 考虑D是一个实例集合，它由k个不同正态分布的混合所得分布生成
- 每个实例使用一个两步骤的过程形成：
  - 首先，随机选择k个正态分布中的一个
  - 其次，随机变量$x_i$按照此选择的分布生成
- 考虑一个简单情形：
  - 单个正态分布的选择基于均匀的概率进行，且k个正态分布有相同的方差
  - 学习任务：输出一个假设$h=<\mu_1 \ldots \mu_k>$，描述k个分布中每个分布的均值，找到极大似然假设，即使得$p(D|h)$最大化的假设

# 估计k个高斯分布的均值（2）

- 当给定从一个正态分布中抽取的数据实例$x_1,\ldots,x_m$时，很容易计算该分布的均值的**极大似然假设**，表示如下

$$\mu_{ML} = \arg\min_{\mu} \sum_{i=1}^{m} (x_i - \mu)^2 = \frac{1}{m}\sum_{i=1}^{m} x_i$$

- 然而，现在的问题涉及k个不同正态分布，而且不知道哪个实例是哪个分布产生的。这是一个涉及隐藏变量的典型例子
- 每个实例的完整描述是**三元组**$<x_i, z_{i1}, z_{i2}>$，其中$x_i$是第i个实例的观测值，$z_{i1}$和$z_{i2}$表示哪个正态分布被用来产生$x_i$，是**隐藏变量**

# 估计k个高斯分布的均值（3）

- 如果$z_{i1}$和$z_{i2}$的值可知，就可用隐变量估计来解决，否则使用EM算法
- EM算法根据当前假设$<\mu_1 \ldots \mu_k>$，不断地再估计隐藏变量$z_{ij}$的期望值，然后用这些隐藏变量的期望值重新计算极大似然假设
- 先将假设初始化为$h=<\mu_1, \mu_2>$
  - 计算每个隐藏变量$z_{ij}$的期望值$E[z_{ij}]$，假定当前假设$h=<\mu_1, \mu_2>$成立
  - 计算一个新的极大似然假设$h'=<\mu'_1, \mu'_k>$，假定每个隐藏变量$z_{ij}$所取值是第一步得到的期望值$E[z_{ij}]$。将假设替换为$h'=<\mu'_1, \mu'_2>$，然后循环

# 两个步骤的计算式

- $E[z_{ij}]$正是实例$x_i$由第j个正态分布生成的概率

$$E[z_{ij}] = \frac{p(x = x_i \mid \mu = \mu_j)}{\sum_{n=1}^{2} p(x = x_i \mid \mu = \mu_n)}$$

$$= \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^{2} e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}}$$

- 第二步，使用第一步得到的$E[z_{ij}]$来导出一新的极大似然假设

$$\mu_j = \frac{\sum_{i=1}^{m} E[z_{ij}]x_i}{\sum_{i=1}^{m} E[z_{ij}]}$$

# 两个步骤的计算式（2）

- EM算法的要点：当前的假设用于估计未知变量，而这些变量的期望值再被用于改进假设

- 可以证明：算法的每一次循环中，EM算法能使似然P(D|h)增加，算法收敛到一个局部最大似然假设

# K均值算法的推导

- 问题框架
  - 要估计k个正态分布的均值$\theta=<\mu_1 \ldots \mu_k>$
  - 观察到的数据是$X=\{<x_i>\}$
  - 隐藏变量$Z=\{<z_{i1},\ldots,z_{ik}>\}$表示k个正态分布中哪一个生成$x_i$
- 用于K均值问题的表达式Q(h'|h)的推导
  - 单个实例的概率

$$p(y_i \mid h') = p(x_i, z_{i1},\ldots, z_{ik} \mid h') = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(x_i-\mu_j')^2}$$

# K均值算法的推导（2）

- 所有实例的概率的对数

$$\ln P(Y \mid h') = \ln \prod_{i=1}^{m} p(y_i \mid h')$$

$$= \sum_{i=1}^{m} \ln p(y_i \mid h')$$

$$= \sum_{i=1}^{m} \left( \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{j=1}^{k} z_{ij} (x_i - \mu_j')^2 \right)$$

- 计算期望值

$$E[\ln P(Y \mid h')] = E\left[ \sum_{i=1}^{m} \left( \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{j=1}^{k} z_{ij} (x_i - \mu_j')^2 \right) \right]$$

$$= \sum_{i=1}^{m} \left( \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \sum_{j=1}^{k} E[z_{ij}] (x_i - \mu_j')^2 \right)$$

# K均值算法的推导（3）

- 求使Q函数最大的假设

$$\arg\max_{h'} Q(h'|h) = \arg\max_{h'} \sum_{i=1}^{m}\left( \ln\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} E\left[z_{ij}\right]\left(x_i - \mu_j^{'}\right)^2 \right)$$

$$= \arg\max_{h'} \sum_{i=1}^{m}\sum_{j=1}^{k} E\left[z_{ij}\right]\left(x_i - \mu_j^{'}\right)^2$$

- 解上式得到

$$\mu_j \leftarrow \frac{\sum_{i=1}^{m} E\left[z_{ij}\right] x_i}{\sum_{i=1}^{m} E[z_{ij}]}$$

- 另外

$$E[zij] \leftarrow \frac{e^{-\frac{1}{2\sigma^2}(x_i-\mu_j)2}}{\sum_{n=1}^{k} e^{-\frac{1}{2\sigma^2}(x_i-\mu_n)^2}}$$