贝叶斯学习

Outline

Topic

贝叶斯学习方法的特性：

- 观察到的每个训练样例可以增量式地降低或升高某假设的估计概率。这提供了一种比其他算法更合理的学习途径。其他算法会在某个假设与任一样例不一致时完全去掉该假设。
- 先验知识可以与观察数据一起决定假设的最终概率。在贝叶斯学习中，先验知识的形式可以是（1）每个候选假设的先验概率（2）每个可能假设在可观察数据上的概率分布。
- 贝叶斯方法可允许假设做出不确定性的预测。（比如这样的假设：这一肺炎病人有 93\
- 新的实例分类可由多个假设一起作出预测，以它们的概率为权重。
- 即使在贝叶斯方法计算复杂度较高时，它们仍可做为一个最优的决策的标准衡量其他方法。

贝叶斯公式

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

- $P(h)$ = 还没有训练数据前，假设 h 的概率。
- $P(D)$ = 训练数据 D 的先验概率
- $P(h|D)$ = D 时 h 成立的概率。
- $P(D|h)$ = 给定假设 h 时观察到数据 D 的概率

最大后验 (Maximum a posteriori,MAP)

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

通常，学习器考虑候选假设集合 H 并在其中寻找给定数据 D 时可能性最大的假设 $h \in H$ 。
这样的具有最大可能性的假设被称为极大后验（maximum a posteriori, MAP）假设：

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(h|D) \\ &= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \arg\max_{h \in H} P(D|h)P(h) \end{aligned}$$

极大似然（Maximum likelihood，ML）

假定 $P(h_i) = P(h_j)$ 则可进一步简化，选取极大似然假设：

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

基本公式

- 乘法公式 (Product rule)：两事件 A 和 B 的交的概率 $P(A \wedge B)$

$$P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$$

- 加法公式 (Sum Rule)：两事件 A 和 B 的并的概率 $P(A \vee B)$

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

- 全概率公式 (Theorem of total probability)：如果事件 $A_1, \ldots, A_n$ 互斥且 $\sum_{i=1}^{n} P(A_i) = 1$，则：

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

Topic

概念

- 假定学习器考虑的是定义在实例空间 X 上的有限的假设空间 H，
- 任务是学习某个目标概念 $c: X \to \{0, 1\}$。
- 为简化讨论，假定实例序列 $\langle x_1, \ldots, x_m \rangle$ 是固定不变的，
- 训练数据 D 可被简单地写作目标函数值序列：$D = \langle c(x_1), \ldots, c(x_m) \rangle$。

Brute-Force MAP 学习算法

◇ 对于 H 中每个假设 h，计算后验概率：

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

◇ 输出有最高后验概率的假设 h_{MAP}

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

Brute-Force MAP 学习算法（续）

选择 P(h) 和 P(D|h) 的概率分布，以描述该学习任务的先验知识：

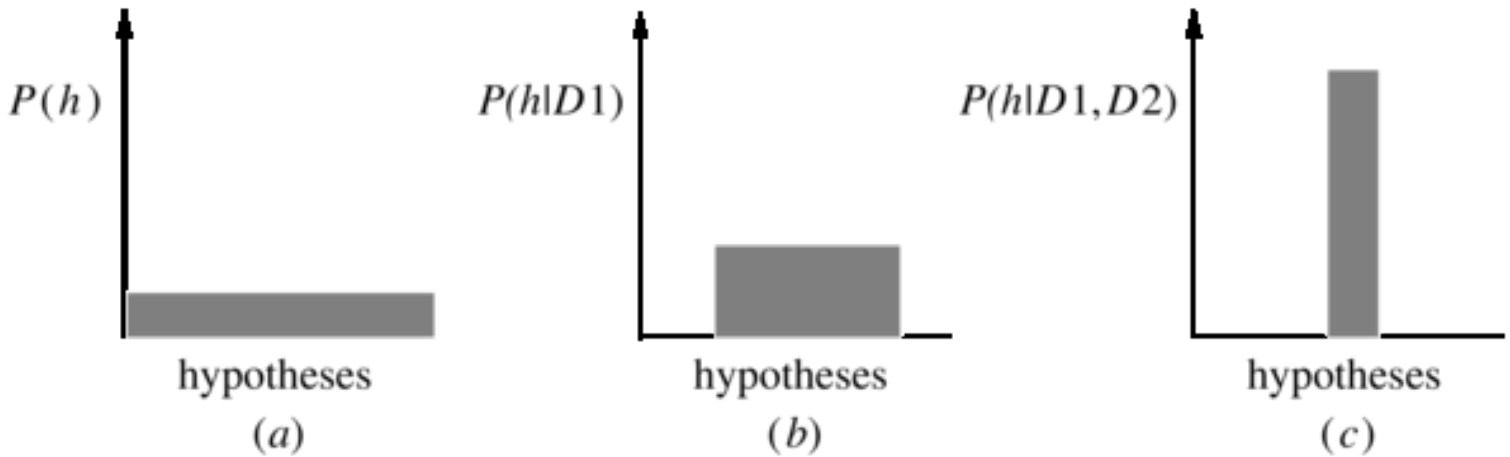
- 训练数据 D 是无噪声的（即 $d_i = c(x_i)$ ）；
- 目标概念 c 包含在假设空间 H 中；
- 没有任何理由认为某假设比其他的假设的可能性大。
- 选取 $P(D|h)$:
  - $P(D|h) = 1$, 若 h 与 D 一致
  - $P(D|h) = 0$, 其它情况
- 选取 $P(h)$ 服从均匀分布
  - $P(h) = \frac{1}{|H|}$, 对 H 中的所有 h

Brute-Force MAP 学习算法（续）

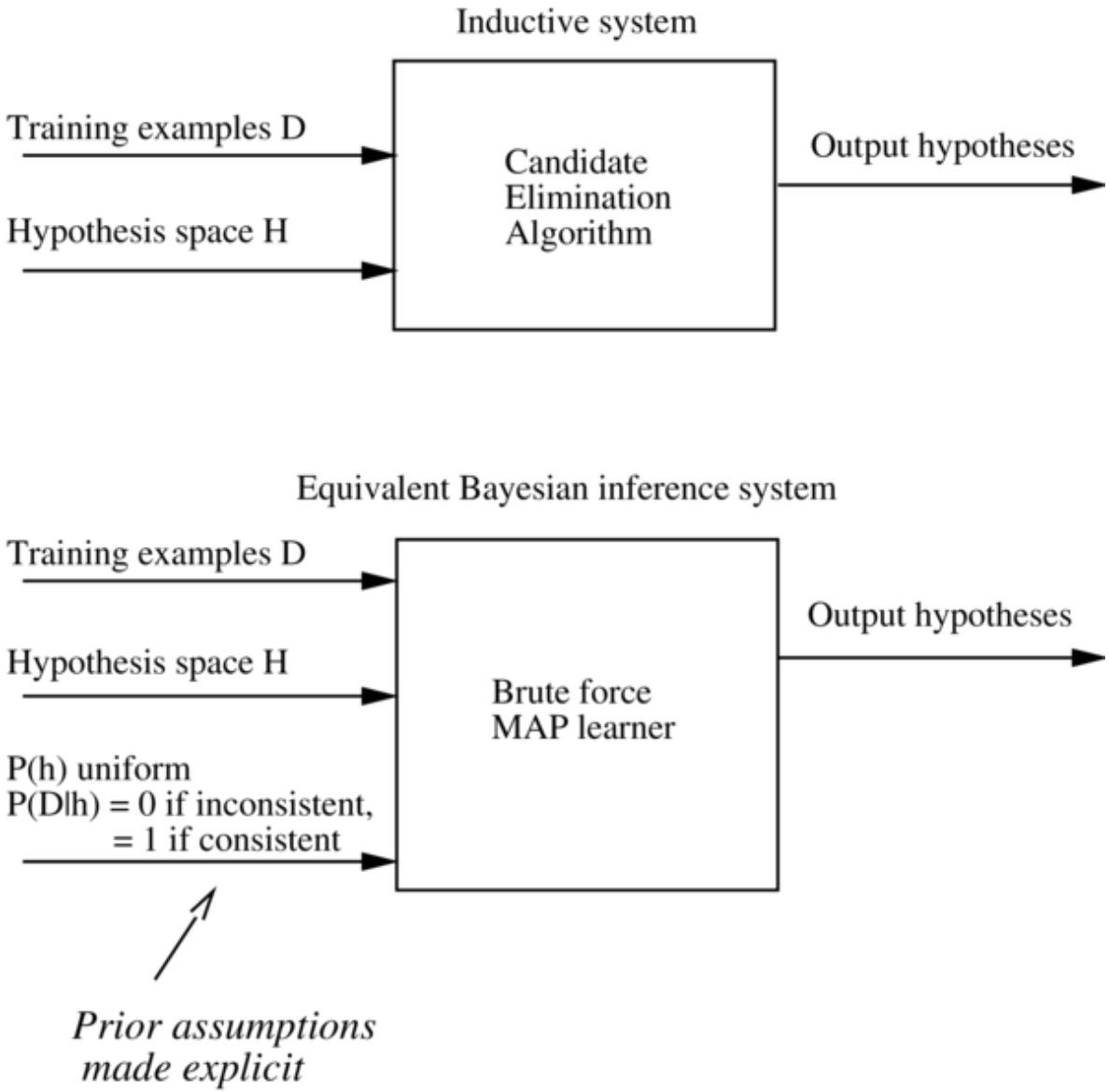
可得：

$$
\begin{aligned}
P(h|D) &= \frac{P(D|h)P(h)}{P(D)} \\
&= \frac{P(D|h)P(h)}{\sum_{h_i \in H} P(D|h_i)P(h_i)} \\
&= \frac{P(D|h) \cdot \frac{1}{|H|}}{\sum_{h_i \in VS_{H,D}} 1 \times \frac{1}{|H|} + \sum_{h_i \notin VS_{H,D}} 0 \times \frac{1}{|H|}} \\
&= \frac{P(D|h)}{|VS_{H,D}|} \\
&= \begin{cases} \frac{1}{|VS_{H,D}|} & \text{if } h \text{ is consistent with } D \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

贝叶斯法则



贝叶斯学习

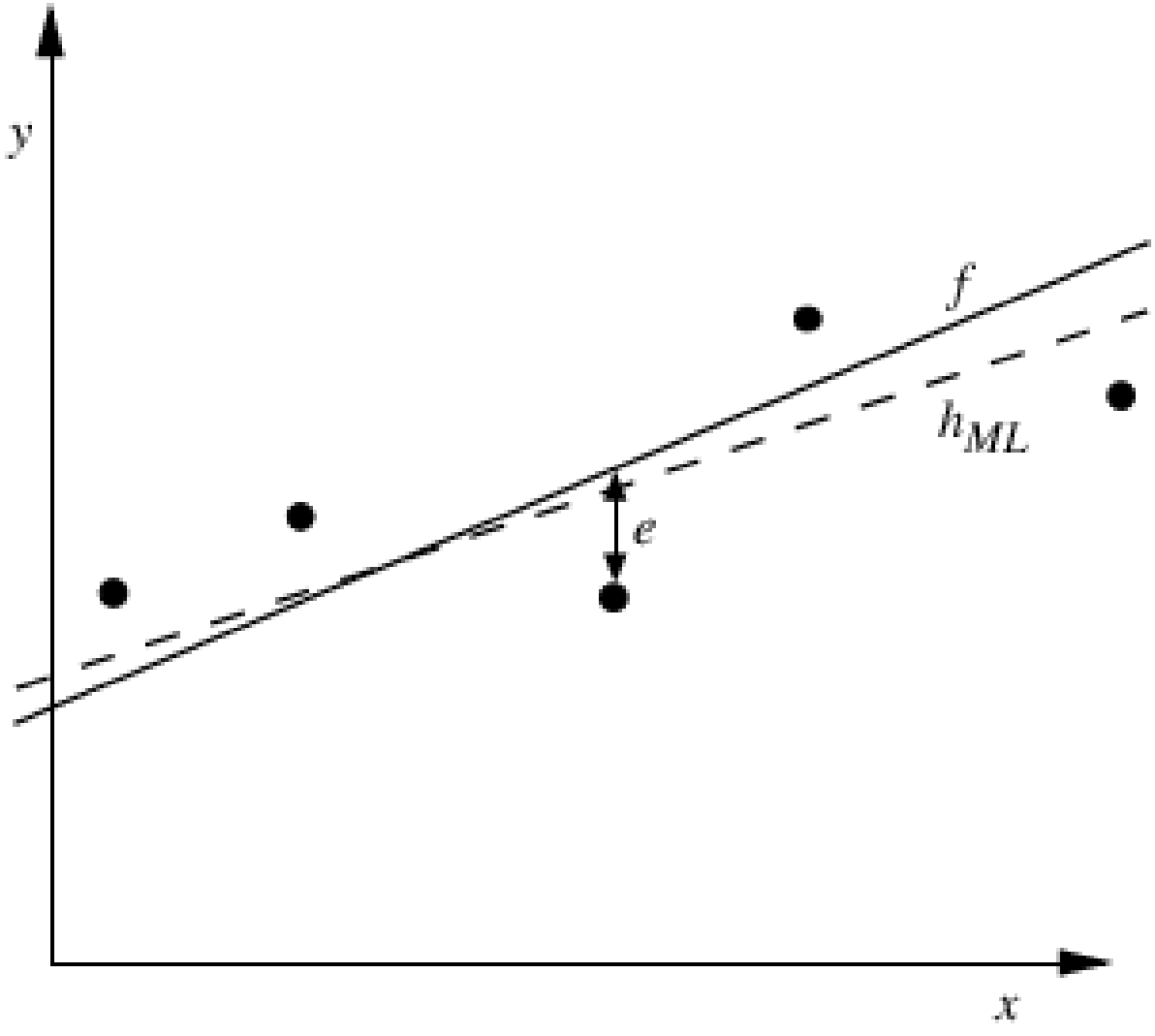

Inductive system
Training examples D
Hypothesis space H
Candidate Elimination Algorithm
Output hypotheses
Equivalent Bayesian inference system
Training examples D
Hypothesis space H
P(h) uniform
P(D|h) = 0 if inconsistent,
= 1 if consistent
Brute force MAP learner
Output hypotheses
Prior assumptions made explicit

MAP 假设和一致学习器

- 在给定条件下，与 D 一致的每个假设都是 MAP 假设。
  - 根据这一结论可直接得到一类普遍的学习器，称为一致学习器。
  - 某学习算法被称为一致学习器，说明它输出的假设在训练例上有零错误率。
- 假定 H 上有均匀的先验概率（即 $P(h_i) = P(h_j)$，对所有的 i, j），
- 且训练数据是确定性的和无噪声的 (即当 D 和 h 一致时，$P(D|h) = 1$，否则为 0）时，
- 任意一致学习器将输出一个 MAP 假设。
- 例如第 2 章讨论的 Find-S 概念学习算法:
  - Find-S 按照特殊到一般的顺序搜索假设空间 H，
  - 并输出一个极大特殊性的一致假设，
  - 可知在上面定义的 $P(h)$ 和 $P(D|h)$ 概率分布下，它输出 MAP 假设。

Topic

学习实值函数



考虑实值函数 f

◇ 训练样例 $\langle x_i, d_i \rangle$, 其中

$$d_i = f(x_i) + e_i$$

⋆ e_i 是随机变量，与 x_i 独立，服从零均值高斯分布

◇ 最大似然估计 h_{ML} :

$$h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

推导：

$$
\begin{aligned}
h_{ML} &= \arg\max_{h\in H} p(D|h) \\
&= \arg\max_{h\in H} \prod_{i=1}^{m} p(d_i|h) \\
&= \arg\max_{h\in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(\frac{d_i-h(x_i)}{\sigma})^2}
\end{aligned}
$$

用自然对数替换，得：

$$
\begin{aligned}
h_{ML} &= \arg\max_{h\in H} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\
&= \arg\max_{h\in H} \sum_{i=1}^{m} -\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\
&= \arg\max_{h\in H} \sum_{i=1}^{m} -\left(d_i - h(x_i)\right)^2 \\
&= \arg\min_{h\in H} \sum_{i=1}^{m} \left(d_i - h(x_i)\right)^2
\end{aligned}
$$

用于预测概率的极大似然假设

考虑从数据中预测概率

- 训练样例 $\langle x_i, d_i \rangle$, 其中 d_i 为 1 或 0
- 训练神经网络根据给定的 x_i 输出一个概率

$$h_{ML} = \arg\max_{h \in H} \prod_{i=1}^{m} h(x_i)^{d_i} (1 - h(x_i))^{1-d_i}$$

$$= \arg\max_{h \in H} \sum_{i=1}^{m} d_i \ln h(x_i) + (1 - d_i) \ln(1 - h(x_i))$$

- sigmoid 单元的权值更新:

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk}$$

其中:

$$\Delta w_{jk} = \eta \sum_{i=1}^{m} (d_i - h(x_i))\ x_{ijk}$$

Topic

奥坎姆剃刀（Occam's razor）

- “为观察到的数据选择最短的解释”。（优先选择短的假设）
- 最小描述长度准则（Minimum Description Length, MDL）：
  - 优先选择最小化

$$h_{MDL} = \arg\min_{h \in H} L_{C_1}(h) + L_{C_2}(D|h)$$

    的假设 h
  - 其中 $L_C(x)$ 是在编码 C 下 x 的描述长度

示例：

- $H =$ 决策树
- $D =$ 训练数据
- $L_{C_1}(h)$ 是 h 的编码长度
- $L_{C_2}(D|h)$ 给定 h 时，D 的编码长度
- 当样例被 h 完美分类时，$L_{C_2}(D|h) = 0$
- h_{MDL} 考虑了树的大小与训练误差

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(D|h)P(h) \\ &= \arg\max_{h \in H} \log_2 P(D|h) + \log_2 P(h) \\ &= \arg\min_{h \in H} -\log_2 P(D|h) - \log_2 P(h) \end{aligned} \quad (1)$$

由信息论可得：

针对以概率 p 发生的事件，最优 (最短期望编码长度) 编码是 $-\log_2 p$ 位.

- ◇ $-\log_2 P(h)$ 是 h 的最优编码长度
- ◇ $-\log_2 P(D|h)$ 是给定 h 后 D 的最优编码长度

→ 优先选择最小化

$$length(h) + length(misclassifications)$$

的假设

Topic

新实例的最大可能分类

- 给定训练数据 D，最可能的假设是什么?(h_{MAP})
- 给定训练数据 D，对新实例 x 的最可能分类是什么？

考虑三个假设：

- $P(h_1|D) = .4,\ P(h_2|D) = .3,\ P(h_3|D) = .3$

对于新的实例 x,

- $h_1(x) = +,\ h_2(x) = -,\ h_3(x) = -$
- x 的最大可能分类是什么？

贝叶斯最优分类器

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i) P(h_i|D)$$

示例:

$$P(h_1|D) = .4, \quad P(-|h_1) = 0, \quad P(+|h_1) = 1$$
$$P(h_2|D) = .3, \quad P(-|h_2) = 1, \quad P(+|h_2) = 0$$
$$P(h_3|D) = .3, \quad P(-|h_3) = 1, \quad P(+|h_3) = 0$$

因此

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) = .4$$
$$\sum_{h_i \in H} P(-|h_i)P(h_i|D) = .6$$

与

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = -$$

Topic

GIBBS 算法

- 虽然贝叶斯最优分类器能从给定训练数据中获得最好的性能，应用此算法的开销可能很大。
- 原因在于它要计算 H 中每个假设的后验概率，然后合并每个假设的预测，以分类新实例。
- 一个替代的、非最优的方法是 Gibbs 算法，定义如下：
  当有一待分类新实例时，Gibbs 算法简单地按照当前的后验概率分布，使用一随机抽取的假设。

Gibbs 算法:

- 按照 H 上的后验概率分布 $P(h|D)$ ，从 H 中随机选择假设 h 。
- 使用 h 来预言下一实例 x 的分类。
- 可证明在一定条件下 Gibbs 算法的误分类率的期望值最多为贝叶斯最优分类器的两倍。
- 更精确地讲，期望值是在随机抽取的目标概念上作出，抽取过程按照学习器假定的先验概率。
- 在此条件下，Gibbs 算法的错误率期望值最差为贝叶斯分类器的两倍。

$$E[error_{Gibbs}] \leq 2E[error_{BayesOptimal}]$$

概念学习问题分析：

- 如果学习器假定 H 上有均匀的先验概率，而且如果目标概念实际上也按该分布抽取
- 那么当前变型空间中随机抽取的假设对下一实例分类的期望误差最多为贝叶斯分类器的两倍。

Topic

朴素贝叶斯分类器（Naive Bayes Classifier ）

- 贝叶斯学习方法中实用性很高的一种为朴素贝叶斯学习器，常被称为朴素贝叶斯分类器（naive Bayes classifier）。在某些领域内其性能可与神经网络和决策树学习相当。
  - 何时使用：
    - 中等或大训练集
    - 描述实例的属性在给定类别后条件独立
- 已成功应用于
  - 诊断
  - 文本分类

描述

- 假定目标函数 $f: X \to V$, 其中每个实例 x 由属性 $\langle a_1, a_2 \ldots a_n \rangle$ 描述.
- $f(x)$ 的最大可能值为:

$$\begin{aligned} v_{MAP} &= \arg\max_{v_j \in V} P(v_j | a_1, a_2 \ldots a_n) \\ v_{MAP} &= \arg\max_{v_j \in V} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)} \\ &= \arg\max_{v_j \in V} P(a_1, a_2 \ldots a_n | v_j) P(v_j) \end{aligned}$$

- Naive Bayes 假定:

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

- 可得:

Naive Bayes classifier: $v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$

算法

- Naive_Bayes_Learn(*examples*) 对每个目标值 v_j
  - $\hat{P}(v_j) \leftarrow$ 估计 $P(v_j)$
  - 对每个属性 a 的每个可能取值 a_i
    - $\hat{P}(a_i|v_j) \leftarrow$ 估计 $P(a_i|v_j)$
- Classify_New_Instance(x)

$$v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

示例

- PlayTennis 中，新实例：

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

- 不同目标值的概率可以基于这 14 个训练样例的频率很容易地估计出：
  - $P(PlayTennis = yes) = 9/14 = 0.64$
  - $P(PlayTennis = no) = 5/14 = 0.36$
- 相似地，可以估计出条件概率，例如对于 Wind=Strong 有：
  - $P(Wind = strong|PlayTennis = yes) = 3/9 = 0.33$
  - $P(Wind = strong|PlayTennis = no) = 3/5 = 0.60$
- 计算:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

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

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

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

Naive Bayes: Subtleties

- 通常不满足独立性假定

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

- 但还是会有很好的表现。注意：不需要估计到的后验概率 $\hat{P}(v_j|x)$ 是正确的，只需要：

$$\arg\max_{v_j \in V} \hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = \arg\max_{v_j \in V} P(v_j) P(a_1 \ldots, a_n | v_j)$$

当目标值为 v_j 的所有训练实例都没有属性值 a_i? 时

$$\hat{P}(a_i|v_j) = 0, \text{ and...}$$

$$\hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = 0$$

典型的解决方法是对 $\hat{P}(a_i|v_j)$ 进行贝叶斯估计

$$\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$$

其中：

- n 是 $v = v_j$ 的训练样例的数量
- n_c 是 $v = v_j$ 且 $a = a_i$ 的样例数量
- p 是对 $\hat{P}(a_i|v_j)$ 的先验估计
- m 是对先验的权重 (等效样本大小)

学习分类文本

- 学习将文本按兴趣分类
- 学习将网页按主题分类

目标概念：$\mathit{Interesting?} : \mathit{Document} \rightarrow \{+, -\}$

将文档表示为单词向量

- one attribute per word position in document
- Learning: Use training examples to estimate
  - $P(+)$
  - $P(-)$
  - $P(doc|+)$
  - $P(doc|-)$

朴素贝叶斯条件独立假定

$$P(doc|v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k|v_j)$$

其中 $P(a_i = w_k|v_j)$ 是给定 v_j 时，位置 i 的单词是 w_k 的概率。
另一假定: $P(a_i = w_k|v_j) = P(a_m = w_k|v_j), \forall i, m$

算法流程

Learn_naive_Bayes_text(*Examples*, V)

- Examples 为一组文本文档以及它们的目标值。
- V 为所有可能目标值的集合。
- 此函数作用是学习概率项 $P(w_k|v_j)$ ，
- 它描述了从类别 v_j 中的一个文档中随机抽取的一个单词为英文单词 w_k 的概率。该函数也学习类别的先验概率 $P(v_j)$ 。

算法流程 (续)

- 收集 Examples 中所有的单词、标点符号以及其他记号
  - *Vocabulary* ← 在 Examples 中任意文本文档中出现的所有单词及记号的集合
- 计算所需要的概率项 $P(v_j)$ 和 $P(w_k|v_j)$
  - 对 V 中每个目标值 v_j
    - $docs_j$ ← Examples 中目标值为 v_j 的文档子集
    - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
    - $Text_j$ ← 将 $docs_j$ 中所有成员连接起来建立的单个文档
    - $n \leftarrow$ 在 $Text_j$ 中不同单词位置的总数 (重复单词多次计算)
    - 对 *Vocabulary* 中每个单词 w_k
      - $n_k \leftarrow$ 单词 w_k 出现在 $Text_j$ 中的次数
      - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

算法流程 (续)

Classify_naive_Bayes_text(*Doc*)

- 对文档 *Doc* 返回其估计的目标值。a_i 代表在 *Doc* 中的第 i 个位置上出现的单词。
  - *positions* $\leftarrow$ 在 *Doc* 中包含的能在 *Vocabulary* 中找到的记号的所有单词位置
  - 返回

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i \in positions} P(a_i|v_j)$$

Twenty NewsGroups

Given 1000 training documents from each group
Learn to classify new documents according to which newsgroup it came from

comp.graphics	misc.forsale
comp.os.ms-windows.misc	rec.autos
comp.sys.ibm.pc.hardware	rec.motorcycles
comp.sys.mac.hardware	rec.sport.baseball
comp.windows.x	rec.sport.hockey
alt.atheism	sci.space
soc.religion.christian	sci.crypt
talk.religion.misc	sci.electronics
talk.politics.mideast	sci.med
talk.politics.misc	
talk.politics.guns	

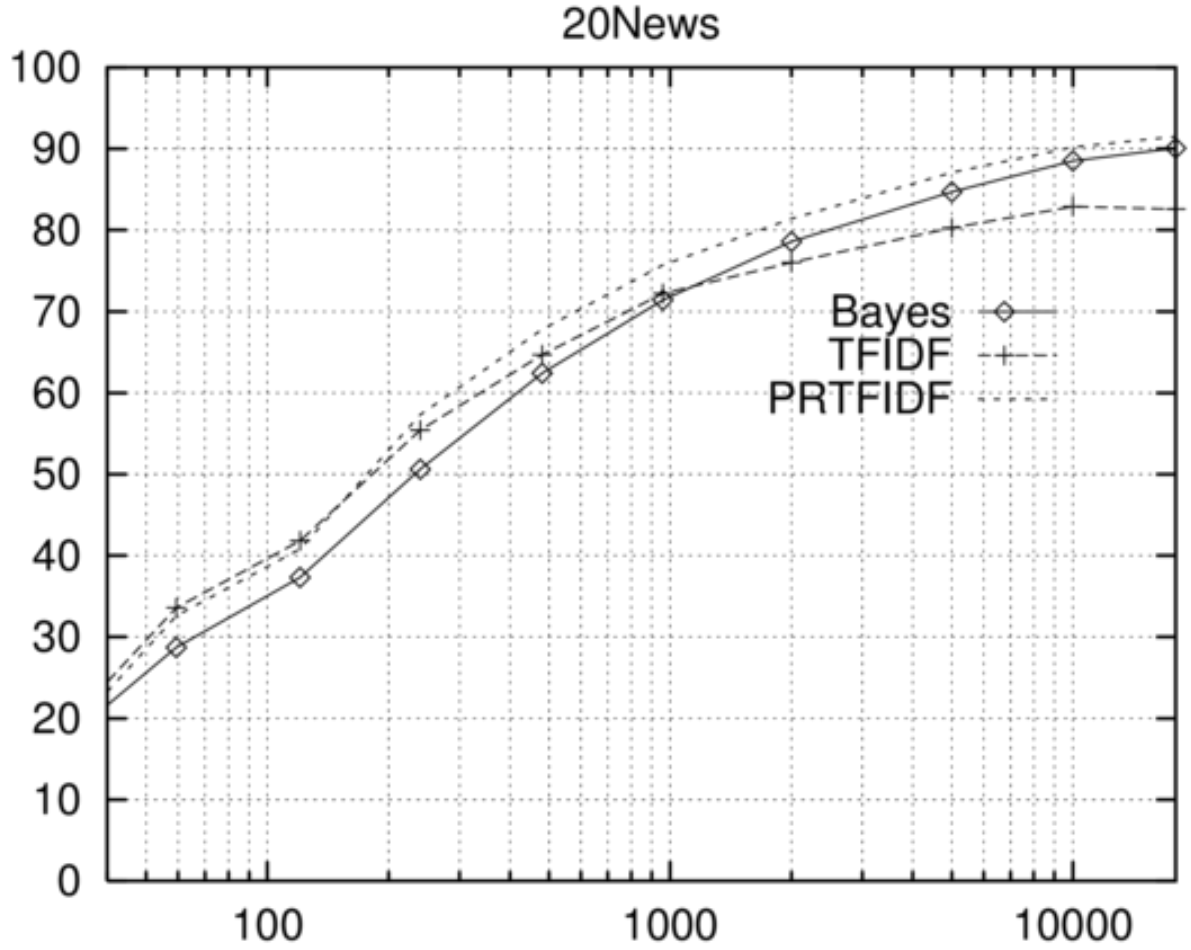
Naive Bayes: 89\

Article from rec.sport.hockey

```
Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.edu!ogicse!uwm.
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opinion)...
Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most
obvious candidate for pleasant surprise is Alex
Zhitnik. He came highly touted as a defensive
defenseman, but he's clearly much more than that.
Great skater and hard shot (though wish he were
more accurate). In fact, he pretty much allowed
the Kings to trade away that huge defensive
liability Paul Coffey. Kelly Hrudey is only the
biggest disappointment if you thought he was any
good to begin with. But, at best, he's only a
mediocre goaltender. A better choice would be
Tomas Sandstrom, though not through any fault of
his own, but because some thugs in Toronto decided
```

Learning Curve for 20 Newsgroups



Accuracy vs. Training set size (1/3 withheld for test)

Topic

贝叶斯信念网 (Bayesian Belief Networks)

- 贝叶斯置信网描述的是一组变量所遵从的概率分布，它通过一组条件概率来指定一组条件独立性假定。
- 朴素贝叶斯分类器假定所有变量在给定目标变量值时为条件独立的，与此不同，贝叶斯置信网中可表述应用到变量的一个子集上的条件独立性假定。
- 因此，贝叶斯置信网提供了一种中间的方法，它比朴素贝叶斯分类器中条件独立性的全局假定的限制更少，又比在所有变量中计算条件依赖更可行。

条件独立

定义: *若给定 Z 的值，X 的概率分布独立于 Y 的值，即:*

$$(\forall x_i, y_j, z_k)\ P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

则称 X 在给定 Z 时条件独立于 Y. 记作:

$$P(X|Y, Z) = P(X|Z)$$

示例:

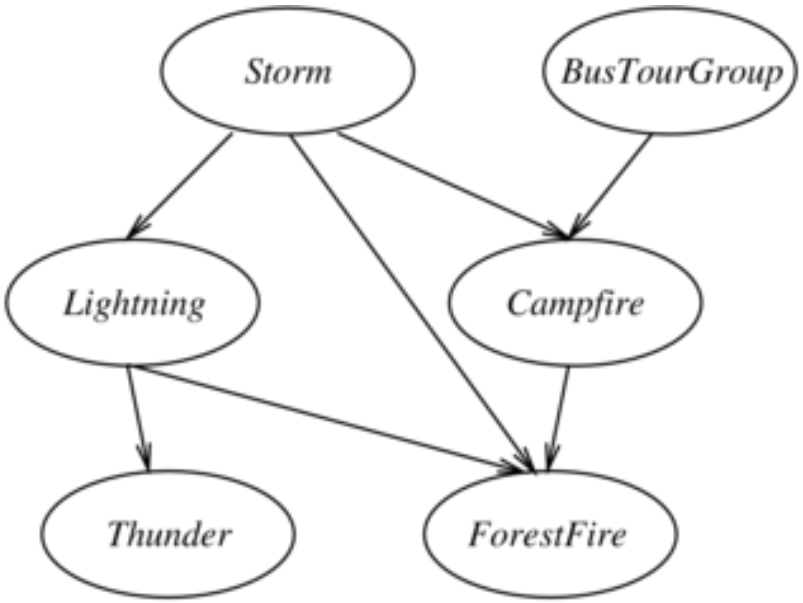
给定 *Lightning* 则 *Thunder* 条件独立于 *Rain*,

$$P(Thunder|Rain, Lightning) = P(Thunder|Lightning)$$

Naive Bayes 推导中使用了条件独立:

$$\begin{aligned} P(X, Y|Z) &= P(X|Y, Z)P(Y|Z) \\ &= P(X|Z)P(Y|Z) \end{aligned}$$

网络



	S,B	$S,\neg B$	$\neg S,B$	$\neg S,\neg B$
C	0.4	0.1	0.8	0.2
$\neg C$	0.6	0.9	0.2	0.8

说明

- 贝叶斯网表示联合概率分布的方法是指定一组条件独立性假定（有向无环图），以及一组局部条件概率集合。
- 联合空间中每个变量在贝叶斯网中表示为一结点。
- 对每一变量需要两种类型的信息。首先，网络弧表示断言“此变量在给定其立即前驱时条件独立于其非后继”。

表示联合概率：

- 例如：$P(Storm, BusTourGroup, \ldots, ForestFire)$
- 对网络变量的元组 $(Y_1, \ldots, Y_n)$ 取值 $(y_1, \ldots, y_n)$ 的联合概率：

$$P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i | Parents(Y_i))$$

其中 $Parents(Y_i)$ 表示网络中 Y_i 的立即前驱的集合。注意 $P(y_i|Parents(Y_i))$ 的值等于与结点 Y_i 关联的条件概率表中的值。

贝叶斯网络推理

- 可以用贝叶斯网在给定其他变量的观察值时推理出某些目标变量（如 ForestFire）的值。
- 由于所处理的是随机变量，所以一般不会赋予目标变量一个确切的值。
- 真正需要推理的是目标变量的概率分布，它指定了在给与其他变量的观察值条件下，目标变量取每一可能值的概率。
- 在网络中所有其他变量都确切知道了以后，这一推理步骤是很简单的。
- 在更通常的情况下，我们希望在知道一部分变量的值（比如 Thunder 和 BusTourGroup 为仅有可用的观察值）时获得某变量的概率分布（如 ForestFire）。
- 一般地，贝叶斯网络可用于在知道某些变量的值或分布时计算网络中另一部分变量的概率分布。

学习贝叶斯网络

- 网络结构预先给出，或可由训练数据中推得。
- 所有的网络变量可以直接从每个训练样例中观察到，或某些变量不能观察到。
- 在网络结构的预先已知，并且变量可以从训练样例中完全获得时，通过学习得到条件概率表就比较简单了。只需要象在朴素贝叶斯分类器中那样估计表中的条件概率项。

若网络结构已知，但只有一部分变量值能在数据中观察到。

- 这一问题在某种程度上类似于在人工神经网络中学习隐藏单元的权值，其中输入和输出结点值由训练样例给出，但隐藏单元的值未指定。
- 梯度上升过程可以学习条件概率表中的项。梯度上升过程搜索一个假设空间，它对应于条件概率表中所有可能的项。
- 在梯度上升中最大化的目标函数是给定假设 h 下观察到训练数据 D 的概率 $P(D|h)$ 。按照定义，它对应于对表项搜索极大似然假设。

梯度上升算法

- 令 w_{ijk} 代表一个条件概率表的一个表项。确切地讲，令 w_{ijk} 为在给定父结点 U_i 取值 u_{ik} 时，网络变量 Y_i 值为 y_{ij} 的概率。

$$w_{ijk} = P(Y_i = y_{ij} | Parents(Y_i) = \text{the list } u_{ik} \text{ of values})$$

若 $Y_i = Campfire$ 则 u_{ik} 可能是
$\langle Storm = T, BusTourGroup = F \rangle$ 例如，若 w_{ijk} 为图中条件概率表中最右上方的表项，那么 Y_i 为变量 $Campfire$，U_i 是其父结点的元组 $< Storm, BusTourGroup >$，$y_{ij} = True$，并且 $u_{ik} =< False, False >$。

梯度上升算法 (续)

- 通过 $\ln P(D|h)$ 的梯度来使 $P(D|h)$ 最大化。
- 重复执行梯度上升
  - 使用训练数据 D 更新所有 w_{ijk}

$$w_{ijk} \leftarrow w_{ijk} + \eta \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik}|d)}{w_{ijk}}$$

  - 重新归一化 w_{ijk}，保证
    - $\sum_j w_{ijk} = 1$
    - $0 \leq w_{ijk} \leq 1$

More on Learning Bayes Nets

可使用 EM 算法

- 假定 h 计算未观测到的变量概率
- 计算新的 w_{ijk} 最大化 $E[\ln P(D|h)]$，其中 D 已包含观测到的与未观测到（但计算出了概率）的变量

当结构未知时

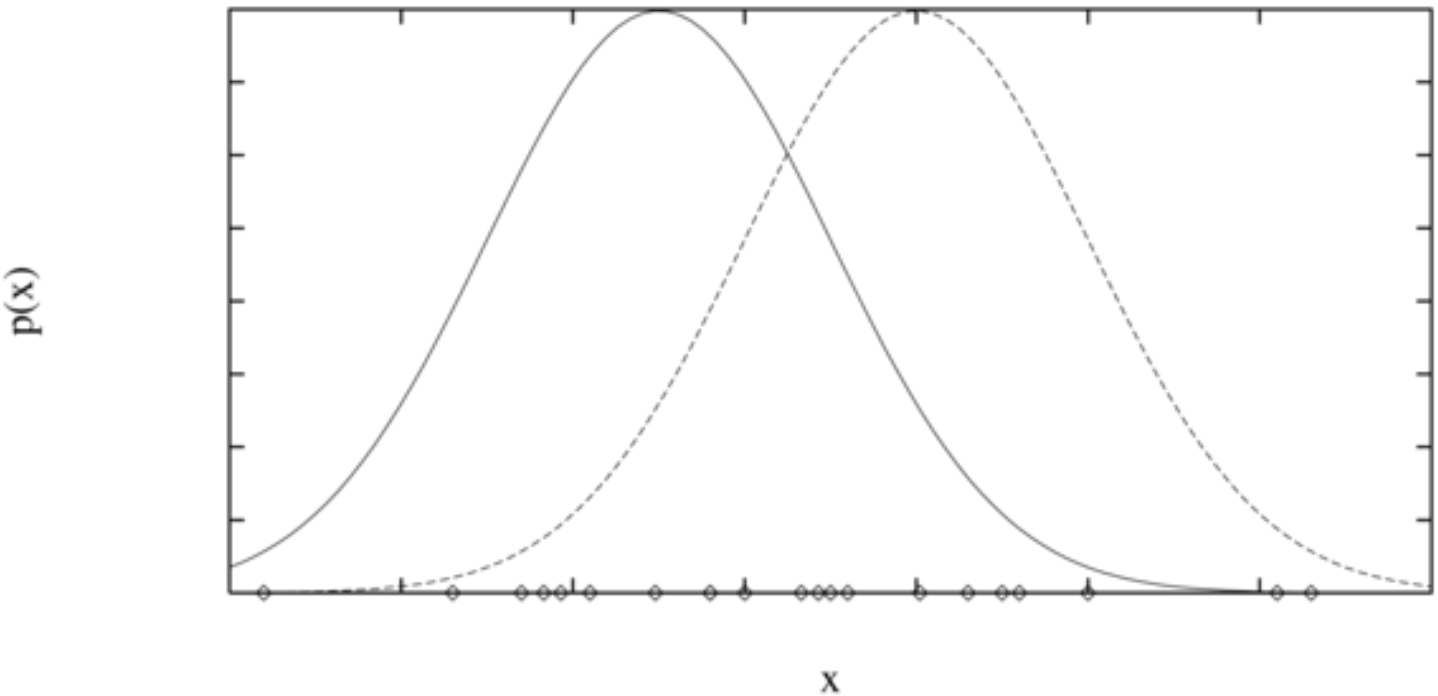
- 可使用贪婪搜索增/删结点与边

Topic

Expectation Maximization (EM)

- 观测到部分数据
- 实例的部分属性未知
- 无监督聚类
- 训练 Bayesian Belief Networks
- 学习 Hidden Markov Models

Generating Data from Mixture of k Gaussians



每个实例 x 按如下方式产生：

- 按均匀分布选取 k 个高斯分布之一
- 按此高斯分布随机产生一个实例

EM for Estimating k Means

已知:

- 从 k 个高斯分布产生的实例 x
- k 个高斯分布的均值 $\langle \mu_1, \ldots, \mu_k \rangle$ 未知
- 不知实例 x_i 从哪个高斯分布产生

求解:

- $\langle \mu_1, \ldots, \mu_k \rangle$ 的最大似然估计

将实例完整描述为 $y_i = \langle x_i, z_{i1}, z_{i2} \rangle$, 其中

- z_{ij} 为 1，当 x_i 由第 j 个高斯分布产生
- x_i 可观测
- z_{ij} 不可观测

EM Algorithm:

随机选取初始值 $h = \langle \mu_1, \mu_2 \rangle$, 然后迭代:

- E step:
  - 计算每个隐藏变量 z_{ij} 的期望值 $E[z_{ij}]$ ，假定当前假设 $h = \langle \mu_1, \mu_2 \rangle$ 成立

$$
\begin{aligned}
E[z_{ij}] &= \frac{p(x = x_i | \mu = \mu_j)}{\sum_{n=1}^{2} p(x = x_i | \mu = \mu_n)} \\
&= \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^{2} e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}}
\end{aligned}
$$

EM Algorithm:

- M step:
  - ⋆ 计算一个新的极大似然假设 $h' = \langle \mu_1', \mu_2' \rangle$ ，
  - ⋆ 假定由每个隐藏变量 z_{ij} 所取的值为 E step 中得到的期望值 $E[z_{ij}]$ ，
  - ⋆ 然后将假设 $h = \langle \mu_1, \mu_2 \rangle$ 替换为新的假设 $h' = \langle \mu_1', \mu_2' \rangle$ ，

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

EM Algorithm

- ◇ Converges to local maximum likelihood h
- ◇ and provides estimates of hidden variables z_{ij}
- ◇ In fact, local maximum in $E[\ln P(Y|h)]$
  - ⋆ Y is complete (observable plus unobservable variables) data
  - ⋆ Expected value is taken over possible values of unobserved variables in Y

General EM Problem

已知:

- 观测数据 $X = \{x_1, \ldots, x_m\}$
- 未观测数据 $Z = \{z_1, \ldots, z_m\}$
- 参数化概率分布 $P(Y|h)$, 其中 $Y = \{y_1, \ldots, y_m\}$ 是数据 $y_i = x_i \cup z_i$ ， h 是参数

求解:

- (局部) 最大化 $E[\ln P(Y|h)]$ 的 h

用于:

- Train Bayesian belief networks
- Unsupervised clustering (e.g., k means)
- Hidden Markov Models

General EM Problem

定义似然函数 $Q(h'|h)$，使用观测到的 X 与当前参数 h 估计 Z，计算 $Y = X \cup Z$

$$Q(h'|h) \leftarrow E[\ln P(Y|h')|h, X]$$

EM Algorithm:

- Estimation (E) step: 使用当前假设 h 和观察到的数据 X 来估计 Y 上的概率分布以计算 $Q(h'|h)$。

$$Q(h'|h) \leftarrow E[\ln P(Y|h')|h, X]$$

- Maximization (M) step:} 将假设 h 替换为使 Q 函数最大化的假设 h'：

$$h \leftarrow \arg\max_{h'} Q(h'|h)$$