七、贝叶斯分类器

主讲教师：周志华

贝叶斯决策论 (Bayesian decision theory)

概率框架下实施决策的基本理论

给定 N 个类别，令 λ_{ij} 代表将第 j 类样本误分类为第 i 类所产生的损失，则基于后验概率将样本 $\boldsymbol{x}$ 分到第 i 类的条件风险为：

$$R(c_i \mid \boldsymbol{x}) = \sum_{j=1}^{N} \lambda_{ij} P(c_j \mid \boldsymbol{x})$$

贝叶斯判定准则 (Bayes decision rule)：

$$h^*(\boldsymbol{x}) = \arg\min_{c \in \mathcal{Y}} R(c \mid \boldsymbol{x})$$

- h^* 称为贝叶斯最优分类器(Bayes optimal classifier)，其总体风险称为贝叶斯风险 (Bayes risk)
- 反映了学习性能的理论上限

判别式 vs. 生成式

$P(c \mid \boldsymbol{x})$ 在现实中通常难以直接获得

从这个角度来看，机器学习所要实现的是基于有限的训练样本尽可能准确地估计出后验概率

两种基本策略：

判别式 (discriminative) 模型

思路：直接对 $P(c \mid \boldsymbol{x})$ 建模

代表：

- 决策树
- BP 神经网络
- SVM

生成式 (generative) 模型

思路：先对联合概率分布 $P(\boldsymbol{x}, c)$ 建模，再由此获得 $P(c \mid \boldsymbol{x})$

$$P(c \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x}, c)}{P(\boldsymbol{x})}$$

代表：贝叶斯分类器

注意：贝叶斯分类器 ≠ 贝叶斯学习 (Bayesian learning)

贝叶斯定理

$$P(c \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x}, c)}{P(\boldsymbol{x})}$$

Thomas Bayes
(1701?-1761)

根据贝叶斯定理，有

$$P(c \mid \boldsymbol{x}) = \frac{P(c)\, P(\boldsymbol{x} \mid c)}{P(\boldsymbol{x})}$$

先验概率（prior）
样本空间中各类样本所占的比例，可通过各类样本出现的频率估计（大数定律）

样本相对于类标记的类条件概率（class-conditional probability），亦称 似然（likelihood）

证据（evidence）因子，与类别无关

主要困难在于估计似然 $P(\boldsymbol{x} \mid c)$

极大似然估计

先假设某种概率分布形式，再基于训练样例对参数进行估计

假定 $P(\boldsymbol{x} \mid c)$具有确定的概率分布形式，且被参数 $\boldsymbol{\theta}_c$ 唯一确定，则任务就是利用训练集 D 来估计参数 $\boldsymbol{\theta}_c$

$\boldsymbol{\theta}_c$ 对于训练集 D 中第 c 类样本组成的集合 D_c 的似然(likelihood)为

$$P(D_c \mid \boldsymbol{\theta}_c) = \prod_{\boldsymbol{x} \in D_c} P(\boldsymbol{x} \mid \boldsymbol{\theta}_c)$$

连乘易造成下溢，因此通常使用对数似然 (log-likelihood)

$$LL(\boldsymbol{\theta}_c) = \log P(D_c \mid \boldsymbol{\theta}_c) = \sum_{\boldsymbol{x} \in D_c} \log P(\boldsymbol{x} \mid \boldsymbol{\theta}_c)$$

于是，$\boldsymbol{\theta}_c$ 的极大似然估计为 $\hat{\boldsymbol{\theta}}_c = \arg\max_{\boldsymbol{\theta}_c} LL(\boldsymbol{\theta}_c)$

估计结果的准确性严重依赖于所假设的概率分布形式是否符合潜在的真实分布

朴素贝叶斯分类器 (naïve Bayes classifier)

$$P(c \mid \boldsymbol{x}) = \frac{P(c)\,P(\boldsymbol{x} \mid c)}{P(\boldsymbol{x})}$$

主要障碍：所有属性上的联合概率难以从有限训练样本估计获得

组合爆炸；样本稀疏

基本思路：假定属性相互独立？

$$P(c \mid \boldsymbol{x}) = \frac{P(c)\,P(\boldsymbol{x} \mid c)}{P(\boldsymbol{x})} = \frac{P(c)}{P(\boldsymbol{x})} \prod_{i=1}^{d} P(x_i \mid c)$$

d 为属性数，x_i 为 $\boldsymbol{x}$ 在第 i 个属性上的取值

$P(\boldsymbol{x})$ 对所有类别相同，于是

$$h_{nb}(\boldsymbol{x}) = \arg\max_{c \in \mathcal{Y}} P(c) \prod_{i=1}^{d} P(x_i \mid c)$$

朴素贝叶斯分类器

- 估计 $P(c)$：$P(c) = \frac{|D_c|}{|D|}$

- 估计 $P(\boldsymbol{x}|c)$：

  - 对离散属性，令 D_{c,x_i} 表示 D_c 中在第 i 个属性上取值为 x_i 的样本组成的集合，则

$$P(x_i \mid c) = \frac{|D_{c,x_i}|}{|D_c|}$$

  - 对连续属性，考虑概率密度函数，假定 $p(x_i \mid c) \sim \mathcal{N}(\mu_{c,i}, \sigma_{c,i}^2)$

$$p(x_i \mid c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma_{c,i}^2}\right)$$

拉普拉斯修正 (Laplacian correction)

若某个属性值在训练集中没有与某个类同时出现过，则直接计算会出现问题，因为概率连乘将“抹去”其他属性提供的信息

例如，若训练集中未出现“敲声=清脆”的好瓜，
则模型在遇到“敲声=清脆”的测试样本时 .....

令 N 表示训练集 D 中可能的类别数，N_i 表示第 i 个属性可能的取值数

$$\hat{P}(c) = \frac{|D_c| + 1}{|D| + N}, \qquad \hat{P}(x_i \mid c) = \frac{|D_{c,x_i}| + 1}{|D_c| + N_i}$$

假设了属性值与类别的均匀分布，这是额外引入的 bias

朴素贝叶斯分类器的使用

- 若对预测速度要求高

  → 预计算所有概率估值，使用时“查表”

- 若数据更替频繁

  → 不进行任何训练，收到预测请求时再估值

  (懒惰学习, lazy learning)

- 若数据不断增加

  → 基于现有估值，对新样本涉及的概率估值进行修正

  (增量学习, incremental learning)

半朴素贝叶斯分类器

朴素贝叶斯分类器的“属性独立性假设”在现实中往往难以成立

半朴素贝叶斯分类器 (semi-naïve Bayes classifier)

基本思路：适当考虑一部分属性间的相互依赖信息

最常用策略：独依赖估计 (One-Dependent Estimator, ODE)

假设每个属性在类别之外最多仅依赖一个其他属性

$$P(c \mid \boldsymbol{x}) \propto P(c) \prod_{i=1}^{d} P(x_i \mid c, pa_i)$$

x_i 的“父属性”

关键是如何确定父属性

两种常见方法

- SPODE (Super-Parent ODE)：

  假设所有属性都依赖于同一属性，称为“超父” (Super-Parent)，然后通过交叉验证等模型选择方法来确定超父属性

- TAN (Tree Augmented naïve Bayes)：

  以属性间的条件”互信息”(mutual information)为边的权重，构建完全图，再利用最大带权生成树算法，仅保留强相关属性间的依赖性

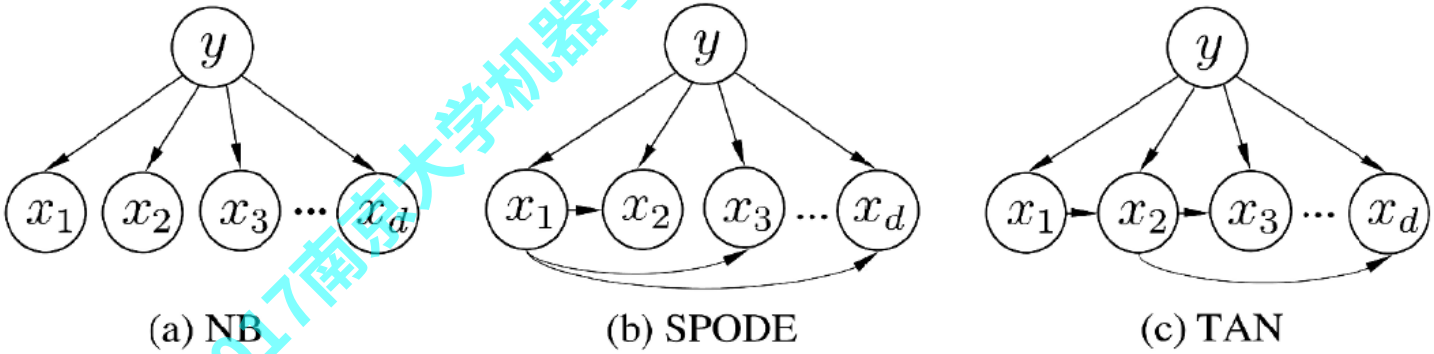

图 7.1 朴素贝叶斯与两种半朴素贝叶斯分类器所考虑的属性依赖关系

AODE (Averaged One-Dependent Estimator)

Geoff Webb
澳大利亚
Monash大学

- 尝试将每个属性作为超父构建 SPODE
- 将拥有足够训练数据支撑的 SPODE 集成起来作为最终结果

$$P(c \mid \boldsymbol{x}) \propto \sum_{\substack{i=1 \\ |D_{x_i}| \geqslant m'}}^{d} P(c, x_i) \prod_{j=1}^{d} P(x_j \mid c, x_i)$$

其中 D_{x_i} 是在第 i 个属性上取值为 x_i 的样本的集合，m' 为阈值常数

$$\hat{P}(c, x_i) = \frac{|D_{c,x_i}| + 1}{|D| + N_i}, \quad \hat{P}(x_j \mid c, x_i) = \frac{|D_{c,x_i,x_j}| + 1}{|D_{c,x_i}| + N_j}$$

D_{c,x_i,x_j} 表示类别为 c 且在第 i 和第 j 个属性上取值分别为 x_i 和 x_j 的样本集合

高阶依赖

能否通过考虑属性间的高阶依赖来进一步提升泛化性能？

例如最简单的做法：ODE → kDE

将父属性 pa_i 替换为包含 k 个属性的集合 $\mathbf{pa}_i$

明显障碍：随着 k 的增加，估计 $P(x_i \mid y, \mathbf{pa}_i)$ 所需的样本数将以指数级增加

- 训练样本非常充分 → 性能可能提升
- 有限训练样本 → 高阶联合概率估计困难

考虑属性间的高阶依赖，需要其他办法

To be continued