目录

第七章　贝叶斯分类器

贝叶斯分类器（Bayes Classifier）是一种通过最大化后验概率进行单点估计的分类器。

0.1　贝叶斯决策论

0.1.1　贝叶斯最优分类器

以多分类任务为例，假设有 N 种标记，即 $\mathcal{Y} = c_1, c_2, ..., c_N$，用 λ_{ij} 表示把一个真实标记为 c_i 的样本误分类为 c_j 所产生的损失。那么将样本 $\mathbf{x}$ 分类为 c_i 的**期望损失**（expected loss）或者说，在样本 $\mathbf{x}$ 上的**条件风险**（conditional risk）：

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

它描述的是，给定一个样本 $\mathbf{x}$，把它分类为 c_i 需要冒多大的风险。或者说，当样本真实标记不是 c_i 时，会有多大的损失。这个损失是一个求和，每一个求和项都是某一类别的后验概率和对应误分类损失代价的积。

在单个样本条件风险的基础上，可以定义总体风险：

$$R(h) = \mathbb{E}_{\mathbf{x}}[R(h(\mathbf{x}) \mid \mathbf{x})]$$

它描述的是，所有样本的条件风险的数学期望。其中 h 是一种用于产生分类结果的判断准则。

贝叶斯判定准则（Bayes decision rule）：要最小化总体风险，只需在每个样本上选择能使对应的条件风险 $R(c \mid \mathbf{x})$ 最小的标记。即：

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

h^* 称为贝叶斯最优分类器（Bayes optimal classifier），对应的总体风险 $R(h^*)$ 称为贝叶斯风险（Bayes risk），而 $1 - R(h^*)$ 则反映了分类器所能达到的最好性能，也即模型精度的理论上限。

如果学习模型的目标是令分类错误率最小，那么分类正确时误分类损失 λ_{ij} 为 0，反之为 1。这是条件风险就是：

$$R(c \mid \mathbf{x}) = 1 - P(c \mid \mathbf{x})$$

要令风险最小，只需要选择使样本 $\mathbf{x}$ 后验概率最大的一个类别标记就可以了。

0.1.2　贝叶斯定理

从概率的角度来理解，机器学习的目标就是基于有限的训练样本集尽可能准确地估计出后验概率（当然，大多数机器学习技术无需准确估计出后验概率）。

获取后验概率主要有两种策略：

- 构建判别式模型（discriminative models）：给定样本 $\mathbf{x}$，直接对后验概率 $P(\mathbf{x} \mid c)$ 建模来预测 c。这类模型包括决策树、BP 神经网络、支持向量机等等。

- 构建生成式模型（generative models）：给定样本 $\mathbf{x}$，先对联合概率分布 $P(\mathbf{x}, c)$ 建模，然后再利用联合概率计算出后验概率 $P(c \mid \mathbf{x})$，也即 $P(c \mid \mathbf{x}) = \frac{P(\mathbf{x},c)}{P(\mathbf{x})}$。

又因为联合概率 $P(\mathbf{x}, c) = P(c \mid \mathbf{x}) \times P(\mathbf{x}) = P(\mathbf{x} \mid c) \times P(c)$，由此，能得到贝叶斯定理：

$$P(c \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c) \times P(c)}{P(\mathbf{x})}$$

其中 $P(c \mid \mathbf{x})$ 是类标记 c 相对于样本 $\mathbf{x}$ 的**条件概率**;$P(\mathbf{x} \mid c)$ 是样本 $\mathbf{x}$ 相对于类标记 c 的**类条件概率**（class-conditional probability），或称为似然（likelihood），也由于得自 c 的取值而被称作 $\mathbf{x}$ 的后验概率。$P(c)$ 是 c 的**先验概率**（也称为边缘概率），之所以称为" 先验" 是因为它不考虑任何 $\mathbf{x}$ 方面的因素。在这里又称为类先验（prior）概率。$P(\mathbf{x})$ 是 $\mathbf{x}$ 的先验概率。在这里是用作归一化的证据（evidence）因子，与类标记无关。

类先验概率 P(c) 表示的是**样本空间中各类样本的比例**，根据大数定律，当训练集包含足够多的独立同分布样本时，类先验概率可以直接通过训练集中各类样本出现的频率进行估计。

0.2 极大似然估计

估计类条件概率的一种常用策略是：先假定该类样本服从某种确定的概率分布形式，然后再基于训练集中的该类样本对假定的概率分布的参数进行估计。

如果类 c 的样本服从参数为 θ_c（可能不止一个参数）的分布，那么从样本空间抽取到该类的某一个样本 $\mathbf{x}$ 的概率就是 $P(\mathbf{x} \mid \theta_c)$。使用 D_c 来表示训练集中类 c 的子集，可以定义数据集 D_c 的**似然（likelihood）**为：

$$P(D_c \mid \theta_c) = \prod_{\mathbf{x} \in D_c} P(\mathbf{x} \mid \theta_c)$$

由于连乘操作容易造成下溢，实际任务中通常使用**对数似然（log-likelihood）**代替：

$$LL(\theta_c) = \log P(D_c \mid \theta_c) = \sum_{\mathbf{x} \in D_c} \log P(\mathbf{x} \mid \theta_c)$$

所谓**极大似然估计**（Maximum Likelihood Estimation，简称 MLE）就是找出令似然最大的参数 θ_c。也即从 θ_c 的所有可能取值中找到一个令所抽取样本出现的可能性最大的值。求解的过程，就是求似然函数的导数，令导数为 0，得到似然方程，解似然方程得到最优解，也即该类样本分布的参数。

0.3 朴素贝叶斯分类器

朴素贝叶斯分类器（naive Bayes clssifier）采用**属性条件独立性假设**（attribute conditional independence assumption）。基于这个假设，可以把类条件概率写成连乘的形式，因此贝叶斯定理可重写为：

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

其中 d 为属性数目，x_i 为样本 $\mathbf{x}$ 在第 i 个属性上的取值。

又因为 $P(\mathbf{x})$ 与类别无关，所以朴素贝叶斯分类器的表达式可以写为：

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

当训练集包含足够多独立同分布样本时，类先验概率 $P(c)$ 可以直接算出，也即训练集该类样本的数目占训练集规模的比例：

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

而条件概率 $P(x_i \mid c)$，根据属性类型分离散和连续两种情况：

- 离散型属性：条件概率 $P(x_i \mid c)$ 可以估计为，在类别 c 的样本子集中，第 i 个属性取值为 x_i 的样本所占的比例：

$$P(x_i \mid c) = \frac{|D_{c,x_i}|}{|D_c|} \tag{1}$$

- 连续性属性：替换为概率密度函数，假设第 i 个属性服从高斯分布 (正态分布)，那么条件概率就写成 $p(x_i \mid c) \sim \mathcal{N}(\mu_{c,i}, \sigma_{c,i}^2)$，其中 $\mu_{c,i}$ 和 $\sigma_{c,i}^2$ 分别是第 c 类样本在第 i 个属性上取值的均值和方差。把属性取值 x_i 代入概率密度函数就可算出条件概率：

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

0.3.1 拉普拉斯修正

若某个属性值在训练集中没有与某个类同时出现过，那么它对应的条件概率 $P(x_i \mid c)$ 就为 0。在连乘中，这就意味着整个式子值为 0 了，其他属性携带的信息都被抹去了。此时，就需要对概率值进行平滑（smoothing）了，最常用的是拉普拉斯修正（Laplacian correction），假设训练集中包含 N 个类别，第 i 个属性包含 N_i 种取值，则拉普拉斯修正把式（1）和式（2）修改为：

$$P(c) = \frac{|D_c| + 1}{|D| + N} \tag{3}$$

$$P(x_i \mid c) = \frac{|D_{c,x_i}| + 1}{|D_c| + N_i} \tag{4}$$

拉普拉斯修正保证了**不会因为训练集样本不充分导致概率估值为零**。但它实际上是假设了类别和属性值是均匀分布的，相当于额外引入了先验，这个假设并不总是成立。

0.3.2 比较

朴素贝叶斯分类器和前面学习的模型有一个不同的地方就是：并不是基于训练集和某些算法来学习模型的参数；而是利用训练集来算出一些概率，在预测时，根据新样本的情况，使用不同的概率计算出它被分到各个类的后验概率，然后取后验概率最大的一个类作为结果。

在实际任务中，有两种使用方式：

- 查表：若对预测速度要求较高，可以先根据训练集把所有涉及到的概率计算出来，然后存储好，在预测新样本时只需要查表然后计算就可以了。
- 懒惰学习：若数据更替比较频繁，也可以理解为用训练集算出的概率可能很快就失效了，更新换代的速度很快，那就采取懒惰学习（lazy learning）的方式，仅当需要预测时才计算涉及到的概率。

特别地，当采取了预先计算所有概率的方式时，如果有新数据加入到训练集，只需要更新新样本涉及到的概率（或者说计数）就可以了，可以很方便地实现**增量学习**。

0.4 半朴素贝叶斯分类器

朴素贝叶斯分类器基于属性条件独立性假设，每个属性仅依赖于类别；有时候属性之间会存在依赖关系，适当考虑一部分属性间的相互依赖信息，这就是半朴素贝叶斯分类器（semi-naive Bayes classifier）的基本思想。

独依赖估计（One-Dependent Estimator，简称 ODE）是半朴素贝叶斯分类器最常用的一种策略，它假设的是每个属性在类别之外最多仅依赖于一个其他属性。也即：

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

其中 pa_i 是属性 x_i 依赖的另一属性，称为 x_i 的父属性。若已知父属性，就可以按式（4）来计算了。

确定每个属性的父属性的方法有 SPODE 和 TAN。

0.5 贝叶斯网

贝叶斯网（Bayesian network）亦称信念网（belief network），它借助**有向无环图**（Directed Acyclic Graph，简称 DAG）来刻画属性之间的依赖关系，并使用**条件概率表**（Conditional Probability Table，简称 CPT）来描述属性的联合概率分布。

贝叶斯网的学习包括结构的学习和参数的学习，而预测新样本的过程则称为推断（inference）。

0.5.1 EM 算法

前面讨论的极大似然估计方法是一种常用的参数估计方法，它是假设分布的形式，然后用训练样本来估计分布的参数。但实际任务中，遇到一个很大的问题就是**训练样本不完整**（样本某些属性的值缺失）。这时就需要用到 EM（Expectation-Maximization）算法了。将每个属性的取值看为一个变量，那么缺失的就可以看作“未观测”变量，称作**隐变量（latent variable）**。

整个训练集可以划分为已观测变量集 X 和隐变量集 Z 两部分。按照极大似然的思路，我们依然是想找出令训练集被观测到的概率最大的参数 Θ。也即最大化对数似然：

$$LL(\Theta \mid X, Z) = \ln P(X, Z \mid \Theta)$$

但是，由于 Z 是隐变量，无法观测到，所以上面这个式子实际是没法求的。

EM 算法的步骤如下：

1. 设定一个初始的 Θ
2. 按当前的 Θ 推断隐变量 Z 的（期望）值 **（E 步）**
3. 基于已观测变量 X 和步骤 2 得到的 Z 对 Θ 做最大似然估计得到新的 Θ **（M 步）**
4. 若未收敛（比方说新的 Θ 与旧的 Θ 相差仍大于阈值），就回到步骤 2，否则停止迭代

EM 算法可以看作是用**坐标下降（coordinate descent）法**来最大化对数似然下界的过程，每次固定 Z 或者 Θ 中的一个去优化另一个，直到最后收敛到局部最优解。

0.6 补充内容

朴素贝叶斯分类器的属性条件独立性假设在现实中很难成立，但事实上它在大多数情形下都有不错的性能。关于这点，有以下两种解释：

1. 对分类任务来说，只需各类别的条件概率排序正确，即使概率值不准确，也可以产生正确的分类结果；

2. 若属性间的相互依赖对所有类别影响都相同，或者依赖关系互相抵消，则属性条件独立性假设在降低开销的同时不会给性能带来负面影响。

注意，本章讨论的**贝叶斯分类器**和一般意义上的**贝叶斯学习**（Bayesian learning）是有很大差别的，本章讨论的贝叶斯分类器只是**通过最大化后验概率来进行单点估计**，获得的仅仅是一个数值；而贝叶斯学习则是进行**分布估计**或者说**区间估计**，获得的是一个分布。