目录

第十一章　特征选择与稀疏学习

特征选择的原因：必要性 (特征过多造成维数灾难) 和有利性 (去除冗余特征减低学习难度)。

0.1　子集搜索与评价

0.1.1　子集搜索

前向搜索：对于给定特征集，第一次选择最优的一个特征，使得在子集评价下最优，第二次从剩下的特征中再选择一个特征，使得这两个特征组成的特征子集在子集评价下最优，并且比一个特征时更优，依次进行，直到多选一个特征构成子集不更优或者选择完所有特征。

后向搜索：从全体特征子集中每次剔除一个特征，直到剩下的特征子集再剔除一个时效果更差。前向搜索和后向搜索结合就是双向搜索：一边增加选定的特征，一边删除无关特征。

无论是前/后/双向搜索都是基于贪心策略的。

0.1.2　子集评价

给定数据集 D，假定样本属性为离散型且 D 中第 i 类样本所占的比例为 $p_i(i=1,2,\cdots,|\mathcal{Y}|)$。对于属性子集 A，假定根据取其值将 D 分成了 V 个子集 $\{D^1,D^2,\cdots,D^V\}$, 每个子集中的样本在 A 上取值相同，计算属性子集 A 的信息增益为:

$$\mathrm{Gain}(A)=\mathrm{Ent}(D)-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\mathrm{Ent}(D^v)$$

其中信息熵定义为:

$$\mathrm{Ent(D)}=-\sum_{k=1}^{|\mathcal{Y}|}p_k\log_2 p_k$$

信息增益 $\mathrm{Gain}(A)$ 越大，意味着特征子集 A 包含的有助于分类的信息越多，所以信息增益越大越好。

0.2　特征选择方法

特征选择方法：特征子集搜索机制与子集评价机制相结合，例如决策树就是一种特征选择方法。常见的特征选择方法可分为三类：过滤式 (filter)，包裹式 (wrapper) 和嵌入式 (embedding)。

0.2.1　过滤式选择

先对数据集进行特征选择，然后再训练学习器，特征选择过程和后续学习器无关。**Relief**(Relevant Features) 是一种著名的过滤式特征选择方法。Relief 给每个属性 j 定义了一个“相关统计量”:

$$\delta^j=\sum_i -\mathrm{diff}(x_i^j,x_{i,nh}^j)^2+\mathrm{diff}(x_i^j,x_{i,nm}^j)^2$$

其中 x_a^j 表示样本 $\boldsymbol{x}_a$ 在属性 j 上的取值，$\boldsymbol{x}_{i,nh}$ 表示 $\boldsymbol{x}_i$ 在同类样本中的最近邻 (称为猜中近邻,near-hit)，$\boldsymbol{x}_{i,nm}$ 表示 $\boldsymbol{x}_i$ 的异类样本中的最近邻 (称为猜错近邻,near-miss)。diff(x_a^j, x_b^j) 取决于属性 j 的类型：若属性 j 为离散型，则 $x_a^j = x_b^j$ 时，diff(x_a^j, x_b^j) 为 0，否则为 1；若属性 j 为连续型，则 diff$(x_a^j, x_b^j$=$|x_a^j - x_b^j|$，注意 x_a^j, x_b^j 均已规范化到区间 $[0, 1]$。

相关统计量越大分类能力越强。Relief 的时间开销随采样次数以及原始特征线性增长。

0.2.2 包裹式选择

包裹式特征选择直接把最终将要使用的学习器性能作为特征子集的评价准则。LVW(Las Vegas Wrapper) 是一个典型的包裹式特征选择方法。它在**拉斯维加斯方法** (Las Vegas method) 框架下使用随机策略来进行子集搜索，并最终分类器的误差为特征子集评价准则。

即：每次随机选择一个子集，如果这个子集在最终学习器上的误差更小，或者误差相同但是子集的基数更小那就替换这个子集为最终要选择的特征子集，直到随机选择的迭代次数到达给定值。

因为包裹式选择是针对学习器优化的，所以一般要比过滤式选择更好；但是因为包裹式选择中需要多次训练学习器，因此包裹式特征选择的**计算开销通常比较大**。

0.2.3 嵌入式选择

嵌入式特征选择是将特征选择过程与学习器训练过程融为一体，即：**引入正则化向**。

对于给定数据集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$，其中 $\boldsymbol{x} \in \mathbb{R}^d, y \in \mathbb{R}$。对于线性回归模型以平方误差为损失函数，则优化目标为：

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (y_i - \boldsymbol{w}^T \boldsymbol{x}_i)^2$$

为了缓解过拟合问题，引入 L2 范数正则化，则有：

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (y_i - \boldsymbol{w}^T \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{w}\|_2^2$$

其中正则化参数 $\lambda > 0$，上式称为**岭回归** (ridge regression)。

若将 L2 正则化替换为 L1 正则化，则称为 LASSO(Least Absolute Shrinkage and Selection Operator)。

0.2.4 L0,L1,L2 正则化比较

L1 和 L2 都可以防止过拟合：L1 是舍弃了一些不重要的点；L2 是控制所有特征的权重。

L1 比 L2 更容易获得“稀疏解”，即它求得的 $\boldsymbol{w}$ 会有更少的非零向量。实际上，L0 比 L1 更容易获得稀疏解，但是通常不用 L0，因为 L0 范数不连续，难以优化求解，通常用 L1 来近似。

理解 L1 比 L2 更容易获得稀疏解即是理解 L1 情况下的损失函数为最小值的 $\boldsymbol{w}$ 解中有更多的 $w_i = 0$；也就是有更多的 $w_i = 0$ 使损失函数成立。因为损失函数是极小值，那么就要使更多的 $w_i = 0$ 的导数为 0，或者导数左右异号。如果原来的导数不为 0，那么 L2 正则化也不会为 0。【因为 $(cw_i{}^2)' = 2cw_i = 0$，相当于加正则项导数为 0。】而 L1 正则化有可能使导数左右异号【因为 $|cw_i|' = c \times \text{sign}(w_i) \times w_i$，只要 c 大于原来 0 处的导数的绝对值，$w_i = 0$ 就是一个极小值。】

0.3 稀疏表示与字典学习

稀疏表示就是当把数据集 D 考虑成一个矩阵时，每一行表示一个样本，每一列表示一个属性，尽可能去除特征无关的列，使得学习任务的难度有所降低，涉及的计算和存储开销也减少。

稀疏表示的好处：例如**线性支持向量机之所以能在文本数据上有很好的性能，恰是因为文本数据在使用字频表示 (每个字看做一个特征，字在文档中出现的频率或次数作为特征的取值) 后具有高度的稀疏性，使大多数问题变得线性可分。**

字典学习 (dictionary learning)：为普通稠密表达的样本找到合适的字典 (以用于类似上面的字频表示)，使样本转化为合适的稀疏表示形式，从而使学习任务得以简化，模型复杂度得以降低。

给定数据集 $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m\}$，字典学习最简单的形式为：

$$\min_{\boldsymbol{B}, \boldsymbol{\alpha}_i} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{\alpha}_i\|_2^2 + \lambda \sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1$$

其中 $\boldsymbol{B} \in \mathbb{R}^{d \times k}$ 为字典矩阵,k 称为字典的词汇量，通常由用户指定，$\boldsymbol{\alpha}_i \in \mathbb{R}^k$ 则是样本 $\boldsymbol{x}_i \in \mathbb{R}^d$ 的稀疏表示。上述式子中的第一项是希望 $\boldsymbol{\alpha}_i$ 能很好地重构 $\boldsymbol{x}_i$，第二项则是希望 $\boldsymbol{\alpha}_i$ 尽量稀疏。

0.4 压缩感知

压缩感知 (compressed sensing) 分为“感知测量”和“重构恢复”两个阶段。感知测量是指对原始信号进行处理以获得稀疏样本表示；而重构恢复则是基于稀疏性从少量观测中恢复原始信号，这是压缩感知的关键部分。

矩阵补全 (matrix completion) 技术：

$$\min_{\boldsymbol{X}} \ \text{rank}(\boldsymbol{X}) \qquad \text{s.t.} \quad (\boldsymbol{X})_{ij} = (\boldsymbol{A})_{ij}, \ (i,j) \in \Omega$$

其中 $\boldsymbol{X}$ 表示要恢复的稀疏信号，$\boldsymbol{A}$ 是已知的信号；Ω 是 $\boldsymbol{A}$ 中已知元素的下标集合。约束项表示恢复出的矩阵中 $(\boldsymbol{X})_{ij}$ 应当与观测到的对应元素相同。

求解上式是 NP 难的，但是注意到 $\text{rank}(\boldsymbol{X})$ 在集合 $\{\boldsymbol{X} \in \mathbb{R}^{m \times n} : \|\boldsymbol{X}\|_F^2 \leq 1\}$($\|\cdot\|_F$ 表示矩阵的 Frobenius 范数，即每个元素平方和后再开平方) 上的凸包是 $\boldsymbol{X}$ 的核范数：

$$\|\boldsymbol{X}\|_* = \sum_{j=1}^{\min\{m,n\}} \sigma_j(\boldsymbol{X})$$

其中 $\sigma_j(\boldsymbol{X})$ 表示 $\boldsymbol{X}$ 的奇异值，即矩阵的核范数为矩阵的奇异值之和，于是可以通过最小化矩阵核范数来近似求解矩阵补全的式子，即：

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_* \qquad \text{s.t.} \quad (\boldsymbol{X})_{ij} = (\boldsymbol{A})_{ij}, \quad (i,j) \in \Omega$$

上面的式子是一个**凸优化**问题，可以通过**半正定规划**求解。