目录

10 特征选择

10.1 特征选择概述

特征选择的目的是数据归约. 具体而言, 特征选择方法是在尽量不损失信息的前提下, 通过删除不重要或不必要的变量, 或者选择出重要的变量, 实现数据维度上的归约, 从而提升建模的效率和模型的解释性能.

特征选择方法可分为三类: 过滤法、封装法和嵌入法.

过滤法(filter) 通过分析变量自身的特点以及与因变量的关系, 在数据预处理阶段过滤掉不重要或不必要的变量. 过滤法的特征选择过程独立于数据预处理之后的模型训练过程, 因此与后续分析阶段模型的选择无关. 过滤法的优点是速度快, 缺点是可能会删除有用的变量. 按照是否采用了因变量的信息, 可以将过滤法分为有监督过滤法和无监督过滤法.

封装法(wrapper) 需要借助于一定的模型或算法估计变量子集的预测能力, 进而选择预测能力强的变量子集. 需要注意的是, 数据分析中最终建立的模型可能与封装法中的模型不同. 所以, 封装法主要作为数据预处理方法. 可以根据最终建立的模型的特点实施封装法. 例如, 对于线性分类模型, 可以采用LASSO方法选择变量; 对于非线性分类模型, 可以采用随机森林等方法选择变量. 封装法的优点是可以选出高质量的变量子集, 缺点是速度比较慢.

嵌入法(embed) 是将特征选择嵌入模型本身, 使得模型具有特征选择能力. 例如, 决策树方法在构建过程中会自动选择最有效的变量. 嵌入法的优点是将特征选择与模型构建完全融合, 一步到位; 缺点是为建模阶段造成了一定的负担, 并且效果依赖于模型. 例如, 决策树CART模型多在局部空间中通过贪心搜索选择变量, 无法保证所选择的变量是全局最优的.

表10.1对三类特征选择方法作了详细的对比. 在实务中进行特征选择, 可以先借由专家知识来初步筛选特征, 然后采用过滤法快速过滤不重要或不必要的变量, 最后采用封装法或嵌入法得到优化的变量子集和模型.

10.2 无监督过滤法

无监督过滤法不依赖于因变量, 仅依据自变量的信息过滤特征. 无监督过滤法主要有四种, 下面将分别介绍.

10.2.1 删除缺失值比例较高的变量

一方面, 缺失值比例较高的变量往往信息量较小, 对预测的贡献可能很有限. 另一方面, 缺失值比例较高的变量增加了数据处理的困难. 因此, 删除缺失值比例较高的变量往往是必要的, 它是一种简单高效的过滤方法.

具体应用中, 需要设置一个阈值. 当变量中缺失值比例大于该阈值时, 删除该变量.

注记10.1. 需要注意缺失值的定义, “正常空缺”的值不属于缺失值. 当计算变量的缺失值比例时, 应排除”正常空缺”的情况.

10.2.2 删除方差几乎为零的变量

我们知道, 具有变异性的变量才能为预测提供信息, 变异性越大的变量所提供的信息往往越多, 从而对预测的贡献可能越大. 因此, 可以考虑删除变异性小、对预测的贡献可能较小的变量.

方差常用于刻画变量的变异性, 方差越小, 说明变量的变异性越小. 但是, 方差是带有变量的单位和量纲的统计量, 不能绝对化地对待. 例如, 分别以米和厘米作为物体的长度单位, 所得到的方差值差异较大. 因此, 对带有单位和

表 10.1: 三类特征选择方法的特点

特征选择方法		优点	缺点	典型方法
过滤法	单变量	(1)速度快 (2)可扩展 (3)与模型独立	(1)忽略了特征间关系 (2)忽略了特征与模型的关系	卡方检验 信息增益 相关系数
	多变量	(1)考虑了特征间关联 (2)与模型独立 (3)计算复杂度低于封装法	(1)计算速度与可扩展性低于单变量方法 (2)忽略了特征与模型的关系	变量聚类筛选
封装法	确定性算法	(1)简单 (2)与模型相关 (3)考虑了特征间相互作用 (4)计算密集度低于随机算法	(1)容易过拟合 (2)相比随机算法, 容易卡在局部最优子集处 (3)依赖模型	循序特征选择 递归特征删除
	随机算法	(1)不易陷入局部最优点 (2)与模型相关 (3)考虑了特征间相互作用	(1)计算密集型 (2)依赖模型 (3)相比确定性算法，过拟合的风险较高	遗传算法 模拟退火
嵌入法		(1)与模型相关 (2)计算复杂度低于封装法 (3)考虑了特征间相互作用	依赖模型	决策树 随机森林 梯度提升树 LASSO

来源: 美团算法团队(2018), 略有改动.

量纲的方差而言, 以“几乎为零”的标准予以判断, 往往不够准确. 实际应用中, 可以先进行数据规范化, 去掉变量的单位和量纲, 再实施该方法; 或者采用不带单位和量纲的变异性刻画指标, 例如, 变异系数. 判断时, 需要设置阈值, 若变异性指标小于该阈值, 则认为变量“方差几乎为零”, 可删除该变量.

此外, 可以通过一些规则判断变量的变异性是否足够小. 例如, 当同时满足以下两个条件时判定变量“方差几乎为零” [1]:

(1) 变量取值个数与样本量之比不超过10%.

(2) 频率最高与频率次高的取值频数之比超过20.

这一规则将变量取值较少且分布过度集中于众数的变量判定为“方差几乎为零”.

10.2.3 去除强相关的自变量

强相关意味着一定的信息冗余, 若保留过多强相关的自变量将导致建模效率低、模型复杂且不易解释. 此外, 在线性回归模型中, 强相关的自变量可能导致系数估计值的误差很大、模型很不稳定, 这就是共线性问题.

去除强相关的自变量, 可以简单有效地解决上述问题. 处理的思路可以有多种, 这里介绍其中一种. 这一思路是在删除尽量少的变量基础上, 保证所有成对变量间的相关系数绝对值低于某一设定的阈值, 例如0.75. 具体的步骤为:

步骤1: 计算相关系数矩阵, 获得所有变量对的相关系数值.

步骤2: 找出最大相关系数绝对值所对应的两个变量, 记为A和B.

步骤3: 计算变量A与其他变量相关系数绝对值的平均值, 对变量B也作同样的计算.

步骤4: 比较步骤3中的结果, 删除相关系数绝对值的平均值最大的变量.

重复步骤1至步骤4, 直到所有变量对的相关系数绝对值低于所设定的阈值.

10.2.4 通过聚类过滤变量

通过聚类过滤变量的主要思路是: 执行变量聚类方法, 将变量分为若干个组, 落入同一组中的变量可能具有一定的同质性. 因此, 可以在每一个组中过滤掉一部分变量, 仅保留一部分具有代表性的变量.

[1]这是R语言caret包中对“方差几乎为零”的判定规则.

10.3 有监督过滤法

与无监督过滤法不同的是, 有监督过滤法依赖于因变量, 实施过滤的依据通常是单一自变量对预测因变量的贡献度, 若变量对预测的贡献度低, 则删除该变量. 有监督过滤法中的关键问题是如何度量单一自变量对预测因变量的贡献度, 以及如何确定应过滤的变量数量. 例如, 可采用以下三种思路:

(1) 依据相关性分析的显著性检验结果. 检验中的原假设是两个变量独立, 从而检验的p值越小说明原假设成立的可能性越小, 变量对预测的贡献可能越大. 因此, 若相关性检验的p值小于显著性水平, 表明自变量与因变量具有显著的相关性, 则应保留该自变量; 反之, 则应过滤该自变量.

(2) 依据相关性度量指标. 采用如Pearson线性相关系数、Spearman等级相关系数、Kendall's τ相关系数等反映相关程度的指标, 并设置希望保留的自变量数量k, 仅保留与因变量相关性最强的k个自变量.

(3) 依据有监督离散化方法结果. 在第6章中提到, 若有监督离散化方法将一个变量的所有取值归并为一个区间, 则表明该变量对预测因变量的贡献很小, 此时可以删除该变量.

表10.2展示了前两种思路下常用的方法. 需要注意的是, 应根据自变量和因变量的类型, 选择恰当的方法. 第三种思路不在本章中赘述.

有监督过滤法的缺陷是未考虑多个自变量的组合与因变量的关系. 例如, 当两个自变量与因变量都不具有显著的相关性, 但是两个自变量的交互作用与因变量显著相关时, 有监督过滤会导致遗漏预测贡献度高的交互作用项. 实际应用中, 建议先作特征衍生, 例如, 获得各变量的交互作用项, 然后再实施有监督过滤法, 从而更大程度地保留对预测有效的变量信息.

对于读者较为熟悉的方法, 这里不作展开. 下面, 着重介绍带有多重检验的有监督过滤方法, 以及依赖于互信息和最大信息系数的有监督过滤方法.

10.3.1 基于多重检验过滤变量

在有监督过滤方法中, 通常需要逐一检验每一个自变量与因变量的相关性. 随着自变量数量的增加, 所进行的检验数量也会增大. 虽然每一个检验的第I类错误概率可以控制得比较小, 但将多个检验合起来看, 整体的第I类错误概率可能会比较大. 例如, 将每一个检验的第I类错误概率控制在0.05的水平, 并且假定各检验相互独立. 那么, 若有10个检验, 将以$1-0.95^{10}\approx 0.4$的概率至少发生一次错误地拒绝原假设; 如果检验的数量达到100个, 那么至少发生一次错误地拒绝原假设的概率将达到0.99. 为整体控制多个检验的错误概率, 多重检验方法被提出. 下面, 介绍最常用的两种多重检验方法.

表 10.2: 基于相关性的有监督过滤方法

序号	自变量X	因变量Y	相关性的显著性检验	相关性度量
1	定性	定性	(1)列联表检验 (2)Fisher精确检验 (3) 基于相关系数的显著性检验	(1)优势比 (2)互信息 (3)相关系数
2	定性	定量	(1)ANOVA (2)Kruskal Wallis检验 (3) 基于相关系数的显著性检验	(1)互信息 (2)相关系数
3	定量	定量	基于相关系数的显著性检验	(1)互信息 (2)最大信息系数 (3)相关系数

[1] 对于含有定性变量的情况, 相关系数、互信息和基于相关系数的显著性检验仅适用于定序变量.
[2] X为定量变量、Y为定性变量所对应的方法与序号2的情况相同.

第一种方法是Bonferroni校正, 其目标是控制簇错误率(Family Wise Error Rate, FWER). FWER是指至少拒绝了一个正确原假设的概率, 即至少犯了一次第I类错误的概率. 假定有m个假设检验问题$\{H_{0i} \text{ vs } H_{1i},\ i = 1, \cdots, m\}$, $p_1, \cdots, p_m$是相应检验的p值. 令显著性水平为α, Bonferroni校正的判断规则是: 当$p_i \le \alpha/m$时, 拒绝原假设H_{0i},否则接受原假设H_{0i}. 设m个原假设中有m_0个原假设成立, 则有

$$\text{FWER} = P\left(\bigcup_{i=1}^{m_0}(p_i \le \alpha/m)\right) \le \sum_{i=1}^{m_0} P\left(p_i \le \alpha/m\right) = m_0 \cdot \alpha/m \le \alpha.$$

可见, Bonferroni校正可以控制m个假设检验问题整体的第I类错误概率.

下面是一个例子, 它将Bonferroni校正应用到有监督过滤中.

例10.1. 假定有五个自变量X_1至X_5, X_1与因变量Y相关性检验的p值为0.04, X_2, X_3, X_4, X_5与Y相关性检验的p值均为0.001. 取显著性水平$\alpha = 0.05$. 若不作Bonferroni校正, 由于所有的p值都小于0.05, 因此应拒绝所有的原假设. 在相关性检验中, 原假设为两个变量独立. 因此, 拒绝所有的原假设就意味着所有的自变量都与因变量显著相关, 从而应保留所有的自变量.

若采用Bonferroni校正, 检验的临界值应为$0.05/5 = 0.01$. 第一个检验的p值大于0.01, 应接受原假设, 即认为X_1与Y独立, 应过滤X_1; 其余四个p值小于0.01, 故应保留X_2, X_3, X_4, X_5.

表 10.3: Benjamini-Hochberg方法中$p(i)$与判断临界值

i	1	2	3	4	5
$p(i)$	0.001	0.001	0.001	0.001	0.04
判断临界值$(i\cdot\alpha)/m$	0.01	0.02	0.03	0.04	0.05

Benjamini & Hochberg (1995) 指出, Bonferroni校正过于保守, 因此提出一个新的方法, 称为Benjamini-Hochberg方法. 给定显著性水平α, 首先将m个检验的p值按从小到大的顺序排列, 并记为$p(i)$. 然后, 找出最大的k,使得$p(k) \le (k\cdot\alpha)/m$. 对于$i = 1,2,\cdots,k$, 拒绝所有对应的$k$个原假设; 对于其余的$m-k$检验, 不拒绝原假设. Benjamini-Hochberg方法是为了控制所有检验的假发现率(False Discovery Rate, FDR). 所谓的"发现"是一个拒绝了原假设的检验, "假发现"则是一个错误地拒绝了原假设的检验. 令D表示"假发现"占全部"发现"的比例, 则D的期望值即为假发现率. Benjamini-Hochberg假发现率控制定理表明, 当原假设为真的检验p值相互独立时, FDR小于等于α.

例10.1 (续) 按照检验的p值从小到大排序获得$p(i)$, 并将$p(i)$与判断临界值$(i\cdot\alpha)/m$作比较(见表10.3). 由于每一个$p(i)$都小于等于$(i\cdot\alpha)/m$, 因此$k = 5$. 此时, 应拒绝所有检验的原假设, 即认为所有的自变量都与因变量显著相关, 从而应保留所有的自变量. 可以看到, 这一结果与Bonferroni校正的结果有所不同.

10.3.2 基于互信息过滤变量

互信息(mutual information) 描述两个变量的关联程度, 需要通过变量的分布计算.

定义10.1. 设两个连续型随机变量X和Y的边际密度函数分别为$p_X(x)$和$p_Y(y)$, 二者的联合密度函数为$p(x,y)$. X和Y的**互信息**为:

$$I(X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log\frac{p(x,y)}{p_X(x)p_Y(y)}\,dy\,dx.$$

为了进一步理解互信息, 引入熵和条件熵两个概念.

定义10.2. 设连续型随机变量X的密度函数为$p_X(x)$. X的**熵(entropy)**为:

$$H(X) = -\int_{-\infty}^{\infty} p_X(x)\log p_X(x)\,dx.$$

熵刻画了随机变量的变异性, 取值大于等于0.

定义10.3. 设有两个连续型随机变量X和Y, Y的边际密度函数为$p_Y(y)$, Y给定时X的边际密度函数为$p_{X|Y=y}(x)$. Y给定时X的**条件熵(conditional entropy)** 为:

$$H(X|Y) = -\int_{-\infty}^{\infty} p_Y(y) \int_{-\infty}^{\infty} p_{X|Y=y}(x) \log p_{X|Y=y}(x)\, dx\, dy.$$

条件熵刻画了给定其他变量的信息后, 一个随机变量所具有的变异性. 条件熵的取值大于等于0.

由于$H(X) = -\int\int p_{Y|X=x}(y)p_X(x) \log p_X(x)\, dy\, dx$, 可以得到

$$\begin{aligned} & H(X) - H(X|Y) \\ = \; & -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{Y|X=x}(y)p_X(x) \log p_X(x)\, dy\, dx \\ & +\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_Y(y)p_{X|Y=y}(x) \log \tfrac{p(x,y)}{p_Y(y)}\, dy\, dx \\ = \; & \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y) \log \tfrac{p(x,y)}{p_X(x)p_Y(y)}\, dy\, dx \\ = \; & I(X,Y). \end{aligned} \tag{10.1}$$

同理, 可以得到

$$I(X,Y) = H(Y) - H(Y|X). \tag{10.2}$$

图10.1展示了互信息、熵和条件熵的关系. 这表明互信息刻画的是X与Y重叠和相依的信息, 因而可用于描述X与Y的关联性. 由互信息的表达式, 以及式(10.1)和(10.2)知, $0 \le I(X,Y) \le \min(H(X), H(Y))$. 当$X$与$Y$独立时, X与Y的联合分布等于二者边际分布的乘积, 因此互信息的值为0, 达到最小; 随着X与Y的联合分布与二者边际分布乘积的差异增大, 互信息的值也变得越大, 表明变量间的关联越大.

对于X和Y中含有一个或两个离散型随机变量的情况, 可类似地定义互信息. 实际应用中, 变量的分布信息往往未知, 因而互信息的计算较为困难. 需要基于一定的分布估计或熵的估计方法, 才能获得互信息的估计值.

例10.2. 假定自变量X和因变量Y都是取0和1两个值的离散型随机变量, 二者的联合分布如表10.4所示. 若采用对数$\log_2$, X与Y的互信息为

$$\begin{aligned} I(X,Y) &= 0.3 \times \log_2 \frac{0.3}{0.6 \times 0.6} + 0.3 \times \log_2 \frac{0.3}{0.6 \times 0.4} \\ &\quad +0.3 \times \log_2 \frac{0.3}{0.4 \times 0.6} + 0.1 \times \log_2 \frac{0.1}{0.4 \times 0.4} \\ &= 0.05. \end{aligned}$$

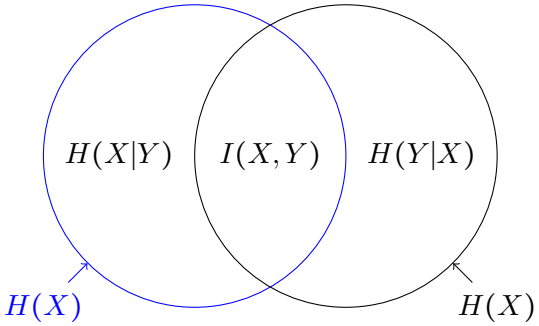


图 10.1: 互信息与熵的关系

表 10.4: X与Y的联合分布

	$Y=0$	$Y=1$	sum
$X=0$	0.3	0.3	0.6
$X=1$	0.3	0.1	0.4
sum	0.6	0.4	1

本例中, X和Y的熵都为1, 因此$I(X,Y)$的取值范围为$[0,1]$. 所得的互信息值为0.05, 接近于0, 表明X和Y的关联性不太强.

将互信息应用于特征选择, 需要计算所有自变量与因变量的互信息值, 过滤互信息值最小的若干个自变量. 但是, 由于互信息的取值范围与自变量的熵有关, 因此这些互信息值的尺度可能具有差异性, 导致不可比的问题. 可以考虑采用归一化的互信息. 当$\min(H(X),H(Y)) < +\infty$时, 归一化的互信息为:

$$I^*(X,Y) = \begin{cases} 0, & \min(H(X),H(Y)) = 0, \\ \frac{I(X,Y)}{\min(H(X),H(Y))}, & 0 < \min(H(X),H(Y)) < +\infty. \end{cases}$$

归一化后的互信息取值范围为$[0,1]$. 当应用于特征选择时, 可以过滤归一化的互信息值最小的若干个自变量. 对于连续型自变量, 可能出现$\min(H(X),H(Y)) = +\infty$的情况, 可将这类自变量的互信息值进行对比, 过滤若干互信息值小的自变量.

10.3.3 基于最大信息系数过滤变量

可以看到, 相对于连续型变量, 离散型变量的分布更易于估计和获取. 那么, 是否可以将连续型变量转换为离散型变量, 从而方便互信息的计算呢?

Reshef et al. (2011) 提出的最大信息系数(Maximal Information Coefficient, MIC) 采用了变量类型转换的思想, 也就是将连续型变量离散化, 从而基于转换后的离散型变量估计互信息. 但这里存在一个问题, 不同的离散化可能得到不同的分布, 从而产生不同的互信息估计值, 在这些估计值中该如何取舍呢? Reshefet al. (2011) 取所有互信息估计值的最大值, 这就是最大信息系数名称的由来.

记X^*和Y^*分别为X和Y离散化后的变量, $\#X$和$\#Y$分别表示X^*和Y^*的取值个数. 定义归一化的互信息值为

$$\frac{I(X^*, Y^*)}{\log(\min(\#X, \#Y))}.$$

其中, 分母部分可以用作归一化, 其原理是: 若离散型随机变量的取值个数为K, 则该变量的熵最大值为$\log K$. 由互信息与熵的关系式(10.1)和(10.2)可知, 互信息一定小于等于其中任何一个变量的熵, 因此互信息小于等于两个变量熵的最小值. 从而, 归一化的互信息取值范围为$[0, 1]$.

定义10.4. 设置离散化网格数量的上限为B $(B > 0)$, 考虑所有$\#X$乘以$\#Y$小于B的离散化, 取其中归一化互信息最大者为**最大信息系数**, 即

$$\text{MIC} = \max_{\#X \cdot \#Y < B} \frac{I(X, Y)}{\log(\min(\#X, \#Y))}.$$

可以看到, 最大信息系数的取值范围为$[0, 1]$.

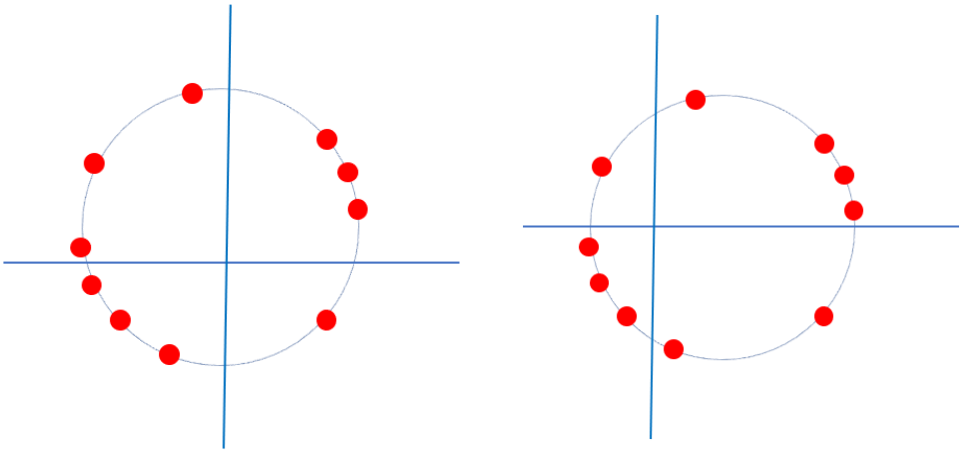
例10.3. 设有10个分布在单位圆上的样本观测, 通过两条相交的直线, 将数据划分入四个区域, 图10.2展示了两种可能的离散化, 表10.5和10.6分别展示了相应的联合分布. 取对数$\log_2$, 第一种离散化对应的归一化互信息为

$$\begin{aligned} & \frac{I(X^*, Y^*)}{\log_2(\min(\#X, \#Y))} \\ = \ & \frac{1}{\log_2(\min(2,2))} \Big(0.3 \times \log_2 \frac{0.3}{0.6 \times 0.6} + 0.3 \times \log_2 \frac{0.3}{0.6 \times 0.4} + 0.3 \times \log_2 \frac{0.3}{0.4 \times 0.6} \\ & + 0.1 \times \log_2 \frac{0.1}{0.4 \times 0.4}\Big) \\ = \ & 0.05. \end{aligned}$$

第二种离散化对应的归一化互信息为

$$\begin{aligned} & \frac{I(X^*, Y^*)}{\log_2(\min(\#X, \#Y))} \\ = \ & \frac{1}{\log_2(\min(2,2))} \Big(0.1 \times \log_2 \frac{0.1}{0.5 \times 0.4} + 0.4 \times \log_2 \frac{0.4}{0.5 \times 0.6} + + 0.3 \times \log_2 \frac{0.3}{0.5 \times 0.4} \\ & 0.2 \times \log_2 \frac{0.2}{0.5 \times 0.6}\Big) \\ = \ & 0.12. \end{aligned}$$

若仅尝试这两种离散化且$B > 4$, $\text{MIC} = \max(0.05, 0.12) = 0.12$.



(a) 离散化1　　(b) 离散化2

Figure 10.2: 2种离散化方式

表10.5: 离散化1对应的联合分布

	Y =上	Y =下	sum
X =左	0.3	0.3	0.6
X =右	0.3	0.1	0.4
sum	0.6	0.4	1

表10.6: 离散化2对应的联合分布

	Y=上	Y=下	sum
X=左	0.1	0.4	0.5
X=右	0.3	0.2	0.5
sum	0.4	0.6	1

最大信息系数可以检测基本上所有类型的函数关系. 若MIC取值为0, 说明两个变量不具有相关关系; 取值为1说明两个变量的相关关系很强. 图10.3将最大信息系数与Pearson线性相关系数作了对比. 可以看到, 最大信息系数适用于一般情况, 尤其是可以识别出非线性相关关系. 但当两个变量具有线性相关关系时, Pearson线性相关系数效果会更好.

将最大信息系数应用于特征选择, 需要计算所有自变量与因变量的MIC值, 过滤MIC值最小的若干个自变量.

10.4 封装法

封装法将需要借助于一定的模型或算法估计变量子集的预测能力, 进而选择预测能力强的变量子集. 可以通过封装法整合不同类型模型的能力, 从而提升预测效果. 例如, Cardie (1993) 发现通过决策树C4.5算法选择变量组合后再实施k近邻算法, 优于单独用C4.5算法或用k近邻算法的预测效果.

封装法针对一个优化问题, 优化目标是找到预测能力最好的变量子集. 实际上, 几乎所有的优化方法都可以应用于该优化问题. 例如, 通过穷举搜索所

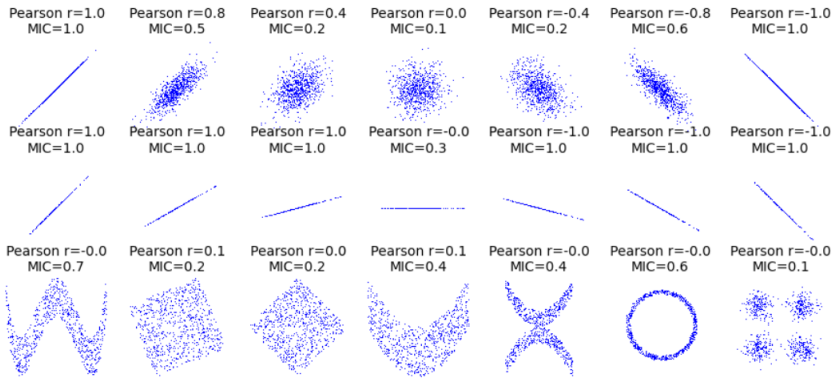


图 10.3: 互信息与Pearson线性相关系数的比较
(来源: https://minepy.readthedocs.io/en/latest/)

有的自变量组合实现, 这就是最优子集方法. 也可以采用诸如遗传算法和模拟退火等搜索算法实现优化, 找到最优或近似最优的变量组合.

下面, 介绍几种典型的封装法. 它们共同的特点是只需要搜索一部分变量组合, 从而提升搜索效率.

10.4.1 由单一模型筛选变量

由单一模型筛选变量是最简单的一种封装法. 此时, 只需要建立一个含有所有自变量的模型, 根据模型得到变量对预测的贡献度(例如, 变量相对重要性), 然后选择贡献度高于某一阈值的若干个自变量.

例10.4. 将四个备选自变量P, D, W, H与因变量Y建立一个模型. 由模型得到各自变量的变量相对重要性分别为: 0.23, 0.71, 0.06和0. 若设置阈值为所得到的变量相对重要性的中位数0.145, 也即选择一半的变量, 则变量P和D入选.

10.4.2 循序特征选择

典型的循序特征选择(Sequential Feature Selection, SFS) 包括向前循序特征选择和向后循序特征选择.

向前循序特征选择(Forward Sequential Feature Selection, FSFS) 构建一个入选变量集V. 初始的V为空集, 然后逐步向V中加入最优的自变量. 具体是, 对每一个尚未进入V的自变量X, 以X与V合并的自变量构建模型, 并评价模型效果, 选择预测效果最好的一个自变量加入V之中. 重复以上步骤, 直到V中的自变量数量已达到预设值.

表 10.7: 单一自变量的模型效果

自变量	P	D	W	H
R^2	0.39	0.59	0.60	0.21

表 10.8: W与1个变量组合的模型效果

自变量组合	$\{W,P\}$	$\{W,D\}$	$\{W,H\}$
修正的R^2	0.55	0.63	0.50

例10.4 (续) 采用向前循序特征选择方法, 设置入选变量数量为2. 假定因变量为定量变量, 采用R^2或修正的R^2作为模型效果的评价指标. 初始的入选变量集$V = \varnothing$.

> 第1步, 从四个备选自变量P, D, W, H中选择一个加入V. 建立并评价所有单一自变量所构建的模型, 所得结果如表10.7所示. 可以看到, 由变量W所构建的模型的R^2为最大, 因此W入选, $V = \{W\}$.
>
> 第2步, 从剩余的三个备选自变量P, D, H中进行选择一个加入V. 建立并评价备选变量加入V所构建的模型, 所得结果如表10.8所示. 可以看到, 以W与D的组合作为自变量的模型, 其修正的R^2达到最大, 因此D入选, $V = \{W, D\}$. 此时, 达到了预设的变量数量, 算法停止.

向前循序特征选择采用的是逐步加入自变量的思路, 这是"向前"的含义. 与之相反的是向后循序特征选择(Backward Sequential Feature Selection, BSFS). 所谓"向后", 就是逐步移除自变量. 向后循序特征选择方法也需要构建一个入选变量集V. 初始的V包含了所有的自变量, 需要逐步从V中移除预测贡献度最小的自变量. 具体是, 对V中的每一个自变量X, 以集合$V\backslash\{X\}$中的自变量构建模型, 并评价模型效果, 更新V为效果最好模型所对应的自变量组合. 重复以上步骤, 直到V中的自变量数量已达到预设值.

例10.4 (续) 采用向后循序特征选择方法, 设置入选变量数量为2. 假定因变量为定量变量, 采用修正的R^2作为模型效果的评价指标. 初始时所有变量都在入选变量集中, $V = \{P, D, W, H\}$.

> 第1步, 建立并评价从$V = \{P, D, W, H\}$移除一个自变量所构建的模型, 所得结果如表10.9所示. 可以看到, 移除H所构建模型的修正的R^2最大, 因此$V = \{P, D, W\}$.

表 10.9: 从$V = \{P, D, W, H\}$移除1个自变量的模型效果

自变量组合	$\{P, D, W\}$	$\{P, D, H\}$	$\{P, W, H\}$	$\{D, W, H\}$
修正的R^2	0.70	0.64	0.52	0.65

表 10.10: 从$V = \{P, D, W\}$移除1个自变量的模型效果

自变量组合	$\{P, D\}$	$\{P, W\}$	$\{D, W\}$
修正的R^2	0.71	0.55	0.63

第2步, 建立并评价从$V = \{P, D, W\}$移除一个自变量所构建的模型, 所得结果如表10.10所示. 可以看到, 移除W所构建模型的修正的R^2最大, 因此$V = \{P, D\}$. 此时, 达到了预设的变量数量, 算法停止.

由上述例子可以看到, 在相同的自变量数量要求下, 向前循序特征选择与向后循序特征选择所获得的结果可能不同.

10.4.3 递归特征删除

递归特征删除(Recursive Feature Elimination, RFE) 由Guyon et al. (2002) 提出. 类似于向后循序特征选择, 递归特征删除以"向后"的方式, 逐步移除变量. 但与向后循序特征选择所不同的是, 递归特征删除并不需要在各种组合中搜索比较, 而是根据各变量的评分, 组合评分最高的若干变量作为入选变量集. 评分可以由变量相对重要性、变量显著性检验结果等构造得到, 评分越高表明变量对预测的贡献越大.

设每一次移除的变量数为k个, 首先将所有变量放入入选变量集V中. 递归特征删除的具体步骤是:

第1步, 采用V中的变量作为自变量, 建立模型, 获得每个变量的评分.

第2步, 移除评分最低的k个变量, 更新V.

重复第1步和第2步, 直到V中的变量数量达到预设值.

例10.4 (续) 采用递归特征删除方法, 设置入选变量数为2个, 每一次移除的变量数$k = 1$. 评分为变量相对重要性. 初始化$V = \{P, D, W, H\}$.

第1步, 建立因变量Y关于P, D, W, H的模型, 获得各变量的评分(如表10.11所示).

表 10.11: Y关于P, D, W, H的模型中各自变量的评分

自变量	P	D	W	H
变量相对重要性	0.20	0.68	0.07	0.05

表 10.12: Y关于P, D, W的模型中各自变量的评分

自变量	P	D	W
变量相对重要性	0.21	0.68	0.11

第2步, 移除评分最低的变量H, 更新后$V = \{P, D, W\}$.

重复第1步, 建立Y关于P, D, W的模型, 获得各变量的评分(如表10.12所示).

重复第2步, 移除评分最低的变量W, 更新后$V = \{P, D\}$. 此时, 达到了预设的入选变量数量, 算法停止.

由这个例子可以看到, 相较于向后循序特征选择, 递归特征删除更为简单易行. 但代价是, 递归特征删除可能错过了更优的变量组合.

注记10.2. 循序特征选择和递归特征删除都涉及对模型效果的评价. 为使得评价指标更为准确客观, 建议采用留出法或交叉验证, 基于训练集建立模型, 并基于验证集计算评价指标.

由单一模型筛选变量、循序特征选择和递归特征删除都有共同的缺陷:

(1) 无法修改前面已加入或删除变量的操作. 例如, 向前循序特征选择中, 已经加入入选变量集V的自变量会永远留在V之中, 也就是说后续的步骤无法修正前序的结果.

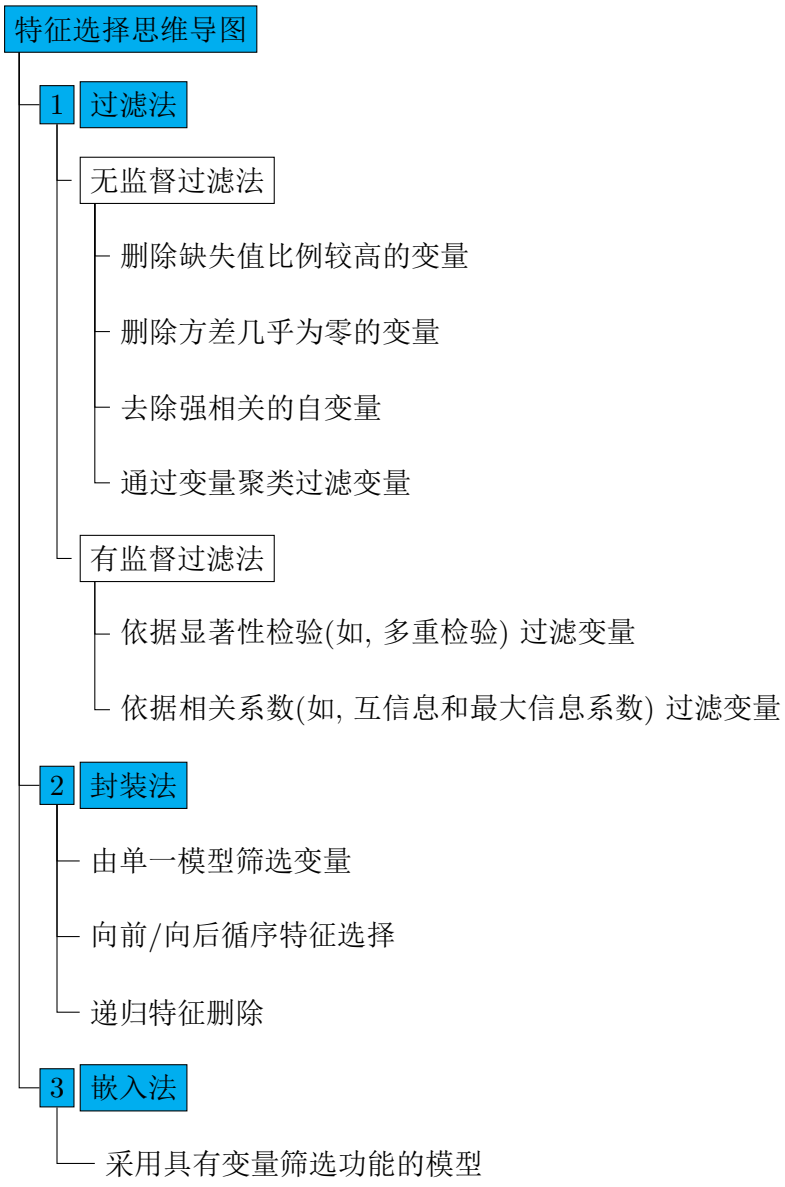
(2) 添加或删除特征是强制的, 不能根据模型效果更改.

(3) 未能穷举搜索所有的变量组合, 因而无法保证所选到的变量子集是最优的.

针对前两个问题, 循序浮动特征选择(Sequential Floating Feature Selection)等方法被提出, 允许在加入或删除变量时进一步调整变量入选的情况, 感兴趣的读者可自行学习.

本章小结

(1) 思维导图



(2) Python实现

表10.13列出了特征选择的Python函数.

表 10.13: 特征选择的Python实现

方法	函数名称与所属模块	重要参数	应用提示
删除方差几乎为零的变量	VarianceThreshold (scikit-learn库feature_selection)	threshold: 判断的阈值, 默认值为0. 可以是向量, 为每个变量设置阈值	(1) 函数不能直接处理定性变量, 需对定性变量作独热编码 (2) 考虑到方差带有单位和量纲, 建议采用最小值-最大值规范化消除单位和量纲
基于显著性检验的有监督过滤	SelectFwe/SelectFdr/SelectFpr (scikit-learn库feature_selection)	(1) score_func: 相关性检验方法. 默认值为'f_classif' (ANOVA, 用于X和Y一个为定性变量、另一个为定量变量的情况).可设置为'f_regression' (Pearson线性相关系数, 用于X和Y都为定量变量的情况); 'chi2' (列联表检验, 用于X和Y都为定性变量的情况) (2)alpha: 显著性水平, 默认值为0.05	(1) SelectFwe函数采用Bonferroni校正, 根据FWER选择变量; SelectFdr函数采用Benjamini–Hochberg方法, 根据FDR选择变量; SelectFpr函数不作校正, 按照显著性水平选择变量 (2) 函数不允许缺失值, 需处理缺失值 (3) 函数不能接受字符串, 应将字符串转换为数值 (4) 注意根据变量的类型选择恰当的相关性检验方法. 每种检验方法都有一定的假定, 例如: Pearson线性相关系数的检验要求变量满足正态性和样本观测之间的独立性; ANOVA要求定量变量满足正态性、方差齐性和样本观测之间的独立性; 列联表检验要求每一格内的频数大于等于5. 若数据不满足假定, 检验结果可能不可靠
基于相关性度量的有监督过滤	SelectKBest/SelectPercentile (scikit-learn库feature_selection)	(1) score_func: 相关性度量方法, 默认值为'f_classif' (ANOVA). 可设置为'f_regression' (Pearson线性相关系数)、'mutual_info_classif' (互信息)、'chi2' (列联表检验). 请注意根据预测变量和目标变量的类型选择恰当的统计量。 (2)k (SelectKBest中): 希望选出的自变量数量, 默认值为10 (3)percentile (SelectPercentile中): 希望选出的自变量所占比例, 默认值为10 (对应于10%)	(1) 函数不允许缺失值, 需处理缺失值 (2) 函数不能接受字符串, 应将字符串转换为数值 (3) 注意根据变量的类型选择恰当的相关性度量方法.
基于最大信息系数的有监督过滤	MINE (minepy)	(1) alpha: 影响离散化网格数量B的参数(记为α), 默认值为0.6. 当$\alpha \in (0,1]$时, $B = \max(n^{\alpha}, 4)$, 其中n为样本量. 当$\alpha > 1$时; $B = \min(\max(\alpha, 4), n)$. (2)$c$: 用于设置行变量取值数量为列变量取值数量的最大倍数, 默认值为15	-
由单一模型筛选变量	SelectFromModel (scikit-learn库feature_selection)	(1) estimator: 所使用的模型 (2) threshold: 阈值, 变量相对重要性高于该阈值的自变量会被选出. 取值为字符串或数值, 默认值为None, 对应的阈值为1e-5. 若取值为'median'和'mean', 分别表示取阈值为所有自变量相对重要性的中位数和均值 prefit: 是否采用预训练的模型进行特征选择. 默认值为False. 若设置为True, 要求模型预先进行训练	数据预处理应根据estimator所设置的模型要求进行
循序特征选择	SequentialFeatureSelector (scikit-learn库feature_selection)	(1) estimator: 所使用的模型 (2) cv: 交叉验证的设置 (3) tol: 停止规则中的阈值. 若前后两次模型效果变化量低于该阈值, 则停止添加或删除变量, 默认值为None. 仅当n_features_to_select='auto'时, tol才会发生作用 (4) n_features_to_select: 选择的自变量数量. 若n_features_to_select='auto', 当tol=None时会选出一半的变量, 当tol不为None时按照tol的指定值停止(入选的自变量数量不确定) (5) direction: 方向, 默认值为'forward', 对应向前循序特征选择. 若取值为'backward', 对应向后循序特征选择	数据预处理应根据estimator所设置的模型要求进行
递归特征删除	RFECV (scikit-learn库feature_selection)	(1) estimator: 所使用的模型 (2) cv: 交叉验证的设置	数据预处理应根据estimator所设置的模型要求进行

习题

References

[1] Reshef D.N., Reshef Y.A., Finucane H.K., Grossman S.R., McVean G., Turnbaugh P.J., Lander E.S., Mitzenmacher M., Sabeti P.C. (2011) Detecting novel associations in large datasets. Science, 334(6062): 1518 – 1524.

[2] Guyon I., Weston J., Barnhill S., Vapnik V. (2002) Gene selection for cancer classification using support vector machines. Machine Learning, 46: 389 – 422.

[3] Cardie C. (1993) Using decision trees to improve case-based learning. In Proceedings of the Tenth International Conference on International Conference on Machine Learning: 25 – 32.

[4] 美团算法团队(2018) 美团机器学习实践. 北京: 人民邮电出版社.

[5] Benjamini Y., Hochberg Y. (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: Series B (Methodological), 57(1): 289 – 300.