

# 综述. 特征选择方法

MF1733062 万晨 weanl\_jc@163.com

2018 年 4 月 10 日

## 1. 介绍

FS 在 ML 中能解决那些问题? 已经解决得怎么样了? 还有那些问题? feature construction = FS + FE 特征选择 (Feature, Variable and Attribution Selection), 是机器学习 feature construction 的重要组成部分。在筛选原始数据, 构造有用的特征集合方面, 特征选择不同于特征提取 (Feature Extraction): 后者会通过线性或非线性的方式从原始数据中构造出全新的特征, 具有特征学习和表示学习的能力 [“Representation Learning: A Review and New Perspectives”]; 特征选择通过设计一些简单高效或者精致巧妙的方法, 实现从原始特征集合中选出最优的特征子集, 能够保持特征对应的原始物理意义。所谓最优特征子集, 理论上定义为没有信息丢失的最小特征子集, 以 Markov blanket 的形式给出 [D. Koller, Toward optimal feature selection] [C. Aliferis, Local causal and markov blanket induction for causal discovery and feature selection]; 实际中理论的 ground-truth 很难找, 所以经验上一般我们用预测器性能 (如分类器的精度) 来评估特征子集的选择结果。特征选择一直以来是一个重要的课题: 为了分析特征间相关性, 早期 [Blum and Langley, 1997, Kohavi and John, 1997] 等在 1997 年提出了特征选择方面课题研究, 当时大多数应用领域下特征维数还不超过 40。后续基因序列分析和 web 文本分类等典型应用不断推动特征选择课题研究: [2001, Feature selection for high-dimensional genomic microarray data] 针对高维的染色体序列数据提出了给予特征选择的基因分析方法, [2015, Deep Feature Selection: Theory and Application to Identify Enhancer and Promoters] 将深层神经网络应用到特征选择中, 实现了基因 Enhancer 和 Promoter 的有效分析。目前特征选择研究有两大趋势: 第一, 应对各种结构化和非结构化的数据设计出一套较为通用的方法, 决策树类和深层神经网络类在特征选择方面的改进是不错的解决方法, [Feature Selection via Regularized Trees] 就是通过修改单棵树的构造算法实现基于随机森林的 Ensemble 类的特征选择方法; 第二, 应对 curse of dimensionality, 设计复杂度较低的算法, 有效地处理高维数据, 改进现有的算法以及组合使用一些简单的算法都是很好的思路。

在数据处理中, 应用特征选择方法概括起来有如下优势:

- 可以过滤无关特征: 采集过程可能引入数据噪声, 从而影响后续的数据处理; 同样与任务显著的无关特征 (irrelevant) 也可以认为是噪声, 特征选择方法一定程度上可以过滤这一部分噪声;
- 可以剔除冗余特征: 相当部分特征虽然与任务相关, 但互相之间存在显著的冗余关系 (redundant), 特征选择方法可以依据实际需求选出代表性的特征, 降低冗余;

- 可以实现特征重要性的评估：一些带有指标（如相关系数、权值）或其他“得分”的特征选择方法，可以在选出的特征子集中按指标或“得分”对特征进行重要性排序。

接下来的部分，我们会分别介绍特征选择典型的三种分类：过滤式 (filter)、包裹式 (wrapper) 和嵌入式 (embedding)。

过滤式：（线性相关系数、互信息系数、relief and relief-F、combination）

包裹式：（一般子集搜索策略）

## 2. 过滤式与包裹式

特征选择的形式化定义：首先将样本空间定义为一个  $N$  维的随机变量  $X = (X_0, X_1, \dots, X_{N-1})$ ，其服从一定的概率分布，通常认为样本实例（数据）都是该概率分布采样所得；特征全集  $S = \{X_0, X_1, \dots, X_{N-1}\}$ ，特征选择考虑的是如何有效地除去无关和冗余的特征，筛选得到“最优”特征子集  $S^* \subset S$ 。显然，暴力地遍历  $S$  所有子集会遇到组合爆炸，是不可行的。实际设计中，常采用如下思路进行搜索：

- 产生初始候选特征子集，进行下一步；
- 评价候选特征子集，如果达到终止条件终止搜索，否则进行下一步；
- 基于上一步评价结果，生成新候选特征子集，返回上一步。

由上，特征选择方法可以依据新候选特征子集的方法 (Search Strategies) 进行分类，通常有 Complete、Sequential 和 Random 等策略，但这种分类方法不是主流的划分方法。主流的分类方法是依据候选子集的评价方法 (Evaluation Criteria) 对特征选择方法进行分类，简单地说，用独立于学习器的方法进行评价的称为过滤式方法，用学习器的性能进行评价的称为包裹式方法，本章节主要介绍这两种很早就被提出的方法。下一章节我们还会介绍后续巧妙独特的嵌入式方法，不同于前两种前两类方法，嵌入式方法将特征选择嵌入到学习器的训练过程中。

### 2.1 过滤式方法

算法的同意形式：

---

**Algorithm 1** Filter Algorithm

---

**Input:**  $D(F_0, F_1, \dots, F_{N-1})$

$S_0$

**Output:**  $S_{best}$

---

算法中  $eval(S_0, D, M)$  说明过滤式采用独立的评价方法，下面介绍几种常见的方法。

#### 2.1.1 基于 pair-wise 的度量

线性相关系数、信息熵方法及 MIC、relief-F

在统计学中，常用线性相关系数 (Pearson correlation coefficient) 衡量两个随机变量间线性相关性。若定义两个随机变量  $A, B$ ，则可有  $M$  个采样  $A: a_0, \dots, a_{M-1}; B: b_0, \dots, b_{M-1}$  计算出两随机变量间的线性相关系数 (的估计值)：

$$r_{AB} = \frac{\sum_{i=0}^{M-1} (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=0}^{M-1} (a_i - \bar{a})^2} \sqrt{\sum_{i=0}^{M-1} (b_i - \bar{b})^2}}$$

$r_{AB}$  在  $[-1, 1]$  上取值，一般做如下判断：如果  $|r_{AB}| < 0.2$ ，认为  $A$  和  $B$  显著地没有线性相关性；如果  $|r_{AB}| > 0.8$ ，认为  $A$  和  $B$  显著地有线性相关性。在特征选择中一般做如下应用：令  $A = X_i, i = 0, 1, \dots, N-1, B = Y$ ，其中  $Y$  为监督学习下给出的数据标签对应的随机变量称为标签变量，满足现实的映射关系  $Y = f(X)$ ；然后计算  $r_{AB}$ ，即计算  $N$  个特征和标签变量的线性相关系数，从而剔除一些  $|r_{AB}|$  过小的特征。该方法在线性假设下能过滤一些无关特征，计算简单应用广泛。这里需要注意的是该方法由如下应用前提： $A, B$  为连续的随机变量； $A, B$  应通过正太分布检验，即大致呈正太分布。

线性相关系数给我们提供了这样一条思路：逐个考虑特征和标签变量间的关系并进行筛选特征，实际上每次计算的是两个随机变量间的关系，该类方法是 pair-wise 的，定义良好的度量关系是关键。在信息论中，基于信息熵的信息增益是常用的衡量两个随机变量可以互相预测 (predict) 性，对于随机变量  $A, B$ ，采用归一化的衡量指标 symmetrical uncertainty[?]：

$$SU(A, B) = 2 \frac{IG(A | B)}{H(A) + H(B)}$$

其中  $IG(A | B) = H(A) - H(A | B)$  为信息增益，对于  $A, B$  是对称的； $H(A) = -\sum_i P(a_i) \log_2(P(a_i))$  为  $A$  的信息熵， $H(B)$  计算类似； $H(A | B) = -\sum_j P(b_j) \sum_i P(a_i | b_j) \log_2(P(a_i | b_j))$  为观测到  $B$  的条件下  $A$  的信息熵，可以证明对于  $A, B$  是对称的。这种归一化的信息增益方法，克服了线性相关系数的较为苛刻的前提假设：不需进行正太分布检验，能很好地度量变量间的非线性关系，同时对于连续的变量可以进行离散化处理以便于计算。当然该方法通用性增加的同时，计算量明显调高了不少。

机器学习中，应对分类任务常用 Relief(Relevant Feature) 进行过滤式的特征选择 [Kira and Rendell, 1992]。延续上述思路，Relief 本质上也是一种度量关系，基于样本的分类标签，可以计算出第  $j$  个特征的度量得分：

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

????????

上述方法可以总结为基于 pair-wise 的度量，显然忽略了特征的组合影响，所以通常用来独立地过滤那些显著无关的特征，不能解决特征冗余的问题；另外这类方法天然的有度量指标，在实际中可以评估特征重要性。

### 2.1.2 组合方法

[2003, Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution] 对于分类任务的特征选择提出了基于纯过滤式方法的 FCBC Algorithm，经过精心的设计可以很好地解决特征冗余的问题。算法采用的是 (2.1.1) 中归一化的信息增益

$SU(A, B)$  作为相关关系度量, 对应记作  $SU_{A,B}$ ; 定义了主导特征 (Predominant Feature)。记特征全集  $S$ , 候选特征子集  $S^*$ , 在考虑把特征  $F_i$  加入  $S^*$  时,  $F_i$  必须为主导特征; 所谓主导特征, 首先满足  $SU_{i,C} > \delta$ , 如果还满足在  $S^*$  中不存在  $F_j$  使得  $SU_{j,i} > SU_{i,C}$ , 那么  $F_i$  是主导特征, 如果不满足第二个条件可以用存在的特征  $F_j$  构建 redundant peer 集合  $S_{P_i}$ , 然后依据算法缩减该集合, 如果能缩减为空集,  $F_i$  也是主导特征。具体算法见 [?]

如果样本的数量规模为  $N$ , 特征规模为  $M$ , FCBC 算法的时间复杂度为  $O(MN \log N)$ , 是比较快速的特征选择算法, 能很好的应用到大量的高维数据。文章用有上千维特征数据的分类任务作为算法的评估, 经验说明该精心设计的算法能取得不错的精度。

## 2.2 包裹式方法

算法的统一形式:

---

**Algorithm 2** Wrapper Algorithm

---

**Input:**  $D(F_0, F_1, \dots, F_{N-1})$

$S_0$

**Output:**  $S_{best}$

---

包裹式方法用学习器的性能对候选特征子集进行评价, 好处在于可以方便地对候选子集所有特征一起做评价, 也就是生成新的候选子集都要训练一遍学习器。显然多次训练学习器的计算开销是巨大的, 所以通常包裹式方法关注的是优化子集搜索 (新候选子集生成) 策略, 提高搜索效率。

## 3. 嵌入式