计算机模式识别与机器学习

—— 特征工程

主讲：图像处理与模式识别研究所
赵群飞

邮　　箱：zhaoqf@sjtu.edu.cn
办公室：电院 2-441
电　　话：13918191860

本章目录

11.1 相关概念

11.2 数据处理与特征构建

11.3 特征提取

11.4 特征选择

11.1 特征工程相关概念

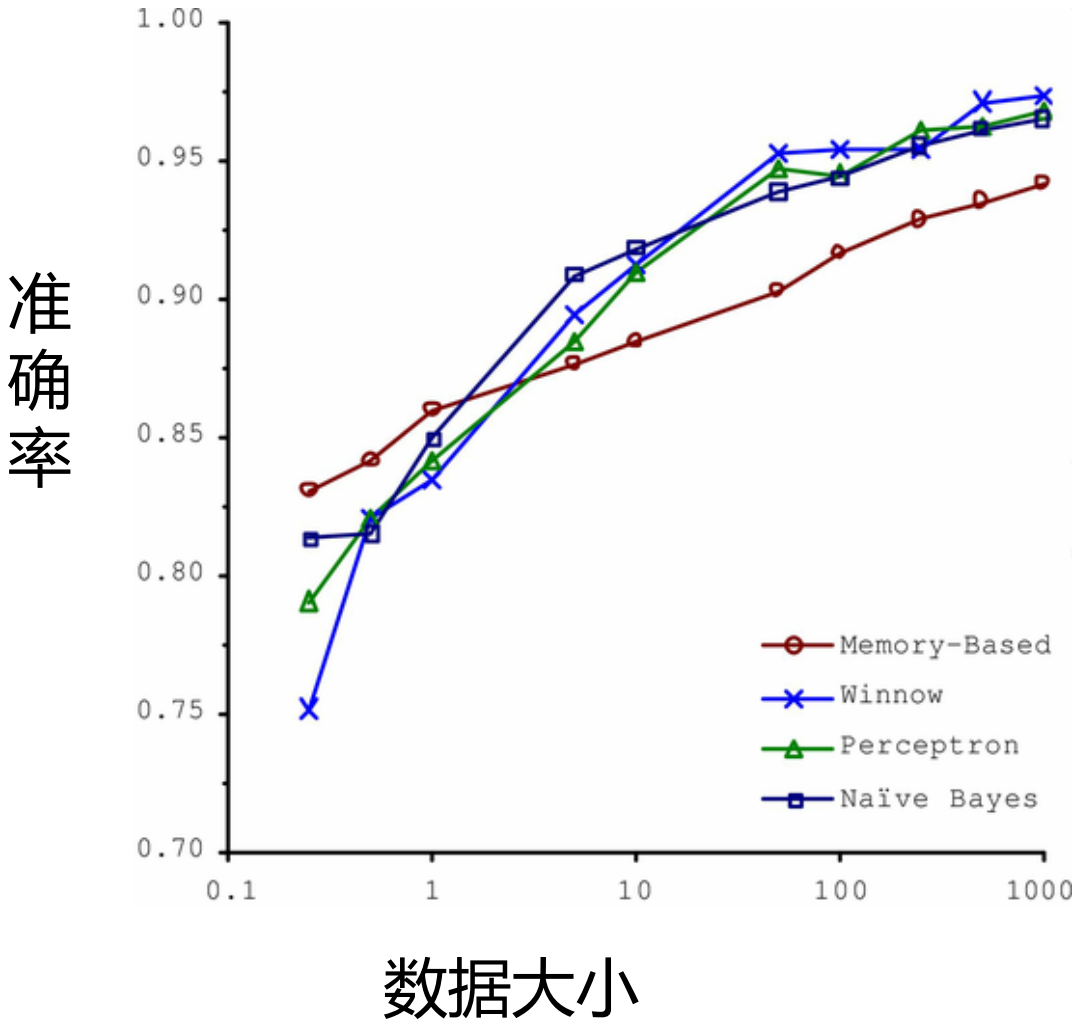
定义 → 是把**原始数据**转变为模型的**训练数据**的过程

目的 → 获取更好的训练数据特征，使得机器学习得到的模型在未知数据上表现更优

作用 →
- 提升模型的性能，直接影响模型的预测结果
- 在机器学习中占有非常重要的作用

构成 →
- 数据处理与特征构建
- 特征提取
- 特征选择

数据决定一切



通过右可以看出，各种不同算法在输入的数据量达到一定级数后，都有相近的高准确度。于是诞生了机器学习界的名言：

成功的机器学习应用不是拥有最好的算法，而是拥有最多的数据！

数据决定一切

- 但是，数据量再多，特征设计不好，或特征选择不当，机器学习的模型预测性能不一定好。

特征越好
- 灵活性越强：轻量、高速、容易维护
- 构建的模型越简单
- 模型的性能越出色

特征工程的内容

就是对数据进行处理，对特征进行进一步分析的方法。

1. 异常处理：
   - 通过箱线图（或 3-Sigma）分析删除异常值；
   - BOX-COX 转换（处理有偏分布）；
   - 长尾截断。
2. 特征归一化/标准化：
   - 标准化（转换为标准正态分布）；
   - 归一化（转换到 [0,1] 区间）；
   - 针对幂律分布，可以采用公式：$\log\left(\frac{1+x}{1+median}\right)$

3. 数据分桶：

- 等频分桶；
- 等距分桶；
- Best-KS 分桶（类似利用基尼指数进行二分类）；
- 卡方分桶。

4. 缺失值处理：

- 不处理（针对类似 XGBoost 等树模型）；
- 删除（缺失数据太多）；
- 插值补全，包括均值/中位数/众数/建模预测/多重插补/压缩感知补全/矩阵补全等；
- 分箱，缺失值一个箱。

5. 特征构建：

- 构建统计量特征，报告计数、求和、比例、标准差等；
- 时间特征，包括相对时间和绝对时间，节假日，双休日等；
- 地理信息，包括分箱，分布编码等方法；
- 非线性变换，包括 log/ 平方/ 根号等；
- 特征组合，特征交叉；
- 仁者见仁，智者见智。

6. 特征选择（筛选）

- 过滤式（filter）：先对数据进行特征选择，然后在训练学习器，常见的方法有 Relief/方差选择发/相关系数法/卡方检验法/互信息法；
- 包裹式（wrapper）：直接把最终将要使用的学习器的性能作为特征子集的评价准则，常见方法有 LVM（Las Vegas Wrapper）；
- 嵌入式（embedding）：结合过滤式和包裹式，学习器训练过程中自动进行了特征选择，常见的有 lasso 回归。

7. 降维

- PCA/ LDA/ ICA；
- 特征选择也是一种降维。

11.1 相关概念

11.2 数据处理与特征构建

11.3 特征提取

11.4 特征选择

11.2 特征构建

在原始数据集中的特征的形式，不适合直接进行建模时，使用一个或多个原特征构建新的特征。可能会比直接使用原有特征更为有效。

特征构建：是指从原始数据中人工的找出一些具有物理意义的特征。

方法：经验、属性分割和结合

操作：使用**混合属性或者组合属性**来创建新的特征，或是**分解或切分**原有的特征来创建新的特征

1. 异常值处理

- 处理数值型的数据，常用的异常值处理操作包括BOX-COX转换（处理有偏分布），箱线图分析删除异常值， 长尾截断等方式。

1）BOX-COX转换

为了使模型满足线性性、独立性、方差齐性以及正态性，需改变数据形式，所以，应用BOX-COX转换

变换公式：

$$y^{(\lambda)} = \begin{cases} \dfrac{(y+c)^{\lambda}-1}{\lambda}, if\ \lambda \neq 0 \\ log(y+c), if\ \lambda = 0 \end{cases}$$

式中$y^{(\lambda)}$为经Box-Cox变换后得到的新变量，y为原始连续因变量， λ为变换参数。

1. 异常值处理

2）其他转换非正态数据分布的方式：

- 对数转换： $y_i = ln(x_i)$
- 平方根转换： $y_i = \sqrt{x_i}$
- 倒数转换： $y_i = 1/x_i$
- 平方根后取倒数： $y_i = 1/\sqrt{x_i}$
- 平方根后再取反正弦： $y_i = arcsin(\sqrt{x_i})$
- 幂转换： $y_i = (x_i^\lambda - 1)/(\tilde{x}^{\lambda+1})$，
  其中$\tilde{x} = (\prod_{i=1}^{n} x_i)^{\frac{1}{n}}$，参数$\lambda \in [-1.5, 1]$

1. 异常值处理

3）特征异常平滑

- 基于统计的异常点检测算法 例如极差，四分位数间距，均差，标准差等，这种方法适合于挖掘单变量的数值型数据。
- 基于距离的异常点检测算法 主要通过距离方法来检测异常点，将数据集中与大多数点之间距离大于某个阈值的点视为异常点，主要使用的距离度量方法有绝对距离(曼哈顿距离)、欧氏距离和马氏距离等方法。
- 基于密度的异常点检测算法 考察当前点周围密度，可以发现局部异常点。

2. 特征归一化（最大 - 最小标准化）

1）数据归一化

使不同规格的数据转换到同一规格，将数据映射到[0,1]区间：

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

- 数据归一化的目的是使得各特征对目标变量的影响一致，会将特征数据进行伸缩变化，所以数据归一化是会改变特征数据分布的。

2）Z-Score标准化

$$x^* = \frac{x - \mu}{\sigma}$$

处理后的数据均值为0，方差为1。

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)^2$$

$$\mu = \frac{1}{m}\sum_{i=1}^{m}x^{(i)}$$

数据标准化为了不同特征之间具备可比性，经过标准化变换之后的特征分布不会发生改变。

- 当数据特征取值范围或单位差异较大时，最好做一下标准化处理。

举个例子，一个包含两个特征的数据，其中一个特征取值范围为5000~10000，另一个特征取值范围仅有0.1-1，实际在建模训练时，无论什么模型，第一个特征对模型结果的影响都会大于第二个特征，这样的模型是很难有效做出准确预测的。

3）定量特征二值化

特征的二值化处理是将数值型数据输出为布尔类型。

设定一个阈值，大于阈值的赋值为1，小于等于阈值的赋值为0。

4）定性特征哑编码

将定性数据编码为定量数据

独热编码(one-hot)，顺序性哑变量

3. 数据分桶

连续值经常离散化或者分离成“箱子”进行分析，为什么要做数据分桶呢？

- 离散后稀疏向量内积乘法运算速度更快，计算结果也方便存储，容易扩展；
- 离散后的特征对异常值更具鲁棒性，如 age>30 为 1 否则为 0，对于年龄为 200 的也不会对模型造成很大的干扰；
- LR 属于广义线性模型，表达能力有限，经过离散化后，每个变量有单独的权重，这相当于引入了非线性，能够提升模型的表达能力，加大拟合；
- 离散后特征可以进行特征交叉，提升表达能力，由M+N个变量编程M*N个变量，进一步引入非线性，提升了表达能力；
- 特征离散后模型更稳定，如用户年龄区间，不会因为用龄长了一岁就变化。

3. 数据分桶

数据分桶的方式：

- 等频分桶
- 等距分桶
- Best-KS分桶（类似利用基尼指数进行二分类）
- 卡方分桶

➢ 最好将数据分桶的特征作为新一列的特征，不要把原来的数据给替换掉。

设成绩为：[63 64 88 71 42 60 99 70 32 88 34 69 83 52 66 92 82 58 66 41]

可以按照区间分桶：

```
bins=[0,59,70,80,90,100]
score_cat = pd.cut(score_list, bins)
print(pd.value_counts(score_cat))
```

(59, 70] 7
(0, 59] 6
(80, 90] 4
(90, 100] 2
(70, 80] 1

也可以等数量分桶：

```
score_cat = pd.qcut(score_list,5)
print(pd.value_counts(score_cat))
```

(31.999, 50.0] 4
(50.0, 63.6] 4
(63.6, 69.4] 4
(69.4, 84.0] 4
(84.0, 99.0] 4

4. 缺失值处理

有些特征可能因为无法采样，或者没有观测值而缺失。

关于缺失值处理的方式， 有以下几种情况：

- 不处理（这是针对xgboost等树模型），有些模型有处理缺失的机制，所以可以不处理；
- 如果缺失的太多，可以考虑删除该列；
- 插值补全（均值，中位数，众数，建模预测，多重插补等）；
- 分箱处理，缺失值一个箱。

5. 聚合特征构建

- 聚合特征构建主要通过对多个特征的分组聚合实现，这些特征通常来自同一张表或者多张表的联立。
- 聚合特征构建使用一对多的关联来对观测值分组，然后计算统计量。
- 常见的分组统计量有中位数、算术平均数、众数、最小值、最大值、标准差、方差和频数等。

6. 转换特征构建

- 相对于聚合特征构建依赖于多个特征的分组统计，通常依赖于对于特征本身的变换。
- 转换特征构建使用单一特征或多个特征进行变换后的结果作为新的特征。
- 常见的转换方法有单调转换（幂变换、log变换、绝对值等）、线性组合、多项式组合、比例、排名编码、独热编码(OneHotEncoder)和标签编码(LabelEncoder）等。

- 在特征构建的时候，需要借助一些背景知识，一般原则就是需要发挥想象力，尽可能多的创造特征，不用先考虑哪些特征可能好，可能不好，先弥补这个广度。
- 特征构建的时候需要考虑数值特征，类别特征，时间特征。
  - 对于数值特征，一般会尝试一些它们之间的加减组合（当然不要乱来，根据特征表达的含义）或者提取一些统计特征；
  - 对于类别特征，一般会尝试之间的交叉组合，embedding也是一种思路；
  - 对于时间特征，在时间序列的预测中这一块非常重要，也会非常复杂，需要就尽可能多的挖掘时间信息，会有不同的方式技巧。

11.1 相关概念

11.2 数据处理与特征构建

11.3 特征提取

11.4 特征选择

11.3 特征提取

提取对象：原始数据（特征提取一般是在特征选择之前）

提取目的：自动地构建新的特征，将原始数据转换为一组具有明显物理意义（比如几何特征、纹理特征）或者统计意义的特征。

常用方法

- 降维方面的PCA、ICA、LDA等
- 图像方面的SIFT、Gabor、HOG等
- 文本方面的词袋模型、词嵌入模型等

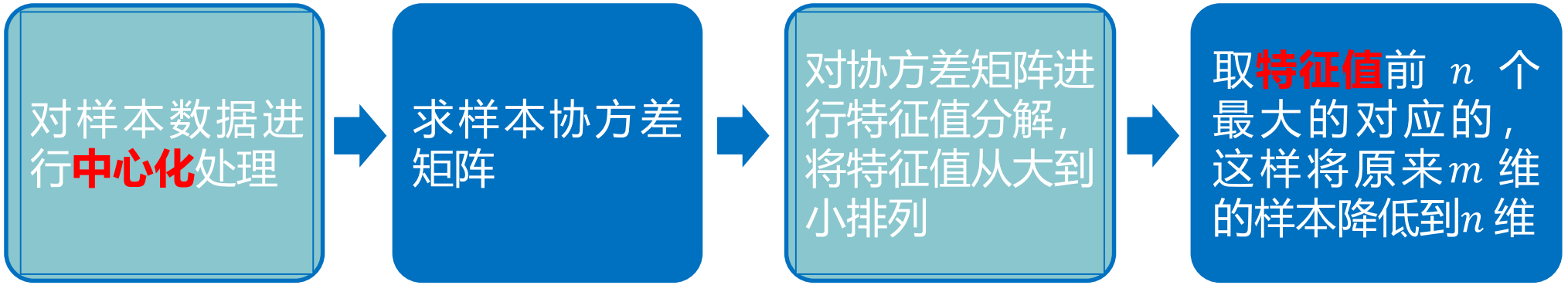
1. 降维

1）主成分分析 （PCA：Principal Component Analysis)

PCA 是降维最经典的方法，它旨在是找到数据中的主成分，并利用这些主成分来表征原始数据，从而达到降维的目的。

PCA 的思想是通过坐标轴转换，寻找数据分布的最优子空间。

步骤

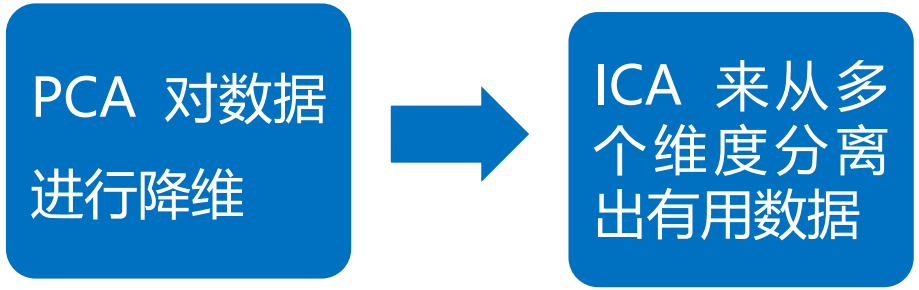


1. 降维

2）独立成分分析 (ICA：Independent Component Analysis)

ICA独立成分分析，获得的是相互独立的属性。ICA算法本质寻找一个线性变换 $z = Wx$，使得 z 的各个特征分量之间的独立性最大。

步骤



PCA 是 ICA 的数据预处理方法

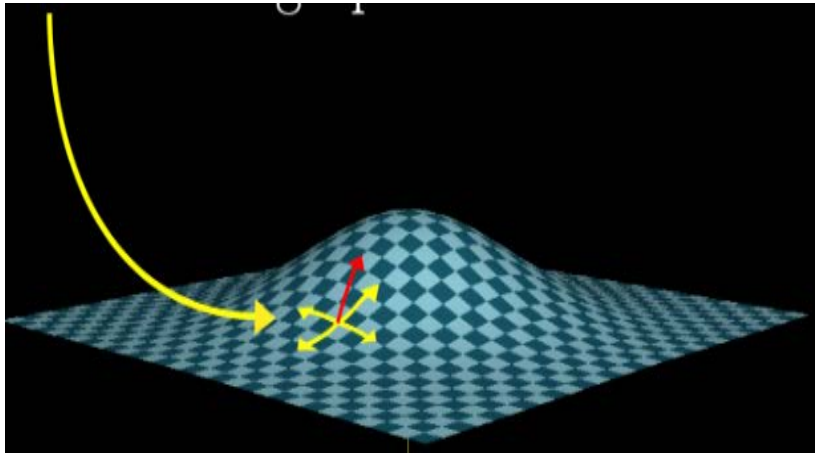
2. 图像特征提取

1） SIFT 特征

优点：

- 具有旋转、尺度、平移、视角及亮度不变性，有利于对目标特征信息进行有效表达；
- SIFT 特征对参数调整鲁棒性好，可以根据场景需要调整适宜的特征点数量进行特征描述，以便进行特征分析。

缺点：不借助硬件加速或者专门的图像处理器很难实现。



步骤

疑似特征点检测 → 去除伪特征点 → 特征点梯度与方向匹配 → 特征描述向量的生成

2） HOG特征

方向梯度直方图(HOG)特征 是针对**行人检测问题**提出的直方图特征，它通过计算和统计图像**局部区域的梯度方向**直方图来实现特征描述。

步骤

3. 文本特征提取

1） 词袋模型

将整段文本**以词为单位**切分开，然后每篇文章可以表示成一个长向量，向量的每一个维度代表一个单词，而该维度的权重反映了该单词在原来文章中的重要程度

采用 TF-IDF 计算权重，公式为 $TF - IDF(t,d) = TF(t,d) \times IDF(t)$

$TF(t,d)$ 表示单词 t 在文档 d 中出现的频率

$IDF(t)$ 是逆文档频率，用来衡量单词 t 对表达语义所起的重要性，其表示为：

$$IDF(t) = \log \frac{\text{文章总数}}{\text{包含单词}t\text{的文章总数} + 1}$$

2） N-gram 模型

- 将连续出现的 n 个词 $(n <= N)$ 组成的词组(N-gram)作为一个单独的特征放到向量表示，构成了 N-gram 模型。
- 另外，同一个词可能会有多种词性变化，但却具有相同含义，所以实际应用中还会对单词进行词干抽取(Word Stemming)处理，即将不同词性的单词统一为同一词干的形式。

11.1 相关概念

11.2 数据处理与特征构建

11.3 特征提取

11.4 特征选择

11.4 特征选择

特征选择(feature selection):

从给定的特征集合中选出相关特征子集的过程

- 原因：维数灾难问题
- 目的：确保不丢失重要的特征的条件下，去除无关特征可以降低学习任务的难度，简化模型，降低计算复杂度，使模型泛化能力更强，减少过拟合。同时，增强对特征和特征值之间的理解。

相关特征	无关特征
• 对当前学习任务有用的属性或者特征	• 对当前学习任务没用的属性或者特征

模型性能	VS	计算复杂度
• 保留尽可能多的特征，模型的性能会提升 • 但同时模型就变复杂，计算复杂度也同样提升		• 剔除尽可能多的特征，模型的性能会有所下降 • 但模型就变简单，也就降低计算复杂度

通常来说，从两个方面考虑来选择特征：

- 特征是否发散：如果一个特征不发散，例如方差接近于0，也就是说样本在这个特征上基本上没有差异，这个特征对于样本的区分并没有什么用。
- 特征与目标的相关性：很显见，与目标相关性高的特征，应当优选选择。

  - 按照特征评价标准分类：选择使分类器的错误概率最小的特征或者特征组合。
  - 利用距离来度量样本之间相似度。
  - 利用具有最小不确定性（Shannon熵、Renyi熵和条件熵）的那些特征来分类。
  - 利用相关系数，找出特征和类之间存在的相互关系。
  - 利用特征之间的依赖关系，来表示特征的冗余性加以去除。

特征选择的三种方法

过滤式(Filter): 先对数据集按照发散性或者相关性对各个特征进行评分，设定阈值或者待选择阈值的个数，进行特征选择，其过程与后续学习器无关，即设计一些统计量来过滤特征，并不考虑后续学习器问题。

包裹式(Wrapper): 根据目标函数或一个分类器，它是将后续的学习器的性能作为特征子集的评价标准，每次选择若干特征，或者排除若干特征。

嵌入式(Embedding): 先使用某些机器学习的算法和模型进行训练，得到各个特征的权值系数，根据系数从大到小选择特征。类似于Filter方法，但是是通过训练来确定特征的优劣。

1. 过滤式

原理：先对数据集进行特征选择，然后再训练学习器

特征选择过程与后续学习器无关

也就是先采用特征选择对初始特征进行过滤，然后用过滤后的特征 训练模型

优点：计算时间上比较高效，而且**对过拟合问题**有较高的鲁棒性

缺点：倾向于选择冗余特征，即没有考虑到特征之间的相关性

1）Relief 方法

定义：Relevant Features是一种著名的过滤式特征选择方法。该方法设计了一个相关统计量来度量特征的重要性。

- 该统计量是一个向量，其中每个分量都对应于一个初始特征。
- 特征子集的重要性，则是由该子集中每个特征所对应的相关统计量分量之和来决定的。
- 最终只需要指定一个阈值 k，然后选择比 k 大的相关统计量分量所对应的特征即可。也可以指定特征个数 m ，然后选择相关统计量分量最大的 m 个特征。

Relief 是为二分类问题设计的，其拓展变体 Relief-F 可以处理多分类问题。

2）方差选择法

先要计算各个特征的方差，然后根据阈值，选择方差大于阈值的特征。

3）相关系数法

先要计算各个特征对目标值的相关系数以及相关系数的 P 值。

4）卡方检验

检验定性自变量对定性因变量的相关性。假设自变量有N种取值，因变量有M种取值，考虑自变量等于 i 且因变量等于 j 的样本频数的观察值与期望的差距，构建统计量：

$$X^2 = \sum \frac{(\mathrm{A} - \mathrm{E})^2}{E}$$

5) 互信息法

概念：经典的互信息也是评价定性自变量对定性因变量的**相关性的**。

为了处理定量数据，最大信息系数法被提出。

互信息计算公式如下：

$$I(\mathrm{X};\mathrm{Y}) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$$

几个有价值的小tricks：

- 对于数值型特征，方差很小的特征可以不要，因为太小没有什么区分度，提供不了太多的信息，对于分类特征，也是同理，取值个数高度偏斜的那种可以先去掉。
- 根据与目标的相关性等选出比较相关的特征（当然有时候根据字段含义也可以选）
- 卡方检验一般是检查离散变量与离散变量的相关性，当然离散变量的相关性信息增益和信息增益比也是不错的选择（可以通过决策树模型来评估来看），person系数一般是查看连续变量与连续变量的线性相关关系。

2. 包裹式

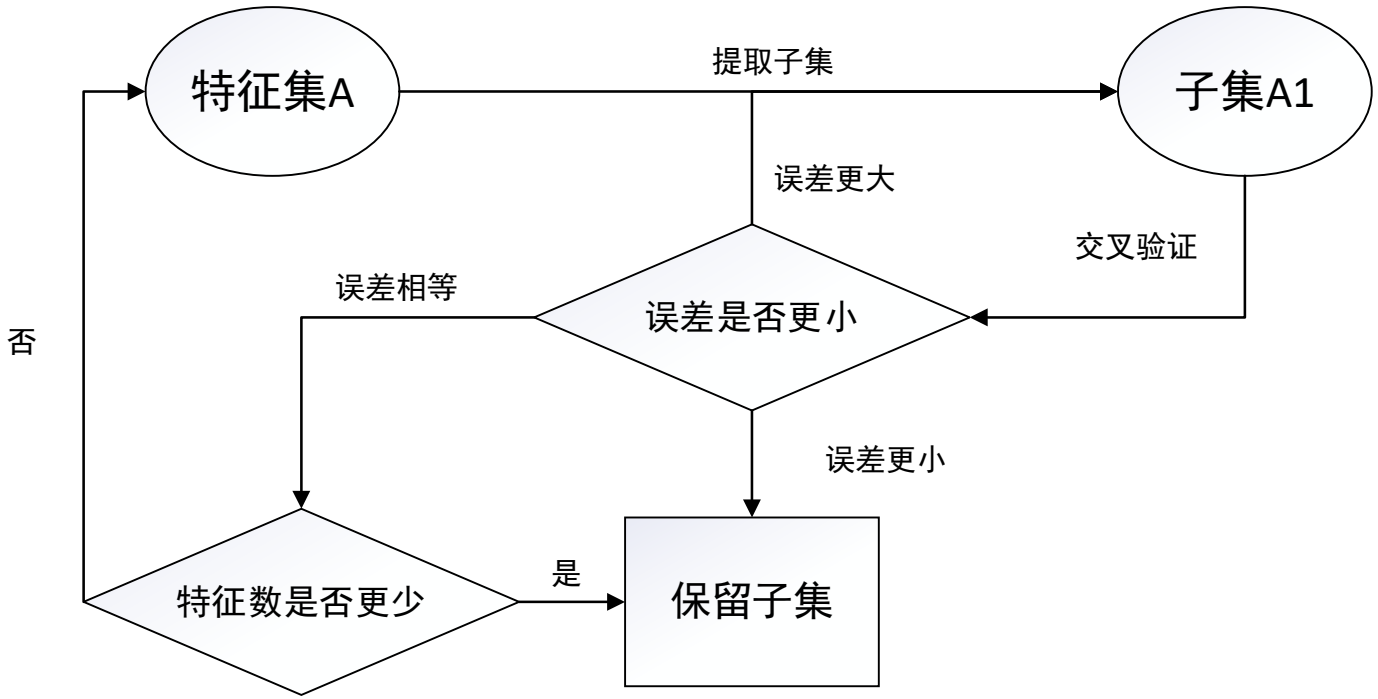
原理：包裹式特征选择**直接把最终将要使用的学习器的性能作为特征子集的评价原则**。其目的是从初始特征集合中不断的选择特征子集，训练学习器，根据学习器的性能来对子集进行评价，直到选择出最佳的子集。

优点：直接针对特定学习器进行优化，考虑到特征之间的关联性，因此通常包裹式特征选择比过滤式特征选择能训练得到一个更好性能的学习器。

缺点：由于特征选择过程需要多次训练学习器，故计算开销要比过滤式特征选择要大得多。

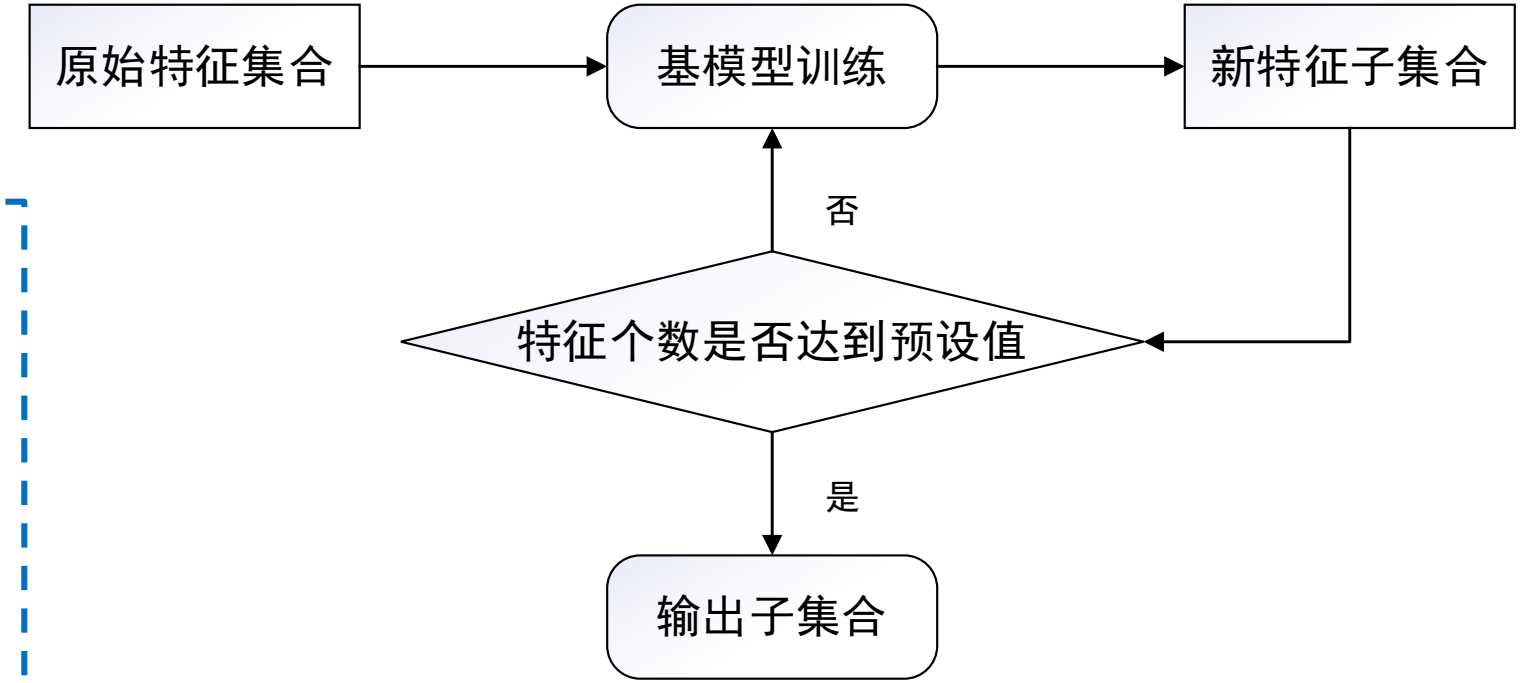
1） LVW

- Las Vegas Wrapper是一个典型的包裹式特征选择方法。使用随机策略来进行子集搜索，并以**最终分类器的误差**作为特征子集的评价标准。
- 由于 LVW 算法中每次特征子集评价都需要训练学习器，计算开销很大，因此要设计一个停止条件控制参数 T。但是如果初始特征数量很多、T 设置较大、以及每一轮训练的时间较长， 则很可能算法运行很长时间都不会停止。



2） 递归特征消除法

- 使用一个基模型来进行多轮训练，每轮训练后，消除若干权值系数的特征，再基于新的特征集进行下一轮训练。



3）嵌入式

原理：嵌入式特征选择是将特征选择与学习器训练过程融为一体，两者在同一个优化过程中完成的。即学习器训练过程中自动进行了特征选择。

常用的方法包括：

- 利用**正则化**，如L1, L2 范数，主要应用于如线性回归、逻辑回归以及支持向量机(SVM)等算法；优点：降低过拟合风险；求得的 w 会有较多的分量为零，即：它更容易获得稀疏解。
- 使用决策树思想，包括决策树、随机森林、Gradient Boosting 等。

常见的嵌入式选择模型：

在 Lasso回归 中，λ 参数控制了稀疏性：

- 如果 λ 越小，则稀疏性越小，被选择的特征越多
- 相反 λ 越大，则稀疏性越大，被选择的特征越少

在 SVM 和 逻辑回归中，参数 C 控制了稀疏性：

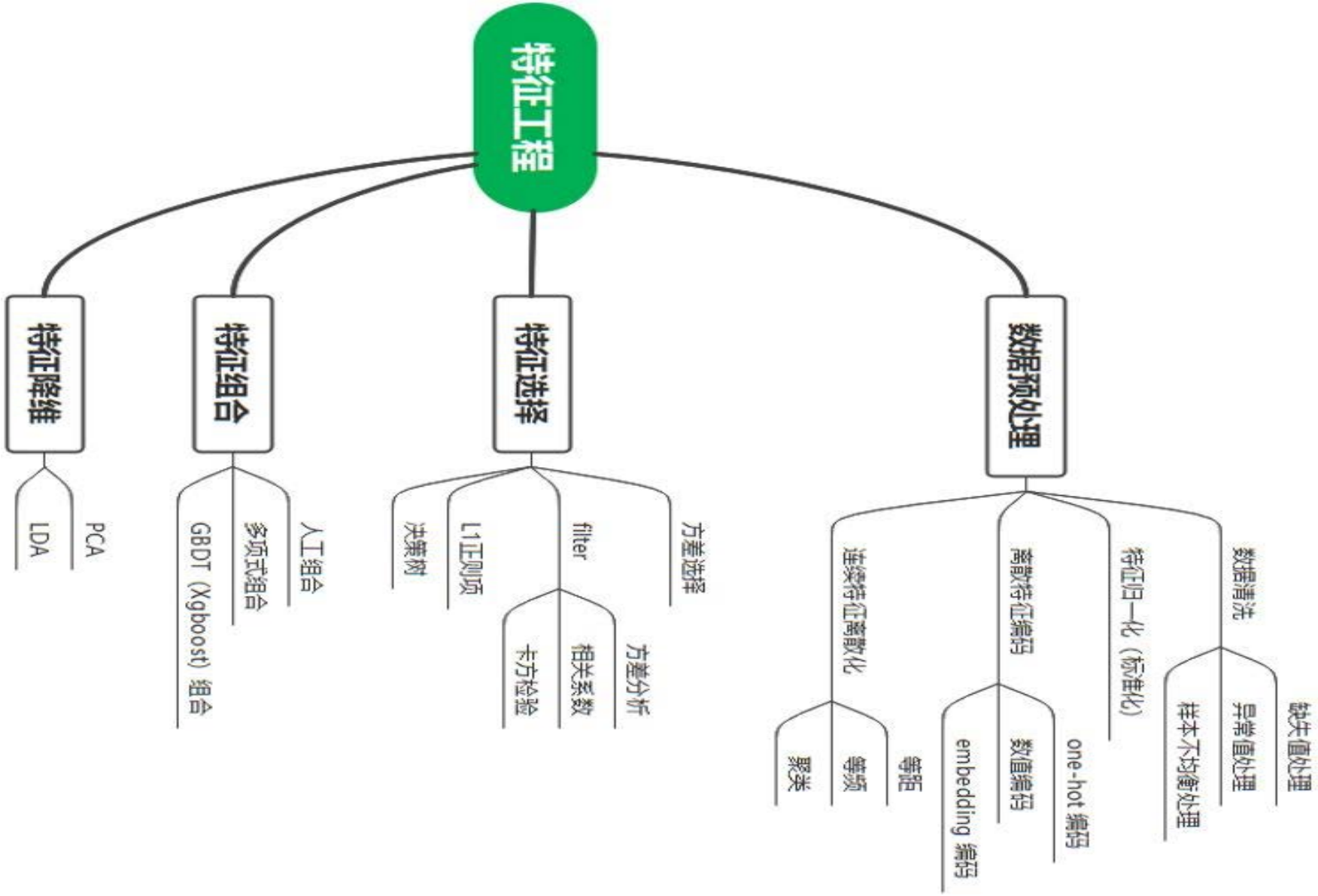
- 如果 C 越小，则稀疏性越大，被选择的特征越少
- 如果 C 越大， 则稀疏性越小，被选择的特征越多

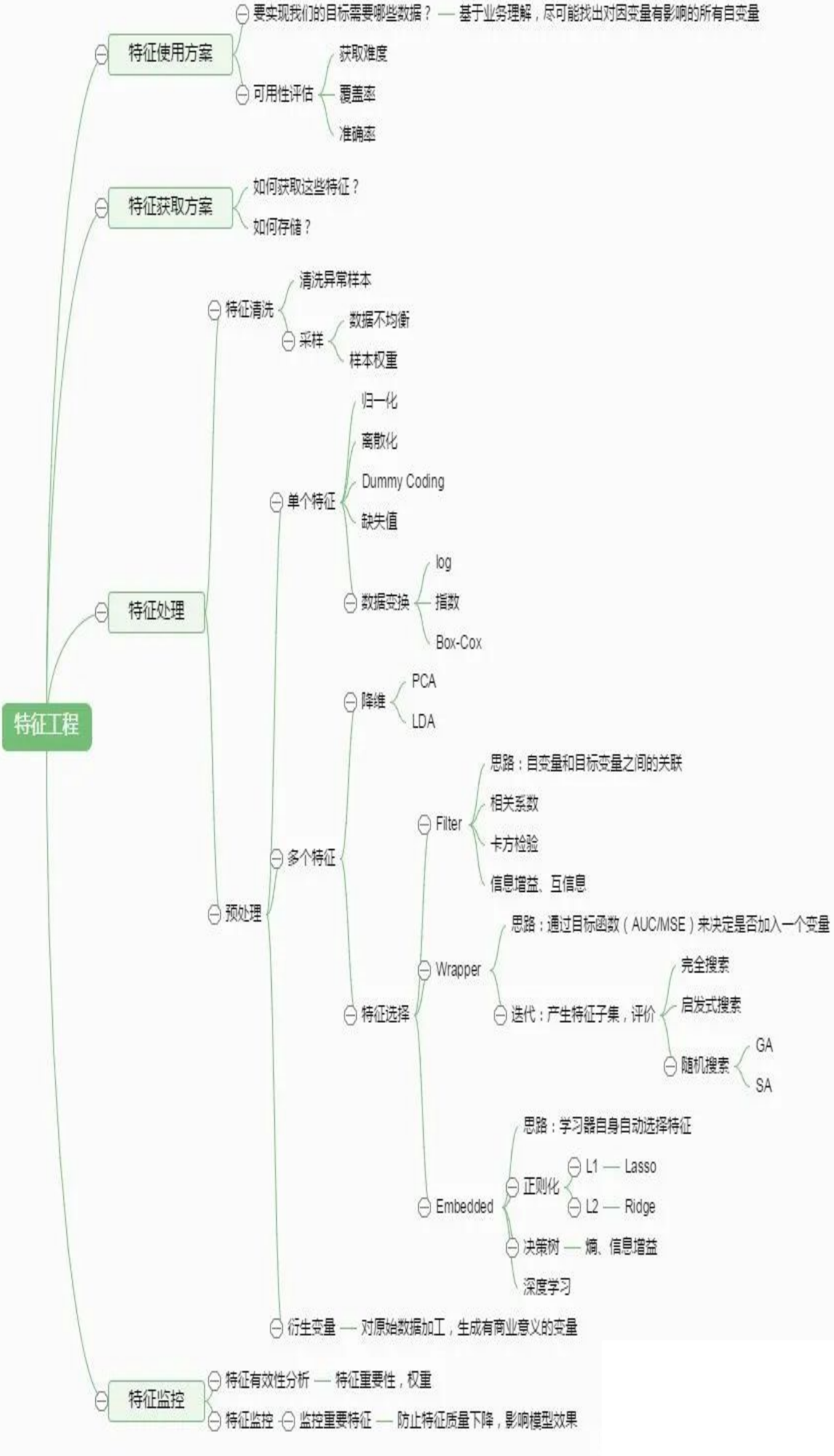
特征工程总结

特征提取VS特征选择

项目	特征提取	特征选择
共同点	都从原始特征中找出最有效的特征 都能帮助减少特征的维度、数据冗余	
区别	➢ 强调通过特征转换的方式得到一组具有明显物理或统计意义的特征 ➢ 有时能发现更有意义的特征属性	➢ 从特征集合中挑选一组具有明显物理或统计意义的特征子集 ➢ 能表示出每个特征对于模型构建的重要性

特征工程总结


特征工程
数据预处理
缺失值处理
异常值处理
样本不均衡处理
数据清洗
特征归一化（标准化）
one-hot 编码
数值编码
embedding 编码
离散特征编码
聚类
分箱
连续特征离散化
分箱
特征选择
方差选择
方差分析
相关系数
卡方检验
filter
L1正则项
决策树
特征组合
人工组合
多项式组合
GBDT (Xgboost) 组合
特征降维
PCA
LDA


特征工程
特征使用方案
要实现我们的目标需要哪些数据？— 基于业务理解，尽可能找出对因变量有影响的所有自变量
可用性评估
获取难度
覆盖率
准确率
特征获取方案
如何获取这些特征？
如何存储？
特征处理
特征清洗
清洗异常样本
采样
数据不均衡
样本权重
预处理
单个特征
归一化
离散化
Dummy Coding
缺失值
数据变换
log
指数
Box-Cox
多个特征
降维
PCA
LDA
特征选择
Filter
思路：自变量和目标变量之间的关联
相关系数
卡方检验
信息增益、互信息
Wrapper
思路：通过目标函数（AUC/MSE）来决定是否加入一个变量
迭代：产生特征子集，评价
完全搜索
启发式搜索
随机搜索
GA
SA
Embedded
思路：学习器自身自动选择特征
正则化
L1 — Lasso
L2 — Ridge
决策树 — 熵、信息增益
深度学习
衍生变量 — 对原始数据加工，生成有商业意义的变量
特征监控
特征有效性分析 — 特征重要性，权重
特征监控
监控重要特征 — 防止特征质量下降，影响模型效果

参考文献

1. https://tianchi.aliyun.com/notebook-ai/detail?spm=5176.12281978.0.0.68021b43MtXMRA&postId=95501
2. https://zhuanlan.zhihu.com/p/111296130

谢 谢！

本课件制作过程中，多处引用了国内外同行的网页、教材、以及课件PPT的内容或图片，没有随处标注，特此说明，并在此向各位作者表示感谢！