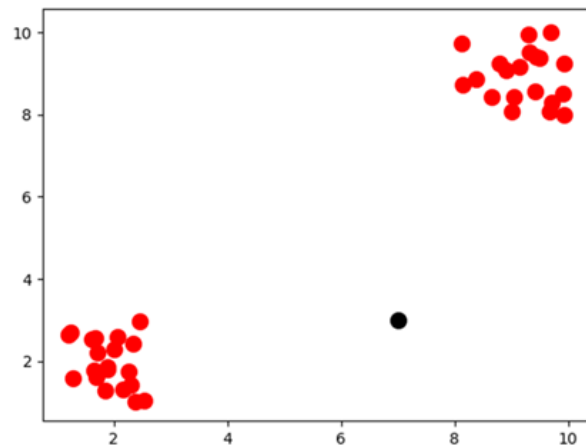


为什么要进行数据预处理

- 现实中的数据由于各种各样的原因，总会存在噪声、偏斜分布、缺失值、维数过高等问题，没有高质量的数据，再好的算法也无法得到令人满意的结果，即使算法得到的结果与数据拟合的很好也可能由于数据中的错误而使得结果的泛化能力很差。比如图中的离群点如果不能进行合适的处理就用k-means算法进行聚类，结果会有很大的偏差。



数据预处理包括哪些内容

- 数据清洗(data cleaning)
- 数据集成(data integration)
- 数据转换(data transformation)
- 数据规约(data reduction)

数据清洗(data cleaning)

数据清洗

- 因为数据仓库中的数据是面向某一主题的数据的集合，这些数据来源的时间与空间都是不同的，这样就避免不了有的数据是无关的、重复的或者错误数据(统计错误、记录错误、计算错误、存储错误等)、还有的数据相互之间有冲突，这些错误的或有冲突的数据显然是我们不想要的，称为“脏数据”。我们要按照一定的规则把“脏数据”“洗掉”，这就是数据清洗。
- 数据清洗主要是删除原始数据集中的无关数据、重复数据，处理无效值、缺失值、噪声值，处理有冲突及不一致的数据等。

无效值的处理

- 由于调查、编码和录入误差等，在数据集中可能会出现一些与期望不符的无效值，比如进行统计时，在性别一栏出现了“**asd**g”等明显无效的数值，对于这些数值可以直接删掉，将其处理为缺失值。

缺失值的处理

- 基于与出现无效值相同的原因，数据集中可能会出现缺失值。出现缺失值时可以通过直接删除的方法进行处理。
- 要删除样本首先确定样本中有多少缺失值
- 统计每一列中有多少缺失值

```
import pandas as pd
```

```
data = pd.read_csv('.\\test.csv', encoding='gbk')  
print(data.isnull().sum())
```

缺失值的处理

- 如果数据集中包含中文，可能会导致报错，需要在打开文件时加上`encoding='某种编码形式'`，与数据集中的中文使用的编码形式有关。
- 统计每一行中有多少缺失值

```
import pandas as pd
```

```
data = pd.read_csv('.\\test.csv', encoding='gbk')  
print(data.isnull().sum(axis=1))
```

- `axis = 1`表示取行，如果取0表示列，不设置参数时，默认为0。

缺失值的处理

- 如果出现缺失值的样本不是很多，即出现缺失值的行占总行数的比例不大时，可以直接删除这些行(假设在文件中每一行是一个样本)。

```
data = pd.read_csv('.\\test.csv', encoding='gbk')  
data = data.dropna()  
print(data)
```

- 如果某一系列里缺失的值很多，即某个属性缺失的值很多，这个属性就可以直接去掉。

缺失值的处理

- 根据条件删除数据：

1.删除全为空值的行：

```
data = pd.read_csv('.\\test.csv', encoding='gbk')  
data = data.dropna(how='all')  
print(data)
```

2.删除数据少于thresh个的行

```
data = pd.read_csv('.\\test.csv', encoding='gbk')  
data = data.dropna(thresh=14)  
print(data)
```

缺失值的处理

3.删除指定列包含缺失值的行:

```
data = pd.read_csv('.\\test.csv', encoding='gbk')  
data = data.dropna(subset=['商品编号'])  
print(data)
```

4.删除某一列

```
data = data.drop(columns='商品编号')  
print(data)
```

缺失值的处理

- 假如有一个数据集包含10000条数据，每条数据包含10个属性，每个属性有200条数据发生了缺失，且每条数据最多只有一个属性缺失。如果把有缺失值的数据全部删掉，数据集将会损失20%，所以这时不能再轻易删除数据。
- 在这种情况下可以对这些缺失的数值利用其他数据在这个属性上已有的数值进行填补，填补的方法主要有以下几种：

缺失值的处理

- 人工填写：比如对于缺失的、无效的性别值，可由人工根据姓名进行填写。
- 均值插补：如果该属性是数值型数据，可以根据其它数据在次属性上的值求平均；如果该属性是非数值型数据，可以取其它数据在此属性上的众数取代缺失值。
- 热卡插补：也叫就近填充，在数据集中找到与包含缺失值的数据最接近的一条或者k条数据(参考k近邻法)，以其属性(如果选择了k条数据则为均值或众数)替代缺失值。

缺失值的处理

- 使用所有可能的值填充：可使用轮盘法(仅供参考)
- 回归算法：把该属性当作标签，其他属性当作属性，执行回归算法。
- 多重替代法：用一系列可能的值来替换每一个缺失值，以反映被替换的缺失数据的不确定性。然后，对多次替换后产生的若干个数据集进行分析。最后，把来自于各个数据集的统计结果进行综合，得到总体参数的估计值。

噪声数据处理

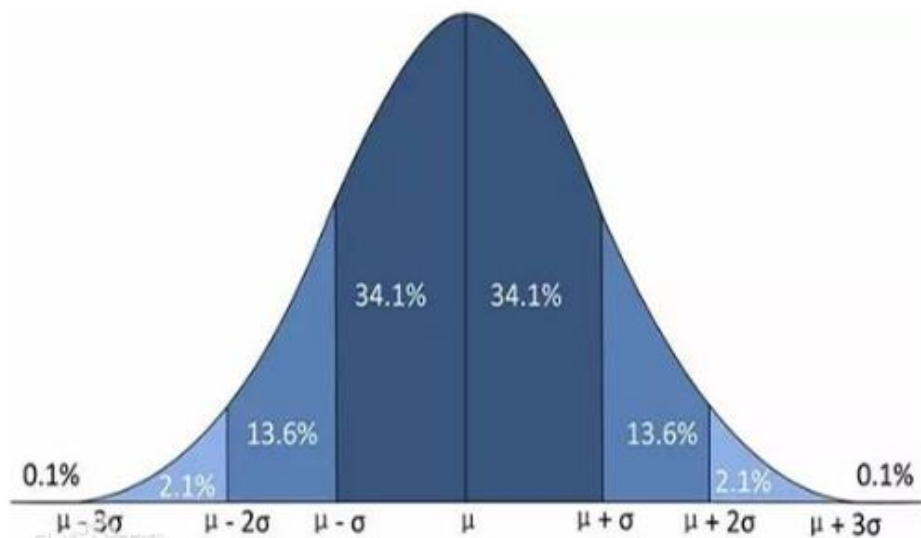
- 噪声数据是指数据中存在着错误或异常(偏离期望值)的数据，这些数据对数据的分析造成了干扰。
- 噪声数据可能来自于错误的数据收集方法、输入错误、传输错误、存储错误等。
- 噪声值在聚类时并没有多少意义，还可能使得聚类结果发生偏差，所以需要对其噪声值进行处理。
- 噪声数据的处理包括直接当作无效值删掉及对其进行平滑处理等方法。

噪声数据处理

- 处理噪声数据首先要找到噪声数据，可以首先假设数据的每个维度都服从正态分布

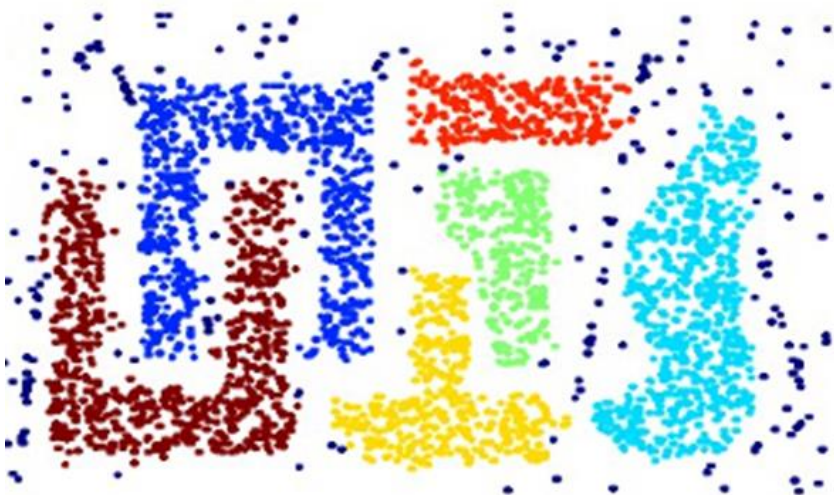
$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- 对于正态分布来说， x 落在 $(\mu - 3\sigma, \mu + 3\sigma)$ 以外的概率小于千分之三，可以认为其为噪声数据。



噪声数据处理

- 除此之外，聚类算法中的基于密度的聚类等也可以识别噪声数据。



- 当然，最重要的是根据实际情况确定如何识别噪声数据。

噪声数据处理

- 在找到哪些数据是噪声数据后，除了直接将这些数据当作无效值删除掉之外，常用分箱、回归、计算机检查和人工检查结合等方法“光滑”数据，去掉数据中的噪声。

噪声数据处理

- 分箱方法是通过对数据进行排序，利用数据“近邻”来光滑有序数据值的一种局部光滑方法。分箱方法主要有四种：等宽分箱、等深分箱、最小熵法和用户自定义区间法。分完箱后，可以使用箱均值、箱中位数或箱边界等进行光滑。箱均值光滑、箱中值光滑分别为对于每个“箱”，使用其均值或中值来代替箱中的值；而箱边界光滑则是指将给定箱中的最大值和最小值被视为箱边界，箱中每一个值都被替换为最近边界。一般而言，宽度越大，光滑效果越明显。

噪声数据处理

- 分箱方法主要有四种：等宽分箱、等深分箱、最小熵法和用户自定义区间法。
- 等宽分箱：每个箱子的取值范围一致，箱子未必是连续的。
- 等深分箱：每个箱子包含相同数目的样本数据。
- 最小熵法：参考决策树算法。
- 用户自定义区间：根据数据集的实际情况，自行确定每个箱子的宽或深。

例 1 分箱方法

- 客户收入属性排序后的值（人民币元）：800, 1000, 1200, 1500, 1500, 1800, 2000, 2300, 2500, 2800, 3000, 3500, 4000, 4500, 4800, 5000。
- 等宽分箱(宽度为1000):
 - 箱子1：800, 1000, 1200, 1500, 1500, 1800
 - 箱子2：2000, 2300, 2500, 2800, 3000
 - 箱子3：3500, 4000, 4500
 - 箱子4：4800, 5000

例 1 分箱方法

- 对于数值明显偏离正常值，不方便分箱的噪声点 (比如身高899cm)，可以通过其他属性进行分箱。
- 数据预处理是一个很灵活的过程，没有一个确定的方法可以在所有数据集上都能得到最好的结果，关键是要根据实际情况调整策略。

噪声数据处理

- 回归是指通过一个函数拟合来对数据进行光滑处理。线性回归涉及找出拟合两个变量的“最佳”直线，使得一个属性可以用来预测另一个；多元线性回归是线性回归的扩充，其中涉及的属性多于两个，并且数据拟合到一个多维曲面。如果各维度数据之间没联系或联系程度很低(比如PCA处理后的数据)，用回归方法处理噪声数据效果并不好。


有冲突的及不一致的数据处理

- 由于数据来源多样、数据存储错误、计算错误等，数据集中可能存在有冲突的及不一致的数据，比如对身高数据的调查，比如人口统计中出现了：男102，女100，合计203。对于这些数据需要根据实际情况对其进行调整。
- 数据冲突已不一致等问题有时也会被归入数据集成等预处理步骤中。

数据清洗总结

- 由于各种原因，实际中使用的数据集总会出现数据不一致、不完整、有噪声等问题，为了机器学习算法能够获得更好的效果，需要对其进行清洗。
- 数据清洗主要包括删除原始数据集中的无关数据、重复数据，处理无效值、缺失值、噪声值，处理有冲突及不一致的数据等。
- 无关数据、重复数据可直接删除。
- 无效值、噪声值可当作缺失值进行处理。

数据清洗总结

- 如果缺失值占总数据量的比例不高，可以直接删除，如果某一属性缺失值过高，则这个属性可以删除，如果有缺失值的数据很多，但缺失的属性比较分散，可以保留这些数据，通过均值插补等方法填补缺失值。
- 可以用中正态分布3  方法或者基于密度的聚类等方法确定哪些数据是噪声点，可以直接删除，也可以通过分箱方法等对噪声点进行平滑处理。
- 有冲突的数据需要根据实际情况灵活处置。

数据集成(data integration)

数据集成

- 机器学习需要的数据集可能会来自多个数据源，通过综合各数据源，将拥有不同结构、不同属性的数据整合归纳在一起，就是数据集成。
- 由于不同的数据源定义属性时命名规则不同，存入的数据格式、取值方式、单位都会有不同，获取数据时的考虑不同等原因，不同的数据源集成到一起时会出现各种问题。因此进行数据集成时需要调整属性名称、单位等，以保证数据集的整体质量。
- 数据集成应该在数据清洗之前进行。

数据集成：属性不一致问题

- 两个数据集中表示同一个属性的数据可能会由于用了不同的名称等原因导致异常，主要问题包括：

同名称属性意义不同：比如两个数据集中都有工资一项，但分别表示税前工资和税后工资。

同意义属性名字不同：比如一个数据集用身高做属性名，另一个用身長做属性名。

属性的数据类型不同：比如同样是学号，一个数据集用的int类型数据，另一个数据集用的string类型数据。

单位不同：比如身高数据一个数据集用cm，一个用m。

数据集成：属性不一致问题

存储格式不同：比如对于数字10000，一个数据集用10,000表示，另一个用 1×10^4 表示，在csv文件中逗号表示分隔，所以10,000表示两个数字。

取值范围不同：比如在mysql数据库中有的数据集属性可以取空值，有的不允许。

- 为了更好的解决这些问题，首先，需要在数据集成前，进行业务调研，确认每个属性的实际意义，不要被不规范的命名误导。其次，可以整理一张专门用来记录属性规则的表格，根据表格对所有的数据集进行考察与调整。

数据集成：属性不一致问题

属性ID	属性名	备注	数据类型	是否为空	格式	单位
001	身高	不穿鞋	float	Not null	178	cm
002	体重		float	Not null	80	kg
003	年龄		int	Not null	12	周岁
004	收入	税前	float	Not null	19000	人民币
005	身份证号		string	Not null	18位数字	

数据集成：属性冗余问题

- 数据集成时，数据集有多个来源，不同的来源最初建立的目的也会有不同，导致数据集之间可能会存在属性冗余问题。
- 比如教育局曾经对全市中学进行过两次统计，分别统计每个学校的学生情况和教师情况，这两个数据集合并时班级数量和班主任数量就几乎可以看成相同的数据。

学生数量	班级数量

教师数量	班主任数量

确定属性之间的关联程度：卡方检验

- 卡方检验是用途非常广的一种假设检验方法，它在分类资料统计推断中的应用包括：两个率或两个构成比比较的卡方检验；多个率或多个构成比比较的卡方检验以及分类资料的相关分析等。

例 2 卡方检验

	体重下降	体重未下降	合计	体重下降率
吃晚饭组	123	467	590	20.85%
不吃晚饭组	45	106	151	29.80%
合计	168	573	741	22.67%

建立假设检验：

H0：不吃晚饭对体重没有影响

H1：不吃晚饭对体重有影响

	体重下降	体重未下降	合计
吃晚饭组	134	456	590
不吃晚饭组	34	117	151
合计	168	573	741

例 2 卡方检验

- 根据卡方检验公式

$$\chi^2 = \sum \frac{(A-T)^2}{T}$$

其中A为实际值，T为理论值
得到卡方值

$$\begin{aligned}\chi^2 &= \frac{(123-134)^2}{134} + \frac{(467-456)^2}{456} + \frac{(45-34)^2}{34} + \frac{(106-117)^2}{117} \\ &= 5.76\end{aligned}$$

例 2 卡方检验

- 该题的自由度为(行数-1)×(列数-1)=1
- 查表可得H0的显著水平小于0.03，拒绝H0，说一体重与是否吃晚饭有关。

卡方检验临界值表

自由度	显著性水平 (α)					
	0.50	0.25	0.10	0.05	0.03	0.01
1	0.455	1.323	2.706	3.841	5.024	6.635
2	1.386	2.773	4.605	5.991	7.378	9.210
3	2.366	4.108	6.251	7.815	9.348	11.345
4	3.357	5.385	7.779	9.488	11.143	13.277
5	4.351	6.626	9.236	11.070	12.833	15.086
6	5.348	7.841	10.645	12.592	14.449	16.812
7	6.346	9.037	12.017	14.067	16.013	18.475
8	7.344	10.219	13.362	15.507	17.535	20.090

数据集成总结

- 数据集成时可能遇到的问题主要为属性不一致问题、属性冗余问题和数据冲突问题。
- 属性不一致问题可以建立一张专门用来记录属性规则的表格，根据表格对所有的数据集进行考察与调整。属性冗余可以通过卡方检验等方法确定属性之间的关联程度。
- 数据冲突问题需要根据实际情况灵活处置。

数据转换(data transformation)

数据转换

- 现有的数据直接使用可能不能很好的满足要求，需要进行数据转换，数据转换主要包括以下几项内容：
- 变量派生：比如销售数据统计中，数据集中统计到的是客户的身份证号，可以从中派生出客户的年龄与籍贯。
- 变量转换：例如为了改变变量的分布，让其近似高斯分布，提升模型自变量的预测能力，有时我们会对变量进行直接变换，常见的手段如下：**核函数**、取绝对值、取对数、取倒数、取指数、取平方、开平方根等。

数据转换

- 分箱转换(离散化): 分箱转换让我们把连续变量转换为离散变量, 以便开展后续分析计算工作。比如朴素贝叶斯分类中, 如果属性取值是连续的, 则任意一点的概率将会为0, 必须将其转换为离散型的。
- 标准化和归一化: PCA中为了简化协方差计算, 将所有的数据减去其期望, 称为标准化, 为了消除量纲的影响, 需要进行归一化。

- 标准化和归一化这两个名词经常有不同的含义，需要看具体情况决定，比如对于将一个向量方向不变，长度变为1的过程，有的资料叫标准化，有的资料叫归一化，还有的资料叫单位化。
- 需要明白何时该对数据做何种处理而不是死记名字。

数据规约(data reduction)

数据规约(data reduction)

- 数据规约方法类似数据集的压缩，它通过数据量的减少或者维度的减少，来达到降低数据规模的目的，方法主要是下面两种：
- 样本规约：用较小的数据表示形式替换原始数据。代表方法为对数线性回归、聚类、抽样等。比如聚类压缩图像就是用聚类进行数据规约，mini-batch k-means就是通过抽样的方式进行样本规约。
- 维度规约：通过降维方法减少数据，如PCA，SVD。

数据预处理总结

- 现实中的数据由于各种各样的原因，总会存在噪声、偏斜分布、缺失值、维数过高等问题，没有高质量的数据，再好的算法也无法得到令人满意的结果，所以在拿到数据集后需要首先进行预处理。
- 数据预处理的过程主要包括：数据集成、数据清洗、数据转换和数据规约。
- 数据集成是将多个数据集合并成一个数据集的过程，主要需要处理属性不一致问题、属性冗余问题等。

数据预处理总结

- 数据清洗包括无效值、缺失值、噪声值等的处理.
- 数据转换包括变量派生、变量转换、离散化、标准化和归一化等。
- 数据规约包括样本规约和维度规约等。