第三章 数据处理

3.1 数据预处理

数据采集后，得到的原始数据常常非常混乱、不全面，机器学习模型往往无法从中有效识别提取信息。

> - 各特征（变量）的尺度（量纲）和数量级差异大
> - 纯在噪声：包含错误和异常值
> - 存在缺失值
> - 存在冗余特征（变量）
> - ...

其中主要问题是**缺失**和**重复**

3.1.1 数据的空值填充方法基本介绍

- 5% 以内空缺可以直接删除
- 10% 以内空缺可以按照常数法填充
- 10% - 30% 以内空缺可以机器学习填充
- 30% 以上空缺删除数据列

常数填充：-1 填充；0 填充；均值填充

插值填充：各种插值方法（后文会详细介绍）

预测填充：利用机器学习算法做预测

离散数据填充：空缺值有时可以当做一个特殊类

连续数据填充：插值方法前向后向是时序性时才能用

3.1.2 异常点的判断与处理

某些情况下的数据，如：远远超过均值或低于均值的点、结果不符合常理的点，可以直接置空或删除

异常点的检测可以通过箱线图来判断：

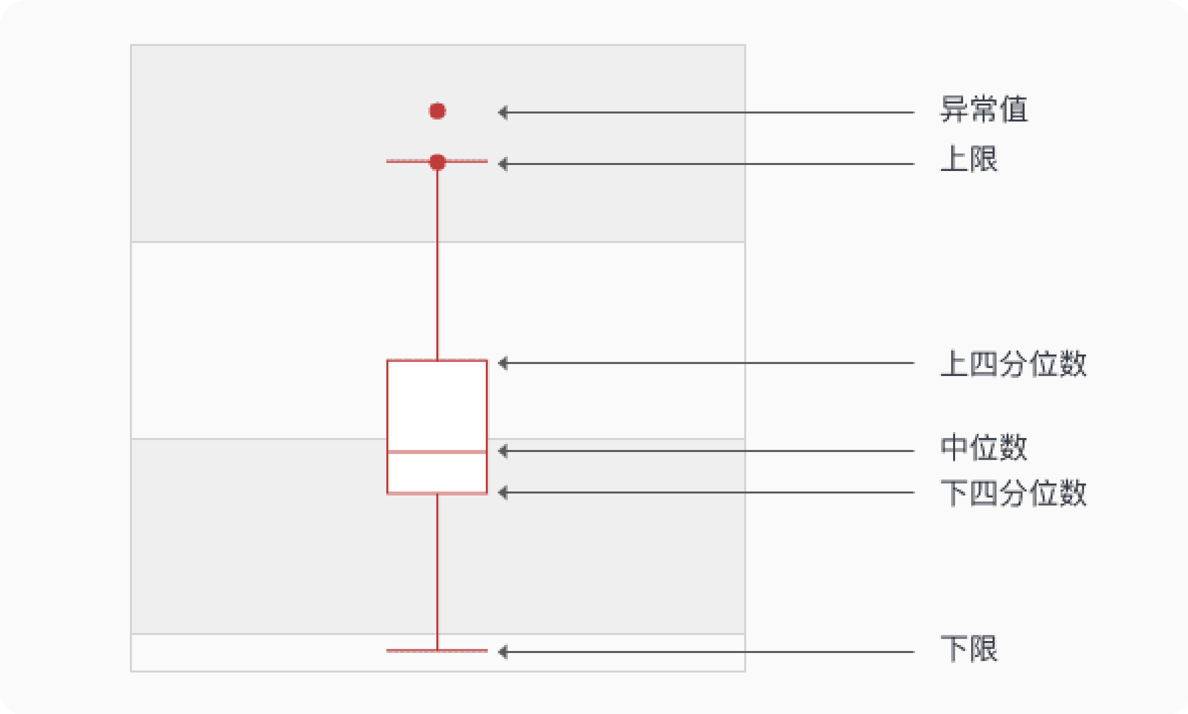

Fig. 1 箱线图

- 中位数（ Q2 / 50 th 百分位数）：数据集的中间值
- 下四分位数（ Q1 / 25 th 百分位数）：最小数（非「最小值」）和数据集中位数之间的中间值
- 上四分位数（ Q3 / 75 th 百分位数）：数据集中位数和最大数（非「最大值」）之间的中间值
- 四分位间距（ IQR ）：上下四分位数之间的距离
- 上限：Q3 + 1.5*IQR
- 下限：Q1 - 1.5*IQR

上下限之外的点即离群点（异常值），可舍去

具体内容见知乎相关文章：如何深刻理解箱线图

3.1.3　数据规约

数据规约的目的：

- 对特征的规约：抛弃冗余特征
- 对数值的规约：数据的分布有偏；数据的范围波动大

1）min - max 规约：

$$x_j^{new} = \frac{x_j - \min(X)}{\max(X) - \min(x)} \in [0, 1]$$

这里的 $\min(X)$ 表示变量 X 的最小值，$\max(X)$ 表示变量 X 的最大值，x_j^{new} 表示标准化之后的数据。

2）Z - score 规约：

$$x_j^{new} = \frac{x_j - \text{mean}(X)}{\text{std}(X)} \in R$$

这里的 $\text{mean}(X)$ 表示变量 X 的均值，$\text{std}(X)$ 表示变量 X 的标准差，x_j^{new} 表示标准化之后的数据。

3.2 插值

3.2.1 线性插值

举一个例子：我们现在有 10 号和 13 号的数据值，但缺少 11 号和 12 号的，该怎么补充呢？

线性插值就是将 10 号与 13 号数值进行连线，计算对应两点直线方程后带入 11 和 12 号横坐标的 y 值即可

$$L(x) = y_k + \frac{y_{k+1} - y_k}{x_{k+1} - x_k}(x - x_k)$$

3.2.2 拉格朗日插值

详细解释见知乎问题：如何直观地理解拉格朗日插值法？

五分钟理解拉格朗日插值法与python实现

这里只提一嘴具体公式计算（以三点为例）：

第一步，拟合函数 f 的具体求法：

$$f(x) = \sum_{i=1}^{3} y_i f_i(x) \quad \text{即} f(x) = y_1 f_1(x) + y_2 f_2(x) + y_3 f_3(x)$$

第二步，中间函数 f_i 的具体求法：

1）f_i 满足的条件

$$f_i(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

2）根据条件可以发现 f_1 的计算式为

$$f_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}$$

3）更一般的，有

$$f_i(x) = \prod_{j \neq i}^{1 \le j \le 3} \frac{x - x_j}{x_i - x_j}$$

第三步，求解其它值：将对应自变量 x 代入求出 $f(x)$ 即可

3.2.3 三次样条插值（效果较好）

假设我们已知 $x_0, x_1, ..., x_n$ 共 n+1 个点的 y 值，则可以将其分为 n 个区间 $[(x_0, x_1), (x_1, x_2), ..., (x_{n-1}, x_n)]$ ，每个区间都可以用一个三次函数进行连接。三次样条就是说每个小区间的曲线是一个三次方程。

由于每个区间中的三次方程都满足：$y = a_i + b_i x + c_i x^2 + d_i x^3$ ，所以**一共有 4n 个未知数**待求解

下面来讨论如何找到 4n 个方程：

- 首先，除了两个端点，内部 $n-1$ 个点满足 $S_i(x_{i+1}) = y_{i+1} 和 S_{i+1}(x_{i+1}) = y_{i+1}$ ，共 $2(n-1)$ 个方程，再加上两个端点分别满足第一个和最后一个方程，共有 $2n$ 个方程
- 其次，内部 $n-1$ 个点的一阶导应该连续，第 i 个区间的末点和第 $i+1$ 个区间的起点是同一个点，它们的一阶导数应该相等，即 $S_i^{'}(x_{i+1}) = S_{i+1}^{'}(x_{i+1})$ 则有 $n-1$ 个方程
- 再者，内部点的二阶导也应该连续，即 $S_i^{''}(x_{i+1}) = S_{i+1}^{''}(x_{i+1})$ 也有 $n-1$ 个方程

现在已经找到 $4n-2$ 个方程了，剩下两个方程可以利用边界条件求解。

有三种边界条件：自然边界、固定边界、非扭结边界

> - 自然边界（ Natural Spline ） 指定端点二阶导数为0，$S^{''}(x_0) = S^{''}(x_n) = 0$
> - 固定边界 （ Clamped Spline ） 指定端点一阶导数，分别定为 A 和 B 。即 $S_0^{'}(x_0) = A, \quad S_{n-1}^{'}(x_{n-1}) = B$
> - 非扭结边界 （ Not-A-Knot Spline ） 强制第一个插值点的三阶导数值等于第二个点的三阶导数值，最后第一个点的三阶导数值等于倒数第二个点的三阶导数值。即 $S_0^{'''}(x_0) = S_1^{'''}(x_1), \quad S_{n-2}^{'''}(x_{n-1}) = S_{n-1}^{'''}(x_n)$

具体解释可见：知乎 三次样条插值

其它的插值方法还有：SMOTE 插值、多维插值、多维线性插值、多维三次样条、最近邻插值、自然插值、傅里叶插值等方法

3.3 拟合

3.3.1 从最小二乘说起

高中时候，我们便已接触过一元线性回归方程的最小二乘公式：

$$\begin{cases} y = \omega x + b \\ \omega = \dfrac{\sum\limits_{i=1}^{n} x_i y_i - n\overline{xy}}{\sum\limits_{i=1}^{n} x_i^2 - n\overline{x}^2} \\ b = \overline{y} - \omega \overline{x} \end{cases}$$

那么，这个公式是怎么得到的呢？能进行扩展吗？

首先，我们的目标是找到一条直线 $y = \omega x + b$ 使得目标 y 与实际 y 偏差值最小，所以采用均方误差作为损失函数来求其最小值：

$$J(\omega, b)=\frac{1}{n}\sum_{i=1}^{n}(y_i-\widehat{y_i})^2$$

其中，$\widehat{y_i}=\omega x_i+b$，故根据二元函数求最小值的方法，我们可以计算其偏导值并令其等于 0，从而求解 ω 和 b

故有：

$$\begin{cases}\dfrac{\partial J}{\partial \omega}=\dfrac{2}{n}\sum\limits_{i=1}^{n}x_i(y_i-\omega x_i-b)=0\\ \dfrac{\partial J}{\partial b}=\dfrac{2}{n}\sum\limits_{i=1}^{n}(y_i-\omega x_i-b)=0\end{cases}$$

由此解出最小二乘公式，根据这种原理自然也能将其推广到更高阶多项式的情况

针对指对数拟合，我们可以先取 $\ln y$ 或 e^y 进行多项式拟合，再进行转化