目录

9 缺失值处理

数据不完整是现实数据中大量存在的问题, 对缺失值的分析和处理形成了统计学的一个重要分支方向. 基于数据处理的需求, 一般按照以下步骤处理缺失值:

(1) 识别缺失值;

(2) 探索缺失模式;

(3) 分析导致缺失值的原因;

(4) 处理缺失值(如, 插补缺失值);

(5) 诊断缺失值处理结果.

本章将首先概述缺失值处理, 之后具体介绍几种缺失值插补方法. 虽然缺失值插补方法占据大量篇幅, 但这并不意味着插补缺失值是最重要的一个环节, 因为缺失值处理需要多个环节有机结合, 具体处理中并无特定范式.

9.1 缺失值处理概述

9.1.1 识别缺失值

识别缺失值, 应首先判别一个值是否为缺失值, 然后分析缺失值的缺失机制.

定义9.1. 缺失值是指未观测到、但如果能观测到则将对分析有帮助的值.

根据定义, 缺失值首先是未观测到的值. 注意, "未观测到"并不意味着"未被记录"或"空缺". 例如, 在对变量取值编码时, 设置"0"表示缺失值[1], 此时取值"0"实际是缺失值, 需要将其识别为缺失值并作相应处理.

其次, 缺失值掩盖了一个有意义的值. 这意味着**未观测到但无意义的值不属于缺失值**. 一个典型的情况是"正常空缺"的值. 例如, 问卷调查中, 若被调查者去过某商场, 则填写对商场的满意度; 否则, 不必填写. 那些没有去过该商场的被调查者未填写满意度, 就属于"正常空缺". 再比如, 我们无法获取病人在死亡时点之后的追踪调查结果, 这也属于"正常空缺". "正常空缺"的值是无意义的, 因此它们不属于缺失值. 但是, "正常空缺"的值可能会影响数据分析的实施, 因此仍然需要作专门的处理, 如:

(1) 通过筛选数据, 避免"正常空缺"值的影响. 例如, 对商场满意度作分析时, 筛选出曾经去过该商场的被调查者分析满意度.

(2) 衍生替代变量. 例如, 为含有"正常空缺"值的定性变量增加类别"正常空缺"; 对于定量变量, 可通过离散化转换为定性变量, 再增加类别"正常空缺". 请注意, 新类别"正常空缺"是为变量增加的一个取值, 并非缺失值. 转换后的变量取值是完整的, 可以正常进行后续分析.

注记9.1. 若采用缺失值插补方法插补"正常空缺"值, 往往被视作过度处理. 这样的做法不是解决了问题, 反而可能引发问题, 应避免.

数据的缺失机制包括三种: 完全随机缺失、随机缺失、非随机缺失. 三者具有顺序关系, 从完全随机缺失到非随机缺失, 缺失值所造成的问题越严重, 对缺失值的处理也越困难.

[1]许多数据库系统会自动将空缺值填补为0.

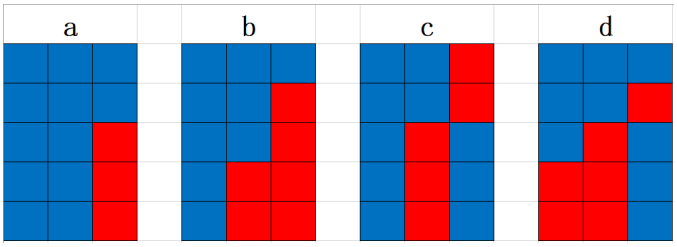


Figure 9.1: 缺失模式示意图

完全随机缺失(Missing Completely At Random, MCAR) 的特点是: 缺失值的发生与任何观测变量无关. 例如, 已知某公司招聘机制是: 应聘者有资格同时参加笔试与面试, 且面试是一次性的、不存在多个关卡. 由于面试成绩中的缺失值与笔试成绩无关, 也与面试成绩本身无关, 所以它属于MCAR. 在MCAR缺失机制下, 缺失值的产生是一个随机事件, 其分布可以看作与已观测数据的分布一致. 这类缺失容易处理, 但过于理想, 在现实中并不经常发生.

随机缺失(Missing At Random, MAR) 的特点是: 变量出现缺失的概率与其他观测变量相关, 但与该变量自身不相关. 继续以招聘为例, 假设招聘机制改为: 应聘者先参加笔试, 笔试合格后才能参加面试, 且面试是一次性的、不存在多个关卡. 由于面试成绩中的缺失值与笔试成绩有关、但与面试成绩本身无关, 所以它属于MAR. 大部分统计分析方法在随机缺失机制假定下进行缺失值处理, 此时往往需要借助于相关变量信息处理缺失值.

非随机缺失(Missing Not At Random, MNAR) 的特点是: 变量出现缺失的概率与该变量的观测值有关. 继续以招聘为例, 假设招聘机制改为: 面试有多个关卡, 上一关面试通过后才能继续参加下一关面试, 否则中途退出, 没有最终的面试成绩. 此时, 由于在前面关卡中面试成绩比较低的应聘者才会出现面试成绩缺失, 因此该缺失与面试成绩本身有关, 属于MNAR. 在非随机缺失机制下, 面临的最大问题是已观测数据的分布与缺失数据分布可能不同, 因而难以通过已观测的数据处理缺失值. 用于非随机缺失数据处理的特定方法较少, 一般需要借助其他变量信息.

可以看到, 在不同的缺失机制下, 缺失值的特点有所不同, 相应的处理也会不同. 因此, 在进行缺失值处理前, 分析缺失机制是必要的.

9.1.2 探索缺失模式

缺失模式有多种分类.

第一种分类是将缺失模式分为单变量缺失与多变量缺失. 当只有一个变

量存在缺失值时, 为单变量缺失模式; 当有两个及以上的变量存在缺失值时, 为多变量缺失模式. 图9.1中, 每一行表示一个样本观测, 每一列表示一个变量, 蓝色表示数值被观测到, 红色表示数值缺失. 可以看到, (a)图为单变量缺失模式, (b)、(c) 、(d)图为多变量缺失模式.

第二种分类是将缺失模式分为单调缺失与非单调缺失. 按照缺失数据比例从小到大的顺序将变量排序后, 若一个样本观测在某一变量上缺失, 则必定也在后续所有变量上缺失, 此为单调缺失模式. 不满足单调缺失模式的情况即为非单调缺失模式, 也称为一般缺失模式. 单调缺失模式常发生在按时间采集的纵向数据之中, 例如, 由于被调查者中途退出导致其某一时间点之后的所有数据都缺失. 图9.1中, (a)图和(b)图为单调缺失模式, (c)图和(d)图为非单调缺失模式.

第三种分类是将缺失模式分为连通缺失与非连通缺失. 对任意观测值, 若可以通过在观测值之间水平或纵向移动到达该观测值, 则称为连通缺失模式; 反之, 称为非连通缺失模式. 非连通缺失模式往往发生在两个变量无法同时被观测的情况, 例如, 试验的观测结果变量与潜在结果变量; 多源数据合并时也可能出现非连通缺失. 图9.1中, (a)-(d)图均为连通缺失模式. 注意到(c)图第二列和第三列观测值必须经由第一列连通, 若将第一列去掉, 则形成非连通缺失模式. 但若去掉(d)图中的第一列, 经由第一个样本观测的连接, 仍为连通缺失模式. 在一些统计推断中, 连通缺失模式往往是必要的. 例如, 为计算两个变量的相关系数, 则要求两个变量直接连通或二者经由其他变量连通.

探索缺失模式有助于发现产生缺失的原因, 以及数据收集过程中的问题等. 探索缺失模式还有一个重要功能: 为缺失值的后续处理, 尤其是缺失值插补, 提供依据. 可以借助一些指标实现. 记r_{ij}为第i个样本观测在第j个变量X_j上的观测情况, $r_{ij}=1$表示第i个样本观测在X_j上有观测值, $r_{ij}=0$表示第i个样本观测在X_j上缺失$(i=1,\cdots,n)$. 首先来看刻画两个变量缺失模式的指标.

(1) 采用X_k插补X_j的可用样本比例(proportion of usable cases):

$$I_{jk}=\frac{\Sigma_{i=1}^{n}(1-r_{ij})r_{ik}}{\Sigma_{i=1}^{n}(1-r_{ij})}.$$

可以看到, I_{jk}等于在X_j上缺失但在X_k上有观测值的样本数量, 除以在X_j上缺失的样本数量. 它刻画了X_k中有多大比例的观测值可用于插补X_j, 表达了变量X_j与其他变量的连通性. 我们往往希望选择信息量更大的变量用于插补, 因此I_{jk}值大的变量应优先用于插补X_j中的缺失值. 可见, I_{jk}有利于快速筛选用于插补的变量. I_{jk}是一种流入统计量(inbound statistic), 显示了其他变量可为插补X_j中缺失值所提供的信息大小. 相对应的, 也可以定义流出统计量.

(2) 变量X_j对X_k的流出统计量(outbound statistic)

$$O_{jk} = \frac{\Sigma_{i=1}^{n} r_{ij}(1 - r_{ik})}{\Sigma_{i=1}^{n} r_{ij}}.$$

O_{jk}等于在X_j上有观测值但在X_k上缺失的样本数量, 除以在X_j上有观测值的样本数量. 它刻画了X_j中有多大比例的观测值可用于插补X_k, 可评估将X_j用于插补X_k的信息量大小.

设变量维度为p, 可相应地定义刻画多个变量缺失模式的指标.

(1) 变量X_j的流入系数(influx coefficient):

$$I_j = \begin{cases} \frac{\Sigma_{k=1}^{p}\Sigma_{i=1}^{n}(1-r_{ij})r_{ik}}{\Sigma_{k=1}^{p}\Sigma_{i=1}^{n} r_{ik}}, & 若\Sigma_{k=1}^{p}\Sigma_{i=1}^{n} r_{ik} > 0 \\ 1, & 若\Sigma_{k=1}^{p}\Sigma_{i=1}^{n} r_{ik} = 0. \end{cases}$$

流入系数等于在X_j上有缺失的样本所拥有的观测值数量, 除以数据集中所有的观测值数量. 若X_j不存在缺失值, 则$I_j = 0$; 若X_j不存在观测值, 则$I_j = 1$. 若两个变量有相同的缺失值比例, 则流入系数更高的变量与其他变量的连通性能更好、更易获得好的插补值.

(2) 变量X_j的流出系数(outflux coefficient):

$$O_j = \begin{cases} \frac{\Sigma_{k=1}^{p}\Sigma_{i=1}^{n} r_{ij}(1-r_{ik})}{\Sigma_{k=1}^{p}\Sigma_{i=1}^{n}(1-r_{ik})}, & 若\Sigma_{k=1}^{p}\Sigma_{i=1}^{n}(1 - r_{ik}) > 0 \\ 1, & 若\Sigma_{k=1}^{p}\Sigma_{i=1}^{n}(1 - r_{ik}) = 0. \end{cases}$$

流出系数等于在X_j上有观测值的样本所拥有的缺失值数量, 除以数据集中缺失值总量. 若X_j不存在缺失值, 则$O_j = 1$; 若X_j不存在观测值, 则$O_j = 0$. 若两个变量有相同的缺失值比例, 则流出系数更高的变量与其他变量的连通性能更好, 能为其他变量的插补提供更多信息.

实践中, 可以从样本观测和变量两个角度分析缺失值情况, 辅助探索缺失模式. 从样本观测角度而言, 主要是探索含有缺失值的样本数量和比例, 以及单元无应答(unit nonresponse) 和项目无应答(item nonresponse) 的样本数量和比例. 单元无应答样本是指绝大部分变量信息缺失的样本, 项目无应答样本是指少数变量信息缺失的样本, 二者往往在后续处理中被区别对待[2]. 从变量角度而言, 需要探索单个变量的缺失值比例, 以及多个变量共同缺失的比例和模式等.

例9.1. 基于*R*语言中的*airquality*数据集*(*包含*Ozone*、*Solar.R*、*Wind*、*Temp*、*Month*和*Day*等*6*个变量和*153*个样本观测*)*, 采用*R*语言*mice*包中的函数探索缺失模式. 图*9.2*和*9.3*中, 红色表示缺失的情况. 图*9.2*中左图展示了各变量缺失值的比例, 可以看到,

[2]通常删除单元无应答样本, 保留项目无应答样本并作缺失值处理.

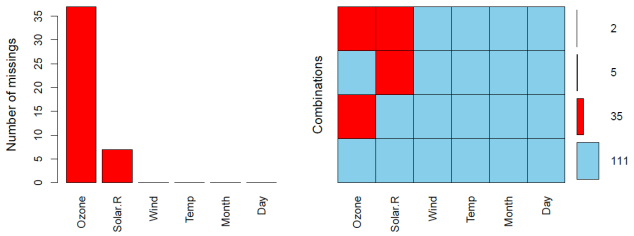


Figure 9.2: 单变量与双变量缺失模式探索

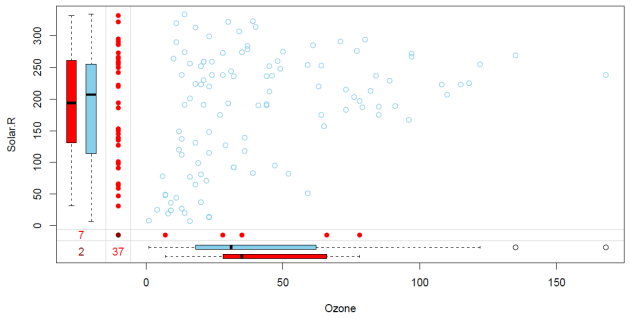


Figure 9.3: 与缺失相关的变量分布探索

*Ozone*的缺失值数量最高, 为*37*个; *Solar.R*的缺失值数量为*7*个; 其他变量不存在缺失值. 图*9.2*中右图展示了两个变量共同缺失的比例和模式, 可以看到, 有*2*个样本观测在*Ozone*和*Solar.R*上的取值同时缺失, 其他的*40*个样本观测仅在一个变量上缺失. 图*9.3*进一步展示了*Ozone*和*Solar.R*的缺失模式, 包括两个变量取值都完整的样本观测的散点图, 以及在某一变量上取值缺失的样本观测在另一变量上的取值分布情况. 例如, 横坐标方向有两张箱线图, 蓝色箱线图是基于*Solar.R*和*Ozone*两个变量都被观测到的样本数据, 展示了它们在*Ozone*变量上的分布. 红色箱线图是基于在*Solar.R*上缺失、但在*Ozone*上被观测到的*5*个样本观测, 展示了它们在*Ozone*变量上的分布. 这样的对比有助于探索含有缺失值的样本观测的分布特点.

9.1.3 分析导致缺失值的原因

分析导致缺失值的原因, 可以帮助我们判断缺失机制, 从而合理选择后续的分析方法. 某些情况下, 分析导致缺失值的原因并从中提取信息, 这本身就是数据分析的一部分. 例如, 生产设备的传感器在设备出现问题时可能产生较多的缺失值, 从而查找导致缺失值的原因有助于分析设备和生产过程出现的问题.

如何分析导致缺失值的原因呢? 我们可以从研究主题和数据收集过程入手, 也可以通过探索缺失模式或者结合业务经验等进行分析. 总之, 需要结合具体的目标和问题, 依据可获得的信息, 尽可能深入地调查分析. 当然, 也可能因信息过少无法分析导致缺失值的原因, 此时可跳过这一步骤.

9.1.4 处理缺失值

对缺失值作处理, 主要有三种策略.

第一种策略是保留缺失值. 即不作任何处理, 将数据不完整问题留给后续的分析任务. 该策略主要用于已纳入了缺失值处理的模型方法, 如, XGBoost和LightGBM等.

第二种策略是删除缺失值. 删除法主要有三种:

(1) 行删除法. 即删除包含缺失值的样本观测, 该策略主要用于单元无应答. 例如, 当因变量取值缺失时, 或者当一个样本观测超过一定比例的变量有缺失时, 可以采用行删除法. 需要注意的是, 删除样本观测后, 样本结构可能发生改变, 可以对剩余样本观测加权予以调整.

(2) 列删除法. 即删除缺失比例较高的变量.

(3) 成对删除法. 其思想可以描述为“即取即用, 用完重置”, 也就是说当分析中需要删除含缺失的样本观测时再作删除, 并且删除操作是暂时的,

不影响数据集与其他的分析. 例如, 当分析A和B两个变量的相关性时, 删除在A和B上至少存在1个缺失值的样本观测; 然后, 回到原始含有全部样本观测的数据集, 等待下一个分析任务. 这样, 可以最大化地保留数据信息.

第三种策略是转换或插补缺失值. 该策略主要用于项目无应答. 转换法是指: 对于定性变量, 增加一个类别, 表示观测值缺失; 对于定量变量, 可将其转换为定性变量, 然后将"缺失"定义为一个新的类别. 插补法是以一定的数值替换缺失值. 后续将着重介绍缺失值插补方法.

9.1.5 诊断缺失值处理结果

缺失值处理的最后一步是诊断缺失值处理结果. 缺失值处理方式不同, 诊断的内容和方法亦不同. 若采用插补法处理缺失值, 诊断的内容包括:

(1) 插补值是否越界. 如果插补值越出了数据集的取值范围, 需要诊断插补值是否符合数据逻辑. 若不符合, 应考虑修改插补值.

(2) 插补值是否产生了不一致问题[3]. 经过插补后的两个样本观测, 其自变量取值完全相同, 但因变量取值存在差异, 这就是由插补所导致的不一致问题. 可以通过编写代码检查这一问题.

(3) 插补值是否产生了取值矛盾. 例如, 对一位未曾有过购物行为的用户插补了大于0的消费额, 为一位已死亡的病人插补了大于0的当前用药剂量, 都属于取值矛盾的插补. 对这类问题的诊断需要借助于业务知识或变量间的逻辑关系.

(4) 是否因插补改变了分析结果. 诊断插补值对分析结果(如, 变量的分布、变量间的关系和目标量的估计等) 的影响, 可以有多种角度. 例如, 将含插补值与不含插补值的分析结果予以对比, 或者将同一插补方法下不同插补值的分析结果予以对比, 以及将不同的插补方法所产生的分析结果予以对比. 当插补值产生了较大影响时, 需要特别谨慎, 应审视缺失值处理是否恰当.

对插补值进行诊断的方法有: 描述插补值的分布, 比较插补值与观测值的差异, 结合后续的数据分析效果作诊断, 等等.

当一个缺失值有多个不完全相同的插补值时, 可以量化诊断缺失值和插补值对最终结果的影响. 其主要思路是通过对目标量估计值的方差作分解,

[3] 1.3节给出了不一致问题的定义

获得各次插补值对目标量估计的影响, 在此基础上构建各种指标进行量化诊断. 记$\hat{Q}$为目标量的估计值, $\boldsymbol{X}_{\text{obs}}$为观察到的数据, $\boldsymbol{X}_{\text{mis}}$为缺失的数据. 根据方差分解公式, 可将方差分解为组内方差(within variance) 和组间方差(between variance), 得到

$$V(\hat{Q}|\boldsymbol{X}_{\text{obs}}) = E[V(\hat{Q}|\boldsymbol{X}_{\text{obs}}, \boldsymbol{X}_{\text{mis}})|\boldsymbol{X}_{\text{obs}}]+V(E[\hat{Q}|\boldsymbol{X}_{\text{obs}}, \boldsymbol{X}_{\text{mis}}]|\boldsymbol{X}_{\text{obs}}). \quad (9.1)$$

(9.1)式右边第一项刻画了插补后所得目标量的平均波动; 第二项刻画了由于各次插补值不同所造成的目标量的波动, 它为诊断缺失值和插补值的影响提供了依据.

设产生了m次插补值, 第l次插补后获得的目标量估计值为$\hat{Q}_l$ ($l = 1, \cdots, m$), 最终的目标量估计值为$\hat{Q} = \sum_{l=1}^{m} \hat{Q}_l/m$. 记$B = \sum_{l=1}^{m}(\hat{Q}_l - \hat{Q})^2/(m-1)$, $\bar{U} = \sum_{l=1}^{m} \bar{U}_l/m$, $\bar{U}_l$是第l次插补中$\hat{Q}_l$的方差. 那么, 总方差$V(\hat{Q}|\boldsymbol{X}_{\text{obs}})$的估计值为:

$$T = \bar{U} + B + B/m. \quad (9.2)$$

对比(9.1)式, (9.2)式多了B/m, 这是由于对期望值$E[\hat{Q}|\boldsymbol{X}_{\text{obs}}]$作估计而额外产生了不确定性[4].

由此可以获得诊断缺失值和插补值影响的指标, 例如:

(1) 缺失导致的变异性比例(proportion of total variance due to missingness):

$$\lambda = \frac{B + B/m}{T}.$$

(2) 无响应导致的变异性相对增量(relative increase in variance due to nonresponse):

$$r = \frac{B + B/m}{\bar{U}} = \frac{\lambda}{1-\lambda}.$$

(3) 无响应导致的目标量信息缺失比例(fraction of information about Q missing due to nonresponse):

$$fmi = \frac{r + 2/(df+3)}{1+r},$$

[4](9.2)式的证明可参见Rubin (1987).

其中df为自由度, 有多种取值方法[5], 例如, 可取$df = \frac{n-k+1}{n-k+3}(n-k)(1-\lambda)$, 其中$n$为样本量, k为分析中需拟合的参数个数. 可以根据fmi的值评估缺失的影响: 当$0.2 \leq fmi < 0.3$时, 认为数据缺失问题是适中的; 当$0.3 \leq fmi < 0.5$时, 认为数据缺失问题适度地大; 当$fmi \geq 0.5$时, 认为数据缺失问题很大[6].

9.2 缺失值插补方法概述

缺失值插补(imputaion) 是缺失值处理中常用的方法. 相对于删除法而言, 插补法不但不会损失已有的数据信息, 还可以更有效地利用已有信息.

以插补的方式处理缺失值也存在弊端. 首先, 插补缺失值改变了数据, 不恰当的插补值可能有损后续的数据分析; 其次, 若借助于某些变量的信息插补缺失值, 可能导致某些变量在数据分析中的作用过度地大, 造成过拟合. 因此, 我们应该谨慎地插补缺失值.

9.2.1 缺失值插补方法分类

缺失值插补的总体思路是从缺失值的预测分布中产生插补值. 可将缺失值插补方法分为两大类.

第一类是单一插补(single imputation), 即以单一值插补一个缺失数据. 由于单一插补无法获得缺失值和插补值对目标量估计值的影响, 因此这一影响将在后续分析中被忽略, 从而可能导致系统性地低估目标量估计值的标准误.

第二类是多重插补(multiple imputation), 即以多个值插补一个缺失数据, 可通过多次执行单一插补实现. 图9.4展示了多重插补的过程, 并与单一插补作比较. 由多重插补可以获得缺失值和插补值对目标量估计的影响, 从而有助于更全面准确地评价分析结果, 9.1.5节中对缺失值处理结果的量化诊断就必须依赖多重插补. 需要注意的是, 只有当单一插补的结果在各次插补中可能发生变化时, 组合多个单一插补结果所得到的多重插补才会有意义.

下面, 仅介绍单一插补方法, 它可以分为三类.

第一类方法是冷卡插补(cold-deck imputation). 它利用历史数据、业务经验或先验信息等外部资源中的数值插补缺失值. 冷卡插补适用于各种缺失机制. 尤其是在非随机缺失机制下, 我们利用外部资源信息推断得到的插补值, 可能比由当前数据中的观测值所获得的插补值更为准确.

第二类方法是以分布的中心趋势值插补. 按照中心趋势值的获得是否基于某种条件, 可分为两种方法. 第一种方法是以非条件中心趋势值插补缺失

[5]可参看Burren (2018) 2.3.6节.

[6]借助于R语言的mice包可以计算出这些指标.

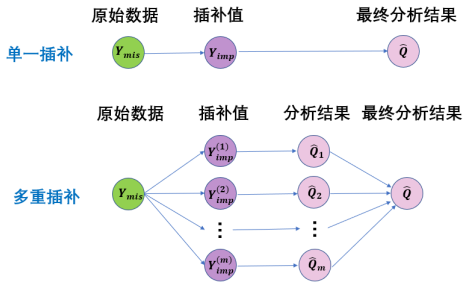


Figure 9.4: 单一插补与多重插补示意图

值, 典型的插补值是全体观测值的均值、中位数和众数等. 第二种方法是以条件中心趋势值插补缺失值. 这里的条件是比较宽泛的, 比如:

(1) 根据若干变量的取值设置条件. 若条件变量的取值(组合) 较少, 可按照条件变量对群体分层, 然后以各子群体的中心趋势值插补缺失值. 例如, 当被调查者的收入缺失时, 可以按照被调查者所在地区人群的平均收入进行插补, 这样的插补值可能比使用全体被调查者的平均收入更接近真实情况. 若条件变量的取值(组合) 较多, 可以条件变量为自变量、以待插补的变量为因变量建立模型, 由模型所得的条件均值的预测值作为插补值. 后面将要介绍的回归插补和缺失森林就属于这种方法.

(2) 以是否为近邻关系作为条件. 此时, 以近邻的中心趋势值插补缺失值, 这就是k近邻插补方法.

第三类方法是基于从预测分布中抽取的样本获得插补值, 典型的方法包括两种: 第一种方法是以模型预测值加随机误差作为插补值, 如随机回归插补和MICE; 第二种方法是基于预测分布抽样以确定邻居, 并以邻居的观测值作为插补值, 如预测均值匹配.

9.2.2 缺失值插补方法应具备的特点

理解缺失值插补方法应具备的特点, 对于选择或设计缺失值插补方法具有重要意义. 一个理想的缺失值插补方法应具备三个特点:

(1) 依据对观测变量的某种条件计算插补值. 其目的是保留待插补变量与观测变量的关联, 以降低不响应造成的偏差, 从而提升精度.

表 9.1: 常用缺失值插补方法的特点

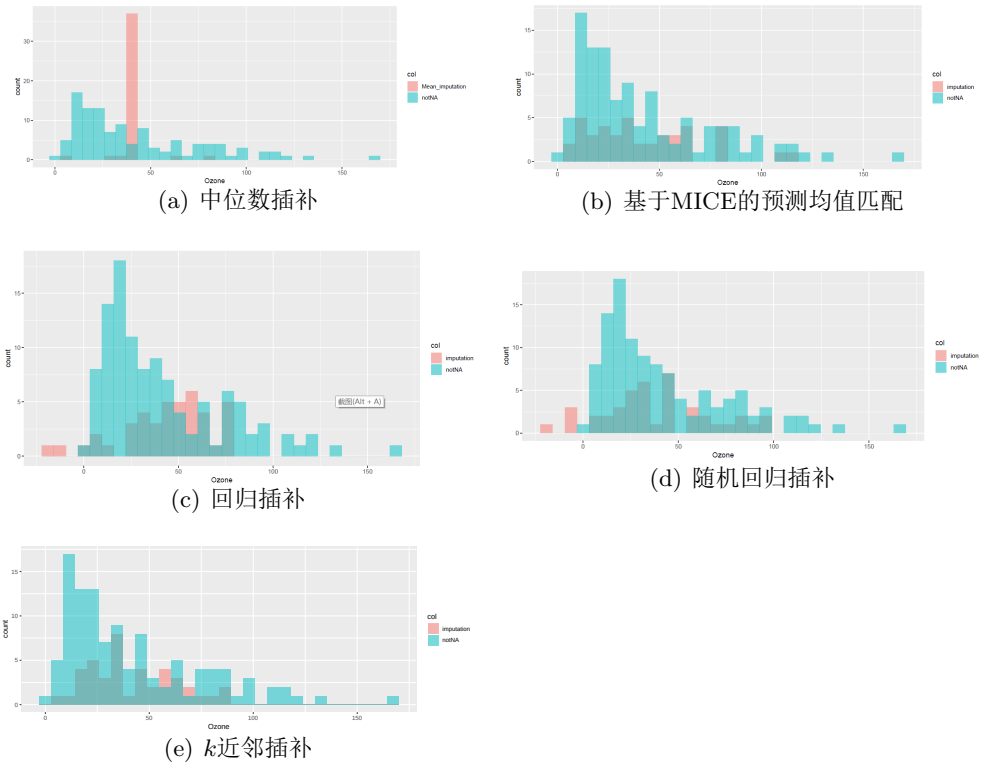
插补方法	以观测变量为条件	保留多元变量插补值之间的关联	基于抽样
以非条件中心插补	×	×	×
以群体分层后的中心插补	✓	×	×
k近邻插补	✓	×	×
回归插补	✓	×	×
随机回归插补	✓	×	✓
缺失森林	✓	✓	×
MICE	✓	✓	✓
预测均值匹配	✓	✓	✓

(2) 当多个变量有缺失值时, 保留多元变量插补值之间的关联性. 关于这一点, 我们针对基于模型的缺失值插补方法予以展开. 为插补缺失值建立模型, 包含两种建模机制. 第一种是独立建模机制. 在独立建模机制下, 将每一个待插补变量作为因变量建立一个模型, 通常排除含缺失值的样本观测, 因而各个变量的插补值相互之间不存在影响关系, 此时往往无法保留多元缺失变量之间的关联. 典型方法是回归插补和随机回归插补. 另一种是联合建模机制, 一种直观的方法是将多元缺失的变量作为因变量, 建立具有多元目标的模型, 它依赖于特定的建模方法, 建模复杂度可能较高. 联合建模机制中更为常用的一种方法采用链式建模, 即轮流以每一待插补变量为因变量建立模型, 使各变量的插补值联动加入建模之中, 将这一过程多次迭代, 直至各模型收敛. 可以看到, 联合建模机制更有利于保留多元缺失变量之间的关联, 典型的方法是MICE.

(3) 基于抽样(而非采用中心趋势值) 获取插补值. 由于抽样具有随机性, 因此从预测分布抽样可能获得不同的插补值, 从而为实现多重插补奠定基础.

表9.1展示了几种常用的缺失值插补方法是否具有以上三个特点. 可以看到, MICE和基于MICE的预测均值匹配同时满足这三个特点.

例9.2. 采用R语言中$airquality$数据集, 分别采用中位数插补、基于$MICE$的预测均值匹配、回归插补、随机回归插补和k近邻插补方法, 插补$Ozone$变



(a) 中位数插补　(b) 基于MICE的预测均值匹配

(c) 回归插补　(d) 随机回归插补

(e) k近邻插补

Figure 9.5: 基于airquality数据Ozone变量的缺失值插补

量的缺失值[7], 结果如图*9.5*所示. 图中, 以绿色直方图展示观测值的分布, 粉色直方图展示插补值的分布. 可以看到, 本例中回归插补和随机回归插补产生了越出观测值范围的插补值; 中位数插补值的分布与观测值的分布差异最大; 相对而言, 基于*MICE*的预测均值匹配和k近邻插补方法所得到的插补值分布与观测值的分布更为接近.

后续几个小节将逐一介绍几种常用的缺失值插补方法.

9.3 k近邻插补

k近邻插补方法由Troyanskaya et al. (2001) 提出, 当时是针对DNA微阵列数据中的缺失值插补问题. 该方法的主要思想是, 根据待插补样本中观测值的信息确定k个近邻, 然后利用k个近邻在待插补变量上的值插补缺失值. 近邻关系由距离决定, 距离最近的k个样本观测即为近邻.

k近邻插补的具体步骤如下:

[7]由R语言mice包中的函数实现.

步骤1: 将所有数据规范化. 这是为了保证在后续的距离运算中, 各维度的变量保持基本一致的作用. 这里, 可以将定量变量作标准化或最小值-最大值规范化处理. 对于定性变量, 应将其转换为虚拟变量.

注记9.2. 将定性变量转换为虚拟变量, 应注意两点:

(1) 虚拟变量个数应等于定性变量的类别数, 这是为了保证当两个样本观测的取值不同时都会产生相同的距离值[8].

(2) 应考虑虚拟变量取值与定量变量尺度的一致性. 根据*Stekhoven & Bühlmann (2012)*, 若采用欧氏距离, 且将定量变量作标准化变换, 则应取虚拟变量值为*-1*和*1*. 其原因是, 假定经标准化变换后的定量变量服从标准正态分布, 则约有*84%*的数据落入$[-\sqrt{2},\sqrt{2}]$, 这一范围内数据的距离平方和最大值为$\left(\sqrt{2}-(-\sqrt{2})\right)^2=8$. 若取虚拟变量值为*-1*和*1*, 定性变量对应的所有虚拟变量上距离平方和的最大值为$(1-(-1))^2\times 2=8$, 与定量变量的作用相当. 实际上, 只要大致遵从一致性的要求, 可以灵活地处理, 例如, 将定性变量变换为取值为*0*和*1*的虚拟变量, 则在欧氏距离下可将定量变量作$[0,\sqrt{2}]$的最小值-最大值规范化处理. 其中的原因, 作为习题留给读者思考.

步骤2: 找出在待插补变量上未缺失的所有样本观测, 记该集合为Q.

步骤3: 计算待插补样本观测与Q中所有样本观测的距离. 欧氏距离是较好的选择. 为解决距离运算中的缺失值问题, 可采用允许缺失值的欧氏距离, 即按照未缺失的数值计算各维度上的距离平方和, 然后乘以调节系数, 它等于总维数除以待插补样本中有观测值的维数. 例如, 考虑两个样本观测$(3,\mathrm{NaN},\mathrm{NaN},6)$和$(1,\mathrm{NaN},4,5)$, 其中NaN表示缺失值, 二者的欧氏距离为

$$\sqrt{\frac{2}{4}\times((3-1)^2+(6-5)^2)}.$$

步骤4: 找出k个近邻, 按照近邻在待插补变量上的加权中心趋势值作为插补值. 例如, 对于定性变量, 可以取近邻的加权众数作为插补值; 对于定量变量, 可以取近邻的加权平均值作为插补值. 这里的权重可以取为$1/k$, 即所有近邻等权重; 也可以采用与距离成反比的权重值, 使得更近的邻居对插补值产生更大的影响.

例9.3. 选取*golf*数据集的前*8*个样本观测, 假定其中含有*3*个缺失值, 如表*9.2*所示. 由于*Play*是后期建模分析中的因变量, 不能参与k近邻插补, 故去掉*Play*. 采用欧氏距离, 将定量变量*Temperature* 和*Humidity* 作标准化变

[8] 5.2.1节作了具体说明.

表 9.2: 含有缺失值的8个样本观测

序号	Temperature	Humidity	Windy	Outlook	Play
1	*	*	False	Sunny	No
2	80	*	True	Sunny	No
3	83	86	False	Overcast	Yes
4	70	96	False	Rainy	Yes
5	68	80	False	Rainy	Yes
6	65	70	True	Rainy	No
7	64	65	True	Overcast	Yes
8	72	95	False	Sunny	No

表 9.3: 转换后的数据

序号	ST	SH	Rainy	Sunny	Overcast	W	NW
1	*	*	-1	-1	1	1	-1
2	-1.23	*	-1	-1	1	-1	1
3	1.68	0.34	1	-1	-1	1	-1
4	-0.25	1.20	-1	1	-1	1	-1
5	-0.55	-0.17	-1	1	-1	1	-1
6	-1.00	-1.03	-1	1	-1	-1	1
7	-1.15	-1.46	1	-1	-1	-1	1
8	0.04	1.11	-1	-1	1	1	-1

换, 变换后的变量分别记为ST和SH. 将定性变量转换为取值为*-1*和*1*的虚拟变量. 将$Outlook$转换为三个虚拟变量$Rainy$, $Sunny$和$Overcast$, 将$Windy$转换为两个虚拟变量W和NW. 转换后的数据如表*9.3*所示.

首先, 插补第*1*个样本观测在$Temperature$上的缺失值. 此时, 序号为*2*至*8*的样本观测组成了备选近邻的集合Q. 采用欧氏距离, 第*1*与第*2*个样本观测的距离为:

$$\sqrt{\frac{6}{5}\left((-1-(-1))^2+(-1-(-1))^2+(1-1)^2+(1-(-1))^2+(-1-1)^2\right)} \approx 3.10.$$

其余的距离可类似计算, 结果展示在表*9.4*中. 根据距离的值, 若取近邻数量$k = 1$, 则第*8*个样本观测为第*1*个样本观测的近邻, 故插补值为*72*. 若取$k = 2$, 则第*8*个样本观测与序号为*2*至*5*中的任意一个样本观测为近邻. 例

表 9.4: 欧氏距离值

序号	2	3	4	5	6	7	8
欧氏距离	3.10	3.10	3.10	3.10	4.38	4.38	0

如, 以序号为*8*和*3*的样本观测为近邻*:* 若采用相等的权重, 则插补值为二者均值*77.5;* 若以距离倒数并作归一化后的值为权重, 由于序号为*8*的样本观测与第*1*个样本观测的距离为*0,* 而其他样本观测与第*1*个样本观测的距离大于*0,* 则序号为*8*的样本观测权重为*1,* 另一个近邻的权重为*0,* 因此插补值为*72.*

注记9.3. 当插补定性变量的缺失值时, 若将定性变量转换为虚拟变量进行处理, k近邻方法可能失效. 这是由于k近邻插补只能独立地处理各个虚拟变量, 有可能导致矛盾的插补值. 例如, 各个虚拟变量的插补值都相同, 那么这些插补值无法对应原始定性变量的某个类别.

Troyanskaya et al. (2001) 认为, 近邻数量k对插补结果影响不太大. 实践中, 一般通过交叉验证选择恰当的近邻数量k, 以实现更好的插补效果.

k近邻插补方法速度快, 在各种缺失规模的情况中都可以呈现较好的插补效果. 但是, k近邻插补方法也存在一些局限性, 包括:

(1) 高维情况下, 所找的近邻可能不准确, 从而导致插补值效果欠佳. 这是因为在高维情况下, 所有的样本观测稀疏地分布于空间中, 可能不存在真正意义上的近邻.

(2) k近邻插补未考虑缺失机制, 可能导致有偏的分析结果.

9.4 回归插补与随机回归插补

回归插补和随机回归插补都适用于待插补变量与其他变量存在关联性和预测关系的情况, 通过建立模型获得插补值. 回归插补和随机回归插补都采用独立建模机制, 即对每一变量独立进行插补, 某一变量的插补值并不影响其他变量的插补值. 在此机制下, 当建立插补模型时, 不能使用含有缺失值的样本观测.

9.4.1 回归插补

回归插补以待插补的变量为因变量建立模型, 以模型的预测值作为插补值. 插补模型可采用任意恰当的模型.

表 9.5: 对定性变量作预处理后的数据

序号	Temperature	Humidity	Windy	Rainy	Sunny
1	*	*	0	0	1
2	80	*	1	0	1
3	83	86	0	0	0
4	70	96	0	1	0
5	68	80	0	1	0
6	65	70	1	1	0
7	64	65	1	0	0
8	72	95	0	0	1

例9.4. 基于表*9.2*中的数据, 采用回归插补方法进行缺失值插补. 由于*Play*是后期建模分析中的目标变量, 不能参与插补模型, 故去掉*Play*. 本例中, 以普通最小二乘回归模型作为插补模型. 根据模型的要求, 应将定性变量转换为虚拟变量, 要特别注意避免共线性问题. 由于*Windy*只有*2*个取值, 故直接将*Windy*的取值转换为*0*和*1*, 其中*0*对应于原取值*False*, *1*对应于*True*. 另一个定性变量*Outlook*, 可任取所产生的*3*个虚拟变量中的*2*个. 例如, 取其中一个虚拟变量*Rainy*, 当*Outlook*取值为*Rainy*时, 该虚拟变量取值为*1*, 否则为*0*; 另一个虚拟变量取为*Sunny*, 取值也为*0*或*1*. 由此得到转换后的数据, 展示在表*9.5*中.

下面插补变量*Humidity*中的缺失值. 考虑两种情况.

第一种情况, 建立一个模型, 插补*Humidity*中所有缺失值. 注意到变量*Temperature*含有缺失值, 因此不能纳入模型. 基于序号为*3*至*7*的样本观测, 采用*Windy*、*Rainy*和*Sunny*构建关于*Humidity*的普通最小二乘回归模型:

$$\widehat{Humidity} = 85.14 - 19.29 \times Windy + 3.29 \times Rainy + 9.86 \times Sunny. \quad (9.3)$$

根据这一模型, 得到序号为*1*和*2*的样本观测*Humidity*插补值分别为*95.00*和*75.71*.

第二种情况, 分别根据每一个样本观测的缺失情况, 构建有针对性的模型. 先考虑序号为*1*的样本观测, 由于其*Temperature*值缺失, 故只能采用*Windy*、*Rainy*和*Sunny*的信息, 所构建的模型为(9.3)式, 插补值也为*95.00*. 再考虑序号为*2*的样本观测, 由于其*Temperature*值已被观测到, 可以采用*Temperature*、*Windy*、*Rainy*和*Sunny*的信息, 构建的最小二乘回归

模型为:

$$\widehat{Humidity} = 50.88 + 0.44 \times Temperature - 14.75 \times Windy + 6.58 \times Rainy + 12.78 \times Sunny. \tag{9.4}$$

由此得到序号为2的样本观测*Humidity*插补值为*84.11*. 可以看到, 它与第一种情况所得的插补值有所不同.

为插补变量*Temperature*中的缺失值, 应以*Temperature*为因变量, 以*Windy*、*Rainy*和*Sunny*为自变量, 基于序号为2至8的样本观测, 建立模型. 具体过程留给读者练习.

9.4.2 随机回归插补

随机回归插补的基本思想是在回归插补的基础上引入随机性, 通常是在回归模型预测值的基础上加入噪声, 将模型预测值的不确定性纳入插补值. 例如, 取噪声为服从正态分布的随机变量, 其均值为0, 方差为回归模型误差项方差的估计值.

例9.4 (续) 采用随机回归插补方法插补变量Humidity中的缺失值. 先考虑序号为1的样本观测. 模型(9.3)误差项的方差估计值$\hat{\sigma}_1^2 = 65.29$. 从$N(0, 65.29)$中随机抽取一个样本, 例如取到14.25. 从而, 序号为1的样本观测Humidity插补值为95.00 + 14.25 = 109.25. 模型(9.4)误差项的方差估计值$\hat{\sigma}_2^2 = 59.00$. 从$N(0, 59.00)$中随机抽取一个样本, 例如取到13.55, 从而, 序号为2的样本观测Humidity插补值为84.11 + 13.55 = 97.66.

回归插补与随机回归插补具有简单快速的特点. 但是, 它们也存在一些局限性, 包括:

(1) 插补的值可能越出观测值的取值范围. 对于这个问题, 解决的办法有多种. 例如, 在回归插补中, 当模型预测值落入已观测值取值范围以外时, 可以取与预测值最为接近的观测值, 也就是观测值中的最大值或最小值, 作为插补值. 在随机回归插补中, 可以基于模型预测值构造截尾型分布, 从中产生随机样本作为插补值.

(2) 建立插补模型时无法使用含有缺失值的样本观测. 当含缺失值的变量较多时, 样本信息的利用可能并不充分. 如何解决这一问题呢? 下面将要介绍的缺失森林和MICE方法或许能提供一些思路.

9.5 缺失森林

缺失森林(Miss Forest) 由Stekhoven & Bühlmann (2012) 提出. 缺失森林采用联合建模机制, 其主要思想是: 对多个含缺失值的变量逐一、轮流进行

多次迭代插补, 所采用的插补模型为随机森林(random forest). 当迭代收敛后, 以模型的预测值作为插补值. 由于随机森林模型可预测定量和定性变量, 故缺失森林方法适用于插补定量和定性数据.

记$\boldsymbol{X} = (X_1, \cdots, X_p)$为$n$行$p$列的数据矩阵. 对其中任一变量$X_s$, 记$X_s$中缺失数据的下标集合为$i_{\text{mis}}^{(s)} \subseteq \{1, \cdots, n\}$, 将数据集分为四个部分:

(1) X_s中的非缺失数据, 记为$y_{\text{obs}}^{(s)}$;

(2) X_s中的缺失数据, 记为$y_{\text{mis}}^{(s)}$;

(3) 下标属于$\{1, \cdots, n\} \backslash i_{\text{mis}}^{(s)}$的观测在除$X_s$以外变量上的数据, 记为$\boldsymbol{X}_{\text{obs}}^{(s)}$ (其中可能含有缺失值);

(4) 下标属于$i_{\text{mis}}^{(s)}$的观测在除X_s以外变量上的数据, 记为$\boldsymbol{X}_{\text{mis}}^{(s)}$.

缺失森林算法的步骤如下:

步骤1: 按缺失数据比例从小到大将所有变量排序, 记排序后的变量为$X_1, \cdots, X_p$. 显然, 未含有缺失数据的变量将排在最前面.

步骤2: 初始插补$\boldsymbol{X}$. 可以非条件中心趋势值作为初始插补值, 例如, 以均值(对定量变量) 或众数(对定性变量) 插补.

步骤3: 对每一个变量X_s $(s = 1, \cdots, p)$, 采用$\left(\boldsymbol{X}_{\text{obs}}^{(s)}, y_{\text{obs}}^{(s)}\right)$建立随机森林模型, 其中$\boldsymbol{X}_{\text{obs}}^{(s)}$为模型的自变量, $y_{\text{obs}}^{(s)}$为模型的因变量. 基于所建立的模型, 获得$\boldsymbol{X}_{\text{mis}}^{(s)}$对应的预测值$\hat{y}_{\text{mis}}^{(s)}$.

重复步骤3, 直到满足停止条件.

这里的停止条件设置规则是, 当迭代前后的插补值差异较小时, 停止迭代. Stekhoven & Bühlmann (2012) 定义了衡量插补值差异的指标. 记N为定量变量的下标集合, F为定性变量的下标集合. 对于定量的待插补变量, 定义:

$$\Delta_N = \frac{\sum_{j \in N} \sum_{i \in i_{\text{mis}}^{(s)}} \left(\hat{y}_{\text{mis},ij}^{(j)\text{new}} - \hat{y}_{\text{mis},ij}^{(j)\text{old}}\right)^2}{\sum_{j \in N} \sum_{i \in i_{\text{mis}}^{(s)}} \left(\hat{y}_{\text{mis},ij}^{(j)\text{new}}\right)^2},$$

其中$\hat{y}_{\text{mis},ij}^{(j)\text{old}}$是前一轮迭代中获得的第$i$个样本观测在变量$X_j$上的插补值, $\hat{y}_{\text{mis},ij}^{(j)\text{new}}$是其下一轮迭代中所获得的插补值. 对于定性的待插补变量, 定义:

$$\Delta_F = \frac{\sum_{j \in F} \sum_{i \in i_{\text{mis}}^{(s)}} I\left(\hat{y}_{\text{mis},ij}^{(j)\text{new}} \neq \hat{y}_{\text{mis},ij}^{(j)\text{old}}\right)}{\#\text{NA}},$$

表 9.6: 步骤2更新后的数据集

序号	Windy	Outlook	Temperature	Humidity
1	False	Sunny	71.71	82.00
2	True	Sunny	80	82.00
3	False	Overcast	83	86
4	False	Rainy	70	96
5	False	Rainy	68	80
6	True	Rainy	65	70
7	True	Overcast	64	65
8	False	Sunny	72	95

其中, $I(\cdot)$为示性函数, #NA为所有定性变量中缺失数据个数.

基于实证结果, Stekhoven & Bühlmann (2012)认为, 在对定量变量的插补中, 迭代5次即可收敛.

例9.5. 基于表*9.2*中的数据, 采用缺失森林方法插补缺失值. 由于*Play*是后期建模分析中的因变量, 不能参与建立插补模型, 故去掉*Play*.

步骤*1:* 按照缺失数据比例从小到大将变量排序. *Outlook*和*Windy*含有的缺失值数量为*0*, 应排在前两位, 二者的顺序可以任意交换. *Temperature*与*Humidity*含有的缺失值数量分别为*1*和*2*, 分别排在第*3*和第*4*位.

步骤*2:* 以非条件中心趋势值对所有缺失数据作初始插补. 由于*Temperature*与*Humidity*都是定量变量, 因此以样本均值作为初始插补值, 得到*Temperature*的初始插补值为*71.71*, *Humidity*的初始插补值为*82*. 基于初始插补值更新数据集, 如表*9.6*所示.

步骤*3:* 通过随机森林模型实施第*1*轮插补. 本例中, 设置随机森林模型的个体学习器数量为*2*, 最大树深度为*1*, 节点中抽取的分枝变量数为*2*, 且不作*Bootstrap*采样. 首先, 以第一个变量为因变量建立随机森林模型. 本例中, 以*Windy*为因变量, 以*Outlook*、*Temperature*和*Humidity*为自变量, 建立随机森林模型. 由于*Windy*中不含有缺失值, 因此, 应采用所有的样本观测建立随机森林模型. 类似地, 再按照变量排序, 分别以第二至第四个变量为因变量, 建立随机森林模型.

注记9.4. 可以看到, 以训练集中不含缺失值的变量为因变量所建立的随机森林模型, 对其他变量的插补模型和插补值并无影响. 那么, 为什么要建立这样的模型呢? 这是因为在未来的应用中, 例如应用于测试集时, 这些变量可能出现缺失值, 因此训练模型时需要为这些缺失值的插补作准备.

表 9.7: 步骤2第三个变量插补后的数据集

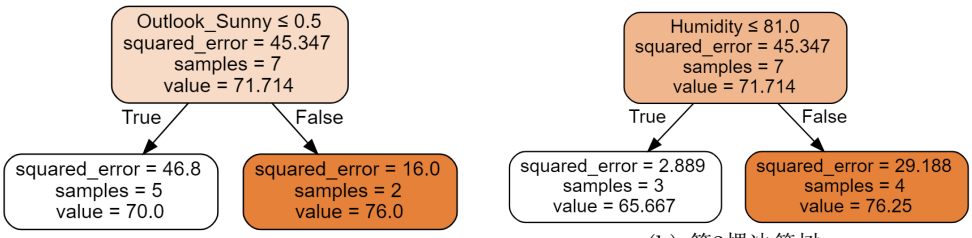
序号	Windy	Outlook	Temperature	Humidity
1	False	Sunny	76.125	82.00
2	True	Sunny	80	82.00
3	False	Overcast	83	86
4	False	Rainy	70	96
5	False	Rainy	68	80
6	True	Rainy	65	70
7	True	Overcast	64	65
8	False	Sunny	72	95

表 9.8: 步骤2第四个变量插补后的数据集

序号	Windy	Outlook	Temperature	Humidity
1	False	Sunny	76.125	79.92
2	True	Sunny	80	90.79
3	False	Overcast	83	86
4	False	Rainy	70	96
5	False	Rainy	68	80
6	True	Rainy	65	70
7	True	Overcast	64	65
8	False	Sunny	72	95

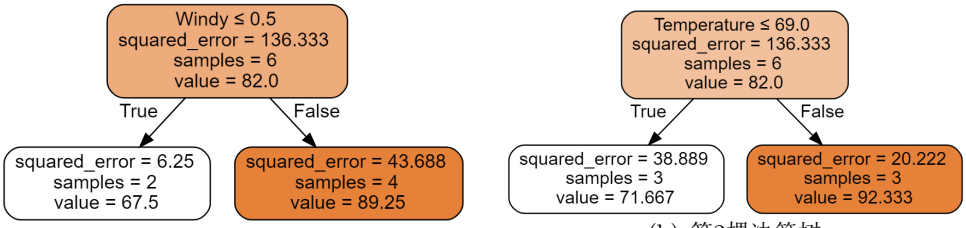
接着, 以第三个变量*Temperature*为因变量, 以*Humidiy*、*Windy*和*Outlook*为自变量, 基于表*9.6*中序号为*2*至*8*的样本观测, 建立随机森林模型. 假定所建立的随机森林模型中的两棵决策树如图*9.6*所示. 由此得到随机森林模型的预测结果为*(76+76.25)/2=76.125*, 因此序号为*1*的样本观测*Temperature*插补值应从初始插补值*71.71*更新为*76.125*. 更新后的数据集如表*9.7*所示.

最后, 以第四个变量*Humidity*为因变量, 以*Temperature*、*Windy*和*Outlook*为自变量, 基于表*9.7*中序号为*3-8*的样本观测建立随机森林模型. 假定所得到的随机森林模型中的两棵决策树如图*9.7*所示. 由此得到随机森林模型的预测结果为分别为*79.92*和*90.79*. 因此, 序号为*1*和*2*的样本观测*Humidity*插补值应从初始插补值*82*分别更新为*79.92*和*90.79*. 更新后的数据集如表*9.8*所示. 至此, 步骤*3*结束, 得到了第*1*轮插补模型和插补结果.



(a) 第1棵决策树　　(b) 第2棵决策树

图 9.6: 以Temperature为因变量的随机森林模型



(a) 第1棵决策树　　(b) 第2棵决策树

图 9.7: 以Humidity为因变量的随机森林模型

可继续执行步骤*3*, 即基于步骤*3*中第一轮迭代产生的结果*(*表*9.8*所示的数据集*)*, 继续迭代更新, 直到满足迭代停止条件. 具体迭代过程留给读者练习.

基于几个实际数据集, Stekhoven & Bühlmann (2012) 展示了缺失森林方法的效果. 从运算效率而言, 缺失森林方法的运行时长远低于MICE, 但高于k近邻插补方法. 从插补效果而言, 若以插补值与真实值之间的误差作为评价指标[9], 总体上看, 缺失森林方法在定量变量和定性变量上的缺失值插补效果优于k近邻插补和MICE.

9.6 MICE与预测均值匹配

本节介绍一种经典的缺失值插补方法——MICE, 以及基于MICE的预测均值匹配方法.

[9]正如Buuren (2021) 所述, 缺失值插补不是一个预测问题. 因此, 以插补值与真实值之间的误差作为评价指标未必恰当(当然也不失为一个评价的角度), 这一点需要注意.

9.6.1 MICE

MICE (Multivariate Imputation by Chained Equations) 这一名称来源于Buuren & Groothuis-Oudshoorn (1999), 但该方法的思想及其别名在1999年之前就已经出现了. MICE类似于缺失森林, 采用联合建模机制, 通过建立待插补变量与其他变量的模型, 逐一迭代更新单一变量的插补值, 最终实现多元变量联动的插补. 与缺失森林不同的是, MICE基于模型预测值的分布, 采用MCMC (Markov Chain Monte Carlo) 抽样产生迭代过程中的插补值.

设有p个向量$X_1,\cdots,X_p$. 记X_j中已有的观测值所构成的向量为X_j^{obs}, X_j中的缺失值在第k轮插补迭代后所形成的插补值向量为$X_j^{*(k)}$, 以$X_j^{(k)}=(X_j^{\text{obs}},X_j^{*(k)})$表示第$k$轮插补迭代后所形成的向量. 首先获得初始插补值, 从X_j^{obs}中随机抽样得到$X_j^{*(0)}$ $(j=1,\cdots,p)$. 设第$k-1$轮迭代后产生了$X_1^{(k-1)},\cdots,X_p^{(k-1)}$, 则第$k$轮迭代过程如下: 以$X_1^{\text{obs}}$为因变量, $X_{2,1}^{(k-1)},\cdots,X_{p,1}^{(k-1)}$为自变量建立模型, 其中$X_{2,1}^{(k-1)},\cdots,X_{p,1}^{(k-1)}$为$X_1^{\text{obs}}$中的样本观测在$X_2^{(k-1)},\cdots,X_p^{(k-1)}$的取值. 由此获得模型参数$\boldsymbol{\theta}_1$估计量的分布, 从该分布中随机抽取样本, 记为$\hat{\boldsymbol{\theta}}_1^{(k)}$. 然后, 将$\hat{\boldsymbol{\theta}}_1^{(k)}$代入模型中, 得到$X_1$各缺失值估计量的分布, 并从该分布中随机抽取样本, 将这些样本作为X_1的第k轮插补值, 从而获得$X_1^{*(k)}$和$X_1^{(k)}$. 继续对$X_2,\cdots,X_p$执行类似操作, 从而获得$X_1^{(k)},\cdots,X_p^{(k)}$.

需要注意的是, 建立插补模型时, 应该使用最近更新的插补值. 例如, 在第k轮中对X_2建立插补模型时, X_1已经更新到第k轮, 此时应以$X_1^{(k)}$作为自变量, 而非$X_1^{(k-1)}$. 当对所有p个变量都执行了插补值更新操作后, 就完成了第k轮迭代. 第k轮迭代过程可表示为:

$$
\begin{aligned}
\text{抽取}\hat{\boldsymbol{\theta}}_1^{(k)} &\sim P\left(\boldsymbol{\theta}_1|X_1^{\text{obs}},X_{2,1}^{(k-1)},X_{3,1}^{(k-1)}\cdots,X_{p,1}^{(k-1)}\right),\\
\text{抽取}X_1^{*(k)} &\sim P\left(X_1^{\text{mis}}|X_1^{\text{obs}},X_{2,1}^{(k-1)},\cdots,X_{p,1}^{(k-1)},\hat{\boldsymbol{\theta}}_1^{(k)}\right),\\
\text{抽取}\hat{\boldsymbol{\theta}}_2^{(k)} &\sim P\left(\boldsymbol{\theta}_2|X_2^{\text{obs}},X_{1,2}^{(k)},X_{3,2}^{(k-1)},\cdots,X_{p,2}^{(k-1)}\right),\\
\text{抽取}X_2^{*(k)} &\sim P\left(X_2^{\text{mis}}|X_2^{\text{obs}},X_{1,2}^{(k)},X_{3,2}^{(k-1)},\cdots,X_{p,2}^{(k-1)},\hat{\boldsymbol{\theta}}_2^{(k)}\right),\\
&\vdots\\
\text{抽取}\hat{\boldsymbol{\theta}}_p^{(k)} &\sim P\left(\boldsymbol{\theta}_p|X_p^{\text{obs}},X_{1,p}^{(k)},X_{2,p}^{(k)},\cdots,X_{p-1,p}^{(k)}\right),\\
\text{抽取}X_p^{*(k)} &\sim P\left(X_p^{\text{mis}}|X_p^{\text{obs}},X_{1,p}^{(k)},X_{2,p}^{(k)},\cdots,X_{p-1,p}^{(k)},\hat{\boldsymbol{\theta}}_p^{(k)}\right),
\end{aligned}
$$

将这一过程多次重复, 直至收敛. 研究发现, 一般迭代10至20次, 即可收敛.

可以通过所建立的模型相关理论获得参数估计量和预测值的分布. 但是, 许多模型(包括大部分机器学习模型) 参数估计量的分布难以获得. 此时, 可将模型参数的抽样与预测值抽样合并, 将第k轮迭代过程简化为:

$$\begin{aligned}
\text{抽取}X_1^{*(k)} &\sim P\left(X_1^{\text{mis}}|X_1^{\text{obs}}, X_{2,1}^{(k-1)}, X_{3,1}^{(k-1)}\cdots, X_{p,1}^{(k-1)}\right),\\
\text{抽取}X_2^{*(k)} &\sim P\left(X_2^{\text{mis}}|X_2^{\text{obs}}, X_1^{(k)}, X_{3,2}^{(k-1)}, \cdots, X_{p,2}^{(k-1)}\right),\\
&\vdots\\
\text{抽取}X_p^{*(k)} &\sim P\left(X_p^{\text{mis}}|X_p^{\text{obs}}, X_1^{(k)}, X_{2,p}^{(k)}, \cdots, X_{p-1,p}^{(k)}\right).
\end{aligned}$$

除了满足一定假设的特定模型(例如, 满足误差正态性、方差齐性和独立性的普通最小二乘回归模型), 许多模型预测值的精确分布不易获得. 解决此问题的一种思路是使用近似分布, 例如, 将模型预测值看作近似服从正态分布. 为避免插补值落入观测值范围以外, 可以使用截尾型分布.

MICE考虑了多元插补值之间的关联, 通过引入随机性减弱了对模型的过度依赖, 也为实现多重插补奠定了基础. MICE中建立模型所使用的自变量可以预先筛选, 以提升插补的准确度和效率.

9.6.2 基于MICE的预测均值匹配

基于模型的插补方法往往存在插补值“外推”的风险, 即插补值是一个从未在已有观测值中出现的值. 注意到, 插补值落入已有观测值的取值范围之外只是“外推”的特例, 前面已经提到, 可通过截尾型分布等方式解决这一问题. 但是, 截尾方法并不能解决插补值在取值范围内的“外推”问题. 例如, 已知某一部门仅有两种级别的员工, 且二者的级别差异较大, 工资存在断档现象, 那么工资插补值落入中间级别区域就是不合理的“外推”.

针对这一问题, 热卡插补(hot-deck imputation) 方法被提出. 其主要思想是从已观测的值中选取一个值作为缺失数据的插补值. 选取值的方法可以是在已观测到的有限个值中进行随机抽样, 也可以通过某种标准选择与缺失数据最匹配的观测值. 例如, 在k近邻插补中设置$k = 1$, 则按照距离最近的标准, 以最近的邻居取值作为插补值, 它属于热卡插补. 若在k近邻插补中采用近邻的众数作为插补值, 那么不论k的取值为多少, 插补值仍然来自于已有的观测值, 也属于热卡插补.

本节介绍热卡插补中的一种方法——预测均值匹配(Predicted Mean Matching, PMM). 该方法由Little (1988) 提出, 通过模型的均值预测值刻画匹配度, 然后以最匹配的观测值作为插补值.

在Little (1988) 中, 预测均值匹配方法与MICE搭配使用. 在MICE为框架下, 预测均值匹配首先从已有的观测值中随机抽样产生初始插补值. 下面介绍预测均值匹配如何实现某一轮迭代中对某一个缺失值的插补. 设待插补的变量为X, 用于建立插补模型的自变量为$\boldsymbol{Z}$. 设全体数据的样本量为n, 在变量X上有观测值的样本下标集合为R, 相应的观测值为x_j $(j \in R)$. 基于下标属于R的样本观测, 以X为因变量, 以$\boldsymbol{Z}$为自变量, 建立插补模型. 根据模型, 获得全体样本观测在X上的预测值$\hat{\mu}_j$, 它是对均值μ_j的预测$(j = 1, \cdots, n)$. 则预测均值匹配方法的插补值为:

$$x_i^* = x_{k_i},\ i \in \bar{R}, \tag{9.5}$$

其中$k_i = \text{argmin}_{s \in R} \left(\tilde{\mu}_i - \hat{\mu}_s\right)^2$, $\tilde{\mu}_i$是从预测值$\hat{\mu}_i$所服从的分布中随机抽取的一个样本.

可以看出, 预测均值匹配与MICE搭配使用, 具有MICE的特点; 同时, 通过从已有观测值中选取插补值, 预测均值匹配可以减弱对插补模型预测值分布的依赖, 并且较好地解决了插补值"外推"问题.

9.6.3 基于普通最小二乘回归模型的特例*

本节针对满足误差正态性、方差齐性和独立性的普通最小二乘回归模型, 介绍如何结合参数估计值和模型预测值的分布实施MICE和基于MICE的预测均值匹配.

设自变量维数为q, 样本量为m, 建立的普通最小二乘回归模型为

$$\hat{\boldsymbol{X}}_{m\times 1} = \boldsymbol{Z}_{m\times(q+1)} \hat{\boldsymbol{\theta}}_{(q+1)\times 1},$$

其中$\hat{\boldsymbol{\theta}}_{(q+1)\times 1}$的第一个维度为截距项, 数据矩阵$\boldsymbol{Z}_{m\times(q+1)}$的第一列是取值全为1的向量. 例如, 对于含有$p$维变量$(X_1, \cdots, X_p)$的数据集, 应用MICE对$X_j$作插补时, 以$X_j$为因变量, $\left(X_1^{(k)}, \cdots, X_{j-1}^{(k)}, X_{j+1}^{(k-1)}, \cdots, X_p^{(k-1)}\right)$为自变量, 建立普通最小二乘回归模型, 系数$\hat{\boldsymbol{\theta}}_j^{(k)}$对应于回归系数的估计值.

当模型误差满足正态性、方差齐性和独立性时, 根据普通最小二乘回归模型理论, 有

$$\hat{\boldsymbol{\theta}} \sim N\left(\boldsymbol{\theta}, \text{COV}\left(\hat{\boldsymbol{\theta}}\right)\right),$$

以及

$$\hat{X}_{\boldsymbol{z}} | \hat{\boldsymbol{\theta}} \sim N\left(\boldsymbol{z}\hat{\boldsymbol{\theta}}, \sigma^2\right),$$

其中$\hat{X}_{\boldsymbol{z}}$是自变量取值为$\boldsymbol{z}$的(个体) 预测值, σ^2是模型误差的方差.

但由于$\text{COV}\left(\hat{\boldsymbol{\theta}}\right)$和$\sigma^2$往往是未知的, 需要先估计未知参数, 并根据相关分布采样. 例如, 当满足一定的条件时, $\hat{\boldsymbol{\theta}}$的样本应基于霍特林$T$方分布产生随

表 9.9: 初始插补后的数据集与Temperature的第1轮预测值

序号	Temperature	Humidity	$\widehat{\text{Temperature}}$
1	65	70	69.14
2	80	80	69.89
3	83	86	72.29
4	70	96	73.79
5	68	80	71.39
6	65	70	69.89
7	64	65	69.14
8	72	95	73.64

机数并作一定转换得到, $\hat{X}_{\boldsymbol{z}}|\hat{\boldsymbol{\theta}}$的样本应基于$t$分布产生随机数并作一定转换得到.

可以看到, 对$\hat{\boldsymbol{\theta}}$采样比较麻烦. 若直接对模型的预测值采样, 会让问题简化. 当模型误差满足正态性、方差齐性和独立性时, 有:

$$\frac{\hat{X}_{\boldsymbol{z}}-X_{\boldsymbol{z}}}{\sqrt{\hat{\sigma}^2\left(\boldsymbol{z}\left(\boldsymbol{Z}^T\boldsymbol{Z}\right)^{-1}\boldsymbol{z}^T+1\right)}}\cdot\sqrt{m-q-1}\sim t\left(m-q-1\right),$$

其中$\hat{\sigma}^2$为模型误差方差的估计值. 因此, 可以从t分布中抽取样本构造插补值$\hat{X}_{\boldsymbol{z}}$.

在基于MICE的预测均值匹配方法中, 需要对均值μ的预测值$\hat{\mu}$采样. 以普通最小二乘回归模型为插补模型, 当模型误差满足正态性、方差齐性和独立性时, 自变量取值为$\boldsymbol{z}$的预测均值$\hat{\mu}_{\boldsymbol{z}}$的分布满足:

$$\frac{\hat{\mu}_{\boldsymbol{z}}-\mu_{\boldsymbol{z}}}{\sqrt{\hat{\sigma}^2\left(\boldsymbol{z}\left(\boldsymbol{Z}^T\boldsymbol{Z}\right)^{-1}\boldsymbol{z}^T\right)}}\cdot\sqrt{m-q-1}\sim t\left(m-q-1\right).$$

可以从t分布中采样构造$\tilde{\mu}$, 并按照(9.5) 式通过匹配获得插补值.

例9.6. 选取*golf*数据集中*2*个定量变量和前*8*个样本观测, 假定其中含有*3*个缺失值, 如表*9.2*前三列所示. 下面, 采用基于*MICE*的预测均值匹配方法插补缺失值, 采用普通最小二乘回归模型作为插补模型.

首先, 从观测值中随机抽样, 作为初始插补值. 假定抽中序号为*6*的样本观测的*Temperature*值, 以及序号为*7*和*6*的*Humidity*值, 初始插补后的数据集如表*9.9*第*2*至*3*列所示.

表 9.10: 第1轮插补Temperature后的数据集

序号	Temperature	Humidity
1	70	65
2	80	70
3	83	86
4	70	96
5	68	80
6	65	70
7	64	65
8	72	95

实施第*1*轮插补. 以*Temperature*为因变量, *Humidity*为自变量, 基于表*9.9*第*2*至*3*列中的所有数据, 建立普通最小二乘回归模型:

$$\widehat{Temperature} = 59.39 + 0.15 \times Humidity.$$

由模型得到*Temperature*的预测值, 列于表*9.9*第*4*列中.

第*1*个样本观测的均值预测值为$\hat{\mu} = 69.14$, 用于建立模型的样本量为*7*, $q = 1$, 因此从$t(5)$分布中抽取一个随机样本, 例如为*1.66*. 模型误差方差的估计值为*295.23*. 因此

$$\tilde{\mu} = 69.14 + \frac{1.66 \times \sqrt{295.23 \times z^T \left(\boldsymbol{Z}^T \boldsymbol{Z}\right)^{-1} z}}{\sqrt{7-1-1}} \approx 77.17,$$

其中, $z = (1, 65)^T$, $\boldsymbol{Z}$的第*1*列是取值全为*1*的向量, 第*2*列是序号为*2*至*8*的样本观测的*Humidity*值. 需要注意的是, 本例中我们并未确认模型误差满足正态性、方差齐性和独立性, 因此, 预测均值的分布可能并不精确, 是近似分布.

可以看到, $\tilde{\mu}$与序号为*4*的样本观测的预测值最为接近, 因此, 序号为*4*的样本观测为邻居, 以它的*Temperature*值*70*作为插补值. 从而, 更新数据集, 如表*9.10*所示.

类似地, 可基于表*9.10*的数据集, 对*Humidity*的插补值进行更新, 留给读者练习.

9.7 案例分析

基于car数据集, 划分训练集和测试集.

本例中, 仅有Reliability含有缺失值. 这里将Reliability作为定量变量处理, 其原因是希望完整地呈现各方法(KNNImputer函数只能插补定量变量的缺失值). 首先采用基于训练集插补缺失值, 然后应用到测试集中.

1. 以非条件中心趋势值插补

采用SimpleImputer函数, 以众数插补缺失值.

```
imputer = SimpleImputer(strategy='most_frequent')
numeric_features = np.where(X_train.dtypes != object)[0]
X_train_imp = pd.DataFrame(imputer.fit_transform(X_train.iloc
    [:,numeric_features]),columns=X_train.iloc[:,
    numeric_features].columns)
```

2. k近邻插补

采用KNNImputer函数实现k近邻插补. 由于涉及欧氏距离运算, 需要先作预处理: (1) 对除待插补变量以外的定量变量标准化; (2) 对除待插补变量以外的定性变量作独热编码, 为使得各原始自变量在距离运算中权重基本相等，将所有独热编码变量变为-1和1两个取值. 将待插补的变量与预处理后的变量合并为X_trainpreprocess.

```
numeric_features = np.where(X_train.drop(['Reliability'], axis
    =1).dtypes != object)[0]
numeric_transformer = Pipeline(steps=[('scaler',
    StandardScaler())])

categorical_features = np.where(X_train.drop(['Reliability'],
    axis=1).dtypes == object)[0]
categorical_transformer = Pipeline(steps=[('onehot',
    OneHotEncoder(handle_unknown='ignore'))])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)
           ])

x1 = X_train[['Reliability']]
x2 = pd.DataFrame(preprocessor.fit_transform(X_train.drop(['
```

```
    Reliability'], axis=1)))
x2.iloc[:,len(numeric_features):] = (x2.iloc[:,len(
    numeric_features):]-0.5)*2
X_trainpreprocess = pd.merge(x1,x2,left_index=True,
    right_index=True, how='outer')
```

取近邻数量$k = 3$, 采用近邻的平均值作为插补值.

```
imputerknn = KNNImputer(n_neighbors=3, weights='uniform')
X_train_impknn = pd.DataFrame(imputerknn.fit_transform(
    X_trainpreprocess))
```

3. 回归插补与随机回归插补

可以将回归插补看作预测的任务, 通过建立模型获得插补值. 本例中, 通过IterativeImputer函数实现回归插补. IterativeImputer函数一般用于联合建模机制中的链式方法, 但通过将迭代次数设置为1, 且放入的数据中只有待插补变量含缺失值(删除在其他变量有缺失值的样本观测), 则可以实现回归插补. 由于本例中仅有待插补变量Reliability含有缺失值, 因此可以使用训练集中所有的样本观测. 为了建立回归模型), 需要将定性变量作独热编码, 注意去掉其中一个独热编码变量, 以避免共线性. 设预处理后的训练集为X_iter. 在IterativeImputer函数中, 设置最大迭代次数max_iter为1; 设置sample_posterior=False, 则采用模型的截尾预测值(不超过所设置的最大和最小值) 作为插补值; 插补的最小值min_value为1, 最大值max_value为5. 这里, 未设置模型, 即以默认的BayesianRidge函数建立模型.

```
iterreg = IterativeImputer(max_iter=1, sample_posterior=False,
     min_value=1, max_value=5)
iterreg.fit(X_iter)
iter_imputation = pd.DataFrame(iterreg.transform(X_iter.iloc[
    X_train['Reliability'].isnull().values==True,:]))
```

类似地, 在IterativeImputer函数中设置max_iter=1和sample_posterior=True, 可实现随机回归插补. 为重现随机抽样的结果, 可以为随机数种子random_state赋值. 设置预测分布为截尾型分布, 其最小值min_value为1, 最大值max_value为5.

```
iterrandreg = IterativeImputer(sample_posterior=True, max_iter
    =1, random_state=0, min_value=1, max_value=5)
```

```
iterrandreg.fit(X_iter)
iter_randimputation = pd.DataFrame(iterrandreg.transform(
    X_iter.iloc[X_train['Reliability'].isnull().values==True
    ,:]))
```

4. 缺失森林

由MissForest函数实现缺失森林. MissForest函数可以直接处理数值型的定性变量, 因此不需要对定性变量作独热编码. 但是, 它不接受字符串, 需先对定性变量作有序编码, 变为数值型.

```
categorical_features = np.where(X_train.dtypes == object)[0]
categorical_transformer = Pipeline(steps=[('ordinal',
    OrdinalEncoder(handle_unknown='use_encoded_value',
    unknown_value=np.nan))])
preprocessor = ColumnTransformer(transformers=[('cat',
    categorical_transformer, categorical_features)])
X_traincategorical = pd.DataFrame(preprocessor.fit_transform(
    X_train), columns=X_train.columns[categorical_features])
```

将定量变量与预处理后的定性变量X_traincategorical合并, 获得预处理后的训练集X_trainpreprocess. 在MissForest函数中, 设置最大迭代次数max_iter为10, 随机数种子random_state为1. 特别注意, 应在fit函数应声明定性变量. 本例中, 定性变量所在的列号为5和6, 因此设置cat_vars=[5,6]. 随后, 通过transform函数可实现对缺失值的插补.

```
imputer = MissForest(max_iter=10, random_state=1)
imputer.fit(X_trainpreprocess, cat_vars=[5,6])
imp_missforest = imputer.transform(X_trainpreprocess)
```

5. MICE与预测均值匹配

可采用两种方式实现MICE. 第一种方式通过IterativeImputer函数实现MICE. 但由于IterativeImputer函数必须采用一个指定的模型, 因此不便于同时插补定性和定量的缺失值(除非建模函数可以自适应因变量类型). 第二种方式通过miceforest包中的函数实现基于MICE. 下面, 采用第二种方式插补缺失值.

miceforest包采用LightGBM模型作为插补模型. 数据预处理应根据LightGBM模型的要求. 这里, 只需要将定性变量的类型设置为'category'即可.

```
X_train.iloc[:,categorical_features] = X_train.iloc[:,
    categorical_features].astype('category')
```

ImputationKernel函数主要用于实现预测均值匹配. 当设置近邻数量为0时, 以模型预测值作为插补值, 此时可以实现MICE. 因此, 设置mean_match_scheme的其他参数为默认值, 近邻数量为0; 设置随机数种子random_state=0; 为实现多重插补, 设置datasets=2, 此时对每一个缺失数据可获得两个插补值.

```
scheme_mmc = mean_match_default.copy()
scheme_mmc.set_mean_match_candidates(0)
miceimp = mf.ImputationKernel(X_train, mean_match_scheme=
    scheme_mmc, random_state=0, datasets=2)
```

需要通过mice函数获得插补值. 设置MICE的迭代次数iterations=5, 以及LightGBM中个体学习器的数量n_estimators=10. 所获得的两组含有插补值的数据集分别记为X_trainmice1和X_trainmice2.

```
miceimp.mice(iterations=5, n_estimators=10)
X_trainmice1 = miceimp.complete_data(0)
X_trainmice2 = miceimp.complete_data(1)
```

类似可以实现预测均值匹配. 这里, 设置近邻数量为1, 随机数种子random_state=0, datasets=1 (为单一插补).

```
scheme_mmc = mean_match_default.copy()
scheme_mmc.set_mean_match_candidates(1)
pmm = mf.ImputationKernel(X_train, mean_match_scheme=
    scheme_mmc, random_state=0, datasets=1)
pmm.mice(iterations=5, n_estimators=10)
X_trainpmm = pmm.complete_data()
```

以上完成了缺失值插补的训练. 采用transform函数可以插补测试集中缺失值. 例如, 采用预测均值匹配插补训练集中的缺失值. 首先对测试集作预处理, 将定性变量设置为'category'. 这里涉及一个问题, 若测试集中出现了训练集中未含有的类别, 运行插补函数时会报错. 因此, 将训练集中未含有的类别设置为NaN.

Table 9.11: 训练集缺失值插补结果

序号	以众数插补	k近邻插补	回归插补	随机回归插补	缺失森林	MICE	预测均值匹配
22	5	4	5	4	4	3	3
58	5	5	4	5	5	5	5
59	5	4	4	5	5	5	5
20	5	3	4	3	2	3	2
57	5	5	4	4	5	5	5
9	5	5	4	4	5	5	5

```
X_country = pd.Categorical(X_train.iloc[:,0])
X_type = pd.Categorical(X_train.iloc[:,3])
cat_country = CategoricalDtype(categories=X_country.categories
    )
cat_type = CategoricalDtype(categories=X_type.categories)
X_test.iloc[:,0] = X_test.iloc[:,0].astype(cat_country)
X_test.iloc[:,3] = X_test.iloc[:,3].astype(cat_type)
```

对预处理后的测试集X_test, 通过transform函数产生插补值.

```
X_testpmm = pmm.transform(X_test)
```

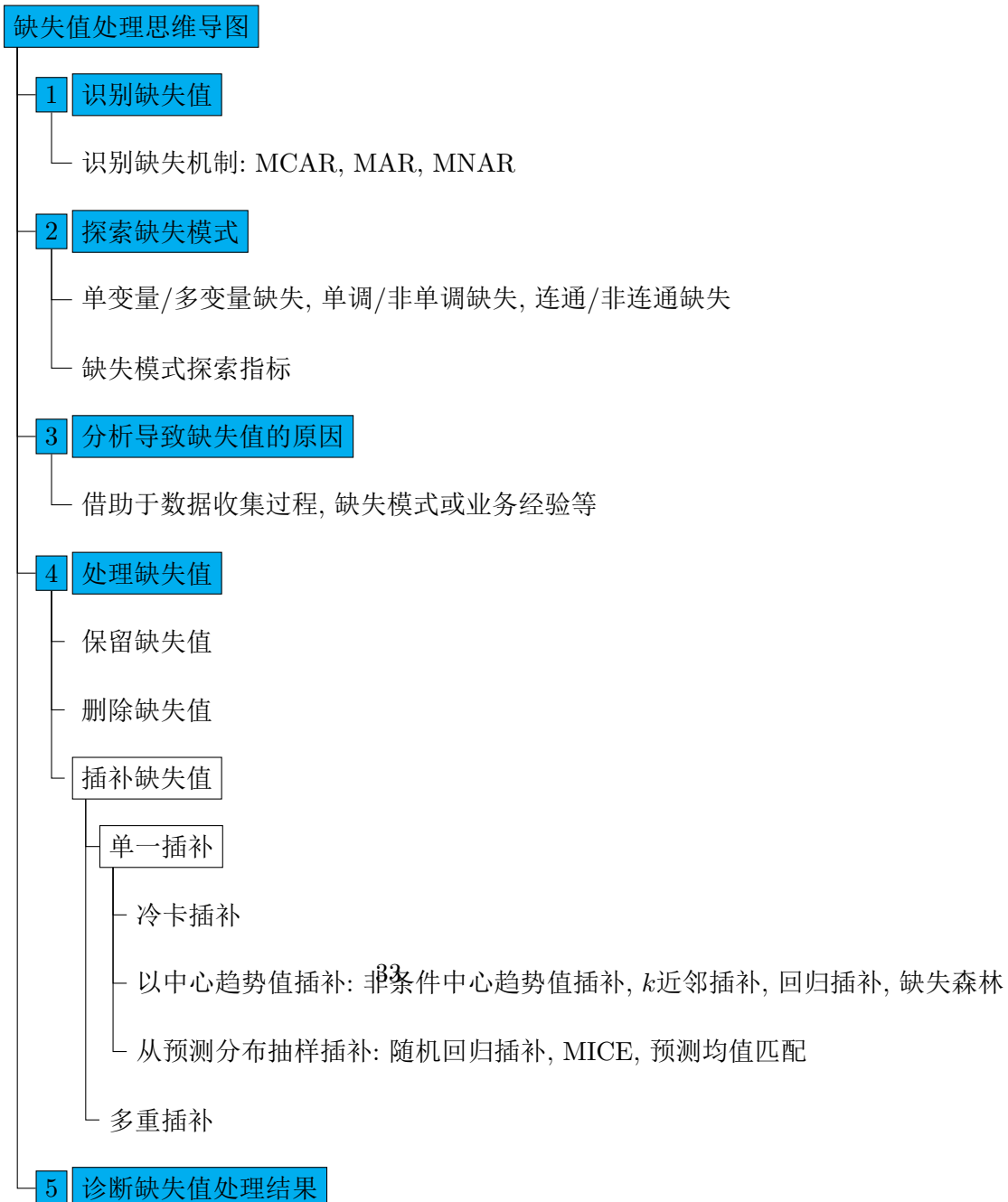
表9.11和9.12分别展示了训练集和测试集的缺失值插补结果, 其中k近邻插补、回归插补、随机回归和MICE插补值四舍五入为整数, 避免产生“外推”的插补值. 显然, 以众数作为插补值不具有变异性; 回归插补和随机回归插补虽然采用了相同的插补模型, 但插补值存在较大差异, 且与其他方法的差异较大; MICE和预测均值匹配都采用LightGBM作为插补模型, 插补值最为接近, 它们与k近邻插补和缺失森林的插补值较为接近.

Table 9.12: 测试集缺失值插补结果

序号	以众数插补	k近邻插补	回归插补	随机回归插补	缺失森林	MICE	预测均值匹配
48	5	2	3	3	2	3	3
44	5	3	3	5	4	4	3
33	5	5	5	2	3	3	3
53	5	3	3	2	2	3	3
21	5	5	4	1	4	5	5

本章小结

(1) 思维导图



(2) Python实现

表9.13列出了缺失值处理的Python函数.

References

[1] Buuren S. van, Groothuis-Oudshoorn K. (1999) Flexible Multivariate Imputation by MICE. Leiden: TNO.

[2] Buuren S. van (2021) Flexible Imputation of Missing Data (second edition). Boca Raton: CRC Press. (https://stefvanbuuren.name/fimd/want-the-hardcopy.html)

[3] Little R.J.A., Rubin D.B.著, 周晓华, 邓宇昊译(2022). 缺失数据统计分析(第三版). 北京: 高等教育出版社.

[4] Stekhoven D.J., Bühlmann P. (2012) MissForest—non-parametric missing value imputation for mixed-type data. Bioinformatics, 28(1):112-118.

[5] Buuren S. van, Groothuis-Oudshoorn K. (2011) mice: Multivariate imputation by chained equations in R.

[6] Little R.J.A. (1988) Missing-data adjustments in large surveys. Journal of Business & Economic Statistics, 6(3): 287 – 296.

[7] Rubin D.B. (1987) Multiple Imputation for Nonresponse in Surveys. New York: John Wiley & Sons.

[8] Troyanskaya O., Cantor M., Sherlock G., Brown P., Hastie T., Tibshirani R., Botstein D., Altman R.B. (2001) Missing value estimation methods for DNA microarrays. Bioinformatics, 17(6):520-525.

表 9.13: 缺失值处理的Python实现

方法	函数名称与所属模块	重要参数	应用提示
删除含缺失值的样本观测或变量	dropna (pandas)	(1) axis: 按行或按列删除缺失值. 默认值为0, 即按行删除含有缺失值的样本观测. 若取值为1, 则按列删除含有缺失值的变量 (2) how: 筛选方式. 默认值为'any', 即样本观测或变量含有1个缺失值, 则删除相应的样本观测或变量. 若取值为'all', 当样本观测或变量中所有数值都缺失时才会删除相应的样本观测或变量 (3) thresh: 非缺失值最低数量. 如果样本观测或变量中的非缺失值小于该值, 会删除相应的样本观测或变量. 默认值为None (4) subset: 限定删除操作所涉及的数据子集, 默认值为None (5) inplace: 是否原地替换. 默认值为False, 即不以删除后的数据集替换原始数据集, 但可以输出展示删除后的数据集. 若取值为True, 以删除后的数据集替换原始数据集. 建议取默认值False	-
以非条件中心趋势值插补缺失值	SimpleImputer (scikit-learn库impute)	strategy: 插补的统计量. 默认值为'mean', 以全体非缺失值的均值插补, 仅适用于定量变量. 还可设置为'median'、'most_frequent'和'constant', 分别对应以中位数、众数和某一设置的常数作为插补值	设置strategy='constant', 可实现冷卡插补
k近邻插补	KNNImputer (scikit-learn库impute)	(1) n_neighbors: 邻居数量, 默认值为5 (2) weights: 加权平均时的权重. 默认值为'uniform', 即各邻居等权重影响插补值. 可设置为'distance', 以距离的倒数为权重	(1) 函数采用近邻(加权) 平均值进行插补, 因而仅适用于定量数据的缺失值插补 (2) 由于涉及距离的计算, 应对定量变量作规范化; 函数不能直接处理定性变量, 应对定性变量作独热编码, 需考虑与定量变量尺度一致的问题
回归插补与随机回归插补	IterativeImputer (scikit-learn库impute)	(1) estimator: 采用的插补模型. 默认值为BayesianRidge(), 属于线性回归模型 (2) max_iter:最大的迭代次数, 默认值为10 (3) sample_posterior: 是否通过从后验分布中抽样产生插补值. 默认值为False, 此时直接使用模型预测值(不超过设置的最大和最小值) 作为插补值. 若设置为True, 则在模型预测值基础上形成截尾正态分布作为预测值后验分布, 从该分布中随机抽样作为插补值. (4) random_state: 随机数种子(与sample_posterior=True搭配使用) (5) min_value: 插补值的最小值, 默认值为-Inf (6) max_value: 插补值的最大值, 默认值为Inf	(1) 设置max_iter=1, 且放入的数据中只有待插补变量含缺失值(删除在其他变量有缺失值的样本观测), 可实现回归插补 (2) 设置max_iter=1和sample_posterior=True, 且放入的数据中只有待插补变量含缺失值, 可实现随机回归插补 (3) 应按照estimator的要求进行数据预处理. 例如, 若estimator=BayesianRidge(), BayesianRidge函数不能直接处理定性变量, 应对定性变量作独热编码, 注意虚拟变量的数量应等于变量取值个数减1, 以避免共线性问题; BayesianRidge函数不允许缺失值, 注意只需删除在除待插补变量以外的变量上有缺失值的样本观测
MICE与预测均值匹配	ImputationKernel (miceforest)	(1) mean_match_scheme: 最为重要的是设置匹配时的近邻数量. 在默认的mean_match_default情况下, 当近邻数量为0时, 则采用模型预测值作为插补值(即不能实现预测均值匹配); 当近邻数量大于0时, 会从这些数量的近邻中随机抽取1个观测值作为插补值 (2) datasets: 需要的插补值数量. 若设置为1, 则为单一插补; 若设置为大于1的值, 可实现多重插补 (3) random_state: 随机数种子	函数采用lightgbm包中的函数建立LightGBM模型作为插补模型, 因此数据预处理应根据lightgbm的要求: 将定性变量的类型设置为'category'. 设置后, LightGBM模型可以自动将'category'类型的变量作为定性变量