【2.5】缺失值的处理

August 20, 2014 na (/categories/na) 阅读量：434 次

对于数据挖掘和分析人员来说，数据准备（Data Preparation，包括数据的抽取、清洗、转换和集成）常常占据了70%左右的工作量。而在数据准备的过程中，数据质量差又是最常见而且令人头痛的问题。本文针对缺失值和特殊值这种数据质量问题，进行了初步介绍并推荐了一些处理方法。

值得注意的是，这里所说的缺失值，不仅包括数据库中的NULL值，也包括用于表示数值缺失的特殊数值（比如，在系统中用-999来表示数值不存在）。如果我们仅有数据库的数据模型，而缺乏相关说明，常常需要花费更多的精力来发现这些数值的特殊含义。而如果我们漠视这些数值的特殊性，直接拿来进行挖掘，那么很可能会得到错误的结论。

还有一种数值缺失的情况，是因为我们要求统计的时间窗口并非对所有数据都适合。例如，我们希望计算出"客户在以前六个月内的最大存款余额"，对于那些建立账户尚不满六个月的客户来说，统计出来的数值与我们想要得到的就可能存在差距。

一般来说，对缺失值的填充方法有多种，用某个常数来填充常常不是一个好方法。最好建立一些模型，根据数据的分布来填充一个更恰当的数值。（例如根据其它变量对记录进行数据分箱，然后选择该记录所在分箱的相应变量的均值或中位数，来填充缺失值，效果会更好一些）

一、造成数据缺失的原因

在各种实用的数据库中，属性值缺失的情况经常发全甚至是不可避免的。因此，在大多数情况下，信息系统是不完备的，或者说存在某种程度的不完备。

缺失值的产生的原因多种多样，主要分为机械原因和人为原因。机械原因是由于机械原因导致的数据收集或保存的失败造成的数据缺失，比如数据存储的失败，存储器损坏，机械故障导致某段时间数据未能收集（对于定时数据采集而言）。人为原因是由于人的主观失误、历史局限或有意隐瞒造成的数据缺失，比如，在市场调查中被访人拒绝透露相关问题的答案，或者回答的问题是无效的，数据录入人员失误漏录了数据

造成数据缺失的原因是多方面的，主要可能有以下几种：

1. 有些信息暂时无法获取。例如在医疗数据库中，并非所有病人的所有临床检验结果都能在给定的时间内得到，就致使一部分属性值空缺出来。又如在申请表数据中，对某些问题的反映依赖于对其他问题的回答。

2. 有些信息是被遗漏的。可能是因为输入时认为不重要、忘记填写了或对数据理解错误而遗漏，也可能是由于数据采集设备的故障、存储介质的故障、传输媒体的故障、一些人为因素等原因而丢失了。

3. 有些对象的某个或某些属性是不可用的。也就是说，对于这个对象来说，该属性值是不存在的，如一个未婚者的配偶姓名、一个儿童的固定收入状况等。

4. 有些信息（被认为）是不重要的。如一个属性的取值与给定语境是无关的，或训练数据库的设计者并不在乎某个属性的取值（称为dont-care value）。

5. 获取这些信息的代价太大。

6. 统实时性能要求较高，即要求得到这些信息前迅速做出判断或决策。

二、数据缺失机制

在对缺失数据进行处理前，了解数据缺失的机制和形式是十分必要的。将数据集中不含缺失值的变量（属性）称为完全变量，数据集中含有缺失值的变量称为不完全变量，Little 和 Rubin定义了以下三种不同的数据缺失机制：

1. 完全随机缺失（Missing Completely at Random，MCAR）

数据的缺失与不完全变量以及完全变量都是无关的。

完全随机缺失（MCAR，Missing Completely At Random），指的是数据的缺失不依赖于自身或者其他变量，完全是随机的。比如你在街头请路人填问卷采集数据，路人很可能由于某种原因（肚子疼，或者他妈喊他回家吃饭等等）中途撤走，那么你的数据就不完整了。再比如你的质谱在采集数据过程中，不小心受到某种程度的扰动（鬼知道仪器经历了什么...），导致你的数据受到影响，最后在搜库匹配时，无法给出对应的定量信息。这种情况指不清道不明，完全随机，所以它对你整个数据的影响没有任何的偏好性，呈现均一分布。

2. 随机缺失（Missing at Random，MAR）。

数据的缺失仅仅依赖于完全变量。

随机缺失（MAR，Missing At Random），指的是数据的缺失不是完全随机的，该类数据的缺失依赖于其他观测变量。比如时间梯度越长的采集越可能有缺失值的出现。这个时候，若是我们将时间变量进行控制，那么数据的缺失也就变成了完全随机的了。所以也有人认为MCAR和MAR二者没啥区别，或者认为MCAR是MAR的一个特例（doi:10.$^{1186}/_{1471}$-2105-13-S16-S5）。

3.非随机、不可忽略缺失（Not Missing at Random,NMAR，or nonignorable）。

不完全变量中数据的缺失依赖于不完全变量本身，这种缺失是不可忽略的。

非随机缺失（MNAR，Missing Not At Random），指的是数据的缺失依赖于观测变量自身。比如在质谱检测的过程中，某些肽段的含量在仪器的检测限以下，这些肽段的定量信息就很有可能丢失，但是你又不能说这些肽段真的不存在，所以这种情况是比较纠结的，这就是所谓的左删失数据（left-censored data）。

从缺失值的所属属性上讲，如果所有的缺失值都是同一属性，那么这种缺失成为单值缺失，如果缺失值属于不同的属性，称为任意缺失。另外对于时间序列类的数据，可能存在随着时间的缺失，这种缺失称为单调缺失。

三、空值语义

对于某个对象的属性值未知的情况，我们称它在该属性的取值为空值(null value)。空值的来源有许多种，因此现实世界中的空值语义也比较复杂。总的说来，可以把空值分成以下三类：

1)不存在型空值。即无法填入的值，或称对象在该属性上无法取值，如一个未婚者的配偶姓名等。

2)存在型空值。即对象在该属性上取值是存在的，但暂时无法知道。一旦对象在该属性上的实际值被确知以后，人们就可以用相应的实际值来取代原来的空值，使信息趋于完全。存在型空值是不确定性的一种表征，该类空值的实际值在当前是未知的。但它有确定性的一面，诸如它的实际值确实存在，总是落在一个人们可以确定的区间内。一般情况下，空值是指存在型空值。

3)占位型空值。即无法确定是不存在型空值还是存在型空值，这要随着时间的推移才能够清楚，是最不确定的一类。这种空值除填充空位外，并不代表任何其他信息。

四、空值处理的重要性和复杂性

数据缺失在许多研究领域都是一个复杂的问题。对数据挖掘来说，空值的存在，造成了以下影响：首先，系统丢失了大量的有用信息；第二，系统中所表现出的不确定性更加显著，系统中蕴涵的确定性成分更难把握；第三，包含空值的数据会使挖掘过程陷入混乱，导致不可靠的输出。

数据挖掘算法本身更致力于避免数据过分适合所建的模型，这一特性使得它难以通过自身的算法去很好地处理不完整数据。因此，空缺的数据需要通过专门的方法进行推导、填充等，以减少数据挖掘算法与实际应用之间的差距。

五、空值处理方法的分析比较

处理不完备数据集的方法主要有以下三大类：

5.1 删除元组

也就是将存在遗漏信息属性值的对象（元组，记录）删除，从而得到一个完备的信息表。这种方法简单易行，在对象有多个属性缺失值、被删除的含缺失值的对象与信息表中的数据量相比非常小的情况下是非常有效的，类标号（假设是分类任务）缺少时通常使用。然而，这种方法却有很大的局限性。它是以减少历史数据来换取信息的完备，会造成资源的大量浪费，丢弃了大量隐藏在这些对象中的信息。在信息表中本来包含的对象很少的情况下，删除少量对象就足以严重影响到信息表信息的客观性和结果的正确性；当每个属性空值的百分比变化很大时，它的性能非常差。因此，当遗漏数据所占比例较大，特别当遗漏数据非随机分布时，这种方法可能导致数据发生偏离，从而引出错误的结论。

5.2 数据补齐

这类方法是用一定的值去填充空值，从而使信息表完备化。通常基于统计学原理，根据决策表中其余对象取值的分布情况来对一个空值进行填充，譬如用其余属性的平均值来进行补充等。数据挖掘中常用的有以下几种补齐方法：

5.2.1 人工填写（filling manually）

由于最了解数据的还是用户自己，因此这个方法产生数据偏离最小，可能是填充效果最好的一种。然而一般来说，该方法很费时，当数据规模很大、空值很多的时候，该方法是不可行的。

5.2.2 特殊值填充（Treating Missing Attribute values as Special values）

将空值作为一种特殊的属性值来处理，它不同于其他的任何属性值。如所有的空值都用“unknown”填充。这样将形成另一个有趣的概念，可能导致严重的数据偏离，一般不推荐使用。

5.2.3 平均值填充（Mean/Mode Completer）

将信息表中的属性分为数值属性和非数值属性来分别进行处理。如果空值是数值型的，就根据该属性在其他所有对象的取值的平均值来填充该缺失的属性值；如果空值是非数值型的，就根据统计学中的众数原理，用该属性在其他所有对象的取值次数最多的值(即出现频率最高的值)来补齐该缺失的属性值。另外有一种与其相似的方法叫条件平均值填充法（Conditional Mean Completer）。在该方法中，缺失属性值的补齐同样是靠该属性在其他对象中的取值求平均得到，但不同的是用于求平均的值并不是从信息表所有对象中取，而是从与该对象具有相同决策属性值的对象中取得。这两种数据的补齐方法，其基本的出发点都是一样的，以最大概率可能的取值来补充缺失的属性值，只是在具体方法上有一点不同。与其他方法相比，它是用现存数据的多数信息来推测缺失值。

5.2.4 热卡填充（Hot deck imputation，或就近补齐）

对于一个包含空值的对象，热卡填充法在完整数据中找到一个与它最相似的对象，然后用这个相似对象的值来进行填充。不同的问题可能会选用不同的标准来对相似进行判定。该方法概念上很简单，且利用了数据间的关系来进行空值估计。这个方法的缺点在于难以定义相似标准，主观因素较多。

5.2.5 K最近距离邻法（K-means clustering）

先根据欧式距离或相关分析来确定距离具有缺失数据样本最近的K个样本，将这K个值加权平均来估计该样本的缺失数据。

同均值插补的方法都属于单值插补，不同的是，它用层次聚类模型预测缺失变量的类型，再以该类型的均值插补。假设X=(X1,X2…Xp)为信息完全的变量，Y为存在缺失值的变量，那么首先对X或其子集行聚类，然后按缺失个案所属类来插补不同类的均值。如果在以后统计分析中还需以引入的解释变量和Y做分析，那么这种插补方法将在模型中引入自相关，给分析造成障碍。

该方法主要是通过预先设定的K个邻居（其它肽段或者蛋白质的表达量），根据这些邻居的信息推算出缺失值的大小。一般的数据处理流程是先计算目标对象（含有缺失值的肽段或者蛋白质）与其他对象之间的距离（一般默认计算的是欧氏距离），计算完成后，选择K个（K值是我们预先设定的）距离最近的对象，然后将对应位置的数值进行平均或者加权，其得到的数值用来表征该缺失值的大小。R代码实现：

```
>library(impute)
>impute.kmm(data, k=10, rowmax=0.5, colmax=0.8)
```

这里面我们用impute包的imput.knn函数，其中data就是要导入的数据，一般是矩阵的形式；k就是我们预先设定的近邻数，默认为10，根据经验一般取值是10到20之间；rowmax和colmax是控制的导入数据中行或者列含有缺失值的比例，比如rowmax=0.5是指当某行的数据缺失值占的比例超过了50%，那么这些缺失值就会用整个样本的均值去填充，而不是我们上面提到的根据K个邻居来填充了；colmax=0.8是指任何一列的数据中当缺失值的比例超过了80%，那么计算就会停止并会报出错误。

5.2.6 使用所有可能的值填充（Assigning All Possible values of the Attribute）

这种方法是用空缺属性值的所有可能的属性取值来填充，能够得到较好的补齐效果。但是，当数据量很大或者遗漏的属性值较多时，其计算的代价很大，可能的测试方案很多。另有一种方法，填补遗漏属性值的原则是一样的，不同的只是从决策相同的对象中尝试所有的属性值的可能情况，而不是根据信息表中所有对象进行尝试，这样能够在一定程度上减小原方法的代价。

5.2.7 组合完整化方法（Combinatorial Completer）

这种方法是用空缺属性值的所有可能的属性取值来试，并从最终属性的约简结果中选择最好的一个作为填补的属性值。这是以约简为目的的数据补齐方法，能够得到好的约简结果；但是，当数据量很大或者遗漏的属性值较多时，其计算的代价很大。另一种称为条件组合完整化方法（Conditional Combinatorial Complete），填补遗漏属性值的原则是一样的，不同的只是从决策相同的对象中尝试所有的属性值的可能情况，而不是根据信息表中所有对象进行尝试。条件组合完整化方法能够在一定程度上减小组合完整化方法的代价。在信息表包含不完整数据较多的情况下，可能的测试方案将巨增。

5.2.8 回归（Regression）

基于完整的数据集，建立回归方程（模型）。对于包含空值的对象，将已知属性值代入方程来估计未知属性值，以此估计值来进行填充。当变量不是线性相关或预测变量高度相关时会导致有偏差的估计。

5.2.9 期望值最大化方法（Expectation maximization，EM）

在缺失类型为随机缺失的条件下，假设模型对于完整的样本是正确的，那么通过观测数据的边际分布可以对未知参数进行极大似然估计（Little and Rubin）。这种方法也被称为忽略缺失值的极大似然估计，对于极大似然的参数估计实际中常采用的计算方法是期望值最大化(Expectation Maximization，EM）。该方法比删除个案和单值插补更有吸引力，它一个重要前提：适用于大样本。有效样本的数量足够以保证ML估计值是渐近无偏的并服从正态分布。但是这种方法可能会陷入局部极值，收敛速度也不是很快，并且计算很复杂。

EM算法是一种在不完全数据情况下计算极大似然估计或者后验分布的迭代算法。在每一迭代循环过程中交替执行两个步骤：E步（Excepctaion step,期望步），在给定完全数据和前一次迭代所得到的参数估计的情况下计算完全数据对应的对数似然函数的条件期望；M步（Maximzation step，极大化步），用极大化对数似然函数以确定参数的值，并用于下步的迭代。算法在E步和M步之间不断迭代直至收敛，即两次迭代之间的参数变化小于一个预先给定的阈值时结束。该方法可能会陷入局部极值，收敛速度也不是很快，并且计算很复杂。

5.2.10 有序K近邻法（Sequential KNN）

方法是对上述KNN法的改进，它在缺失值占比稍微大些的数据中表现依然良好。实现流程是先根据数据中每个对象缺失值的比例进行排序（这里也就体现出“Sequential”的思想），从比例最小的那个对象开始计算，根据预先设定的K个近邻（注意这里是指没有缺失值的，KNN法里面并没有强调这一点）的值进行加权或者平均计算填充。此外当该对象填充完毕后，也会加入后续其他对象缺失值填充的计算当中。R代码实现：

```
>library(SeqKnn)
>SeqKNN(data，k=10)
```

这里我们用SeqKnn包的SeqKNN函数，其中data就是我们要导入的数据，一般是矩阵的形式；k就是我们预先设定的近邻数，根据经验一般取值是10到20之间。

5.2.10 多重填补（Multiple Imputation， MI）

多值插补的思想来源于贝叶斯估计，认为待插补的值是随机的，它的值来自于已观测到的值。具体实践上通常是估计出待插补的值，然后再加上不同的噪声，形成多组可选插补值。根据某种选择依据，选取最合适的插补值。

多重填补方法分为三个步骤：;为每个空值产生一套可能的填补值，这些值反映了无响应模型的不确定性；每个值都被用来填补数据集中的缺失值，产生若干个完整数据集合。;每个填补数据集合都用针对完整数据集的统计方法进行统计分析。;对来自各个填补数据集的结果进行综合，产生最终的统计推断，这一推断考虑到了由于数据填补而产生的不确定性。该方法将空缺值视为随机样本，这样计算出来的统计推断可能受到空缺值的不确定性的影响。该方法的计算也很复杂。

多重插补方法分为三个步骤：①为每个空值产生一套可能的插补值，这些值反映了无响应模型的不确定性；每个值都可以被用来插补数据集中的缺失值，产生若干个完整数据集合。②每个插补数据集合都用针对完整数据集的统计方法进行统计分析。③对来自各个插补数据集的结果，根据评分函数进行选择，产生最终的插补值。

假设一组数据，包括三个变量Y1，Y2，Y3，它们的联合分布为正态分布，将这组数据处理成三组，A组保持原始数据，B组仅缺失Y3，C组缺失Y1和Y2。在多值插补时，对A组将不进行任何处理，对B组产生Y3的一组估计值（作Y3关于Y1，Y2的回归），对C组作产生Y1和Y2的一组成对估计值（作Y1，Y2关于Y3的回归）。

当用多值插补时，对A组将不进行处理，对B、C组将完整的样本随机抽取形成为m组（m为可选择的m组插补值），每组个案数只要能够有效估计参数就可以了。对存在缺失值的属性的分布作出估计，然后基于这m组观测值，对于这m组样本分别产生关于参数的m组估计值，给出相应的预测即，这时采用的估计方法为极大似然法，在计算机中具体的实现算法为期望最大化法（EM）。对B组估计出一组Y3的值，对C将利用 Y1,Y2,Y3它们的联合分布为正态分布这一前提，估计出一组(Y1，Y2）。

上例中假定了Y1,Y2,Y3的联合分布为正态分布。这个假设是人为的，但是已经通过验证（Graham和Schafer于1999），非正态联合分布的变量，在这个假定下仍然可以估计到很接近真实值的结果。

多重插补和贝叶斯估计的思想是一致的，但是多重插补弥补了贝叶斯估计的几个不足。

(1)贝叶斯估计以极大似然的方法估计，极大似然的方法要求模型的形式必须准确，如果参数形式不正确，将得到错误得结论，即先验分布将影响后验分布的准确性。而多重插补所依据的是大样本渐近完整的数据的理论，在数据挖掘中的数据量都很大，先验分布将极小的影响结果，所以先验分布的对结果的影响不大。

(2)贝叶斯估计仅要求知道未知参数的先验分布，没有利用与参数的关系。而多重插补对参数的联合分布作出了估计，利用了参数间的相互关系。

```
>library(mice)
>data(sleep，package="VIM")
>imp<-mice(sleep，m=5，defaultMethod="pmm"，seed=1234)


>fit<-with(imp,lm(dream~span+Gest))
>pooled<-pool(fit)
>summary(pooled)
```

其中是以sleep数据（VIM包中动物睡眠数据）为例展开的，mice（）就是进行插补的函数，它里面m参数是指生成完整数据集的个数，defaultMethod参数是指选择填充缺失值的方法，seed参数是为了保证结果的重复性；with（）就是我们进行标准方法流程对填充好的完整数据集进行分析，这里面我们选择的是线性模型来分析做梦（Dream）变量与另外两个变量（Span，寿命；Gest，妊娠期）之间的线性关系；pool（）函数整合最终的结果。

5.2.11 C4.5方法

通过寻找属性间的关系来对遗失值填充。它寻找之间具有最大相关性的两个属性，其中没有遗失值的一个称为代理属性，另一个称为原始属性，用代理属性决定原始属性中的遗失值。这种基于规则归纳的方法只能处理基数较小的名词型属性。

就几种基于统计的方法而言，删除元组法和平均值法差于hot deck、EM和MI；回归是比较好的一种方法，但仍比不上hot deck和EM；EM缺少MI包含的不确定成分[46]。值得注意的是，这些方法直接处理的是模型参数的估计而不是空缺值预测本身。它们合适于处理无监督学习的问题，而对有监督学习来说，情况就不尽相同了。譬如，你可以删除包含空值的对象用完整的数据集来进行训练，但预测时你却不能忽略包含空值的对象。另外，C4.5和使用所有可能的值填充方法也有较好的补齐效果，人工填写和特殊值填充则是一般不推荐使用的。

补齐处理只是将未知值补以我们的主观估计值，不一定完全符合客观事实，在对不完备信息进行补齐处理的同时，我们或多或少地改变了原始的信息系统。而且，对空值不正确的填充往往将新的噪声引入数据中，使挖掘任务产生错误的结果。因此，在许多情况下，我们还是希望在保持原始信息不发生变化的前提下对信息系统进行处理。这就是第三种方法：

5.3 不处理

直接在包含空值的数据上进行数据挖掘。这类方法包括贝叶斯网络和人工神经网络等。

贝叶斯网络是用来表示变量间连接概率的图形模式，它提供了一种自然的表示因果信息的方法，用来发现数据间的潜在关系。在这个网络中，用节点表示变量，有向边表示变量间的依赖关系。贝叶斯网络仅适合于对领域知识具有一定了解的情况，至少对变量间的依赖关系较清楚的情况。否则直接从数据中学习贝叶斯网的结构不但复杂性较高（随着变量的增加，指数级增加），网络维护代价昂贵，而且它的估计参数较多，为系统带来了高方差，影响了它的预测精度。当在任何一个对象中的缺失值数量很大时，存在指数爆炸的危险。

人工神经网络可以有效的对付空值，但人工神经网络在这方面的研究还有待进一步深入展开。人工神经网络方法在数据挖掘应用中的局限性

讨论

我们如何识别缺失值呢？在R语言中，

1. 缺失值被标记为NA，判断一个值是不是缺失值用is.na函数
2. NaN是说这个值不是一个数，代表的是不可能的值，判断用is.nan函数
3. infinite表示无穷值，Inf表示正无穷，-Inf表示负无穷，判断使用is.infinite函数

总结

大多数数据挖掘系统都是在数据挖掘之前的数据预处理阶段采用第一、第二类方法来对空缺数据进行处理。并不存在一种处理空值的方法可以适合于任何问题。无论哪种方式填充，都无法避免主观因素对原系统的影响，并且在空值过多的情形下将系统完备化是不可行的。从理论上来说，贝叶斯考虑了一切，但是只有当数据集较小或满足某些条件（如多元正态分布）时完全贝叶斯分析才是可行的。而现阶段人工神经网络方法在数据挖掘中的应用仍很有限。值得一提的是，采用不精确信息处理数据的不完备性已得到了广泛的研究。不完备数据的表达方法所依据的理论主要有可信度理论、概率论、模糊集合论、可能性理论，D-S的证据理论等。

参考资料：

- **http://www.wenhuaxuan.com/shujuchuli/2013-09-27/7019.html（超赞）(http://www.wenhuaxuan.com/shujuchuli/2013-09-27/7019.html（超赞）)**
- **http://www.itongji.cn/article/100311B2012.html (http://www.itongji.cn/article/100311B2012.html)**
- **https://mp.weixin.qq.com/s?__biz=MzI3MTM3OTExNQ==&mid=2247484057&idx=1&sn=0a3fa0da1dde77f0e977cb3fcb573a66&chksm=eac3fd5dddb4744b440845fb75d994cbd0c773e8280cd907c2f72da77515c39fdeae544ce312&scene=21#wechat_redirect (https://mp.weixin.qq.com/s?__biz=MzI3MTM3OTExNQ==&mid=2247484057&idx=1&sn=0a3fa0da1dde77f0e977cb3fcb573a66&chksm=eac3fd5dddb4744b440845fb75d994cbd0c773e8280cd907c2f72da77515c39fdeae544ce312&scene=21#wechat_redirect)**

NA
(/TAGS/NA/)

About Sam

专注生物信息 专注转化医学

«PREVIOUS
【2.6】数据标准化
(/math/statics_topic/data-normalization/)

NEXT»
【2.4】异常值(离异值)检测
(/math/statics_topic/outlier-detection/)