《R 语言数据分析方法与实验》

小组课程设计指南

高宝俊

2021 年 4 月

一、问题的提出与背景

在做数据分析时，我们经常会遇到以下情景：所研究的变量分别在不同的数据集里。在这种情况下，我们常常需要通过两个不同数据集中的关联字段进行连接匹配，最终使多个数据集能够正确连接起来。连接两张表的公共字段，通常为其中一个表的主键（PK），另一张表的外键（FK），并且这些字段为数值型或比较短的字符型。此时，连接是比较容易实现的，连接结果的质量也是比较有保证的。

然而，在真实的场景下，往往需要研究者首先判断、筛选、并不断改进连接条件。连接的字段不一定是主键和外键，而可能是任意字段；字段类型可能不是简单的数值和短文本类型的变量，而是相对比较复杂的长文本类型。文本类型的连接字段可能并不是干净的数据，因此首先需要做初步的清洗。因此，提出了本课程设计的**问题**：通过多种方法匹配 TripAdvisor 和酒店纳税两个数据集。

问题的背景

互联网特别是移动互联网的发展，催生了用户生成的在线评论等社交媒体的繁荣。在线评论有助于消除在线环境下的信息不对称，影响用户的决策和行为，有利于产品的销售，具有重要的理论和实践价值。这一问题近十年在营销、信息系统、产业组织等领域产生了大量的研究成果。要探究在线评论如何影响产品或服务的销售，就需要获得评论的数据和销售的数据，并且将二者连接起来。有关这一领域的研究，可以参考如下文献。

References：

Babić Rosario, A., Sotgiu, F., De Valck, K., & Bijmolt, T. H. (2016). The effect of electronic word of mouth on sales: A meta-analytic review of platform, product, and metric factors. Journal of Marketing Research, 53(3), 297-318.

Hollenbeck, B. (2018). Online reputation mechanisms and the decreasing value of chain affiliation. Journal of Marketing Research, 55(5), 636-654.

二、教学目标

知识目标

本部分旨在训练以下知识点：

（1）readr：　文件的读取和存储；向量解析等

（2）dplyr：　数据转换；表的连接；表的合并等

（3）stringr：　字符串的提取、合并；正则表达式模式匹配等

（4）其他：　常见的 R base 函数、处理日期时间的 lubridate 包（可选），以及模糊匹配的方法和文本处理的基础技能等。

技能目标

（1）如何判断并改进连接条件

（2）如何通过较长的文本字段连接两个数据集

（3）如何用多种方法（精确、模糊匹配，辅以人工验证）提高连接成功率

（4）如何采用多种方法验证连接的准确性

能力目标

（1）数据质量与数据意识

（2）在实践中学习新知识的能力

（3）解决复杂实际问题的能力

三、数据集介绍

本部分涉及两个数据集，以下是数据集中关联字段的介绍。

数据集 1：TripAdvisor Data

数据来源：https://www.tripadvisor.com/

数据内容：美国 Texas 州酒店的相关信息

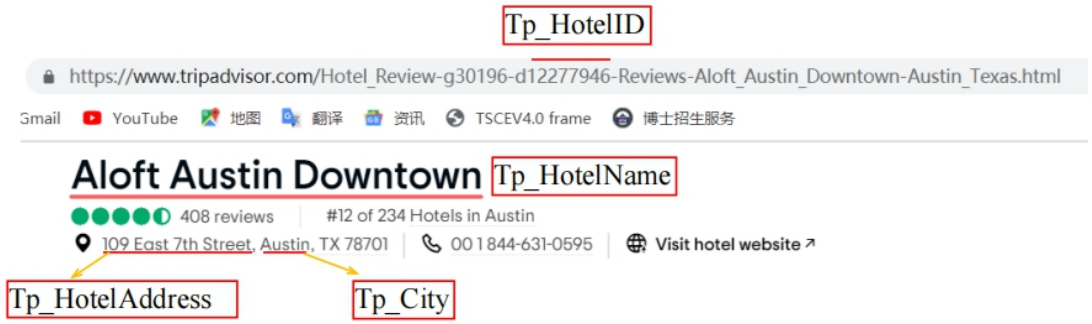
字段信息（部分字段）

Tp_HotelName：酒店名称

Tp_HotelAddress：酒店地址

Tp_City: 酒店所在城市

Tp_HotelID：TripAdvisor 中酒店的 ID ，如图所示，HotelUrl 中“12277946”为 Tp_HotelID。它是 TripAdvisor 数据集中酒店的唯一标识，即主键。



数据集 2：Texas tax Data

数据来源：Texas 州税务部门网站

数据内容：美国 Texas 州酒店税收的相关信息

字段信息（部分字段）

Tax_HotelID: 设定的酒店 ID，作为该数据集中每个酒店的唯一标识

Tax_HotelName：酒店名称

Tax_HotelAddress：酒店地址

Tax_City：酒店所在城市

四、解题思路及重难点

1 字段的清洗处理

提示：相关关联字段可能存在重复、缺失、大小写不一致、字符前后有空格等问题，需要清洗处理后再进行后续匹配工作。文本预处理涉及的主要有：

（1）查找是否有重复、缺失等（distinct、unique、is.na 等）

（2）大小写统一问题（toupper、tolower 等）

（3）去空格（str_trim）

2 设计匹配策略

提示：

（1）仅根据酒店地址匹配可能会导致错误，因为酒店地址不能唯一标识一家酒店，有些地址只精确到街道，同一条街道上有多家酒店。

（2）仅根据酒店名称匹配也是不行的，因为一方面酒店名称不能唯一标识一家酒店，比如希尔顿酒店，很多城市都有希尔顿酒店；另一方面名称在两个数据集中的拼写方式也可能不同。

（3）因此，建议按照酒店名称和酒店地址双重条件进行匹配，即两个数据集中地址和名称两个字段均能匹配上的记录才是正确的匹配。

（4）由于匹配的字段是文本，而表达同样含义的文本可能存在微小差异。

这样，如果只考虑精确匹配（匹配条件为完全相等），无论是使用内连接还是左连接，匹配上的样本将非常少。因此，除了精确匹配之外，还需要考虑模糊匹配。

（5）由于要考虑模糊匹配，因此在匹配之前，首先基于两个数据集生成一个笛卡尔乘积的连接结果，即表 A 的每一行记录都与表 B 的任意一行记录是生成一个连接结果。

（6）由于生成的笛卡尔乘积数据集的规模会非常大（m*n 行），为了降低复杂度，在生成笛卡尔乘积时候，可以首先按照城市分块，两个数据集中同一城市的记录才会进行笛卡尔乘积连接。

（7）得到笛卡尔乘积的连接结果之后，所有的操作都在这一数据集上。建议按精确匹配、模糊匹配和人工匹配三个步骤的顺序进行（这样可以减少在进行模糊匹配时的笛卡尔乘积），逐步筛选出匹配好的结果，并逐步将已经匹配好的记录从笛卡尔乘积数据集中删除。

3 精确匹配

提示：

Tp_HotelAddress == Tax_HotelAddress & Tp_HotelName == Tax_HotelName

4 模糊匹配

模糊匹配简介

本部分的模糊匹配实际上是字符序列间相似度的比较，相似度高的字符序列可一定程度上看作“匹配上”了。字符序列间差别的度量有很多方式，常见的一种便是 Levenshtein Distance。

Levenshtein Distance 是一个度量两个字符序列之间差别的字符串度量标准，两个字符序列之间的 Levenshtein Distance 是将一个字符序列转换为另外一个字符序列所需的单字符编辑（插入、删除或替换）的最小数量，它是 1965 年由苏联数学家 Vladimir Levenshtein 发明的。Levenshtein Distance 也被称为编辑距离（Edit Distance）。

原理

首先考虑极端状况，当 a 或 b 长度为 0 时，那么需要编辑的次数就是另一个字符串的长度。而后再考虑通常状况，此时分为三种状况：

1.在 k 个操作中，将 a[1...i]转换为 b[1...j-1]

2.在 k 个操作中，将 a[1...i-1]转换为 b[1...j]

3.在 k 个操作中，将 a[1...i-1]转换为 b[1...j-1]

针对第一种状况，只须要在 a[1...i]后加上字符 b[j]，便可完成 a[1..i]到 b[1...j]的转换，总共须要的编辑次数即为 k+1。

针对第二种状况，只须要在 a[i]字符从 a 中移除，便可完成 a[1..i]到 b[1...j]

的转换，总共须要的编辑次数即为 k+1。

针对第三种状况，只须要将 a[i]转换为 b[j]，便可完成 a[1..i]到 b[1...j]的转换，若是 a[i]与 b[i]的字符相同，则总共须要的编辑次数为 k，不然即为 k+1

上述三种状况分别对应于插入、删除、替换操作。

为了保证将 a[1..i]转换为 b[1..j]的操作数是最少的，只需要从三种状况中选择操做次数最少的状况，同理为了保证三种状况的操作数也是最小的，只需要按此逻辑进行迭代保证每一步的操作数都是最小的便可。

R 语言中有实现该功能的包和方法（RecordLinkage::levenshteinSim），RecordLinkage 度量字符序列间差异的有 levenshteinDist 和 levenshteinSim 两个函数。其中 levenshteinDist 会生成编辑距离，但它不能用作准确比较字符序列间相似度的标准，在此基础上，levenshteinSim 做了改进，它的计算方式是 1 - d(str1,str2) / max(A,B)，其中 d 是编辑距离，A 和 B 是字符序列的长度，levenshteinSim 的取值范围为[0,1]，越接近 1，相似度越大。

References：

Winkler, W.E.: String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In: Proceedings of the Section on Survey Research Methods, American Statistical Association (1990), S. 354‐369.

除了 Levenshtein Distance，还有很多其他度量字符序列间差异的方式，除了本文提供的度量方式，大家也可以尝试采取其他方法来进行模糊匹配，具体可参考：https://zhuanlan.zhihu.com/p/91645988。

模糊匹配

提示：

（1）Tp_HotelAddress = Tax_HotelAddress，Tp_HotelName 和 Tax_HotelName 有差别，通过比较两者的 levenshteinSim。

（2）Tp_HotelName = Tax_HotelName，Tp_HotelAddress 和 Tax_HotelAddress 有差别，通过比较两者的 levenshteinSim。

可以将 levenshteinSim 计算得到的结果设置一个相似度的临界（cutoff），比如设置 levenshteinSim > 0.7，在此区间的可以暂时看作匹配上了，然后通过人工核查最终确定匹配上的样本。

模糊匹配得到的结果并不一定是正确的，因此人工核查对于最终数据质量至关重要。对于不同匹配方式得到的结果，我们的信心是不同的。因此，在生成数据的过程中，可以设置一系列标记字段来进行区分。特别是接受模糊匹配的原因要记录下来，以便于交叉验证。这是数据分析实践中重要的质量提升准则，在代码和最终结果中加上如下字段：

是否精确匹配（If_Exact）：逻辑型

模糊匹配是否高于阈值（If_Above）：逻辑型

模糊匹配 Name 的相似度（Lev_Sim_Name）：数值型

模糊匹配 Address 的相似度（Lev_Sim_Address）：数值型

接受模糊匹配的依据（Reason）：长文本，给出原因。

5 模糊匹配进一步探索（可选）

提示：可能存在 Tp_HotelAddress 和 Tax_HotelAddress，Tp_HotelName 和 Tax_HotelName 都有差别的情况，这种情况下可以合并 Tp_HotelAddress 和 Tp_HotelName，Tax_HotelAddress 和 Tax_HotelName 形成两个新的字段，比较两个新字段的 levenshteinSim。

6 寻求其他方法，人工匹配

上述精确与模糊匹配不可能实现两个数据集百分之百的匹配。当自动化方法无效时，应当人工介入，通过搜索、寻找证据、判断等方法，进一步提升数据的匹配精度。

重难点：

模糊匹配

五、示例

Texas tax 中有一条数据是

LocationName	LocationAddress
RED ROOF INN	2960 W SAM HOUSTON PKWY S

TripAdvisor 有一条数据是

HotelName	HotelAddress
Red Roof Inn Houston - Westchase	2960 W Sam Houston Pkwy S

这两条数据有可能是同一家酒店。

由此可见，在 Name 上是一个模糊匹配。这个例子也给了我们一些启示：如果通过观察发现，TripAdvisor 的 Name 后面总是带有城市的名称的话，首先去掉城市名称可能能实鲜精确匹配，至少能提高模糊匹配的精度。

六、提交内容

本次作业需要用 Markdown 撰写，作业中加入必要的描述性文字。提交 Markdown 文件和编译好的 html 文件，以及匹配好的 CSV 文件。