特征工程（中）-特征表达

原创 stephenDC 大数据与人工智能 2019-07-31

作者丨stephenDC

这是作者的第14篇文章

在上一篇《特征工程（上）—特征选择》中，我们解决了从哪些维度去刻画一个对象的问题。

在本篇中我们聊一下特征表达（或者说特征编码）的问题，即从这些选定的维度，如何去刻画特定的对象。

01

特征表达要考虑哪些方面?

从一个完整的机器学习任务来看，在选择完特征之后，特征表达的任务就是要将一个个的样本抽象成数值向量，供机器学习模型使用。因此，特征表达就要兼顾特征属性和模型需求这两个方面。

- **特征属性**

特征按其取值类型不同，可以简单分为连续型和离散型。而离散型特征，又可以分为类别型和序列型。下面依次简要说明。

连续型特征：取值为连续实数的特征。
比如，身高，175.4cm。

类别型特征：取离散值，表示没有比较关系的类型。
比如，血型有A型、B型、AB型和O型4种，它们各自为一个独立类型。

序列型特征：取离散值，表示有比较关系的类型。
比如，还是身高，但取值为“高”、“中”、“低”3种类型。

- **模型需求**

如果你在公司负责建模调优，那你对负责特征工程的同事，会有什么样的需求呢？换言之，你希望他们给你什么样的特征呢？

应该不外乎这么几点，**类型匹配、特征准确性、特征完备性和方便模型训练。**

类型匹配：对一个树模型而言，其原理是对特征进行切分，因此特征的值是否是数值类型，以及是否有缺失值，都可以不影响模型运算；但对其他基于数学运算的模型，则特征必须转化为数值类型，且对缺失值要有相应的处理逻辑。

准确性：拿身高来说，粗略地分为“高”、“中”、“低”3个类型，大致是对的，但并不足够准确。比如，在“高”这个类型中的人，因为分类太粗糙，已经没办法再进行比较了。

完备性：完备性是说，你的特征是否可以尽可能多方面的刻画一个对象。这一点跟所使用的模型有很大的关系。对简单模型来说，希望特征可以足够复杂，这就可能需要考虑高阶特征。

（比如，对一个电影而言，主演=“杨洋”和类型=“功夫”，单独来看这两者对你可能都很有吸引力，但都是从“主演+类型”这个维度，还是算了吧。）

但对复杂模型而言，如FM和神经网络，其模型结构本身就有一定的特征交叉功能，因此可以不用再考虑高阶特征。

方便模型训练：从模型参数的求解来看，如果特征做过归一化处理，可以有效避免在解空间中形成“峡谷”，从而加速参数求解过程。

探讨完特征表达需要考虑的因素，下面我们就可以有的放矢，讨论一下特征表达的技术问题。

02

连续型特征

上面说过，根据模型的需要，特征需要做连续化或者离散化的处理。连续特征已无需再做连续化处理，可以把特征的值直接拿来用，最多再做个归一化什么的就够了。

- **连续特征的离散化**

方法主要有两种，阈值分组和模型离散。

先说阈值分组，以出生日期为例，如果模型不需要知道一个人具体在哪一分哪一秒出身，很多情况下以年月为阈值划分就足够了。

模型离散的话，我们以树模型为例。树模型是靠对特征空间进行分割，并在每个子空间中用常量建模，得到预测结果的。

特征划分的结果，最终反映为树的叶子结点，因此用某个连续值特征被划分到哪个叶子结点，自然就实现了连续特征的离散化。

03

离散型特征

对离散特征，我们考虑其连续化和离散化的过程。

- **离散特征的连续化**

一篇文章，由很多不同的单词组成；一个视频，则可以有很多的标签，如演员、导演、地区、语言、豆瓣评分等。单词和标签都是离散的，如何得到一个取连续值的特征呢？

下面介绍One hot、TF-IDF和embedding三种方法。

1.One hot 编码

One hot编码依赖一个由所有“单词”组成的“词典”。将词典里的单词排一个固定顺序，假设有10000个单词，即对应一个10000维的向量。对不同的文章，如果对应的单词在文章中出现，即将相应的维度编码为1，否则为0。这样所有的文章都会转化为10000维的向量。

2.TF-IDF编码

One hot编码有两个问题，一是没有考虑某个单词在一篇文章中出现的次数，二是没有考虑不同单词的刻画能力大小。

某个单词在一篇文章中出现的次数除以文章里单词总数，称之为“词频”，即TF: Term Frequency。

不同的单词，对属性的刻画能力差别很大。比如，“的”这种词，可能在各种文章中都会频繁出现，但不足以说明这篇文章的类型；但“债券”、“期权”这种词，在一篇文章中出现，就说明这篇文章很大可能是在讲金融相关。

所以，一个单词在多少比例的文章中出现过，也是一个很重要的因素。对这个比例取倒数，然后取对数，称之为“逆向文本频率”，即IDF: Inverse document frequency。

综合考虑“词频”和“逆向文本频率”，求两者的乘积，就有了TF-IDF的编码方式。

3、Embedding编码

One hot和TF-IDF的编码，都把每个单词或标签当成一个独立的个体（在特征空间中是彼此正交的），而没有考虑它们之间的联系。比如，“成龙”、“李连杰”作为电影的标签，两者之间的联系，显然比和“巩俐”这个标签的联系更大。

Embedding的编码，考虑了各个单词之间的联系，将这些单词嵌入到了一个低维的特征空间中，从而实现了一种既能表征彼此之间联系又能降维的编码方式。限于篇幅，这里不再详述，有兴趣的同学可以去研究Word2vec等Embedding模型。

- **离散特征的离散化**

离散特征的取值，不一定都需要转化为数值，比如上文提到的树模型；但对更多模型而言，特征的值是需要做数值运算的，因此对离散特征有时也需要做离散化的处理。对类别型特征和序列型特征，其处理方式又有所不同。

对类别型特征，可以采用上文提到的One hot编码。

比如，对类型1、2、3和4，可以分别编码为（1 0 0 0）、（0 1 0 0）、（0 0 1 0）和（0 0 0 1）。为了缩短编码的长度，也可以采用二进制编码，则同样的这4个类型，可以分别编码为（0 0 1）、（0 1 0）、（0 1 1）和（1 0 0）。

对序列型特征，在编码的时候，需要考虑维持原来特征的大小关系。比如，对身高的“高”、“中”、“低”而言，有“高”> “中”>“低”的关系，那么编码出来也要维持这种关系。如果分别编码为t1、t2和t3，则t2必需要介于t1和t3之间。

04

特殊特征的处理

有时候，根据模型的需要，需要对一些特征做特殊处理。这里以时间特征和地理特征为例，进行说明。

对时间特征，有时候模型用到的并不是其绝对量，而是相对量，这个情况下就需要求差值。

比如，想知道一个影片热度，需要参考上映年份这个特征，但模型要的并不是“2018”这个值本身，以2000年为基准的话，“2018”转化为“18”才是需要的。另外，时间也可以按“年-月-日-小时”等拆解为层级特征，这样金融危机可能就跟年份有关，而气温和降雨量等具有周期性的量，就会和月份有关。

对地理特征，也需要根据模型的需要，选择特定的处理方式。比如，对一个全球的GPS系统应用，地理特征很好的标示是（经度，维度）。但对一个邮政或者外卖这样的系统而言，更好用的还是拆解的层级特征，“省-市-区-县-乡-镇”等。

05

缺失值处理

数据来源及处理过程中的各种因素，都可能会导致特征出现缺失值的问题。对树模型来说，缺失值不是个问题，模型可以自行处理。但对其他大多数模型而言，缺失值是需要处理掉的。

缺失值处理的方式主要有两种，筛除和填充。

如果样本足够多，而有缺失值的样本占总体的比例并不大，这种情况下，可以采用最简单粗暴的方式，直接筛除掉就好了。但如果，样本本来就不够充分，又或者有缺失值的样本占了很大比例，这时候就需要考虑对缺失值进行填充了。对连续型特征，一般用均值或者中位数进行填充；而对离散型特征，则更多用众数进行填充。

小结

本文在特征选择的基础上，进一步讨论了特征表达的问题，主要涉及连续和离散型特征的编码方式、特殊特征的处理和缺失值处理等方面。

对文中提到的归一化，我们认为也是特征表达的一个方面，但这个问题不太核心，且限于篇幅，不再详述。下篇文章将是特征工程系列的最后一篇，届时会讨论特征评估的问题。