031 | 文档理解的关键步骤：文档聚类

2017-12-13 洪亮劼

AI技术内参 进入课程 >

讲述：初明明
时长 07:24 大小 3.40M

周一我们分享了文档理解最基本的一个步骤，那就是给文档分类（Classification），主要是看不同文档表达什么类别的信息。今天我就来聊一聊文档理解的另外一个重要组件：**文档聚类**（Document Clustering）。

文档聚类的类型

和了解文档分类的思路相似，我们先来看看文档聚类的分类。一般来说，可以把文档聚类看作非监督学习的典型代表。

先说一种直观的分类方法。如果把文档分为“互不相关”的几个聚类，那就叫作“**扁平聚类**”（Flat Clustering）；如果这些聚类相互之间有一定的结构关系，那就叫作“**层次聚类**”（Hierarchical Clustering）。

“扁平聚类”中的“互不相关”是说文档所划分进去的聚类之间本身没有重合。而“层次聚类”的特点是，希望在聚类之间找到关系，从而把这些文档组织到一个有层次的结构中。在这种层级结构里，根节点所代表的内容往往比较抽象，而叶节点所表达的内容则比较具体。

值得注意的是，不管是“扁平聚类”还是“层次聚类”，相较于文档分类来说，这里最大的不同就是这些聚类以及它们之间的关系都不是事先定义好的，或者说研发人员事先并不知道这些聚类的存在。从这个角度来看，聚类的确是比分类要困难的任务，难在如何衡量聚类的好坏。

除了“扁平聚类”和“层次聚类”这种区分以外，聚类方法中还有一个类似的区分，那就是“**硬聚类**”（Hard Assignment）和“**软聚类**”（Soft Assignment）的区别。

顾名思义，“硬聚类”是说对于每一个文档，不管是“扁平聚类”还是“层次聚类”，都确定性地分配到一个或者一组聚类中。而“软聚类”则往往学习到文档分配到聚类的一个分布，也就是说所有的分配都是以某种概率存在的。

文档聚类的应用

在搜索系统为背景的场景中，我们为什么要强调文档聚类？

首先，文档聚类可以帮助文档提取和排序。很多文档能够聚合到一个类别肯定是因为文档在某种情况下“相似”。相似的文档很可能都满足用户的某种“信息需求”（Information Needs）。实际上，在类似“语言模型”（Language Model）或者其他概率模型的场景中，对文档相关度的预测经常需要从相似文档群体中寻找额外信息。

举个例子，在“语言模型”中，我们需要估计文档相对于查询关键字的相关度。单独的某一个文档，数据信息可能比较匮乏，因此一个常用的策略就是从整个数据集中补充信息。如果我们已经有了文档的聚类，那自然就可以从这些聚类中补充，而不需要数据全集。

其次，文档聚类能够帮助整理搜索结果。在最普通的搜索结果上，如果只是完全“平铺”所有的结果，用户很可能对成百上千的结果“不得要领”。因此，在这些结果上体现某种结构就成为了很多搜索引擎提升用户体验的一种方法。

当然，这里可以用我们之前提到的“文档分类”的方法，把返回的结果按照类别组织。这样，哪一个类别有什么结果就清清楚楚。在这里，文档聚类相比于文档分类的优势是，聚类

更能反应文档之间更本质的联系，而不是类似于分类这样“先入为主”地对文档的关系有一个定义。

文档聚类不仅仅是搜索结果的展示利器，**很多时候，文档聚类还可以帮助研究人员来浏览一个文档集合，而不需要太多的先期假设。**在有“层次聚类”的帮助下，研发人员可以很容易地根据层次之间的关系来对一个文档集合进行分析。利用文档聚类来浏览文档集合常常是发现问题，并且进行下一步工作的有效步骤。

文档聚类的基本模型

最基础的文档“扁平聚类”方法当属“K 均值算法”（K-Means）。

首先，一个最基本的步骤就是要把文档表示成“特性向量”（Feature Vector）。具体的做法可以采用我们周一讲过的几个方式，比如最基本的“词袋模型”（Bag Of Word），这是一种把文字顺序完全打乱的方式。在“词袋模型”中，每个词的权重可以用我们之前介绍过的 TF-IDF 或是语言模型对单词进行加权。当然，还有“N 元语法”（N-gram）和“递归神经网络”（RNN）两种思路，这一部分可以回到我们周一的内容再复习一下。

把文档表达成为“特征向量”之后，就可以开始聚类了。“K 均值算法”的基本思路是这样的。给定一个数据样本集，K 均值算法尝试把所有的样本划分为 K 个聚类。每个聚类都是互斥的，也就是说样本都被有且唯一地分配到这些聚类中。K 均值算法在优化一个目标函数，那就是每个样本到目标聚类中心的平均平方误差最小。

这里，目标聚类中心是指当前这个样本被分配到的聚类；而聚类中心则是所有被分配到这个聚类的样本的均值。很明显，根据不同的样本被分配到不同的聚类，聚类中心也会随之发生变化。通俗地说，K 均值算法的目标函数要达到的目的是，让聚类内部的样本紧紧围绕在聚类的均值向量周围。整个目标函数的值越小，聚类内样本之间的相似度就越高。

和我们熟悉的线性回归模型（Linear Regression）以及对数几率回归（Logistic Regression）一样，目标函数本身仅仅描述了当最终的聚类分配最佳时的一种情况，并没有描述如何能够得到最佳聚类分配的情况。实际上，对于 K 均值算法而言，直接最小化这个目标函数并不容易，一般来说，找到它的最优解是一个 NP 难的问题。

不过幸运的是，**贪心算法一般能够找到不错的近似解。**下面我就介绍一个通过迭代优化来近似求解目标函数的算法。

首先，我们对均值向量进行初始化。比较简单的初始化方法就是直接随机地选择某几个点来当做聚类均值。**然后，我们依次对每一个样本点进行聚类划分。**每个数据点被分配到距离某一个均值向量最近的那个聚类里。当我们进行了所有的分配之后再对均值向量更新。这就完成了一次迭代，整个算法需要进行多次迭代更新。**若迭代更新后聚类结果保持不变，就将当前聚类划分结果返回。**

文档聚类的难点

在今天分享的最后，我想来谈一谈文档聚类的一些难点。

首先，怎样衡量聚类的质量好坏，也就是如何评价聚类算法以及比较不同的算法，一直都是聚类模型，甚至说是无监督机器学习算法的共同问题。有一些评价手段基于定义聚类内部数据的相似度，并且认为聚类内部数据应该比聚类之间的数据更加相似。然而，这样的定义并不能真正反映聚类的质量。

其次，在聚类算法中，往往有一个参数非常难以决定，那就是聚类的个数。对于一个决定的数据集来说，我们不可能事先知道这个参数。当聚类的个数过少的时候，我们可能无法对数据集进行比较完备的 K 均值算法描述。而聚类的个数过多的时候，可能数据又被切割成过多的碎片。因此，要确定这个参数就成了聚类算法研究的一个核心难点。

小结

今天我为你讲了文档理解中的文档聚类问题。一起来回顾下要点：第一，简要介绍了文档聚类的类型。第二，详细介绍了文档聚类的应用场景。第三，讲解来一个基本的文档聚类 K 均值算法。第四，简要提及了文档聚类的一些难点。

最后，给你留一个思考题，当得到文档聚类的结果以后，能否把这些结果用在其他任务中呢？如果可以，如何利用？

欢迎你给我留言，和我一起讨论。

上一篇　030 | 文档理解第一步：文档分类

下一篇　032 | 文档理解的重要特例：多模文档建模

精选留言 (1)

写留言

极客星星
2017-12-13

1

文档聚类的结果 我理解应该可以作为排序模型的一个特征 帮助更好的排序。此外,是不是召回时也可以利用这个信息

作者回复: 是的。