机器学习算法系列（23）：TF-IDF与余弦相似度

title:
date: 2017-07-07 23:14:45
categories: 机器学习
tags:
- TF-IDF
- 余弦相似度
- 文档检索

mathjax2: true

TF-IDF(term frequency=inverse document frequency)是一种用于资讯检索与文本挖掘的常用加权技术。TF-IDF是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。TF-IDF加权的各种形式常备搜索引擎应用，作为文件与用户查询之间相关程度的度量或评级。

一、原理

设想现在我们正在阅读新闻，如何最快速的了解新闻的主旨？毫无疑问——关键词。TF-IDF就具有这样的能力：提取关键词。

1.1 TF

假设一个词在一篇文章中出现的次数越多，那么它就越”紧扣主题”。以本文为例，我们可以统计词频(TF)，不难发现“TF-IDF”,“应用”、“原理”是出现频率很高的词，后文称keywords。这符合我们的假设，但是有些词却出现的次数更多，如：的、是、有等。这类词语没有明确意义，我们称为停顿词(Stopwords)。

如果单纯按照词频算关键词，你会发现几乎所有的文章都是stopwords的词频最高。换句话说，像这种”万金油”，是没有区分度的词语，不能很好的起到将文章分类的作用。

此外，抛开停用词，如果该文档中的几个词出现的频率一样，也不意味着，作为关键词，它们的

重要性是一致的。比如这篇文档中，“TF-IDF”、“意义”、“文档”这三个词的词频出现的次数一样多，但因为“意义”是很常见的词，相对而言，“TF-IDF”、“文档”不那么常见。即使它们的词频一样，我们也有理由认为，“TF-IDF”和“文档”的重要性大于“意义”，也就是使，在关键词排序上，“TF-IDF”和“文档”也应该排在“意义”的前面。

所以，我们需要一个重要性调整系数，衡量一个词是不是常见词。如果某个词比较少见，但是它在这篇文章中多次出现，那么它很可能就反映了这篇文章的特性，正是我们所需要的关键词。这时就需要祭出逆文档频率(IDF)来解决词语权重的问题。

1.2 IDF

用统计学语言表达，就是在词频的基础上，要对每个词分配一个"重要性"权重。最常见的词（"的"、"是"、"在"）给予最小的权重，较常见的词（"中国"）给予较小的权重，较少见的词（"蜜蜂"、"养殖"）给予较大的权重。这个权重叫做"逆文档频率"（Inverse Document Frequency，缩写为IDF），它的大小与一个词的常见程度成反比。

知道了"词频"（TF）和"逆文档频率"（IDF）以后，将这两个值相乘，就得到了一个词的TF-IDF值。某个词对文章的重要性越高，它的TF-IDF值就越大。所以，排在最前面的几个词，就是这篇文章的关键词。

1.3 公式化表达

对于在某一特定文件里的词语t_i来说，它的重要性可表示为：

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

以上式子中$n_{i,j}$是该词在文件d_j中的出现次数而分母则是在文件d_j中所有字词的出现次数之和。

逆向文件频率（inverse document frequency，idf）是一个词语普遍重要性的度量。某一特定词语的idf，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取对数得到：

$$IDF_i = log\frac{|D|}{|j : t_i \in d_j|}$$

其中

- $|D|$：语料库中的文件总数
- $|j : t_i \in d_j|$：包含词语t_i的文件数目（即$n_{i,j} \neq 0$的文件数目）如果该词语不在语料库中，就会导致分母为零，因此一般情况下使用$1 + |j : t_i \in d_j|$

然后

$$TF - IDF = TF_{i,j} \times IDF_i$$

某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的tf-idf。因此，tf-idf倾向于过滤掉常见的词语，保留重要的词语。

1.4 应用

我们通过Google搜索结果数为例，将含有中文“的”结果数15.8亿作为整个语料库大小，计算一些关键词和停用词的TF-IDF值。为了计算简便，假设全文分词后一共500词，则结果如下：

	包含该词的文章(百万)	IDF	TF	TF-IDF
TF-IDF	0.497	3.502	0.018	0.063
原理	24.4	1.811	0.008	0.014
应用	82.8	1.280	0.008	0.010
是	363	0.638	0.028	0.018
有	482	0.515	0.026	0.013
的	1580	0.000	0.080	0.000

TF-IDF的优点是计算简单，利于理解，性价比极高。但是它也有缺陷，首先单纯依据文章中的TF来衡量重要性，忽略了位置信息。如段首，句首一般权重更高；其次，有的文章可能关键词只出现1-2次，但可能通篇都是围绕其进行阐述和解释，所以单纯靠TF仍然不能解决所有的情况。

二、余弦相似度

余弦相似性通过测量两个向量的夹角的余弦值来度量它们之间的相似性。0度角的余弦值是1，而其他任何角度的余弦值都不大于1；并且其最小值是-1。从而两个向量之间的角度的余弦值确定两个向量是否大致指向相同的方向。两个向量有相同的指向时，余弦相似度的值为1；两个向量夹角为90°时，余弦相似度的值为0；两个向量指向完全相反的方向时，余弦相似度的值为-1。这结果是与向量的长度无关的，仅仅与向量的指向方向相关。余弦相似度通常用于正空间，因此给出的值为0到1之间。

注意这上下界对任何维度的向量空间中都适用，而且余弦相似性最常用于高维正空间。例如在信息检索中，每个词项被赋予不同的维度，而一个文档由一个向量表示，其各个维度上的值对应于该词项在文档中出现的频率。余弦相似度因此可以给出两篇文档在其主题方面的相似度。

2.1 定义

两个向量间的余弦值可以通过使用欧几里得点积公式求出：

$$a \cdot b = |a| \cdot |b| \, cos\theta$$

给定两个属性向量A和B，其余相似性θ由点积和向量长度给出，如下所示：

$$similarity = cos(\theta) = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

这里的A_i和B_i分别代表向量A和B的各分量。

给出的相似性范围从-1到1：-1意味着两个向量指向的方向正好截然相反，1表示它们的指向是完全相同的，0通常表示它们之间是独立的，而在这之间的值则表示中间的相似性或相异性。

对于文本匹配，属性向量A 和B 通常是文档中的词频向量。余弦相似性，可以被看作是在比较过程中把文件长度正规化的方法。

在信息检索的情况下，由于一个词的频率（TF-IDF权）不能为负数，所以这两个文档的余弦相似性范围从0到1。并且，两个词的频率向量之间的角度不能大于90°。

由此，我们就得到了"找出相似文章"的一种算法：

- 1）使用TF-IDF算法，找出两篇文章的关键词；
- 2）每篇文章各取出若干个关键词（比如20个），合并成一个集合，计算每篇文章对于这个集合中的词的词频（为了避免文章长度的差异，可以使用相对词频）；
- 3）生成两篇文章各自的词频向量；
- 4）计算两个向量的余弦相似度，值越大就表示越相似。

"余弦相似度"是一种非常有用的算法，只要是计算两个向量的相似程度，都可以采用它。