知乎 首发于 机器学习和自然语言处理

文本特征选择

 宝神 关注他

在做文本分类聚类的任务时，常常需要从文本中提取特征，提取出对学习有价值的分类，而不是把所有的词都用上，那样会造成维度灾难。因此一些词对分类的作用不大，比如“的、是、在、了”等停用词。这里介绍三种常用的特征选择方法：

无监督方法：

- TF-IDF

监督方法：

- 卡方
- 信息增益
- 互信息

一、TF-IDF

一个容易想到的思路，就是找到出现次数最多的词。如果某个词很重要，它应该在这篇文章中多次出现。于是，我们进行"词频"（Term Frequency，缩写为TF）统计。

结果你肯定猜到了，出现次数最多的词是----"的"、"是"、"在"----这一类最常用的词。它们叫做“停用词”（stop words），表示对找到结果毫无帮助、必须过滤掉的词。

的重要性是一样的?

显然不是这样。因为"中国"是很常见的词，相对而言，"蜜蜂"和"养殖"不那么常见。如果这三个词在一篇文章的出现次数一样多，有理由认为，"蜜蜂"和"养殖"的重要程度要大于"中国"，也就是说，在关键词排序上面，"蜜蜂"和"养殖"应该排在"中国"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。**

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。

第一步，计算词频。

$$词频(TF) = 某个词在文章中的出现次数$$

考虑到文章有长短之分，为了便于不同文章的比较，进行"词频"标准化。

$$词频(TF) = \frac{某个词在文章中的出现次数}{文章的总词数}$$

或者

$$词频(TF) = \frac{}{该文出现次数最多的词的出现次数}$$

如果一个词越常见，那么分母就越大，逆文档频率就越小越接近0。分母之所以要加1，是为了避免分母为0（即所有文档都不包含该词）。log表示对得到的值取对数。

第二步，计算逆文档频率。

这时，需要一个语料库（corpus），用来模拟语言的使用环境。

$$逆文档频率(IDF) = \log(\frac{语料库的文档总数}{包含该词的文档数 + 1})$$

如果一个词越常见，那么分母就越大，逆文档频率就越小越接近0。分母之所以要加1，是为了避免分母为0（即所有文档都不包含该词）。log表示对得到的值取对数。

第三步，计算TF-IDF。

$$TF - IDF = 词频(TF) \times 逆文档频率(IDF)$$

可以看到，TF-IDF与一个词在文档中的出现次数成正比，与该词在整个语言中的出现次数成反比。所以，自动提取关键词的算法就很清楚了，就是计算出文档的每个词的TF-IDF值，然后按降序排列，取排在最前面的几个词。

还是以《中国的蜜蜂养殖》为例，假定该文长度为1000个词，"中国"、"蜜蜂"、"养殖"各出现20次，则这三个词的"词频"（TF）都为0.02。然后，搜索Google发现，包含"的"字的网页共有250亿张，假定这就是中文网页总数。包含"中国"的网页共有62.3亿张，包含"蜜蜂"的网页为0.484亿张，包含"养殖"的网页为0.973亿张。则它们的逆文档频率（IDF）和TF-IDF如下：

	档数（亿）	IDF	TF-IDF
中国	62.3	0.603	0.0121
蜜蜂	0.484	2.713	0.0543
养殖	0.973	2.410	0.0482

从上表可见，"蜜蜂"的TF-IDF值最高，"养殖"其次，"中国"最低。（如果还计算"的"字的TF-IDF，那将是一个极其接近0的值。）所以，如果只选择一个词，"蜜蜂"就是这篇文章的关键词。

除了自动提取关键词，TF-IDF算法还可以用于许多别的地方。比如，信息检索时，对于每个文档，都可以分别计算一组搜索词（"中国"、"蜜蜂"、"养殖"）的TF-IDF，将它们相加，就可以得到整个文档的TF-IDF。这个值最高的文档就是与搜索词最相关的文档。

TF-IDF算法的优点是简单快速，结果比较符合实际情况。缺点是，单纯以"词频"衡量一个词的重要性，不够全面，有时重要的词可能出现次数并不多。而且，这种算法无法体现词的位置信息，出现位置靠前的词与出现位置靠后的词，都被视为重要性相同，这是不正确的。（一种解决方法是，对全文的第一段和每一段的第一句话，给予较大的权重。）

TF-IDF算法可以用于无监督学习，不需要知道文档的类别，但是对同一个词来说，它在不同的文档中有不同的TF-IDF值，我这里处理的策略是每篇文档取top K，然后做一个去重。

二、卡方检验

开方检验其实是数理统计中一种常用的检验两个变量独立性的方法。

开方检验最基本的思想就是**通过观察实际值与理论值的偏差来确定理论的正确与否**。具体做的时候常常先假设两个变量确实是独立的（行话就叫做“原假设”），然后观察实际值（也可以叫做观察值）与理论值（这个理论值是指“如果两者确实独立”的情况下应该有的值）的偏差程度，如果偏差足够小，我们就认为误差是很自然的样本误差，是测量手段不够精确导致或者偶然发生的，两者确确实实是独立的，此时就接受原假设；如果偏差大到一定程度，使得这样的误差不太可能是偶然产生或者测量不精确所致，我们就认为两者实际上是相关的，即否定原假设，而接受备择假设。

那么用什么来衡量偏差程度呢？假设理论值为E（这也是数学期望的符号哦），实际值为x，如果仅

来衡量，单个的观察值还好说，当有多个观察值x1，x2，x3的时候，很可能x1-E，x2-E，x3-E的值有正有负，因而互相抵消，使得最终的结果看上好像偏差为0，但实际上每个都有偏差，而且都还不小！此时很直接的想法便是使用方差代替均值，这样就解决了正负抵消的问题，即使用

$$\sum_{i=1}^{n}(x_i-E)^2$$

这时又引来了新的问题，对于500的均值来说，相差5其实是很小的（相差1%），而对20的均值来说，5相当于25%的差异，这是使用方差也无法体现的。因此应该考虑改进上面的式子，让均值的大小不影响我们对差异程度的判断

$$\sum_{i=1}^{n}\frac{(x_i-E)^2}{E} \qquad \text{式（1）}$$

上面这个式子已经相当好了。**实际上这个式子就是开方检验使用的差值衡量公式。**当提供了数个样本的观察值x1，x2，……xi，……xn之后，代入到式（1）中就可以求得开方值，用这个值与事先设定的阈值比较，如果大于阈值（即偏差很大），就认为原假设不成立，反之则认为原假设成立。

在文本分类问题的特征选择阶段，我们主要关心一个词t（一个随机变量）与一个类别c（另一个随机变量）之间是否相互独立？如果独立，就可以说词t对类别c完全没有表征作用，即我们根本无法根据t出现与否来判断一篇文档是否属于c这个分类。但与最普通的开方检验不同，我们不需要设定阈值，因为很难说词t和类别c关联到什么程度才算是有表征作用，我们只想借用这个方法来选出一些最最相关的即可。

此时我们仍然需要明白对特征选择来说原假设是什么，因为计算出的开方值越大，说明对原假设的偏离越大，我们越倾向于认为原假设的反面情况是正确的。我们能不能把原假设定为“词t与类别c相关“？原则上说当然可以，这也是一个健全的民主主义社会赋予每个公民的权利（笑），但此时你会发现根本不知道此时的理论值该是多少！你会把自己绕进死胡同。所以我们一般都使用”词t与类别c不相关“来做原假设。选择的过程也变成了为每个词计算它与类别c的开方值，从大到小排个序（此时开方值越大越相关），取前k个就可以（k值可以根据自己的需要选，这也是一个健全的民主主义社会赋予每个公民的权利）。

好，原理有了，该来个例子说说到底怎么算了。

比如说现在有N篇文档，其中有M篇是关于体育的，我们想考察一个词“篮球”与类别“体育”之间的相关性（任谁都看得出来两者很相关，但很遗憾，我们是智慧生物，计算机不是，它一点也看不出来，想让它认识到这一点，只能让它算算看），我们有四个观察值可以使用：

2. 包含“篮球”但不属于“体育”类别的文档数，命名为B

3. 不包含“篮球”但却属于“体育”类别的文档数，命名为C

4. 既不包含“篮球”也不属于“体育”类别的文档数，命名为D

用下面的表格更清晰：

特征选择	1．属于“体育”	2．不属于“体育”	总计
1．包含“篮球”	A	B	A+B
2．不包含“篮球”	C	D	C+D
总数	A+C	B+D	N

如果有些特点你没看出来，那我说一说，首先，A+B+C+D=N（这，这不废话嘛）。其次，A+C的意思其实就是说“属于体育类的文章数量”，因此，它就等于M，同时，B+D就等于N-M。

好，那么理论值是什么呢？以包含“篮球”且属于“体育”类别的文档数为例。如果原假设是成立的，即“篮球”和体育类文章没什么关联性，那么在所有的文章中，“篮球”这个词都应该是等概率出现，而不管文章是不是体育类的。这个概率具体是多少，我们并不知道，但他应该体现在观察结果中（就好比抛硬币的概率是二分之一，可以通过观察多次抛的结果来大致确定），因此我们可以说这个概率接近

$$\frac{A+B}{N}$$

（因为A+B是包含“篮球”的文章数，除以总文档数就是“篮球”出现的概率，当然，这里认为在一篇文章中出现即可，而不管出现了几次）而属于体育类的文章数为A+C，在这些个文档中，应该有

$$E_{11} = (A+C)\frac{A+B}{N}$$

篇包含“篮球”这个词（数量乘以概率嘛）。

但实际有多少呢？考考你（读者：切，当然是A啦，表格里写着嘛……）。

同样，我们还可以计算剩下三种情况的差值D12，D21，D22，聪明的读者一定能自己算出来（读者：切，明明是自己懒得写了……）。有了所有观察值的差值，就可以计算“篮球”与“体育”类文章的开方值

$$\chi^2(\text{篮球},\text{体育})=D_{11}+D_{12}+D_{21}+D_{22}$$

把D11，D12，D21，D22的值分别代入并化简，可以得到

$$\chi^2(\text{篮球},\ \text{体育})=\frac{N(AD-BC)^2}{(A+C)(A+B)(B+D)(C+D)}$$

词t与类别c的开方值更一般的形式可以写成

$$\chi^2(t,\ c)=\frac{N(AD-BC)^2}{(A+C)(A+B)(B+D)(C+D)}$$

实际上式（2）还可以进一步化简，注意如果给定了一个文档集合（例如我们的训练集）和一个类别，则N，M，N-M（即A+C和B+D）对同一类别文档中的所有词来说都是一样的，而我们只关心一堆词对某个类别的开方值的大小顺序，而并不关心具体的值，因此把它们从式（2）中去掉是完全可以的，故实际计算的时候我们都使用

$$\chi^2(t,\ c)=\frac{(AD-BC)^2}{(A+B)(C+D)}$$

针对英文纯文本的实验结果表明：作为特征选择方法时，开方检验和信息增益的效果最佳（相同的分类算法，使用不同的特征选择算法来得到比较结果）；文档频率方法的性能同前两者大体相当，术语强度方法性能一般；互信息方法的性能最差（文献[17]）。

但开方检验也并非就十全十美了。回头想想A和B的值是怎么得出来的，它统计文档中是否出现词t，却不管t在该文档中出现了几次，这会使得他对低频词有所偏袒（因为它夸大了低频词的作用）。甚至会出现有些情况，一个词在一类文章的每篇文档中都只出现了一次，其开方值却大过了在该类文章99%的文档中出现了10次的词，其实后面的词才是更具代表性的，但只因为它出现的文档数比前面的词少了“1”，特征选择的时候就可能筛掉后面的词而保留了前者。这就是开方检验著名的“低频词缺陷“。因此开方检验也经常同其他因素如词频综合考虑来扬长避短。

三、信息增益

因此先回忆一下信息论中有关信息量（就是“熵”）的定义。说有这么一个变量X，它可能的取值有n多种，分别是x1，x2，……，xn，每一种取到的概率分别是P1，P2，……，Pn，那么X的熵就定义为：

$$H(X)=-\sum_{i-1}^{n}P_i \bullet \log_2 P_i$$

意思就是一个变量可能的变化越多（反而跟变量具体的取值没有任何关系，只和值的种类多少以及发生概率有关），它携带的信息量就越大。

$$H(C)=-\sum_{i-1}^{n}P(C_i) \bullet \log_2 P(C_i)$$

信息增益是针对一个一个的特征而言的，就是看一个特征t，系统有它和没它的时候信息量各是多少，两者的差值就是这个特征给系统带来的信息量，即增益。系统含有特征t的时候信息量很好计算，就是刚才的式子，它表示的是包含所有特征时系统的信息量。

问题是当系统不包含t时，信息量如何计算？我们换个角度想问题，把系统要做的事情想象成这样：说教室里有很多座位，学生们每次上课进来的时候可以随便坐，因而变化是很大的（无数种可能的座次情况）；但是现在有一个座位，看黑板很清楚，听老师讲也很清楚，于是校长的小舅子的姐姐的女儿托关系（真辗转啊），把这个座位定下来了，每次只能给她坐，别人不行，此时情况怎样？对于座次的可能情况来说，我们很容易看出以下两种情况是等价的：（1）教室里没有这个座位；（2）教室里虽然有这个座位，但其他人不能坐（因为反正它也不能参与到变化中来，它是不变的）。

对应到我们的系统中，就是下面的等价：（1）系统不包含特征t；（2）系统虽然包含特征t，但是t已经固定了，不能变化。

我们计算分类系统不包含特征t的时候，就使用情况（2）来代替，就是计算当一个特征t不能变化时，系统的信息量是多少。这个信息量其实也有专门的名称，就叫做“条件熵”，条件嘛，自然就是指“t已经固定“这个条件。

但是问题接踵而至，例如一个特征X，它可能的取值有n多种（x1，x2，……，xn），当计算条件熵而需要把它固定的时候，要把它固定在哪一个值上呢？答案是每一种可能都要固定一下，计算n个值，然后取均值才是条件熵。而取均值也不是简单的加一加然后除以n，而是要用每个值出现的概率来算平均（简单理解，就是一个值出现的可能性比较大，固定在它上面时算出来的信息量占的

$$H(C|X=x_i)$$

这是指特征X被固定为值xi时的条件熵。

$$H(C|X)$$

这是指特征X被固定时的条件熵，注意与上式在意义上的区别。从刚才计算均值的讨论可以看出来，第二个式子与第一个式子的关系就是：

$$H(C|X)=P_1H(C|X=x_1)+P_2H(C|X=x_2)+...+P_nH(C|X=x_n)$$

$$=\sum_{i=1}^{n}P_iH(C|X=x_i)$$

具体到我们文本分类系统中的特征t，t有几个可能的值呢？注意t是指一个固定的特征，比如他就是指关键词“经济”或者“体育”，当我们说特征“经济”可能的取值时，实际上只有两个，“经济”要么出现，要么不出现。一般的，t的取值只有t（代表t出现）和t_cat（代表t不出现），注意系统包含t但t 不出现与系统根本不包含t可是两回事。

因此固定t时系统的条件熵就有了，为了区别t出现时的符号与特征t本身的符号，我们用T代表特征，而用t代表T出现，那么

$$H(C|T)=P(t)H(C|t)+P(\bar{t})H(C|\bar{t})$$

因此特征T给系统带来的信息增益就可以写成系统原本的熵与固定特征T后的条件熵之差：

$$IG(T)=H(C)-H(C|T)$$

$$=-\sum_{i=1}^{n}P(C_i)\log_2P(C_i)+$$

$$P(t)\sum_{i=1}^{n}P(C_i|t)\log_2P(C_i|t)+P(\bar{t})\sum_{i=1}^{n}P(C_i|\bar{t})\log_2P(C_i|\bar{t})$$

小就要把大小的影响加进去）。再比如P(t)，就是特征T出现的概率，只要用出现过T的文档数除以总文档数就可以了，再比如P(Ci|t)表示出现T的时候，类别Ci出现的概率，只要用出现了T并且属于类别Ci的文档数除以出现了T的文档数就可以了。

从以上讨论中可以看出，信息增益也是考虑了特征出现和不出现两种情况，与开方检验一样，是比较全面的，因而效果不错。但信息增益最大的问题还在于它只能考察特征对整个系统的贡献，而不能具体到某个类别上，这就使得它只适合用来做所谓“全局”的特征选择（指所有的类都使用相同的特征集合），而无法做“本地”的特征选择（每个类别有自己的特征集合，因为有的词，对这个类别很有区分度，对另一个类别则无足轻重）。

四、互信息

一个常用的方法是计算文档中的词项t与文档类别c的互信息MI，MI度量的是词的存在与否给类别c带来的信息量，互信息的基本定义如下：

$$I(X;Y)=\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$$

应用到文本特征选择:

$$I(U;C) = \sum_{e_t\in\{1,0\}}\sum_{e_c\in\{1,0\}}P(U=e_t,C=e_c)\log_2\frac{P(U=e_t,C=e_c)}{P(U=e_t)P(C=e_c)},$$

U、C都是二值随机变量，当文档包含词项t时，U的取值为et=1，否则et=0；当文档属于类别c时，C的取值ec=1，否则ec=0，用最大似然估计时，上面的概率值都是通过统计文档中词项和类别的数目阿里计算的。于是实际计算公式如下：

$$\begin{aligned}I(U;C) = {} & \frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_{1.}N_{.1}}+\frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_{0.}N_{.1}}\\ & +\frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_{1.}N_{.0}}+\frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_{0.}N_{.0}}\end{aligned}$$

我们可以对每一个类计算各个词项与其的互信息，并选取值最大的k个词项，当然有可能两个类会选取相同的特征词，去重一下即可。

知乎 首发于 机器学习和自然语言处理

五、N-Gram

基于N-Gram的方法是把文章序列，通过大小为N的窗口，形成一个个Group，然后对这些Group做统计，滤除出现频次较低的Group，把这些Group组成特征空间，传入分类器，进行分类。

reference

TF-IDF与余弦相似性的应用（一）：自动提取关键词 - 阮一峰的网络日志

文本分类入门（十）特征选择算法之开方检验 - Jasper's Java Jacal - BlogJava

文本分类入门（十一）特征选择方法之信息增益 - Jasper's Java Jacal - BlogJava

文本特征选择 - CodeMeals - 博客园

编辑于 2017-08-03

机器学习 自然语言处理 特征选择

文章被以下专栏收录

机器学习和自然语言处理
整理和输出知识的地方。

关注专栏

推荐阅读