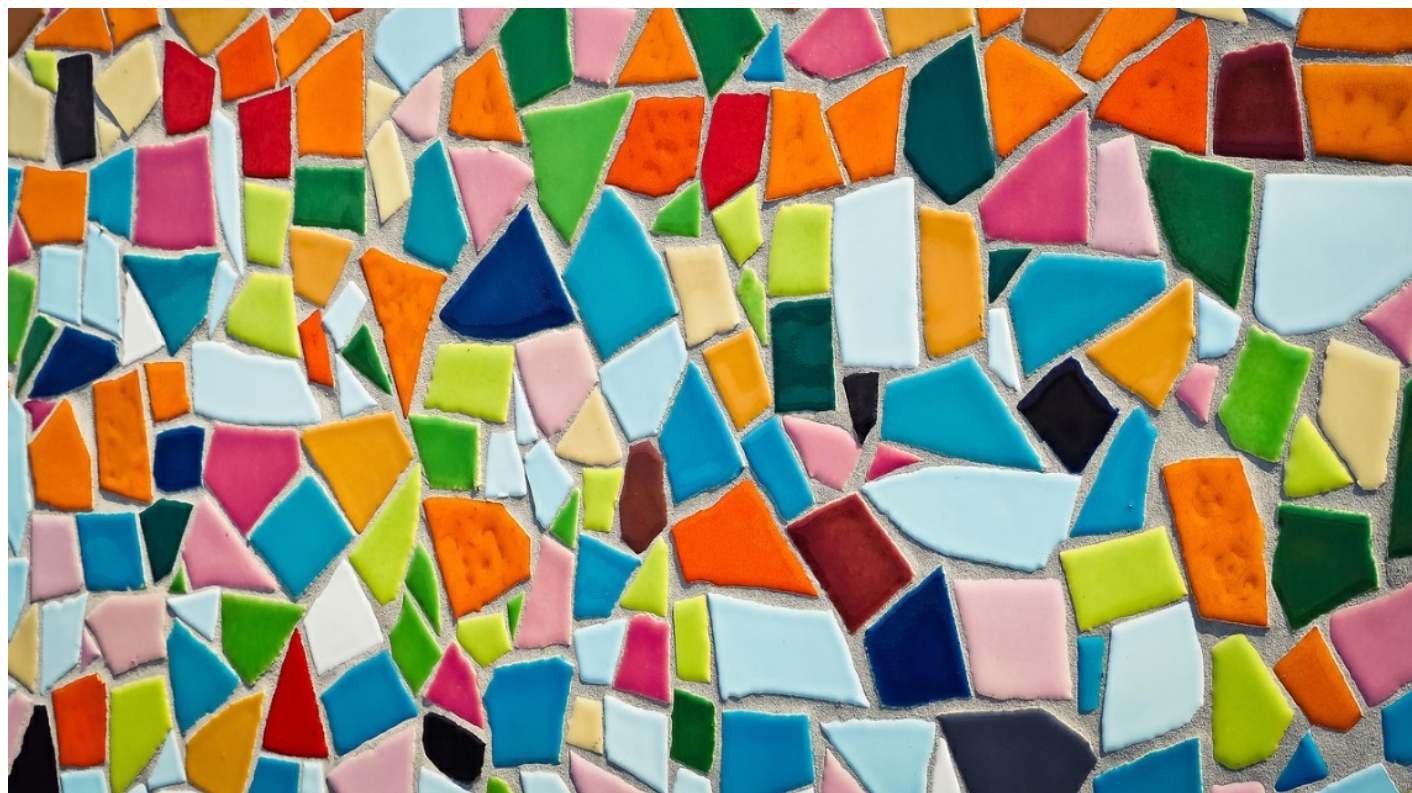


24讲语言模型：如何使用链式法则和马尔科夫假设简化概率模型



你好，我是黄申。

之前我给你介绍了用于分类的朴素贝叶斯算法。我们讲了，朴素贝叶斯算法可以利用贝叶斯定理和变量之间的独立性，预测一篇文章属于某个分类的概率。除了朴素贝叶斯分类，概率的知识还广泛地运用在其他机器学习算法中，例如语言模型、马尔科夫模型、决策树等等。

今天我就来说说，基于概率和统计的语言模型。语言模型在不同的领域、不同的学派都有不同的定义和实现，因此为了避免歧义，我这里先说明一下，我们谈到的语言模型，都是指基于概率和统计的模型。

语言模型是什么？

在解释语言模型之前，我们先来看两个重要的概念。第一个是链式法则，第二个是马尔科夫假设及其对应的多元文法模型。为什么要先说这两个概念呢？这是因为链式法则可以把联合概率转化为条件概率，而马尔科夫假设通过变量间的独立性来减少条件概率中的随机变量，两者结合就可以大幅简化计算的复杂度。

1. 链式法则

链式法则是概率论中一个常用法则。它使用一系列条件概率和边缘概率，来推导联合概率，我用一个公式来给你看看它的具体表现形式。

$$P(x_1, x_2, \dots, x_n) = P(x_1) \times P(x_2 | x_1) \times P(x_3 | x_1, x_2) \times \dots \times P(x_n | x_1, x_2, \dots, x_{n-1})$$

其中， x_1 到 x_n 表示了n个随机变量。

这个公式是怎么来的呢？你还记得联合概率、条件概率和边缘概率之间的“三角”关系吗？我们用这三者的关系来推导一下，最终我们可以得到链式法则。

$$\begin{aligned}
& P(x_1, x_2, \dots, x_n) \\
&= P(x_1, x_2, \dots, x_{n-1}) \times P(x_n | x_1, x_2, \dots, x_{n-1}) \\
&= P(x_1, x_2, \dots, x_{n-2}) \times P(x_{n-1} | x_1, x_2, \dots, x_{n-2}) \times P(x_n | x_1, x_2, \dots, x_{n-1}) \\
&= \dots \\
&= P(x_1) \times P(x_2 | x_1) \times P(x_3 | x_1, x_2) \times \dots \times P(x_n | x_1, x_2, \dots, x_{n-1})
\end{aligned}$$

推导的每一步，都是使用了三种概率之间的关系，这个应该不难理解。

2.马尔科夫假设

理解了链式法则，我们再来看看马尔可夫假设。这个假设的内容是：任何一个词 w_i 出现的概率只和它前面的1个或若干个词有关。基于这个假设，我们可以提出**多元文法（Ngram）模型**。Ngram中的“N”很重要，它表示任何一个词出现的概率，只和它前面的N-1个词有关。

我以二元文法模型为例，来给你解释。按照刚才的说法，二元文法表示，某个单词出现的概率只和它前面的1个单词有关。也就是说，即使某个单词出现在一个很长的句子中，我们也只需要看前面那1个单词。用公式来表示出来就是这样：

$$P(w_n | w_1 w_2, \dots, w_{n-1}) \approx P(w_n | w_{n-1})$$

如果是三元文法，就说明某个单词出现的概率只和它前面的2个单词有关。即使某个单词出现在很长的一个句子中，它也只看相邻的前2个单词。用公式来表达就是这样：

$$P(w_n | w_1 w_2, \dots, w_{n-1}) \approx P(w_n | w_{n-1}, w_{n-2})$$

你也许会好奇，那么一元文法呢？按照字面的意思，就是每个单词出现的概率和前面0个单词有关。这其实说明，每个词的出现都是相互独立的。用公式来表达就是这样的：

$$P(w_1, w_2, \dots, w_n) = P(w_1) \times P(w_2) \times P(w_3) \times \dots \times P(w_n)$$

弄明白链式法则和马尔科夫假设之后，我们现在来看语言模型。

假设我们有一个统计样本文本 d ， s 表示某个有意义的句子，由一连串按照特定顺序排列的词 $w_{\{1\}}, w_{\{2\}}, \dots, w_{\{n\}}$ 组成，这里 n 是句子里单词的数量。现在，我们想知道根据文档 d 的统计数据， s 在文本中出现的可能性，即 $P(s|d)$ ，那么我们可以把它表示为 $P(s|d) = P(w_{\{1\}}, w_{\{2\}}, \dots, w_{\{n\}} | d)$ 。假设我们这里考虑的都是集合 d 的情况下发生的概率，所以可以忽略 d ，写为 $P(s) = P(w_{\{1\}}, w_{\{2\}}, \dots, w_{\{n\}})$ 。

到这里，我们碰到了第一个难题，就是如何计算 $P(w_{\{1\}}, w_{\{2\}}, \dots, w_{\{n\}})$ 要在集合中找到一模一样的句子，基本是不可能的。这个时候，我们就需要使用链式法则。我们可以把这个式子改写为：

$$P(w_1, w_2, \dots, w_n) = P(w_1) \times P(w_2 | w_1) \times P(w_3 | w_1, w_2) \times P(w_4 | w_1, w_2, w_3) \times \dots \times P(w_n | w_1, w_2, \dots, w_{n-1})$$

乍一看，问题似乎是解决了。因为通过文档集合 C ，你可以知道 $P(w_{\{1\}})$ ， $P(w_{\{2\}} | w_{\{1\}})$ 这种概率。不过，再往后看，好像 $P(w_{\{3\}} | w_{\{1\}}, w_{\{2\}})$ 出现概率很低， $P(w_{\{4\}} | w_{\{1\}}, w_{\{2\}}, w_{\{3\}})$ 出现的概率就更低了。一直到 $P(w_{\{n\}} | w_{\{1\}}, w_{\{2\}}, \dots, w_{\{n-1\}})$ ，基本上又为0了。我们可以使用上一节提到的平滑技巧，减少0概率的出现。不过，如果太多的概率都是通过平滑的方式而得到的，那么模型和真实的数据分布之间的差距就会加大，最终预测的效果也会很差，所以平滑也不是解决0概率的最终办法。

除此之外， $P(w_{\{1\}}, w_{\{2\}}, \dots, w_{\{n\}})$ 和 $P(w_{\{n\}} | w_{\{1\}}, w_{\{2\}}, \dots, w_{\{n-1\}})$ 还不只会导致0概率，它还会使得模型存储空间

的急速增加。

为了统计现有文档集合中 $P(w_{\{1\}}, w_{\{2\}}, \dots, w_{\{n\}})$ 这类值，我们就需要生成很多的计数器。我们假设文档集合中有 m 个不同的单词，那么从中挑出 n 个单词的可重复排列，数量就是 m^n 。此外，还有 m^{n-1} ， m^{n-2} 等等。这也意味着，如果要统计并存储的所有 $P(w_{\{1\}}, w_{\{2\}}, \dots, w_{\{n\}})$ 或 $P(w_{\{n\}}|w_{\{1\}}, w_{\{2\}}, \dots, w_{\{n-1\}})$ 这类概率，就需要大量的内存和磁盘空间。当然，你可以做一些简化，不考虑单词出现的顺序，那么问题就变成了可重复组合，但是数量仍然非常巨大。

如何解决0概率和高复杂度的问题呢？马尔科夫假设和多元文法模型能帮上大忙了。如果我们使用三元文法模型，上述公式可以改写为：

$$P(w_1, w_2, \dots, w_n) \approx P(w_1) \times P(w_2 | w_1) P(w_3 | w_2, w_1) P(w_4 | w_3, w_2) \times \dots \times P(w_n | w_{n-1}, w_{n-2})$$

这样，系统的复杂度大致在 $(C(m, 1) + C(m, 2) + C(m, 3))$ 这个数量级，而且 $P(w_{\{n\}}|w_{\{n-2\}}, w_{\{n-1\}})$ 为0的概率也会大大低于 $P(w_{\{n\}}|w_{\{1\}}, w_{\{2\}}, \dots, w_{\{n-1\}})$ （其中 $n \gg 3$ ）为0的概率。当然，多元文法模型中的 N 还是不能太大。随着 N 的增大，系统复杂度仍然会快速升高，就无法体现出多元文法的优势了。

语言模型的应用

基于概率的语言模型，本身不是新兴的技术。它已经在机器翻译、语音识别和中文分词中得到了成功应用。近几年来，人们也开始在信息检索领域中尝试语言模型。下面我就来讲讲语言模型在信息检索和中文分词这两个方面里是如何发挥作用的。

1.信息检索

信息检索很关心的一个问题就是相关性，也就是说，给定一个查询，哪篇文档是更相关的呢？为了解决相关性问题，布尔模型和向量空间检索模型都是从查询的角度出发，观察查询和文档之间的相似程度，并以此来决定如何找出相关的文档。这里的“相似程度”，你可以理解为两者长得有多像。那么，语言模型如何来刻画查询和文档之间的相关度呢？

它不再使用相似度定义，而是采用了概率。一种常见的做法是计算 $P(d|q)$ ，其中 q 表示一个查询， d 表示一篇文档。 $P(d|q)$ 表示用户输入查询 q 的情况下，文档 d 出现的概率是多少？如果这个概率越高，我们就认为 q 和 d 之间的相关性越高。

通过我们手头的文档集合，并不能直接获得 $P(d|q)$ 。好在我们已经学习过了贝叶斯定理，通过这个定理，我们可以将 $P(d|q)$ 重写如下：

$$P(d|q) = \frac{P(q|d) \times P(d)}{P(q)}$$

对于同一个查询，其出现概率 $P(q)$ 都是相同的，同一个文档 d 的出现概率 $P(d)$ 也是固定的。因此它们可以忽略，我们只要关注如何计算 $P(q|d)$ 。而语言模型，为我们解决了如何计算 $P(q|d)$ 的问题，让 $k_{\{1\}}, k_{\{2\}}, \dots, k_{\{n\}}$ 表示查询 q 里包含的 n 个关键词。那么根据之前的链式法则公式，可以重写为这样：

$$P(q|d) = P(k_1, k_2, k_3, \dots, k_n | d) = P(k_1 | d) \times P(k_2 | k_1, d) \times P(k_3 | k_1, k_2, d) \times \dots \times P(k_n | k_1, k_2, \dots, k_{n-1}, d)$$

为了提升效率，我们也使用马尔科夫假设和多元文法。假设是三元文法，那么我们可以写成这样：

$$P(q|d) = P(k_1, k_2, k_3, \dots, k_n | d) = P(k_1 | d) \times P(k_2 | k_1, d) \times P(k_3 | k_2, k_1, d) \times \dots \times P(k_n | k_{n-1}, k_{n-2}, d)$$

最终，当用户输入一个查询 q 之后，对于每一篇文档 d ，我们都能获得 $P(d|q)$ 的值。根据每篇文档所获得的 $P(d|q)$ 这个值，由高到低对所有的文档进行排序。这就是语言模型在信息检索中的常见用法。

2.中文分词

和拉丁语系不同，中文存在分词的问题。如果想进行分词，你就可以实用语言模型。我举个例子给你解释一下，你就明白了。

最普遍的分词方法之一是基于常用词的词典。如果一个尚未分词的句子发现了存在于字典里的词，我们就认为找到一个新的词，并把它切分出来。这种切分不会出现完全离谱的结果，但是无法解决某些歧义。我下面来举个例子，原句是“乒乓球拍卖完了”。我在读的时候，会有所停顿，你就能理解分词应该如何进行。可是，仅仅从书面来看，至少有以下几种分词方式：

第一种，乒乓|球|拍卖|完了

第二种，乒乓球|拍卖|完了

第三种，乒乓|球拍|卖完|了

第四种，乒乓|球拍|卖|完了

上面分词的例子，从字面来看都是合理的，所以这种歧义无法通过这句话本身来解决。那么这种情况下，语言模型能为我们做什么呢？我们知道，语言模型是基于大量的语料来统计的，所以我们可以使用这个模型来估算，哪种情况更合理。

假设整个文档集合是D，要分词的句子是s，分词结果为w₁, ... w_n，那么我们可以求P(s)的概率为：

$$P(s|D) = P(w_1, w_2, w_3, ..., w_n | D) = P(w_1 | D) \times P(w_2 | w_1, D) \times P(w_3 | w_1, w_2, D) \times ... \times P(w_n | w_1, w_2, ..., w_{n-1}, D)$$

请注意，在信息检索中，我们关心的是每篇文章产生一个句子（也就是查询）的概率，而这里可以是整个文档集合D产生一个句子的概率。

根据链式法则和三元文法模型，那么上面的式子可以重写为：

$$P(s|D) = P(w_1, w_2, w_3, ..., w_n | D) = P(w_1 | D) \times P(w_2 | w_1, D) \times P(w_3 | w_2, w_1, D) \times ... \times P(w_n | w_{n-1}, w_{n-2}, D)$$

也就是说，语言模型可以帮我们估计某种分词结果，在文档集中出现的概率。但是由于不同的分词方法，会导致w₁到w_n的不同，因此就会产生不同的P(s)。接下来，我们只要取最大的P(s)，并假设这种分词方式是最合理的，就可以在一定程度上解决歧义。我们可以使用这个公式来求解：

$$\arg \max P(W_i | D)$$

其中，W_i表示第i种分词方法。

回到“乒乓球拍卖完了”这句话，如果文档集合都是讲述的有关体育用品的销售，而不是拍卖行，那么“乒乓|球拍|卖完|了”这种分词的可能性应该更高。

小结

这一节，我介绍了基于概率论的语言模型，以及它在信息检索和中文分词领域中的应用。这一节的公式比较多，你刚开始看可能觉得有点犯晕。不用急，我给你梳理了几个要点，你只要掌握这几个要点，依次再进行细节学习，就会事半功倍。

第一，使用联合概率，条件概率和边缘概率的“三角”关系，进行相互推导。链式法则就是很好的体现。

第二，使用马尔科夫假设，把受较多随机变量影响的条件概率，简化为受较少随机变量影响的条件概率，甚至是边缘概率。

第三，使用贝叶斯定理，通过先验概率推导后验概率。在信息检索中，给定查询的情况下推导文档的概率，就需要用到这个定理。

如果你记住了这几点，那么不仅能很快的理解本篇的内容，还能根据实际需求，设计出满足自己需要的语言模型。

思考题

在中文分词的时候，我们也可以考虑文章的分类。比如，这样一句话“使用纯净水源浇灌的大米”，正确的切分应该是：

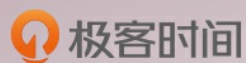
使用 | 纯净 | 水源 | 浇灌 | 的 | 大米

如果我们知道这句描述来自于“大米”类商品，而不是“纯净水”类商品，那么就不会错误地切分为：

使用 | 纯净水 | 源 | 浇灌 | 的 | 大米

想想看，如何对我介绍的语言模型加以改进，把分类信息也包含进去？

欢迎留言和我分享，也欢迎你在留言区写下今天的学习笔记。你可以点击“请朋友读”，把今天的内容分享给你的好友，和他一起精进。



程序员的数学基础课

在实战中重新理解数学

黄申

LinkedIn 资深数据科学家



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

精选留言



acheng

换句话说：其实我是想问，如何能更好的利用全文或者说全部训练集的语义信息？

2019-02-16 07:39

作者回复

如果是词包模型，确实对语义没有太多的理解。可以加入一些基于文法甚至是领域知识的语义分析，不过这个和具体的应用有关系，可能不是语法模型本身能很好解决的。例如，评论中的情感分析（sentiment analysis），我们可以考虑表达情感的词在否定句式中的表达等等。

2019-02-18 00:57



acheng

这篇文章和上篇文章中介绍的分类型基本都是在假定文章中的词语相互独立的情况下进行的，虽然马尔科夫模型用到前面词的概率，这可以说是结合了部分上下文之间的语义关系，只使用了和前面词的语义关系，对文本分类其实效果已经很好，尤其是在语料充足的情况下可以使用机器学习的方法让分类模型自迭代优化，目前很多机器翻译就是这么干的。但是我想问下老师，针对语料不是很充足的文本集，如何使用全篇文章的语义（不只是前面词的上下文关系）来设计一个文本分类器？

2019-02-15 07:56

作者回复

我想你说的是针对分类问题对吧？那么语料不充足，是指只有少数的标注数据（文章），是吧？如果是这样，一种做法是增大ngram里面的n，因为标注数据不多，增加n不会增加太多的存储空间。另外，也可以使用少量数据训练得到的分类器，对新的数据进行分类，然后获得一些分类结果后，人工再进行复查，把正确的结果再次纳入训练数据。

2019-02-16 01:11



?

文本分类器，对给定文本进行判断。用特征词代表该文本。应该和上篇文章分类的计算有类似之处。计算每个特征词出现在该类文章的概率。然后根据权重分类？或者根据每个词的词频。

（我也很迷糊）那中文中有时词的顺序错乱也能表达一个意思。

比如，密码是123和321是密码；蹦迪坟头和坟头蹦迪。

比如（相互和互相；代替和替代）比

如，纳税可以是一个专有名词，也可以是人名，姓纳名税。还有那遇到多音字咋办？

2019-02-08 11:09

作者回复

这个问题很好，确实中文比较特殊，和拉丁文不太一样。

我觉得你的问题是：中文里的歧义或者分词错误，是不是会影响分类？

你说的这几种情况，我简单分为以下几种：

分词：如果我们能知道123或321代表一个字符串，而不是单个的数字，那么就不会切分它们。再例如“相互”也不会切为“相”和“互”。当然中文分词本身不是件容易的事情，我这里提到概率语言模型，如果语料里有相关的信息，那么可以在一定程度上提升分词效果。

同义词：如果我们能正确切分出“相互”和“互相”，那么还需要把它们关联为同义词。基本的做法是使用词典。

语义：“纳税”的问题就更复杂一些，需要计算机理解上下文关系和语义。从统计的角度而言，那还是要看语料里“纳税”这个词哪种情况的概率更高。

所以，自然语言处理，尤其是中文的处理，是件相当复杂也是相当有趣的事情。“词包”模型只是最基本的模型，如果我们想优化它在分类问题上的表现，需要解决好中文分词、消除歧义、同义词/近义词等问题。每个问题都是值得研究，并且可以提升的。如果每个点都能得到优化，那么最终分类的效果也会得到优化。

总结一下，文本分类涉及的面很广，不仅受到分类算法的影响，还受到其他许多自然语言处理的影响。由于这个系列专栏的主题是数学，所以我这讲只能把概率和相关分类算的核心思想体现出来。如果你对自然语言处理有兴趣，我可以在加餐或者其他专栏中和你分享。

2019-02-09 01:38