WIKIPEDIA

向量空间模型

维基百科，自由的百科全书

向量空间模型是一个把文本文件表示为标识符（比如索引）向量的代数模型。它应用于信息过滤、信息检索、索引以及相关排序。SMART是第一个使用这个模型的信息检索系统。

目录

定义

文档和查询都用向量来表示。

$$d_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$
$$q = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$$

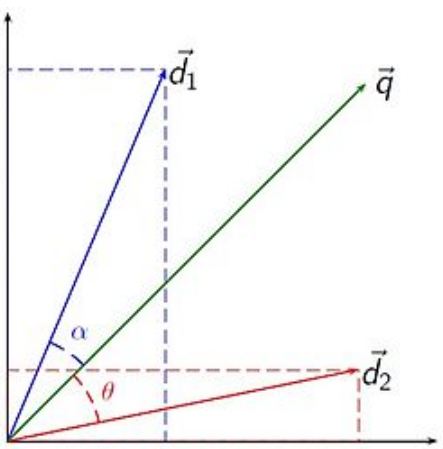
每一维都对应于一个个别的词组。如果某个词组出现在了文档中，那它在向量中的值就非零。已经发展出了不少的方法来计算这些值，这些值叫做（词组）权重。其中一种最为知名的方式是tf-idf权重（见下面的例子）。

词组的定义按不同应用而定。典型的词组就是一个单一的词、关键词、或者较长的短语。如果将词语选为词组，那么向量的维数就是词汇表中的词语个数（出现在语料库中的不同词语的个数）。

通过向量运算，可以对各文档和各查询作比较。

应用

据文档相似度理论的假设，如要在一次关键词查询中计算各文档间的相关排序，只需比较每个文档向量和原先查询向量（跟文档向量的类型是相同的）之间的角度偏差。



实际上，计算向量之间夹角的余弦比直接计算夹角本身要简单。

$$\cos\theta = \frac{\mathbf{d_2} \cdot \mathbf{q}}{\|\mathbf{d_2}\| \, \|\mathbf{q}\|}$$

其中$\mathbf{d_2} \cdot \mathbf{q}$是文档向量（即右图中的$d_2$）和查询向量（图中的q）的点乘。$\|\mathbf{d_2}\|$是向量$d_2$的模，而$\|\mathbf{q}\|$是向量q的模。向量的模通过下面的公式来计算：

$$\|\mathbf{v}\| = \sqrt{\sum_{i=1}^{n} v_i^2}$$

由于这个模型所考虑的所有向量都是每个元素严格非负的，因此如果余弦值为零，则表示查询向量和文档向量是正交的，即不符合（换句话说，就是检索项在文档中没有找到）。如果要了解详细的信息可以查看余弦相似性这条目。

范例：tf–idf权重

在Salton，Wong和Yang [1]提出的传统向量空间模型中，一个词组在文档向量中的权重就是局部参数和全局参数的乘积，这就是著名的tf-idf模型（词频-逆向文档频率）。文档的权重向量d就是$\mathbf{v_d} = [w_{1,d}, w_{2,d}, \ldots, w_{N,d}]^T$，其中

$$w_{t,d} = \mathrm{tf}_{t,d} \cdot \log \frac{|D|}{|\{d' \in D \,|\, t \in d'\}|}$$

- $\mathrm{tf}_{t,d}$是词组t在文档d中出现的频率（一个局部参数）
- $\log \frac{|D|}{|\{d' \in D \,|\, t \in d'\}|}$是逆向文档频率（一个全局参数）。$|D|$是文档集中的文档总数；$|\{d' \in D \,|\, t \in d'\}|$是含有词组$t$的文档数。

文档d_j和查询q之间的余弦相似度通过以下公式来计算：

$$\mathrm{sim}(d_j, q) = \frac{\mathbf{d_j} \cdot \mathbf{q}}{\|\mathbf{d_j}\| \, \|\mathbf{q}\|} = \frac{\sum_{i=1}^{N} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}}$$

在较简单的词组计数模型（Term Count Model）中，词组的权重不包含全局参数，而是单纯的计算词组出现的次数：$w_{t,d} = \mathrm{tf}_{t,d}$。

优点

相对于标准布尔模型（Standard Boolean model），向量空间模型具有如下优点：

1. 基于线性代数的简单模型
2. 词组的权重不是二元的
3. 文档和查询之间的相似度取值是连续的
4. 允许根据文档间可能的相关性来进行排序
5. 允许局部匹配

局限

向量空间模型有如下局限：

1. 不适用于较长的文档，因为它的相似值不理想（过小的内积和过高的维数）。
2. 检索词组必须与文档中出现的词组精确匹配；词语子字串可能会导致“假阳性”匹配。
3. 语义敏感度不佳；具有相同的语境但使用不同的词组的文档不能被关联起来，导致“假阴性匹配”。
4. 词组在文档中出现的顺序在向量形式中无法表示出来。
5. 假定词组在统计上是独立的。
6. 权重是直观上获得的而不够正式。

然而，这些局限中的多数能够通过集合各种方法来解决，包括数学上的技术（比如奇异值分解）和词汇数据库（比如WordNet）。

基于及扩展了向量空间模型的模型

基于及扩展了向量空间模型的模型包括：

- 广义向量空间模型
- （增强的）基于主题的向量空间模型
- 潜在语义学

- 潜在语义索引
- DSIR模型
- 词汇鉴别（Term Discrimination）
- Rocchio分类

以向量空间模型为工具的软件

使用向量空间模型做实验或者想基于它们实现研究服务的人或许会对以下的这些软件包感兴趣。

免费开放的软件资源

- Apache Lucene.这是一个高性能的软件，用java写的功能全面的文本搜索引擎。
- SemanticVectors (https://web.archive.org/web/20080828220200/http://semanticvectors.googlecode.com/).语义向量索引，将随机投影算法（类似于潜在的语义分析）应用于Apache Lucene构建的文本词组矩阵。
- Gensim (http://nlp.fi.muni.cz/projekty/gensim)是一个Python+NumPy的向量空间模型的框架。它包含对Tf–idf、潜在的语义索引、随机投影和潜在的狄利克雷边界的增值算法（有效利用内存空间）。
- Antonio Gulli开发的Compressed vector space in C++ (http://codingplayground.blogspot.com/2010/03/compressed-vector-space.html)
- Text to Matrix Generator (TMG) (http://scgroup.hpclab.ceid.upatras.gr/scgroup/Projects/TMG/)用于一系列特殊文本挖掘的matlab工具箱。（1）指标化（2）检索（3）降维（4）聚类（5）分类。大多数的TMG都是用matlab编写的，小部分是用Perl编写的。它包括了LSI的实现和聚类、NMF以及其他方法。
- SenseClusters (http://senseclusters.sourceforge.net)，通过潜在的语义分析和单词的同现矩阵来进行文本和词组聚类的一个公开软件包。
- S-Space Package (http://code.google.com/p/airhead-research/)，通过“统计语义”实现的的检索程序集成。

进一步参考

- G. Salton, A. Wong, and C. S. Yang (1975), "A Vector Space Model for Automatic Indexing (http://www.cs.uiuc.edu/class/fa05/cs511/Spring05/other_papers/p613-salton.pdf)," *Communications of the ACM*, vol. 18, nr. 11, pages 613–620. *(Article in which a vector space model was presented)*
- David Dubin (2004), The Most Influential Paper Gerard Salton Never Wrote (http://www.ideals.uiuc.edu/bitstream/2142/11697/2/Dubin748764.pdf) *(Explains the history of the Vector Space Model and the non-existence of a frequently cited publication)*
- Description of the vector space model (http://isp.imm.dtu.dk/thor/projects/multimedia/textmining/node5.html)
- Description of the classic vector space model by Dr E Garcia (http://www.miislita.com/term-vector/term-vector-3.html)

另见

- 反向索引
- 复合词组处理（Compound term processing）

参考文献

1. G. Salton , A. Wong , C. S. Yang, A vector space model for automatic indexing (http://doi.acm.org/10.1145/361219.361220), Communications of the ACM, v.18 n.11, p.613-620, Nov. 1975