简介

LSA和传统向量空间模型(vector space model)一样使用向量来表示词(terms)和文档(documents)，并通过向量间的关系(如夹角)来判断词及文档间的关系；不同的是，LSA 将词和文档映射到潜在语义空间，从而去除了原始向量空间中的一些“噪音”，提高了信息检索的精确度。

LSA原理

通过对大量的文本集进行统计分析，从中提取出词语的上下文使用含义。技术上通过SVD分解等处理，消除了同义词、多义词的影响，提高了后续处理的精度。  
流程：  
（1）分析文档集合，建立词汇-文本矩阵A。  
（2）对词汇-文本矩阵进行奇异值分解。  
（3）对SVD分解后的矩阵进行降维  
（4）使用降维后的矩阵构建潜在语义空间

SVD分解，即奇异值分解，听起来很高大上，其实就是将一个矩阵用其他几个矩阵的乘积来表示。假设有 m×n 的矩阵 A，那么 SVD 就是要找到如下式的这么一个分解，将 A分解为 3 个矩阵的乘积：  
A m x n = Um x k X ∑ k x k X Vk x nT  
性质：对于奇异值,它跟我们特征分解中的特征值类似，在奇异值矩阵中也是按照从大到小排列，而且奇异值的减少特别的快，在很多情况下，前10%甚至1%的奇异值的和就占了全部的奇异值之和的99%以上的比例。由于这个重要的性质，SVD可以用于PCA降维，来做数据压缩和去噪。也可以用于推荐算法，将用户和喜好对应的矩阵做特征分解，进而得到隐含的用户需求来做推荐。同时也可以用于NLP中的算法，比如潜在语义索引（LSI）。

（1）词汇-文本矩阵A是一个稀疏矩阵，其行代表词语，其列代表文档。一般情况下，词-文档矩阵的元素是该词在文档中的出现次数，也可以是是该词语的tf-idf(term frequency–inverse document frequency)。

（2）小矩阵X是对词进行分类的一个结果，它的每一行表示一个词，每一列表示一个语义相近的词类，这一行中每个非零元素表示每个词在每个语义类中的重要性（或者说相关性）

如: X =  
[0.7 0.15  
0.22 0.49  
0.3 0.03]

则第一个词和第一个语义类比较相关，第二个词正好相反，第三个词与两个语义都相关性都不强。

第二个小矩阵B表示词的类和文章的类之间的相关性

如B =  
[0.7 0.21  
0.18 0.63]

则第一个词的语义类和第一个主题相关，和第二个主题没有太多关系，第二个词的语义类则相反

矩阵Y是对文本进行分类的一个结果，它的每一行表示一个主题，每一列表示一个文本，这一列每个元素表示这篇文本在不同主题中的相关性  
如Y =  
[0.7 0.15 0.22  
0 0.92 0.08]

则第一篇文章属于第一个主题，第二篇文章和第二个主题非常相关，第三篇文章与两个主题都不相关

（3）在构建好词-文档矩阵之后，LSA将对该矩阵进行降维，来找到词-文档矩阵的一个低阶近似。降维的原因有以下几点：

a. 原始的词-文档矩阵太大导致计算机无法处理，从此角度来看，降维后的新矩阵式原有矩阵的一个近似。  
b. 原始的词-文档矩阵中有噪音，从此角度来看，降维后的新矩阵式原矩阵的一个去噪矩阵。  
c. 原始的词-文档矩阵过于稀疏。原始的词-文档矩阵精确的反映了每个词是否“出现”于某篇文档的情况，然而我们往往对某篇文档“相关”的所有词更感兴趣，因此我们需要发掘一个词的各种同义词的情况。

降维的结果是不同的词或因为其语义的相关性导致合并，如：  
{(car), (truck), (flower)} --> {(1.3452 \* car + 0.2828 \* truck), (flower)}  
将维可以解决一部分同义词的问题，也能解决一部分二义性问题。具体来说，原始词-文档矩阵经过降维处理后，原有词向量对应的二义部分会加到和其语义相似的词上，而剩余部分则减少对应的二义分量。

### 应用

可对文档进行比较，进而可用于文档聚类和文档分类。

在翻译好的文档上进行训练，可以发现不同语言的相似文档，可用于跨语言检索。

发现词与词之间的关系，可用于同义词、歧义词检测。.

通过查询映射到语义空间，可进行信息检索。

从语义的角度发现词语的相关性，可用于“选择题回答模型”（multi choice qustions answering model）。

### LSA的优点

1）低维空间表示可以刻画同义词，同义词会对应着相同或相似的主题。  
2）降维可去除部分噪声，是特征更鲁棒。  
3）充分利用冗余数据。  
4）无监督/完全自动化。  
5）与语言无关。

### LSA的缺点

1）LSA可以处理向量空间模型无法解决的一义多词(synonymy)问题，但不能解决一词多义(polysemy)问题。因为LSA将每一个词映射为潜在语义空间中的一个点，也就是说一个词的多个意思在空间中对于的是同一个点，并没有被区分。  
2）SVD的优化目标基于L-2 norm 或者 Frobenius Norm 的，这相当于隐含了对数据的高斯分布假设。而 term 出现的次数是非负的，这明显不符合 Gaussian 假设，而更接近 Multi-nomial 分布。  
3）特征向量的方向没有对应的物理解释。  
4）SVD的计算复杂度很高，而且当有新的文档来到时，若要更新模型需重新训练。  
5）没有刻画term出现次数的概率模型。  
6）对于count vectors 而言，欧式距离表达是不合适的（重建时会产生负数）。  
7）维数的选择是ad-hoc的。  
8）LSA具有词袋模型的缺点，即在一篇文章，或者一个句子中忽略词语的先后顺序。  
9）LSA的概率模型假设文档和词的分布是服从联合正态分布的，但从观测数据来看是服从泊松分布的。因此LSA算法的一个改进PLSA使用了多项分布，其效果要好于LSA。