潜在语义索引(LSI)

　　　　潜在语义索引(Latent Semantic Indexing,以下简称LSI)，有的文章也叫Latent Semantic  Analysis（LSA）。其实是一个东西，后面我们统称LSI，它是一种简单实用的主题模型。LSI是基于奇异值分解（SVD）的方法来得到文本的主题的。SVD：对于一个m×nm×n的矩阵AA，可以分解为下面三个矩阵：

Am×n=Um×mΣm×nVTn×nAm×n=Um×mΣm×nVn×nT

　　　　有时为了降低矩阵的维度到k，SVD的分解可以近似的写为：

Am×n≈Um×kΣk×kVTk×nAm×n≈Um×kΣk×kVk×nT

　　　　如果把上式用到我们的主题模型，则SVD可以这样解释：我们输入的有m个文本，每个文本有n个词。而AijAij则对应第i个文本的第j个词的特征值，这里最常用的是基于预处理后的标准化TF-IDF值。k是我们假设的主题数，一般要比文本数少。SVD分解后，UilUil对应第i个文本和第l个主题的相关度。VjmVjm对应第j个词和第m个词义的相关度。ΣlmΣlm对应第l个主题和第m个词义的相关度。

　　　　也可以反过来解释：我们输入的有m个词，对应n个文本。而AijAij则对应第i个词档的第j个文本的特征值，这里最常用的是基于预处理后的标准化TF-IDF值。k是我们假设的主题数，一般要比文本数少。SVD分解后，UilUil对应第i个词和第l个词义的相关度。VjmVjm对应第j个文本和第m个主题的相关度。ΣlmΣlm对应第l个词义和第m个主题的相关度。

　　　　这样我们通过一次SVD，就可以得到文档和主题的相关度，词和词义的相关度以及词义和主题的相关度。

　　　　在上面我们通过LSI得到的文本主题矩阵可以用于文本相似度计算。而计算方法一般是通过余弦相似度。

　　　　LSI是最早出现的主题模型了，它的算法原理很简单，一次奇异值分解就可以得到主题模型，同时解决词义的问题，非常漂亮。但是LSI有很多不足，导致它在当前实际的主题模型中已基本不再使用。

　　　　主要的问题有：

　　　　1） SVD计算非常的耗时，尤其是我们的文本处理，词和文本数都是非常大的，对于这样的高维度矩阵做奇异值分解是非常难的。

　　　　2） 主题值的选取对结果的影响非常大，很难选择合适的k值。

　　　　3） LSI得到的不是一个概率模型，缺乏统计基础，结果难以直观的解释。

　　　　对于问题1），主题模型非负矩阵分解（NMF）可以解决矩阵分解的速度问题。对于问题2），这是老大难了，大部分主题模型的主题的个数选取一般都是凭经验的，较新的层次狄利克雷过程（HDP）可以自动选择主题个数。对于问题3），牛人们整出了pLSI(也叫pLSA)和隐含狄利克雷分布(LDA)这类基于概率分布的主题模型来替代基于矩阵分解的主题模型。

　　　　回到LSI本身，对于一些规模较小的问题，如果想快速粗粒度的找出一些主题分布的关系，则LSI是比较好的一个选择。