https://www.cnblogs.com/pinard/p/6908150.html

[**用scikit-learn学习LDA主题模型**](http://www.cnblogs.com/pinard/p/6908150.html)

　　　　在[LDA模型原理篇](http://www.cnblogs.com/pinard/p/6831308.html)我们总结了LDA主题模型的原理，这里我们就从应用的角度来使用scikit-learn来学习LDA主题模型。除了scikit-learn,  还有spark MLlib和gensim库也有LDA主题模型的类库，使用的原理基本类似，本文关注于scikit-learn中LDA主题模型的使用。

**1. scikit-learn LDA主题模型概述**

　　　　在scikit-learn中,LDA主题模型的类在sklearn.decomposition.LatentDirichletAllocation包中，其算法实现主要基于原理篇里讲的变分推断EM算法，而没有使用基于Gibbs采样的MCMC算法实现。

　　　　而具体到变分推断EM算法，scikit-learn除了我们原理篇里讲到的标准的变分推断EM算法外，还实现了另一种在线变分推断EM算法，它在原理篇里的变分推断EM算法的基础上，为了避免文档内容太多太大而超过内存大小，而提供了分步训练(partial\_fit函数)，即一次训练一小批样本文档，逐步更新模型，最终得到所有文档LDA模型的方法。这个改进算法我们没有讲，具体论文在这：[“Online Learning for Latent Dirichlet Allocation”](https://www.cs.princeton.edu/~blei/papers/HoffmanBleiBach2010b.pdf) 。

　　　　下面我们来看看sklearn.decomposition.LatentDirichletAllocation类库的主要参数。

**2. scikit-learn LDA主题模型主要参数和方法**

　　　　我们来看看LatentDirichletAllocation类的主要输入参数:

　　　　1) **n\_topics:**即我们的隐含主题数KK,需要调参。KK的大小取决于我们对主题划分的需求，比如我们只需要类似区分是动物，植物，还是非生物这样的粗粒度需求，那么KK值可以取的很小，个位数即可。如果我们的目标是类似区分不同的动物以及不同的植物，不同的非生物这样的细粒度需求，则KK值需要取的很大，比如上千上万。此时要求我们的训练文档数量要非常的多。

　　　　2) **doc\_topic\_prior:**即我们的文档主题先验Dirichlet分布θdθd的参数αα。一般如果我们没有主题分布的先验知识，可以使用默认值1/K1/K。

　　　　3) **topic\_word\_prior:**即我们的主题词先验Dirichlet分布βkβk的参数ηη。一般如果我们没有主题分布的先验知识，可以使用默认值1/K1/K。

　　　　4) **learning\_method:**即LDA的求解算法。有 ‘batch’ 和 ‘online’两种选择。 ‘batch’即我们在原理篇讲的变分推断EM算法，而"online"即在线变分推断EM算法，在"batch"的基础上引入了分步训练，将训练样本分批，逐步一批批的用样本更新主题词分布的算法。默认是"online"。选择了‘online’则我们可以在训练时使用partial\_fit函数分布训练。不过在scikit-learn 0.20版本中默认算法会改回到"batch"。建议样本量不大只是用来学习的话用"batch"比较好，这样可以少很多参数要调。而样本太多太大的话，"online"则是首先了。

　　　　5）**learning\_decay：**仅仅在算法使用"online"时有意义，取值最好在(0.5, 1.0]，以保证"online"算法渐进的收敛。主要控制"online"算法的学习率，默认是0.7。一般不用修改这个参数。

　　　　6）**learning\_offset：**仅仅在算法使用"online"时有意义，取值要大于1。用来减小前面训练样本批次对最终模型的影响。

　　　　7） **max\_iter** ：EM算法的最大迭代次数。

　　　　8）**total\_samples：**仅仅在算法使用"online"时有意义， 即分步训练时每一批文档样本的数量。在使用partial\_fit函数时需要。

　　　　9）**batch\_size:**仅仅在算法使用"online"时有意义， 即每次EM算法迭代时使用的文档样本的数量。

　　　　10）**mean\_change\_tol** :即E步更新变分参数的阈值，所有变分参数更新小于阈值则E步结束，转入M步。一般不用修改默认值。

　　　　11） **max\_doc\_update\_iter:**即E步更新变分参数的最大迭代次数，如果E步迭代次数达到阈值，则转入M步。

　　　　从上面可以看出，如果learning\_method使用"batch"算法，则需要注意的参数较少，则如果使用"online",则需要注意"learning\_decay", "learning\_offset"，“total\_samples”和“batch\_size”等参数。无论是"batch"还是"online", n\_topics(KK), doc\_topic\_prior(αα), topic\_word\_prior(ηη)都要注意。如果没有先验知识，则主要关注与主题数KK。可以说，主题数KK是LDA主题模型最重要的超参数。

**3. scikit-learn LDA中文主题模型实例**

　　　　下面我们给一个LDA中文主题模型的简单实例，从分词一直到LDA主题模型。

　　　　我们的有下面三段文档语料，分别放在了nlp\_test0.txt, nlp\_test2.txt和 nlp\_test4.txt：

　　　　沙瑞金赞叹易学习的胸怀，是金山的百姓有福，可是这件事对李达康的触动很大。易学习又回忆起他们三人分开的前一晚，大家一起喝酒话别，易学习被降职到道口县当县长，王大路下海经商，李达康连连赔礼道歉，觉得对不起大家，他最对不起的是王大路，就和易学习一起给王大路凑了5万块钱，王大路自己东挪西撮了5万块，开始下海经商。没想到后来王大路竟然做得风生水起。沙瑞金觉得他们三人，在困难时期还能以沫相助，很不容易。

　　　　沙瑞金向毛娅打听他们家在京州的别墅，毛娅笑着说，王大路事业有成之后，要给欧阳菁和她公司的股权，她们没有要，王大路就在京州帝豪园买了三套别墅，可是李达康和易学习都不要，这些房子都在王大路的名下，欧阳菁好像去住过，毛娅不想去，她觉得房子太大很浪费，自己家住得就很踏实。

　　　　347年（永和三年）三月，桓温兵至彭模（今四川彭山东南），留下参军周楚、孙盛看守辎重，自己亲率步兵直攻成都。同月，成汉将领李福袭击彭模，结果被孙盛等人击退；而桓温三战三胜，一直逼近成都。

　　　　首先我们进行分词，并把分词结果分别存在nlp\_test1.txt, nlp\_test3.txt和 nlp\_test5.txt：

[复制代码](javascript:void(0);)

# -\*- coding: utf-8 -\*-

import jieba

jieba.suggest\_freq('沙瑞金', True)

jieba.suggest\_freq('易学习', True)

jieba.suggest\_freq('王大路', True)

jieba.suggest\_freq('京州', True)

#第一个文档分词#

with open('./nlp\_test0.txt') as f:

document = f.read()

document\_decode = document.decode('GBK')

document\_cut = jieba.cut(document\_decode)

#print ' '.join(jieba\_cut)

result = ' '.join(document\_cut)

result = result.encode('utf-8')

with open('./nlp\_test1.txt', 'w') as f2:

f2.write(result)

f.close()

f2.close()

#第二个文档分词#

with open('./nlp\_test2.txt') as f:

document2 = f.read()

document2\_decode = document2.decode('GBK')

document2\_cut = jieba.cut(document2\_decode)

#print ' '.join(jieba\_cut)

result = ' '.join(document2\_cut)

result = result.encode('utf-8')

with open('./nlp\_test3.txt', 'w') as f2:

f2.write(result)

f.close()

f2.close()

#第三个文档分词#

jieba.suggest\_freq('桓温', True)

with open('./nlp\_test4.txt') as f:

document3 = f.read()

document3\_decode = document3.decode('GBK')

document3\_cut = jieba.cut(document3\_decode)

#print ' '.join(jieba\_cut)

result = ' '.join(document3\_cut)

result = result.encode('utf-8')

with open('./nlp\_test5.txt', 'w') as f3:

f3.write(result)

f.close()

f3.close()

[复制代码](javascript:void(0);)

　　　　现在我们读入分好词的数据到内存备用，并打印分词结果观察：

[复制代码](javascript:void(0);)

with open('./nlp\_test1.txt') as f3:

res1 = f3.read()

print res1

with open('./nlp\_test3.txt') as f4:

res2 = f4.read()

print res2

with open('./nlp\_test5.txt') as f5:

res3 = f5.read()

print res3

[复制代码](javascript:void(0);)

　　　　打印出的分词结果如下：

　　　　沙瑞金 赞叹 易学习 的 胸怀 ， 是 金山 的 百姓 有福 ， 可是 这件 事对 李达康 的 触动 很大 。 易学习 又 回忆起 他们 三人 分开 的 前一晚 ， 大家 一起 喝酒 话别 ， 易学习 被 降职 到 道口 县当 县长 ， 王大路 下海经商 ， 李达康 连连 赔礼道歉 ， 觉得 对不起 大家 ， 他 最 对不起 的 是 王大路 ， 就 和 易学习 一起 给 王大路 凑 了 5 万块 钱 ， 王大路 自己 东挪西撮 了 5 万块 ， 开始 下海经商 。 没想到 后来 王大路 竟然 做 得 风生水 起 。 沙瑞金 觉得 他们 三人 ， 在 困难 时期 还 能 以沫 相助 ， 很 不 容易 。  
　　　　沙瑞金 向 毛娅 打听 他们 家 在 京州 的 别墅 ， 毛娅 笑 着 说 ， 王大路 事业有成 之后 ， 要 给 欧阳 菁 和 她 公司 的 股权 ， 她们 没有 要 ， 王大路 就 在 京州 帝豪园 买 了 三套 别墅 ， 可是 李达康 和 易学习 都 不要 ， 这些 房子 都 在 王大路 的 名下 ， 欧阳 菁 好像 去 住 过 ， 毛娅 不想 去 ， 她 觉得 房子 太大 很 浪费 ， 自己 家住 得 就 很 踏实 。  
　　　　347 年 （ 永和 三年 ） 三月 ， 桓温 兵至 彭模 （ 今 四川 彭山 东南 ） ， 留下 参军 周楚 、 孙盛 看守 辎重 ， 自己 亲率 步兵 直攻 成都 。 同月 ， 成汉 将领 李福 袭击 彭模 ， 结果 被 孙盛 等 人 击退 ； 而 桓温 三 战三胜 ， 一直 逼近 成都 。

　　　　我们接着导入停用词表，这里的代码和[中文文本挖掘预处理流程总结](http://www.cnblogs.com/pinard/p/6744056.html)中一样，如果大家没有1208个的中文停用词表，可以到之前的这篇文章的链接里去下载。

[复制代码](javascript:void(0);)

#从文件导入停用词表

stpwrdpath = "stop\_words.txt"

stpwrd\_dic = open(stpwrdpath, 'rb')

stpwrd\_content = stpwrd\_dic.read()

#将停用词表转换为list

stpwrdlst = stpwrd\_content.splitlines()

stpwrd\_dic.close()

[复制代码](javascript:void(0);)

　　　　接着我们要把词转化为词频向量，注意由于LDA是基于词频统计的，因此一般不用TF-IDF来做文档特征。代码如下：

[复制代码](javascript:void(0);)

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.decomposition import LatentDirichletAllocation

corpus = [res1,res2,res3]

cntVector = CountVectorizer(stop\_words=stpwrdlst)

cntTf = cntVector.fit\_transform(corpus)

print cntTf

[复制代码](javascript:void(0);)

　　　　输出即为所有文档中各个词的词频向量。有了这个词频向量，我们就可以来做LDA主题模型了，由于我们只有三个文档，所以选择主题数KK=2。代码如下：

lda = LatentDirichletAllocation(n\_topics=2,

learning\_offset=50.,

random\_state=0)

docres = lda.fit\_transform(cntTf)

　　　　通过fit\_transform函数，我们就可以得到文档的主题模型分布在docres中。而主题词 分布则在lda.components\_中。我们将其打印出来：

print docres

print lda.components\_

　　　　文档主题的分布如下：

[[ 0.00950072 0.99049928]

[ 0.0168786 0.9831214 ]

[ 0.98429257 0.01570743]]

　　　　可见第一个和第二个文档较大概率属于主题2，则第三个文档属于主题1.

　　　　主题和词的分布如下：

[[ 1.32738199 1.24830645 0.90453117 0.7416939 0.78379936 0.89659305

1.26874773 1.23261029 0.82094727 0.87788498 0.94980757 1.21509469

0.64793292 0.89061203 1.00779152 0.70321998 1.04526968 1.30907884

0.81932312 0.67798129 0.93434765 1.2937011 1.170592 0.70423093

0.93400364 0.75617108 0.69258778 0.76780266 1.17923311 0.88663943

1.2244191 0.88397724 0.74734167 1.20690264 0.73649036 1.1374004

0.69576496 0.8041923 0.83229086 0.8625258 0.88495323 0.8207144

1.66806345 0.85542475 0.71686887 0.84556777 1.25124491 0.76510471

0.84978448 1.21600212 1.66496509 0.84963486 1.24645499 1.72519498

1.23308705 0.97983681 0.77222879 0.8339811 0.85949947 0.73931864

1.33412296 0.91591144 1.6722457 0.98800604 1.26042063 1.09455497

1.24696097 0.81048961 0.79308036 0.95030603 0.83259407 1.19681066

1.18562629 0.80911991 1.19239034 0.81864393 1.24837997 0.72322227

1.23471832 0.89962384 0.7307045 1.39429334 1.22255041 0.98600185

0.77407283 0.74372971 0.71807656 0.75693778 0.83817087 1.33723701

0.79249005 0.82589143 0.72502086 1.14726838 0.83487136 0.79650741

0.72292882 0.81856129]

[ 0.72740212 0.73371879 1.64230568 1.5961744 1.70396534 1.04072318

0.71245387 0.77316486 1.59584637 1.15108883 1.15939659 0.76124093

1.34750239 1.21659215 1.10029347 1.20616038 1.56146506 0.80602695

2.05479544 1.18041584 1.14597993 0.76459826 0.8218473 1.2367587

1.44906497 1.19538763 1.35241035 1.21501862 0.7460776 1.61967022

0.77892814 1.14830281 1.14293716 0.74425664 1.18887759 0.79427197

1.15820484 1.26045121 1.69001421 1.17798335 1.12624327 1.12397988

0.83866079 1.2040445 1.24788376 1.63296361 0.80850841 1.19119425

1.1318814 0.80423837 0.74137153 1.21226307 0.67200183 0.78283995

0.75366187 1.5062978 1.27081319 1.2373463 2.99243195 1.21178667

0.66714016 2.17440219 0.73626368 1.60196863 0.71547934 1.94575151

0.73691176 2.02892667 1.3528508 1.0655887 1.1460755 4.17528123

0.74939365 1.23685079 0.76431961 1.17922085 0.70112531 1.14761871

0.80877956 1.12307426 1.21107782 1.64947394 0.74983027 2.03800612

1.21378076 1.21213961 1.23397206 1.16994431 1.07224768 0.75292945

1.10391419 1.26932908 1.26207274 0.70943937 1.1236972 1.24175001

1.27929042 1.19130408]]

　　　　在实际的应用中，我们需要对K,α,ηK,α,η进行调参。如果是"online"算法，则可能需要对"online"算法的一些参数做调整。这里只是给出了LDA主题模型从原始文档到实际LDA处理的过程。希望可以帮到大家。