

火光摇曳

夜幕降临之际，火光摇曳妩媚、灿烂多姿，是最美最美的... ..

[LDA工程实践之算法篇-1]算法实现正确性验证

🕒 2014/07/21 📁 机器学习、自然语言处理 🔖 LDA、perplexity、toy data、合成实验 👤 xueminzhao

研究生二年级实习（2010年5月）开始，一直跟着王益（yiwang）和新志辉（rickjin）学习LDA，包括对算法的理解、并行化和应用等等。毕业后进入了腾讯公司，也一直在从事相关工作，后边还在yiwang带领下，与孙振龙、严浩等一起实现了一套大规模并行的LDA训练系统——Peacock。受rick影响，决定把自己对LDA工程实践方面的一些理解整理出来，分享给大家，其中可能有一些疏漏和错误，还请批评指正。

Rickjin在《LDA数学八卦》[1]一文中已经对LDA的数学模型以及基本算法介绍得比较充分了，但是在工程实践上，我们还是有一些需要注意的问题，比如：

- 怎样验证算法实现的正确性？
- 怎样加速Gibbs sampling？
- 在线推断（inference）时，需要注意些什么问题？
- 超参数对模型的影响以及怎样做超参数优化？

本文将涉及以上内容，不包括：LDA并行化和应用，后续会在文章《LDA工程实践之架构篇》和《LDA工程实践之应用篇》中进行介绍。

为了方便大家理解，本文所有数学符号和 [2] 保持一致，具体见表 1。

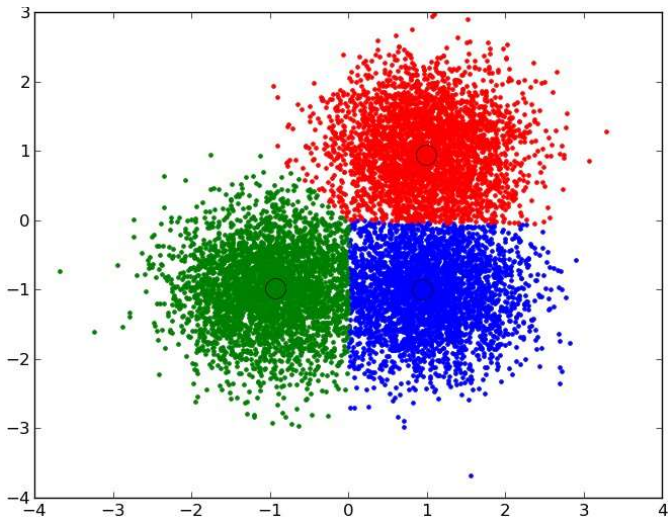
M	训练集文档数，标量。
N_m	文档 m 的长度，标量。
B	训练算法迭代次数，标量。
K	模型主题数，标量。
V	词典大小，标量。
$\vec{\alpha}$	文档主题的超参数， K 维向量。
$\vec{\beta}$	词的超参数， V 维向量。文中特意选取为对称的，即每个词对应的 β 都相同， $\vec{\beta} = \beta \vec{1}$ ，简写为标量 β 。
$\vec{\varphi}_k$	主题 k 的分布， V 维向量。模型 $\Phi = \{\vec{\varphi}_k\}_{k=1}^K$ 。
$\vec{\vartheta}_m$	文档 m 的主题分布， K 维向量。
\vec{w}	文档的词向量。 $\{\vec{w}\}$ 表示文档集合，比如训练集等。
\vec{z}	文档的词对应的主题向量。 $\{\vec{z}\}$ 表示文档的词向量对应的主题向量集合。
$w_{m,n}$	文档 m 中第 n 个词的 indicator。
$z_{m,n}$	文档 m 中第 n 个词对应主题的 indicator。
n_k^t	集合中词 t 赋予主题 k 的频次。
n_k	集合中主题 k 出现的频次， $n_k = \sum_{t=1}^V n_k^t$ 。
n_m^k	文档 m 中主题 k 的频次。

Table 1: Symbols

1 算法实现正确性验证

在实现机器学习算法的时候，由于数值算法特有的收敛性问题，让这项本来相对简单的工作增加了难度。这其中的典型是多层次神经网络的优化算法——反向传播（Back Propagation，BP）算法，由于神经网络的强大表述能力，即使实现有误，在简单数据实验上，我们可能也发现不了问题。LDA算法的实现较BP简单，工作中我们常采用如下几个方法进行算法正确性的先期验证。

1.1 Toy data实验



是可预测的（表 2 数据收敛后，Doc1-3的词赋予的主题应该都是1，Doc4-6的词赋予的主题应该都是2，或者二者主题互换）。

文档 Id	文档内容
Doc1	apple orange banana
Doc2	apple orange
Doc3	orange banana
Doc4	cat dog
Doc5	dog tiger
Doc6	tiger cat

Table 1: LDA toy data

随机算法在开发调试过程中，稳定不变的随机数序列是非常重要的，这样有利于定位问题。获取稳定不变的随机数非常简单，只需要我们额外提供一个伪随机数种子的命令行参数。

1.2 合成实验

算法包最终实现，toy data实验符合预期，此时如果我们想进一步验证LDA算法的效果呢？考虑到LDA是一种生成模型[3]，Griffiths等人[4]在论文中采用合成实验来演示模型的效果，当然，这也可以作为算法正确性的验证。

假设已知LDA模型的主题数 K 、词典大小 V 、文档主题的超参数 α 、主题 k 的分布 φ_k ，生成 M 个长度为 N_m 文档的过程如算法LdaGenerate。

假设已知LDA模型主题数 K 、文档主题超参数 α 、词的超参数 β ，训练文档 $\{\vec{w}\}$ ，以及迭代次数 B ，LDA训练算法如算法LdaGibbs（注意：此处主要给出框架，具体算法可以参考[4,2]）。其中具体符号含义可以参考表1， i 表示当前词即 $i = (m, t)$ ， $\neg i$ 表示剔除词 i （比如 $n_{k, \neg i}$ 表示剔除词 i 的主题后主题 k 的频次，因此在具体实现算法时，通常采样之前会对 n_m^k 、 n_k^t 和 n_k 进行减减操作）。训练算法除了输出模型，还可以得到一个副产物 $\vec{\vartheta}_m$ ——训练文档 m 的主题分布（Eq. 3）。

$$p(z_i = k | \{\vec{z}\}_{\neg i}, \{\vec{w}\}) = \frac{n_{k, \neg i}^t + \beta}{n_{k, \neg i} + \beta V} \cdot \frac{n_{m, \neg i}^k + \alpha_k}{N_m - 1 + \sum_{k=1}^K \alpha_k}$$
$$\propto \frac{n_{k, \neg i}^t + \beta}{n_{k, \neg i} + \beta V} (n_{m, \neg i}^k + \alpha_k)$$

(1)

$$\varphi_{k\ t} = \frac{n_k^t + \beta}{n_k + \beta V}$$

(2)

时，我们将训练主题数设置为真实值，如果不是真实值会怎么样呢？小伙伴们赶紧自己动手试一试吧！

Figure 2: Griffiths Ground truth Φ

Griffiths等人 [4] 为了使合成实验更加直观，将每个 $\vec{\varphi}_k$ 渲染成大小为 $\sqrt{V} \times \sqrt{V}$ 图片，其中 φ_k^t 为对应图片位置的像素值（相当于将 V 维向量 $\vec{\varphi}_k$ Reshape成大小为 $\sqrt{V} \times \sqrt{V}$ 的图像矩阵；同时，因为 $\vec{\varphi}_k$ 为浮点向量，图像像素值为0-255整数，我们需要将向量 $\vec{\varphi}_k$ 元素值 Normalized成0-255的整数， φ_k^t 值越大，对应图像像素越“亮”。Griffiths等人选用的真实 Φ 是10张包含白色Bar的图片，相当于 $K = 10$ ，详情见图2（真实模型 $\Phi_{10 \times V}$ 的一行对应一张图片）。

Figure 3: Griffiths Synthesis Experiment [4]

图3给出了预估 $\tilde{\Phi}$ 渲染成的图像随训练迭代的变化情况，可以看到预估 $\tilde{\Phi}$ 渲染成的图像逐渐“清晰”，模型质量越来越好。我们采用中国的十二生肖图像，重复了Griffiths等人的实验，如图4和5。

Figure 4: Ground truth Φ

Figure 5: Estimated $\tilde{\Phi}$

合成实验过程中需要用到Dirichlet采样，一般的标准库中没有提供：对c/c++来说，gsl [5] 是不错的选择；对python来说，numpy [6] 有提供实现。

合成实验的基本原理就是这样，虽然简单，但是如果我们善加利用，却可以得出许多有用的结论，比如利用合成实验来模拟LDA算法在“真实”的互联网语料数据上的表现。互联网语料的模型 Φ 至少应该具有如下性质：主题数 K 非常大（比如几十万的级别）；主题之间具有层次关系；语料中主题和词的出现频次应该满足长尾分布。给定满足这些性质的模型 Φ 以后，生成语料，我们就可以实验不同的训练算法变形的具体效果，以及各种算法参数对预估模型质量的影响。

1.3 Perplexity曲线

在自然语言处理中，Perplexity [7] 常用来度量语言模型的质量，值越小，模型质量越好。Perplexity定义为模型在给定测试集上每个词似然度（likelihood）的几何平均的倒数 [2,8]。在给定测试集 $\{\vec{w}\}$ 上，模型的Log Likelihood为 Eq. 4，Perplexity和Log Likelihood之间满足Eq. 5。

$$Loglikelihood(\{\vec{w}\}|\mathcal{M}) = \sum_{m=1}^M \sum_{n=1}^{N_m} \log(p(w_{m,n}|\mathcal{M})) \quad (4)$$

$$Perplexity(\{\vec{w}\}|\mathcal{M}) = \exp\left(-\frac{Loglikelihood(\{\vec{w}\}|\mathcal{M})}{\sum_{m=1}^M N_m}\right) \quad (5)$$

具体到LDA模型，Perplexity计算公式如Eq. 6。训练过程中，计算Perplexity严谨的做法应该使用

$$\begin{aligned}
Perplexity(\{\vec{w}\}|\mathcal{M}) &= \left(\prod_{m=1}^M \prod_{n=1}^{N_m} p(w_{m,n}|\mathcal{M}) \right)^{-\sum_{m=1}^M 1/N_m} \\
&= \exp \left(\log \left(\left(\prod_{m=1}^M \prod_{n=1}^{N_m} p(w_{m,n}|\mathcal{M}) \right)^{-\sum_{m=1}^M 1/N_m} \right) \right) \\
&= \exp \left(- \frac{\sum_{m=1}^M \sum_{n=1}^{N_m} \log(p(w_{m,n}|\mathcal{M}))}{\sum_{m=1}^M N_m} \right) \\
&= \exp \left(- \frac{\sum_{m=1}^M \sum_{n=1}^{N_m} \log \sum_{k=1}^K p(w_{m,n}, z_{m,n} = k|\mathcal{M})}{\sum_{m=1}^M N_m} \right) \\
&= \exp \left(- \frac{\sum_{m=1}^M \sum_{n=1}^{N_m} \log \sum_{k=1}^K p(w_{m,n} = t|z_{m,n} = k)p(z_{m,n} = k|m)}{\sum_{m=1}^M N_m} \right) \\
&= \exp \left(- \frac{\sum_{m=1}^M \sum_{n=1}^{N_m} \log \sum_{k=1}^K \varphi_{k,t} \vartheta_{k,m}}{\sum_{m=1}^M N_m} \right)
\end{aligned} \tag{6}$$

分享到...

LDA模型训练过程中，随着迭代的进行，模型的Perplexity曲线会逐渐收敛。因此，我们通常会根据训练过程中模型的Perplexity曲线是否收敛来判定模型是否收敛。Perplexity曲线收敛性也从侧面可以证明算法实现的正确性。图6给出了一次模型训练过程的LogLikelihood和Perplexity曲线（主题数 $K = 10,000$ ，迭代130左右的曲线突变将在第四章给出解释）。

Figure 6: LogLikelihood and perplexity curve


注意：合成实验小节中图3同时给出了模型训练过程中，Log Likelihood取值和预估 $\tilde{\Phi}$ 的图像情况，可以看到Log Likelihood曲线收敛后，预估 $\tilde{\Phi}$ 的图像任然有非常正向的变化，说明模型还在优化。因此，模型训练时，工程上一般在Log Likelihood曲线收敛后，任然继续进行一定量的迭代再输出最终模型。至于Log Likelihood曲线的收敛和模型的收敛之间的关系究竟如何呢，小伙伴们知道么？

参考文献

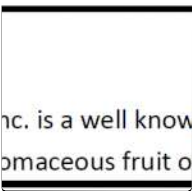
- [1] 靳志辉. LDA数学八卦. <http://cos.name/2013/03/lda-math-lda-text-modeling>.
- [2] Gregor Heinrich. Parameter estimation for text analysis. Technical Report, 2009.
- [3] Generative model. http://en.wikipedia.org/wiki/Generative_model.
- [4] Thomas L. Griffiths, and Mark Steyvers. Finding scientific topics. In PNAS '2004.
- [5] http://www.gnu.org/software/gsl/manual/html_node/The-Dirichlet-Distribution.html.
- [6] <http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.dirichlet.html>.
- [7] Perplexity. <http://en.wikipedia.org/wiki/Perplexity>.
- [8] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. In JMLR '2003.

本文链接：[\[LDA工程实践之算法篇-1\]算法实现正确性验证](#)

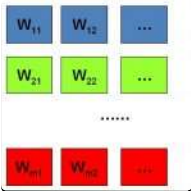
相关文章



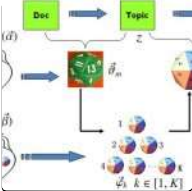
2014/10/27
[LDA工程实践之算法篇-2]
SparseLDA算法




2015/03/02
Peacock: 大规模主题模型及其在腾讯业务中的应用



2014/06/17
[LDA数学八卦-4]
文本建模



2014/06/17
[LDA数学八卦-5]LDA 文本建模




2014/06/19
[我们这样理解语言的-1]文本分析平台TextMiner



10 条评论


最新 最早 最热



数急

这种LDA模型在实践中有个疑问：1.主题数K如何设置？有动态确定的办法吗？ 2.如何设置超参数alpha和beta

2014年8月4日 回复 顶 转发



xueminzhao

1. 对lda，K一般还是根据应用来cross validation，没有非常好的办法。
那些不设定K的算法或者有别的参数，或者工程实现上复杂度较高（比如hdp），我们没有采用。
如果最优的K=1000，我们设置成了2000，其实效果上也差不多太多，所以。。。当然这个问题有学术上的价值。

2. alpha/beta我们是自动优化的，当然初始值也有一些讲究，后边会有介绍。



是根据Parameter estimation for text analysis中的介绍自动优化的吗？优化，是在Training还是在Inference中？

2014年10月30日 回复 顶
转发



MC、兜

博主。。。可不可以请教你几个问题。。。关于MCMC的。博主能否给我举一个其他方法不能采样的 而能用MCMC-Metropolis-Hastings（其实是不能理解其用途），另外算法中为什么要设置一个接受率。。。博主可否给一个你的联系方式 有问题 可以找你请教一下？

2014年8月14日 回复 顶 转发



lightman

引入接受率，构造新的转移矩阵，让新构造的转移矩阵对应的平稳分布为 $p(x)$

2015年4月1日 回复 顶 转发



lightman

引入接受率，构造新的转移矩阵，让新构造的转移矩阵对应的平稳分布为 $p(x)$

2015年4月1日 回复 顶 转发



aeolus

Log Likelihood曲线收敛后，模型不是也应该收敛了吗

2015年5月14日 回复 顶 转发



laoyang945

博主你好，请问如何确定训练文本的分类和LDA的分类的对应关系呢？我想的是用LDA再分类一下训练文本，然后就知道关系了。有没有更快的方法呢？谢谢！

2015年7月28日 回复 顶 转发



谢晨阳

博主你好~对于perplexity的计算还是有很多盲点，我用的是TEH的MATLAB的代码，里边没有给出perplexity的计算方法，自己去写的时候又不知道从何下手，您有时间的话可以



说点什么吧...

发布

火光摇曳正在使用多说

分享到
...