一种基于动态词汇表的在线LDA算法

张健伟 严建峰 刘晓升 杨 璐
(苏州大学计算机科学与技术学院 苏州 215006)

摘 要 目前的在线潜在狄利克雷分布模型(LDA)算法大多是基于固定的词汇表,在实际应用中经常会出现词汇表和处理的语料不匹配的情况,影响了模型的实用性。针对这个现象,在置信传播算法(BP)的框架下,使主题单词分布服从狄利克雷过程,重新推导公式,使得词汇表在模型运行之前为空,并且在处理时不断向词汇表中增加发现的新词。实验证明,这种新的基于动态词汇表的算法不仅使得词汇表与语料的贴合度更高,而且使其在混淆度以及互信息指数这两个指标上能够比基于固定词汇表的LDA模型表现得更加优越。

关键词 潜在狄利克雷分配,动态词汇表,狄利克雷过程,流处理

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.12.021

Online LDA on Dynamic Vocabulary

ZHANG Jian-wei YAN Jian-feng LIU Xiao-sheng YANG Lu
(School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

Abstract Most of the online LDA algorithms are based on the fixed vocabulary table currently. The vocabulary table may not often match the processed corpus in practice which has a bad effect on the precision of LDA. To solve this problem, we let the topic words distribution subject to the dirichlet process (DP) and re-deduce the model under the framework of BP algorithm. So that we can make the vocabulary table empty before the algorithm running and it can continually add new words to table. Results from the experiments show that, our new algorithm can make the vocabulary table match the corpus better and the dynamic vocabulary table makes the new algorithm achieve better performance on perplexity and PMI compared with other state-of-the-art fixed vocabulary online algorithms.

Keywords Latent dirichlet allocation, Dynamic vocabulary, Dirichlet process, Streaming process

1 引言

潜在狄利克雷分布(Latent Dirichlet Allocation, LDA)[1]是目前非常流行的一种矩阵分解算法,它能把高维稀疏的文档单词矩阵分解为两个低维稠密的文档主题矩阵和主题单词矩阵。批处理LDA算法处理的是固定数量文档的小语料。在面对数量级很大的语料或是实时的流数据时,批处理LDA算法就会出现问题,因为批处理LDA算法需要在算法运行之前把数据完整地保存到内存,而大语料可能造成内存负担过重,实时的流数据也不可能在一个时间点全部到达内存。在线LDA算法[8-10,13]就是针对这个问题提出的解决方案。在线LDA算法首先会将大语料切分成若干份小语料,然后顺序处理这些小语料。所以算法运行的某个时间点上只需要将一个小语料放入内存即可,这样就缓解了批处理LDA算法内存开销过大的问题。并且,在线LDA算法的这种处理方式天然地解决了批处理LDA算法在处理实时数据上的局限,实时的流数据可以按时间点组织语料,在线LDA算法即可对实时数据按时间点一批一批的处理。

在目前的LDA算法中,词汇表都是固定的,并且在算法执行前就要确定,这就造成了在算法扫描文档时可能会忽略一些不在词汇表中但是很重要的词。这在面对确定的语料时不是非常严重,因为可以在算法执行前先去扫描一遍语料,将所有出现的词组成词汇表。但是在面对实时的流数据时这个问题就会很严重,因为无法确定每批进来的数据会有哪些单词。如果将词汇表变得很大,能覆盖所有可能出现的单词,就会造成主题单词矩阵过大,内存负担过重。因此固定的词汇表是阻碍在线LDA算法实用性的一个瓶颈。

造成这个问题的主要原因是传统的LDA算法中的主题单词分布服从狄利克雷分布,而狄利克雷分布需要确定原子的个数。本文针对这个问题,提出使主题单词分布服从狄利克雷过程[12],在置信传播算法[2,13]的框架下重新推导LDA模型,使得词汇表具有了灵活性,从而使得LDA模型中的词汇表与处理的语料更加贴合。从实验中也可以发现,更加灵活的词汇表使得LDA模型可以在混淆度和互信息指数这两个评价LDA算法常用的指标上比固定词汇表的主题模型表现得更加优越。

到稿日期:2015-12-22 返修日期:2016-03-25 本文受国家自然科学基金(61373092,61572339,61272449),江苏省科技支撑计划重点项目(BE2014005)资助。

张健伟(1992—),男,硕士生,主要研究方向为机器学习;**严建峰**(1978—),男,副教授,硕士生导师,主要研究方向为机器学习,E-mail:yanjf@suda.edu.cn(通信作者);**刘晓升**(1976—),男,博士生,主要研究方向为机器学习;**杨 璐**(1982—),女,副教授,硕士生导师,主要研究方向为机器学习与软件工程。

2 相关工作

2.1 潜在狄利克雷分布

LDA 模型是一种无监督的概率图模型[1]。模型假定每篇文档(d)都可以用一个长度为 K 的主题分布(θ_d)表示，K 为设定的主题数。每个主题可以用一个长度为词汇表长度的单词分布(ϕ_k)表示。

$$\theta_d \sim Dir(\alpha), \phi_k \sim Dir(\beta), z_i \sim \theta_d, x_i \sim \phi_{z_i} \tag{1}$$

LDA 模型假定 θ_d 和 ϕ_k 都服从狄利克雷分布。在 LDA 模型中，一篇文档的生成过程如下：从一个先验为 α 的狄利克雷分布中采样一篇文档 d 的主题分布 θ_d，从一个先验为 β 的狄利克雷分布中采样每个主题 k 的分布 ϕ_k，接下来从 θ_d 中采样文档 d 中的每个单词 t 的主题 z_i，再从主题单词分布 ϕ_{z_i} 中采样一个单词 x_i，重复这样的过程直到生成所有的文档。图 1 是 LDA 模型的图模型。表 1 列出了本文中与 LDA 模型相关的一些参数。

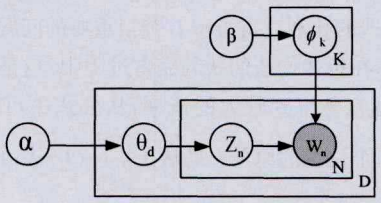


图 1 LDA 图模型

表 1 符号标签

符号	意义
$1\leqslant d\leqslant D$	语料库文本索引
$1\leqslant w\leqslant W$	单词表中单词索引，W 为词汇表的长度
$1\leqslant k\leqslant K$	主题索引
$x_{w,d}$	索引为{w,d}的单词的词频
$z^k_{w,d}$	文本 d 中所有的单词 w 属于主题 k 的个数
z	记录语料中所有词的主题，是 $z^k_{w,d}$ 的集合
x	所有的 $x_{w,d}$ 的集合
θ_d	文本 d 的主题分布
$\theta_d(k)$	文本 d 的主题分布中主题 k 的概率
$\varphi(k)$	主题 k 的单词分布
$\varphi_w(k)$	主题 k 的单词分布中单词 w 的概率
$\hat{\theta}_d(k)$	文本 d 的主题分布中在主题 k 的概率计数
$\hat{\varphi}_w(k)$	主题 k 的单词分布中单词 w 的概率计数
$\mu^k_{w,d}$	文档 d 中的单词 w 属于主题 k 的概率
α,β	狄利克雷分布的超参
$V_w(k)$	主题 k 下单词 w 的折棒产生的权重概率
$\hat{V}_w(k)$	$V_w(k)$ 的概率计数

2.2 置信传播算法

主题模型的任务可以理解为给每篇文档的每个单词贴主题标签 $z^k_{w,d}$。利用积分去除完整的 LDA 模型的似然函数 $p(x,z,\theta,\varphi|\alpha,\beta)$ 中的 θ 和 φ 后，可以得到塌陷的 LDA 模型的似然函数：

$$\begin{aligned} p(x,z|\alpha,\beta) &= p(w|z,\beta)\cdot p(z|\alpha) \\ &\propto \prod_d \prod_k \Gamma(\sum_w x_{w,d} z^k_{w,d} + \alpha) \times \prod_d \prod_k \Gamma(\sum_d x_{w,d} z^k_{w,d} + \\ &\beta) \times \prod_k (\sum_{w,d} x_{w,d} z^k_{w,d} + w\beta)^{-1} \end{aligned} \tag{2}$$

其中，$\Gamma(\cdot)$ 是伽马函数。为了最大化式子中 z 的概率，置信传播算法会计算 $\mu^k_{w,d}$ 的后验概率，$\mu^k_{w,d} = p(z^k_{w,d}=1 | z^k_{-(w,d)}, x)$，在置信传播算法中 $\mu^k_{w,d}$ 被称为消息，并且：

$$\mu^k_{w,d} = \frac{(\hat{\theta}_{-w,d}(k)+\alpha)\times(\hat{\varphi}_{w,-d}(k)+\beta)}{\hat{\varphi}_{-(w,d)}(k)+W\beta} \tag{3}$$

其中，$\hat{\theta}_d(k)$ 和 $\hat{\varphi}_w(k)$ 的计算公式为：

$$\hat{\theta}_{-w,d}(k) = \sum_{-w} x_{w,d}\mu^k_{w,d} \tag{4}$$

$$\hat{\varphi}_{w,-d}(k) = \sum_{-d} x_{w,d}\mu^k_{w,d} \tag{5}$$

其中，$-w$ 代表文档中所有的单词除了 w，$-d$ 代表所有的文档除了 d。可以从上面的公式看到消息 $\mu^k_{w,d}$ 依赖其他所有相邻的消息 $\mu_{-(w,d)}$。两个重要的参数文档 d 的主题分布 θ_d 和主题 k 的单词分布 $\varphi(k)$，可以通过狄利克雷归一化 $\hat{\theta}$ 和 $\hat{\varphi}$ 得到：

$$\theta_d(k) = \frac{\hat{\theta}_d(k)+\alpha}{\sum_k \hat{\theta}_d(k)+K\alpha} \tag{6}$$

$$\varphi_w(k) = \frac{\hat{\varphi}_w(k)+\beta}{\sum_w \hat{\varphi}_w(k)+W\beta} \tag{7}$$

置信传播算法会在训练语料上不停地计算式(3)，直到 $\theta_d(k)$ 和 $\varphi_w(k)$ 收敛到稳定的值。

3 动态词汇表的 LDA 算法

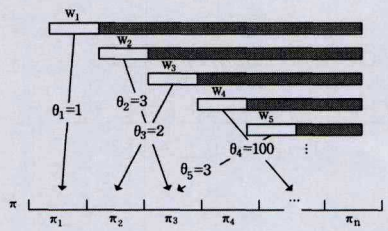
3.1 狄利克雷过程

狄利克雷过程(Dirichlet Process，DP)[12]可以看成狄利克雷分布在无限维上的实现。狄利克雷过程是一个概念，需要一种确定的方案实现，目前主要有 3 种方案可以构造：折棒(Stick-breaking)构造[4]、中国餐馆(Chinese Restaurant)构造、波利亚罐(Polya urn)构造。因为折棒(Stick-breaking)构造的理论更加清晰简单，所以本文模型使用 Stick-breaking 构造狄利克雷过程。

折棒构造的狄利克雷过程是一个二参数的分布，表示为 $DP(\alpha,G_0)$，其中 α 是贝塔分布的先验参数，G_0 是狄利克雷过程的基分布。如果 π 服从 $DP(\alpha,G_0)$，那么 π 可以通过如下方程来构造：

$$\begin{aligned} &\pi = \sum_{i=1}^{\infty}(V_i \prod_{j=1}^{i-1}(1-V_j)\delta_{\theta_j}) \\ &V_i \sim Beta(1,\alpha), \theta_j \sim G_0 \end{aligned} \tag{8}$$

其中，$V_i\prod_{j=1}^{i-1}(1-V_j)$ 被称为折棒构造产生的权重系数概率。在第 i 步，V_i 服从贝塔分布，是从完整的单位长度的棒的剩余长度 $1\times\prod_{j=1}^{i-1}(1-V_j)$ 中折下来的。G_0 是基分布，用于确定每一步产生的权重系数概率分配给 π 中的哪个原子。图 2 展示了折棒构造的示意图。



$w_1 = v_1$

$w_2 = v_2(1-v_1)$

$w_3 = v_3(1-v_2)(1-v_1)$

$w_4 = v_4(1-v_3)(1-v_2)(1-v_1)$

$w_5 = v_5(1-v_4)(1-v_3)(1-v_2)(1-v_1)$

…

$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \cdots \sim G_0$

图 2 折棒构造示意图

3.2 基于动态词汇表的 LDA 模型

本文提出的模型 dvOBP(Dynamic Vocabulary Online Belief Propagation)主要改变了 LDA 模型中主题单词分布服从的分布，将狄利克雷分布换成狄利克雷过程，$\varphi(k)\sim DP(\alpha^\beta, G_0)$。在置信传播[2,13]的算法框架下重新推导模型。基于动态词汇表的 LDA 模型中一篇文章的生成过程与传统的 LDA 模型中的文档生成过程类似，不过基于动态词汇表的 LDA 模型是从一个狄利克雷过程中采样单词。

在 dvOBP 的 $\varphi(k)$ 中，主题 k 中的每个单词对应的概率 $\varphi_w(k)$ 只对应一个折棒产生的权重系数，所以每个 $\varphi_w(k)$ 只对应一个 $V_w(k)$，并且：

$$p(V_w(k)|\alpha^\beta)=Beta(1,\alpha^\beta) \tag{9}$$

并且主题 k 下，单词 w 对应的概率 $\varphi_w(k)$ 依据折棒构造的公式为：

$$\varphi_w(k)=V_w(k)\prod_{j=1}^{LOC(w,k)}(1-V_{WORD(j,k)}(k)) \tag{10}$$

其中，$LOC(w,k)$ 函数是定位单词 w 在主题 k 的单词分布中的位置。而 $WORD(j,k)$ 是确定主题 k 的单词分布中坐标为 j 的单词。设定 LOC 函数和 $WORD$ 函数是因为每个主题的单词分布中单词的位置是不固定的。

在式(2)中，LDA 的似然函数 $p(x,z|\alpha,\alpha^\beta)$ 可分解为 $p(z|\alpha)$ 和 $p(w|z,\alpha^\beta)$ 的乘积，由于只是改变了主题单词分布所服从的分布，文档主题分布依旧服从狄利克雷分布，故在 dvOBP 中 $p(z|\alpha)$ 与传统的 LDA 算法一样，而 $p(w|z,\alpha^\beta)$ 变为：

$$\begin{aligned}
&p(w|z,\alpha^\beta)\\
&=\int p(w,V\mid z,\alpha^\beta)\mathrm{d}V\\
&=\int p(w\mid z,V)p(V\mid\alpha^\beta)\mathrm{d}V\\
&=\prod_k\prod_w\frac{\Gamma(1+\alpha^\beta)}{\Gamma(1)\Gamma(\alpha^\beta)}\times\\
&\frac{\Gamma(1+\hat{V}_w(k))\Gamma(\alpha^\beta+\sum_{j=LOC(w,k)+1}^{\infty}\hat{V}_{WORD(j,k)}(k))}{\Gamma(1+\alpha^\beta+\sum_{j=LOC(w,k)}^{\infty}\hat{V}_{WORD(j,k)}(k))}
\end{aligned} \tag{11}$$

其中，V 是所有的 $V_w(k)$ 的集合，$\hat{V}_w(k)$ 是 $V_w(k)$ 的概率计数。与置信传播算法的处理方式一样，为了最大化 z 的似然，通过计算式(12)：

$$\begin{aligned}
\mu^k_{w,d}&=p(z^k_{w,d}=1|z^k_{-(w,d)},x)\\
&=\frac{p(W|z)}{p(W_{-(w,d)}|z_{-(w,d)})}\frac{p(z)}{p(z_{-(w,d)})}\\
&\propto\frac{(1+\hat{V}_{w,-d}(k))\times(\hat{\theta}_{-w,d}(k)+\alpha)}{1+\alpha^\beta+\hat{V}_{-(w,d)}(k)}
\end{aligned} \tag{12}$$

其中：

$$\begin{aligned}
&\hat{\theta}_{-w,d}(k)=\sum_{-w}x_{w,d}\mu^k_{w,d}\\
&\hat{V}_{w,-d}(k)=\sum_{-d}x_{w,d}\mu^k_{w,d}\\
&\hat{V}_{-(w,d)}(k)=\sum_{w',w'\neq w}\hat{V}_{w'}(k)+\hat{V}_{w,-d}(k)
\end{aligned} \tag{13}$$

可以发现，dvOBP 的消息更新式(12)和置信传播的消息更新式(3)非常相似，只是公式分母上的概率平滑项有所不同，在置信传播算法中为 $W\beta$，而在 dvOBP 中为 $1+\alpha^\beta$。在两个重要参数文档 d 的主题分布 θ_d 和主题 k 的单词分布 $\varphi(k)$ 上，θ_d 的计算公式与式(6)一样，而 $\varphi(k)$ 的计算公式为：

$$\begin{aligned}
V_w(k)&=\frac{1+\hat{V}_w(k)}{1+\alpha^\beta+\sum_{j=LOC(w,k)}^{\infty}\hat{V}_{WORD(j,k)}(k)}\\
\varphi_w(k)&=V_w(k)\prod_{j=1}^{LOC(w,k)-1}(1-V_{WORD(j,k)}(k))
\end{aligned} \tag{14}$$

3.3 dvOBP 算法核心实现

3.3.1 dvOBP 在基分布上的选择

在 dvOBP 中，基分布 G_0 选择的是均匀分布。Ishwaran 和 Zarepour[11]证明了当基分布为均匀分布且 π 中的原子的数量级很大时，上述多个权重系数概率分配给 π 中同一个原子的情况出现的概率趋向于 0，我们有理由设定主题下的每个单词只会分配到一个折棒产生的概率，从而使得主题单词分布基于狄利克雷过程的 LDA 模型推导具有了可实施性。

3.3.2 对于狄利克雷过程位置偏置问题的处理方式

折棒构造的狄利克雷过程有个很重要的问题就是位置偏置问题[4,12]，在折棒构造的狄利克雷过程中，越是靠前位置折棒的原子，越容易得到较大的概率，从公式中可以直观地看出，处于第 i 个位置的原子 i 是从 $1\times\prod_{j=1}^{i-1}(1-V_j)$ 剩余的概率中折下一段概率，i 越小，$1\times\prod_{j=1}^{i-1}(1-V_j)$ 就越大，留下来的概率就越大。在 dvOBP 中把位置偏置作为主题分布下每个单词的先验的一部分。在 dvOBP 中处理完每批数据时，对每个主题 k 下的单词分布中的单词的位置进行重新排列，将 $\hat{V}_w(k)$ 从大到小排列，使得在每个主题下比较重要的词(概率大)拥有较大的先验知识。

3.3.3 单词表的截断

由于狄利克雷过程是无限维的，而在将狄利克雷过程应用到 LDA 模型中时，无需使得主题单词分布中包含所有可能出现的词，只需要包含在模型处理的文档中出现的所有单词。针对这个情况，可以采取截断技术使得主题单词分布仅包含已经发现的单词。故在 dvOBP 中，会给出每个主题 k 的单词分布中处于最后一个位置的单词 w 之前所有单词折掉的概率的剩余概率：

$$\varphi_w(k)=1-\sum_{j=1}^{LOC(w,k)-1}\varphi_{WORD(j,k)}(k) \tag{15}$$

这样可以使得 dvOBP 的每个主题的单词分布不再是无限维的，使得算法具有了可实现性，并且使 $\sum_w\varphi_w(k)=1$，保证了概率分布的严谨性。

3.4 dvOBP 的算法流程

基于动态词汇表的在线 LDA 算法如下所示。

算法 基于动态词汇表的在线 LDA 算法

初始化：词汇表为 0，将大语料或者流数据组织成一批一批的小量级的文档的集合 Ds

对于每一批到来的文档集 Ds：

1. 将数据导入到内存
2. 扫描 Ds，将所有的新单词插入到词汇表中
3. 随机给每个单词 $x^s_{w,d}$ 分配主题，并初始化消息参数 $\mu^k_{w,d}$
4. for t=1 to T do;//迭代循环，T 为最大迭代次数
5. Residual=0 //Residual 是残差变量

6. for d∈D_s,w∈d do:
7. //对 D_s 中每篇文档的每个单词的迭代
8. for k=1 to K;//更新每个 $\mu^k_{w,d}$
9. 通过消息更新式(12)更新 $\mu^k_{w,d}$
10. 用 $\mu^k_{w,d}$来更新 $\theta_d(k)$和 $V_w(k)$
11. 把 $\mu^k_{w,d}$的变化量加到 Residual 变量
12. If Residual / (D_s 中所有单词数)<0.01:
13. //收敛条件
14. Break //退出循环
15. D_s 移出外存
16. 根据 V(k)从大到小排列每个主题单词分布下每个单词的位置
17. 将 D_s 上的 θ 矩阵保存,将 V 留在内存处理下一批数据

3.5 与同类算法的比较

Ke Zhai[5]提出了在服从狄利克雷过程的主题单词分布之上的 LDA 模型(Infvoc-LDA)。本文提出的模型和 Infvoc-LDA 相比主要有 3 点不同。

Infvoc-LDA 是基于稀疏随机推断(Sparse Stochastic Inference,SOI)[6]算法框架的,而本文的模型是基于置信传播算法框架的。Zeng[10]证明了置信传播算法比 SOI 算法具有更好的精度以及收敛速度[10]。

Infvoc-LDA 会在算法开始前在一份英文字典上训练一个 n 元语言模型(N-gram Language Model),之后在算法运行时,Infvoc-LDA 会给出在 n 元语言模型中概率高的单词在主题单词分布中较大的先验。由于训练 n 元语言模型的字典和 LDA 算法处理的语料相关性不大,因此 Infvoc-LDA 可能会给出一些在 LDA 处理的语料中很重要的词很低的先验,从而可能丢失一些信息。另外,由于需要计算一个 n 元语言模型,Infvoc-LDA 的时间复杂度也较高。而本文提出的模型中每个单词在主题单词分布中的先验是由处理的语料决定的,更加贴合处理的语料。

Infvoc-LDA 会在处理完每批数据之后,淘汰每个主题下的单词分布中概率较小的一部分单词,但是这样可能会造成信息流失。因为有一部分词在每批数据中可能不会出现得很多,并且在 n 元语言模型中的概率较少造成先验很小,这些词不太会在某一批数据处理完之后在某个主题下概率变得很大,所以它们会被一直淘汰。本文提出的模型不会在每批数据处理完之后进行单词淘汰,这样前文提出的这种类型的单词可能会累积概率,最后在某个主题下拥有很大的概率。虽然这种处理方式会造成词汇表越来越大,对内存造成较大的负担,但是主题单词分布矩阵不一定需要完全放在内存中,可以把主题单词分布矩阵放在外存中,在每次处理一批数据时,取出对应的需要计算的一部分数据。

4 实验分析

本文所提算法主要的应用场景是处理大数据或是流数据,故实验采用了两个数量级很大的语料 NYTIMES 和 PUBMED,设定几种不同的长度值,将大语料切分成若干批小语料(minibatch)。语料集的相关信息如表 2 所列,其中 D 代表语料集内的总文档数,W 代表语料中词汇表的长度,NNZ 表示语料的非零单词数。

表 2 数据集

数据集	D	W	NNZ
NYTIMES	269771	102660	62707275
PUBMED	7790000	141043	459283331

实验中的对比算法是传统的固定词汇表的在线 LDA 算法,具体是在线置信传播算法(Online Belief Propagation,OBP)[10,13]、在线吉布斯采样算法(Online Gibbs Sampling,OGS)[9]、在线变分推断算法(Online Variational Inference,OVB)[8]以及同样基于动态词汇表的无限词汇表算法(Infvoc-LDA)[5]。本文主要在混淆度、互信息指数以及算法收敛时间上进行对比,在混淆度的对比上没有对比 Infvoc-LDA,因为 Infvoc-LDA 算法不是以最大化单词的似然为优化目标的,并且会淘汰在主题下概率很小的单词。故用判断单词产生概率的混淆度指标来对比 Infvoc-LDA 算法是不公平的。而在互信息指数指标上对比了 Infvoc-LDA 算法。

实验中基于固定词汇表的 LDA 算法的先验参数初始化都设置为 0.01,在主题单词分布基于折棒构造的狄利克雷过程的 LDA 算法中 $\alpha=0.01$,$\alpha^\beta=1000$。在每批流数据中的文档数(*minibatch*)与混淆度的实验中固定主题数 $K=300$,在不同的主题数与混淆度的实验中固定 *minibatch*=1024。在互信息指数实验中固定 $K=100$,*minibatch*=1024。在计算算法复杂度的实验中固定 *minibatch*=1024,并且算法读入数据到内存的用时不计算在算法运行时间内。

4.1 评价指标

本文中使用了混淆度和互信息指数两个评价指标。

混淆度是测试 LDA 最常用的指标,用来评测单词在 LDA 输出的文档主题分布 θ_d 和主题单词分布 φ_k 下产生的概率似然大小,以此来评价 LDA 算法的建模好坏。混淆度的计算公式如下:

$$Perp=\exp\{-\frac{\sum_{w,d}x_{w,d}\log[\sum_k\theta_d(k)\varphi_w(k)]}{\sum_{w,d}x_{w,d}}\} \tag{16}$$

公式中的符号的意义与表 1 中的相同。本文测试混淆度时,会从完整的语料中抽出 10000 篇文档作为测试集,测试集不参与模型训练过程。实验采用训练集上训练出来的模型在测试集上计算混淆度。

互信息指数(Pointwise Mutual Information,PMI)[7]用于评测 LDA 算法产生的每个主题中概率最高的 n 个词的相关性。其中主题 k 的互信息指数的计算公式为:

$$PMI(k)=\sum_{i=2}^{n}\sum_{j=1}^{i-1}\log(\frac{Q(w_i^k,w_j^k)+\varepsilon}{Q(w_i^k)Q(w_j^k)}) \tag{17}$$

其中,w_i^k 表示的是主题 k 下概率第 i 高的单词,$Q(\cdot)$函数是在语料中寻找某个单词出现的文档的个数,$Q(\cdot,\cdot)$是在语料中寻找某两个单词一起出现的文档的个数,ε 为分子上的平滑项。实验中,n 设定为 10,把每个主题的互信息指数取平均值作为语料的 PMI 值。

4.2 模型精度

与固定词汇表之上的 LDA 算法相比,动态词汇表之上的 LDA 算法最大的优势是能够使得 LDA 算法更加贴近当前所处理的语料。表 3 模拟了在实际情况中会发生的问题。xxx_1 基于的词汇表与数据是匹配的,其中包含的词都是在数据中

出现的，这是最理想的情况。xxx_2 基于的词汇表模拟的是词汇表和数据不匹配的情况，在 xxx_1 中用的词汇表之上增加了一些数据中并没有出现过的单词。在 xxx_2 中用的词汇表所包含的词的数量是 xxx_1 所用的词的数量的两倍。其他参数设置相同。

表 3　基于固定词汇表的 LDA 算法在不同的词汇表上精度的表现

数据集	NYTIMES		PUBMED	
算法	K=100	K=500	K=100	K=500
OBP_1	4354.214	3107.765	3075.325	2765.232
OBP_2	**6241.35**	**5436.225**	**4924.22**	**4325.251**
OGS_1	4733.234	3423.854	3403.454	2810.565
OGS_2	**7125.32**	**6024.325**	**5321.59**	**4031.692**
OVB_1	4803.742	6104.12	5600.762	6209.101
OVB_2	**6254.85**	**7924.251**	**6425.65**	**7425.352**

从表 3 中可以看到当词汇表与处理的数据不匹配时，模型输出的混淆度在不同的 K 下都是上升的，这说明了词汇表的重要性；当词汇表与处理的数据比较匹配时，模型可以达到最优的精度。其中的原因可以从主题单词分布的狄利克雷分布归一化公式中找到一部分，当词汇表的长度较大时，分母上的平滑项 $W\beta$ 也会变得很大，这样就会使得主题单词分布趋向于均匀分布，进而造成模型精度下降。

图 3－图 6 中所有固定词汇表的 LDA 算法的词汇表都是最优词汇表，只包含所有在语料中出现的单词。图 3 和图 4 中固定 $K=300$，图 5 和图 6 中固定每批流数据中包含 1024 篇文档（$minibatch=1024$）。可以发现在不同的 $minibatch$ 以及不同的 K 下，dvOBP 都能达到最优的精度，因为固定词汇表的 LDA 算法是在最优的词汇表下运行的，所以在实际情况中，dvOBP 在混淆度上的优势会更加明显。

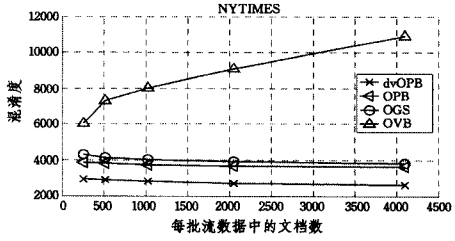


图 3　NYTIMES 在不同大小的流数据尺寸下的混淆度变化

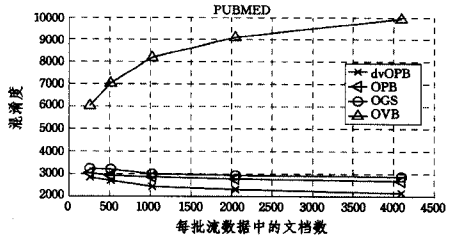


图 4　PUBMED 在不同大小的流数据尺寸下的混淆度变化

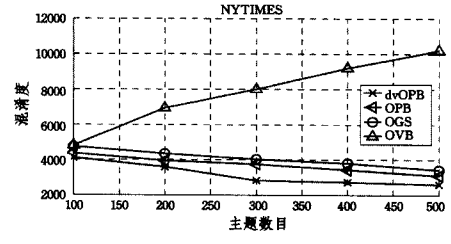


图 5　NYTIMES 在不同主题数下的混淆度变化

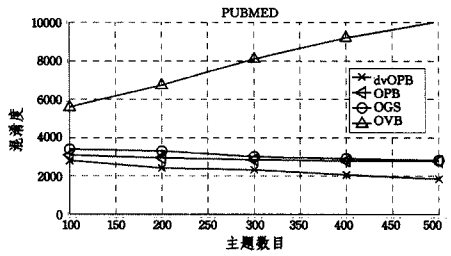


图 6　PUBMED 在不同主题数下的混淆度变化

4.3　互信息指数

表 4 是在 NYTIMES 和 PUBMED 下计算的互信息指数，从实验结果来看，在 NYTIMES 和 PUBMED 上基于动态词汇表的 dvOBP 和 Infvoc-LDA 的表现比基于固定词汇表的算法的表现更好。这说明了在主题单词分布上选用狄利克雷过程使得词汇表具有灵活性的方案是有效的，能使主题下的热词（概率最高的一部分词）的相关性更高。而 dvOBP 在 PMI 指数上表现得比 Infvoc-LDA 略好一些。这说明了在处理折棒构造中存在的位置偏置问题上，dvOBP 通过 $\hat{V}(k)$ 来给主题 k 单词分布下的每个单词从大到小排序，从而使得 $\hat{V}_w(k)$ 高的单词 w 拥有更高的先验这一处理方法比 Infvoc-LDA 中通过在一个英文字典上训练一个语言模型来确定每个单词的先验的方法要更优越。

表 4　LDA 算法在两个数据集上的 PMI 指数上的表现

数据集	NYTIMES	PUBMED
算法	K=100	K=100
dvOBP	**−7.1315**	**−4.012**
Infvoc-LDA	−7.21434	−4.2314
OBP	−8.2334	−6.242454
OGS	−8.102	−5.5332
OVB	−9.21313	−7.998

4.4　算法复杂度分析

表 5 是在 NYTIMES 和 PUBMED 两个数据集上计算的收敛时间。可以发现基于动态词汇表的 dvOBP 比在线置信传播算法和在线吉布斯采样算法慢一些，这是由于在 dvOBP 迭代完一遍语料时，需要对每个主题下的单词分布中的单词进行排序。而同样基于动态词汇表的 Infvoc-LDA 比 dvOBP 更加慢一点，这是由于 Infvoc-LDA 不仅仅在迭代完语料之后需要对主题分布下的单词进行排序，还需要在训练开始前训练一个 n 元主题模型来确定单词的先验。

表 5　LDA 算法在两个数据集上的运行时间（s）

数据集	NYTIMES		PUBMED	
算法	K=100	K=500	K=100	K=500
OBP	9998	60453	49522	245658
OGS	11786	62476	53276	279646
OVB	16986	127865	152625	845523
dvOBP	15367	115467	136543	786549
Infvoc-LDA	21654	216549	268904	1009890

在算法空间复杂度上，dvOBP 与在线置信传播算法一样，需要保存 $\mu_{w,d}^k$，故它们的空间复杂度是每个非零词属于每个主题的概率、文档主题矩阵和主题单词矩阵的累加，为 $O(K*(NNZ+D+W))$，其中 NNZ 为数据集中非零词的数目。在线变分推断算法和在线吉布斯采样都不需要保存

（下转第 134 页）

续的影响，因此建立了在不同影响滞差情况下，利用加权贝叶斯方法预测组织行为的模型。

参 考 文 献

[1] Serra E, Subrahmanian V S. A Survey of Quantitative Models of Terror Group Behavior and an Analysis of Strategic Disclosure of Behavioral Models[J]. IEEE Transactions on Computational Social Systems, 2014, 1(1): 66-88

[2] Subrahmanian V S. Handbook of Computaional Approaches to Counterterrorism[M]. New York: Springer Press, 2013: 99-268

[3] Xue An-rong, Wang Wei, Zhang Ming-cai. Terrorist Organization Behavior Prediction Algorithm Based on Context Subspace [M]// Advanced Data Mining and Applications. Springer Berlin Heidelberg, 2011: 332-345

[4] Li Xiao-chen, Mao Wen-ji, Zeng D, et al. Performance evaluation of machine learning methods in cultural modeling[J]. Journal of Computer Science and Technology, 2009, 24(6): 1010-1017

[5] Subrahmanian V S. Cultural Modeling in real time[J]. Science, 2007, 317(5844): 1509-1510

[6] Khuller S, Martinez V, Nau D, et al. Finding Most Probable Worlds of Probabilistic Logic Programs[C]// 1st International Conference on Scalable Uncertainty Management. Washington DC: IEEE Press, 2007: 45-57

[7] Subrahmanian V S, Albanese M, Martinez M V, et al. CARA: a cultural adversarial reasoning architecture[J]. IEEE Intel Syst, 2007, 22(2): 12-16

[8] Martinez M V, Simari G I, Sliva A, et al. CONVEX: context vectors as a similarity-based paradigm for forecasting group behaviors[J]. IEEE Intel Syst, 2008, 23(4): 51-57

[9] Martinez M V, Sliva A, Simari G I, et al. CAPE: automatically predicting changes in group behavior[C]// Mathematical Methods in Counterterrorism. Springer Vienna, 2009: 253-269

[10] Victor A, Pate A, Wilkenfeld J. Minorities at risk organization behavior data and codebook version 9[EB/OL]. http://www.cidcm.umd.edu/mar

[11] Guo Hong-yu. Research on weighted feature selection algorithm based on information entropy theory[J]. Computer Engineering and Applications, 2013, 49(10): 140-146(in Chinese)
郭红钰. 基于信息熵理论的特征权重算法研究[J]. 计算机工程与应用, 2013, 49(10): 140-146

[12] Liu Hua-wen. Research on feature selection algorithm based on information entropy[D]. Changchun: Jilin University, 2010(in Chinese)
刘华文. 基于信息熵的特征选择算法研究[D]. 长春: 吉林大学, 2010

[13] He Qing, Li Ning, Luo Wen-juan, et al. Summary of machine learning algorithms in large data[J]. Pattern Recognition and Artificial Intelligence, 2014, 27(4): 327-336(in Chinese)
何清, 李宁, 罗文娟, 等. 大数据下的机器学习算法综述[J]. 模式识别与人工智能, 2014, 27(4): 327-336

[14] Han Jia-wei, Kamber M. 数据挖掘：概念与技术(3rd ed)[M]. 范明, 等译. 北京: 机械工业出版社, 2012

(上接第 124 页)

$\mu_{w,d}^{k}$，故它们的空间复杂度都为 $O(K*(D+W))$。而 Infvoc-LDA 对于每个主题分布中的单词需要保存两个参数，空间复杂度为 $O(K*(D+2*W))$。由于都是在线算法，可以通过调整 *minibatch* 来控制 dvOBP 的空间复杂度，但是在 Infvoc-LDA 中有内存隐患，因为主题单词矩阵是全局变量，需要常驻内存。

结束语 本文首先提出了传统的 LDA 算法中固定词汇表的问题，然后提出了一种新的基于动态词汇表的 LDA 算法 dvOBP，并给出了详细的推导过程。之后将其与其他固定词汇表的传统 LDA 算法以及同样基于动态词汇表的 Infvoc-LDA 算法在混淆度、互信息指数以及收敛时间上进行比较，可以发现，dvOBP 在词汇表上的解决方案虽然存在着时间复杂度比传统的 LDA 算法偏高的问题，但是确实可以解决实际问题中固定词汇表所造成的问题，在混淆度、互信息指数上优于传统的 LDA 算法。

参 考 文 献

[1] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. The Journal of Machine Learning Research, 2003, 3(1): 993-1022

[2] Zeng J, Cheung W K, Liu J. Learning topic models by belief propagation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(5): 1121-1134

[3] Heinrich G. Parameter estimation for text analysis[R]. Technical Report, 2005

[4] Sethuraman J. A constructive definition of Dirichlet priors[R]. Florida State Univ Tallahassee Dept of Statistics, 1991

[5] Zhai K, Boyd-Graber J. Online Latent Dirichlet Allocation with Infinite Vocabulary[C]// Proceedings of The 30th International Conference on Machine Learning. 2013: 561-569

[6] Mimno D, Hoffman M, Blei D. Sparse stochastic inference for latent Dirichlet allocation[J]. arXiv, 2012(3): 362-365

[7] Newman S K D, Cavedon L. External evaluation of topic models [C]// Australasian Document Computing Symposium. 2012: 11-18

[8] Hoffman A F M, Blei D. Online inference of topics with latent dirichlet allocation[C]// NIPS. 2010: 856-864

[9] Yao L, Mimno D, McCallum A. Efficient methods for topic model inference on streaming document collections[C]// Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2009: 937-946

[10] Zeng J, Liu Z Q, Cao X Q. Online belief propagation for topic modeling[J]. arXiv preprint arXiv: 1210. 2179, 2012

[11] Ishwaran H, Zarepour M. Dirichlet prior sieves in finite normal mixtures[J]. Statistica Sinica, 2002, 12(3): 941-963

[12] Mei S Y, Wang F, Zhou S G. Dirichlet process mixture model, extensions and appication[J]. Chin Sci Bull, 2012, 57(34): 3243-3257(in Chinese)
梅素玉, 王飞, 周水庚. 狄利克雷过程混合模型、扩展模型及应用[J]. 科学通报, 2012, 57(34): 3243-3257

[13] Gong Sheng-rong, Ye Yun, Liu Chun-ping, et al. Topic Tracking Based on Online Belief Propagation[J]. Chinese Journal of Computers, 2015, 38(2): 249-260(in Chinese)
龚声蓉, 叶芸, 刘纯平, 等. 基于在线消息传递的主题追踪方法[J]. 计算机学报, 2015, 38(2): 249-260