## 2相关工作

在本节中，我们从以下两个角度简要总结相关工作：普通文本的主题模型和短文本的主题模型。

#### 2.1普通文本的主题模型

主题模型被广泛用于揭示文本语料库中的潜在语义结构。挖掘文本集合中的语义结构的工作可以追溯到潜在语义分析（LSA）[17]，它采用奇异值分解将文档投影到较低维的空间，称为潜在语义空间。概率潜在语义分析(PLSA)[6]使用基于从潜在类模型导出的混合分解的可靠概率模型改进了LSA。在PLSA中，文档被表示为主题的混合，而主题是单词的概率分布。扩展PLSA，潜在狄利克雷分配(LDA)[7]为特定于文档的主题混合添加了狄利克雷先验，从而可以生成看不见的文档。LDA由于其良好的泛化能力和可扩展性，在文本挖掘领域取得了巨大的成功。

在过去的十年中，主题模型得到了广泛的研究。标准LDA模型的许多复杂变体和扩展已经被提出，这可以在综合综述中找到[18]。这里我们只列出一些与我们密切相关的工作。Wallach[19]通过将二元统计纳入主题建模，提出了扩展LDA的二元主题模型，但其细节与我们的有很大不同。二元主题模型旨在通过利用文档级顺序模式来捕获单词之间的顺序依赖关系（在普通文本中），而我们的模型是专门为短文本设计的，并试图通过利用语料库级单词共生来捕获单词之间的语义依赖关系。发生模式。此外，最近提出的两个模型，即正则化主题模型[20]和广义Polya模型[21]，都具有利用单词共现（即biterm）统计来增强主题学习的相同想法。然而，它们都只是使用单词共现信息作为先验来指导单词的生成，而不是直接对共现进行建模。最重要的是，所有这些模型都只处理普通文本，而没有考虑短文本的特殊性。

#### 2.2短文本主题模型

早期的研究主要集中在利用外部知识来丰富短文本的表示。例如，Phan等人[22]基于在另一个用于短文本分类的大规模数据集上估计的传统主题模型来推断短文本的主题。金等人[23]提出了一种基于LDA的模型，联合学习短文本和相关长文本的主题。期望利用从长文本中学到的主题知识来帮助完成短文本的主题学习任务。然而，这些方法只有在辅助数据与原始数据密切相关时才有效。有时，找到此类辅助数据可能会很昂贵，甚至是不可能的。相比之下，我们的模型仅依赖于语料库内单词共现的统计，这与上述模型是互补的。因此，在未来的工作中将它们结合在一起是有希望的。

近年来，随着社交媒体的出现，主题模型已被用于各种任务中的社交媒体内容分析，例如内容表征[1]，[2]，事件跟踪[5]，内容推荐[24]，[25]]，以及有影响力的用户预测[4]。然而，由于缺乏针对短文本的特定主题模型，一些研究人员直接应用常规（或稍加修改）的主题模型[1]，[26]。其他一些人尝试根据一些附加信息将短文本聚合成冗长的伪文档，然后训练传统的主题模型[4]，[2]。洪等人。[10]对Twitter中的主题建模进行了全面的实证研究，并提出需要新的短文本主题模型。

在我们之前的工作中，我们发现全局单词共现有助于短文本聚类[27]和主题学习[28]、[29]。本文扩展了我们之前的会议文章[29]，并进行了以下改进。1）我们引入了两种BTM在线算法来处理大规模数据集。2）对两种在线算法的能力进行了实证验证。3）进行了更全面的实验，并报告了新的发现。

## 3BITERM主题模型

在大多数主题模型中，主题被表示为相关单词组，其相关性基本上通过文档中的单词共现模式来揭示。例如，一旦观察到“ipad”和“iphone”这两个词经常同时出现，人们就可以看出它们的含义很接近，并且可能属于同一主题，即使他/她不知道确切的含义其中。传统的主题模型利用单词共现模式，通过对每个文档中单词的生成进行建模，以隐式方式揭示语料库的潜在语义结构。这些方法对文档的简短程度很敏感，因为单个短文档中的单词共现模式是稀疏且不可靠的。相反，如果我们聚合语料库中所有单词的共现模式，它们的频率会更稳定，更清楚地揭示单词之间的相关性。带着这个想法，我们开发了双项主题模型，它采用一种新颖的方式通过直接建模单词共现模式的生成来揭示语料库中的潜在主题组件。

#### 3.1双项提取

在详细介绍模型之前，我们首先介绍“biterm”的表示法，它表示在短上下文中同时出现的无序单词对（即单词共现模式的实例）。这里，短上下文是指术语序列上的一个小的、固定大小的窗口。在文档长度有限的短文本中，例如推文和短信，我们可以简单地将每个文档作为单独的上下文单元。在这种情况下，文档中的任何两个不同的单词都会构成一个双项。例如，包含三个不同单词的文档将生成三个双项：



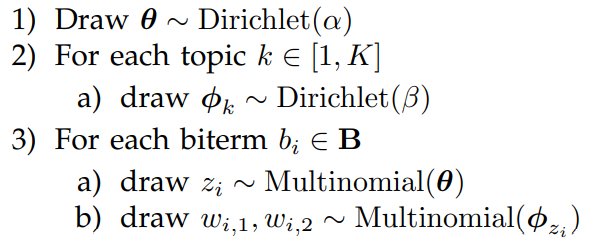
其中(·，·)是无序的。在提取每个文档中的双术语后，整个语料库现在变成了双术语集。双项提取过程可以通过对文档的单次扫描来完成。

#### 3.2型号说明

与大多数通过对文档的生成进行建模来学习语料库中的潜在主题组件的主题模型不同，BTM通过对双项的生成进行建模来执行此任务。关键思想是，如果两个单词同时出现的频率更高，则它们更有可能属于同一主题。基于这个想法，我们假设双项中的两个单词是独立于主题抽取的，其中主题是从整个语料库的主题混合中采样的。

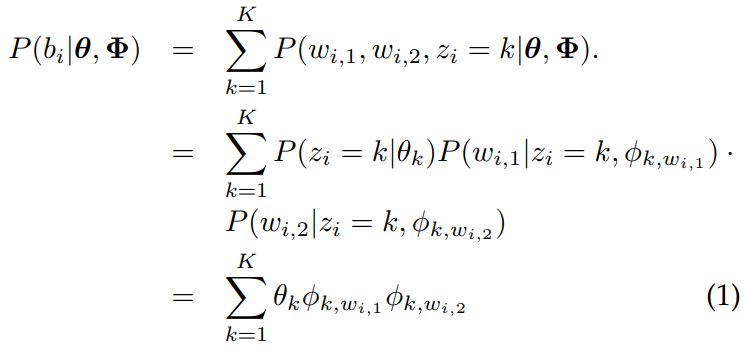
给定一个包含ND文档的语料库，假设它包含NB位术语，其中，并且通过词汇表中的W个唯一单词表达K个主题。令z∈[1，K]为主题指示变量，我们可以用K维多项分布来表示语料库中主题的流行程度（即P(z)），其中且。主题的单词分布（即P(w|z)）可以用K×W矩阵Φ表示，其中第k行是W具有条目且的维多项分布。

遵循LDA[30]的约定，我们对θ和使用对称狄利克雷先验，分别具有单值超参数α和β。形式上，BTM的生成过程描述如下。

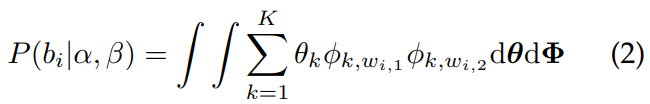


其图形表示如图1(c)所示。请注意，为了简单起见，我们假设双项是独立生成的。

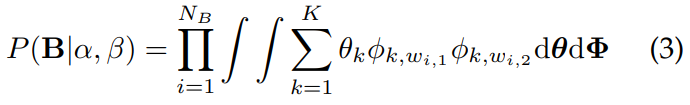
按照上述过程，我们可以写出以模型参数θ和Φ为条件的双项bi的概率：



给定超参数α和β，我们可以通过对θ和Φ进行积分来获得bi的概率：



将单个双项的概率乘积，我们得到整个语料库的概率：



#### 3.3型号比较

为了更好地理解BTM的独特性，我们将其与两种典型的主题学习模型进行比较，即LDA[7]和一元词混合模型[12]。在文献中，LDA和一元语法的混合都被用于短文本的主题发现[1]、[2]、[31]、[26]。图1显示了三个模型的图形表示。

如图1（a）所示，LDA对文档d的生成进行建模，如下所示：对于d中的每个单词，我们首先从文档特定主题分布θd中绘制一个主题z，然后从主题z中绘制一个单词w。从图中我们可以看出，单词w的主题z通过共享主题分布θd来依赖于同一文档中的其他单词。因此，LDA过度依赖文档级上下文来推断z和θd。当文档较短时，LDA容易受到数据稀疏问题的影响，导致z和θd的估计较差，进而损害主题词分布Φ的学习。

如图1(b)所示，混合一元组也对每个文档的生成进行建模，但方式不同。它假设文档中的所有单词共享相同的主题z，而z是从全局主题分布θ中采样的。换句话说，它将整个语料库而不是文档建模为主题的混合体。通过利用整个语料库的信息，它缓解了短文本主题推理中的稀疏问题。然而，文档具有单个主题的约束过于严格（我们知道即使是短文档也可能包含多个主题），这妨碍了它对文档中的精细主题进行建模。

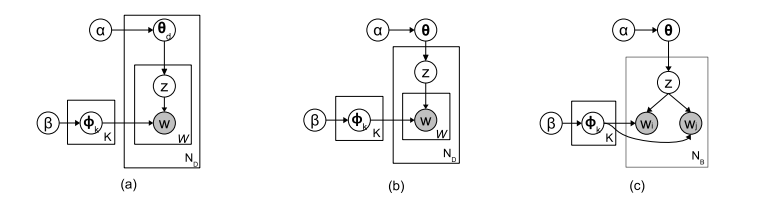


图1.(a)LDA、(b)一元组混合和(c)BTM的图形表示。图中的每个节点表示一个随机变量，其中阴影表示观察到的变量。板表示其中模型的复制品。重复次数在板的右下角给出。

总之，LDA和一元词混合的主要问题在于对短文档建模不当。对于如此极其稀疏的数据，很难直接建模和推断单个短文档中的潜在主题。然而，我们认为没有必要为语料库中的主题发现对文档进行建模。如图1(c)所示，BTM只是选择了另一种方式来通过对双术语（而不是文档）的生成进行建模来发现主题。与LDA相比，BTM通过学习全局主题分布θ来避免数据稀疏问题，就像一元词混合一样。同时，通过将每个文档分解为双词，并为每个双词分配一个主题，BTM允许一个文档（具有多个双词）能够展示多个主题，克服了一元词混合的缺陷。

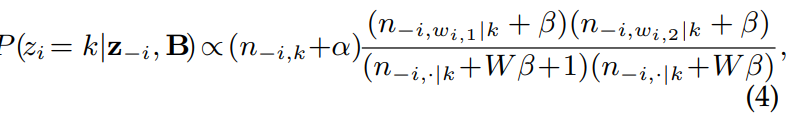
## 4参数估计

在本节中，我们描述了BTM中估计参数{Φ，θ}的算法，并将其复杂度与LDA进行了比较。

#### 4.1吉布斯采样算法

与LDA类似，通过最大化等式中的似然来精确求解耦合参数θ和Φ是很困难的。（3）。按照[30]，我们使用吉布斯采样[14]对θ和Φ进行近似推断，该采样使用从潜在变量的后验分布中抽取的样本来估计参数，这些样本依次以所有其他变量和数据的当前值为条件。

在BTM的设置中，需要估计三类变量（即z的主题分配、多项分布参数Φ和θ）。但是使用折叠吉布斯采样技术[32]，由于使用了共轭先验，可以将Φ和θ积分出来。因此，对于bittermbi，我们只需要根据以下条件分布对其主题zi进行采样（补充材料中提供了推导）：



其中z−i表示除bi之外的所有位项的主题分配，是分配给主题k（不包括bi）的位项数量，是单词w分配给主题k的次数（不包括bi，且。等式的右手。(4)相当直观：第一部分与语料库中主题k的概率成正比，第二部分表示主题k下wi，1和wi，2的概率的乘积。

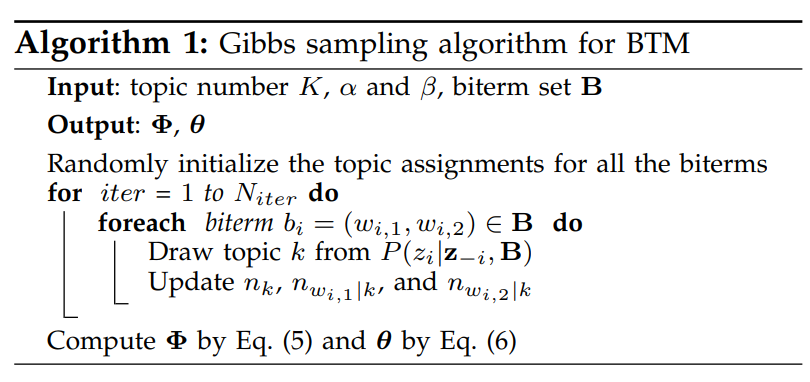
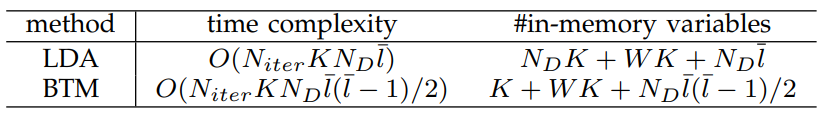
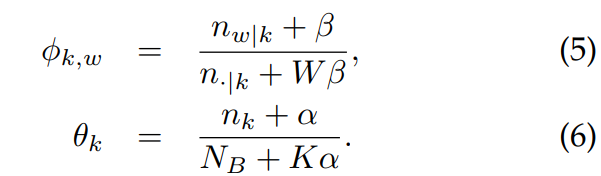


表1LDA和BTM中的时间复杂度和内存变量数量



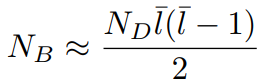
我们总结了算法1中吉布斯采样的整体过程。首先，我们为每个双项随机分配一个主题作为初始状态。在每次迭代中，我们通过检查等式更新每个双项的主题分配。(4)依次。经过足够次数的迭代后，我们计算每个主题k中的双项数量，用nk表示，以及每个单词w分配给主题k的次数，用nw|k表示。这些计数用于估计Φ和θ，如下（补充材料中提供了推导）：



#### 4.2复杂性分析

现在我们比较BTM吉布斯采样算法与LDA的运行时间和内存需求。我们在表1中列出了LDA和BTM的吉布斯采样过程的时间复杂度和内存变量的数量，其中表示平均单词数。

具体来说，两种算法中主要耗时的部分是主题分配的条件概率的计算，这需要O(K)时间。请记住，LDA为每个出现的单词绘制一个主题，给出总体时间复杂度。相反，BTM为每个双项绘制一个主题，总时间复杂度为。请注意，具有¯l个不同单词的文档将生成l(l−1)/2个双项，我们大致有（这里我们简单地假设文档具有几乎相同数量的不同单词。由于文档非常短，因此对于短文本来说这是合理的。）



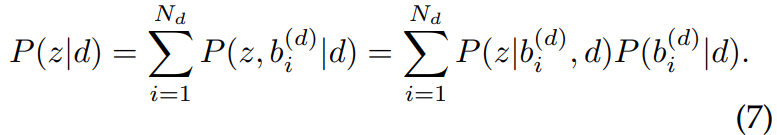
我们可以看到BTM的时间复杂度约为LDA的(¯l−1)/2倍。对于短文本，由于文档的平均长度非常小，例如Tweets2011集合中的¯l=5.21，因此BTM的运行时间仍然与LDA相当。

在两种吉布斯采样算法中，需要缓存的变量是计数和主题作为符号。在LDA中，我们需要维护计数（文档d中分配给主题k的单词数），（单词w分配给主题k的次数），以及每个单词出现的主题分配[33]，总共变量，内存中。在BTM中，我们需要将计数、以及每个双项的主题分配（总共K+WK+NB变量）保存在内存中。与BTM相比，我们可以看到，当ND和K变大时，LDA的内存成本会迅速增加，使其内存效率低于BTM。我们将在实验部分进一步演示它。

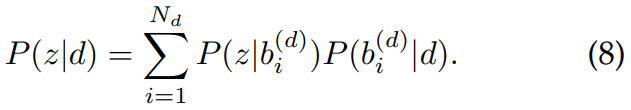
5推断文档中的主题

除了学习主题组件（即）之外，主题模型中的另一个常见任务是推断文档中的主题，即评估文档d的主题后验P(z|d)。然而，由于BTM不对文档进行建模，我们无法直接从估计模型中获得P(z|d)。幸运的是，我们可以通过bitterm的主题推导出文档的主题比例。

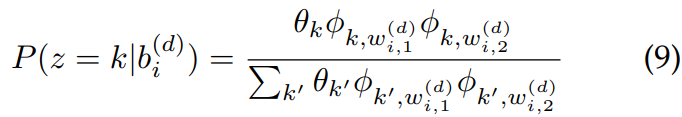
假设d包含Nd个位项，，使用链式法则



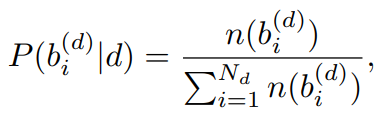
给定双项，我们假设其主题z有条件独立于d，即。那么，我们可以简化上面的方程：



在等式中。(8)可以根据BTM中学习到的参数通过贝叶斯公式计算：



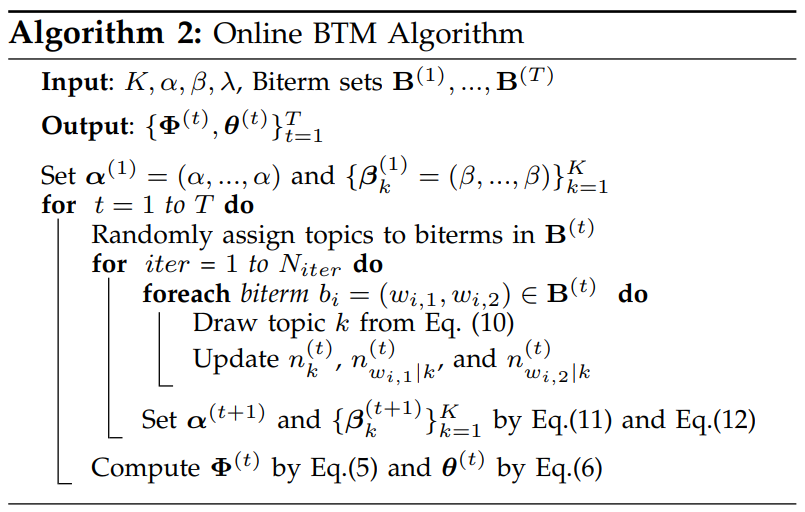
同时，可以根据经验估计：



其中是d中双项的频率。

## 6BTM在线算法

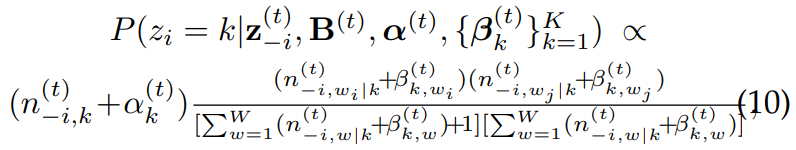
在微博等现实应用中，短文本通常数量巨大、源源不断，并且随着时间的推移而快速增长。在这种情况下，批处理算法不再适合主题学习。首先，由于内存的限制，重复扫描整个数据集是不切实际的。其次，当新数据不断到达时，希望使模型保持最新。由于这些原因，我们引入了两种BTM在线算法，称为在线BTM(oBTM)和增量BTM(iBTM)。在线算法只需要动态存储一小部分数据来进行模型更新，这比大规模数据集上的批处理算法要高效得多。



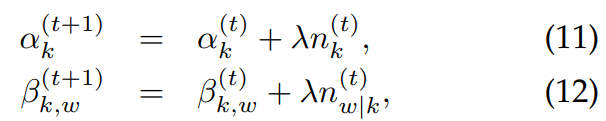
#### 6.1在线BTM算法（oBTM）

oBTM算法受到[34]中提出的在线LDA算法的启发，该算法假设文档按时间片（例如一天）划分，并且文档在时间片内是可交换的。oBTM的主要思想是在时间片t中的数据上拟合BTM模型，并使用当前时间片中的计数和来调整下一个时间片的狄利克雷超参数时间片。oBTM的总体流程如算法2所示

在运行oBTM之前，我们需要将时间片t中的文档转换为双项集B(t)。令α(t)为θ(t)的K维狄利克雷超参数，为的W维狄利克雷超参数。我们通过设置α(1)=(α，...，α)和β(1)k=(β，...，β)使用对称狄利克雷分布作为初始先验。给定α(t)和，我们根据条件分布迭代地绘制每个双项bi∈B(t)的主题分配：



迭代完成后，我们获得计数和，并利用它们来调整超参数和时间片t+1通过设置：

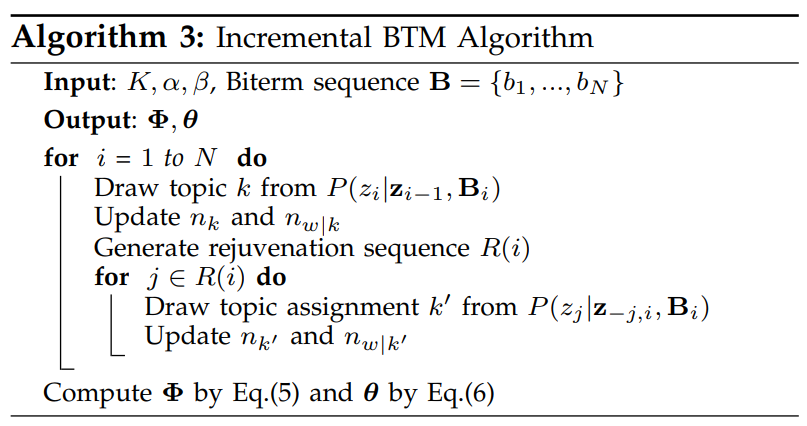


其中λ∈[0，1]是衰减权重。如[35]、[34]中所述，凭借狄利克雷多项式共轭性质，超参数和可以被视为先前观测值的计数和，分别。因此，方程（11-12）可以解释为将历史主题分配作为下一个时间片的先前观察。此外，衰减权重λ控制历史主题分配的影响强度。如果λ=0，则不同时间片训练的模型完全独立；如果0<λ<1，则历史影响力会随着经过的时间片数量呈指数衰减；如果λ=1，主题分配的历史计数只是简单累积，没有任何衰减。

通过对每个时间片中的数据顺序运行批量吉布斯采样器，oBTM简单且易于实现。然而，在某些应用中，例如微博中的实时主题跟踪，希望在新文档到达时立即更新模型。在这种情况下，oBTM无能为力。因此，我们转向另一种更适合此类任务的在线算法

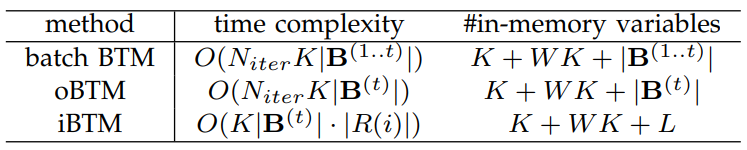
#### 6.2增量BTM算法（iBTM）

iBTM不断更新模型，即通过一种称为增量吉布斯采样器的技术，每当双项到达时立即更新参数Φ和θ[16]。具体来说，当bitermbi到来时，iBTM分两步更新模型。首先，我们从中得出bi的主题分配，其中表示之前的所有主题分配，。其次，我们随机选择一些先前的双项来构建双项序列，称为再生序列R(i)，以对它们的主题分配进行重新采样。对于每个双项bj∈R(i)，我们从重新采样其主题分配zj。iBTM的流程如算法3所示。



模型更新中不同时间收到的双项数。例如，我们可以从之前的双项的衰减分布（例如指数和逆多项式分布[36]）中选择R(i)中的条目，以支持更新的历史数据。在这项工作中，R(i)是根据覆盖最近双项的固定大小滑动窗口上的均匀分布生成的。这种方法不仅通过存储一小部分历史数据来减少内存和时间成本，而且由于它不断更新模型，因此模型比oBTM对数据中主题的动态变化更加敏感。

表2时间片t内批量和在线BTM算法的时间复杂度和内存变量数量



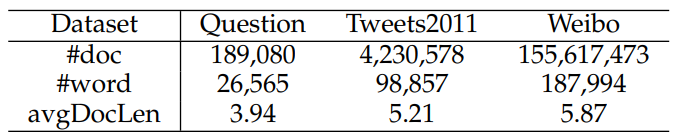
#### 6.3复杂度比较

在线算法相对于批处理算法的一个主要优点是它们可以很好地扩展到海量数据集，因为它们只需要存储一小部分数据来进行模型更新。通过分析时间复杂度和内存消耗，我们可以更清楚这一点。为了便于比较，我们假设语料库按T个时间片组织，然后估计时间片t内模型更新的运行时间和内存需求。

批量BTM需要运行时间片t之前观察到的所有双项（即，花费时间，并且必须记录内存中的变量，其中表示集合中元素的数量。相反，oBTM仅在当前时间片中迭代运行双项集B(t)，时间复杂度为和内存中的变量。对于iBTM，它单次运行在B(t)上，但与|R(i)|一起运行。每个双项的重采样次数。因此其时间复杂度为，内存变量数量为K+WK+L，其中L是滑动窗口的长度。请注意在大型数据集上。

表2总结了批量BTM算法、oBTM和iBTM的时间复杂度和内存消耗。我们可以看到，批量BTM算法的时间和内存成本随着t的增长而线性增加。相反，两种在线BTM算法需要几乎恒定的时间和内存成本来更新模型，因为每个时间片中的位项数量通常是稳定的。因此，这两种在线算法可以通过增量处理数据来有效地处理大规模数据集

表3三篇短文集的摘要



## 7实验

在本节中，我们实证评估BTM的有效性和效率，包括批量BTM算法和在线BTM算法

#### 7.1实验设置

数据集。为了展示我们的方法在不同短文本数据集上的有效性，我们使用三个短文本集合进行评估。

问题集包括从热门中文问答网站3抓取的648，514个问题。在此集合中，每个问题都有一个由其作者从35个类别中选择的标签

Tweets2011合集是TREC2011微博track4上发布的标准短文本合集，提供了2011年1月23日至2月8日期间采样的大约1600万条推文。除了内容之外，每条推文还包含一个用户ID和一个时间戳。

微博合集是2011年8月1日至2012年7月31日期间从weibo.com收集的微博的子集。原始数据量约为600G。

这些集合的原始数据非常嘈杂。为了进行预处理，我们删除了无意义的单词，例如停用词、低频词以及非拉丁语或中文的字符。为了过滤掉低质量的文档，我们删除了重复文档和单个单词的文档。表3列出了预处理后三个集合的文档和不同单词的数量以及文档的平均长度（即单词数）。

我们首先评估Question和Tweets2011集合上的批处理BTM，因为微博集合太大，运行批处理算法的成本非常昂贵。然后，我们使用Tweets2011和Weibo集合来检查两种在线算法的性能，因为它们具有帖子时间戳，并且可以视为文本流

基线方法。对于批量BTM，我们将其与当今短文本主题建模的三种典型方法进行了比较：

Mix表示一元模型的混合，它假设每个文档仅展示一个主题。