**文章编号**：1003-0077(2018)07-0116-12

# 在线技术社区的用户技能与兴趣发现

张东雷，林友芳，万怀宇，马语丹，陆金梁

（北京交通大学 计算机与信息技术学院，北京 100044）

**摘　要**：在线技术社区是技术爱好者或者从业者进行技术交流、咨询和分享的重要平台。社区运营者如果能够准确掌握每个用户的技能和兴趣，对用户进行画像，将有助于为用户提供精准的推荐和个性化服务，从而增加用户的黏性和社区的活跃度。考虑到社区用户既是内容的生产者（作者）又是内容的消费者（读者），生产者体现用户技能，消费者体现用户兴趣，从而提出了一种作者—读者—话题（author-reader-topic，ART）模型，同时对用户的技能和兴趣进行建模。该模型可以将文档的作者和读者关联起来，因而能够提升话题的聚集效果，产生更准确的作者话题分布和读者话题分布。该文基于 CSDN 技术社区的真实数据集进行了实验对比和分析，实验结果表明，该文提出的 ART 模型能够有效地发现用户的技能和兴趣，明显优于现有的各种话题模型。

**关键词**：在线技术社区；用户画像；用户技能；用户兴趣

**中图分类号**：TP391　　**文献标识码**：A

## Discovering Users's Expertises and Interests in Online Technology Communities

ZHANG Donglei, LIN Youfang, WAN Huaiyu, MA Yudan, LU Jinliang

(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

**Abstract**: Online communities are important platforms for technology enthusiasts or practitioners to exchange and share information. To capture the expertise and interests of each user simultaneously, this paper proposes an Author-Reader-Topic (ART) model based on the fact that a user in a community is both a producer (author) and a consumer (reader) of contents. By linking the authors and the readers of the documents, this model can improves the topic clustering, and achieves more accurate author topic distribution and reader topic distribution. We conducted an experimental comparison and analysis based on a real data set collected from the CSDN community. The experiments show that the proposed model can effectively discovery users' expertises and interests, outperforming the existing methods significantly.

**Key words**: online technology community; user profiling; user expertise; user interest

## 0　引言

作为一种集成化的信息与知识传播和共享服务平台，在线技术社区为用户提供了技术交流、咨询和共享空间，深受技术爱好者和从业者的青睐。用户可以在社区发表博客或帖子来记录或分享自己对某一问题的经验或看法，可以浏览或收藏自己感兴趣的内容，可以针对自己的疑问提出咨询，也可以参与相关话题的讨论。例如，全球最大的中文 IT 技术社区 CSDN，拥有数千万用户，每天产生大量的博客和帖子，以及浏览、顶踩、评论、收藏等行为。准确地了解和掌握每个用户的技能和兴趣，对用户进行准确的画像，对技术社区的运营者来说十分重要，有助于他们为用户提供精准推荐和个性化服务，从而增加用户的黏性和社区的活跃度。例如，社区可以根据用户的兴趣为其推荐内容、好友、活动信息、技术专家等，也可以根据用户的技能为其推荐合适的工作机会。然而，社区中通常只有少部分用户提供了自定义的技能标签或兴趣标签，而且标签的可信度也存疑。因此，如何基于用户产生的内容和行为信息，准确地发现用户的技能和兴趣，就显得尤为

**收稿日期**：2017-11-24　**定稿日期**：2017-12-16

**基金项目**：国家自然科学基金(61603028)

必要。

以用户技能(或兴趣)建模为目的的文本挖掘近年来受到了研究者的广泛关注[1-3]，涌现出了大量的相关模型，这些模型大致可以分为有监督、无监督和半监督的用户技能/兴趣建模。有监督和半监督的建模方法[4-5]需要利用训练语料来训练生成文本分类器，进而对用户进行分类，一般具有较高的准确率，然而获取训练样本的昂贵代价极大地限制了此类方法的可应用性。因此以 LDA[6]、AT[7]、CAT[8]和 ACT[9]等话题模型为代表的无监督用户技能/兴趣建模方法近年来受到更多的关注。但是，当前这些模型主要考虑从用户发表的文章来对其技能或兴趣进行建模，没有将用户的技能和兴趣区别开来，因此还不能更准确地同时捕获用户的技能和兴趣。

事实上，社区用户既是社区内容的生产者，又是消费者。生产者是指用户以作者身份发表内容，主要体现了用户的技能；消费者是指用户以读者身份去阅读、顶踩、评论和收藏各种内容，主要体现了用户的兴趣。通常情况下，用户的技能比较集中，而用户的兴趣则相对宽泛。基于这一假设，本文提出了一种作者—读者—话题(author-reader-topic，ART)模型，来同时对用户的技能和兴趣进行建模。该模型在经典的 LDA 模型基础上增加了作者和读者信息，在建模文档生成过程中，同步建模作者的话题分布和读者的话题分布。该模型可以捕获文档的作者和读者之间的关联关系，因而能够进一步提升话题的聚集效果，从而产生更准确的作者话题分布和读者话题分布。

我们采用吉布斯采样的方法对 ART 模型进行推导和求解，通过不断地采样语料库中每个词的主题指派和读者指派来近似推断语料库中主题、词和读者的联合分布。吉布斯采样收敛后，我们就可以根据语料库中每个词的采样结果来估计出作者话题分布和读者话题分布。

我们从 CSDN 技术社区采集了一个真实数据集进行了实验，并跟已有的用户技能/兴趣发现方法进行了对比和分析。实验结果表明，本文提出的 ART 模型能够更有效地发现用户的技能和兴趣，明显优于现有的各种话题模型。

本文的主要贡献包括以下两点：

(1) 将在线技术社区用户的技能和兴趣区别开来，并根据用户作为内容的生产者和消费者两种角色，提出了一种同时对用户技能和兴趣进行建模的话题模型。

(2) 从 CSDN 社区采集了一个高质量的真实数据集，对提出的模型进行了大量的实验，并通过案例分析和各种评价指标验证了本文提出的模型的有效性。

本文的剩余部分组织如下：第一节简要介绍相关工作；第二节详细描述我们提出的模型；第三节进行实验对比和结果分析；最后在第四节对全文进行总结。

## 1　相关工作

以用户技能(或兴趣)建模为目的的文本挖掘近年来受到了研究者的广泛关注[1-3]，涌现出了大量的相关模型。早期研究者主要利用有监督或半监督的模型来挖掘用户的技能或兴趣，取得了较高的准确率。例如，He 等人[5]通过形式概念分析技术从正例文档中建立用户兴趣模型，并采用形式化的概念来代表用户感兴趣的话题。Yang 等人[4]考虑到 Twitter 用户发表微博的周期性模式，提出了利用时间序列把用户微博进行归类，将微博内容转化为时间序列特征，采用时间序列的分类方法对 Twitter 用户的兴趣进行分类，和传统的基于文本特征对用户兴趣进行分类的方法相比，取得了较高的分类准确率。然而，有监督和半监督的建模方法需要大量的标注语料来训练分类器，虽然具有较高的准确率，但是获取训练样本的昂贵代价极大地限制了此类方法的可应用性。

以话题模型为代表的无监督方法避免了获取训练样本的昂贵代价，因此在文本挖掘领域得到广泛的研究和使用。LDA(latent dirichlet allocation)模型[6]是最经典的话题模型之一，由 Blei 等人于 2003 年提出，它采用了“词袋”假设，即忽略一篇文档的词序、语法和句法，仅仅将其看作是一个词集合。LDA 是一种层次式的贝叶斯模型，其核心思想是将文档看作隐话题的分布，而将每个隐话题看作词的分布。由于 LDA 具有良好的数学基础和灵活的可拓展性，目前国内外已有大量的研究者基于 LDA 及其拓展模型来对用户技能或兴趣进行建模。Weng 等人[10]将每个用户发表的 tweets 融合起来，并使用 LDA 来发现用户感兴

趣的话题。Rosen-Zvi 等人[7]提出了作者—话题(author-topic,AT)模型,将文档的作者信息加入话题的建模过程中,同时对作者和话题进行建模,实验表明加入作者先验信息可以增强话题的聚集效果,从而有效地计算作者感兴趣的话题分布。Hong 等人[11]将 AT 模型应用到 Twitter 上,结果表明作者信息的加入有助于对 Twitter 用户的兴趣话题进行建模。Xu 等人[12]认为用户发表的 tweets 并不都能体现用户的兴趣,通过在 AT 模型中引入一个隐变量来指示一篇 tweet 是否和用户的兴趣相关,据此提出的 twitter-user 模型在发现用户兴趣上要优于 AT 模型。Li 等人[13]提出用户—话题(user-topic,UT)模型对微博中的用户兴趣进行建模,按照微博生成机制的不同将用户的兴趣分为原创兴趣和转发兴趣,分别对应用户的原创博文和转发博文,实验表明该模型发现的用户兴趣涵盖的范围更全面、更准确。Tu 等人[8]在 AT 模型的基础上增加论文的引用作者信息,提出了引用—作者—主题(citation-author-topic,CAT)模型,来更好地刻画作者的技能分布,从而服务于专家发现。Tang 等人[9]在 AT 模型的基础上增加了出版地信息,提出了作者—会议—主题(author-conference-topic, ACT)模型,来更好地对学术领域中的作者、话题和出版地进行建模,进而用于学术领域专家发现,取得了比其他话题模型更好的效果。

上述模型在分析用户技能(或兴趣)方面仍然存在一些不足之处。这些模型都是从内容生产者的角度考虑了用户发表或者转发的文档,而没有站在内容消费者的角度考虑用户阅读、评论或收藏的文档。事实上,用户生产的内容更多地反映了用户的技能,而用户消费的内容更多地反映了用户的兴趣。通常情况下,用户的技能比较集中,而用户的兴趣则相对比较分散。因此,本文综合考虑文档的作者信息和读者信息,提出了一种新颖的作者—读者—话题(ART)模型来同时对在线技术社区用户的技能和兴趣进行建模。

## 2 用户技能及兴趣发现

本节首先简单介绍两个关于用户技能/兴趣建模的基础模型 LDA 模型和 AT 模型,然后将详细描述本文提出的作者—读者—话题(ART)模型,并对模型进行推导。表 1 列出了本文主要使用的符号及其含义说明。

**表 1 相关符号说明**

| 符号 | 含义 |
|---|---|
| $D$ | 文档数 |
| $T$ | 话题数 |
| $A$ | 作者数 |
| $R$ | 读者数 |
| $V$ | 文档词库的词数 |
| $N_d$ | 第 $d$ 篇文档的词数 |
| $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\eta}$ | Dirichlet 超参数 |
| $\boldsymbol{\theta}$ | 作者(或文档)—话题分布 |
| $\boldsymbol{\phi}$ | 话题—词分布 |
| $\boldsymbol{\varphi}$ | 话题—读者分布 |
| $a_d$ | 第 $d$ 篇文档的作者 |
| $\boldsymbol{r}_d$ | 第 $d$ 篇文档的读者集合 |
| $\boldsymbol{z}$ | 语料库中所有词的话题指派 |
| $\boldsymbol{w}$ | 语料库中的所有词 |
| $\boldsymbol{x}$ | 语料库中所有词的读者指派 |
| $\boldsymbol{a}$ | 语料库中的作者集合 |
| $\boldsymbol{r}$ | 语料库中的读者集合 |
| $z_{dn}$ | 第 $d$ 篇文档第 $n$ 个词的话题指派 |
| $x_{dn}$ | 第 $d$ 篇文档第 $n$ 个词的读者指派 |
| $w_{dn}$ | 第 $d$ 篇文档中的第 $n$ 个词 |

### 2.1 基础模型

#### 2.1.1 LDA 模型

LDA 模型[6]是一种层次式的贝叶斯概率模型,包含词、话题和文档三层结构,语料库中的每篇文档被建模为隐话题的多项式分布,每个话题又被建模为词的多项式分布,每篇文档中的每个词都是通过“以一定概率选择了某个话题,并从这个话题中以一定概率选择某个词”这样一个过程得到。LDA 模型的盘式表示如图 1 所示。为了生成一篇文档,首先根据文档的话题分布采样生成一个话题,然后根据该话题的词分布采样生成一个词。重复上述过程直到文档中所有词均已生成。由于 LDA 模型中存在隐变量,直接求解模型参数非常困难,因此 LDA 模型的推导一般采用变分法或吉布斯采样进行近似推断[6,14]。

采用 LDA 进行用户技能或兴趣发现时,在模

型中不考虑文档的用户信息，而是在求解模型得出每篇文档的话题分布之后，对每个用户对应的全部文档的话题分布求平均，形成用户的话题分布，进而根据用户—话题分布和话题—词分布生成用户的技能表示或兴趣表示。

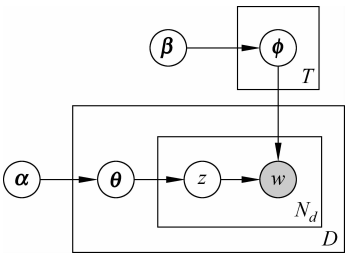


图 1　LDA 模型的盘式表示

2.1.2　作者—话题模型

作者—话题（author-topic，AT）模型[7]是对 LDA 模型的一种拓展变形，是一种较新颖的话题模型，它包含词、话题、文档和作者四层结构，在建模过程中加入了文档的作者信息。该模型假设语料库中的每个作者都对应一个隐话题的多项式分布，每个话题都对应一个词的多项式分布。AT 模型与 LDA 模型的不同之处在于它用作者—话题分布替换了文档—话题分布，并且每个词对应两个隐变量，即话题和作者。AT 模型的盘式表示如图 2 所示，其文档生成过程与 LDA 的区别在于，它首先从文档的作者集合中随机选择一个作者，然后根据该作者的话题分布采样生成一个话题，最后再根据该话题的词分布采样生成一个词。重复上述过程直到文档中所有词均已生成。同样，AT 模型的推导通常也采用变分法或吉布斯采样进行近似推断。

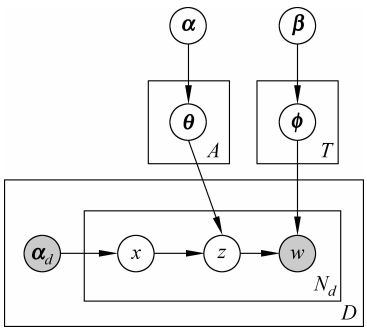


图 2　AT 模型的盘式表示

AT 模型由于在建模过程中加入了作者信息，通常一个作者的话题分布是比较稳定的，因此有助于增强话题的聚集效果。得益于作者信息的加入，相比 LDA 模型，AT 模型能够更准确地发现用户的技能分布。AT 模型虽然引入了文档的作者信息，但没有考虑文档的读者信息对话题的聚集效果，以及进一步的提升作用。此外，用户分别作为作者和读者时，其话题分布也是不一样的。作为作者时对应的是其技能的话题分布，作为读者时则对应其兴趣的话题分布。因此，本文进一步提出了作者—读者—话题模型，对用户的技能和兴趣分别进行建模。

## 2.2　作者—读者—话题模型

2.2.1　模型描述

在技术社区中，用户经常将自己的技术知识和经验以博客或者帖子的形式发表出来，供其他用户学习、参考或讨论，此时用户作为生产者所发表的内容通常体现了他们所具有的技能。另一方面，用户也经常搜索、浏览、评论、顶踩、收藏自己感兴趣的内容，此时用户作为消费者所关注的内容则体现了他们所拥有的兴趣爱好。一般情况下，用户的技能比较集中于少量的一个或几个话题，而用户的兴趣则可能相对比较广泛地分布于多个话题。基于这一事实，我们提出将用户的技能话题分布和兴趣话题分布区别对待，分别使用生产者（作者）和消费者（读者）的身份来发现用户的技能和兴趣。我们以 LDA 模型为基础，同时加入文档的作者和读者信息，形成作者—读者—话题（author-reader-topic，ART）模型。ART 模型将用户作为作者和读者的两种身份信息加入到话题的建模过程中，不仅可以进一步增强话题的聚集效果，还可以同步分别建模用户的技能和兴趣。该模型的直观含义是：文档的作者决定了文档的话题，而文档的话题决定了词的生成并且吸引对该话题感兴趣的用户对该文档进行阅读。

与 AT 模型类似，ART 模型仍然是一种层次式的贝叶斯概率模型，它包含词、话题、文档、作者和读者五层结构，其盘式表示如图 3 所示。在 ART 模型中，每篇文档 $d$ 对应一个作者 $a_d$ 和多个读者 $\boldsymbol{r}_d$，每个作者 $a$ 对应的话题的多项式分布为 $\boldsymbol{\theta}_a$，每个话题 $t$ 对应的词的多项式分布为 $\boldsymbol{\phi}_t$ 以及读者的多项式分布为 $\boldsymbol{\varphi}_t$。该模型的文档生成过程的形式化描述见算法 1：首先，根据 Dirichlet 超参数分别采样作者—话题分布 $\boldsymbol{\theta}$、话题—词分布 $\boldsymbol{\phi}$ 以及话题—读者分布 $\boldsymbol{\varphi}$，其分别服从 Dirichlet 分布 Dir($\boldsymbol{\alpha}$)、Dir($\boldsymbol{\beta}$)和 Dir($\boldsymbol{\eta}$)；然后，对于每篇文档中的每个词，根据文档对应作者的作者—话题分布 $\boldsymbol{\theta}$ 采样生成一个话题 $z$，$z$ 服从多项式分布 Mul($\boldsymbol{\theta}$)；接下来，基于生成的话题 $z$ 分别独立地从话题—词分布 $\boldsymbol{\phi}$ 和话题—读者分布 $\boldsymbol{\varphi}$ 中采样生成一个词 $w$ 和一个读者 $x$，$w$ 和 $x$

分别服从多项式分布 Mul($\boldsymbol{\phi}$)和 Mul($\boldsymbol{\varphi}$)。

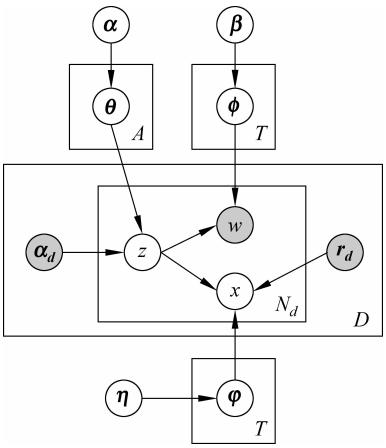


图3 ART模型的盘式表示

**算法1 ART模型的文档生成过程**

**for** each author $a \in A$ **do**
  //draw a distribution over topics
  $\boldsymbol{\theta}_a \sim \mathrm{Dir}(\boldsymbol{\alpha})$
**end for**
**for** each topic $t \in T$ **do**
  //draw a distribution over words
  $\boldsymbol{\phi}_t \sim \mathrm{Dir}(\boldsymbol{\beta})$
  //draw a distribution over readers
  $\boldsymbol{\varphi}_t \sim \mathrm{Dir}(\boldsymbol{\eta})$
**end for**
**for** each document $d \in [1, D]$ and its author $a_d$ **do**
  **for** each word $n \in [1, N_d]$ **do**
    assign a topic $z_{dn} \sim \mathrm{Mul}(\boldsymbol{\theta}_{a_d})$;
    draw a word $w \sim \mathrm{Mul}(\boldsymbol{\phi}_{z_{dn}})$;
    draw a reader $x \sim \mathrm{Mul}(\boldsymbol{\varphi}_{z_{dn}})$;
  **end for**
**end for**

给定超参数 $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\eta}$ 以及文档 $d$ 的作者 $a_d$ 和读者 $\boldsymbol{r}_d$,语料库的生成概率如式(1)所示。

$$P(\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\varphi},\boldsymbol{z},\boldsymbol{w},\boldsymbol{x} \mid \boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\eta},\boldsymbol{a},\boldsymbol{r}) = \prod_{a=1}^{A} p(\boldsymbol{\theta}_a \mid \boldsymbol{\alpha}) \prod_{t=1}^{T} p(\boldsymbol{\phi}_t \mid \boldsymbol{\beta}) \prod_{t=1}^{T} p(\boldsymbol{\varphi}_t \mid \boldsymbol{\eta}) \cdot \prod_{d=1}^{D} \prod_{n}^{N_d} (p(z_{dn} \mid \boldsymbol{\theta}_{a_d}) p(x_{dn} \mid \boldsymbol{\varphi}_{z_{dn}}, \boldsymbol{r}_d) p(w_{dn} \mid \boldsymbol{\phi}_{z_{dn}})) \tag{1}$$

### 2.2.2 模型推导

我们采用吉布斯采样方法来近似推导ART模型。吉布斯采样是一种高效的MCMC(Markov Chain Monte Carlo)采样方法,它通过迭代采样方式对复杂的概率分布进行推断。为了得到参数 $\boldsymbol{\theta}$、$\boldsymbol{\phi}$ 和 $\boldsymbol{\varphi}$,需要计算词 $w_{dn}$ 的话题指派和读者指派的条件分布 $p(z_{dn}, x_{dn} \mid \boldsymbol{z}_{-dn}, \boldsymbol{x}_{-dn}, \boldsymbol{w}, a_d, \boldsymbol{r}_d, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta})$,其中,$\boldsymbol{z}_{-dn}$ 和 $\boldsymbol{x}_{-dn}$ 分别指除文档 $d$ 中第 $n$ 个词以外的其他所有词的话题指派和读者指派(符号 $-dn$ 表示排除当前词 $w_{dn}$)。为了简化公式的描述,我们引入 $\Delta$ 函数(the Dirichlet delta function)[15],对于含有 $V$ 维的Dirichlet先验参数 $\boldsymbol{\delta}$,$\Delta$ 函数定义,如式(2)所示。

$$\Delta(\boldsymbol{\delta}) = \prod_{k=1}^{V} \Gamma(\delta_k) / \Gamma\left(\sum_{k=1}^{V} \delta_k\right) \tag{2}$$

其中,$\Gamma(\cdot)$是伽玛函数。

基于图3中概率图模型的独立性假设,给定超参数,则话题、读者和词的联合分布可形式化推导,如式(3)所示。

$$\begin{aligned} & P(\boldsymbol{z},\boldsymbol{x},\boldsymbol{w} \mid \boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\eta}) \\ &= P(\boldsymbol{z} \mid \boldsymbol{\alpha}) P(\boldsymbol{w} \mid \boldsymbol{z},\boldsymbol{\beta}) P(\boldsymbol{x} \mid \boldsymbol{z},\boldsymbol{\eta}) \\ &= \prod_{a=1}^{A} \frac{\Delta(\boldsymbol{n}_a + \alpha)}{\Delta(\boldsymbol{\alpha})} \prod_{tw=1}^{T} \frac{\Delta(\boldsymbol{n}_{tw} + \boldsymbol{\beta})}{\Delta(\boldsymbol{\beta})} \prod_{tr=1}^{T} \frac{\Delta(\boldsymbol{n}_{tr} + \boldsymbol{\eta})}{\Delta(\boldsymbol{\eta})} \end{aligned} \tag{3}$$

其中 $\boldsymbol{n}_a = \{n_a^t\}_{t=1}^{T}$,$n_a^t$ 表示作者 $a$ 在话题 $t$ 下产生的词的个数;$\boldsymbol{n}_{tw} = \{n_{tw}^w\}_{w=1}^{V}$,$n_{tw}^w$ 表示话题 $tw$ 产生词 $w$ 的个数;$\boldsymbol{n}_{tr} = \{n_{tr}^r\}_{r=1}^{R}$,$n_{tr}^r$ 表示话题 $tr$ 产生读者 $r$ 的次数。根据式(3)的联合分布以及Markov链式法则,可以推导出上述的条件分布,如式(4)所示。

$$\begin{aligned} & p(z_{dn}, x_{dn} \mid \boldsymbol{z}_{-dn}, \boldsymbol{x}_{-dn}, \boldsymbol{w}, a_d, \boldsymbol{r}_d, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}) \\ &= \frac{p(\boldsymbol{z},\boldsymbol{x},\boldsymbol{w} \mid a_d, \boldsymbol{r}_d, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta})}{p(\boldsymbol{z}_{-dn}, \boldsymbol{x}_{-dn}, \boldsymbol{w} \mid a_d, \boldsymbol{r}_d, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta})} \\ &\propto \frac{n_{a_d}^{z_{dn},-dn} + \alpha_{z_{dn}}}{\sum_{t=1}^{T} (n_{a_d}^{t,-dn} + \alpha_t)} \frac{n_{z_{dn}}^{w_{dn},-dn} + \beta_{w_{dn}}}{\sum_{w=1}^{V} (n_{z_{dn}}^{w,-dn} + \beta_w)} \\ &\quad \frac{n_{z_{dn}}^{r_{dn},-dn} + \eta_{r_{dn}}}{\sum_{r=1}^{R} (n_{z_{dn}}^{r,-dn} + \eta_r)} \end{aligned} \tag{4}$$

其中,$\boldsymbol{n}_{a_d}^{z_{dn},-dn}$ 表示除了当前词 $w_{dn}$ 外,所有文档中同时属于作者 $a_d$ 和话题 $z_{dn}$ 的词的频数;$n_{z_{dn}}^{w_{dn},-dn}$ 表示除了当前词 $w_{dn}$ 外,所有文档中词 $w_{dn}$ 属于话题 $z_{dn}$ 的频数;$n_{z_{dn}}^{r_{dn},-dn}$ 表示除了当前词 $w_{dn}$ 外,所有文档中同时属于读者 $x_{dn}$ 和话题 $z_{dn}$ 的词的频数。

吉布斯采样收敛后,我们就可以根据采样的结果估计作者—话题分布 $\boldsymbol{\theta}$,话题—词分布 $\boldsymbol{\phi}$ 以及话题—读者分布 $\boldsymbol{\varphi}$,分别如式(5)~(7)所示。

$$\theta_{az} = \frac{n_a^z + \alpha_z}{\sum_{t=1}^{T} (n_a^t + \alpha_t)} \tag{5}$$

$$\phi_{zw} = \frac{n_z^w + \beta_w}{\sum_{v=1}^{V} (n_z^v + \beta_v)} \tag{6}$$

$$\varphi_{zx}=\frac{n_z^x+\eta_x}{\sum_{r=1}^{R}(n_z^r+\eta_r)} \tag{7}$$

ART 模型的吉布斯采样方法的详细推导见附录。我们使用吉布斯采样进行参数估计的算法流程详见算法 2。

**算法 2　ART 模型的参数估计**

initialize the reader and topic assignments randomly for all tokens;
//Gibbs sampling over burn-in period and sampling period
**while** not finished **do**
　**for** all documents $d\in[1,D]$ **do**
　　**for** all words $n\in[1,N_d]$ in document $d$ **do**
　　draw $x_{dn}$ and $z_{dn}$ from Eq. (4);
　　update $n_{d}^{z_{dn}}$, $n_{a_d}$, $n_{z_{dn}}^{w_{dn}}$, $n_{z_{dn}}$, $n_{z_{dn}}^{x_{dn}}$;
　　**end for**
　**end for**
　//check convergence and read out parameters
　**if** converged and $L$ sampling iterations since last read out **then**
　　//the different parameters read outs are averaged
　　read out parameter set $\boldsymbol{\theta}$ according to Eq. (5);
　　read out parameter set $\boldsymbol{\phi}$ according to Eq. (6);
　　read out parameter set $\boldsymbol{\varphi}$ according to Eq. (7);
　**end if**
**end while**

## 3　实验

本节将详细介绍基于 CSDN 技术社区数据集的实验过程及结果分析，将本文提出的 ART 模型与经典的 LDA 模型、在技能发现方面表现最好的 AT 模型以及衍生的读者—话题(RT)模型三种基准方法进行了对比，验证了 ART 模型在用户技能和兴趣发现方面的优势。

### 3.1　数据集

本文使用的数据集来自全球最大的中文 IT 技术社区 CSDN。我们从 CSDN 采集了 2015 年 01 月至 2016 年 7 月之间部分活跃用户产生的内容和行为记录，其中用户产生的内容包含用户在该时间段内发表的所有博客，以及用户的自定义技能标签和兴趣标签；用户行为记录包括在该时间段内用户对博客的浏览、顶踩、评论和收藏行为的日志记录。其中，用户自定义技能标签和兴趣标签用来评估本文提出的技能与兴趣发现方法的效果，用户浏览、顶踩、评论和收藏过的文档均视为该用户读过的文档。值得注意的是，所有涉及用户个人隐私的信息，均不包含在采集的数据集之中。

数据集共包含与 4 357 位用户相关的 27 880 篇博客文档，其中部分用户既是作者又是读者。我们对所有文档进行了必要的预处理，包括去除文档中的 HTML 标记、程序代码块以及 URL 链接，然后采用 NLPIR 分词工具[16]进行分词，并在分词过程中使用了清华大学开放 IT 词库[17]，最后去除停用词并去除 TF-IDF 值较低的词。处理后的数据集统计信息如表 2 所示。

**表 2　数据集统计信息**

| 项目 | 数量 |
|---|---|
| 博客文档数 | 27 880 |
| 用户数 | 4 357 |
| 作者用户数 | 2 826 |
| 读者用户数 | 2 232 |
| 作者与读者交集用户数 | 701 |
| 词数 | 37 493 |
| 数据集总词数 | 9 065 864 |
| 用户自定义技能标签数 | 38 400 |
| 用户自定义兴趣标签数 | 52 949 |

### 3.2　基准方法

我们在实验过程中采用三种基准方法与本文提出的 ART 模型进行对比。除了经典的 LDA 模型和当前在技能发现方面表现最好的 AT 模型，我们还基于 AT 模型和 ART 模型衍生出一种读者—话题(reader-topic，RT)模型，用来单独对用户的兴趣进行建模。RT 模型的盘式表示如图 4 所示，它和 ART 模型的区别在于没有将文档的作者信息加入到模型中，只保留了读者信息，因此只用来建模用户兴趣。RT 模型的文档生成过程和 ART 模型类似，首先，根据 Dirichlet 超参数分别采样文档—话题分布 $\boldsymbol{\theta}$、话题—词分布 $\boldsymbol{\phi}$ 以及话题—读者分布 $\boldsymbol{\varphi}$，其分别服从 Dirichlet 分布 Dir($\boldsymbol{\alpha}$)、Dir($\boldsymbol{\beta}$)和 Dir($\boldsymbol{\eta}$)；然后，对于每篇文档中的每个词，从文档—话题分布 $\boldsymbol{\theta}$ 中采样生成一个话题 $z$，$z$ 服从多项式分布 Mul($\boldsymbol{\theta}$)，进而基于生成的话题 $z$ 分别独立地从话题—词分布 $\boldsymbol{\phi}$ 和话题—读者分布 $\boldsymbol{\varphi}$ 中采样生成一个词 $w$ 和一个读者 $x$，$w$ 和 $x$ 分别服从 Mul($\boldsymbol{\phi}$)和 Mul($\boldsymbol{\varphi}$)分布。RT 模型的文档生成过程以

及吉布斯采样的推导过程和ART模型类似，本文在此不再赘述。

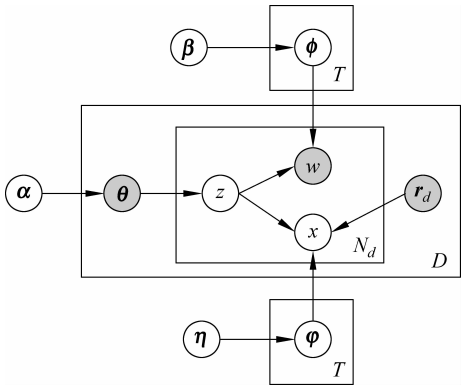


图4 RT模型的盘式表示

### 3.3 实验参数设置

在实验过程中，为了对四种模型进行相互比较，我们对四种模型的话题数目及其他超参数进行了相同的设置。对于话题数目，我们根据数据集的话题分布情况经验性地设置话题数 $T=100$；对于其他超参数，我们尝试了不同的超参数设置，发现模型效果并没有受到超参数的较大影响，因此根据文献[14]，我们将超参数设置为固定值：$\alpha=50/T$、$\beta=0.01$、$\eta=0.1$。在模型训练过程中，我们发现模型迭代1 500次左右就基本达到收敛状态，为了统一标准并确保所有模型都能达到收敛，我们对四种模型均设置迭代次数为2 000次。

### 3.4 话题聚集结果

ART模型迭代收敛后，可以利用式(6)来提取整个数据集的话题。我们首先从100个话题中筛选出10个话题，并尽量让这些话题覆盖不同的技术领域，然后列出每个话题的前10个词，如表3所示。通过分析每个话题下代表性词汇的含义，给出了每个话题的语义。我们发现，ART模型聚集出来的话题比较容易理解，每个话题下的大部分词汇在语义上都与话题有较强的相关性。

**表3 数据集话题聚集结果**

| 话题 | 语义 | 前10个话题词 |
|---|---|---|
| topic 2 | 深度学习 | 神经网络、训练、特征、caffe、卷积、cnn、深度学习、模型、参数、神经元 |
| topic 5 | 大数据处理 | hadoop、大数据、hive、map、hdfs、文件、hbase、reduce、mapreduce、namenode |
| topic 8 | 游戏开发 | 游戏、cocos2d-x、场景、动作、lua、unity、图片、物体、精灵、碰撞 |
| topic 14 | 多媒体处理 | 视频、播放、音频、ffmpeg、sdl、编码、播放器、解码、媒体、声音 |
| topic 15 | 多线程 | 线程、等待、同步、调用、线程池、synchronized、多线程、队列、阻塞、资源 |
| topic 22 | 图像处理 | 图像、矩阵、opencv、区域、变换、坐标、灰度、参数、图片、检测 |
| topic 27 | Web开发 | js、页面、浏览器、javascript、html、前端、html5、加载、react、css |
| topic 42 | 网络与通信 | 发送、数据包、tcp、网络、ip、udp、主机、连接、路由器、传输 |
| topic 96 | 数据库 | oracle、数据库、sql、备份、database、select、orcl、恢复、alter、日志 |
| topic 97 | 移动开发 | android、设置、效果、fragment、控件、显示、布局、listview、activity、添加 |

### 3.5 用户技能词和兴趣词提取结果

通过式(6)我们可以得到每个话题的词分布 $\boldsymbol{\phi}$，通过式(5)和(7)可以分别得到每个作者的话题分布 $\boldsymbol{\theta}$ 和每个读者的话题分布 $\boldsymbol{\varphi}$，然后通过整合作者—话题分布和话题—词分布来计算用户的技能词，通过整合读者—话题分布和话题—词分布来计算用户的兴趣词，形式化公式分别如式(8)和式(9)所示。计算用户技能词和兴趣词的核心思想在于计算作者(或读者)、话题和词的联合分布的积分，式(8)中 $\Omega_{aw}$ 表示作者 $a$ 在技能词 $w$ 上的相关度，式(9)中 $H_{rw}$ 表示读者 $r$ 在兴趣词 $w$ 上的相关度。

$$\Omega_{aw}=\int_t(\phi_{tw}\cdot\theta_{at})\mathrm{d}t=\sum_{t=1}^{T}(\phi_{tw}\cdot\theta_{at}) \quad (8)$$

$$H_{rw}=\int_t(\phi_{tw}\cdot\varphi_{tr})\mathrm{d}t=\sum_{t=1}^{T}(\varphi_{tw}\cdot\varphi_{tr}) \quad (9)$$

通过式(8)～(9)我们可以计算每个作者对词汇集 $V$ 中每个词的技能相关度以及每个读者对词汇集 $V$ 中每个词的兴趣相关度，从而得到和每个用户相关度比较高的技能词和兴趣词。表4和表5分别列出了四个代表性用户的前十个技能词和前十个兴趣词，这四个用户的自定义技能标签和兴趣标签如表6所示。

**表 4　各种模型发现的用户技能词**

| | | |
|---|---|---|
| LDA | user1 | hadoop、数据、文件、配置、spark、安装、代码、hdfs、hive、输入 |
| | user2 | 文件、数据、mysql、事务、命令、进程、内存、系统、性能、索引 |
| | user3 | 对象、代码、函数、数据、view、android、游戏、安装、ios、编译 |
| | user4 | 对象、文件、标签、javascript、html、字符串、浏览器、css、数据、实例 |
| RT | user1 | hadoop、数据、文件、spark、配置、安装、代码、hdfs、命令、函数 |
| | user2 | 索引、数据、mysql、文件、命令、进程、内存、innodb、函数、数据库 |
| | user3 | 代码、对象、函数、游戏、数据、安装、view、mysql、图片、android |
| | user4 | javascript、对象、html、浏览器、文件、标签、xml、jquery、css、字符串 |
| AT | user1 | hadoop、spark、数据、配置、hdfs、命令、scala、安装、集群、hive |
| | user2 | mysql、索引、数据、结点、内存、文件、进程、磁盘、系统、sql |
| | user3 | 游戏、代码、对象、图片、算法、unity、加载、安装、渲染、opengl |
| | user4 | html、javascript、浏览器、对象、jquery、标签、css、文件、样式、选择器 |
| ART | user1 | spark、hadoop、hdfs、配置、hive、集群、yarn、scala、java、zookeeper |
| | user2 | oracle、mysql、数据库、文件、innodb、索引、进程、sql、日志、事务 |
| | user3 | android、游戏、cocos2d-x、图片、unity、应用程序、opengl、cocos2d、动画、数据 |
| | user4 | javascript、浏览器、html、css、jquery、对象、数据、样式、选择器、标签 |

**表 5　各种模型发现的用户兴趣词**

| | | |
|---|---|---|
| LDA | user1 | hadoop、数据、文件、spark、配置、安装、代码、hdfs、hive、输入 |
| | user2 | 数据、文件、命令、事务、mysql、代码、进程、内存、系统、数据库 |
| | user3 | 代码、对象、函数、数据、view、android、游戏、安装、ios、输入 |
| | user4 | 对象、函数、文件、标签、javascript、html、浏览器、字符串、数据、css |
| AT | user1 | hadoop、数据、spark、文件、代码、配置、hdfs、命令、安装、输入 |
| | user2 | 数据、索引、mysql、文件、内存、命令、系统、进程、线程、服务器 |
| | user3 | 游戏、代码、数据、对象、函数、图片、view、安装、android、mysql |
| | user4 | 对象、函数、javascript、html、文件、浏览器、标签、jquery、字符串、实例 |
| RT | user1 | hadoop、数据、spark、配置、hdfs、命令、安装、代码、hive、集群 |
| | user2 | 索引、mysql、数据、文件、内存、进程、命令、数据库、innodb、事务 |
| | user3 | 游戏、算法、函数、mysql、图片、代码、数据、unity、内存、动画 |
| | user4 | html、标签、javascript、对象、函数、浏览器、css、jquery、样式、文件 |
| ART | user1 | spark、hadoop、hdfs、配置、hive、命令、集群、yarn、安装、scala |
| | user2 | oracle、mysql、数据库、文件、innodb、索引、进程、命令、sql、系统、磁盘 |
| | user3 | 游戏、android、数据、cocos2d-x、图片、函数、mysql、应用程序、unity、动画 |
| | user4 | javascript、浏览器、html、css、数据、jquery、对象、模块、标签、样式 |

**表 6　用户自定义技能和兴趣标签**

| | | |
|---|---|---|
| user1 | 技能标签 | hadoop、hive、spark、zookeeper、java、scala、mysql、数据库 |
| | 兴趣标签 | hive、spark、zookeeper、scala、hadoop、大数据、数据库、mysql、系统运维 |

续表

| | | |
|---|---|---|
| user2 | 技能标签 | oracle、mysql、shell、linux、os |
| | 兴趣标签 | oracle、mysql、mongodb、mybatis、数据库、shell、linux、centos、os、php、html |
| user3 | 技能标签 | android、游戏开发、unity、ios 开发、phonegap、uikit、opengl |
| | 兴趣标签 | 游戏开发、ios 开发、android、unity、phonegap、uikit、opengl、sql |
| user4 | 技能标签 | web 开发、javascript、html5、css3、jquery |
| | 兴趣标签 | web 开发、html5、css3、javascript、jquery、设计模式 |

将表 4 和表 5 的结果分别与用户自定义技能标签和兴趣标签进行比较，我们发现 ART 模型发现的技能词和兴趣词与用户的自定义标签相关度较高。例如，user1 的自定义技能和兴趣标签体现在“大数据”领域，而 ART 发现的诸如“hadoop”“hive”“spark”等词与此的相关度较高；user2 的自定义技能和兴趣标签体现在“数据库”和“操作系统”领域，而 ART 发现的诸如“oracle”“mysql”“进程”“命令”等词与此相关度较高；user3 的自定义技能和兴趣标签体现在“移动开发”和“游戏开发”领域，而 ART 发现的诸如“android”“游戏”等词与此相关度较高；user4 的自定义技能和兴趣标签体现在“web 开发”领域，而 ART 发现的诸如“javascript”“html”等词与此相关度较高。

将表 4 和表 5 中 ART 模型发现的用户技能和兴趣之间进行比较，我们发现用户的技能和兴趣相似度很高，但用户的技能更加专一，用户的兴趣则相对广泛。例如，user1 的技能和兴趣都体现在“大数据”领域，但 user1 还对“系统运维”比较感兴趣；user3 的技能和兴趣都体现在“游戏开发”“移动开发”领域，但 user3 还对“数据库”比较感兴趣；user4 的技能和兴趣都体现在“web 开发”领域，但 user4 还对“设计模式”比较感兴趣。为了进一步分析用户技能分布和兴趣分布之间的差异，我们分别计算作者技能分布熵（简称技能熵）和读者兴趣分布熵（简称兴趣熵），如式(10)～(11)所示。

$$E_a = -\sum_{t=1}^{T}(\theta_{at} \cdot \log\theta_{at}) \tag{10}$$

$$E_r = -\sum_{t=1}^{T}(\varphi_{tr} \cdot \log\varphi_{tr}) \tag{11}$$

我们计算了 701 个既是作者又是读者的用户的技能熵和兴趣熵，并进行了相关统计，结果如表 7 所示。从表中可以看出，兴趣熵的平均值要高于技能熵，这进一步表明了用户技能的专一性和用户兴趣的广泛性。

**表 7 技能熵与兴趣熵统计值比较**

| | 平均值 | 最大值 | 中位数 | 最小值 |
|---|---|---|---|---|
| 技能熵 | 0.897 | 2.452 | 0.806 | 0.061 |
| 兴趣熵 | 1.291 | 3.865 | 1.168 | 0.044 |

表 4 和表 5 分别列出了 LDA、AT 和 RT 三种基准方法提取的用户技能词和兴趣词结果，从表中可以看出，在用户技能发现方面，相比于其他三个模型，ART 模型发现的技能词与用户自定义技能标签相关度更高，而且相关度高的词排序更加靠前；同样，在用户兴趣发现方面，ART 模型也要优于其他三个模型。为了定量评价四种模型在发现用户技能和兴趣方面的优劣，我们将四种模型发现的技能词和兴趣词分别和用户自定义技能标签和兴趣标签求交集，计算技能发现和兴趣发现的准确率和召回率，如式(12)～(15)所示：

$$\text{Precision}_{\text{exp}} = \frac{\sum_{u=1}^{U_a}\text{hit}_{\text{exp}}^{u}/K}{U_a} \tag{12}$$

$$\text{Recall}_{\text{exp}} = \frac{\sum_{u=1}^{U_a}\text{hit}_{\text{exp}}^{u}/\text{label}_{\text{exp}}^{u}}{U_a} \tag{13}$$

$$\text{Precision}_{\text{int}} = \frac{\sum_{u=1}^{U_r}\text{hit}_{\text{int}}^{u}/K}{U_r} \tag{14}$$

$$\text{Recall}_{\text{int}} = \frac{\sum_{u=1}^{U_r}\text{hit}_{\text{int}}^{u}/\text{label}_{\text{int}}^{u}}{U_r} \tag{15}$$

其中，$U_a$ 和 $U_r$ 分别指作者用户数和读者用户数；$\text{hit}_{\text{exp}}^{u}$ 和 $\text{hit}_{\text{int}}^{u}$ 指模型发现的用户 $u$ 的技能词和兴趣词分别和用户自定义技能标签和兴趣标签的交集数；$\text{label}_{\text{exp}}^{u}$ 和 $\text{label}_{\text{int}}^{u}$ 分别指用户 $u$ 的自定义技能标签数和兴趣标签数。

具体地，我们分别计算出各种模型发现的每个用户的前 $K$($K$=5,10,20,50,100)个技能词和兴趣词，与该用户自定义技能标签和兴趣标签求交集，计算每个用户技能发现和兴趣发现的准确率和召回率，然后对所有用户求平均。四种模型技能发现的准确率和召回率如表 8 和表 9 所示，兴趣发现的准确率和召回率如表 10 和表 11 所示。从表中可以看出，在技能发现方面，ART 要显著优于 AT 模型，AT 模型要优于 RT 模型，LDA 模型效果最差；在兴趣发现方面，ART 要显著优于 RT 模型，RT 模型要优于 AT 模型，同样 LDA 模型效果最差。需要说明的是，准确率和召回率整体不高的原因，一方面是因为用户自定义的技能标签和兴趣标签通常更抽象，而模型发现的技能词和兴趣词通常更具体；另一方面，用户自定义标签也存在更新不及时等问题，因此两者的交集偏少，从而导致准确率和召回率偏低，但这并不会影响其作为模型评价标准的客观性和公正性。

**表 8　四种模型技能发现的准确率比较**

单位：%

| 模型 | Top 5 | Top 10 | Top 20 | Top 50 | Top 100 |
|---|---|---|---|---|---|
| LDA | 8.42 | 6.96 | 5.37 | 3.41 | 2.29 |
| AT | 13.92 | 11.26 | 8.56 | 5.49 | 3.68 |
| RT | 12.19 | 10.57 | 8.28 | 5.32 | 3.59 |
| ART | 18.60 | 17.22 | 14.27 | 10.16 | 7.48 |

**表 9　四种模型技能发现的召回率比较**

单位：%

| 模型 | Top 5 | Top 10 | Top 20 | Top 50 | Top 100 |
|---|---|---|---|---|---|
| LDA | 7.61 | 11.08 | 14.93 | 20.40 | 24.56 |
| AT | 11.44 | 16.41 | 22.05 | 30.04 | 36.46 |
| RT | 10.64 | 15.93 | 21.80 | 29.66 | 36.40 |
| ART | 16.06 | 20.59 | 25.32 | 32.84 | 40.29 |

**表 10　四种模型兴趣发现的准确率比较**

单位：%

| 模型 | Top 5 | Top 10 | Top 20 | Top 50 | Top 100 |
|---|---|---|---|---|---|
| LDA | 8.33 | 7.47 | 6.05 | 4.11 | 2.85 |
| AT | 16.07 | 14.68 | 12.08 | 8.12 | 5.61 |
| RT | 16.57 | 14.85 | 12.12 | 8.13 | 5.64 |
| ART | 19.91 | 19.19 | 16.48 | 11.91 | 8.91 |

**表 11　四种模型兴趣发现的召回率比较**

单位：%

| 模型 | Top 5 | Top 10 | Top 20 | Top 50 | Top 100 |
|---|---|---|---|---|---|
| LDA | 3.13 | 5.20 | 7.65 | 11.86 | 15.49 |
| AT | 5.53 | 9.40 | 14.12 | 22.02 | 28.68 |
| RT | 5.74 | 9.61 | 14.31 | 22.06 | 28.72 |
| ART | 11.02 | 14.21 | 18.20 | 24.87 | 32.00 |

以上实验结果证明，随着作者信息和读者信息的加入，ART 模型提高了话题聚集效果，能够更准确地同时对用户的技能和兴趣进行建模，显著优于其他现有的技能或兴趣发现方法。

## 4　结论

本文提出了一个新颖的作者—读者—话题(ART)模型来同步发现在线技术社区中用户的技能和兴趣。该模型能够有效地将文档的作者信息和读者信息关联起来，提升话题聚集效果，产生更准确的作者技能话题分布和读者兴趣话题分布。

在 CSDN 社区的真实数据集上的实验结果表明，本文提出的 ART 模型能够有效地发现用户的技能和兴趣，提取的用户技能词和兴趣词比其他现有的技能或兴趣挖掘方法更准确。与此同时，我们也验证了用户技能相对集中、用户兴趣相对分散的假设。本文提出的方法可以广泛应用于在线社区的用户技能与兴趣挖掘，服务于社区运营者进行用户画像，向用户提供精准推荐和个性化服务。

## 参考文献

[1] Bhattacharya P, Zafar M B, Ganguly N, et al. Inferring user interests in the Twitter social network[C]// Proceedings of the 8th ACM Conference on Recommender systems. ACM, 2014: 357-360.

[2] Wen Z, Lin C Y. On the quality of inferring interests from social neighbors[C]//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2010: 373-382.

[3] Pennacchiotti M, Silvestri F, Vahabi H, et al. Making your interests follow you on Twitter[C]//Proceedings of the 21st ACM International Conference on Information and Knowledge Management. ACM, 2012: 165-174.

[4] Yang T, Lee D, Yan S. Steeler nation, 12th man, and

boo birds: classifying Twitter user interests using time series[C]//Proceedings of 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. IEEE, 2013: 684-691.

[5] He H, Hai H, Rujing W. FCA-based web user profile mining for topics of interest[C]//Proceedings of 2007 IEEE International Conference on Integration Technology. IEEE, 2007: 778-782.

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

[7] Rosen-Zvi M, Griffiths T, Steyvers M, et al. The author-topic model for authors and documents[C]//Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence. AUAI Press, 2004: 487-494.

[8] Tu Y, Johri N, Roth D, et al. Citation author topic model in expert search[C]//Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics, 2010: 1265-1273.

[9] Tang J, Zhang J, Yao L, et al. Arnetminer: Extraction and mining of academic social networks[C]//Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2008: 990-998.

[10] Weng J, Lim E P, Jiang J, et al. Twitter rank: Finding topic-sensitive influential Twitterers[C]//Proceedings of the 3rd ACM International Conference on Web Search and Data Mining. ACM, 2010: 261-270.

[11] Hong L, Davison B D. Empirical study of topic modeling in Twitter[C]//Proceedings of the 1st Workshop on Social Media Analytics. ACM, 2010: 80-88.

[12] Xu Z, Ru L, Xiang L, et al. Discovering user interest on Twitter with a modified author-topic model[C]//Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01. IEEE Computer Society, 2011: 422-429.

[13] Li H, Yan J, Weihong H, et al. Mining user interest in Microblogs with a user-topic model[J]. China Communications, 2014, 11(8): 131-144.

[14] Griffiths T L, Steyvers M. Finding scientific topics[J]. Proceedings of the National Academy of Sciences, 2004, 101(suppl 1): 5228-5235.

[15] Heinrich G. Parameter estimation for text analysis[R]. Technical Report, 2008.

[16] NLPIR 汉语分词系统[CP/OL]: http://ictclas.nlpir.org/.

[17] 韩世依, 张钰晖, 马云山, 等. THUOCL: 清华大学开放中文词库[DB], 2016.

张东雷(1993—),硕士研究生,主要研究领域为文本挖掘、社交网络挖掘。
E-mail: dongleizhang@bjtu.edu.cn

万怀宇(1981—),通信作者,副教授,主要研究领域为社交网络挖掘、文本挖掘、交通数据挖掘等。
E-mail: hywan@bjtu.edu.cn

林友芳(1971—),教授,主要研究领域为数据仓库与数据挖掘、大数据技术、智能技术等。
E-mail: yflin@bjtu.edu.cn

## 附录 ART 模型吉布斯采样推导

为了进行吉布斯采样,我们需要计算词 $w_{dn}$ 的话题指派和读者指派的条件分布 $p(z_{dn}, x_{dn} \mid \boldsymbol{z}_{-dn}, \boldsymbol{x}_{-dn}, \boldsymbol{w}, a_d, \boldsymbol{r}_d, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta})$,其中,$\boldsymbol{z}_{-dn}$ 和 $\boldsymbol{x}_{-dn}$ 分别指除文档 $d$ 中第 $n$ 个词以外的其他词的话题指派和读者指派(符号 $-dn$ 表示排除当前词 $w_{dn}$),详细的推导过程如下所示:

$$
\begin{aligned}
& p(z_{dn}, x_{dn} \mid \boldsymbol{z}_{-dn}, \boldsymbol{x}_{-dn}, \boldsymbol{w}, a_d, \boldsymbol{r}_d, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}) \\
&= \frac{p(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{w} \mid a_d, \boldsymbol{r}_d, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta})}{p(\boldsymbol{z}_{-dn}, \boldsymbol{x}_{-dn}, \boldsymbol{w} \mid a_d, \boldsymbol{r}_d, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta})} \\
&= \frac{p(\boldsymbol{z} \mid a_d, \boldsymbol{\alpha})}{p(\boldsymbol{z}_{-dn} \mid a_d, \boldsymbol{\alpha})} \cdot \frac{p(\boldsymbol{w} \mid \boldsymbol{z}, \boldsymbol{\beta})}{p(\boldsymbol{w}_{-dn} \mid \boldsymbol{z}_{-dn}, \boldsymbol{\beta}) \cdot p(w_{dn})} \cdot \frac{p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\eta}, \boldsymbol{r}_d)}{p(\boldsymbol{x}_{-dn} \mid \boldsymbol{z}_{-dn}, \boldsymbol{\eta}, \boldsymbol{r}_d)}
\end{aligned}
$$

$$\propto \frac{\Delta(\boldsymbol{n}_{a_d}+\boldsymbol{\alpha})}{\Delta(\boldsymbol{n}_{a_d,-dn}+\boldsymbol{\alpha})}\frac{\Delta(\boldsymbol{n}_{tw}+\boldsymbol{\beta})}{\Delta(\boldsymbol{n}_{tw,-dn}+\boldsymbol{\beta})}\frac{\Delta(\boldsymbol{n}_{tr}+\boldsymbol{\eta})}{\Delta(\boldsymbol{n}_{tr,-dn}+\boldsymbol{\eta})}$$

$$\propto \frac{\Gamma(n_{a_d}^{z_{dn}}+\alpha_{z_{dn}})\Gamma\left(\sum_{t=1}^{T}(n_{a_d}^{t}+\alpha_t)\right)}{\Gamma(n_{a_d,-dn}^{z_{dn}}+\alpha_{z_{dn}})\Gamma\left(\sum_{t=1}^{T}(n_{a_d}^{t}+\alpha_t)\right)}$$

$$\cdot\frac{\Gamma(n_{z_{dn}}^{w_{dn}}+\beta_{w_{dn}})\Gamma\left(\sum_{w=1}^{V}(n_{z_{dn}}^{w}+\beta_w)\right)}{\Gamma(n_{z_{dn},-dn}^{w_{dn}}+\beta_{w_{dn}})\Gamma\left(\sum_{w=1}^{V}(n_{z_{dn}}^{w}+\beta_w)\right)}$$

$$\cdot\frac{\Gamma(n_{z_{dn}}^{x_{dn}}+\eta_{x_{dn}})\Gamma\left(\sum_{r=1}^{R}(n_{z_{dn}}^{r}+\eta_r)\right)}{\Gamma(n_{z_{dn},-dn}^{x_{dn}}+\eta_{x_{dn}})\Gamma\left(\sum_{r=1}^{R}(n_{z_{dn}}^{r}+\eta_r)\right)}$$

$$\propto \frac{n_{a_d,-dn}^{z_{dn}}+\alpha_{z_{dn}}}{\sum_{t=1}^{T}(n_{a_d,-dn}^{t}+\alpha_t)}\frac{n_{z_{dn},-dn}^{w_{dn}}+\beta_{w_{dn}}}{\sum_{w=1}^{V}(n_{z_{dn},-dn}^{w}+\beta_w)}\frac{n_{z_{dn},-dn}^{x_{dn}}+\eta_{x_{dn}}}{\sum_{r=1}^{R}(n_{z_{dn},-dn}^{r}+\eta_r)}$$