上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR

论文题目　基于多视图信息的大型社交网络数据挖掘

学生姓名　王　灏

学生学号　5090309591

指导教师　李武军

专　　业　计算机科学与技术

学院（系）电子信息与电气工程学院

基于多视图信息的大型社交网络数据挖掘

摘　要

随着近几年来社交网络（social networks）的流行，人们对信息的搜索、获取以及社交的方式都有了较大的改变。从另一个角度讲，它的兴起也为研究者们和工业界的分析积累了足够多的数据。除了国外的 Twitter、Facebook 等与国内的微博这些狭义的社交网络外，我们平时购买的（特别是在网上购买的）商品之间、发表的学术论文等之间也存在着广义的社交网络。比如学术论文之间的引用关系就构成了一个巨大的引用社交网络。而且随着在线社交网络的发展，社交网络不仅产生了许多的随着时间变化的链接（linkage）信息，还生成了许多内容（content）方面的信息。这些都是各种不同视图的信息。如何最大程度地利用社交网络的信息与其他视图的信息为现有的系统（如推荐系统）服务，甚至开发新的服务，是一个值得深入、长期研究的话题。

在本文中，我们分别拓展了协同话题回归（Collaborative Topic Regression，即 CTR）与动态自中心模型（Dynamic Egocentric Models，即 DEM），将社交网络信息，甚至是微细粒度的动态社交网络信息与其他信息无缝地联合建模，提出了关系型协同话题回归（Relational Collaborative Topic Regression，即 RCTR）、带社交正则化的协同话题回归（Collaborative Topic Regression with Social Regularization，即 CTR-SR）与在线自中心模型（Online Egocentric Models，即 OEM）。实验表明新模型大大提高了原模型的推荐准确率。

关键词： 社交网络, 推荐系统, 数据挖掘, 机器学习

Large-Scale Social Network Data Mining with Multi-View Information

ABSTRACT

The development of social networks has changed how people retrieve information and even the way of socialization. Besides, it also provides large amount of available data for academia and industry. Besides the social networks existing in social media like Twitter, Facebook, and Weibo, the networks also exist between the items that people bought, especially online, and between scientific articles. What's more, there is also lots of dynamic linkage information and content information. These are all different views of information. It has become a topic worth digging into how to fully utilize the social network information and other views of information to boost current systems (such as recommendation systems) or even to develop new services.

In this thesis, we extend Collaborative Topic Regression (CTR) and Dynamic Egocentric Models (DEM) to jointly model social network information (even dynamic social network information of extremely fine granularity) and other information and propose Relational Collaborative Topic Regression (RCTR), Collaborative Topic Regression with Social Regularization (CTR-SR), and Online Egocentric Models (OEM). Experiments show that our models outperform the original ones and other state-of-the-art baselines.

Key words: social networks, recommendation systems, data mining, machine learning

目　　录

表格索引

插图索引

第一章　绪论

近年来，随着 Twitter，微博等社交网络的兴起，人们的日常生活发生了巨大的变化。人们通过社交网络传播信息、推荐服务或者商品、结交朋友。社交网络的流行也给学术界与工业界积累了足够多的数据。这些社交网络相关的数据不仅可以用于研究人与人之间的社交关系等，还可以为各种网络服务（比如商品推荐和广告投放）提供宝贵的资料。其实从广义的角度讲，网上商城的商品之间、发表的学术论文之间也存在着社交网络，比如，学术论文之间的社交网络便是文章间复杂的引用关系。除了我们经常看到的静态的社交网络，随着时间的推移，社交网络本身的演变（比如好友关系的建立与消失）也为社交网络分析提供了很重要的一维（也可以称为一个视图，即 view）信息。如何最大程度地利用社交网络的这些多视图信息为现有的系统（如推荐系统）服务，一直是一个值得深入探讨的问题。

另一方面，推荐系统（Recommender systems）在我们日常的信息获取中扮演着十分重要角色。比如，亚马逊（Amazon）[1] 利用推荐系统为用户做商品推荐，Netflix[2] 用它做电影、剧集的推荐。现存的推荐系统可以粗略地分为三类 [3, 4]：基于内容的方法、基于协同过滤（collaborative-filtering-based or CF-based）的方法与混合方法。基于内容（content-based）的方法 [5, 6] 采用用户或产品自身的信息进行推荐。基于协同过滤 [7–10] 的方法利用用户过去对商品的评价或者偏好进行预测。混合方法则是结合了基于内容的方法与基于协同过滤的方法。由于隐私保护等原因，用户的信息变得越来越难以收集，因此基于协同过滤的方法比起基于内容的方法在近些年来更加受到大家的重视。

在大部分传统的协同过滤方法中，只有评分矩阵（记录用户对产品的评分）这种信息被用来训练模型与预测用户喜好。一般来说，评分矩阵都十分稀疏，大部分的用户其实只是评价了非常少数量的产品，这样使得协同过滤的方法的效果受到严重的限制。更加具体地讲，协同过滤是假设如果两个用户过去对产品的评价相似，那么他们往往对产品有着相似的偏好。显然，假如一个用户只评价过一两个产品，推荐系统如果想要为这个用户做推荐是相当困难的。不幸的是，在实际应用中，大部分的用户都只有一两个评价。况且，对

于那些只有一两个评价的新用户来说，提供准确的推荐显得更加重要。新用户能否变成老用户（常客）很大程度上取决于推荐系统推荐的准确度。而对于老用户来说，他们已经对现有的系统足够满意，提高推荐的准确度获得的边际效益其实要小得多。假如我们能够大幅度地提高对新用户的推荐准确度，那么就会有更多的新用户成为老用户，进而便会有更多的数据，从而提高整个推荐系统的准确度。这样一来，整个推荐系统便进入了一种良性循环。

为了缓解基于协同过滤的方法对于稀疏性的敏感，很多研究者提出将附加的信息结合到原有的模型里面。有些方法 [11–13] 利用物品（item）或者产品的内容（属性）来辅助协同过滤的训练。在这些方法里面协同话题回归（collaborative topic regression or CTR）[12] 是最近提出的一个著名模型。它对用户 -物品矩阵（user- item matrix）和物品的属性联合建模，无缝地结合了话题模型 [14]（topic models）与协同过滤，同时提高了模型的预测性能与可解释性（interpretability）。对于未评价的物品，CTR 能够利用物品的属性直接进行矩阵外预测（out-of-matrix prediction or cold start prediction）[13, 15]。另一些方法 [16–18] 则使用了社交网络的信息来提高性能。在这些方法中，CTR-SMF （CTR with social matrix factorization）[18] 通过对 CTR 的扩展，加进了用户之间的社交网络信息, 提高了 CTR 的性能。

在许多实际应用中，除了用户评分（rating）信息与物品的属性信息，还有许多物品之间的网络关系（社交网络信息），这些信息对于精确推荐物品十分有用。比如，假如我们想为 CiteULike （一个文献搜索推荐系统）的用户推荐文献，文献之间的引用网络就能提供很好的信息。因此我们可以建立一个层级的贝叶斯模型（名为关系协同话题回归或 Relational Collaborative Topic Regression，以下简称 RCTR）来将物品关系（item relations）无缝结合到模型中以大幅提高推荐的准确度和可解释性。又或者我们想为 CiteULike 或者其他系统的物品推荐标签（tag），物品之间的关系网络同样可以提供很好的信息，同样，我们也可以建立另一个层级的贝叶斯模型（名为带社会正则化的协同话题回归或 Collaborative Topic Regression with Social Regularization，一下简称 CTR-SR）。虽然乍一看 RCTR 与 CTR-SR 同样是结合了物品间的社交网络信息，然而两个模型有着很大的不同。从贝叶斯建模的角度讲，RCTR 将物品之间的链接关系视为已观察的变量，而 CTR-SR 将这些关系处理为一个拉普拉斯矩阵，作为决定物品隐向量（latent vector）的一个先验

(prior)。从优化算法的角度讲，RCTR 使用直接的 Coordinate ascent 方法，而 CTR-SR 利用了 Steepest descent 讲每步优化的时间复杂度从三次方降到线性。从应用的角度讲，RCTR 倾向于向用户推荐物品，而 CTR-SR 倾向于向物品推荐标签（tag)。

值得注意的是，标签推荐系统（tag recommender systems）是推荐系统中一个十分重要的分支，与普通的推荐系统类似，标签推荐系统使用的方法也大致可以分成三类 [19]: 基于内容的方法 [20–25]、基于同现的方法 [26–34] 和混合方法 [35, 36]。由于标签推荐是一种比较特殊的推荐，单单使用内容或者同现关系都无法得到很好的效果。因此，近来的趋势是使用混合方法进行标签推荐。然而想要将物品 -标签矩阵（item-tag matrix）与物品属性信息有效的结合到标签推荐系统中仍然是一个很大的挑战。再者，在一些应用中可能还存在着许多物品之间的网络信息（社交网络信息)。比如，如果我们希望为 CiteULike 里面的文献加标签（tag)，我们可以认为有引用关系的文献会有更多共同的标签。所以，如果将社交网络信息集成进标签推荐系统是另一个挑战。因此我们首先将 CTR 加以调整后引用与标签推荐系统中，然后在 CTR 的基础上提出了上文的 CTR-SR，旨在解决前文提到的两个标签推荐中的挑战。实验结果表明 RCTR 与 CTR-SR 在推荐上的准确度都明显地超过了 CTR 以及其他基线。

前面提到的社交网络信息属于静态的信息，然而随着大规模动态网络的出现，动态网络分析（Dynamic Network Analysis，即 DNA）已经成为近年来一个非常热门的研究话题。虽然很多动态网络分析的方法已经被各个领域的研究者们提出，但是其中绝大多数只能对极粗的细粒度的动态网络信息进行建模。近来，一些模型被提出来用于大规模的微细粒度的论文引用网络。然而，这些模型的准确度会随着时间的推移而明显地下降，这是因为在模型的预测过程中中，学习到的参数或者节点特征是静态的（固定的)。因此，我们提出了在线自中心模型（OEM）以学习动态引用网络中随着时间变化的参数及节点特征。实验结果表明，OEM 不仅能够防止预测准确率随时间下降，而且能够揭示引用网络中话题随着时间的变化。

下文的组织如下：第二章提出了模型 RCTR，实验表明模型通过与（静态）社交网络信息的结合，很好地解决了 CF、CTR 中的数据稀疏问题。第三章提出了模型 CTR-SR，实验说明通过与（静态）物品网络信息的整合，能很好的解决数据稀疏的问题，达到高于基线的准确率。第四章提出了模型 OEM，实验说明 OEM 通过即时高效地调整动态网络的

参数与特征，使得系统保持较高的预测准确率。第五章是对全文的总结。

第二章　关系型协同话题回归

2.1　引言

推荐系统在我们对信息的有效利用和搜索中扮演着十分重要的角色。比如亚马逊 [1] 利用推荐系统来为它的客户做产品推荐，而 Netflix [2] 也利用推荐系统做电影、剧集的推荐。现今的推荐系统可以粗略地归为三类 [3, 37]：基于内容的方法、基于协调过滤（CF）的方法与混合方法。基于内容的方法 [5, 6, 38] 利用用户或者产品的概要信息做推荐。基于协同过滤的方法 [7–10] 利用用户过去的活动或者偏好（如用户对物品的评分）来做预测，而没有使用任何用户或者产品的概要信息。混合方法则将基于内容的方法和基于协同过滤的方法结合起来。由于隐私保护的问题，一般来说收集用户的概要信息远远比收集用户的过去活动与偏好信息难，因此基于协同过滤的方法在近年来要比基于内容的方法受欢迎得多。

在大多数传统的基于协同过滤的方法中，只有包含用户对物品的评价信息的评价矩阵（rating matrix）被用于训练和预测。一般来说，评价矩阵是十分稀疏的，这意味着大多数用户只评价了极少的物品。由于这个稀疏性的问题，传统的只用到评价矩阵的协同过滤的效果会大打折扣。具体地讲，协同过滤假设对物品有相似评价的用户往往会喜欢相似的物品。因此，如果一个用户只是评价了一两个物品，那么要预测他的喜好则变得十分困难。不幸的是，在现实世界中，我们发现大多数的用户只对一小部分的物品有评价。再者，为目前只有极少数评价的新用户提供精确的推荐要比老用户重要的多，因为新用户是否会继续来这个网站或者继续使用这个服务很大程度上取决于系统推荐的准确度。然而对于老用户来说，我们可以认为他们已经对系统的服务足够满意了（他们为系统提供的信息已经足以让系统做出精确的推荐)。如果我们能够提高对新用户的推荐准确度，那么就会有更多的新客户会变成老客户，这会使得系统积累更多的训练数据从而为全体用户做出更好的推荐。可以说，在极其稀疏的数据设定下提高推荐准确度是使得推荐系统进入良性循环的关键。

为了缓解基于协同过滤的模型的稀疏性问题，许多研究者提出将附加信息整合进模型

的训练和预测的过程中。其中一些方法 [12, 13, 39] 利用物品的内容来加强协同过滤的训练。在这些方法中协同话题回归 (CTR) [12] 是最近一个将用户 -物品评价矩阵（user-item rating matrix）与物品的内容信息（文章的文本）联合建模的模型。协同话题回归无缝地讲话题模型 [40] 与协同过滤结合起来，不仅提高了预测的准确度，而且大大增强了模型的可解释性。对于未被评价的（新的）物品，CTR 可以只使用内容信息而进行矩阵外预测（out-of-matrix prediction or cold-start prediction） [13, 15]. 一些其他的方法 [16–18] 使用用户之间的社交网络关系来提高预测性能。在这些方法中，CTR-SMF [18] 拓展了 CTR，将用户之间的社交网络信息以社交矩阵分解 (SMF)[17] 的方式集成到 CTR 中，达到了优于 CTR 的推荐性能。

在很多真实应用中，除了评价矩阵和物品的内容信息，还可能存在物品之间的关系（或者叫社交网络），这些对提高推荐准确度会很有帮助。比如，如果我们想在CiteULike[1]中为用户推荐学术论文（的引用），学术论文之间的引用关系是十分有用的。在这部分工作中，我们提出一个层级的贝叶斯模型，关系型协同话题回归（RCTR）来将物品之间的社交网络集合到推荐过程中。RCTR 的主要贡献如下：

- 通过拓展 CTR，RCTR 无缝地将用户 -物品评价信息、物品内容信息与物品间的关系（网络）结构整合到一个层级贝叶斯模型中。
- 即使一个新用户只评价了一两个物品，RCTR 仍然能够有效利用物品网络的信息来缓解协同过滤中的数据稀疏问题，从而明显地提高了推荐的准确度。
- 在 RCTR 中，我们提出了一族链接概率函数以对物品之间的关系建模。这个从离散链接概率函数 [41] 到一族连续的链接概率函数的拓展大大增强了 RCTR 的建模能力和预测能力。
- 与 CTR 相比，RCTR 需要更少的迭代次数即能达到令人满意的预测准确度。这使得 RCTR 的总学习时间复杂度要远低于 CTR，即使 RCTR 的每次迭代时间要长于 CTR。
- RCTR 能够学习出一个具有相当高解释性的隐结构，这大大增强了推荐系统的用户体验。
- 在真实数据上的实验表明，RCTR 能够达到比最新模型更高的准确度。

[1] http://www.citeulike.org/

本章的其他部分组织如下。在第 2.2节中，我们将简要地介绍关于 CF 与 CTR 的背景知识。2.3中主要是我们提出的模型 RCTR 的细节。第 2.4节是实验结果。本章的小结则在最后的第 2.5节。

2.2 背景

这一节会先简单地介绍 RCTR 的背景知识，包括基于协同过滤的的推荐、基于矩阵分解（也叫隐变量模型）的 CF 方法 [9, 42] 与 CTR [12]。

2.2.1 基于 CF 的推荐

CF 的任务是基于用户过去的偏好为用户推荐物品。比如，我们可以部署一个为研究者推荐学术论文（引用文献）的推荐系统在 CiteULike 上。在这里，用户则是研究者而物品则是学术论文。与 [12] 类似，我们本章中假设有 I 个用户 J 个物品。用户 i 对物品 j 的评价用 $_{ij}$ 表示。虽然我们的模型可以用于评分为 1 到 5 的整数的情况，我们这里先如 CTR [12] 假设 $r_{ij} \in \{0, 1\}$。这意味着我们的模型的目的是预测一个用户是否喜欢一个物品。在训练数据中，$r_{ij} = 1$ 意味着用户 i 喜欢物品 j。$r_{ij} = 0$ 意味着对应的行为未被观察到（即我们并不知道用户 i 是否喜欢 j）。正如前文所讲，CF 方法只是用了评价矩阵 $\{r_{ij}|i = 1, 2, \cdots, I; j = 1, 2, \cdots, J\}$ 来做训练和预测。

这里提到的预测包含两种预测 [12]：矩阵内预测（in-matrix prediction）与矩阵外预测（our-of-matrix prediction）。矩阵内预测是为至少被用户评价过一次的物品做预测。相反地，矩阵外预测则是为从没有被评价过的物品做预测。矩阵外预测即在其他许多文献中提到的所谓冷启动推荐 [13, 15]。

2.2.2 用于 CF 的矩阵分解

现存的 CF 方法主要可以成两大类 [43]：基于记忆的方法（memory-based methods）[1, 8, 10] 和基于模型的方法 [9, 42, 44]。基于记忆的方法采用相似的用户或者物品的评分的加权平均来做预测，基于模型的方法则是从训练数据中学习出一个统计模型，然后再做预测。许多研究工作已经证实了一般来说，基于模型的方法有比基于记忆的方法更好的性能。因此近年来，基于模型的方法远比另一种方法受欢迎。矩阵分解 [9, 42] 以及其拓展如

概率矩阵分解 [9] 是基于模型的方法中最具代表性的。它们已经被证明了可以在实际应用中达到十分可观的性能。矩阵分解方法的基本思想是使用低维的隐向量来表示用户和物品。具体地说，我们可以一个 K 维的隐向量 $u_i \in R^K$ 来表示用户 i，用一个 K 维的隐向量 $v_j \in R^K$ 来表示物品 j。用户 i 对物品 j 的评价预测值可以用以下式子计算：

$$\hat{r}_{ij} = u_i^T v_j.$$

如果我们用两个隐矩阵 $U = (u_i)_{i=1}^I$ 和 $V = (v_j)_{j=1}^J$ 来分别表示所有的用户和物品的隐向量，矩阵分解的学习过程即是寻找最优的 U 和 V 以最优化下面的目标函数：

$$\min_{U,V} \sum_{i=1}^{I} \sum_{j=1}^{J} (r_{ij} - u_i^T v_j)^2 + \lambda_u \sum_{i=1}^{I} \|u_i\|^2 + \lambda_v \sum_{j=1}^{J} \|v_j\|^2, \tag{2–1}$$

其中 $\|\cdot\|$ 表示一个向量的 Frobenius norm，λ_u 和 λ_v 是控制模型复杂度的正则项系数。目标函数（3–1）对应概率矩阵分解（PMF）模型 [9] 的最大后验 (MAP) 估计。

[12] 中提出了一种 PMF 模型的扩展：

$$\begin{aligned} u_i &\sim \mathcal{N}(0, \lambda_u^{-1} I_K), \\ v_j &\sim \mathcal{N}(0, \lambda_v^{-1} I_K), \\ r_{ij} &\sim \mathcal{N}(u_i^T v_j, c_{ij}^{-1}), \end{aligned} \tag{2–2}$$

其中 $\mathcal{N}(\cdot)$ 表示高斯分布，I_K 是一个有 K 行 K 列的单位矩阵，我们定义 c_{ij} 为：

$$c_{ij} = \begin{cases} a, & \text{if } r_{ij} = 1, \\ b, & \text{if } r_{ij} = 0, \end{cases}$$

其中 a 和 b 是可调的参数且 $a > b > 0$。

矩阵分解方法在实际应用中已经取得的很好的效果。但是，它也有这严重的稀疏性问题。还有另一个缺点就是直接使用矩阵分解方法是无法做矩阵外预测的。

2.2.3 协同话题回归

协同话题回归（CTR） [12] 可以用于为用户推荐科学文献，它将评价矩阵与物品（文章）内容信息无缝地整合起来以解决基于矩阵分解的 CF 遇到的问题。由于整合了矩阵分解和 LDA [40]，CTR 能达到比基于矩阵分解的 CF 高的推荐准确度而且能够得到更有解释性的结果。不仅如此，利用物品的内容使得 CTR 也可以做矩阵外预测。

CTR 的概率图模型如图 2–1。CTR 引进了一个物品隐偏移 ϵ_j 的概念。ϵ_j 在 LDA 中的话题比例（topic proportion）θ_j 与 CF 中的物品隐向量 v_j 之间。这个偏移可以被解释为从文本看文章的主题（用 θ_j 表示）与用户们心目中文章的主题（用 v_j 表示）之间的差别，具体的解释在 [12] 中有提到。如果我们使用 $\boldsymbol{\beta} = \beta_{1:K}$ 来表示 K 个话题，那么 CTR 的产生过程（generative process）如下：

1. 从分布中取得每个用户 i 的隐向量：$u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$。
2. 对于每一个物品 j，
   (a) 取得话题比例 $\theta_j \sim \text{Dirichlet}(\alpha)$。
   (b) 取得物品隐偏移 $\epsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$，令物品隐向量 $v_j = \epsilon_j + \theta_j$。
   (c) 对于文档（物品）$\mathbf{w}_j$ 中的每一个词 w_{jn}，
      i. 取话题分配（topic assignment）$z_{jn} \sim \text{Mult}(\theta)$。
      ii. 取得词 $w_{jn} \sim \text{Mult}(\beta_{z_{jn}})$。
3. 取得每个用户 -物品对 (i, j) 的评价（rating）r_{ij}，

$$r_{ij} \sim \mathcal{N}(u_i^T v_j, c_{ij}^{-1}).$$

正如 [12] 中提到的，CTR 的关键在于物品隐偏移 ϵ_j。这个隐变量使得物品隐向量 v_j 距离话题比例 θ_j 足够接近，但是必要时又可以使两者距离变远。参数 λ_v 控制着 v_j 与 θ_j 的距离。

在 CiteULike 的科学文章推荐上的实验表明，CTR 的准确率要优于基于矩阵分解的 CF 方法。

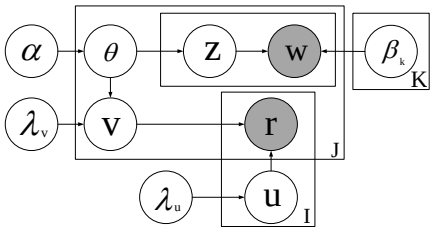


图 2–1 协同话题回归（CTR）的概率图模型。

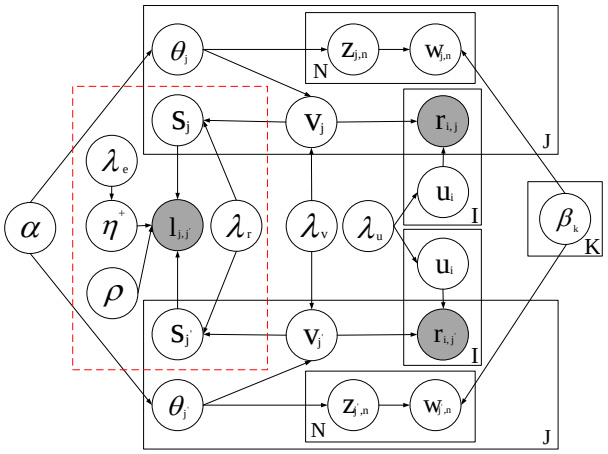


图 2–2 RCTR 的概率图模型，其中 $r_{i,j}$ 与 $r_{i,j'}$ 被观察到的评价，$l_{j,j'}$ 是观察到的物品 j 与物品 j' 之间的关系（0 或 1）。虚线部分是区别 RCTR 与 CTR 的地方。

2.3 关系型协同话题回归

在本节中，我们会具体描述提出的新模型关系型协同话题回归（RCTR）。除了 CTR 建模的评价矩阵和物品内容，RCTR 同时将物品间的社交网络纳入模型中，大大提高了预测准确度。

2.3.1 模型推导

为了更好地展示 RCTR 的概率图，这里使用了一个不同于图 2–1的方法。同样的方法也被 CTR [12] 的作者使用过。RCTR 的概率图如图 2–2。图中的虚线矩形即 RCTR 与 CTR 的区别。

RCTR 的产生过程如下：

1. 取得每个用户 i 的隐向量：$u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$。

2. 对于每一个物品 j，

   (a) 取得话题比例 $\theta_j \sim \text{Dirichlet}(\alpha)$。

   (b) 取得物品隐偏移 $\epsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$，令物品隐向量 $v_j = \epsilon_j + \theta_j$。

   (c) 取得物品关系偏移 $\tau_j \sim \mathcal{N}(0, \lambda_r^{-1} I_K)$，令物品关系向量 $s_j = \tau_j + v_j$。

   (d) 对于文档（物品）$\mathbf{w}_j$ 的每一个词 w_{jn}，

      i. 取得话题分配（topic assignment）$z_{jn} \sim \text{Mult}(\theta)$。

      ii. 取得词 $w_{jn} \sim \text{Mult}(\beta_{z_{jn}})$。

3. 取得参数 $\eta^+ \sim \mathcal{N}(0, \lambda_e^{-1} I_{K+1})$。

4. 对于每一个物品对 (j, j')，取得二进制链接量

$$l_{j,j'} | s_j, s_{j'} \sim \psi(\cdot | s_j, s_{j'}, \eta^+)$$

5. 对于每一个用户 -物品对，取得评价量

$$r_{ij} \sim \mathcal{N}(u_i^T v_j, c_{ij}^{-1})$$

在上面的产生过程中，链接概率函数定义如下：

$$\psi(l_{j,j'} = 1 | s_j, s_{j'}, \eta^+) = \left[\sigma\left(\eta^T (s_j \circ s_{j'}) + \nu\right)\right]^\rho, \tag{2–3}$$

其中 $l_{j,j'}$ 是一个二进制值（只能取 0 或 1），$\sigma(\cdot)$ 表示 sigmoid 函数，ν 是一个标量，表示偏移量，$\eta^+ = \langle \eta, \nu \rangle$ 中 $\langle \cdot \rangle$ 表示将一个标量加到向量的尾部，运算符 $\circ$ 表示对于向量的每个元素的逐个相乘。注意到如果 $\rho = 1$，链接概率函数则退化成关系型话题模型 （RTM）[41] 中提到的链接概率函数之一。

注意上述步骤 2（c）、3、4 即是 RCTR 与 CTR 的区别。步骤 2（c）中的物品关系偏移 τ_j 是 RCTR 的关键性质之一。与物品隐偏移相似，τ_j 使得 s_j 与 v_j 足够接近，但是必要时又能够拉开距离。物品隐向量 v_j 表示用户们心目中物品 j 的话题，而物品关系向量 s_j 表示从物品 j 的社交网络看它的话题是什么。λ_r 越大，v_j 与 s_j 就有越大的概率彼此接

近。当 λ_r 趋向正无穷时，RCTR 就退化成如图 2–3的模型。第 2.4节的实验（图 2–15）表明 RCTR 的准确度要明显好于退化的模型 2–3，从而说明了物品关系偏移 τ_j 的重要性。

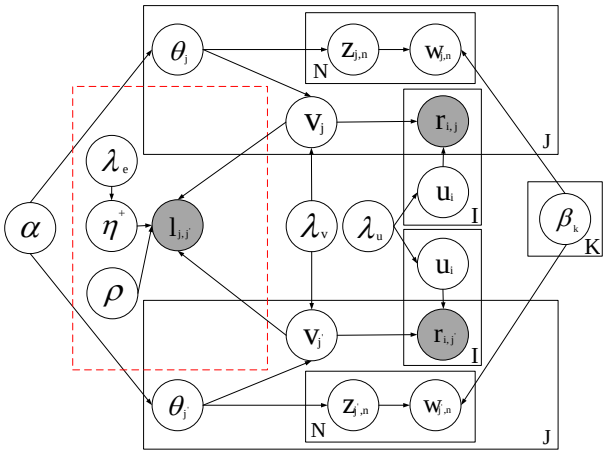


图 2–3 退化的 RCTR 的概率图模型。其中 $r_{i,j}$ 和 $r_{i,j'}$ 是已观察到的评价，$l_{j,j'}$ 是已观察到的物品 j 与 j' 之间的链接关系。虚线框里面的部分是 RCTR 与 CTR 之间的区别。

2.3.2 学习

由于 u_i, v_j, s_j 和 θ_j 的全后验概率无法得到，我们在参数学习中使用了最大后验（MAP）估计。这里的 MAP 估计等价于最大化给定超参数 ρ, λ_u，λ_v，λ_r，λ_e，和 β 时 U，V，$s_{1:J}$，$\theta_{1:J}$ 和评价矩阵的完全 log-likelihood：

$$\begin{aligned}\mathscr{L} = &\rho \sum_{(j,j')} \log \sigma(\eta^T(s_j \circ s_{j'}) + \nu) - \frac{\lambda_u}{2}\sum_i u_i^T u_i \\ &- \frac{\lambda_v}{2}\sum_j (v_j - \theta_j)^T(v_j - \theta_j) \\ &- \frac{\lambda_r}{2}\sum_j (s_j - v_j)^T(s_j - v_j) - \frac{\lambda_e}{2}\eta^{+T}\eta^+ \\ &+ \sum_j \sum_n \log(\sum_k \theta_{jk}\beta_{k,w_{jn}}) - \sum_{i,j}\frac{c_{ij}}{2}(r_{ij} - u_i^T v_j)^2. \end{aligned} \tag{2–4}$$

这里省略了一个常数，话题模型中的超参数 α 被设为 1。这个目标函数可以用交替上升法（coordinate ascent）优化。因为 $\mathscr{L}$ 并不是对所有变量联合凸的，我们设计了一个交替的算法来学习参数。更具体地，每一次我们固定其他参数，只对某一个参数进行优化。

我们取目标函数关于 u_i 和 v_j 的梯度然后将之设为 0，可以得到下面的更新公式：

$$u_i \leftarrow (VC_iV^T + \lambda_u I_K)^{-1} VC_iR_i,$$
$$v_j \leftarrow (UC_iU^T + \lambda_v I_K + \lambda_r I_K)^{-1}(UC_jR_j + \lambda_v\theta_j + \lambda_r s_j),$$

其中 C_i 是一个对角矩阵，元素 $\{c_{ij}|j = 1, \ldots, J\}$，而 $R_i = \{r_{ij}|j = 1, 2, \cdots, J\}$ 是一个包含用户 i 所有评分的列向量。注意 c_{ij} 如 (2–2) 所定义，是一个反映确信度的超参数，由 a 和 b 控制 [45]。

对于 s_j 与 η^+，由于我们无法直接计算 $\mathscr{L}$ 关于它们的梯度，这里使用了梯度上升（gradient ascent）的方法来更新变量。$\mathscr{L}$ 关于 s_j 梯度是：

$$\nabla_{s_j}\mathscr{L} = \rho \sum_{l_{j,j'}=1} (1 - \sigma(\eta^T(s_j \circ s_{j'}) + \nu))(\eta \circ s_{j'}) - \lambda_r(s_j - v_j).$$

其中 $\pi^+_{j,j'} = \langle s_j \circ s_{j'}, 1\rangle$。

对于 θ_j，我们先如 [12] 令 $q(z_{jn=k}) = \psi_{jnk}$。然后在分离出关于 θ_j 的项后使用 Jensen 不等式，

$$\begin{aligned}\mathscr{L}(\theta_j) &\geq -\frac{\lambda_v}{2}(v_j - \theta_j)^T(v_j - \theta_j) \\ &\quad + \sum_n \sum_k \phi_{jnk}(\log \theta_{jk}\beta_{k,w_{jn}} - \log \phi_{jnk}) \\ &= \mathscr{L}(\theta_j, \boldsymbol{\phi}_j).\end{aligned}$$

这里 $\boldsymbol{\phi}_j = (\phi_{jnk})^{N\times K}_{n=1,k=1}$。显然 $\mathscr{L}(\theta_j, \boldsymbol{\phi}_j)$ 是 $\mathscr{L}(\theta_j)$ 的一个紧下界，我们可以使用投影梯度（projection gradient）来优化 θ_j。ϕ_{jnk} 的最优值为

$$\phi_{jnk} \propto \theta_{jk}\beta_{k,w_{jn}}.$$

至于参数 β 的学习，则与 LDA 中的 M 步的更新相同，

$$\beta_{kw} \propto \sum_j \sum_n \phi_{jnk} 1[w_{jn} = w].$$

2.3.3 预测

令 D 为已观察数据（observed data）。类似于 [12]，我们使用 u_i, θ_j 和 ϵ_j 的点估计来计算评价的预测值:

$$E[r_{ij}|D] \approx E[u_i|D]^T (E[\theta_j|D] + E[\epsilon_j|D]),$$

其中 $E(\cdot)$ 表示取期望的运算。

对于矩阵内预测

$$r^*_{ij} \approx (u^*_j)^T (\theta^*_j + \epsilon^*_j) = (u^*_i)^T v^*_j.$$

对于矩阵外预测

$$r^*_{ij} \approx (u^*_i)^T \theta^*_j.$$

2.3.4 时间复杂度

根据 RCTR 学习过程中的更新规则，可以知道每次迭代中，更新 η 的时间复杂度为 $O(KQ)$，其中 K 是隐变量空间的维数，Q 是社交网络中链接的个数。更新物品关系矩阵 $S = \{s_j|j = 1, 2, \cdots, J\}$ 的花费也是 $O(KQ)$。更新其他变量的时间复杂度与 CTR 中的相同。对于 U，时间复杂度是 $O(IK^3 + IJK^2)$，对于 V 时间复杂度是 $O(JK^3 + IJK^2)$，其中 I 为用户的个数，J 为物品的个数。在每次迭代中，RCTR 相对 CTR 多了 $O(KQ)$ 的时间复杂度。由于一般来说，物品间的社交网络是十分稀疏的，这意味着 Q 可以当成是 J 的常数倍，相对于 CTR，RCTR 多出的时间开销非常小。

从实验可以发现，相比 CTR，RCTR 需要更少的迭代次数即可达到令人满意的预测效果。这意味着即使乍一看，RCTR 的每个迭代的时间复杂度要大于 CTR，然而前者的总学

习时间要比后者还要短。这个结果会在第 2.4.6节中具体阐述。

2.3.5 关于链接概率函数的讨论

除了物品关系偏移，另一个 RCTR 的关键性质是链接概率函数族的使用。[41] 中发现不同的链接概率函数能够有不同的预测性能。于是在 RCTR 中我们使用单个参数 ρ 来控制链接概率函数的选择。由于 ρ 是一个非负实数，链接概率函数族实际上包含着无数个候选的链接概率函数。相对于 [41] 的两个链接概率函数，这里提出的链接概率函数族能够大大地增强 RCTR 建模的能力，从而使得它有更高的预测准确度。从优化的角度，ρ 可以被当成是一个控制链接与其他观察量的权衡的超参数。ρ 取不同值时链接概率函数的差别如图 2–4所示。从图中可以看出，我们提出链接概率函数族足够灵活，能够对各种情况进行建模。

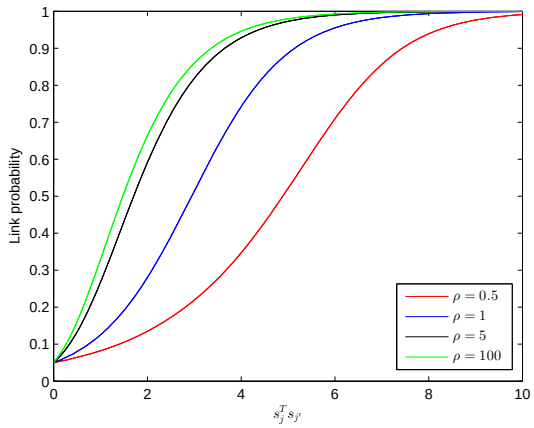


图 2–4 ρ 取不同值时各个链接概率函数的比较。曲线显示的是 $l_{j,j'} = 1$ 的概率关于 s_j 和 $s_{j'}$ 的内积的函数。η 被固定为 1，ν 被调整到使得所有函数的起始点相同。

2.4 实验

我们设计了几组实验以比较 RCTR 与其他最先进的基线在两个数据集上的预测性能。我们想要通过实验回答的问题是：

- RCTR 的预测性能能够比最先进的基线好多少，特别是当数据极其稀疏的时候？
- 链接概率函数族对预测性能的影响有多大？

- 关系参数 λ_r 及其他参数如何影响模型的预测性能？

2.4.1 数据集

我们使用了两个数据集来进行实验。两个数据集都是来自于 CiteULike[2]，然而它们是用不同的方法搜集的，规模大小与稀疏度的大小都不同。第一个数据集 *citeulike-a*来自 [12]。注意原来的数据集不包含物品之间的社交网络。我们直接从 CiteULike 与 Google Scholar[3] 收集网络信息。第二个数据集 *citeulike-t*是我们自己独立于第一个数据集收集的。我们先手动选择了 273 个种子标签，然后收集所有带这些标签的文章。同时，我们还在 Google Scholar 上爬取了这些文章之间的引用关系。注意最终所有文章的标签数是远远大于 273 的。与 [12] 相似，我们先去掉评价物品少于 3 的用户以得到一个更大、更稀疏的数据集。两个数据集的描述详见表 2–1。可以看出我们收集的数据集 *citeulike-t*中的用户数与物品数都远大于 *citeulike-a*。而且，用户 -物品的评价矩阵中，*citeulike-a*与 *citeulike-t*中非空元素（等于 1-稀疏度）所占的比例分别是 0.0022 和 0.0007。这意味着第二个数据集要远远比第一个稀疏。

*citeulike-a*的文本信息是用与 [12] 相同的过程预处理的。与 [12] 相同，我们使用标题与摘要作为 *citeulike-t*的文本信息。在去除了停词之后，我们根据 tf-idf 值选取了 20000 个词作为词典。

除了论文中的引用关系，我们发现文章的标签信息量也比较大。所以，我们同时使用了引用关系与标签来建立单个物品间的社交网络（图）。对于每个数据集，我们先构造一个阈值为 4 的标签图，具体地说，即是如果两篇文章有大于等于 4 个共同标签，那么在标签图中它们之间就存在着一条边。然后我们对标签图与引用图施加 OR 运算，得到最终的社交网络。在最终网络中链接的总个数见表 2–1的最后一行。

2.4.2 评价标准

对每个数据集，我们为每个用户随机选取 P 个评价的物品以构造训练集，使用其他的数据用于测试集，正如前文第 2.1节所说，为新用户提供更准确的推荐的意义要比为老

[2]CiteULike 中，用户可以自己建立自己的论文集。每篇文章都有摘要、标题、标签等信息。其它信息如作者、群组、上传时间、关键词等没有出现在这部分工作中，我们不予讨论。详情可见 http://www.citeulike.ort/faq/data.adp

[3]大多数 CiteULike 上的文章没有提供引用信息，这意味着我们必须自己收集论文间的社交网络信息。在本章中，引用信息是从 Google Scholar 即 http://scholar.google.com 中爬取的

表 2–1 数据集描述

	citeulike-a	citeulike-t
#users	5551	7947
#items	16980	25975
#tags	19107	52946
#citations	44709	32565
#user-item pairs	204987	134860
sparsity	99.78%	99.93%
#relations	549447	438722

用户的大得多。因此我们更感兴趣的是，在极其稀疏的设定下，推荐算法的性能如何。实验中 P 取值 1 到 10，P 越小，表示训练集越稀疏。注意到当 $P = 1$ 时，只有 2.7% 的非空的评价矩阵元素被放入数据集 *citeulike-a*。对于 *citeulike-t*同样的数字是 5.8%。对于每个 P 我们重复评测 5 遍，每遍中的训练数据都是重新随机选取的，最终报告的是平均的性能。

如 [12] 和 [18] 中提到的，我们使用召回率（recall）作为推荐效果的评价标准，因为评价（评分）为零可能是因为用户不喜欢某一个物品，也可能是因为用户不知道这个物品的存在，这意味着精确率（precision）不是一个合适的标准。正如大多数推荐系统所做的，我们将一个用户对各个物品评分的预测值排序后，为用户推荐前 M 个物品（文章)。对于每个用户 recall@M 定义为

$$\text{recall@M} = \frac{\text{前 M 个物品中用户喜欢的个数}}{\text{用户喜欢的总物品数}}$$

最后展示的结果是所有用户的平均召回率。

上面的公式计算的是面向用户的召回率，相似的，我们也可以为目标物品推荐用户。在这样的设定下，面向物品的召回率定义如下:

$$\text{i-recall@M} = \frac{\text{前 M 个用户中喜欢这个物品的人数}}{\text{喜欢物品的总人数}}$$

2.4.3 基线与实验设置

我们使用 CTR [12] 与基于矩阵分解的 CF [42] 作为基线。在用一个验证数据集（validation set）找到 CTR 和 CF 最优的超参数 λ_u 和 λ_v 后，我们固定 λ_u 和 λ_v 为 CTR 的最优值，然后调其他的超参数。

使用 grid search，我们可以发现当 $\lambda_v = 100$、$\lambda_u = 0.01$、$a = 1$、$b = 0.01$、$K = 200$ 时 CF 与 CTR 达到了很好的预测性能。这里的 a 与 b 控制着确定度参数 c_{ij}。对于 RCTR，我们令参数 $\lambda_v = 100$、$\lambda_u = 0.01$、$a = 1$、$b = 0.01$、$K = 200$ 然后改变其他参数（包括控制链接概率函数的参数 ρ）以研究链接概率函数的选择如何影响预测性能并研究模型对于参数 λ_r 与 λ_e 的敏感度。

2.4.4 性能

正如第 2.4.2小节描述的，我们使用两个不同的推荐设置：面向用户的推荐与面向物品的推荐。

2.4.4.1 面向用户的推荐

面向用户的推荐试图为目标用户推荐物品。我们比较了 RCTR 与 CF、CTR 以观察 RCTR 的性能可以超过基线多少。图2–5与图2–6分别是数据集 *citeulike-a*与 *citeulike-t*上当 P 为 1、2、5、8、10 时的 recall@300[4]。对于数据集 *citeulike-a*我们的模型 RCTR 一致地取得比 CTR 更好的预测性能，超过 CTR1.4% $\sim$ 5.0%。对于数据集 *citeulike-t*，两者之间的差别在 2.8% $\sim$ 8.5%。CF 在 P 很小是表现很差，当 P 变大时 CF 的性能逐渐变好。recall@300 的随机基线在两个数据集上分别是（即不用任何算法，进行随机的推荐）1.77% 与 1.15%。正如图中所示，虽然 CF 比随机基线好，但是 CF 由于稀疏问题性能比较差。而 CTR 由于使用了物品内容信息，能够取得比 CF 好的性能，我们的 RCTR 通过将物品社交网络整合进模型，是的推荐性能有进一步的明显提高。

考虑到我们划分训练集与测试集的方法，CTR 与 RCTR 在本章中被设置为当有未评价物品时自动进行矩阵外预测。例如，当使用 *citeulike-t*、$P = 1$ 时，接近 69% 的物品是未被评价的，这意味着大部分的预测（推荐）都是矩阵外预测。

图2–7与图2–8是当 P 固定为 1、M = 50，100，150，200，250，300 时 RCTR、CTR 与 CF 的召回率。对于 RCTR$\lambda_r = 1$、$\lambda_e = 1000$、$\rho = 100$。在这里 RCTR 又一次一致而且明显地超过 CTR 与 CF。相似的现象在 P 为其他值是也可以发现，这里由于篇幅所限不予讨论。

[4]注意由于所有设置下的标准差就很小（在 $[3.53 \times 10^{-5}, 6.59 \times 10^{-3}]$ 的范围内），为了避免混乱，本章中的图不会单独报告标准差

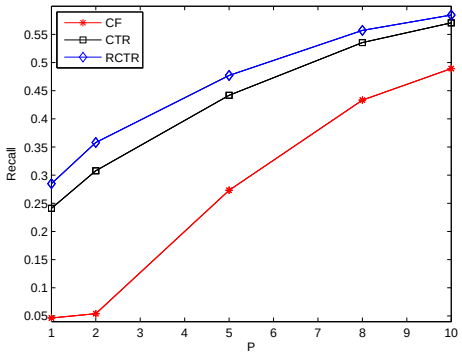


图 2–5 RCTR、CTR 与 CF 的面向用户的 recall@300。P 取值在 1 到 10 之间。使用的数据集是*citeulike-a*。随机基线是 1.77%。

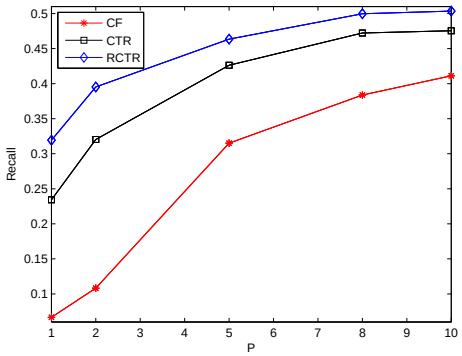


图 2–6 RCTR、CTR 与 CF 的面向用户的 recall@300。P 取值在 1 到 10 之间。使用的数据集是*citeulike-t*。随机基线是 1.15%。

2.4.4.2 面向物品的推荐

面向物品的推荐试图为目标物品推荐合适的用户。比如对于科学文献，这个可以被应用于推荐 coauthor 与 reviewer。这里图 2–9、图 2–10、图 2–11与图 2–12报告了面向物品推荐的性能。

与面向用户的推荐的结果相似，RCTR 的性能一致地明显地高于 CTR 与 CF。

2.4.5 参数敏感度

图2–13说明了链接概率函数的选择如何影响预测性能。令 $\lambda_r = 1$、$\lambda_e = 1000$、$P = 1$，令 $\rho = 1, 10, 100, 1000, 10000$ 分别计算 recall@M。当 $\rho = 1$ 时，链接概率函数等价于 [41] 提出的函数之一。可以看出，在五个选择中，$\rho = 100$ 对应着最优的链接概率函数。当 ρ 太小时，RCTR 的性能与 CTR 十分接近。注意 [41] 中提出的链接概率函数在 5

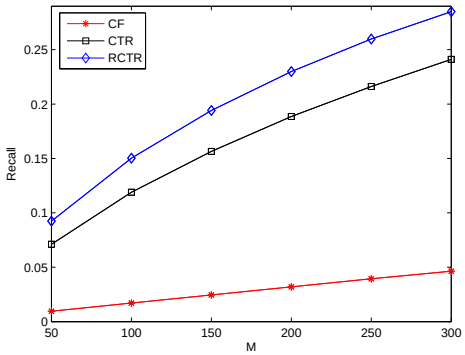


图 2–7 RCTR、CTR 与 CF 的面向用户的召回率。M 取值 50 到 300 之间。使用的数据集为 *citeulike-a*。P 固定为 1。相似的现象在 P 取其它值时也可以观察到。

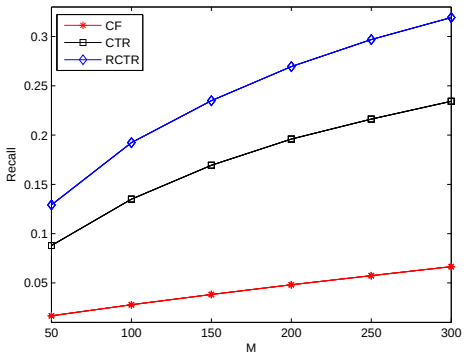


图 2–8 RCTR、CTR 与 CF 的面向用户的召回率。M 取值 50 到 300 之间。使用的数据集为 *citeulike-t*。P 固定为 1。相似的现象在 P 取其它值时也可以观察到。

个中预测性能是最差的，这也从一个角度说明了选择最优的链接概率函数的重要性。而 RCTR 提供了一个足够灵活可调的链接概率函数族。

为了研究 RCTR 对参数 λ_r 与 λ_e 的敏感度，我们进行了两组实验，使用的是 $P = 1$ 的训练数据。首先令 $\lambda_r = 1$ 并研究预测性能如何随着 λ_e 变化。图2–14即是数据集 *citeulike-t*上的 recall@300。可以发现对于 λ_e 的变化 RCTR 的性能基本保持稳定。如果令 $\lambda_e = 1000$ 然后研究 λ_r 如何影响预测性能。图2–15即是对应的 recall@300，使用的数据集同样是 *citeulike-t*。可以发现 RCTR 的预测性能对于 λ_r 较敏感。对于固定的 ρ，召回率先随着 λ_r 的增大而上升，之后在接近 $\lambda_r = 1$ 的地方开始下降。预测性能在 λ_r 太大是一直保持着较低的水平。更大的 λ_r 意味着物品关系向量 s_j 会更加接近物品隐向量 v_j。当 $\lambda_r = 0$ 时 RCTR 退化为 CTR，当 $\lambda_r = \infty$，RCTR 退化成图 2–3的模型。一个有趣的性质是，最好的性能总是在 $\lambda_r = 1$ 附近达到，无论其他的参数如何改变。因此，我们在实验中将 λ_r

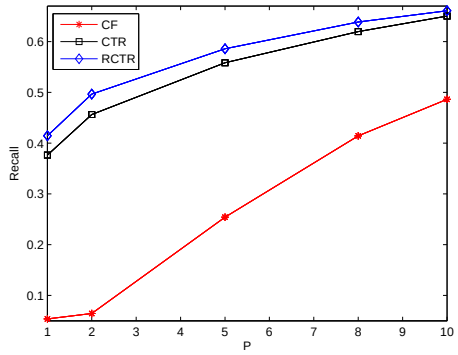


图 2–9 RCTR、CTR 与 CF 的面向物品的 i-recall@300。P 取值于 1 到 10 之间，使用的数据集是 *citeulike-a*。随机基线的是 5.40%。

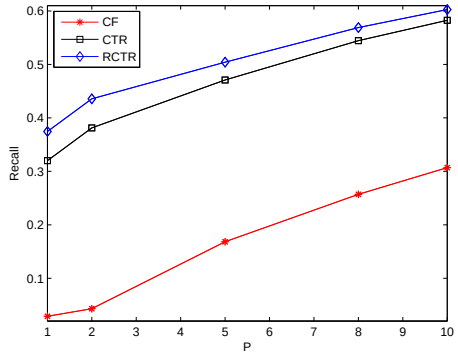


图 2–10 RCTR、CTR 与 CF 的面向物品的 i-recall@300。P 取值于 1 到 10 之间，使用的数据集是 *citeulike-t*。随机基线的是 3.78%。

固定为 1。

2.4.6 计算时间

表2–2与表2–3中是 CTR 与 RCTR 的平均训练时间（与标准差），单位为秒。如第 2.3.4节所说，虽然在 RCTR 中有更多的信息输入，每次迭代需要更多的时间，它的总时间复杂度依然要远低于 CTR。主要的原因是要达到令人满意的预测性能，RCTR 所需的迭代次数要比 CTR 的少。

表 2–2 数据集 *citeulike-a* 的训练时间 (秒)

P	1	2	5	8	10
CTR	3387 ± 582	5336 ± 434	11136 ± 1124	16750 ± 529	18931 ± 264
RCTR	2655 ± 295	3009 ± 246	2832 ± 336	2891 ± 323	2655 ± 295

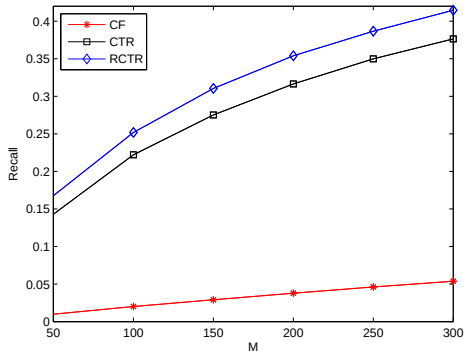


图 2–11 RCTR、CTR 与 CF 的面向物品的召回率。M 取值 50 到 300 之间。使用的数据集为 *citeulike-a*。P 固定为 1。相似的现象在 P 取其它值时也可以观察到。

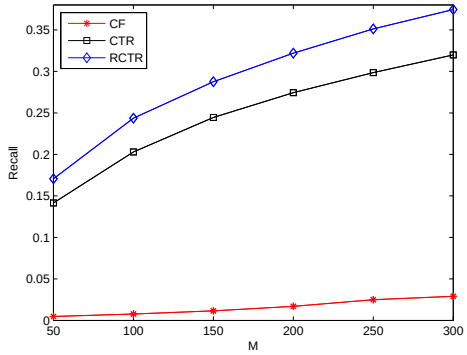


图 2–12 RCTR、CTR 与 CF 的面向物品的召回率。M 取值 50 到 300 之间。使用的数据集为 *citeulike-t*。P 固定为 1。相似的现象在 P 取其它值时也可以观察到。

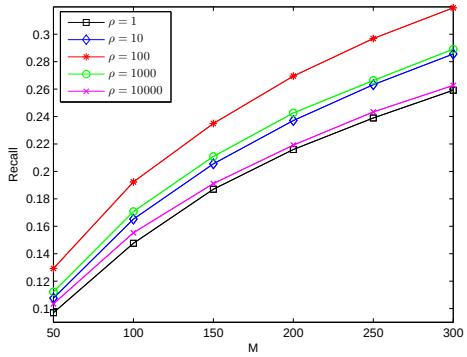


图 2–13 当 M 取值 50 到 300 之间时参数 ρ 对 RCTR 的影响。使用的数据集是 *citeulike-t*。P 设为 1。$\lambda_v = 100$、$\lambda_u = 0.01$、$\lambda_r = 1$、$\lambda_e = 1000$。

表 2–3 数据集 *citeulike-t*的训练时间 (秒)

P	1	2	5	8	10
CTR	16250 ± 1114	15251 ± 1010	24442 ± 379	28638 ± 706	28971 ± 706
RCTR	5285 ± 657	5140 ± 594	7095 ± 412	6806 ± 472	7095 ± 412

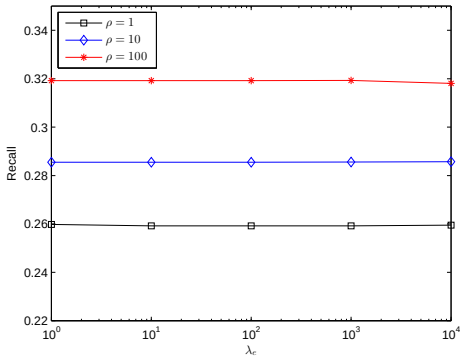


图 2–14 参数 λ_e 对 RCTR 的影响。使用的数据集是 *citeulike-t*。P 设为 1。$\lambda_v = 100$、$\lambda_u = 0.01$、$\lambda_r = 1$。

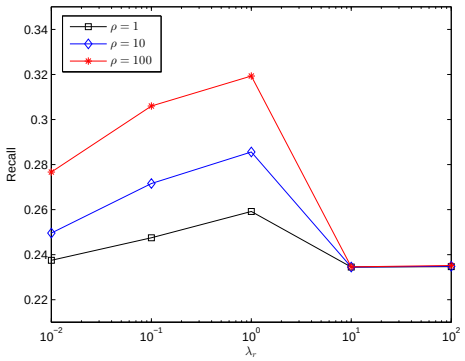


图 2–15 参数 λ_r 对 RCTR 的影响。使用的数据集是 *citeulike-t*。P 设为 1。$\lambda_v = 100$、$\lambda_u = 0.01$、$\lambda_e = 1000$。

2.4.7 可解释性

RCTR 的学习结果有很好的解释性。更具体地说，用户的隐向量可以被解释为从数据中学习到的一些话题。为了更加深入地了解 RCTR，这里展示了两个实例用户的概要信息，其中有对应用户的前三个匹配话题及被 RCTR 与 CTR 分别推荐的文章列表。在这个实例中，我们在极其稀疏的条件 $P = 1$ 下训练 RCTR 与 CTR 后，为用户推荐文章。注意在训练数据中，每个用户是评价了 1 篇文章，这使得准确的推荐十分具有挑战性。正如表2–4所示，用户 I 是一个研究神经科学的研究者，这个可以从 RCTR 返回的第一个与 CTR 返回的第三个话题看出。对于用户 I，RCTR 与 CTR 在前 10 篇文章中的精确率（precision）分别为 80% 与 30%。相似地，可以发现用户 II 是一个研究 RNA 的生物学家。RCTR 与 CTR 的精确率分别为 90% 与 40%。

仔细观察训练数据可以知道，用户 I 只评价了一篇名为“*Neural Correlations, Population Coding and Computation*”的文章。在全部 8 篇 RCTR 推荐正确的文章中，有 6 篇是与

已评价文章“*Neural Correlations, Population Coding and Computation*”有直接的链接，这意味着 RCTR 成功地将社交网络信息整合到模型中并大大提高了预测性能。相似地，在训练集中用户 II 评价的文章是“*A Combined Computational-Experimental Approach Predicts Human MicroRNA Targets*”。在 RCTR 全部 9 篇正确推荐的文章中，有 4 篇是直接与它有链接的。更具体地，有 3 篇是在标签图与引用图都连向“*A Combined Computational-Experimental Approach Predicts Human MicroRNA Targets*”的，有 1 篇是只在标签图中与它相连。

2.5 本章小结

本章中我们为推荐系统提出了一个新型的层级贝叶斯模型（RCTR）。RCTR 能够无缝地将用户 -物品评价信息、物品内容信息与物品间社交网络整合进同一个模型中。RCTR 能够很好地利用附加的信息以缓解传统的 CF 方法与 CTR 面临的稀疏问题。在真实数据集上进行的实验表明，我们的模型能够达到比最先进方法更高的预测准确度，却只需要更低的时间复杂度。再者，RCTR 还能够提供可解释的训练结果，这对于推荐系统来说是十分有用的。

RCTR 的贝叶斯构造使得我们可以将之加以改进并对不止一个物品间社交网络进行建模。比如，我们可以对标签图与引用图分别建模，而不是将它们先合并成一个单一的图。这会在我们未来的工作中实现并严重。另外，从优化的过程可知，RCTR 的算法可以比较容易地被并行化以用来处理极大的数据集。

此章的工作已经整理成文并将投稿于国际顶级期刊 TKDE（IEEE Transactions on Knowledge and Data Engineering）。

表 2–4 学习出来的隐结构的可解释性

	user I (RCTR)	in user's lib?
top 3 topics	1. activity, neural, neurons, cortex, cortical, neuronal, stimuli, spike, visual, stimulus 2. processing, conditions, sensitivity, perception, music, sound, filters, filter, simultaneous, auditory 3. positive, correlation, hypothesis, negative, correlations, bias, intrinsic, costs, codon, aggregation	
top 10 articles	1. The variable discharge of cortical neurons	yes
	2. Refractoriness and neural precision	no
	3. Neural correlates of decision variables in parietal cortex	yes
	4. Neuronal oscillations in cortical networks	yes
	5. Synergy, redundancy, and independence in population codes	yes
	6. Entropy and information in neural spike trains	no
	7. The Bayesian brain: the role of uncertainty in neural coding and computation	yes
	8. Activity in posterior parietal cortex is correlated with the relative subjective desirability of action	yes
	9. Psychology and neurobiology of simple decisions	yes
	10. Role of experience and oscillations in transforming a rate code into a temporal code	yes
	user I (CTR)	in user's lib?
top 3 topics	1. coding, take, necessary, place, see, regarding, reason, recognized, mediated, places 2. genetic, variation, population, populations, variants, snps, individuals, genetics, phenotypes, phenotypic 3. activity, neural, neurons, cortex, cortical, neuronal, stimuli, spike, visual, stimulus	
top 10 articles	1. Chromatin modifications and their function	no
	2. Mistranslation-induced protein misfolding as a dominant constraint on coding-sequence evolution	no
	3. Lateral habenula as a source of negative reward signals in dopamine neurons	yes
	4. Two types of dopamine neuron distinctly convey positive and negative motivational signals	no
	5. Proportionally more deleterious genetic variation in European than in African populations	no
	6. The primate amygdala represents the positive and negative value of visual stimuli during learning	yes
	7. Genetic variation in an individual human exome	no
	8. Behavioural report of single neuron stimulation in somatosensory cortex	no
	9. Reward-dependent modulation of neuronal activity in the primate dorsal raphe nucleus	no
	10. Uniform inhibition of dopamine neurons in the ventral tegmental area by aversive stimuli	yes
	user II (RCTR)	in user's lib?
top 3 topics	1. sites, target, site, targets, mirnas, predicted, mirna, conserved, seed, figure 2. rna, mirnas, mirna, rnas, mrna, microrna, microrna, translation, mir, mrnas 3. human, identification, humans, targeted, curves, curve, assay, roc, uniquely, receiver	
top 10 articles	1. Combinatorial microRNA target predictions	yes
	2. Prediction of mammalian microRNA target	yes
	3. Conserved seed pairing indicates that thousands of human genes are microRNA targets	yes
	4. Animal microRNAs confer robustness to gene expression	yes
	5. Silencing of microRNAs in vivo with 'antagomirs'	no
	6. Microarray analysis shows that some microRNAs downregulate large numbers of target mRNAs	yes
	7. Mammalian microRNAs derived from genomic repeats	yes
	8. Identification of hundreds of conserved and nonconserved human microRNAs	yes
	9. The widespread impact of mammalian microRNAs on mRNA repression and evolution	yes
	10. A microRNA polycistron as a potential human oncogene	yes
	user II (CTR)	in user's lib?
top 3 topics	1. rna, mirnas, mirna, rnas, mrna, microrna, microrna, translation, mir, mrnas 2. sites, target, site, targets, mirnas, predicted, mirna, conserved, seed, figure 3. human, identification, humans, targeted, curves, curve, assay, roc, uniquely, receiver	
top 10 articles	1. Regulation by let-7 and lin-4 miRNAs results in target mRNA degradation	no
	2. Getting to the root of miRNA-mediated gene silencing	no
	3. Prediction of mammalian microRNA target	yes
	4. Conserved seed pairing indicates that thousands of human genes are microRNA targets	yes
	5. Animal microRNAs confer robustness to gene expression	yes
	6. Large-scale sequencing reveals 21U-RNAs and additional microRNAs and endogenous siRNAs in C. elegans	no
	7. MicroRNA control in the immune System: basic principles	no
	8. Concordant regulation of translation and mRNA abundance for hundreds of targets of a human microRNA	no
	9. Endogenous siRNAs from naturally formed dsRNAs regulate transcripts in mouse oocytes	yes
	10. Dual role for argonautes in microRNA processing	no

第三章　带社会正则化的协同话题回归

3.1　引言

标签系统在分类和组织系统方面扮演着重要的角色。比如说，Flickr 使用标签来组织分类图片，Last.fm 利用标签来分类艺术家和音乐。CiteULike 允许用户对文章标签。通过标签系统，用户可以更好地组织他们的信息，更容易的找到相关物品或信息。

然而，找到准确的描述物品的标签是件很难的事。正因如此，标签推荐 (TR)[19, 46] 变的越加重要。通过标签推荐系统，用户只需很少的点击来完成标注过程。并且，不同用户生成的标签可能千差万别。不同的用户可能使用不同的文字来形容相同的意向，这些都给使用标签造成了障碍。标签推荐系统可以帮助缩小词汇范围，从而减轻这类问题。同时它可以帮助排除一些错拼和无意义的单词。因此，TR[19] 最近成为了非常热的话题。

现有的标签推荐方法可以简单分成三类 [19]：基于内容的方法，co-occurrence 的方法，混合方法。基于内容的方法 [20–25] 直接利用物品的内容信息来做推荐，比如论文的摘要和内容，图像信息和图像描述。基于 co-occurrence 的方法 [26–34] 主要利用 tag 在物品中共同出现的次数的记录来做标签推荐。事实上，co-occurrence 方法背后的原理和协同过滤方法（CF）相似。因为 TR 问题非常复杂和难，无论是纯粹的基于内容的方法还是基于 co-occurrence 的方法都无法取得满意的效果。因此最近的趋势是使用混合的方法 [33, 35, 36, 47]，这些方法同时利用了物品 -标签矩阵和物品的个体信息来做推荐。

在一些应用中，我们也许可以得到物品之间的网络关系。比如说，如果我们要在 CiteULike 里对文章进行标注，文章之间会有引用信息。通常两篇有相关联系的文章更有可能是关于同意的话题的，从而也更有可能有相同的标签。因此，如何有效的整合物品之间的社交网络信息成为了一个新的挑战。

在本章中，我们提出了一个新的模型来解决这个挑战。文章主要的贡献在于以下几点：

- 我们利用协同话题回归（CTR）模型作为基础模型，成功的结合了物品 -标签矩阵和物品的内容信息。

- 通过对 CTR 进行拓展，提出了一种层级式的贝叶斯模型，称为有社交正则化的 CTR 模型（CTR-SR）。它有效地整合了物品 -标签矩阵, 物品内容信息，并利用了物品之间的网络关系。

- 对现实数据的实验显示，CTR 模型相比只单独使用内容或 co-occurrence 信息的方法要更优，而 CTR-SR 可以有效的利用物品之间的社交网络，从而进一步改善表现。

3.2 问题描述

假设我们有一个需要标注的物品集合 $W = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_J]$，其中 $\mathbf{w}_j \in \mathbb{R}^d$ 表示了物品 j 的内容或属性。比如说，如果我们希望标注文章，那么物品即文章，而内容可有是文章的摘要。假设有 I 个标签 $\{\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_I\}$。那么我们可以用矩阵 $R = [r_{ij}]_{I \times J}$ 来代表所有物品的标签信息。r_{ij} 是个二元变量，其中 $r_{ij} = 1$ 表示物品 $\mathbf{w}_j$ 有标签 $\mathbf{t}_i$。标签推荐任务即预测 $r_j = [r_{1j}, r_{2j}, \cdots, r_{Ij}]^T$ 中的未知值。注意我们本文关注的是对文章的标注问题。然而，我们的模型同样可以被应用到图片和影像的标注任务上。

基于内容的方法只使用了内容信息来进行推荐。比如说，如果我们希望给物品 $\mathbf{w}_j$ 推荐标签，我们可以使用与 $\mathbf{w}_j$ 内容最相近的物品同样的标签。我们也可以把每一个标签作为 label, 然后通过基于内容来训练分类器的方法来进行推荐。

基于 co-occurrence 的方法只使用了矩阵 R 来做推荐。比如说，如果 $\mathbf{t}_i$ 与 $\mathbf{t}_k$ 同时在多篇文章中作为 tag，并且已知 $\mathbf{t}_i$ 是 $\mathbf{w}_j$ 的 tag，那么我们也应该给 $\mathbf{w}_j$ 推荐 $\mathbf{t}_k$。可以看到 co-occurrence 方法背后的原理与协同过滤的方法 (Collaborative filtering) 非常相似。

不管是 co-occurrence 方法还是基于内容的方法，都忽略了一些有用的信息，因此，它们在应用时无法达到足够令人满意的效果。

3.3 协同话题回归

协同话题回归预测（CTR）模型 [12] 可以用来做推荐。CTR 模型最初通过利用用户 -文章的评价信息和文章内容，来对文章进行推荐。这篇文章中，我们将 CTR 运用到标签推荐中来。

CTR 的图模型如图3–1。假设有 K 个话题 $\boldsymbol{\beta} = \beta_{1:K}$，CTR 模型的生成过程如下:

1. 为每个标签生成隐含变量:

$$u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K),$$

2. 对于每个物品 j:

   (a) 生成主题分布 $\theta_j \sim \text{Dirichlet}(\alpha)$。

   (b) 生成物品的隐含偏移量 $\epsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$，并且设置物品的隐含变量为 $v_j = \epsilon_j + \theta_j$。

   (c) 对于文章 $\mathbf{w}_j$ 的每个单词 w_{jn}:

      i. 生成话题 $z_{jn} \sim \text{Mult}(\theta_j)$。

      ii. 生成单词 $w_{jn} \sim \text{Mult}(\beta_{z_{jn}})$。

3. 对于每个标签 -物品对 (i, j)，生成标签信息:

$$r_{ij} \sim \mathcal{N}(u_i^T v_j, c_{ij}^{-1}), \tag{3–1}$$

其中 c_{ij} 反映了 r_{ij} 的置信度:

$$c_{ij} = \begin{cases} a, & \text{if } r_{ij} = 1, \\ b, & \text{if } r_{ij} = 0, \end{cases}$$

其中 a 与 b 是参数，$a > b > 0$。

我们可以采用最大后验估计的方法 (MAP) 来学习 CTR 的参数。

容易看到上述过程结合了基于矩阵分解的协同过滤方法 [48] 和话题模型方法 (Topic Model)。

3.4 带社交正则化的协同话题回归

通过拓展 CTR，我们提出了一个层级化的贝叶斯模型，成为有社交正则化的 CTR 模型 (CTR-SR)，用以无缝地整合物品 -标签矩阵，物品内容信息，以及物品之间的社交网

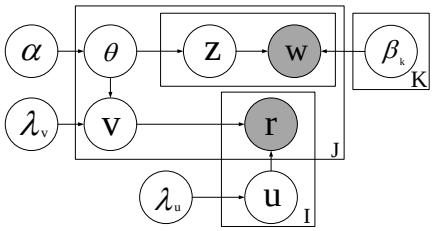


图 3–1 CTR 的概率图模型

络。CTR-SR 的图模型如图3–2。

CTR-SR 模型的生成过程如下:

1. 为每个标签 $\mathbf{t}_i$ 生成隐含变量:

$$u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K).$$

2. 对于每个物品 j,

   (a) 生成话题分布 $\theta_j \sim \text{Dirichlet}(\alpha)$.

   (b) 对于文章 $\mathbf{w}_j$ 中的每一个词 w_{jn},

      i. 生成话题 $z_{jn} \sim \text{Mult}(\theta_j)$.

      ii. 生成单词 $w_{jn} \sim \text{Mult}(\beta_{z_{jn}})$.

3. 按矩阵的高斯分布生成社交隐含矩阵 $S = [s_1, s_2, \cdots, s_J]$[49]:

$$S \sim \mathcal{N}_{K,J}(0, I_K \otimes (\lambda_l \mathscr{L}_a)^{-1}). \tag{3–2}$$

4. 为物品 j 通过两个高斯分布的乘积 (PoG) 来生成隐含向量:[50]:

$$v_j \sim \text{PoG}(\theta_j, s_j, \lambda_v^{-1} I_K, \lambda_r^{-1} I_K). \tag{3–3}$$

5. 对于每个标签 -物品对 (i,j)，生成标签信息:

$$r_{ij} \sim \mathcal{N}(u_i^T v_j, c_{ij}^{-1}).$$

在上述生成过程中，S 表示社交隐含矩阵，每列代表物品 j 的社交隐含向量 s_j。公式（3–2）中的 $\mathcal{N}_{K,J}(0, I_K \otimes (\lambda_l \mathscr{L}_a)^{-1})$ 表示矩阵变量正态分布 [49]:

$$\begin{aligned} p(S) &= \mathcal{N}_{K,J}(0, I_K \otimes (\lambda_l \mathscr{L}_a)^{-1}) \\ &= \frac{\exp\{\mathrm{tr}[-\frac{\lambda_l}{2} S \mathscr{L}_a S^T]\}}{(2\pi)^{JK/2} |I_K|^{J/2} |\lambda_l \mathscr{L}_a|^{-K/2}}, \end{aligned} \tag{3–4}$$

其中 $\otimes$ 表示两个矩阵的 Kronecker 积 [49]，$\mathrm{tr}(\cdot)$ 表示矩阵的迹，$\mathscr{L}_a = D - A$，其中 D 是对角矩阵，满足 $D_{ii} = \sum_j A_{ij}$。这里 A 是社交网络的邻接矩阵。如果 i 与 j 连边，则 $A_{ij} = 1$，否则 $A_{ij} = 0$。公式（3–3）中的 $\mathrm{PoG}(\theta_j, s_j, \lambda_v^{-1} I_K, \lambda_r^{-1} I_K)$ 表示高斯分布 $\mathcal{N}(\theta_j, \lambda_v^{-1} I_K)$ 与 $\mathcal{N}(s_j, \lambda_r^{-1} I_K)$ 的乘积。它同时也满足高斯分布 [50]，对应的高斯分布为 $\mathcal{N}(\mu_{vr}, \lambda_{vr}^{-1} I_K)$，其中

$$\begin{aligned} \mu_{vr} &= \frac{\theta_j \lambda_v + s_j \lambda_r}{\lambda_v + \lambda_r}, \\ \lambda_{vr} &= \frac{\lambda_v \lambda_r}{\lambda_v + \lambda_r}. \end{aligned}$$

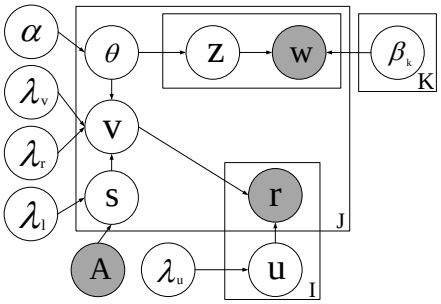


图 3–2 CTR-SR 的概率图模型

其中如公式（3–2）与图3–2显示的那样，CTR-SR 中，通过将邻接矩阵的 Laplacian 矩阵作为 S 的先验分布，社交网络信息被无缝的整合进了模型。其中的物理意义为是将隐含

向量（s_j 和 v_j）连接的尽量紧密。

因为计算 u_i、v_j、s_j 与 θ_j 完全后验不可行，我们提出了一种 EM 类型的算法来计算最大后验估计 (MAP)。我们可以通过计算给定参数情况下 $U = [u_1, u_2, \cdots, u_I]$, $V = [v_1, v_2, \cdots, v_J]$、$S$、$\theta_{1:J}$ 与 R 的最大完全 loglikelihood，来最大化后验概率：

$$\begin{aligned}\mathscr{L} = &-\frac{\lambda_l}{2}\text{tr}(S\mathscr{L}_a S^T) - \frac{\lambda_r}{2}\sum_j (s_j - v_j)^T(s_j - v_j) \\ &- \frac{\lambda_u}{2}\sum_i u_i^T u_i - \frac{\lambda_v}{2}\sum_j (v_j - \theta_j)^T(v_j - \theta_j) \\ &+ \sum_j \sum_n \log(\sum_k \theta_{jk}\beta_{k,w_{jn}}) - \sum_{i,j}\frac{c_{ij}}{2}(r_{ij} - u_i^T v_j)^2.\end{aligned} \tag{3–5}$$

其中省略了常数，且 CTR 中 topic model 的参数都设置成了为 1。注意第一项对应了去掉常数项后的 $\log p(S)$，并且：

$$\begin{aligned}\text{tr}(S\mathscr{L}_a S^T) &= \frac{1}{2}\sum_{j=1}^{J}\sum_{j'=1}^{J} A_{jj'}||S_{*j} - S_{*j'}||^2 \\ &= \frac{1}{2}\sum_{j=1}^{J}\sum_{j'=1}^{J}[A_{jj'}\sum_{k=1}^{K}(S_{kj} - S_{kj'})^2] \\ &= \frac{1}{2}\sum_{k=1}^{K}[\sum_{j=1}^{J}\sum_{j'=1}^{J} A_{jj'}(S_{kj} - S_{kj'})^2] \\ &= \sum_{k=1}^{K} S_{k*}^T \mathscr{L}_a S_{k*},\end{aligned} \tag{3–6}$$

我们可以看到最大化 $-\frac{\lambda_l}{2}\text{tr}(S^T\mathscr{L}_a S)$ 会使得当物品 j 与 j' 相连时 s_j 与 $s_{j'}$ 尽可能相近。

公式（3–5）中的函数 $\mathscr{L}$ 可以通过梯度上升的方法来优化。我们首先固定参数 β 然后迭代优化协同过滤的变量 $\{u_i, v_j, s_j\}$ 与话题比例 θ_j。参数 β 在每次 $\{u_i, v_j, s_j\}$ 与 θ_j 被优化时更新。

u_i 与 v_j 的更新规则如下：

$$\begin{aligned}u_i &\leftarrow (VC_iV^T + \lambda_u I_K)^{-1}VC_iR_i, \\ v_j &\leftarrow (UC_iU^T + \lambda_v I_K + \lambda_r I_K)^{-1}(UC_jR_j + \lambda_v\theta_j + \lambda_r s_j),\end{aligned}$$

其中 C_i 是一个对角矩阵，$\{c_{ij}, j = 1, \ldots, J\}$ 为其对角元素。R_j 是 R 的第 j 行。

对于社交隐含矩阵 S，我们固定除了第 k 行的所有行，并且更新第 k 行。通过对 L 关于 S_{k*} 求梯度并置为 0，我们得到了如下的线性系统:

$$(\lambda_l \mathscr{L}_a + \lambda_r I) S_{k*} = \lambda_r V_{k*}. \tag{3–7}$$

一个直接的解线性系统的方法是令 $S_{k*} = \lambda_r(\lambda_l \mathscr{L}_a + \lambda_r I_J)^{-1} V_{k*}$。但是每次更新的时间复杂度达到 $O(J^3)$，其中 J 是物品数。受 [51] 启发，我们使用最速下降法 [52] 来迭代更新 S_{k*}:

$$\begin{aligned} S_{k*}(t+1) &\leftarrow S_{k*}(t) + \delta(t) r(t) \\ r(t) &\leftarrow \lambda_r V_{k*} - (\lambda_l \mathscr{L}_a + \lambda_r I_J) S_{k*}(t) \\ \delta(t) &\leftarrow \frac{r(t)^T r(t)}{r(t)^T (\lambda_l \mathscr{L}_a + \lambda_r I_J) r(t)} \end{aligned}$$

使用最速下降法而不是直接线性系统可以大大减少每轮迭代的计算量。从 $O(J^3)$ 降为 $O(J)$。

对于 θ_j，我们首先定义 $q(z_{jn=k}) = \psi_{jnk}$，将包含 θ_j 的项分离出来后引用 Jensen 不等式得到，

$$\begin{aligned} \mathscr{L}(\theta_j) &\geq -\frac{\lambda_v}{2}(v_j - \theta_j)^T (v_j - \theta_j) \\ &\quad + \sum_n \sum_k \phi_{jnk} (\log \theta_{jk} \beta_{k, w_{jn}} - \log \phi_{jnk}) \\ &= \mathscr{L}(\theta_j, \boldsymbol{\phi}_j). \end{aligned} \tag{3–8}$$

这里 $\phi_j = (\phi_{jnk})_{n=1,k=1}^{N \times K}$。显然 $\mathscr{L}(\theta_j, \boldsymbol{\phi}_j)$ 是 $\mathscr{L}(\theta_j)$ 的紧下界。我们可以用投影梯度来优化 θ_j。ϕ_{jnk} 的最优值为

$$\phi_{jnk} \propto \theta_{jk} \beta_{k, w_{jn}}.$$

至于参数 β，我们使用 LDA 中相同的 M 步来更新，

$$\beta_{kw} \propto \sum_j \sum_n \phi_{jnk} 1[w_{jn} = w].$$

3.5 实验

我们在两个数据集上进行了实验，实验显示我们的方法相当有效。虽然我们关注的重点是推荐文章的标签，但我们的模型依旧可以较好的拓广到其它类型的数据上。

3.5.1 数据集

两个数据集都来自 CiteULike。对于第一个数据集来自 [12]，并且我们自行爬到了相应的标签数据。第二个数据集由我们自行收集。具体的，第一个数据集共 19107 个标签，第二个数据有 52946 个标签。我们将出现次数少于 5 次的标签剔除，最终分别得到了 7386 和 8311 个标签。两个数据集分别有 16980 和 25975 篇文章。两个数据集对应的 R 矩阵的稀疏度分别为 0.00145 和 0.00104。

我们使用 [12] 同样的方法对文本信息进行预处理，我们使用了标题和摘要信息作为内容。

因为 CiteULike 并不提供引用信息，我们通过用户 -文章信息来构建文章之间的网络。对于每个数据集，如果两篇文章有 4 个以上的共同读者，我们将其连边。这样做是因为拥有类似读者的两篇文章更有可能具有相似的话题。见图完成后，两个数据集分别有 259344 和 150567 条边。

3.5.2 评测方案

对于每个数据集，我们对于每个标签随机选择 P 篇文章作为训练数据，剩下的都作为训练集。我们从 1 到 10 变化 P，P 越小，训练数据越稀疏。注意当 $P = 1$ 时，只有 4.1% 的标签被放入了 *citeulike-a*的训练集，3.7% 的标签被放入了 *citeulike-t*的训练集。对于每个 P 我们都重复进行了五次实验，并且取平均值。

与 [12]、[29] 相似，我们使用 recall 来作为评判标准。与大多数推荐系统类似，我们

将备选标签按评分排序，并且推荐前 M 个标签。对于每个物品，我们定义 recall@M 为:

$$\text{recall@M} = \frac{\text{前 M 个推荐标签中对应目标物品的标签数}}{\text{目标物品总标签数}}.$$

最终的结果是多次实验后的平均值。

除此之外，如 [30] 提到的，我们使用 success@M 为另一个评测标准。success@M 定义为在推荐的前 M 个标签中出现至少一个正确标签的概率。

3.5.3 基线与实验设置

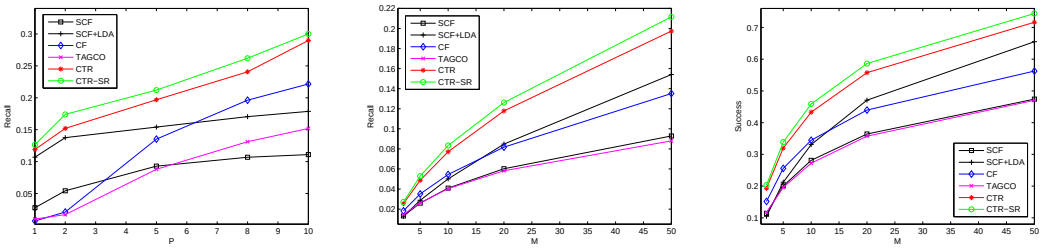
我们使用下列方法与 CTR-SR 模型进行比较:

- TAGCO：基于 co-occurrence 的方法 [30]。
- SCF: 基于相似度的协同过滤方法 [29]。他找到文章最相似的 k 篇文章，并依据这 k 篇文章的 tag 进行推荐。
- CF：基于矩阵分解的协同过滤方法 [48]。它将训练矩阵分解为两个低秩矩阵 U，V，并且用 UV^T 来近似目标矩阵。
- SCF+LDA: 这个方法集成了 SCF 和 LDA 方法。它属于 [35] 提出的混合方法。
- CTR：之前提到过的模型。

我们使用 validation 集来找到最优的参数。具体的，我们发现 $\lambda_v = 10$、$\lambda_u = 0.1$、$a = 1$、$b = 0.01$、$K = 200$ 时 CTR 取得较好的效果。对于 CF 方法，$\lambda_v = 1$、$\lambda_u = 1$、$a = 1$、$b = 0.01$、$K = 200$ 时最好。而对于 CTR-SR 模型，参数为 $\lambda_v = 10$、$\lambda_u = 0.1$、$a = 1$、$b = 0.01$、$K = 200$、$\lambda_r = 100$、$\lambda_l = 10$。

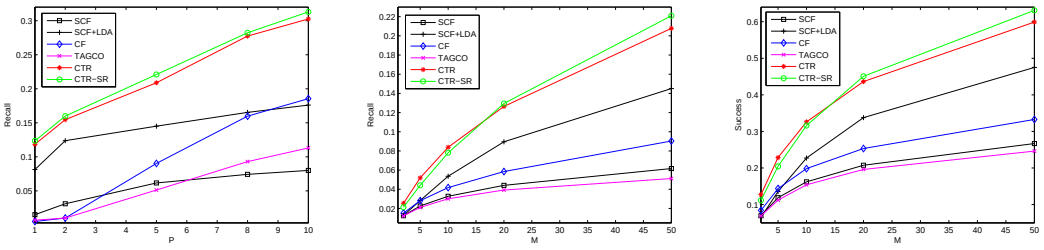
3.5.4 预测性能

图3–3（a）与图3–4（a）分别是数据集 *citeulike-a*与 *citeulike-t*上当 P 设为 1、2、5、8、10 时的 recall@50。两个数据集的随机基线分别为 0.68% 与 0.60%。由图可知，混合方法 SCF+LDA 的预测性能超过了所有非混合方法，CTR 的性能超过了 SCF+LDA。最后我



(a) (b) (c)

图 3–3 *citeulike-a*上的实验结果。(a) 是所有方法的 recall@50。(b) 是所有方法在 P 固定为 5 时的 recall@M，M 取值 2 到 50。(c) 所有方法的 success@M，P 固定为 5，M 取值 2 到 50。



(a) (b) (c)

图 3–4 *citeulike-t*上的实验结果。(a) 是所有方法的 recall@50。(b) 是所有方法在 P 固定为 5 时的 recall@M，M 取值 2 到 50。(c) 所有方法的 success@M，P 固定为 5，M 取值 2 到 50。

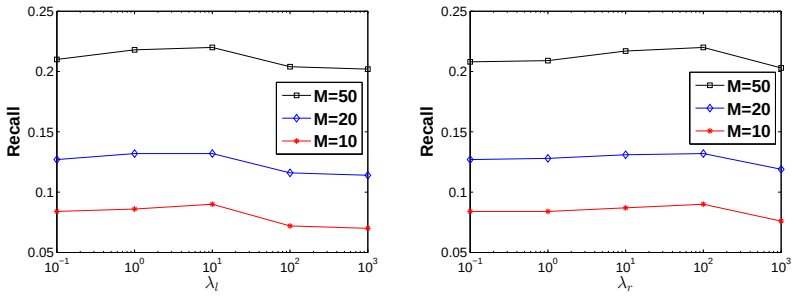
们的 CTR-SR 模型通过将物品间的社交网络整合入模型中，在绝大多数情况下达到了最好的预测性能。

图3–3（b）与图3–4（b）是 P 固定为 5 时所有方法的 recall，M 在两个数据集中都取值为 2、5、10、20、50。图3–3（c）与图3–4（c）是当 P 固定为 5 时所有方法的 success@M，M 在两个数据集中都取值为 2、5、10、20、50。同样地，我们可以看见 CTR 的预测性能超过了其他基线，而 CTR-SR 明显地在绝大多数情况下达到了比其他所有基线高的预测性能。P 取其它值时相似的现象也可以观察到，由于篇幅所限，本文不予讨论。

3.5.5 参数敏感度

图3–5（a）表示的是参数 λ_l 如何影响CTR-SR的预测性能。P 固定为 5，$\lambda_v = 10$，$\lambda_u = 0.1$，$\lambda_r = 100$。由图可知，预测性能先随着 λ_l 上升，而在 $\lambda_l = 10$ 之后开始轻微下降。对于所有的 M 值都是如此。这里可以看出，模型当 λ_l 在很大的一个范围内都不敏感。

图3–5（b）展示的是模型 CTR-SR 对于参数 λ_r 的敏感度。在实验中，P 同样被固定



(a) (b)

图 3–5 参数敏感度。(a) 是参数 λ_l 对 CTR-SR 的影响。(b) 是参数 λ_r 对 CTR-SR 的影响。

为 5，$\lambda_v = 10$，$\lambda_u = 0.1$，$\lambda_l = 10$。如图所示，模型的预测性能先随着 λ_r 增大而提高，而在 $\lambda_r = 100$ 之后开始下降。对于所有的 M 值都是如此。同样，模型当 λ_r 在很大的一个范围内都不敏感。

表 3–1 对示例文章的标签推荐

Article I	Title: How much can behavioral targeting help online advertising?			
	Top topic 1: web, search, engine, pages, keyword, click, hypertext, html, searchers, crawler			
	Top topic 2: mobile, phones, attitudes, advertising, consumer, marketing, commerce, sms, m-learning			
	True tags: behavioral_targeting, advertising, ads, computational_advertising, recommend, user-behavior, user_profile			
Top 10 recommended tags	CTR	True tag?	CTR-SR	True tag?
	1. random-walks	no	1. behavioral_targeting	yes
	2. page-rank	no	2. ads	yes
	3. computational_advertising	yes	3. computational_advertising	yes
	4. citizen-science	no	4. random-walks	no
	5. natural_history	no	5. page-rank	no
	6. search_engine	no	6. developing	no
	7. engine	no	7. recommend	yes
	8. searchengine	no	8. advertising	yes
	9. what	no	9. what	no
	10. re-ranking	no	10. need	no
Article II	Title: Lowcost multitouch sensing through frustrated total internal reflection			
	Top topic 1: molecular, molecules, surface, chemical, formation, forces, reaction, shapes, sensing, kinetics			
	Top topic 2: design, interface, principles, interfaces, interactive, devices, usability, application			
	True tags: tech, screen, gestures, touch, interface, multitouch, multi-touch, table, visualization, computer_vision			
Top 10 recommended tags	CTR	True tag?	CTR-SR	True tag?
	1. guide	no	1. touch	yes
	2. gamma	no	2. field	no
	3. optical	no	3. gestures	yes
	4. nanoparticles	no	4. table	yes
	5. nano	no	5. multi-touch	yes
	6. dna-nanotecnology	no	6. screen	yes
	7. tirf	no	7. multitouch	yes
	8. sms	no	8. dna-nanotecnology	no
	9. touch	yes	9. nano	no
	10. field	no	10. superlist	no

3.5.6 可解释性

除了良好的表现，我们的模型同时提供了很好的解释性。两个样例文章如表3–1所示。注意到虽然 CTR 和 CTR-SR 学习出来的话题分布比重不同，但排名基本类似。CTR-SR 和

CTR 的样例都来自 $P = 1$ 的稀疏情况。也就是说训练集中每个标签只对应了一篇文章。从表中可以看出 u，在第一篇文章中，CTR-SR 和 CTR 模型的准确度分别为 50% 和 10%；第二篇文章中准确度分别为 60% 和 10%。我们可以发现社交网络信息非常具有信息量，而 CTR-SR 模型很好地利用了这一点。更仔细地观察发现，第一篇文章“How much can behavioral targeting help online advertising?”主要是关于在线广告的，而 CTR 的模型更加关注在文章的技术细节上，而 CTR-SR 得到的标签更关注文章的本质。同样的，对于第二篇文章“Lowcost multitouch sensing through frustrated total internal reflection”关注的是多点触控。CTR 推荐的大多为 nanoparticles 之类的专业术语，相反 CTR-SR 则准确的推荐出了 multi-touch 和 screen 等更准确的标签。

3.6 本章小结

本章中，我们首先提出将 CTR 模型应用到标签推荐系统任务中来，进一步，我们扩展了 CTR 模型，提出了一种新的层级式的贝叶斯模型，称为社交正则化协同话题回归方法（CTR-SR），用以无缝整合物品 -标签矩阵，内容信息以及物品之间的社交网络关系。并且我们通过实验显示了 CTR-SR 模型的有效性。

本章的工作已经发表在国际顶级会议 IJCAI（International Joint Conference on Artificial Intelligence）2013 上。论文题目为“Collaborative Topic Regression with Social Regularization for Tag Recommendation”。本文作者为第一作者。

第四章　在线自中心模型

4.1　引言

网络分析 [51, 53–62]，特别是动态网络分析 （Dynamic Network Analysis，即 DNA）在包括社会科学与生物学在内的许多领域中已经显得越来越重要。虽然现在已经有不少关于动态网络分析的工作，但是其中绝大多数要不就是只关注极粗的细粒度下的大规模数据 [63–70]，要不就是只关注在一个很小的网络中的微细粒度的分析 [71, 72]。近年来，有人提出了动态自中心模型（Dynamic Egocentric Model，即 DEM）[73]，这个模型基于多变量计数过程并成功地对微细粒度的大规模的时变引用网络进行建模。

虽然 DEM 能够动态地更新节点（在原文中表示文章）的链接特征，DEM[1]学习出来的参数与话题特征在预测过程中却是固定的。因此 DEM 随着时间时间的推移，预测的准确度会严重地下降，因为实际上话题特征与参数都应该是随着时间变化的。比如，模型的链接特征之一是截至某个时间点节点的入度（文章被引用的次数），随着时间的推移，一篇文章的被引用次数会变得越来越多，因此整个数据集中引用数的分布也会随着时间而改变。这样的结果就是，对应这个特征的参数，甚至是其他参数，也应该跟着改变。另外，关于话题特征，虽然乍一看，一篇文章的话题特征会随着时间改变可能显得有点不可思议，因为按常理来讲，一篇发表的文章的文字都是不会随着时间改变的。然而，引用这篇文章的许多文章却时时在变化。我们认为（后面的实验也证明了）将引用信息与文本内容信息结合起来决定一篇文章的话题特征要更加合理。比如，一篇关于神经网络的文章在 20 世纪 50 年代可能会被认为是与心理学或者生物学高度相关的，但是在今天，它却更可能被划分为关于机器学习的文章，因为几十年来有越来越多发表的文章引用了它。由此可知，一篇文章的话题特征显然是会随着时间的改变的，只是幅度的大小不同而已。由于无法对时变的参数与话题参数建模，DEM 并没法很好地对动态网络进行精确的建模从而使得准确度会随着时间而下降。这个现象也可以在论文 [73] 的图 2 看到。

本章中，我们提出了 DEM 的一个在线拓展，名为在线自中心模型（Online Egocentric

[1]在 [73] 中，有两个 DEM 的变种。一个只对链接特征进行建模，另一个同时对链接特征与话题特征（文本信息）进行建模。由于后者的准确度远高于前者并且一篇文章的文本信息是较容易得到的，除非特殊说明，下文中的 DEM 指的是后者。

Model，即 OEM)。这个模型可以对时变的话题特征与模型参数进行建模。这个模型的贡献如下:

- OEM 同时考虑话题特征与模型参数的时变特点，从而使得随着时间推移模型预测的准确度不会下降。
- 在 OEM 的在线训练过程中，我们还揭示了文章话题特征的变化以及文章间话题特征的传递。
- 大量的在两个真实数据集上的实验说明了 OEM 模型的有效性。

4.2 动态自中心模型（DEM）

在这节中，我们会简要的说明一下 DEM [73]。DEM 将是我们本章工作的基础。为了便于理解这里使用了与 [73] 相同的符号。

n 是网络中节点（文章）的总数。DEM 试图通过在每个节点 i （$i = 1, 2, \cdots, n$）上放置一个计数过程 $N_i(t)$ 以对动态网络进行建模。其中 $N_i(t)$ 表示节点 i 上“事件”的截止时间 t 的累计发生次数。这里“事件”的定义要取决于上下文。比如，在引用网络中，一个“事件”可以对应着一次引用。

虽然可以最大化这些计数过程的全概率，推出一个连续时间的模型，但是对于引用网络来说，显然通过最大化偏概率的方法来估计那些与时变统计量相关的参数会更加实际。所以 DEM 试图最大化下面整个网络的 likelihood:

$$L(\boldsymbol{\beta}) = \prod_{e=1}^{m} \frac{\exp(\boldsymbol{\beta}^T \mathbf{s}_{i_e}(t_e))}{\sum_{i=1}^{n} Y_i(t_e) \exp(\boldsymbol{\beta}^T \mathbf{s}_i(t_e))}, \tag{4–1}$$

其中 m 是引用事件的总次数，e 是每次引用事件的索引，i_e 表示在事件 e 中被引用的文章，t_e 表示事件 e 发生的时间，$Y_i(t)$ 的值当节点 i 在时间 t 存在是为 1，否则为 0。$\mathbf{s}_i(t_e)$ 表示节点 i 在时间 t_e 的特征向量。$\boldsymbol{\beta}$ 是需要学习的参数向量。

$\mathbf{s}_i(t_e)$ 中的向量可以分为两类。一类称为“链接特征（统计量)”，另一类称为“话题特征”。在 [73] 中有 8 个链接特征，包括三个 preferential attachment 统计量、三个 triangle 统计量与两个 out-path 统计量。另外还通过对文章的摘要运行 LDA 对每篇文章提取了 50

个话题特征。更具体地，假设在时间 t_e 新到的文章为 i，我们可以如下计算任何已有文章 j 的话题特征：

$$\mathbf{s}_j^{LDA}(t_e) = \boldsymbol{\theta}_i \circ \boldsymbol{\theta}_j,$$

其中 $\boldsymbol{\theta}_i$ 表示文章 i 的话题比例，$\circ$ 意为向量间的元素逐项相乘。

由上可知，$\mathbf{s}_i(t_e)$ 是一个含有 58 个特征的向量，其中前 8 个特征为链接特征，后面 50 个为话题特征。对应地，$\boldsymbol{\beta}$ 为一个长度为 58 的参数向量。关于特征的更多详情可参考论文 [73]。

4.3 在线自中心模型（OEM）

在动态网络的预测过程中，DEM 的链接特征会自动更新。然而模型的参数 $\boldsymbol{\beta}$ 与话题特征 $\boldsymbol{\theta}_i$ 都不会。这使得 DEM 的准确度会随着时间推移而下降。在本节中，我们提出了在线自中心模型（OEM）以解决这个问题。OEM 的基本思想就是在新事件发生后交替地更新参数与话题特征。

虽然我们可以从整个文章的集合中完整地学习 LDA，但是显然如果我们直接使用在线的 LDA 模型 [74] 的话会十分的耗费时间。因此，在这个工作中，我们先固定话题后再学习话题比例 $\boldsymbol{\theta}$。因为在引用网络中，即使一些文章本身的话题比例会随着时间而改变，主要的话题是相对稳定不变的，所以这么做是合理的。我们只需要在每隔一段比较长的时间更新全部的话题。从实验可以看出，这样做依然可以达到很好的准确度。

因此，在某些新事件发生后，OEM 试图最小化下面的目标函数：

$$\begin{aligned} &minimize \quad -\log L(\boldsymbol{\beta}, \boldsymbol{\omega}) + \lambda \sum_{k=1}^{n} \|\boldsymbol{\omega}_k - \boldsymbol{\theta}_k\|_2^2 \\ &subject\ to: \ \boldsymbol{\omega}_k \succeq \mathbf{0},\ \mathbf{1}^T \boldsymbol{\omega}_k = 1, \end{aligned} \tag{4–2}$$

其中 $\boldsymbol{\omega}_k$ 是待学习的节点 k 的新话题比例，$\boldsymbol{\theta}_k$ 是节点 k 当前的话题比例，$\boldsymbol{\omega} = \{\boldsymbol{\omega}_k\}_{k=1}^n$，$L(\boldsymbol{\beta}, \boldsymbol{\omega})$ 的定义与式子（4–1）中的 $L(\boldsymbol{\beta})$ 相同，除了这里将 $\boldsymbol{\beta}$ 与话题比例都作为变量[2]，$\boldsymbol{\omega}_k \succeq \mathbf{0}$ 表示 $\boldsymbol{\omega}_k$ 中的每一个元素都是非负的，$\mathbf{1}$ 是一个元素全为 1 的向量，这些限制用于保证 $\boldsymbol{\omega}_k$ 中的所有元素都是非负的而且元素和为 1。λ 是一个控制两个项之间权重的超参

[2]注意 $L(\boldsymbol{\beta}, \boldsymbol{\omega})$ 与 $L(\boldsymbol{\beta})$ 是不同的，在 $L(\boldsymbol{\beta})$ 中，只有 $\boldsymbol{\beta}$ 是变量而 $\boldsymbol{\omega}$ 是常数

数。

当一个新事件或者一系列新事件被观察到，式子（4–2）中的第二项会保证更新后的话题比例 $\boldsymbol{\omega}_k$ 不会距离目前的话题比例 $\boldsymbol{\theta}_k$ 太远。除此之外，我们使用旧的 $\boldsymbol{\beta}$ 作为初始值来更新 $\boldsymbol{\beta}$。综上，通过有效利用现有事件的信息，我们可以提出一个在线学习算法。

显然可见，式子（4–2）的优化问题并不是对 $(\boldsymbol{\beta}, \boldsymbol{\omega})$ 联合凸的。但是好在我们可以证明这个目标函数是在一个变量固定时，关于另外一个变量是凸的。在本章中，我们设计了一个交替投影算法（alternating projection）以找出最优解。更具体地，每次迭代中，我们固定两个变量中的一个并更新另一个。算法的概要如下：

- 在线 $\boldsymbol{\beta}$ 步（online $\boldsymbol{\beta}$ step）：固定 $\boldsymbol{\omega}$ 后使用牛顿法更新参数 $\boldsymbol{\beta}$，初始化用的是当前的 $\boldsymbol{\beta}$；
- 在线话题步（online topic step）：固定 $\boldsymbol{\beta}$ 后在当前话题比例 $\boldsymbol{\theta}_k$ 的基础上，最小化式子（4–2）以获得更新后的话题比例 $\boldsymbol{\omega}_k$。

上述过程需要重复几次知道符合终止条件。

Mini-batches: 在上面的 OEM 中，每次一篇新文章 i 出现，我们可以将它加入原引用网络中后马上使用在线 $\boldsymbol{\beta}$ 步与在线话题步直至收敛。但是，这对于大规模的引用网络来说是十分耗时间的。因此我们可以等新文章积累到一定数量后才开始更新。这种 mini-batch 技巧不仅可以节省计算时间，而且可以减少噪声的影响 [74]。因此在我们的模型实现中，我们在每 q 次引用事件后更新一次而非每次事件后更新一次。q 在实验中设置为 1500 左右。

本节接下来的内容讲详细地说明如何进行在线的 $\boldsymbol{\beta}$ 的学习与话题比例的学习。

4.3.1 在线 $\boldsymbol{\beta}$ 步

固定 $\boldsymbol{\omega}$ 后，学习 $\boldsymbol{\beta}$ 的目标函数如下：

$$L(\boldsymbol{\beta}) = \prod_{e=x}^{x+q-1} \frac{\exp(\boldsymbol{\beta}^T \mathbf{s}_{i_e}(t_e))}{\sum_{i=1}^{n} Y_i(t_e) \exp(\boldsymbol{\beta}^T \mathbf{s}_i(t_e))},$$

其中 x 是 mini-batch 中的第一个事件，q 是 mini-batch 中的事件数。

为了避免在更新 $\boldsymbol{\beta}$ 时遍历所有之前的引用事件，我们用了一个训练窗口，使得在训练参数 $\boldsymbol{\beta}$ 时只需要考虑引用事件中的一个比较小的子集。若训练窗口的宽度为 W_t $(1 \leq W_t \leq q)$，可以通过优化下面式子来学习 $\boldsymbol{\beta}$:

$$L_w(\boldsymbol{\beta}) = \prod_{e=x+q-W_t}^{x+q-1} \frac{\exp(\boldsymbol{\beta}^T \mathbf{s}_{i_e}(t_e))}{\sum_{i=1}^{n} Y_i(t_e) \exp(\boldsymbol{\beta}^T \mathbf{s}_i(t_e))}.$$

而且我们还可以缓存每个节点的链接特征以进一步减小计算负担，正如 [73] 所做的。

4.3.2 在线话题步

在本小节中，我们先提出“满话题步（full topic step）”，然后再推导出“近似在线话题步（approximative online topic step）”以加速优化过程。

4.3.2.1 满在线话题步（Full Online Topic Step）

如果一次性地更新 $\boldsymbol{\omega}$ 中的所有话题比例将会极其耗费时间。因此我们设计了一个交替的算法来更新 $\boldsymbol{\omega}$。更具体地，每一次只更新一篇文章的话题比例 $\boldsymbol{\omega}_k$，在更新 $\boldsymbol{\omega}_k$ 时，其他文章的话题比例 $\{\boldsymbol{\omega}_i | i \neq k\}$ 保持不变。如果在一个大小为 q 的 mini-batch 中，节点 k 在引用事件 $e_1, e_2, \ldots, e_p$ 中被引用而在时间 $e_{p+1}, e_{p+2}, \ldots, e_q$ 没有被引用（注意 e_2 发生的时间不一定在 e_{p+2} 之前，虽然前者的下标较后者小），我们需要优化的目标函数 $f(\boldsymbol{\omega}_k)$ 是:

$$\begin{aligned} &- \log\Big(\prod_{i=1}^{p} \frac{\alpha_i \exp(\mathbf{a}_i^T \boldsymbol{\omega}_k)}{A_i + \alpha_i \exp(\mathbf{a}_i^T \boldsymbol{\omega}_k)} \prod_{u=p+1}^{q} \frac{C_u}{B_u + \gamma_u \exp(\mathbf{b}_u^T \boldsymbol{\omega}_k)}\Big) \\ &+ \lambda \|\boldsymbol{\omega}_k - \boldsymbol{\theta}_k\|_2^2, \end{aligned} \tag{4–3}$$

其中

$$\begin{aligned}
\alpha_i &= \exp(\boldsymbol{\beta}_l^T \mathbf{s}_k^l(t_{e_i})),\\
\gamma_u &= \exp(\boldsymbol{\beta}_l^T \mathbf{s}_k^l(t_{e_u})),\\
A_i &= \sum_{j\neq k} Y_j(t_{e_i}) \exp(\boldsymbol{\beta}^T \mathbf{s}_j(t_{e_i})),\\
B_u &= \sum_{j\neq k} Y_j(t_{e_u}) \exp(\boldsymbol{\beta}^T \mathbf{s}_j(t_{e_u})),\\
\mathbf{a}_i &= \boldsymbol{\beta}_t \circ \boldsymbol{\theta}_i,\\
\mathbf{b}_u &= \boldsymbol{\beta}_t \circ \boldsymbol{\theta}_u.
\end{aligned}$$

这里，$\boldsymbol{\beta}_l$ 包含着参数 $\boldsymbol{\beta}$ 的前 8 个元素（对应着链接特征），$\boldsymbol{\beta}_t$ 包含着参数 $\boldsymbol{\beta}$ 的后 50 个元素（对应的是话题特征），$\boldsymbol{\theta}_i$ 是引用事件 e_i 的引用者的话题比例，$\mathbf{s}_k^l(t_{e_i})$ 是引用事件 e_i 中的节点 k 的链接特征（前 8 个特征），C_u 是一个与 $\boldsymbol{\omega}_k$ 无关的常数。

式子（4–3）的一阶与二阶偏导如下:

$$\begin{aligned}
\frac{\partial f}{\partial \boldsymbol{\omega}_k} =& -\sum_{i=1}^{p} \mathbf{a}_i + \sum_{i=1}^{p} \frac{\mathbf{a}_i \alpha_i \exp(\mathbf{a}_i^T \boldsymbol{\omega}_k)}{A_i + \alpha_i \exp(\mathbf{a}_i^T \boldsymbol{\omega}_k)}\\
&+ \sum_{u=p+1}^{q} \frac{\mathbf{b}_u \gamma_u \exp(\mathbf{b}_u^T \boldsymbol{\omega}_k)}{B_u + \gamma_u \exp(\mathbf{b}_u^T \boldsymbol{\omega}_k)}\\
&+ 2\lambda(\boldsymbol{\omega}_k - \boldsymbol{\theta}_k),
\end{aligned} \tag{4–4}$$

$$\begin{aligned}
\frac{\partial^2 f}{\partial \boldsymbol{\omega}_k^2} =& \sum_{i=1}^{p} \frac{A_i \alpha_i \mathbf{a}_i \mathbf{a}_i^T \exp(\mathbf{a}_i^T \boldsymbol{\omega}_k)}{(A_i + \alpha_i \exp(\mathbf{a}_i^T \boldsymbol{\omega}_k))^2}\\
&+ \sum_{u=p+1}^{q} \frac{B_u \gamma_u \mathbf{b}_u \mathbf{b}_u^T \exp(\mathbf{b}_u^T \boldsymbol{\omega}_k)}{(B_u + \gamma_u \exp(\mathbf{b}_u^T \boldsymbol{\omega}_k))^2} + 2\lambda \mathbf{I},
\end{aligned}$$

其中 $\mathbf{I}$ 是单位矩阵。

从上面式子可以看出 Hessian 矩阵正定（PD）的，因此（4–3）的函数是凸的。我们可以直接使用 solver 来找到全局最优解。

4.3.2.2 近似在线话题步（Approximative Online Topic Step）

在（4–4）中，A_i 远大于 $\mathbf{a}_i\alpha_i \exp(\mathbf{a}_i^T\boldsymbol{\omega}_k)$ 与 $\alpha_i \exp(\mathbf{a}_i^T\boldsymbol{\omega}_k)$，且 p 在每个 batch 中都相对较小。同理，B_u 远大于 $\mathbf{b}_u\gamma_u \exp(\mathbf{b}_u^T\boldsymbol{\omega}_k)$ 与 $\gamma_u \exp(\mathbf{b}_u^T\boldsymbol{\omega}_k)$，而 $(q-p)$ 也相对较小。因此，(4–4) 中的第二与第三项要远小于其它两项。这意味着我们可以删去较小的两项以得到一个近似的梯度:

$$\frac{\partial f}{\partial \boldsymbol{\omega}_k} \approx -\sum_{i=1}^{p} \mathbf{a}_i + 2\lambda(\boldsymbol{\omega}_k - \boldsymbol{\theta}_k).$$

基于上面的近似梯度，我们可以恢复（4–2）的近似目标函数:

$$\begin{aligned} &minimize \quad -\sum_{i=1}^{p} \mathbf{a}_i^T\boldsymbol{\omega}_k + \lambda\sum_{k=1}^{n} \|\boldsymbol{\omega}_k - \boldsymbol{\theta}_k\|_2^2 \\ &subject\ to: \ \ \boldsymbol{\omega}_k \succeq \mathbf{0}, \mathbf{1}^T\boldsymbol{\omega}_k = 1. \end{aligned} \tag{4–5}$$

我们将（4–5）这个 OEM 的变种称为“近似 OEM”（approximative OEM），而将原来的 OEM 称为“满 OEM”（full OEM）。在实验中可以发现近似 OEM 可以达到与满 OEM 接近的准确度而需要少很多的时间。

4.3.3 收敛分析

在每次迭代中，学习的算法保证目标函数的值总是下降。而且目标函数值总是大于等于 0，因此算法是收敛的。

4.4 实验

我们将 DEM 与 OEM 应用于两个引用网络中并比较两个模型的实验结果。我们还分析了文章话题比例的演变。

4.4.1 数据集

由于本章主要做的是引用网络分析（动态网络分析中最重要的应用之一），这里用的是两个引用网络的数据集arXiv-TH与arXiv-PH。两个数据集都是从arXiv[3]爬取的。数据集

[3]http://snap.stanford.edu/data

表 4–1 数据集信息

Data Set	#Papers	#Citations	#Unique Times
arXiv-TH	14226	100025	10500
arXiv-PH	16526	125311	1591

的主要信息见表4–1。

arXiv-TH 数据集是关于高能物理理论的一系列文章。时间的范围是 1993 年到 1997 年，这个数据集有很高的时间解析度（精确到毫秒）。arXiv-PH数据集是关于高能物理现象的一系列文章，时间范围为 1993 年到 1997 年，时间精确到每天。由于数据集中的时间解析度非常高，我们可以假设每篇新文章都在不同的时间加入到网络中而且显然同一个时间中可能有不止一个引用事件。正如前一节提到的，我们一个 batch 一个 batch 地更新话题比例与参数。更具体地，我们将数据集划分成一个个的 mini-batch，每个 mini-batch 中包含着在一段时间中发生的引用时间。对于arXiv-TH每个 mini-batch 中的时间戳数为 100，而对于 arXiv-PH 是 20。对应与每一个 mini-batch 的事件数大约为 1500。

4.4.2 基线

在实验中我们比较了下面 4 个模型的性能：

- **DEM**：原来的有 8 个链接特征与 50 个话题特征的 DEM。注意原来的 DEM 并不是在线（online），参数与话题特征在训练后是固定的。
- **OEM-β**：只带有在线 β 步的 OEM，这个模型中，β 会随随时间更新但是话题特征不会。
- **OEM-full**：带有在线 β 步与话题步的满 OEM，话题特征与参数都会随着时间改变，使用了目标函数（4–2）。
- **OEM-appr**：带有在线 β 步与近似话题步的 OEM，话题特征与参数都会随着时间改变，使用了目标函数（4–5）

4.4.3 评测标准

与 [73] 类似，我们用下面三个标准来评测上面的模型：

表 4–2 数据集建立、训练、测试阶段的分割

DATA SETS	BUILDING	TRAINING	TESTING
ARXIV-TH	62239	1465	36328
ARXIV-PH	82343	1739	41229

- 平均测试 log-likelihood（Average held-out log-likelihood）：在每个测试引用事件中对（4–1）中的 likelihood$L(\boldsymbol{\beta})$ 取 log 后即可得到测试 log-likelihood。将所有测试事件的测试 log-likelihood 的和除以本 batch 中事件的总数，即可以得到平均测试 log-likelihood。这个数值越高，则说明测试准确度越高。

- 召回率 @K（Recall of top-K recommendation list）：这里的召回率定义为 K 个最可能的引用事件中真实发生的比例。这里的 K 是一个切分点（cut-point）。

- 平均测试正规排名（Average held-out normalized rank）：这里每个引用事件的排名（rank）指的是这个引用在已排序好的推荐列表中的实际位置。这个排名除以可能引用事件的总数即得到正规化（normalize）后的排名。这个数值越低，表示预测性能越好。

4.4.4 结果与分析

如 DEM[73]，我们将每个数据集分为三个部分：建立阶段、训练阶段与测试阶段。建立阶段主要是为了建立起引用网络的统计量，一般它的时间范围会较长以缓解截断效应（1993 年前的引用时间没有出现在数据集中）并避免 bias。在训练阶段中，我们训练出初始的模型参数与话题特征。为了更加全面的展示并比较模型的预测性能，这里的测试阶段比较长。测试阶段被划分为 24 个 batch。注意统计量（链接特征）在训练阶段与测试阶段中都是会动态改变的。每个阶段的数据大小（用引用事件数表示）如表4–2所示。

为了进一步减少 OEM 训练与测试的时间，我们在每个 batch 中只随机选取了一部分的时间中的引用事件来优化文章的话题比例。比如当优化文章 i 的话题比例时，在第 1 个 batch 到达后，我们随机选取 10%（这里我们将 10% 称为 citer 百分比，下文亦然）的引用者（citer）而不是全部引用者。这可以一定程度加速计算。在 OEM 中，我们设超参数 $\lambda = 0.1$，设 citer 百分比为 10%，除非另外说明。超参数 citer 百分比与 λ 对模型的影响会在接下来的实验中具体说明。

OEM 的测试过程的细节如下。我们先用建立阶段与训练阶段的数据训练一个初始的 OEM。显然此时这个初始的 OEM 等价于 DEM。然后我们评测这个模型在 Batch 1 的预测性能（注意到我们在训练时并没有用到 Batch 1 的数据)。之后我们再将 Batch 1 的数据吸收为额外的训练数据并更新 OEM 的参数与特征。然后我们再接着使用现在这个已更新的 OEM 来预测 Batch 2。由此可见，在测试某一个 batch 之前，我们并没有将这个 batch 的数据用于训练。因此测试的结果会真实地反映 OEM 的泛化/预测能力。

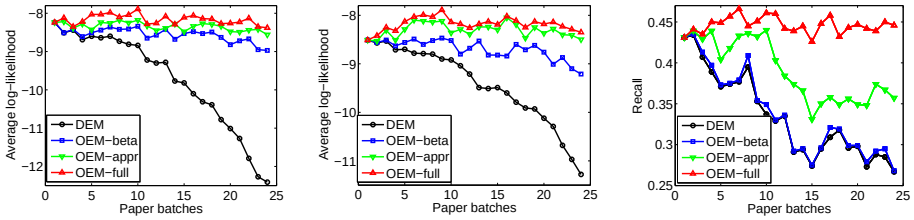
图 4–1（a）和（b）是所有模型的平均测试 log-likelihood。由于初始的 OEM 与 DEM 是等价的，我们可以看到所有的模型在测试 Batch 1 时的性能都是相同的。然而，随着时间的推移，DEM 的预测性能会严重地下降，而 OEM 的各个变种则不会。比如，从图4–1（a）可以看出，DEM 的 log-likelihood 随着时间下降十分明显，而 OEM-β 只是从 -8.24 下降到-8.97。我们的 OEM-full 的预测能力超过了前面两个模型，log-likelihood 的范围是 -7.89 到 -8.38。OEM-appr 则从 -8.24 下降到 -8.56。

图4–1（c）和（d）是前 K 推荐列表中的召回率，K 取值 250。可以发现 DEM、OEM-β 与 OEM-appr 的性能都随着时间而下降，然而 OEM-full 却不会。虽然OEM-appr的预测性能也会随着时间下降，但是它的性能依然明显超过 DEM。OEM-β 的性能与 DEM 差不多，都不理想。这意味着话题特征的信息量是十分大的，只是更新 β 是远不够的。注意 K 取其他值时也可以得到相似的结果，由于篇幅所限本文不予讨论。

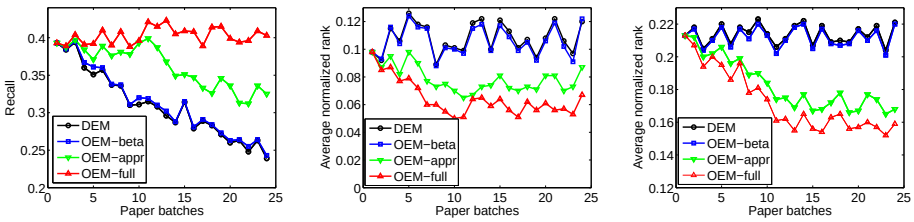
图4–1（e）与（f）是平均测试正规排名。可以发现 DEM 与 OEM-β 的性能无法随着时间而提高。而 OEM-full 与 OEM-appr 则可以。注意排名数值越低意味着预测能力越高。与前面相似，OEM-β 的不理想效果进一步说明了话题特征的更新对这一项评测标准的重要性。因为越到后面的 batch，候选的引用事件数会越多，如果用绝对的排名，DEM 的性能实际上是随着时间而下降的。但是OEM-full却可以防止性能的下降，即使是从绝对排名的角度来看。这个与图4–1（a)、(b)、(c）与（d）的结果相符。

表4–3比较了 OEM 与近似 OEM 的计算消耗。由表可知，虽然近似 OEM 比满 OEM 预测性能稍差，但是却节省了 50% 的时间。

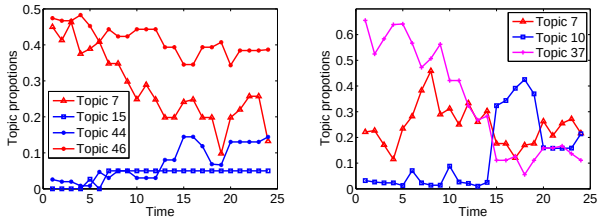
为了研究超参数（citer 百分比与 λ）对预测性能的影响，我们使用 arXiv-TH 数据集并计算了 citer 百分比与 λ 取不同值时所有测试 batch 的平均测试 log-likelihood。结果详见表4–4 与表4–5。由表4–4可知，0.1 为 λ 的最优值。从表4–5可以看出在 citer 百分比大于



(a) arXiv-TH (b) arXiv-PH (c) arXiv-TH(K=250)



(d) arXiv-PH(K=250) (e) arXiv-TH (f) arXiv-PH



(g) 第 8001 个时间点被引用的文章集合 (h) 第 8005 个时间点被引用的文章集合

图 4–1 （a）与（b）是测试引用事件的平均测试 log-likelihood。（c）与（d）前 K 推荐列表中的召回率。（e）与（f）为平均测试正规排名。由于所有的模型在建立阶段与训练阶段后的初始参数相同，它们在第 1 个测试 batch 的性能是相同的。这个从（a）到（f）可以看到。（g）与（h）是在第 8001 与第 8005 个时间点是被引用的两个文章集的话题演变。为了防止图像的混乱，我们只画出了比例最高的前几个话题。

表 4–3 $\lambda = 0.1$ 时 OEM-full 与 OEM-appr 的计算时间（秒）

CITER PERCENTAGE	2%	5%	10%	20%	30%	50%	100%
OEM-FULL	0.13	0.43	0.87	1.42	1.96	2.61	3.91
OEM-APPR	0.06	0.22	0.41	0.70	0.95	1.29	1.94

表 4–4 citer 百分比为 10% 时的平均测试 log-likelihood

λ	10^{-4}	0.01	0.1	0.5	1	2	10^{4}
LOG-LIKELIHOOD	-8.61	-8.33	-8.15	-8.28	-8.33	-8.35	-8.56

10% 后，预测性能随着 citer 百分比的提高较小，而时间消耗却有很大的增加，这意味着选择 10% 为 citer 百分比是合理的。总而言之，我们的模型对于这些超参数并不敏感。

表 4–5 $\lambda = 0.1$ 时的平均测试 log-likelihood

CITER PERCENTAGE	2%	5%	10%	20%	30%	50%	100%
LOG-LIKELIHOOD	-8.94	-8.43	-8.15	-8.10	-8.09	-8.03	-7.98
AVERAGE TIME	0.13	0.43	0.87	1.42	1.96	2.61	3.91

我们从 arXiv-TH 数据集中选择了 2 个文章集合来说明文章的话题演变。为了避免混乱，我们对每个文章集合的话题比例取平均，图中只画出了平均的话题比例。由于话题数共有 50 个，我们只选择了占的比例最大的话题。具体地说，令 $S_t = \{r_1, r_2, \ldots, r_l\}$ 表示在时间 t 被引用的文章集合（同一个集合中的文章被用一篇文章引用)。$\phi_t = \frac{1}{l}\sum_{i=1}^{l} \boldsymbol{\omega}_{r_i}$ 则是文章集合 S_t 的平均话题向量。这里选择了 S_{8001} 与 S_{8005} 作为说明的例子，如图4–1（g）与（h)。

从图4–1（g）可知，话题 7 的比例（即 $\phi_{8001}^{(7)}$）与话题 46 的比例（即 $\phi_{8001}^{(46)}$）是随着时间下降的。然而话题 15 的比例（$\phi_{8001}^{(15)}$）与话题 44 的比例（$\phi_{8001}^{(44)}$）则相反。一个解释是这个在第 8001 个时间点被引用的文章集合原来是关于某个物理学的子领域，但是随着时间的推移，这些文章的价值被其他子领域的研究者发现了。再被其他子领域的文章引用了足够多次之后，这个文章集合的话题开始从原话题（话题 7 与话题 46）向新话题（话题 15 与话题 44）转移。同样的事情会发生在统计学、心理学等领域（原领域）与机器学习等领域（新领域）上面。在第 8005 个时间点被引用的文章集合（S_{8005}）的话题演变与第 8001 个时间点的类似，如图4–1（h）所示。

4.5 相关工作

动态网络分析（DNA）已经广泛地被诸如社交网络、引用网络、邮件网络 [75–77] 等领域的相关研究者所研究。然而，大多数现有的工作 [63–65, 78–80] 的关注点要不就在微细粒度的小型网络上，要不就在极粗的细粒度下的大型网络上。虽然 DEM[73] 可以处理微细粒度下大型网络的数据，然而由于它的参数与话题特征固定，使得预测性能随着时间下降严重。针对纵向时间网络的连续时间回归模型 [81] 允许参数随时间变化，但是不同的是，模型使用的是边而非节点作为建模的中心，更重要的是，这个模型中没有考虑到话题特征的时变性质。但是在我们的模型 OEM 中，话题信息的变化被很好的整合到模型的建立、训练与测试三个阶段。

关于我们工作的另一条研究主线是话题模型（topic models)。通过拓展原来的 LDA 话

题模型 [82]，研究者们提出了许多模型以对话题的演变进行建模 [83–86]。还有一些方法可以对网络结构与节点特征同时建模 [87–89]。一般来说，在时变数据中同时更新话题与话题比例是十分耗费时间的。

我们选择直接地调整文章的话题比例而非使用现有的在线 LDA 模型 [74, 90]。这是因为 LDA 的在线推断（online inference）涉及到文章本身的文本内容信息，这将会花费很多时间在更新 LDA 向量上。然而在 OEM 中，我们只需要解几个小型的优化问题即可更新向量。

4.6 本章小结

在本章中，我们提出了在线自中心模型（OEM）以对时变的引用网络进行建模。通过随着时间调整、学习模型参数与话题特征，使得 OEM 克服了 DEM 的缺点（准确率随着时间严重下降)。在两个真实数据集上的实验结果表明，OEM 在实际应用中能达到十分可观的预测性能。

虽然本章的实验仅限于文章引用网络，如 [73] 所说，我们的模型也可以适用于其他类型的网络。这也会成为我们未来研究的问题之一。

本章的工作已经发表在国际顶级会议 IJCAI（International Joint Conference on Artificial Intelligence）2013 上。论文题目为“Online Egocentric Models for Citation Networks”。本文作者为第一作者。

第五章　全文总结

本文提出了 RCTR、CTR-SR 与 OEM 等三个模型。RCTR 通过“物品关系偏移”向量将物品间的社交网络信息整合进层级贝叶斯模型中，同时提出了“链接概率函数族”的概念，在大大提高预测性能的同时降低了训练的总时间。CTR-SR 通过将物品间的社交网络信息处理成拉普拉斯矩阵后作为先验，无缝地整合入层级贝叶斯模型中，缓解了推荐系统的稀疏问题，在真实数据上的相关实验也表明，CTR-SR 的预测性能要明显地高于其他最先进的基线而且能得到具有很好解释性的结果。需要强调的是，虽然乍一看 RCTR 与 CTR-SR 同样是结合了物品间的社交网络信息，然而两个模型有着很大的不同。从贝叶斯建模的角度讲，RCTR 将物品之间的链接关系视为已观察的变量，而 CTR-SR 将这些关系处理为一个拉普拉斯矩阵，作为决定物品隐向量（latent vector）的一个先验（prior）。从优化算法的角度讲，RCTR 使用直接的 Coordinate ascent 方法，而 CTR-SR 利用了 Steepest descent 讲每步优化的时间复杂度从三次方降到线性。从应用的角度讲，RCTR 倾向于向用户推荐物品，而 CTR-SR 倾向于向物品推荐标签（tag）。RCTR 与 CTR-SR 关注的都是静态的社交网络信息，在动态网络分析方面，OEM 通过将在线学习模型参数与话题特征转化为凸优化问题，通过交替地迭代，成功地对大规模微细粒度的动态社交网络进行建模，解决了 DEM 的模型缺陷，随着时间学习动态引用网络中随着时间变化的参数及节点特征，同样地，真实数据上的实验表明，OEM 不仅能够防止预测准确率随时间下降，而且能够揭示引用网络中话题随着时间的变化。

参考文献

[1] Linden G, Smith B, York J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering[J]. IEEE Internet Computing, 2003, 7(1):76–80.

[2] Bennett J, Lanning S. The Netflix prize[C]//Proceedings of KDD Cup and Workshop. .[S.l.]: [s.n.] , 2007.

[3] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6):734–749.

[4] Su X, Khoshgoftaar T M. A Survey of Collaborative Filtering Techniques[J]. Adv. Artificial Intellegence, 2009, 2009.

[5] Balabanović M, Shoham Y. Fab: content-based, collaborative recommendation[J]. Commun. ACM, 1997, 40(3):66–72.

[6] Lang K. Newsweeder: Learning to filter netnews[C]//ICML. .[S.l.]: [s.n.] , 1995.

[7] Breese J, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]//UAI. .[S.l.]: [s.n.] , 1998.

[8] Herlocker J L, Konstan J A, Borchers A, et al. An Algorithmic Framework for Performing Collaborative Filtering[C]//SIGIR.[S.l.]: ACM, 1999.

[9] Salakhutdinov R, Mnih A. Probabilistic Matrix Factorization[C]//. Platt J C, Koller D, Singer Y, et al. NIPS.[S.l.]: Curran Associates, Inc., 2007.

[10] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//WWW. .[S.l.]: [s.n.] , 2001.

[11] Basilico J, Hofmann T. Unifying collaborative and content-based filtering[C]//ICML. .[S.l.]: [s.n.] , 2004.

[12] Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles[C]//KDD. .[S.l.]: [s.n.] , 2011.

[13] Zhen Y, Li W J, Yeung D Y. TagiCoFi: tag informed collaborative filtering[C]//. Bergman L D, Tuzhilin A, Burke R D, et al. RecSys.[S.l.]: ACM, 2009.

[14] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet Allocation[J]. Journal of Machine Learning Research, 2003, 3:993–1022.

[15] Schein A I, Popescul A, Ungar L H, et al. Methods and metrics for cold-start recommendations[C]//SIGIR. .[S.l.]: [s.n.] , 2002.

[16] Jamali M, Ester M. *TrustWalker*: a random walk model for combining trust-based and item-based recommendation[C]//. IV J F E, Fogelman-Soulié F, Flach P A, et al. KDD.[S.l.]: ACM, 2009.

[17] Ma H, Yang H, Lyu M R, et al. SoRec: social recommendation using probabilistic matrix factorization[C]//. Shanahan J G, Amer-Yahia S, Manolescu I, et al. CIKM.[S.l.]: ACM, 2008.

[18] Purushotham S, Liu Y, Kuo C C J. Collaborative Topic Regression with Social Matrix Factorization for Recommendation Systems[C]//ICML. .[S.l.]: [s.n.] , 2012.

[19] Wang M, Ni B, Hua X S, et al. Assistive tagging: A survey of multimedia tagging with human-computer joint exploration[J]. ACM Comput. Surv., 2012, 44(4):25.

[20] Chen H M, Chang M H, Chang P C, et al. SheepDog: group and tag recommendation for flickr photos by automatic search-based learning[C]//ACM Multimedia. .[S.l.]: [s.n.] , 2008:737–740.

[21] Lipczak M, Hu Y, Kollet Y, et al. Tag Sources for Recommendation in Collaborative Tagging Systems[C]//ECML PKDD Discovery Challenge 2009 (DC09). 2009. .[S.l.]: [s.n.] , CEUR-WS.org, vol. 497. http://ceur-ws.org/Vol-497/paper_19.pdf.

[22] SHEN Y, FAN J. Leveraging loosely-tagged images and inter-object correlations for tag recommendation[C]//ACM Multimedia. .[S.l.]: [s.n.] , 2010:5–14.

[23] LEE S, NEVE W D, PLATANIOTIS K N, et al. MAP-based image tag recommendation using a visual folksonomy[J]. Pattern Recognition Letters, 2010, 31(9):976–982.

[24] TODERICI G, ARADHYE H, PASCA M, et al. Finding meaning on YouTube: Tag recommendation and category discovery[C]//CVPR. .[S.l.]: [s.n.] , 2010:3447–3454.

[25] CHEN L, XU D, TSANG I W H, et al. Tag-based web photo retrieval improved by batch mode re-tagging[C]//CVPR. .[S.l.]: [s.n.] , 2010:3440–3446.

[26] BENZ D, TSO K H L, SCHMIDT-THIEME L. Automatic Bookmark Classification - A Collaborative Approach[C]//Proceedings of the 2nd Workshop in Innovations in Web Infrastructure (IWI2) at WWW2006. Edinburgh, Scotland: [s.n.] , 2006.

[27] XU Z, FU Y, MAO J, et al. Towards the semantic web: Collaborative tag suggestions[C]//Proceedings of the Collaborative Web Tagging Workshop at the WWW 2006. Edinburgh, Scotland: [s.n.] , 2006.

[28] HOTHO A, JÄSCHKE R, SCHMITZ C, et al. Information Retrieval in Folksonomies: Search and Ranking[C]//ESWC. .[S.l.]: [s.n.] , 2006:411–426.

[29] MARINHO L B, SCHMIDT-THIEME L. Collaborative Tag Recommendations[C]//GFKL. .[S.l.]: [s.n.] , 2007:533–540.

[30] SIGURBJÖRNSSON B, VAN ZWOL R. Flickr tag recommendation based on collective knowledge[C]//WWW. .[S.l.]: [s.n.] , 2008:327–336.

[31] GARG N, WEBER I. Personalized, interactive tag recommendation for flickr[C]//RecSys. .[S.l.]: [s.n.] , 2008:67–74.

[32] WEINBERGER K Q, SLANEY M, VAN ZWOL R. Resolving tag ambiguity[C]//ACM Multimedia. .[S.l.]: [s.n.] , 2008:111–120.

[33] WU L, YANG L, YU N, et al. Learning to tag[C]//WWW. .[S.l.]: [s.n.] , 2009:361–370.

[34] Rendle S, Schmidt-Thieme L. Pairwise interaction tensor factorization for personalized tag recommendation[C]//WSDM. .[S.l.]: [s.n.] , 2010:81–90.

[35] Sevil S G, Kucuktunc O, Duygulu P, et al. Automatic tag expansion using visual similarity for photo sharing websites[J]. Multimedia Tools Appl., 2010, 49(1):81–99.

[36] Lops P, de Gemmis M, Semeraro G, et al. Content-based and collaborative techniques for tag recommendation: an empirical evaluation[J]. J. Intell. Inf. Syst., 2013, 40(1):41–61.

[37] Su X, Khoshgoftaar T M. A Survey of Collaborative Filtering Techniques[J]. Advances in Artificial Intelligence, 2009.

[38] Mooney R J, Roy L. Content-based book recommending using learning for text categorization[C]//ACM DL. .[S.l.]: [s.n.] , 2000.

[39] Basilico J, Hofmann T. Unifying collaborative and content-based filtering[C]//. Brodley C E. 2004. ICML.[S.l.]: ACM, ACM International Conference Proceeding Series, vol. 69.

[40] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet Allocation[J]. Journal of Machine Learning Research, 2003, 3:993–1022.

[41] Chang J, Blei D M. Relational Topic Models for Document Networks[J]. Journal of Machine Learning Research - Proceedings Track, 2009, 5:81–88.

[42] Koren Y, Bell R M, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. IEEE Computer, 2009, 42(8):30–37.

[43] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]//KDD. .[S.l.]: [s.n.] , 2008.

[44] Hofmann T. Latent semantic models for collaborative filtering[J]. ACM Trans. Inf. Syst., 2004, 22(1):89–115.

[45] Hu Y, Koren Y, Volinsky C. Collaborative Filtering for Implicit Feedback Datasets[C]//ICDM.[S.l.]: IEEE Computer Society, 2008.

[46] Gupta M, Li R, Yin Z, et al. Survey on social tagging techniques[J]. SIGKDD Explorations, 2010, 12(1):58–72.

[47] Lops P, de Gemmis M, Semeraro G. Content-based Recommender Systems: State of the Art and Trends.[M]//. Ricci F, Rokach L, Shapira B, et al. Recommender Systems Handbook.[S.l.]: Springer, 2011:73–105.

[48] Koren Y, Bell R M, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. IEEE Computer, 2009, 42(8):30–37.

[49] Gupta A, Nagar D. Matrix Variate Distributions[M], Chapman & Hall/CRC Monographs and Surveys in Pure and Applied Mathematics.[S.l.]: Chapman & Hall, 2000. http://books.google.com.hk/books?id=PQOYnT7P1loC.

[50] Gales M J F, Airey S S. Product of Gaussians for speech recognition[J]. CSL, 2006, 20(1):22–40.

[51] Li W J, Yeung D Y. Relation Regularized Matrix Factorization[C]//. Boutilier C. IJCAI. .[S.l.]: [s.n.] , 2009.

[52] Shewchuk J R. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain[R]. Pittsburgh, PA, USA: Carnegie Mellon University, 1994.

[53] Goldenberg A, Zheng A X, Fienberg S E, et al. A Survey of Statistical Network Models[J]. Foundations and Trends in Machine Learning, 2009, 2:129–233.

[54] Li W J, Yeung D Y, Zhang Z. Probabilistic Relational PCA[C]//NIPS. .[S.l.]: [s.n.] , 2009:1123–1131.

[55] Li W J, Zhang Z, Yeung D Y. Latent Wishart Processes for Relational Kernel Learning[J]. Journal of Machine Learning Research - Proceedings Track, 2009, 5:336–343.

[56] Wang C, Han J, Jia Y, et al. Mining advisor-advisee relationships from research publication networks[C]//ACM SIGKDD. .[S.l.]: [s.n.] , 2010.

[57] Li W J, Yeung D Y, Zhang Z. Generalized Latent Factor Models for Social Network Analysis[C]//IJCAI. .[S.l.]: [s.n.] , 2011:1705–1710.

[58] Li W J, Yeung D Y. Sparse Probabilistic Relational Projection[C]//AAAI. .[S.l.]: [s.n.] , 2012.

[59] Zhu J. Max-Margin Nonparametric Latent Feature Models for Link Prediction[C]//ICML. .[S.l.]: [s.n.] , 2012.

[60] McAuley J J, Leskovec J. Learning to Discover Social Circles in Ego Networks[C]//NIPS. .[S.l.]: [s.n.] , 2012.

[61] Kim M, Leskovec J. Latent Multi-group Membership Graph Model[C]//ICML. .[S.l.]: [s.n.] , 2012.

[62] Myers S A, Zhu C, Leskovec J. Information diffusion and external influence in networks[C]//ACM SIGKDD. .[S.l.]: [s.n.] , 2012.

[63] Fu W, Song L, Xing E P. Dynamic mixed membership blockmodel for evolving networks[C]//ICML. .[S.l.]: [s.n.] , 2009.

[64] Wyatt D, Choudhury T, Bilmes J. Discovering long range properties of social networks with multi-valued time-inhomogeneous models[C]//AAAI. .[S.l.]: [s.n.] , 2010.

[65] Hanneke S, Fu W, Xing E P. Discrete temporal models of social networks[J]. Electronic Journal of Statistics, 2010, 4:585–605.

[66] Richard E, Gaiffas S, Vayatis N. Link prediction in graphs with autoregressive features[C]//NIPS. .[S.l.]: [s.n.] , 2012.

[67] Sarkar P, Chakrabarti D, Jordan M I. Nonparametric link prediction in dynamic networks[C]//ICML. .[S.l.]: [s.n.] , 2012.

[68] Jin Y, Lin C Y, Matsuo Y, et al. Mining longitudinal network for predicting company value[C]//IJCAI. .[S.l.]: [s.n.] , 2011.

[69] Wang S, Groth P T. A framework for longitudinal influence measurement between communication content and social networks[C]//IJCAI. .[S.l.]: [s.n.] , 2011.

[70] Nori N, Bollegala D, Ishizuka M. Interest prediction on multinomial, time-evolving social graph[C]//IJCAI. .[S.l.]: [s.n.] , 2011.

[71] Wasserman S. Analyzing social networks as stochastic processes[J]. Journal of the American Statistical Association, 1980, 75(370):280–294.

[72] Snijders T A B. Models for longitudinal network data[J]. Models and Methods in Social Network Analysis, 2005:215–247.

[73] Vu D, Asuncion A U, Hunter D, et al. Dynamic egocentric models for citation networks[C]//ICML. .[S.l.]: [s.n.] , 2011.

[74] Hoffman M D, Blei D M, Bach F R. Online learning for latent Dirichlet allocation[C]//NIPS. .[S.l.]: [s.n.] , 2010.

[75] Leskovec J, Kleinberg J M, Faloutsos C. Graphs over time: densification laws, shrinking diameters and possible explanations[C]//ACM SIGKDD. .[S.l.]: [s.n.] , 2005.

[76] Kossinets G, Watts D J. Empirical analysis of an evolving social network[J]. Science, 2006, 311(5757):88–90.

[77] Viswanath B, Mislove A, Cha M, et al. On the evolution of user interaction in Facebook[C]//ACM Workshop on Online Social Networks. .[S.l.]: [s.n.] , 2009.

[78] Sarkar P, Moore A W. Dynamic social network analysis using latent space models[J]. SIGKDD Explorations, 2005, 7(2):31–40.

[79] Foulds J R, DuBois C, Asuncion A U, et al. A dynamic relational infinite feature model for longitudinal social networks[C]//AISTATS. .[S.l.]: [s.n.] , 2011.

[80] Ho Q, Song L, Xing E P. Evolving cluster mixed-membership blockmodel for time-evolving networks[C]//AISTATS. .[S.l.]: [s.n.] , 2011.

[81] Vu D, Asuncion A U, Hunter D, et al. Continuous-time regression models for longitudinal networks[C]//NIPS. .[S.l.]: [s.n.] , 2011.

[82] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3:992–1022.

[83] Wang C, Blei D M, Heckerman D. Continuous Time Dynamic Topic Models[C]//UAI. .[S.l.]: [s.n.] , 2008.

[84] Wang C, Thiesson B, Meek C, et al. Markov topic models[C]//AISTATS. .[S.l.]: [s.n.] , 2009.

[85] Chen C, Ding N, Buntine W L. Dependent hierarchical normalized random measures for dynamic topic modeling[C]//ICML. .[S.l.]: [s.n.] , 2012.

[86] Dubey A, Hefny A, Williamson S, et al. A non-parametric mixture model for topic modeling over time[C]//SDM. .[S.l.]: [s.n.] , 2013.

[87] Kataria S, Mitra P, Caragea C, et al. Context sensitive topic models for author influence in document networks[C]//IJCAI. .[S.l.]: [s.n.] , 2011.

[88] Hu Y, John A, Wang F, et al. ET-LDA: joint topic modeling for aligning events and their twitter feedback[C]//AAAI. .[S.l.]: [s.n.] , 2012.

[89] Krafft P, Moore J, Wallach H, et al. Topic-partitioned multinetwork embeddings[C]//NIPS. .[S.l.]: [s.n.] , 2012.

[90] Canini K R, Shi L, Griffiths T L. Online inference of topics with latent Dirichlet allocation[C]//AISTATS. .[S.l.]: [s.n.] , 2009.

[91] Wang H, Li W J. Online Egocentric Models for Citation Networks[C]//IJCAI. .[S.l.]: [s.n.] , 2013.

致　　谢

感谢我的导师李武军老师，他两年来不辞辛劳的指导让我一步步从对机器学习这个领域入门到初有成果，同时也让我学到了许多东西，没有李老师也就不可能有前面的这些工作。感谢陈彬毅同学在 CTR-SR 部分所做的工作，他强大的编程能力使得这部分工作如期发表成为可能。除此之外还要感谢实验室的其他成员，是大家的谅解和帮助使得大量相关的实验可以有条不紊地进行。

论文发表

[1] Hao Wang（王灏）, Binyi Chen, Wu-Jun Li. Collaborative Topic Regression with Social Regularization for Tag Recommendation. Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI), 2013.

[2] Hao Wang（王灏）, Wu-Jun Li. Online Egocentric Models for Citation Networks. Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI), 2013.

[3] Hao Wang（王灏）, Wu-Jun Li. Relational Collaborative Topic Regression for Recommendation Systems. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2013. (submitted)

Large-Scale Social Network Data Mining with Multi-View Information

In recently years, social network services (SNS) have become increasingly popular. Social network sites like Twitter, Facebook are playing more and more important roles in people's lives and have dramatically changed our ways of communication, purchasing goods, and many other things. Besides, the fever of SNS also provides plenty of data for academia and industry. These data can not only be used to research social relationships between people, but also provide precious evidence for all kinds services such as product recommendation and online advertising. Actually in a broader sense, there also exist social networks between products in the online malls or between scientific articles. What's more, social networks themselves are evolving over time, which means the trace of their evolution can also be a source of information (here we call it a view). Thus how to fully utilize the available social network information (both static information and dynamic information) should be a problem worth diving into.

On the other hand, recommender systems (RS) are playing very important role for us to make effective use of information. For example, Amazon [1] adopts RS for product recommendation, and Netflix [2] uses RS for movie recommendation. Existing RS methods can be roughly categorized into three classes [3, 37]: content-based methods, collaborative filtering (CF) based methods, and hybrid methods. Content-based methods adopt the profile of the users or products for recommendation. CF based methods use the past activities or preferences, such as the ratings on items given by users, for prediction, without using any user or product profiles. Hybrid methods combine both content-based methods and CF based methods by ensemble techniques.

In most traditional CF methods, only the rating matrix, which contains the ratings on the items given by users, is used for training and prediction. Typically, the rating matrix is very sparse, which means that most users only rate very few items.

To overcome the sparsity problem of CF-based models, many researchers have proposed to integrate auxiliary information into the model training and prediction procedure. Some methods [12, 13, 39] utilize the item content (attributes) to facilitate the CF training. Among these methods, collaborative topic regression (CTR) [12] is the most recent one which jointly models the user-item rating matrix and the item content information (texts of articles). CTR seamlessly incorporates topic modeling [40] with CF to improve the performance and interpretability. For unrated (new) items, CTR is able to perform out-of-matrix prediction (cold-start prediction) [13, 15] using only the content information. Some other methods [16–18] try to use social networks among *users* to improve the performance. Among these methods, CTR-SMF [18] extends CTR by integrating the social network among users into CTR with social matrix factorization (SMF)[17] techniques, which has achieved better performance than CTR.

In many real applications, besides the rating and item content information, there may exist relations (or called social networks) among the *items* which can also be very helpful for recommendation. For example, if we want to recommend papers (references) to users in CiteULike[1], the citation relations between papers are very informative for recommending papers with similar topics. In this paper, we develop a novel hierarchical Bayesian model, called Relational Collaborative Topic Regression (RCTR), to incorporate *item relations* for recommendation. The main contributions of RCTR are outlined as follows:

- By extending CTR, RCTR seamlessly integrates the user-item rating information, item content information and relational (network) structure among items into a principled hierarchical Bayesian model.
- Even if a new user has only rated one or two items, RCTR can still make effective use of the information from the item network to alleviate the data sparsity problem in CF, which will consequently improve the recommendation accuracy.
- In RCTR, a family of link (relation) probability functions is proposed to model the relations between items. This extension from discrete link probability functions like those in [41] to

[1] `http://www.citeulike.org/`

a family of link probability functions increases the modeling capacity of RCTR with better performance.

- Compared with CTR, a smaller number of learning iterations are needed for RCTR to achieve satisfactory accuracy. As a consequence, the total learning time complexity of RCTR is much lower than that of CTR even if the time complexity of each iteration of RCTR is slightly higher than that of CTR.

- RCTR can learn very good interpretable latent structures which are very useful for recommendation.

- Experiments on real-world datasets show that RCTR can achieve higher prediction accuracy than the state-of-the-art methods.

Besides item recommender system, tagging systems have been playing very important role for us to better categorize and organize information. For example, Flickr[2] uses tags to label and organize photos, Last.fm[3] adopts tags to categorize artists and music, and CiteULike[4] allows users to tag articles. With the tagging systems, users are able to better organize their own content and find relevant resources (content) more easily.

However, finding the set of proper words (tags) to describe the resources often requires high mental focus. That is why tag recommendation (TR) [19, 46] has become more and more important on the Internet. With the tag recommendation system, users only need a few clicks to finish the tagging process. Existing tag recommendation methods can be roughly categorized into three classes [19]: content-based methods, co-occurrence based methods, and hybrid methods. Because the TR problem is very complex and difficult, neither co-occurrence based methods nor content based methods can achieve satisfactory performance in real TR applications. Hence, the recent trend in TR research is to use hybrid methods which try to combine both item-tag matrix and item content information together for recommendation.

However, it is still a challenge to find an effective way to combine both item-tag matrix and

[2] http://www.flickr.com
[3] http://www.lastfm.com
[4] http://www.citeulike.org

item content information for TR. Furthermore, in some applications there may exist social networks (relations) between items. For example, if we want to tag articles in CiteULike, there are citation relations or other social networks between articles [57, 91]. Typically, two articles with relation between them might be most likely to be about the same topic [54, 55], and consequently they should have similar tags. Hence, how to effectively integrate social networks between items for tagging is another challenge.

To solve the above challenges we propose a model called CTR-SR. The main contributions of this model can be outlined as follows:

- We adapt the *collaborative topic regression* (CTR) model [12], which has been successfully applied for article recommendation, to combine both item-tag matrix and item content information for tag recommendation in a principled way.
- By extending CTR, we propose a novel hierarchical Bayesian model, called *CTR with social regularization* (CTR-SR), to seamlessly integrate the item-tag matrix, item content information, and social networks between items into the same principled model.
- Extensive experiments on real-world data sets show that CTR can outperform the baselines which use only one kind of information, either item-tag matrix or item content information. Furthermore, CTR-SR can effectively utilize the social networks between items to further improve the performance.

Besides static social network information mentioned above, there also exists dynamic information, namely the trace of evolution of networks. In fact Dynamic Network Analysis (DNA) has already become a very hot research topic recently. Although there have been a lot of works on DNA, most of them either focus on large-scale data at a very rough temporal granularity [63–70] or focus on small networks at a fine temporal granularity [71, 72]. Recently, dynamic egocentric model (DEM) [73], which is based on multivariate counting processes, has been successfully proposed to model large-scale evolving citation networks at a fine temporal granularity of individual time-stamped events.

Although DEM can dynamically update the *link features* (statistics) of the nodes (papers), the

learned *parameters* and *topic features* of DEM are static (fixed) during the prediction process for evolving networks. Hence, DEM suffers from a decrease of accuracy over time because typically both the *parameters* and the *topic features* of the papers will evolve over time. Without the ability to adaptively learn the parameters and topic features, DEM fails to model the evolution of networks. This phenomenon of decreasing prediction accuracy over time can also be observed from the experimental results in Figure 2 of [73].

Thus we propose an online extension of DEM, called *online egocentric model* (OEM), to capture the evolution of both topic features and model parameters. The contributions of this paper are briefly outlined as follows:

- OEM takes the evolution of both topic features and parameters into consideration and maintains high prediction accuracy regardless of the elapse of time.
- During the online training of OEM, we can also uncover the evolution of topic features for each paper and the propagation of topic features between pairs of papers.
- Extensive experiments on two real-world citation networks are performed to demonstrate the effectiveness of our novel model.

In summary we propose three separate models (RCTR, CTR-SR, and OEM) to apply social network analysis and integrate the social information into principled models to solve existing problems while significantly improve models' predictive performance.