异构信息网络中基于元结构的协同过滤算法

王 旭[1] 庞 巍[2] 王 喆[1]
(吉林大学计算机与科学技术学院 长春 130012)[1]
(阿伯丁大学自然与计算科学学院 阿伯丁 AB253TN)[2]

摘 要 近年来,异构信息网络由于包含丰富的语义信息引起了众多研究者的关注。已有的研究已经证实异构信息网络中丰富的关系信息能够提高推荐效果。作为一种挖掘异构信息网络中关系信息的重要工具,元路径已经被广泛地应用到许多算法中,然而元路径受到线性结构的限制,不能表示更加复杂的关系信息。为了解决这一问题,文中提出了一种新的推荐系统算法,即 MetaStruct-CF。该算法利用元结构来挖掘异构信息网络中丰富的关系信息。不同于现有的一些算法,该算法结合了多种信息,以有效地利用异构信息网络中丰富的信息。两个真实世界数据集上的大量实验表明,MetaStruct-CF 能够有效地提高推荐效果。

关键词 异构信息网络,推荐系统,协同过滤,元结构
中图法分类号 TP391 **文献标识码** A

MetaStruct-CF: A Meta Structure Based Collaborative Filtering Algorithm in Heterogeneous Information Networks

WANG Xu[1] PANG Wei[2] WANG Zhe[1]
(College of Computer Science and Technology, Jilin University, Changchun 130012, China)[1]
(School of Natural and Computing Sciences, University of Aberdeen, Aberdeen AB253TN, United Kingdom)[2]

Abstract In recent years, heterogeneous information networks (HINs) have received a lot of attention as they contain rich semantic information. Previous works have demonstrated that the rich relationship information in HINs can effectively improve the recommendation performance. As an important tool for mining relationship information in HINs, meta-path has been widely used in many algorithms. However, because of its simple linear structure, meta-path may not be able to express complex relationship information. To address this issue, this paper proposed a new recommendation algorithm, Metastruct-CF, which applies Meta structure to capture the accurate relationship information among data objects. Different from existing methods, the proposed combines algorithm multiple relationships to effectively utilize the information in HINs. Extensive experiments on two real world datasets show that this algorithm achieves better recommendation performance than several popular or state-of-the-art methods.

Keywords Heterogeneous information network, Recommendation systems, Collaborative filtering, Meta structure

1 引言

随着大数据时代的到来,互联网用户经常遇到信息过载的问题。推荐系统作为解决这个问题的有效工具之一,已经被许多公司广泛使用,如亚马逊[1]和腾讯[2]。在许多提出的推荐算法中,协同过滤[3]算法被广泛使用,它使用历史用户的历史评分来为用户推荐可能感兴趣的物品。但是由于物品数量庞大,用户经常只能对小部分的物品进行评分,从而导致了数据稀疏问题[4]。另外,对于一个新用户,由于缺乏评分信息,很难对其做出合适的推荐,因此推荐系统往往还面临着冷启动问题。

为了解决数据稀疏和冷启动问题,研究者们提出了许多不同的算法。研究人员发现,他们可以通过利用用户或物品之间的关系来提高推荐效果[5-6]。因为具有相似兴趣的人倾向于喜欢相同的物品,具有相似特征的物品更有可能被同一用户喜欢。而异构信息网络中蕴含着丰富的关系信息,可以用来提高推荐效果。在异构信息网络中,节点之间存在不同类型的节点和链接,它们表示不同种类的关系并蕴含了丰富的语义信息。一些研究人员开始提出基于异构信息网络的推荐算法。Yang 等[7]提出 TrustMF 推荐算法,该算法引入了社交网络中用户之间的关系。Yu 等[8]提出了一种个性化推荐算法,通过组合用户之间的不同关系来提高推荐效果。Jamali 等[9]提出了一个上下文相关矩阵分解模型 HETEROMF,综合考虑了全局信息和上下文相关的因素。上述研究表明,通过从异构信息网络中提取到的更多信息,我们可以设计出更好的推荐算法。

在异构信息网络的关系信息提取的过程中,如何计算网络中节点之间的相似性是一个重要的问题。为了解决这个问

本文受国家自然科学基金项目(61472159),吉林省大数据智能计算重点实验室(20180622002JC),吉林省自然科学基金(20180101036JC)资助。
王 旭(1994—),男,硕士,CCF 会员,主要研究方法为数据挖掘;庞 巍(1979—),男,博士,主要研究方向为数据挖掘、机器学习;王 喆(1974—),男,博士,副教授,主要研究方向为社交网络与数据挖掘,E-mail:wz2000@jlu.edu.cn(通信作者)。

题，一些研究人员提出了很多方法。个性化 PageRank[10] 通过从源节点随机游走到目标节点的概率来计算节点之间的相似度。SimRank[11] 通过节点邻居的相似性来计算两个节点之间的相似度。但这些方法并未考虑网络节点和链接的类型信息，这意味着它们可能不适用于异构信息网络。为了解决这个问题，Sun 等[12] 提出了基于元路径的相似性度量 PathSim，以计算基于元路径的同类型节点之间的相似性。之后，一些研究人员提出了基于元路径的其他相似度度量。Lao 等[13] 提出了路径约束随机游走(PCRW)模型来测量标记有向图中的节点相似度。还有一些研究人员通过引入更丰富的信息[14] 扩展了 PathSim，如传递相似性和时间动力学等。Shi 等[6] 提出的加权异构信息网络和加权元路径概念以更加细微的方式描绘了异构信息网络中的语义信息。Shi 等[14] 提出 HeteSim 来测量任意元路径下任何类型节点之间的相似度。总的来说，这些工作通过结合网络结构信息和其他信息来计算异构信息网络中节点之间的相似性，元路径在这些工作中起着重要的作用。

虽然元路径已被证明在许多算法中非常有效，但它仍然有局限性，只能表示两个网络节点之间的简单关系。因此，元路径可能无法表示更加精细的语义信息。为了更有效地在异构信息网络中获取语义信息，Huang 等[15] 提出了一种更有效的工具 Meta Structure，其本质上是一个有向无环图。与元路径不同，元结构可以在网络节点之间表示更复杂的结构，从而形成更丰富的信息。

在提出元结构之前，已经存在一些使用元路径的基于异构信息网络的推荐算法，例如上述的一些算法；然而，据我们所知，没有基于 Meta Structure 的推荐算法。MetaStruct-CF 首先将 Meta Structure 应用于协同过滤推荐的算法。为了充分利用异构信息网络中包含的关系信息，我们在算法中引入了用户之间的关系信息和物品之间的关系信息。最后，我们将该方法应用于两个现实世界的数据集。实验结果证明了所提算法的有效性。

本文的贡献如下：

1)将元结构应用到推荐系统中，通过元结构获取更丰富的关系信息来提高推荐效率。

2)为了充分利用异构网络中丰富的关系信息，我们在 MetaStruct-CF 中引入了两种类型的关系信息，即用户之间的关系和物品之间的关系。

3)在两个真实数据集上进行了大量实验，评估了算法在多种指标下的性能。实验显示，MetaStruct-CF 可以有效地提高推荐性能。

本文第 2 节介绍了关于算法的背景知识；第 3 节介绍了 MetaStruct-CF 算法的框架；第 4 节进行了相关实验；最后，总结全文并探讨了未来的工作。

2 相关概念

本节介绍了本研究的背景知识，包括异构信息网络、元路径、元结构和协同过滤算法等。

2.1 异构信息网络

异构信息网络是包含多种类型节点和多种类型链接的信息网络[16]。异构信息网络的定义如下。

定义 1(异构信息网络[16]) 异构信息网络是一个有向图 $G=(V,E,A,R)$，其中 V 是顶点集，E 是边集，A 是顶点的类型，R 是边的类型。存在着从 V 到 A 的映射 $\varphi:V\rightarrow A$ 和从 E 到 R 的映射 $\psi:E\rightarrow R$，即网络中每个顶点都对应着 A 中的一种类型，每一条边都对应着 R 中的一种类型。其中，$|A|>1$，$|R|>1$。

图 1 给出了一个典型的异构信息网络，它是一个 DBLP 数据集的网络。在这个网络中，有 4 种类型的节点：作者、论文、关键词和会议。在这些网络节点之间存在着不同种类的关系，例如作者发表的论文、论文包含的关键词和会议包含的论文。

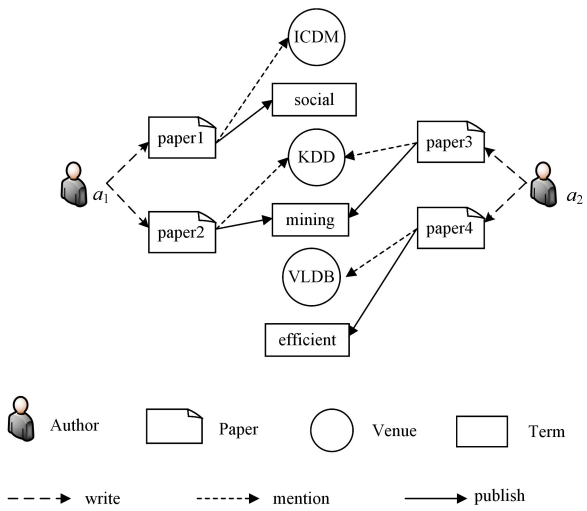

图 1 一个简单的异构信息网络

为了给出异构信息网络的元级描述，使用网络模式(Network Schema)的概念来描述网络的结构。其定义如下。

定义 2(网络模式)[17] 对于一个异构信息网络 $G=(V,E,A,R)$，它的网络模式 $T_G=(A,R)$ 是一个定义在 A 和 R 上的有向图。

图 2 给出了 DBLP 网络的模式。异构信息网络的网络模式描述了网络中顶点的类型和网络顶点之间的关系类型。例如，在 DBLP 网络中只有 4 种类型的顶点：作者(Author)、论文(paper)、会议(Conference)和关键词(Term)。

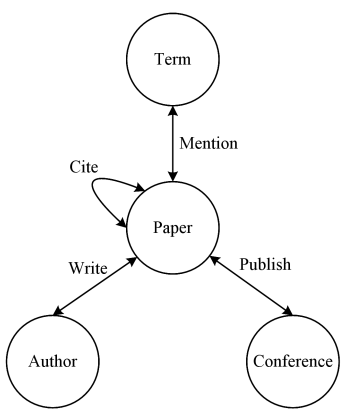

图 2 DBLP 网络模式

定义 3(元路径，Meta path)[17] 在一个异构信息的网络模式 $T_G=(A,R)$ 中，一个元路径 P 表示两个顶点之间的复合关系，$P=R_1\circ R_2\circ\cdots\circ R_{l-1}$，它的形式是 $A_1-A_2-\cdots-A_i$。

元路径表示两个网络顶点之间的关系，不同种类的元路径代表顶点之间的不同含义。因此，元路径可以有效地描述

顶点之间的不同语义信息,这使得它成为异构信息网络中挖掘关系信息的重要工具。

尽管 meta-path 具有很多优点,但仍然存在一些局限性:由于其结构简单,元路径不能描述顶点之间更复杂的语义信息。例如,图 1 中,作者 a_1 和 a_2 都在 KDD 中发表了关于"数据挖掘"的论文,单个元路径不能表示这种情况。

为了解决上述问题,Huang 等[15] 提出了元结构(Meta Structure)的概念。其定义如下。

定义 4(元结构)[15]　给定一个网络模式 $T_G=(A,R)$,一个元结构 $S=(N,M,n_s,n_t)$是一个有向无环图。元结构 S 从源节点 n_s 出发,到终点 n_s 结束。其中,N 是 A 的子集,M 是 R 的子集。

图 3 给出了元结构的一个例子。与 meta-path 不同,元结构具有更复杂和更灵活的结构,可以使它表示更复杂的关系。例如,图 3 所示的元结构包含的含义是 A_1 和 A_2 在同一会议中发表关于同一主题的论文,在这个元结构中,A 代表"作者",P 代表"论文",V 代表"会议",T 代表"关键词"。

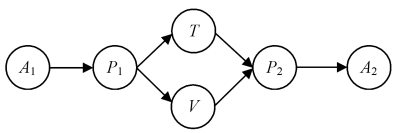

图 3　DBLP 中一个简单的元结构

基于元结构的节点相似性计算方法有 3 种[15],包括 StructCount,SCSE 和 BSCSE。其中 StructCount 和 SCSE 是 BSCSE 的特例,因此这里只介绍 BSCSE 及其相关概念。

定义 5(Biased Structure Constrained Subgraph Expansion,BSCSE)　BSCSE 的计算公式如下:

$$BSCSE(g,i|S,o_t)=\frac{\sum_{g'\in\sigma(g,i|S,G)}BSCSE(g',i+1|S,o_t)}{\|\sigma(g,i|S,G)\|^{\theta}} \tag{1}$$

其中,S 代表元结构;$o_s\in V$ 是源节点;$o_t\in V$ 是目标节点;g 是元结构的子图,由源节点 o_s 扩展而来;i 表示子图 g 的层数;$\theta\in(0,1)$是 BSCSE 中的一个参数,当 $\theta=0$ 时,BSCSE 等价于 StructCount,当 $\theta=1$ 时,BSCSE 等价于 SCSE。

2.2　矩阵分解

矩阵分解是推荐系统中一种有效的方法[17]。对于一个评分矩阵,矩阵分解算法学习到用户和物品的低维表示 U 和 V,使得:

$$R\approx U^{\mathrm{T}}V \tag{2}$$

其中,U 和 V 可以通过最小化以下损失函数得到:

$$\min_{U,V}\sum_{i=0}^{m}\sum_{j=0}^{n}(U_i^{\mathrm{T}}V_j-R_{i,j})^2+\lambda(\|U\|_F^2+\|V\|_F^2) \tag{3}$$

在以上的损失函数中,$\|*\|_F^2$ 是 Frobenius 范式。$\lambda(\|U\|_F^2+\|V\|_F^2)$是 L_2 正则化项。$U=[U_1,U_2,\cdots U_m]$和 $V=[V_1,V_2,\cdots V_n]$代表用户和物品的低维表示。

3　MetaStruct-CF 算法

本节将详细介绍 MetaStruct-CF 算法。与其他推荐算法不同,MetaStruct-CF 算法将元结构应用到推荐系统中。此外,我们将用户之间的关系和物品之间的关系统一到一个模型中。之后,介绍模型的学习方法,最后给出完整的 MetaStruct-CF 算法。

3.1　用户关系建模

在模型中引入用户之间的关系信息,推荐算法能够表现得更好[7]。因为在现实中物以类聚、人以群分,关系紧密的人通常有类似的兴趣。

例如,两位作者汤姆和彼得,他们有类似的研究领域,并且对机器学习感兴趣,如果汤姆经常在 ICML 中发表他的论文,Peter 也可能会在 ICML 中发表他的论文。在这种情况下,两位作者通过以下元路径连接:作者-论文-关键词-论文-作者,并且两个作者的相似性可以通过 PathSim[12] 计算得到。然而,元路径仍然存在着一些局限性,它们只能描述两个节点之间的简单关系。例如,汤姆和彼得都对机器学习感兴趣,他们可能会在同一会议发表论文。在这种情况下,一个元路径不能精确地捕获这种信息,但是图 4 中的元结构可以通过并入更复杂的结构来包含更多的信息。

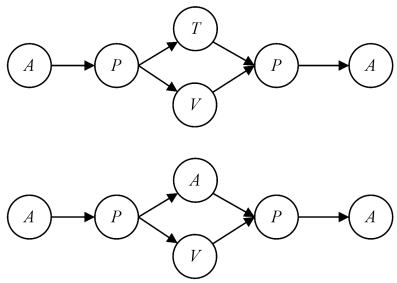

图 4　作者之间的两种元结构

通过图正则化[18]的方式将用户之间的关系信息加入到矩阵分解模型中,得到了式(4):

$$J=\sum_{i=0}^{m}\sum_{j=0}^{n}(U_i^{\mathrm{T}}V_j-R_{i,j})^2+\sum_{k=0}^{N_A}\alpha_k\sum_{i=0}^{m}\sum_{j=0}^{m}S_A^k(i,j)\|U_i-U_j\|_F^2+\lambda(\sum n_{user_i}\|U\|_F^2+\sum n_{item_i}\|V\|_F^2+\|A\|_F^2) \tag{4}$$

其中,U,V,R 与上文提到的含义相同,分别代表着用户、物品和评分;α_k 表示第 k 个元结构的重要性;$A=[\alpha_1,\alpha_2,\cdots,\alpha_{NA}]$,其中 N_A是元结构的种类数;S_A^k 是用户在第 k 个元结构下的相似度矩阵。为了避免算法训练中的过拟合,我们在目标函数的正则化项中引入了系数 n_{user_i} 和 n_{item_i}。n_{user_i} 是用户 i 评分的次数,n_{user_i} 是物品 i 被评分的次数。

3.2　物品关系建模

我们还可以通过引入物品之间的关系信息来提高推荐算法的性能[19]。例如,如果作者经常在 ICML 中发表论文,则可以向他推荐 NIPS。因为 NIPS 和 ICML 都是机器学习的顶级会议。在这种情况下,ICML 和 NIPS 之间的关系可以通过元路径:会议-论文-作者-论文-会议来描述。

但是,元路径只能使用有限的信息。例如,作者可能对几个不同的领域感兴趣,并在不同领域的会议发表论文。在这种情况下,一个元路径不能表达这样的信息。通过图 5 中的元结构,我们可以更有效地利用会议之间的关系信息。

类似地,我们给出了以下目标函数,该目标函数引入了物品之间的关系信息:

$$J=\sum_{i=0}^{m}\sum_{j=0}^{n}(U_i^{\mathrm{T}}V_j-R_{i,j})^2+\sum_{k=0}^{N_B}\beta_k\sum_{i=0}^{n}\sum_{j=0}^{n}S_B^k(i,j)\|V_i-V_j\|_F^2+\lambda(\sum n_{user_i}\|U\|_F^2+\sum n_{item_i}\|V\|_F^2+\|B\|_F^2) \tag{5}$$

其中,β_k表示第 k 个元结构的重要性,$B=[\beta_1,\beta_2,\cdots,\beta_{NB}]$,$N_B$

表示两个物品之间元结构的种类数，S_B^k 是物品在第 k 个元结构下的相似度矩阵，n_{user_i} 和 n_{user_j} 与式(4)具有相同的含义。

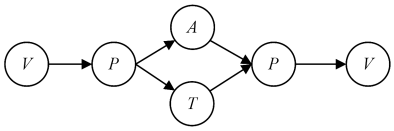

图 5 会议之间的元结构

3.3 MetaStruct-CF 模型

3.1 节和 3.2 节分别对用户关系和物品关系进行了建模，本小节将两种模型整合在一起得到 MetaStruct-CF 模型。最终的目标函数如下：

$$\min_{U,V,A,B,W} \sum_{i=0}^{m}\sum_{j=0}^{n}(U_i^{\mathrm{T}}V_j-R_{i,j})^2+\sum_{k=0}^{N_A}\alpha_k\sum_{i=0}^{m}\sum_{j=0}^{m}S_A^k(i,j)\parallel U_i-U_j\parallel_F^2+\sum_{k=0}^{N_B}\beta_k\sum_{i=0}^{n}\sum_{j=0}^{n}S_B^k(i,j)\parallel V_i-V_j\parallel_F^2+\lambda(\sum n_{user_i}\parallel U\parallel_F^2+\sum n_{item_i}\parallel V\parallel_F^2+\parallel A\parallel_F^2+\parallel B\parallel_F^2) \tag{6}$$

然而，这个目标函数不能直接进行优化，因此我们将图正则化项改写为迹的形式[19]，转换公式如下：

$$\sum_{k=0}^{N_A}\alpha_k\sum_{i=0}^{m}\sum_{j=0}^{m}S_A^k(i,j)\parallel U_i-U_j\parallel_F^2=Tr(U^{\mathrm{T}}(\sum_{k=0}^{N_A}\alpha_kL_A^k)U) \tag{7}$$

$$\sum_{k=0}^{N_A}\beta_k\sum_{i=0}^{n}\sum_{j=0}^{n}S_B^k(i,j)\parallel V_i-V_j\parallel_F^2=Tr(V^{\mathrm{T}}(\sum_{k=0}^{N_B}\beta_kL_B^k)V) \tag{8}$$

其中，L_A^k 是一个对角矩阵，$L_A^k=\sum_{j=0}^{m}S_A^k(i,j)-S_A^k$，相似地，$L_B^k$ 也是一个对角矩阵，$L_B^k=\sum_{j=0}^{m}S_B^k(i,j)-S_B^k$。

然后，基于式(6)－式(8)，最终的损失函数 J 为：

$$J=\sum_{i=0}^{m}\sum_{j=0}^{n}(U_i^{\mathrm{T}}V_j-R_{i,j})^2+Tr(U^{\mathrm{T}}(\sum_{k=0}^{N_A}\alpha_kL_A^k)U+Tr(V^{\mathrm{T}}(\sum_{k=0}^{N_B}\beta_kL_B^k)V)+\lambda(\sum n_{user_i}\parallel U\parallel_F^2+\sum n_{item_i}\parallel V\parallel_F^2+\parallel A\parallel_F^2+\parallel B\parallel_F^2) \tag{9}$$

我们采用随机梯度下降算法来最小化目标函数，函数 J 对 U,V,α,β 的偏导数如下：

$$\frac{1}{2}\frac{\partial J}{\partial U_i}=\sum_{j=0}^{n}(U_i^{\mathrm{T}}V_j-R_{i,j})V_j+U_i\sum_{k=0}^{N_A}\alpha_kL_A^k+\lambda n_{user_i}U_i \tag{10}$$

$$\frac{1}{2}\frac{\partial J}{\partial V_j}=\sum_{i=0}^{m}(U_i^{\mathrm{T}}V_j-R_{i,j})U_i+V_j\sum_{k=0}^{N_B}\beta_kL_B^k+\lambda n_{item_j}V_j \tag{11}$$

$$\frac{1}{2}\frac{\partial J}{\partial \alpha}=Tr(U^{\mathrm{T}}(\sum_{k=0}^{N_A}\alpha_kL_A^k)U)+\lambda\alpha \tag{12}$$

$$\frac{1}{2}\frac{\partial J}{\partial \beta}=Tr(V^{\mathrm{T}}(\sum_{k=0}^{N_B}\beta_kL_B^k)V)+\lambda\beta \tag{13}$$

然后，在每一次迭代中，U,V,α,β 通过如下方式进行更新：

$$U_i\leftarrow U_i-\alpha^s\frac{\partial J}{\partial U_i},i=0,\cdots,m-1 \tag{14}$$

$$V_i\leftarrow V_i-\alpha^s\frac{\partial J}{\partial V_i},i=0,\cdots,n-1 \tag{15}$$

$$\alpha_i\leftarrow\alpha_i-\alpha^s\frac{\partial J}{\partial \alpha_i},i=0,\cdots,N_A-1 \tag{16}$$

$$\beta_i\leftarrow\beta_i-\alpha^s\frac{\partial J}{\partial \beta_i},i=0,\cdots,N_B-1 \tag{17}$$

其中，$\alpha^s\in(0,1)$用来控制学习的速率。

通过式(14)－式(17)，我们得到最终的 U,V，进而得到预测的评分 $R=U^{\mathrm{T}}V$。

最后，我们给出了 MetaStruct-CF 算法的整体描述，如算法 1 所示。

算法 1 MetaStruct-CF 算法

输入：异构信息网络 G=(V,E,A,R)，用户之间的元结构，商品之间的元结构，实际评分矩阵 R

输出：预测的评分矩阵 R_pred

1. 随机初始化 U,V,α,β；
2. While U,V,α,β,两次迭代的差异大于指定的阈值
   2.1 使用式(14)对 U 进行更新；
   2.2 使用式(15)对 V 进行更新；
   2.3 使用式(16)对 α 进行更新；
   2.4 使用式(17)对 β 进行更新；
   End while
3. 评分矩阵 R_pred=U^{T}V。

4 实验结果

为了证明所提算法的有效性，我们在两种不同类型的数据集上进行测试，并且在多种评价标准下与其他算法做了对比。

4.1 数据集

实验中，我们选择 DBLP 和 Meetup 两个现实世界的异构信息网络数据集，来测试包括 MetaStruct-CF 在内的不同的推荐算法。

DBLP 数据集[20]已被广泛应用于异构信息网络研究。我们使用 DBLP 的一个子集作为实验数据，其中包含 10352 篇论文、2566 位作者和 18 个会议。DBLP 的网络模式如图 2 所示。

另一个数据集是 Meetup 数据集，它是从在线社交网站 Meetup 上抓取的，其包括在纽约市举行的活动，包含 46895 位用户和 398 个小组。Meetup 网络模式如图 6 所示。其中，U 代表“用户”，L 代表“地点”，G 代表“小组”，E 代表“事件”。

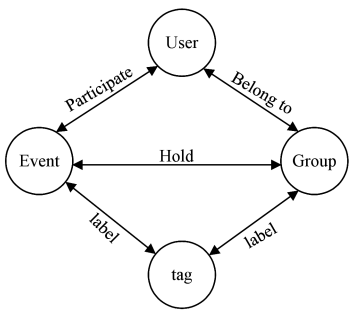

图 6 Meetup 网络模式

4.2 对比算法和评价标准

我们选取以下几种推荐算法作为对比算法。

UserMean：使用用户评分的平均值作为预测值。

ItemMean：使用物品评分的平均值作为预测值。

NMF：非负矩阵分解算法。

Trust-MF：Trust-MF[7]利用了用户之间的社交关系来提高推荐效果。

Hete-MF：Hete-MF[19]利用物品之间的关系来提高推荐效果。

我们使用了 4 种评价指标来评估不同算法的性能，分别是 MAE(平均绝对误差)、RMSE(均方根误差)、Precision(精确度)、Recall(召回率)。4 种指标比较全面地比较了算法之间的效果差异。

4.3　实验方案

在 DBLP 数据集中,我们的任务是向作者推荐其可能感兴趣的会议;在 Meetup 数据集中,我们的任务是向用户推荐其可能感兴趣的小组。对于 DBLP 和 Meetup 数据集,我们随机选择 60%的数据集作为训练数据,40%的数据集作为测试数据,使用 4.2 节提到的评价指标来衡量 MetaStruct-CF 算法和对比算法的推荐效果,每个实验结果是 5 次实验结果的平均值。

对于 MetaStruct-CF 算法,元结构的选择对算法的效果有着较大的影响。对于元结构的选择,我们考虑两个方面:一方面,在元结构中节点类型的数量应该尽可能的多,以使元结构可以包含更多需要的语义信息;另一方面,我们应该选择尽可能少的元结构类别,从而节省计算时间。现实网络中元结构的种类可能较多,在选择元结构时不可能选取所有的元结构类型,因此,对于 DBLP 数据集,我们根据以上思路选择 3 个具有代表性的元结构,如图 4 和图 5 所示。对于 Meetup 数据集,我们选取的元结构如图 7 所示。

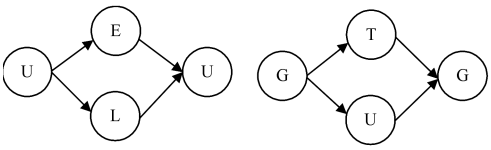

图 7　Meetup 元结构

4.4　实验结果与分析

实验结果如表 1 和表 2 所列。表 1 列出了 DBLP 中的算法性能,表 2 列出了 Meetup 中的算法性能。在 DBLP 数据集中,我们设置特征向量的维数 $d=5$。因为 DBLP 数据集包含 5 个数据区域。在 Meetup 数据集中,我们设置 $d=20$。因为在 Meetup 中,用户有更广泛的兴趣。从结果可以看出,我们的方法在大多数评估指标下胜过其他算法。Hete-MF,Trust-MF 和 MetaStruct-CF 的表现优于 NMF。因为相比 NMF,这些算法引入了更多的信息来提高算法的效果。Trust-MF 引入了用户之间的关系信息,提高了算法的推荐效果。Hete-MF 使用了物品之间的关系信息,同样也使算法的效果得到了较高的提升。这表明充分利用网络中用户之间或商品之间的信息对推荐系统有较大帮助。更重要的是,MetaStruct-CF 在这些指标下胜过 Hete-MF 和 Trust-MF。因为 MetaStuct-CF 在进行推荐的过程中引入了更加丰富的信息,进一步证明了充分利用用户之间和商品之间的关系信息,有助于提高推荐效果,同时也证明了所提算法的有效性。

另外,我们研究了 BSCSE 公式中 θ 对算法结果的影响。当 $\theta=1$ 时,算法 BSCSE 等价于 SCSE,当 $\theta=0$ 时,算法 BSCSE 等价于 StructCount。如图 8 所示,随着参数 θ 的增加,算法的性能得到有效提高,然后趋于稳定。因此,在 θ 较大时,SCSE 的效果与 BSCSE 的效果相似。因此,可以看出在本算法中 SCSE 和 BSCSE 的效果比 StructCount 的效果更好。原因是,StructCount 采用两个节点之间元结构的实例的个数来衡量相似性。SCSE 表示的是从一个节点出发,沿着元结构到达另一个节点的概率。在网络中,单纯的节点间的链接个数并不能很好地衡量节点的相似度。

另外,我们研究了 BSCSE 公式中 θ 对算法结果的影响。当 $\theta=1$ 时,算法 BSCSE 等价于 SCSE,当 $\theta=0$ 时,算法 BSCSE 等价于 StructCount。如图 8 所示,随着参数 θ 的增加,算法的性能得到有效提高,然后趋于稳定。因此,在 θ 较大时,SCSE 的效果与 BSCSE 的效果相似。因此,可以看出在本算法中 SCSE 和 BSCSE 的效果比 StructCount 的效果更好。原因是,StructCount 采用两个节点之间元结构的实例的个数来衡量相似性。SCSE 表示的是从一个节点出发,沿着元结构到达另一个节点的概率。在网络中,单纯的节点间的链接个数并不能很好地衡量节点的相似度。

表 1　算法在 DBLP 数据集上的结果

算法	UserMean	ItemMean	NMF	Trust-MF	Hete-MF	MetaStruct-CF
MAE	0.942	1.075	2.156	0.841	0.933	**0.841**
RMSE	1.216	1.123	2.395	1.003	1.100	**1.001**
Precision@3	0.191	0.185	0.233	0.313	0.317	**0.320**
Precision@5	0.152	0.164	0.230	0.289	0.301	**0.321**
Recall@3	0.090	0.087	0.215	0.289	0.293	**0.304**
Recall@5	0.102	0.110	0.353	0.444	0.463	**0.493**

表 2　算法在 Meetup 数据集上的结果

算法	UserMean	ItemMean	NMF	Trust-MF	Hete-MF	MetaStruct-CF
MAE	1.076	0.953	1.135	0.889	1.002	**0.853**
RMSE	1.154	1.143	1.356	1.025	1.105	**0.984**
Precision@3	0.101	0.122	0.216	0.255	**0.332**	**0.330**
Precision@5	0.090	0.114	0.201	0.236	0.301	**0.317**
Recall@3	0.040	0.067	0.210	0.224	0.308	**0.314**
Recall@5	0.124	0.112	0.338	0.341	0.368	**0.384**

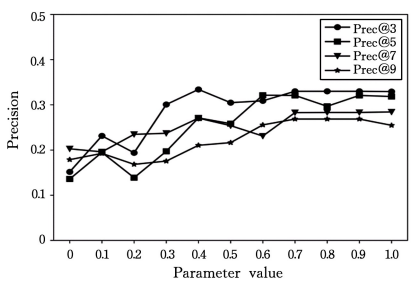

图 8　θ 对结果的影响

结束语　本文介绍了基于异构信息网络的推荐系统的发展,并提出了 MetaStruct-CF,它是第一个在异构信息网络中应用元结构来提高推荐性能的算法。在 MetaStruct-CF 中,我们使用 BSCSE 来计算数据对象之间的相似度,并分析在不同参数下 BSCSE 对结果的影响。我们通过元结构将用户之间的关系信息和物品之间的关系信息统一到我们的模型中。最后,通过在两个数据集上的实验证明了 MetaStruct-CF 算法的有效性。

(下转第 416 页)