

ISSN 2096-742X

CN 10-1649/TP



文献 CSTR:

32002.14.jfdc.

CN10-1649/

TP.2024.05.010

文献 DOI:

10.11871/jfdc.issn.

2096-742X.2024.

05.010

页码: 102-110

获取全文



## 基于图划分的分布式推荐系统

杨锦光<sup>1</sup>, 熊菲<sup>1\*</sup>, 顾峻瑜<sup>2,3</sup>, 席炜亭<sup>4</sup>

1. 北京交通大学, 北京 100044

2. 中国科学院计算机网络信息中心, 北京 100083

3. 中国科学院大学, 北京 100049

4. 华北电力大学, 北京 100096

**摘要:**【目的】设计一个数据处理效率高的推荐系统具有重要的意义。【方法】使用图结构来模拟推荐系统中的用户偏好关系, 将其通过图划分算法处理, 可以更深层次地挖掘推荐系统中数据的信息价值, 并将得到的负载均衡的子图数据作为分布式系统的输入, 最终经过一个自适应聚合模块的融合实现了一个分布式推荐系统。【结果】该系统可以提高推荐算法对于大规模数据的处理效率, 在预测精度不下降的前提下, 算法在一个由 16 个 CPU 构成的集群训练相比于单个 CPU 训练可提高 6.4 倍的效率。【结论】实验结果证明了该系统于推荐效率方面的有效性。

**关键词:** 推荐系统; 图划分; 负载均衡; 分布式系统

## A Distributed Recommender System Based on Graph Partition

YANG Jingguang<sup>1</sup>, XIONG Fei<sup>1\*</sup>, GU Junyu<sup>2,3</sup>, XI Weiting<sup>4</sup>

1. Beijing Jiaotong University, Beijing 100044, China

2. Computer Network Information Center, Chinese Academy of Sciences, Beijing 100083, China

3. University of Chinese Academy of Sciences, Beijing 100049, China

4. North China Electric Power University, Beijing 100096, China

**Abstract:** [Objective] It is of great significance to design a recommender system with high data processing efficiency. [Methods] The graph structure is used to simulate the user preference relationship in the recommender system. Through the graph partition algorithm processing, the information value of the data in the recommender system can be further mined, and the obtained sub-graph data with load balancing can be used as the input of the distributed system. Finally, a distributed recommender system is implemented through the fusion of an adaptive aggregation module. [Results] The system can improve the processing efficiency of the recommender algorithm for large-scale data. On the premise that the prediction accuracy does not decline, the algorithm can improve the efficiency 6.4 times in a cluster training consisting of 16 CPUs com-

**基金项目:** 国家自然科学基金(61872033); 国家自然科学基金(72004009); 国家重点研发计划(2018YFC0832304); 北京市科技新星计划(Z201100006820015)

**\*通信作者:** 熊菲 (E-mail: xiongf@bjtu.edu.cn)

pared with a single CPU training. [Conclusions] The experimental results show that the system is effective in recommendation efficiency.

**Keywords:** recommender system; graph partition; load balancing; distributed system

## 引言

在互联网高速发展的年代,搜索引擎已经不能完全满足用户的需求<sup>[1]</sup>。因而,推荐系统应运而生,推荐系统是一种为用户在互联网时代过滤冗余信息、提供个性化服务,用以缓解信息过载的手段<sup>[2]</sup>。推荐系统因其有效性和实用性得到了广泛的关注,在电子商务<sup>[3]</sup>、社交媒体<sup>[4]</sup>、新闻门户<sup>[5]</sup>等广泛的应用中发挥着越来越重要的作用。美国知名电影公司网飞公司估计,其个性化推荐系统每年可能为其业务节省10亿美元。亚马逊网络书城的推荐算法每年贡献30个百分点的创收。从本质上来讲,推荐系统是对用户某种行为的模拟,其根据用户的历史购买需求、行为记录或者相似偏好进行建模,从而发现符合用户偏好的需求,再为之推荐给用户。

迄今为止,科研人员已经提出了大量的推荐算法。协同过滤(Collaborative Filtering, CF)<sup>[6]</sup>是推荐系统中使用广泛的算法之一。其可分为基于内存的算法和基于模型的算法。矩阵分解(Matrix Factorization, MF)<sup>[7]</sup>是一种经典的基于模型的协同过滤算法,它将一个高秩矩阵分解为若干个低秩矩阵。其主要思想是通过矩阵计算的手段来降低计算复杂度,同时提高预测精度。由于矩阵计算的成熟发展,基于矩阵分解的推荐算法得到了蓬勃的扩展,例如 LightGCN<sup>[8]</sup>、MA-GNN<sup>[9]</sup>、卷积矩阵分解<sup>[10]</sup>、贝叶斯个性化排名(Bayesian Personalized Ranking, BPR)<sup>[11]</sup>等。但矩阵分解也存在训练时间长及难以处理大规模数据的问题。虽然隐语义模型<sup>[12]</sup>相比一般的矩阵分解将分解得到的矩阵个数减少,降低了一定的复杂度,然而在实际的推荐过程中,不可避免的是数据的规模通常很大,导致分解得到的矩阵规模依旧很大,而且是稠密矩阵,导致产生较高的成本。另一方面,图学习(Graph Learning, GL)涉及应

用于图形结构数据的机器学习,是一种新兴的人工智能技术,近年来显示出巨大的能力。一种新兴的基于图学习的推荐范式,即基于图学习的推荐系统(Graph Learning based Recommender Systems, GLRS)<sup>[13]</sup>,用来学习由图建模的特定类型的关系,并已被证明是有效的<sup>[14]</sup>。因此,使用图结构来模拟推荐系统中的各种关系是一种自然和令人信服的选择。然而数据规模大同样往往制约着它的发展,因为图结构数据相比一般数据通常更大,需要更多的时间和空间。因此,在深度学习的背景下,对图结构数据执行复杂的机器学习技术具有挑战性。

为了提高推荐系统对大规模数据的处理和学习能力,及时为用户生成推荐列表,本文结合了与图结构数据这类非欧几里得结构数据密切相关的图划分算法缓解推荐系统在面临大规模数据所处的困境。图划分问题是经典的 NP (Non-deterministic Polynomial) 完全问题,通常很难在有限的时间内找到图划分的最优解。虽然在图划分问题上大都需要较多的内存和存在着较高的计算复杂度的问题,但图结构数据在现实世界中具有重要的价值和意义。基于此,国内外研究者提出了很多优异可靠的算法,如几何方法、启发式方法、智能优化算法、多层划分算法等。图划分的目标是将输入的图结构数据分割为若干个规模相当的子图,并且跨越子图有最少的边数。

在实际的推荐过程中,往往存在大规模数据,且带来较长的学习和处理时间,难以及时地为用户生成可靠的推荐列表。基于上述问题,本文提出了基于图划分的分布式推荐系统。首先,利用比一般的欧几里得结构数据更具信息价值的图结构数据模拟推荐系统中的关系,可以更好地捕捉用户的局部特征,有助于提高推荐性能。接着,结合图划分算法将输入的图数据划分为负

载均衡的子图数据,可以进一步提高分布式系统的效率。最后,设计一种基于注意力的自适应聚合模块融合学习到的特征表示,以便进一步学习分布式系统的子图信息。

## 1 相关工作

### 1.1 Top-N推荐算法

在推荐系统中,Top-N推荐的目的是从大量的数据中识别出用户可能偏爱的一小部分数据,结果以推荐列表的形式给用户推送信息。

贝叶斯个性化排序,是当下推荐系统中Top-N推荐常用的一种推荐算法。与其他的基于用户评分矩阵的方法不同的是贝叶斯个性化排序算法主要采用用户的隐式反馈(如点击、收藏、加入购物车等),通过对问题进行贝叶斯分析得到的最大后验概率来对项目进行排序,进而产生推荐。

在该算法中,将任意用户 $u$ 对应的项目进行标记,如果用户 $u$ 在同时有项目 $i$ 和 $j$ 的时候对 $i$ 产生了行为(浏览或购买等),那么就得到了一个三元组 $\langle u, i, j \rangle$ ,表示对用户 $u$ 来说,项目 $i$ 的排序要比项目 $j$ 靠前。但如果一个用户对两个项目同时产生过行为,或同时没有产生行为,则无法构成偏好对。如果对于用户 $u$ 来说有 $m$ 组的反馈,那么就可以得到 $m$ 组用户 $u$ 对应的训练样本。

### 1.2 图划分

图划分的目标是将输入的图结构数据分割为若干个规模相当的子图,并且跨越子图有最少的边数。按照对图数据的切分方式分类,可以分为点划分和边划分。

本文选择一种高质量和轻量级的最小切割图划分算法:Metis<sup>[15]</sup>。从本质上讲,Metis是一种点划分算法,是多层划分算法的一种。点划分是将图的节点分配到各个子图中,维持节点之间子图的完整性,此时可能造成某些节点之间的边被切掉。Metis的核心思想是对于给定图结构数据

进行多层次的稀疏化融合节点和边来降低原图的规模大小,然后达到一定程度后对于缩减后的图结构数据进行划分,最后将划分后的子图还原成原始的图结构数据,保证每份子图的数据均衡。

## 2 模型概述

基于图划分的分布式推荐系统由3部分组成。如图1所示。

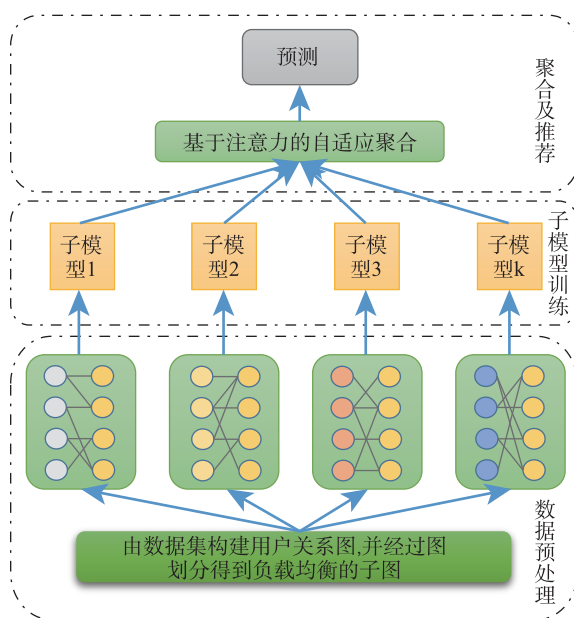


图1 基于图划分的分布式推荐系统

Fig.1 Distributed recommender system based on graph partition

(1)数据预处理:由原始数据构成用户关系图,利用图划分算法对其进行分区处理。

(2)子模型训练:选取子模型训练原图划分之后的子图数据,获取用户特征表示和项目特征表示。

(3)聚合及推荐:使用基于注意力的自适应模块分别对由各个子模型训练得到的用户特征和项目特征进行聚合,随后得到推荐结果。

表1描述了本文中使用的符号和概念描述。使用 $U$ 和 $V$ 分别表示用户集合和项目集合,两者

表1 符号说明

Table 1 Symbol description

符号	描述
$U$	用户集合
$V$	项目集合
$Y$	用户-项目交互
$N$	分区数量
$P^i$	第 <i>i</i> 个分区的用户嵌入表示
$Q^i$	第 <i>i</i> 个分区的项目嵌入表示
$P'_i$	转移到相同特征空间上第 <i>i</i> 个分区的用户嵌入表示
$Q'_i$	转移到相同特征空间上第 <i>i</i> 个分区的项目嵌入表示
$P$	聚合后的用户嵌入表示
$Q$	聚合后的项目嵌入表示

之间的交互使用 $Y$ 进行表示。推荐任务是给定一个用户 $u$ ,推荐一个用户 $u$ 感兴趣的项目列表。

## 2.1 数据预处理

原始数据集是由各个类似三元组[用户,项目,评分]构成的,在进行训练前先将其转换为模型所需的格式。

首先,构建用户相似度图:使用余弦相似度计算用户之间的相似度。给定用户 $u$ 和用户 $v$ ,令 $N_u$ 表示用户 $u$ 曾经有交互的物品集合,令 $N_v$ 表示用户 $v$ 曾经有交互的物品集合。其余弦相似度如公式(1):

$$w_{uv} = \frac{|N_u \cap N_v|}{\sqrt{|N_u| |N_v|}} \quad (1)$$

通过计算就可以得到每个用户与其他用户的相似度,这里选取相似度由高到低排列的前10个用户作为相似用户。经过如上计算得到一个原始用户相似图。

接下来使用图划分算法将原始用户相似图进行划分。如图2所示,将包含10个用户的用户图划分为2个子图,使得子图之间的连边最少。也就是说子图内部的用户关系链接度较高,子图与子图之间的关系链接度低。

图2所示的原始用户关系图可将其解释为:在推荐过程中,左边5个用户之间兴趣爱好较为一致,右边5个用户兴趣爱好较为一致,两个用

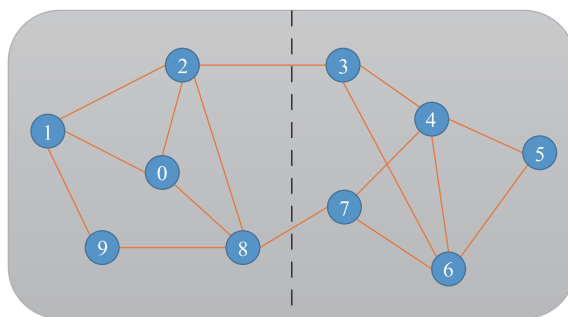


图2 使用图划分将10个用户划分为2个子图

Fig.2 Divide 10 users into 2 subgraphs using graph partition

户簇之间的兴趣爱好差异较大。在训练过程中,由于用户之间的偏好力度的不一致,例如用户A点击了5个项目,用户B点击了60个项目,导致出现负载不均衡的现象。为了更好地满足分布式系统对于负载均衡的要求,有必要将用户的偏好力度进行量化,将用户-项目交互数映射到一个等差区间内,将每个区间视为一个偏好力度区间,将其所对应的值作为其权重,结合Metis中的顶点带权重的划分方式可以解决训练时数据负载不均衡的问题。

如图3所示,将一个包含16个用户的关系图使用Metis算法将其划分为4个子图,每个子图

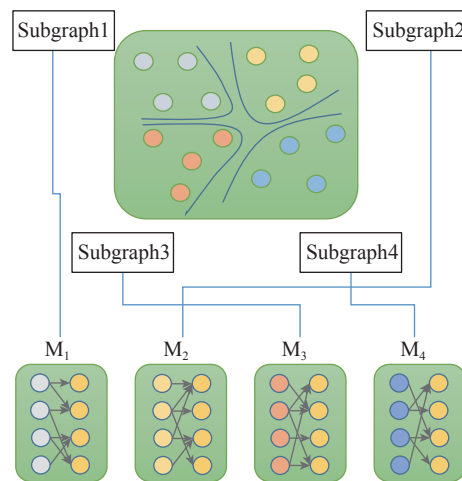


图3 4个子图输入到对应的子模型

Fig.3 Input the four subgraphs into the corresponding submodel



的用户数为4。之后将对应的子图作为对应的子模型的输入。

## 2.2 子模型训练

对数据进行划分后,在每个子模型中训练一个子图数据。子模型的选取可以选择不同的推荐系统算法模型,这里选取的是BPR算法获取用户和项目在特征空间的表示。BPR算法的核心思想是将原始数据集转化为三元组对地进行训练。

一般来说,不同用户对不同项目有不一样程度的偏好。考虑到不同子模型学习到的用户和项目的表示可能会嵌入到不同的空间中,使用公式(2)和(3)将它们转移到相同的表示空间中。

$$P_t^i = W^i P^i + b^i \quad (2)$$

$P^i$ 是各个子模型训练得到的用户嵌入表示; $W^i$ 是对应的转移矩阵; $b^i$ 是偏差矩阵。

$$Q_t^i = W^i Q^i + b^i \quad (3)$$

$Q^i$ 是各个子模型训练得到的项目嵌入表示; $W^i$ 是对应的转移矩阵; $b^i$ 是偏差矩阵。

## 2.3 聚合与推荐

为了进一步提高模型的推荐性能,设计了一个基于注意力的自适应聚合模块。使用公式(4)和(5)分别计算用户和商品的聚合嵌入表示:

$$P = \sum_{i=1}^N \alpha_i P_t^i \quad (4)$$

$$Q = \sum_{i=1}^N \beta_i Q_t^i \quad (5)$$

其中 $\alpha_i$ 和 $\beta_i$ 是聚合用户和项目嵌入表示的注意力权重,定义为公式(6)~(9):

$$\alpha_i^* = h_1^T W_1 P_t^i \quad (6)$$

$$\alpha_i = \frac{\exp(\alpha_i^*)}{\sum_{j=1}^N \exp(\alpha_j^*)} \quad (7)$$

$$\beta_i^* = h_2^T W_2 Q_t^i \quad (8)$$

$$\beta_i = \frac{\exp(\beta_i^*)}{\sum_{j=1}^N \exp(\beta_j^*)} \quad (9)$$

其中, $h_1$ 和 $W_1$ 是用户注意力的参数; $h_2$ 和 $W_2$ 是项

目注意力的参数;所有分区上的注意权重都通过softmax函数进行了归一化处理。

该模块训练采用BPR配对损失函数去训练,如公式(10)所示。

$$\min_{\theta} \zeta(P, Q, Y) + \lambda \|\theta\|_2^2 \quad (10)$$

其中, $\zeta$ 表示BPR配对损失函数; $P$ 是用户的聚合嵌入表示; $Q$ 是项目的聚合嵌入表示; $Y$ 是用户-项目之间的交互; $\theta$ 是注意力模块的参数,加入带系数 $\lambda$ 的L2正则化防止过拟合。

## 3 实验

实验所使用单机环境配置如下:(处理器:1颗32核心/2.5 GHz主频/X86架构国产处理器);内存(128 GB)。实验所使用分布式环境配置如下:计算节点数量规模为{2,4,8,16};处理器(1颗32核心/2.5 GHz主频/X86架构国产处理器);内存(128 GB)。

整个模型的训练是一个两步训练过程。第一步是使用分布式环境训练子模型;第二步是在一个计算节点内训练基于注意力的自适应聚合模块。

### 3.1 数据集和评估指标

在两个真实世界和公开的数据集上进行了实验,这些数据集已被广泛应用于评估推荐系统模型:MovieLens-1 m、MovieLens-10 m<sup>[16]</sup>。这些数据集都是电影数据集,包括用户对电影的5级评分。所有数据集的格式都形如 $[u, v, r_{uv}]$ 。由于是Top-N推荐任务,因此遵循广泛的数据处理方法,将这些数据集转化为隐式反馈。详细地讲,是将数据集中的具体评分转换为0或1来表示用户与项目之间是否有交互。表2总结了这些数据集的统计细节。

MovieLens-1 m数据集共包含6,940个用户和3,706个项目,用户和项目之间的交互数量为1,000,209,交互矩阵的密度为3.89%。而MovieLens-10 m数据集共包含71,567个用户和10,681

表2 数据集的统计细节

Table 2 Statistical details of datasets

数据集	用户	项目	用户-项目交互	密度
Movielens-1 m	6,940	3,706	1,000,209	3.89%
Movielens-10 m	71,567	10,681	10,000,054	1.31%

个项目,用户和项目之间的交互数量为10,000,054,交互矩阵的密度为1.31%。

对于每个数据集,随机选择80%的数据作为训练集,剩下20%的数据作为测试集。使用五折交叉验证法来进行训练和测试。具体来说,将数据随机分成5份,并重复测试5次。在每次实验中,使用4份作为训练集,剩下的1份作为测试集。然后,计算在测试集上的平均结果,以确定最终的结果。选择2个经典的和常见的指标,即召回率 recall 公式(11)和精度 precision 公式(12)。一般来说,这2个评估指标的值越高,意味着推荐性能越好。

$$\text{recall}(L) = \frac{|R_u \cap T_u|}{|T_u|} \quad (11)$$

$$\text{precision}(L) = \frac{|R_u \cap T_u|}{|R_u|} \quad (12)$$

其中, $L$ 是推荐列表的长度; $R_u$ 是根据训练集给出的推荐列表; $T_u$ 是训练集中的行为列表(真实数据)。

### 3.2 参数设置

在第一步训练过程中,批量大小设置为512,学习速率设置为0.05,特征嵌入维度设置为64,对数据进行划分的数量为{2,4,8,16},与之对应的分布式环境规模大小也为{2,4,8,16};在第二步训练过程中,批量大小设置为1,024,学习速率设置为0.05,特征嵌入维度设置为32。最后的结果采用内积形式的评分函数完成。推荐列表 $L$ 的长度取10和20。为了优化目标函数,使用小批量的Adagrad<sup>[17]</sup>作为优化器。其主要优点是在模型学习过程中学习速率可以自适应地调整。

### 3.3 结果与分析

表3和表4分别是数据集Movielens-1 m和Movielens-10 m使用图划分算法生成不同分区的子图的推荐性能。具体设置为数据分区个数 $N=2,4,8,16$ 和推荐列表长度 $L=10,20$ , $N=1$ 表示不划分数据或者划分结果为1个子图。由表3和表4可以看出 $N=2,4,8,16$ 时,recall和precision的值几乎都优于 $N=1$ 时的值,除了 $N=16$ 时数据集Movielens-10 m的recall@10和precision@10略低。实验结果显著的表现可以归结为两方面的原因:(1)使用图结构模拟推荐系统中的各种关系,例如用户偏好相似关系,进而使用图划分算法划分数据,通过保留数据中用户之间的偏好相似的信息提高推荐性能;(2)使用一个基于注意力的自适应聚合模块进一步学习各个子模型学习到的特征表示,考虑不同的子模型的贡献不同,进一步提升整体的性能。

表3 数据集Movielens-1 m不同子图数量的推荐性能

Table 3 Recommended performance of dataset Movielens-1 m with different subgraph numbers

分区数 $N$	1	2	4	8	16
recall@10	0.0693	0.0740	0.0757	0.0723	0.0714
recall@20	0.1135	0.1178	0.1282	0.1265	0.1250
precision@10	0.1681	0.1763	0.1803	0.1776	0.1749
precision@20	0.1422	0.1457	0.1550	0.1501	0.1492

表4 数据集Movielens-10 m不同子图数量的推荐性能

Table 4 Recommended performance of dataset Movielens-10 m with different subgraph numbers

分区数 $N$	1	2	4	8	16
recall@10	0.1050	0.1101	0.1180	0.1206	0.1029
recall@20	0.1684	0.1835	0.1899	0.1940	0.1814
precision@10	0.1715	0.1738	0.1826	0.1845	0.1701
precision@20	0.1437	0.1503	0.1550	0.1563	0.1445

图4-图7是数据集使用随机划分和图划分的两种划分方式的性能比较。随机划分是指将原始数据随机打乱,使之平均分配为若干份数据。图4和图5是数据集Movielens-1 m使用两

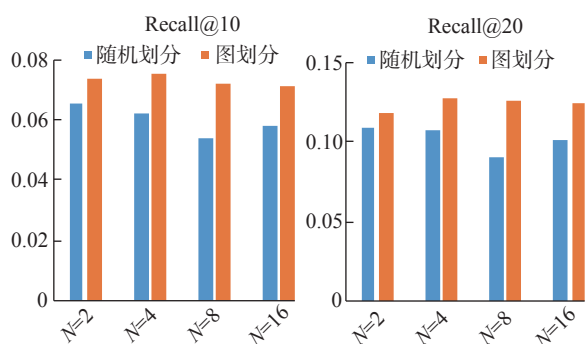


图4 Movielens-1 m使用两种不同的划分方法 recall值比较

Fig.4 Comparison of recall values of Movielens-1 m using two different division methods

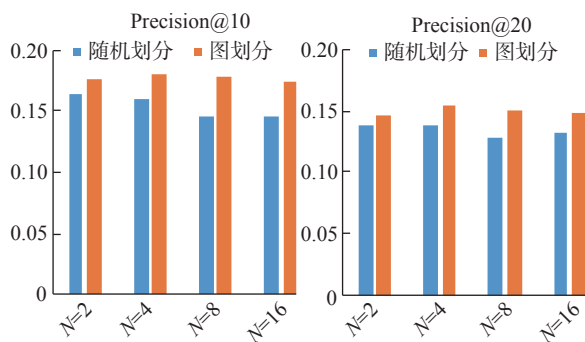


图5 Movielens-1 m使用两种不同的划分方法 precision值比较

Fig.5 Comparison of precision values of Movielens-1 m using two different division methods

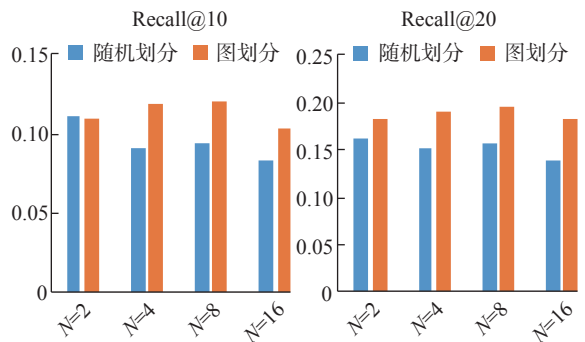


图6 Movielens-10 m使用两种不同的划分方法 recall值比较

Fig.6 Comparison of recall values of Movielens-10 m using two different division methods

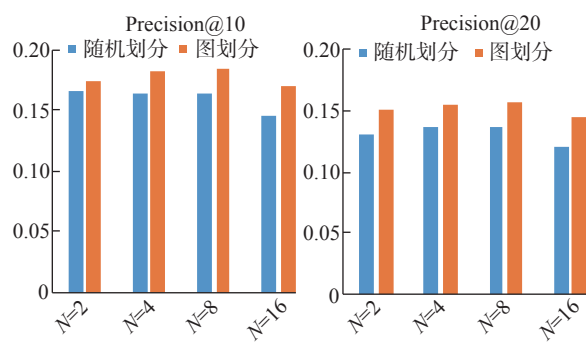


图7 Movielens-10 m使用两种不同的划分方法 precision值比较

Fig.7 Comparison of precision values of Movielens-10 m using two different division methods

种不同的划分方法的性能比较;图6和图7是数据集 Movielens-10 m使用两种不同的划分方法的性能比较。由图4~7对比得到:使用图划分的方法其性能结果基本上都优于使用随机划分的方法,进一步表明使用图结构数据结合图划分的方法对于提升推荐性能起着关键作用。使用图划分的方法对于性能的提升有着一定的上限,由图4和图5可以得到 Movielens-1 m在  $N=4$  时性能优于其他分区,由图6和图7可以得到 Movielens-10 m在  $N=8$  时的性能优于其他分区。一个分区数量的确定,意味着分区内部的数据规模的确定,不同的数据集往往存在着一个适宜的数据规模,有助于提升推荐性能,表明分区数量的增加与性能的提升并不存在着正相关的关系。

表5和表6是数据集运行时间的体现。整体的运行时间由子模型训练和聚合训练两部分构成。计算得到使用分布式环境可以提升训练效率: Movielens-1 m加速比为2.2~6.4, Movielens-10 m加速比为1.6~4.7。本文采用的两步训练方式在计算节点较少时子模型训练时间相比较长,在计算节点较多时聚合训练时间占比较大。然而随着分布式环境中计算节点的增多,整体运行时间的加速比会下降(与理想线性加速比相比),这是因为一个确定的数据集其聚合训练的数据量是确定不变的,也就是说聚合训练时间是确定的,其时间占比随之升高,因此加速比呈现出逐

表5 Movielens-1 m不同分区数量运行时间(分钟)

Table 5 Running time of Movielens-1 m different partitions (minute)					
	1	2	4	8	16
子模型训练	—	28.3	15	7.8	4.2
聚合训练	—	8.3	8.3	8.3	8.3
总计	80	36.6	23.3	16.1	12.5

表6 Movielens-10 m不同分区数量运行时间(分钟)

Table 6 Running time of Movielens-10 m different partitions (minute)					
	1	2	4	8	16
子模型训练	—	400	216.7	125	75
聚合训练	—	83.3	83.3	83.3	83.3
总计	750	483.3	300	208.3	158.3

渐下降的趋势。结合前文不同分区数量的性能比较,可以得到:一个适宜的由图划分算法生成的数据分区个数对推荐性能和推荐效率上有着良好的帮助。

#### 4 结论与展望

本文设计了一种基于图划分的推荐系统。其使用图结构和图划分模拟推荐系统中的用户偏好相似关系,可以更好地学习用户的特征表示。将得到的负载均衡的数据应用于分布式系统中,可以进一步保证推荐效率,随之构建的基于注意力的自适应聚合模块可以更好地融合模型学习到的特征表示,有助于保证推荐性能。模型考虑了推荐性能和推荐效率两者之间的兼容。实验结果表明,本文提出的模型在不降低性能的前提下可以提高大规模数据的处理效率。未来将探讨利用推荐系统的数据特点与多重划分方式的结合,进一步扩展到不仅限于二部图这一种异构图上。

#### 利益冲突声明

所有作者声明不存在利益冲突关系。

#### 参考文献

- [1] GRECHANIK M, FU C, XIE Q, et al. A search engine for finding highly relevant applications[C]//Acm/ieee International Conference on Software Engineering. ACM, 2010: 475-484.
- [2] JIE L, DIANSHUANG W, MINGSONG M Z, et al. Recommender system application developments: A survey[J]. Decision Support Systems, 2015,74: 12-32.
- [3] FUYU L, TAIWEI J, CHANGLONG Y, et al. SDM: Sequential Deep Matching Model for Online Large-scale Recommender System[C]//Conference on Information and Knowledge Management, 2019: 2635-2643.
- [4] COVINGTON P, ADAMS J, SARGIN E. Deep Neural Networks for YouTube Recommendations[C]//Acm Conference on Recommender Systems. ACM, 2016: 191-198.
- [5] SHUMPEI O, YUKIHIRO T, SHINGO O, et al. Embedding-based News Recommendation for Millions of Users[C]// Knowledge Discovery and Data Mining, 2017: 1933-1942.
- [6] MCLAUGHLIN M R, HERLOCKER J L. A collaborative filtering algorithm and evaluation metric that accurately model the user experience[C]//International Acm Sigir Conference on Research & Development in Information Retrieval. ACM, 2004: 329-336.
- [7] KOREN Y, BELL R, VOLINSKY C. Matrix Factorization Techniques for Recommender Systems[J]. IEEE, 2009, 42(8): 30-37.
- [8] HE X, DENG K, WANG X, et al. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation[C]//International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020: 639-648.
- [9] CHEN M, LIHENG M, YINGXUE Z, et al. Memory Augmented Graph Neural Networks For Sequential Recommendation[C]//National Conference on Artificial Intelligence, 2020, 34: 5045-5052.
- [10] DONG H K, CHANYOUNG P, JINOH O, et al. Convolutional Matrix Factorization for Document Context-Aware Recommendation[C]//Conference on Recommender Systems, 2016: 233-240.
- [11] STEFFEN R, CHRISTOPH F, ZENO G, et al. BPR:



- Bayesian personalized ranking from implicit feedback [C]//Uncertainty in Artificial Intelligence, 2012: 452-461.
- [12] ZHANG W, WANG J Y, FENG W. Combining latent factor model with location features for event-based group recommendation[M]. Knowledge Discovery and Data Mining, 2013: 910-918.
- [13] GUO Q Y, ZHUANG F Z, QIN C, et al. A Survey on Knowledge Graph-Based Recommender Systems[J]. IEEE Annals of the History of Computing, 2022, 34 (8): 3549-3568.
- [14] WU Z H, PAN S R, CHEN F W, et al. A Comprehensive Survey on Graph Neural Networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(1): 4-24.
- [15] GEORGE K, VIPIN K. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs[J]. SIAM Journal on Scientific Computing, 1999, 20(1): 359-392.
- [16] MAXWELL H F, JOSEPH A K. The MovieLens Datasets: History and Context[J]. ACM transactions on interactive intelligent systems, 2016, 5(4): 1-19.
- [17] JOHN C D, ELAD H, YORAM S. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization[J]. Journal of Machine Learning Research, 2011, 12(61): 2121-2159.

收稿日期: 2023 年 1 月 15 日

杨锦光, 北京交通大学, 硕士研究生, 主要研究方向为人工智能、推荐系统。

本文承担工作为: 模型设计, 模型算法实现。

YANG Jinguang is a master's student at Beijing Jiaotong University.

His main research interests are artificial intelligence and recommender systems.

In this paper, he is mainly responsible for model design and model algorithm realization.

E-mail: yangjg@bjtu.edu.cn



熊菲, 北京交通大学, 博士生导师, 主要研究方向为人工智能、网络内容安全、推荐系统等。

本文承担工作为: 指导优化模型和模型设计。

XIONG Fei is a Ph.D. supervisor at Beijing Jiaotong University. His

main research interests are artificial intelligence, network content security, and recommender systems.

In this paper, he is mainly responsible for providing guidance for optimizing and designing models.

E-mail: xiongf@bjtu.edu.cn



引文格式: 杨锦光, 熊菲, 顾峻瑜, 席炜亭. 基于图划分的分布式推荐系统[J]. 数据与计算发展前沿, 2024, 6(5): 102-110. DOI: 10.11871/jfdc.issn.2096-742X.2024.05.010. <https://cstr.cn/32002.14.jfdc.CN10-1649/TP.2024.05.010>.

YANG Jinguang, XIONG Fei, GU Junyu, XI Weiting. A Distributed Recommender System Based on Graph Partition[J]. Frontiers of Data & Computing, 2024, 6(5): 102-110. DOI: 10.11871/jfdc.issn.2096-742X.2024.05.010. <https://cstr.cn/32002.14.jfdc.CN10-1649/TP.2024.05.010>.