

文章编号:1007-5321(2021)03-0021-06

DOI:10.13190/j.jbupt.2020-226

# 卷积记忆图协同过滤

刘国桢, 陈鸿龙

(中国石油大学(华东) 控制科学与工程学院, 青岛 266580)

**摘要:** 针对推荐系统中用户和项目的向量表示问题,提出了一种端到端的具有记忆单元的图神经网络. 在图神经网络中引入门控循环单元解决高阶连通节点间信息损失问题,可以使得用户和项目节点从高阶邻居获得更加完整的特征信息,然后利用卷积神经网络对网络输出层间的特征向量进行融合以获得不同阶段下用户的偏好. 实验结果表明,与最优对比算法相比,采用所提卷积记忆图协同过滤推荐算法在4个数据集上的评分预测性能分别提升了1.98%,4.17%,9.27%和2.70%.

**关键词:** 图神经网络; 门控循环单元; 卷积神经网络; 评分预测; 推荐系统

中图分类号: TP301

文献标志码: A

## Convolutional Memory Graph Collaborative Filtering

LIU Guo-zhen, CHEN Hong-long

(College of Control Science and Engineering, China University of Petroleum(East China), Qingdao 266580, China)

**Abstract:** An end-to-end graph neural networks with memory unit is proposed for user vector representations and items in recommender systems. Gated recurrent unit is introduced to reduce the information loss between high-order connected nodes. This enables users and items nodes to obtain more complete feature information from high-order neighbor nodes. The convolutional neural networks are used to fuse feature vectors between different output layers to obtain users' preferences at different stages. Experiments on 4 datasets show that compared with the optimal comparison algorithms, the performance of proposed algorithm achieves gain of 1.98%, 4.17%, 9.27% and 2.7%, respectively.

**Key words:** graph neural networks; gated recurrent unit; convolutional neural networks; rating prediction; recommender systems

在互联网时代,网络数据规模呈现出爆发式的增长,使得用户正面临着信息过载的困境. 推荐系统可以帮助用户决策,很好地缓解了这一问题,因此被广泛地应用在电商平台、社交网络和短视频分享平台中.

推荐系统利用用户和项目的描述信息,和它们

之间的历史交互记录去估计用户对项目的偏好程度. 协同过滤(CF, collaborative filtering)算法是推荐系统领域的经典算法,在理论研究和实际应用中取得了巨大的成功. 矩阵分解(MF, matrix factorization)是CF算法中的一种,其利用评分矩阵将用户和项目分别映射为一个 $d$ 维向量,而后通过对用户

收稿日期: 2020-11-03

基金项目: 国家自然科学基金项目(61772551); 国家自然科学基金国际合作与交流项目(62111530052); 中石油重大科技项目(ZD2019-183-003)

作者简介: 刘国桢(1995—),男,硕士生.

通信作者: 陈鸿龙(1984—),男,副教授,博士生导师, E-mail: chenhl@upc.edu.cn.

向量与项目向量进行内积运算获得预测分数。

通常来说,设计一个推荐系统算法有2个核心问题:用户和项目的嵌入表示问题和用户项目间交互预测建模问题。CF利用用户或者项目间的相似性生成推荐<sup>[1]</sup>。因子分解机(FM, factorization machines)算法对用户(项目)的属性之间进行二阶特征交互,以生成更多特征信息,从而提高推荐效果。在神经图协同过滤(NGCF, neural graph collaborative filtering)算法中,Wang等<sup>[2]</sup>利用图神经网络(GNN, graph neural networks)获取用户和项目之间潜在的高阶协同关系,却忽视了高阶协同信息在远距离节点间传递中的衰减问题;而且,对每一层网络的嵌入向量进行简单地拼接无法获得不同层嵌入向量维度间的交互信息。

针对以上问题,提出了卷积记忆图协同过滤(CMGCF, convolutional memory graph collaborative filtering)推荐算法,通过在GNN中引入门控循环单元(GRU, gated recurrent unit)降低了节点间信息传递的损失,可以使节点嵌入向量捕获更完整的高阶邻居信息。利用卷积神经网络(CNN, convolutional neural networks)对不同层的嵌入向量进行融合,以学习不同时期下节点向量维度间的交互特征,从而使得节点向量包含更多的偏好信息。在4个数据集上进行实验,结果显示CMGCF性能优于对比算法。

## 1 相关工作

基于深度神经网络的推荐系统能够综合各种类型的多源异构数据,有效捕捉到用户的不同偏好,从而提高推荐的准确性,该系统正被广泛研究。

在用户和项目表示学习的问题上,Xue等<sup>[3]</sup>提出了深度矩阵分解(DeepMF, deep matrix factorization)算法,采用2个并行的多层感知机(MLP, multi-layer perceptron)将用户和项目映射到同一个低维空间。受到GNN的启发,Li等<sup>[4]</sup>认为用户和项目的特征字段间存在着图结构,提出了图特征交互(FiGNN, feature interactions via graph neural networks)算法,采用GNN对特征字段间的关系进行建模。

学习用户特征和项目特征之间的关系对数据建模起着重要的作用<sup>[5]</sup>。He等<sup>[6]</sup>提出了神经协同过滤(NCF, neural collaborative filtering)算法,利用MLP学习用户和项目特征间的交互关系。He等<sup>[7]</sup>提出了神经因子分解机(NFM, neural factorization machines)算法,采用MLP对FM的输出进行高阶非

线性交互,其在嵌入层之上连接一个双线性池化交互层,可以使得模型在采用较少隐藏层的情况下获得更好的性能。He等<sup>[8]</sup>提出了基于张量积的神经协同过滤(ONCF, outer product-based neural collaborative filtering)算法,其对用户和项目的嵌入向量进行张量积运算形成一个用户项目的关系矩阵,从而可以利用CNN学习用户项目嵌入向量各维度间的信息。Xin等<sup>[9]</sup>提出了卷积因子分解机(CFM, convolutional factorization machines)算法,采用张量积运算将特征向量交互成为一系列的关系矩阵,以便于更有效地捕捉特征向量各维度间的二阶关系,进一步将关系矩阵堆叠,利用3D CNN显式学习特征向量维度间的高阶交互关系。

## 2 GNN理论

GNN是一种连接模型,可以捕捉图中节点间的依赖信息<sup>[10]</sup>。目前,GNN被广泛应用在自然语言处理<sup>[11]</sup>和计算机视觉<sup>[12]</sup>等领域。GNN的目标是为节点学习一个 $d$ 维的嵌入向量 $e_n$ 。设想用户 $u$ 对项目 $i$ 有交互关系,那么在图中交互信息将被抽象化成一条连接用户与项目的边,如图1所示。

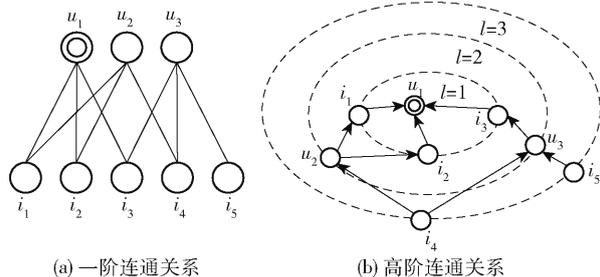


图1 用户-项目的连通关系

### 2.1 GNN中的高阶连通关系传递

将图1(a)的一阶连通关系展开成高阶连通关系,如图1(b)所示。图1(b)中高阶连通关系指的是一个节点经过 $l > 1$ 的路径长度到达目标节点 $u_1$ 。因此,通过高阶连通关系使得目标节点嵌入向量包含了丰富的语义信息。用户 $u_1$ 和项目 $i_5$ 的距离为3,其特征信息传递路径为 $i_5 \rightarrow u_3 \rightarrow i_3 \rightarrow u_1$ 。这表示 $i_5$ 作为 $u_1$ 的三阶邻居,其特征信息经过 $u_3$ 和 $i_3$ 两个节点传递到目标节点 $u_1$ 。

### 2.2 GNN中嵌入向量的信息聚合过程

对于一个包含 $N$ 个用户和 $M$ 个项目的集合,该集合的嵌入矩阵 $E^{(0)}$ 将作为CMGCF的原始输入,即:

$$\mathbf{E}^{(0)} = [\mathbf{e}_{u_1}^{(0)}, \mathbf{e}_{u_2}^{(0)}, \dots, \mathbf{e}_{u_N}^{(0)}; \mathbf{e}_{i_1}^{(0)}, \mathbf{e}_{i_2}^{(0)}, \dots, \mathbf{e}_{i_M}^{(0)}] \quad (1)$$

其中  $\mathbf{e}_n^{(l)}$  为节点  $n$  的  $l$  阶嵌入向量。

对于一组连接的用户  $u$  和项目  $i$ , 将从  $i$  流向  $u$  的信息表示为

$$T_{u \leftarrow i} = \frac{1}{|N_u|} \mathbf{e}_i \quad (2)$$

其中:  $N_u$  为用户  $u$  的一阶邻居集合,  $\frac{1}{|N_u|}$  为折损因子。随后将用户  $u$  周围节点的信息进行聚合, 完成一次用户  $u$  的嵌入向量更新:

$$\mathbf{e}_u^{(1)} = \text{LeakyReLU} \left( \sum_{i \in N_u} T_{u \leftarrow i} \right) \quad (3)$$

其中:  $\mathbf{e}_u^{(1)}$  为用户  $u$  在获得一阶邻居的信息聚合后更新的嵌入向量; LeakyReLU 为激活函数, 可同时对正负信号进行编码。重复  $l$  次一阶邻居信息聚合, 可以得到全局节点的  $l$  阶嵌入向量, 图 2 所示为用户  $u_1$  的三阶嵌入向量生成过程。

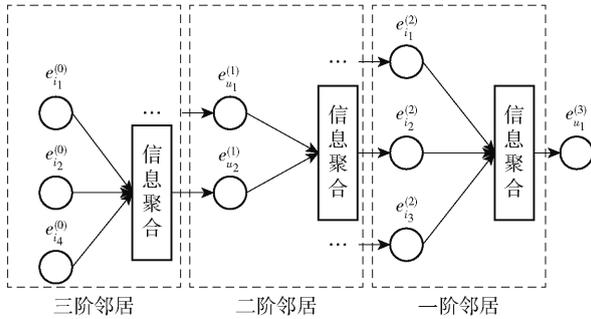


图2 三阶 GNN 的框架

## 3 CMGCF 算法与实现

### 3.1 GRU 缓解节点信息传递中的损失

GRU 利用重置门  $r_t$  控制保存的历史信息数量, 更新门  $z_t$  决定遗忘信息的数量, 并加入最新状态信息  $\tilde{\mathbf{h}}_t$ , 得到 GRU 的输出  $\mathbf{h}_t$ , 其数学表示式为

$$\mathbf{r}_t = \sigma(\mathbf{W}_r [\mathbf{h}_{t-1}, \mathbf{x}_t]) \quad (4)$$

$$\mathbf{z}_t = \sigma(\mathbf{W}_z [\mathbf{h}_{t-1}, \mathbf{x}_t]) \quad (5)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_{\tilde{\mathbf{h}}} [\mathbf{r}_t \circ \mathbf{h}_{t-1}, \mathbf{x}_t]) \quad (6)$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \circ \mathbf{h}_{t-1} + \mathbf{z}_t \circ \tilde{\mathbf{h}}_t \quad (7)$$

其中:  $\mathbf{W}_r, \mathbf{W}_z, \mathbf{W}_{\tilde{\mathbf{h}}}$  为权重矩阵,  $\mathbf{h}_{t-1}$  为上一时刻的输出状态,  $\mathbf{x}_t$  为本时刻的输入,  $[\cdot, \cdot]$  为对 2 个向量进行拼接,  $\circ$  为哈达玛积。为了表示方便, 将式(8)形式化为

$$\mathbf{h}_t = f_{\text{GRU}}(\mathbf{h}_{t-1}, \mathbf{x}_t) \quad (8)$$

其中  $f_{\text{GRU}}(\cdot)$  为 GRU 网络。

GNN 每一次迭代都会使全局节点的嵌入向量得到更新。高阶节点信息在 GNN 中递归传递的形式与神经网络(RNN, recurrent neural networks)处理序列数据的形式相似, GRU 同样可以解决 GNN 中远距离节点间的信息依赖问题, 因此提出了基于记忆单元的图神经网络(MGNN, memory unit-based graph neural networks)算法, 如图 3 所示。

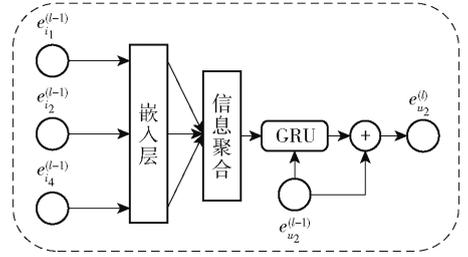


图3 MGNN 的一阶信息聚合

在 MGNN 迭代过程中, 用户  $u_2$  利用前一时刻邻居节点的信息聚合和自身前一时刻的状态  $\mathbf{e}_{u_2}^{(l-1)}$  来共同更新本时刻的状态  $\mathbf{e}_{u_2}^{(l)}$ 。

通过将  $l$  阶 MGNN 堆叠后, 使得用户(项目)节点获得  $l$  阶邻居信息。所提算法不仅考虑了目标节点周围邻居的信息, 并且保留了该节点前一时刻的信息, 以此来更新其嵌入向量:

$$\mathbf{e}_u^{(l)} = \text{LeakyReLU} \left( \mathbf{e}_u^{(l-1)} + \right.$$

$$\left. f_{\text{GRU}} \left( \mathbf{e}_u^{(l-1)}, \frac{1}{|N_u|} \sum_{i \in N_u} \mathbf{e}_i^{(l-1)} \right) \right) \quad (9)$$

$\mathbf{e}_u^{(l)}$  为用户  $u$  在获得了  $l$  阶邻居信息后生成的嵌入向量, 随后  $\mathbf{e}_u^{(l)}$  将会参与到第  $(l+1)$  阶节点向量的更新中。最后将节点的向量更新描述为矩阵形式:

$$\mathbf{E}^{(l)} = \text{LeakyReLU}(\mathbf{P}\mathbf{E}^{(l-1)} + f_{\text{GRU}}(\mathbf{E}^{(l-1)}, \mathbf{P}\mathbf{E}^{(l-1)})) \quad (10)$$

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}, \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{Y} \\ \mathbf{Y}^T & \mathbf{0} \end{bmatrix} \quad (11)$$

其中:  $\mathbf{E}^{(l)}$  为激活后用户和项目的  $l$  阶嵌入向量矩阵,  $\mathbf{D}$  为度矩阵,  $\mathbf{A}$  为邻接矩阵,  $\mathbf{Y}$  为评分矩阵。

### 3.2 CNN 提取节点最终嵌入向量

在 MGNN 中经过  $L$  次的信息聚合后, 得到用户  $u$  (项目  $i$ ) 的多个嵌入向量  $[\mathbf{e}_u^{(1)}, \mathbf{e}_u^{(2)}, \dots, \mathbf{e}_u^{(L)}]$  ( $[\mathbf{e}_i^{(1)}, \mathbf{e}_i^{(2)}, \dots, \mathbf{e}_i^{(L)}]$ )。不同阶的嵌入向量代表了用户在不同阶段的偏好信息, MGNN 低阶嵌入向量包含的信息不够丰富, 并且已被包含在高阶嵌入向

量中. 如果简单地将各阶嵌入向量进行拼接, 会使得信息冗余, 且无法获得向量各维度间的交互信息, 因此采用第  $(L-1)$  阶和第  $L$  阶的嵌入向量交互作为用户的最终嵌入向量. 首先对  $e_u^{(L-1)}$  和  $e_u^{(L)}$  进行张量积运算获得交互矩阵:

$$\mathbf{M} = e_u^{(L-1)} \otimes e_u^{(L)} = e_u^{(L-1)} (e_u^{(L)})^T \quad (12)$$

其中:  $\otimes$  为张量积. 采用张量积对 2 个向量交互有 2 个优势: 一是考虑了向量不同维度间的关系, 从而可以编码更多的信号; 二是与仅保留原始嵌入信息的拼接方式相比, 能生成新信息. 文献[8]的研究结果表明, 对嵌入向量进行显式的特征交

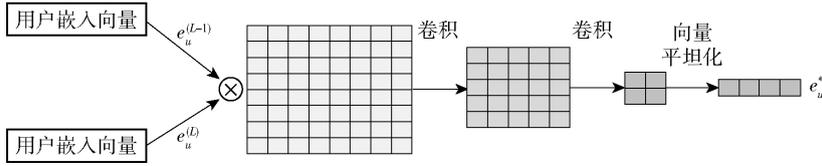


图 4 卷积模块

### 3.3 算法预测

为了便于对整体算法有清晰的认识, 本节首先

互对于深度学习算法的性能提升有很大帮助.

特征交互矩阵  $\mathbf{M}$  中每一个元素都是嵌入向量维度间的交互信息, 因此可以将其看作是一张“图片”. 许多功能强大的学习二维数据的算法都采用 CNN 模型, 如丰富的卷积神经网络 (R-CNN, rich CNN)<sup>[13]</sup>, 多分支卷积网络 (MBCN, multi-branch convolutional network)<sup>[14]</sup> 等. 在推荐系统中构建出二维交互特征, 可以利用 CNN 来学习一个数据间关系的方程, 提高推荐性能. 如图 4 所示, 通过卷积模块获得用户  $u$  的最终嵌入向量  $e_u^*$ , 同理可以获得项目  $i$  的最终嵌入向量  $e_i^*$ .

给出 CMGCF 的整体结构框架, 如图 5 所示.

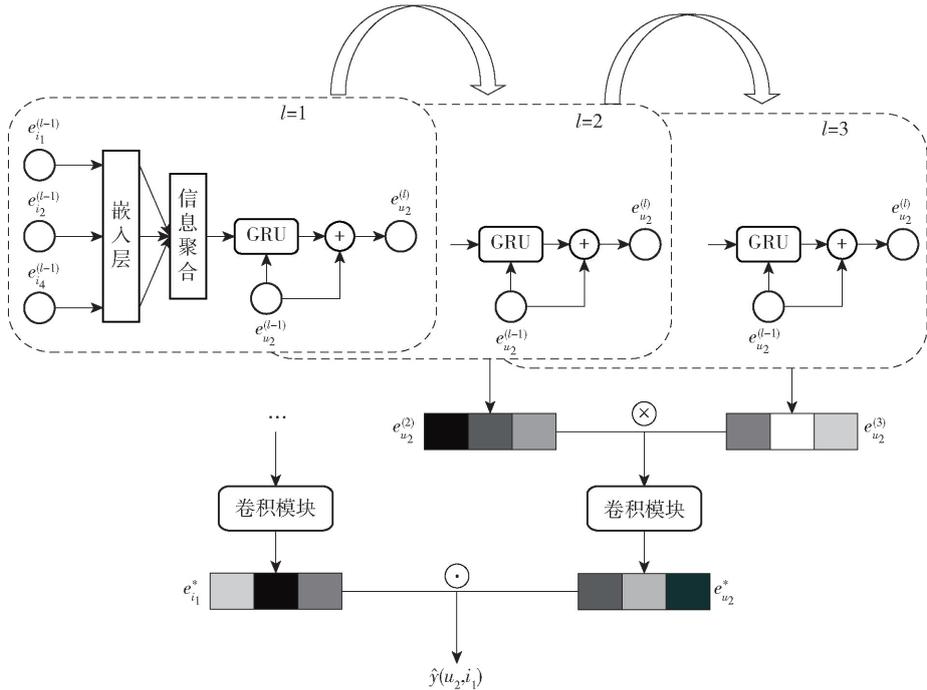


图 5 三阶 CMGCF 结构图

在 3.2 节中利用卷积模块获得用户和项目的最终嵌入向量, 即  $e_u^*$  和  $e_i^*$ . 在预测环节, CMGCF 采用内积 ( $\odot$ ) 方程衡量  $e_u^*$  和  $e_i^*$  的匹配度, 以此估计用户  $u$  对项目  $i$  的喜好程度, 预测函数为

$$\hat{y}_{ui} = \hat{y}(u, i) = e_u^* \odot e_i^* \quad (13)$$

预测分数可以作为推荐系统是否向用户推送项目的依据.

### 3.4 CMGCF 训练

评分预测是一个线性回归问题, 通常采用平方误差作为损失函数:

$$L = \sum_{u,i \in S} (\hat{y}_{ui} - y_{ui})^2 \quad (14)$$

其中:  $\hat{y}_{ui}$  为 CMGCF 的预测分数,  $y_{ui}$  为用户  $u$  对项目  $i$  的真实评分,  $S$  为数据集, 并且采用自适应矩估计优化器去优化损失数.

## 4 实验结果及分析

### 4.1 数据集与参数设置

选择了 4 个具有代表性的公共数据集进行评估, 分别是 Movielens-100k (ML-100K), Movielens-1M (ML-1M), Yahoo Music 和 Personality, 各数据集的详细参数如表 1 所示.

表 1 4 个数据集的参数

| 数据集         | 用户数量  | 项目数量   | 评分数量      | 数据密度/% | 评分范围  |
|-------------|-------|--------|-----------|--------|-------|
| ML-100K     | 943   | 1 682  | 100 000   | 6.30   | 1~5   |
| ML-1M       | 6 040 | 3 883  | 1 000 209 | 4.26   | 1~5   |
| Yahoo Music | 7 642 | 11 916 | 211 231   | 0.23   | 1~5   |
| Personality | 1 820 | 35 196 | 1 028 751 | 1.60   | 0.5~5 |

在实验中, 随机地将每一个数据集的 80% 划分为训练集, 其余的 20% 做为测试集. 在不同的数据集上采用不同的嵌入维度和迭代次数, 参数设置如表 2 所示.

表 2 CMGCF 在 4 个数据集上的参数设置

| 参数   | ML-100K | ML-1M | Yahoo Music | Personality |
|------|---------|-------|-------------|-------------|
| 嵌入维度 | 81      | 81    | 108         | 81          |
| 迭代次数 | 60      | 150   | 120         | 120         |

在 4 个数据集上采用的 CMGCF 阶数均为 3, 批尺寸为 2 048, 学习率为 0.01, 在卷积模块中, 2 次卷积的卷积核大小均为 3, 步长均为 3.

### 4.2 对比算法与评估指标

为了检验 CMGCF 的评分预测性能, CMGCF 对比了现有比较优秀的 3 个推荐算法: NCF 算法<sup>[6]</sup>, 神经矩阵分解 (NeuMF, neural matrix factorization) 算法<sup>[6]</sup> 和 NGCF 算法<sup>[7]</sup>.

NCF 算法利用深度神经网络提取用户与项目之间的交互特征; NeuMF 算法同时利用 MLP 和内积作为用户和项目的交互方程, 并对嵌入向量进行拼接作为交互特征; NGCF 算法对用户和项目间的高阶协同信息编码来更新嵌入向量, 使用内积方程做预测函数.

采用均方根误差 (RMSE, root mean squared error) 作为预测值的评估指标对算法性能进行评价,

其定义为

$$E_{\text{RMSE}} = \sqrt{\frac{1}{n} \sum_{u,i \in S} (\hat{y}_{ui} - y_{ui})^2} \quad (15)$$

其中  $n$  为数据集  $S$  的容量.

### 4.3 性能对比

所提算法与对比算法相比, 在 4 个公共数据集上的评分预测性能均有不同程度提升, 如表 3 所示, 其中“提升”一行所示为所提算法相较于最优对比算法 NGCF 算法的性能改善. 4 种算法在 Yahoo Music 数据集上的性能表现最差, 主要是因为 Yahoo Music 数据集的稀疏度远远高于其他 3 个数据集, 但是 CMGCF 利用 GRU 缓解了 GNN 中高阶邻居节点间信息传递过程中的信息丢失问题, 因此 CMGCF 算法在 Yahoo Music 数据集上的性能比 NGCF 算法提升了 9.27%. 4 个算法在 Personality 数据集上的预测准确度均比在其他数据集上的预测效果更好, 是因为 Personality 具有最大规模的项目数量, 因此推荐算法可以得到更好的泛化, 从而更精准地预测用户的偏好.

表 3 不同算法在 4 个数据集上的 RMSE 对比

| 算法    | ML-100K | ML-1M   | Yahoo Music | Personality |
|-------|---------|---------|-------------|-------------|
| NCF   | 0.917 4 | 0.924 4 | 1.013 2     | 0.824 0     |
| NeuMF | 0.916 4 | 0.922 8 | 1.003 4     | 0.824 8     |
| NGCF  | 0.897 9 | 0.907 5 | 0.993 2     | 0.778 7     |
| CMGCF | 0.880 1 | 0.869 7 | 0.901 1     | 0.757 7     |
| 提升/%  | 1.98    | 4.17    | 9.27        | 2.70        |

### 4.4 稳定性验证

CMGCF 的稳定性是指在不同实验下, RMSE 指标是否保持稳定, 将 CMGCF 在 4 个数据集上进行 3 次实验, 记录 3 次实验在不同训练迭代次数下的 RMSE 平均值. 由图 6 可知, 所提算法在 4 个数据集上的预测性能整体比较稳定, 随着迭代次数的增加,

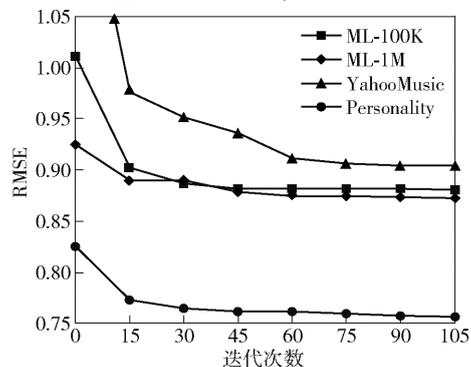


图 6 4 个数据集中 CMGCF 算法的 RMSE 随迭代次数的变化

RMSE 越来越小. 在训练前期, RMSE 下降明显; 训练后期, RMSE 的值逐渐趋于平缓. 这说明 CMGCF 算法逐渐学习到用户和项目之间的关系, 从而提高了项目被用户采纳的概率.

## 5 结束语

CMGCF 算法旨在解决 GNN 中高阶邻居节点间信息传递过程中的信息丢失问题. 首先将用户和项目关系转化为图结构, 采用 GRU 增强高阶邻居节点间的信息传递, 利用 CNN 学习一个提取 MGNN 各阶嵌入向量维度间关系的方程, 并生成用户项目的最终嵌入向量. 在仅利用内积作为评分预测方程的情况下, 预测性能均有提升. 在下一步工作中, 将对用户和项目的评分预测方程的构造进行进一步优化.

### 参考文献:

- [1] Srifi M, Oussous A, Lahcen A A, et al. Recommender systems based on collaborative filtering using review texts: a survey[J]. *Information*, 2020, 11(6): 317.
- [2] Wang Xiang, He Xiangnan, Wang Meng, et al. Neural graph collaborative filtering[C]// Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2019: 165-174.
- [3] Xue Hongjian, Dai Xinyu, Zhang Jianbing, et al. Deep matrix factorization models for recommender systems[C]// International Joint Conference on Artificial Intelligence. Stockholm: IJCAI, 2017: 3203-3209.
- [4] Li Zekun, Cui Zeyu, Wu Shu, et al. Fi-GNN: modeling feature interactions via graph neural networks for CTR prediction[C]// International Conference on Information and Knowledge Management. New York: ACM, 2019: 539-548.
- [5] Chen Cheng, Meng Xiangwu, Xu Zhenghua, et al. Location-aware personalized news recommendation with deep semantic analysis[J]. *IEEE Access*, 2017, 5(99): 1624-1638.
- [6] He Xiangnan, Liao Lizi, Zhang Hanwang, et al. Neural collaborative filtering[C]// The Web Conference. New York: ACM, 2017: 173-182.
- [7] He Xiangnan, Chua T S. Neural factorization machines for sparse predictive analytics[C]// Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2017: 355-364.
- [8] He Xiangnan, Du Xiaoyu, Wang Xiang, et al. Outer product-based neural collaborative filtering[C]// International Joint Conference on Artificial Intelligence. Stockholm: IJCAI, 2018: 2227-2233.
- [9] Xin Xin, Chen Bo, He Xiangnan, et al. CFM: convolutional factorization machines for context-aware recommendation[C]// International Joint Conference on Artificial Intelligence. Stockholm: IJCAI, 2019: 3926-3932.
- [10] Lamb C L, Garcez S A, Gori M, et al. Graph neural networks meet neural-symbolic computing: a survey and perspective[C]// International Joint Conference on Artificial Intelligence. Stockholm: IJCAI, 2020: 4877-4884.
- [11] Zhang Yufeng, Yu Xueli, Cui Zeyu, et al. Every document owns its structure: inductive text classification via graph neural networks[C]// Annual Meetings of Association for Computational Linguistics. Stroudsburg: ACL, 2020: 334-339.
- [12] Qi Xiaojuan, Liao Renjie, Jia Jiaya, et al. 3D graph neural networks for RGBD semantic segmentation[C]// IEEE International Conference on Computer Vision. New York: IEEE, 2017: 5209-5218.
- [13] Girshick B R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// Computer Vision and Pattern Recognition. New York: IEEE, 2014: 580-587.
- [14] Guo Wei, Zhang Can, Guo Huifeng, et al. Multi-branch convolutional network for context-aware recommendation[C]// Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2020: 1709-1712.