智能系统学报
CAAI TRANSACTIONS ON INTELLIGENT SYSTEMS

基于用户记忆矩阵的长序列推荐算法

鹿祥志, 孙福振, 王绍卿, 董家玮, 吴相帅

引用本文:
鹿祥志,孙福振,王绍卿,董家玮,吴相帅. 基于用户记忆矩阵的长序列推荐算法[J]. 智能系统学报, 2023, 18(3): 517–524.
LU Xiangzhi,SUN Fuzhen,WANG Shaoqing,DONG Jiawei,WU Xiangshuai. Long sequence recommendation algorithm based on user memory matrix[J]. *CAAI Transactions on Intelligent Systems*, 2023, 18(3): 517–524.

在线阅读 View online: https://dx.doi.org/10.11992/tis.202110003

DOI: 10.11992/tis.202110003
网络出版地址: https://kns.cnki.net/kcms/detail/23.1538.TP.20230411.1511.002.html

基于用户记忆矩阵的长序列推荐算法

鹿祥志, 孙福振, 王绍卿, 董家玮, 吴相帅
(山东理工大学 计算机科学与技术学院, 山东 淄博 255000)

摘　要: 传统的循环神经网络, 如长短期记忆网络和门控循环单元, 记忆能力有限而且记忆数据的存取不够灵活, 对较长序列的特征捕捉有着先天的不足。记忆网络具有存储长时记忆的特点, 而且对于记忆数据的存取更加灵活多变, 因此本文在基于会话的推荐算法中引入了记忆网络。本文设计了一个层次化的推荐模型, 模型分为 2 层。第 1 层为会话级的 GRU 模型, 此模型用来刻画当前会话的序列特征, 从而预测下一个项目。第 2 层为用户级的记忆网络模型, 这个模型用来刻画用户长期兴趣的变化。本文提出的模型能有效地捕捉到用户的短期和长期兴趣, 进而提升推荐的性能。公开数据集上的实验证明, 在会话个数为 10 相对于会话个数为 5 的性能提升对比中, 本文所提带有用户记忆矩阵的分层网络算法在召回率和平均倒数排名的提升度上相对于分层门控循环单元都有 4% 的增加。

关键词: 记忆网络; 层次化; 长期兴趣; 短期兴趣; 长短期记忆网络; 门控循环单元; 长序列推荐; 会话推荐
中图分类号: TP309.2　　**文献标志码:** A　　**文章编号:** 1673−4785(2023)03−0517−08

中文引用格式: 鹿祥志, 孙福振, 王绍卿, 等. 基于用户记忆矩阵的长序列推荐算法 [J]. 智能系统学报, 2023, 18(3): 517–524.
英文引用格式: LU Xiangzhi, SUN Fuzhen, WANG Shaoqing, et al. Long sequence recommendation algorithm based on user memory matrix[J]. CAAI transactions on intelligent systems, 2023, 18(3): 517–524.

Long sequence recommendation algorithm based on user memory matrix

LU Xiangzhi, SUN Fuzhen, WANG Shaoqing, DONG Jiawei, WU Xiangshuai
(College of Computer Science and Technology, Shandong University of Technology, Zibo 255000, China)

Abstract: Traditional recurrent neural networks, such as long-term and short-term memory networks (LSTM) and gated recurrent unit (GRU), have limited memory capacity and inflexible access to memory data, which have inherent shortcomings in capturing features of longer sequences. The memory network has the characteristics of storing long-term memory, and the access to memory data is more flexible and changeable. Therefore, this paper introduces the memory network in the session-based recommendation algorithm. In this paper, we design a hierarchical recommendation model, which is divided into two layers. The first layer is the session-level GRU model, which is used to characterize the sequence of current session and predict the next item. The second layer is the user-level memory network model, which is used to describe the changes in users' long-term interests. The model proposed in this paper can effectively capture the short-term and long-term interests of users and thus improve the performance of recommendations. The experiments on public data sets demonstrate that the proposed hierarchical network with user memory (HNUM) algorithm has 4% increase in both recall rate and mean inverse ranking improvement relative to hierarchical gated recurrent unit (HGRU) for a performance improvement comparison of 10 sessions versus 5 sessions.

Keywords: memory network; hierarchy; long-term interest; short-term interest; long short-term memory network; gated recurrent unit; long sequence recommendation; session recommendations

伴随大数据时代的发展和智能移动终端的普及, 推荐系统仍然是解决信息过载问题的有效手段[1]。基于会话的推荐系统 (session-based recommender systems, SBRS) 是序列推荐系统 (sequential recommender systems, SRSs)[2] 的一个分支, 是当前推荐算法研究的热点之一[3]。深度学习技术在近年来在学术界和工业界掀起一股热潮, 越来越多的学者将深度学习技术应用到推荐系统中, 而且取得了丰硕的成果[4]。深度学习模型拥有更

收稿日期: 2021−10−05.　**网络出版日期:** 2023−04−11.
基金项目: 国家自然科学基金项目 (61841602); 山东省自然科学基金项目 (ZR2020MF147).
通信作者: 孙福振. E-mail: sunfuzhen@sdut.edu.cn.

加强大的学习能力，避免了传统基于机器学习的推荐模型需要人工设计特征的问题[5]。循环神经网络善于处理序列任务，因此在具有序列特征的推荐任务中有着良好的表现[6]。虽然在许多基于会话的推荐域中很难找到用户标识符，但也有一些领域中用户配置文件是随时可用的，这时可以对用户历史信息加以利用[7]。因此，Quadrana 等[8]提出一种层次化的推荐模型，该模型设计了两个层次的循环神经网络（recurrent neural network, RNN），分别为用户级的 RNN 模型和会话级的 RNN 模型。用户级的 RNN 模型主要根据用户的一系列会话捕捉存储用户的长期兴趣。之前的工作都是考虑用户在当前会话中的顺序行为，并未考虑到用户的主要目的，Li 等[9]在会话推荐的基础上提出神经注意力推荐机（NARM），使用带有注意力机制的混合编码器结构捕捉用户的在会话中主要目的。

当今，由于网络资源的随手可得，人们喜欢利用碎片化的时间进行各种网上行为，如浏览新闻、阅读文章、选购商品等[10]。碎片化时间使得产生的会话更多、更零散，用户的兴趣漂移也变得更快，用户的长期兴趣变得难以捕捉[11]。虽然 RNN 在序列任务上有着比较良好的表现，但是，RNN 可以存储的信息是有限的，随着记忆单元存储的内容越来越多，其丢失的信息也越来越多[12]。即使是循环神经网络的变体长短期记忆网络 (long short-term memory, LSTM) 也因为无法存储长期记忆而只能被称为长的短期记忆网络。Weston 等[13]介绍了一种新的学习模型——记忆网络，同年 Google DeepMind 团队提出了神经图灵机[14]，它们都使用外部存储器来进行记忆。神经图灵机设计了基于注意力读写操作，使得对记忆的读取更加灵活。2015 年 Sukhbaatar 等[15]又提出一种端到端的记忆网络。外部存储空间其实是一个记忆矩阵。2021 年 Tan 等[16]提出一种基于动态记忆网络的动态注意力网络（DMAN），将用户的短期和长期兴趣存储在与循环相连接的记忆网络中，实现有效地联合推荐。记忆网络已经证实在推荐系统可以实现存储大量的用户历史交互序列，并且可以对记忆网路有选择的记性地读取，避免了对整个序列进行操作[17]。同时，记忆网络可以解决会话推荐中长序列特征无法捕捉的问题，因此，结合 Quadrana 等[8]的工作，本文提出一种具有用户记忆矩阵的层次化网络 (hierarchical networks with user memory matrix, HNUM)，使用记忆矩阵作为记忆网络中记忆空间的具体实现，能够有效捕捉和刻画用户的短期和长期兴趣，并在公开数据集 Movie Lens-25M 和 Adressa 上设计实验验证和分析算法的有效性。

1　算法描述

首先介绍本文提出模型的整体结构，然后分别阐述其中的每个模块。HNUM 模型分为两层，模型的总体结构如图 1 所示。

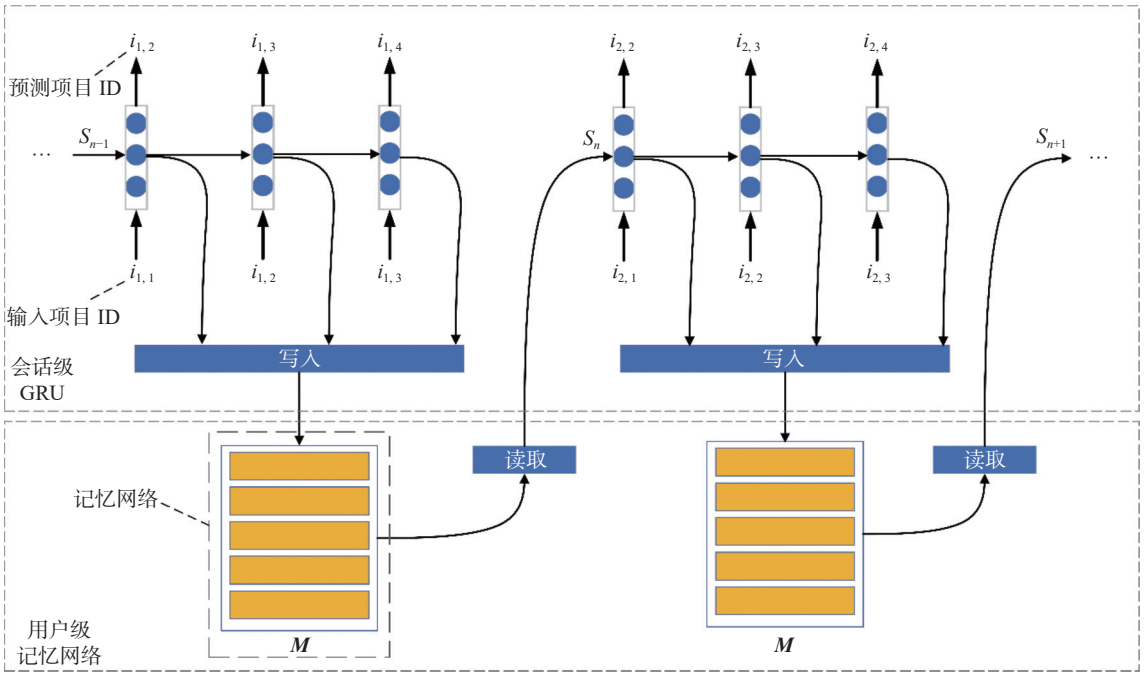

图 1　HNUM 框架

Fig. 1　HNUM framework

模型第一层为会话级的门控循环单元(gated recurrent unit, GRU)模型, 此模型用来刻画当前会话的序列特征, 从而预测下一个项目。模型第二层为用户级的记忆网络模型, 记忆网络用来刻画用户长期兴趣的变化。在用户的一个会话中, 会话开始后, 读取模块读取当前用户对应的记忆矩阵$\boldsymbol{M}$中的记忆向量, 读取得到的记忆作为用户的偏好向量, 用于对 GRU 单元的隐藏层进行初始化。在每个时间步结束后, 得到 GRU 的隐状态, 将其通过写入模块存储到记忆矩阵$\boldsymbol{M}$中。当用户的一个会话结束下一个会话开始时, 再次进行同样的流程。

1.1　HNUM 形式化描述

定义$U=\{u_1,u_2,\cdots,u_N\}$为所有用户的集合, $S^u=\{s_1^u, s_2^u,\cdots\}$为用户$u$的会话集合, $V^s=\{v_1,v_2,\cdots\}$为用户的某个会话s中产生交互的项目序列, 其中, v_i是整个系统所有项目中的一个被交互过的项目。$\boldsymbol{M}^u=[m_1^u\ m_2^u\ \cdots\ m_K^u]\in\mathbf{R}^{D\times K}$是用户$u$的记忆矩阵, $m_k^u\in\mathrm{R}^D$是$\boldsymbol{M}^u$的第k个记忆向量, 用来存储用户的长期兴趣。$\boldsymbol{M}^u$的形状大小取决于记忆矩阵的记忆向量数量K和矩阵中记忆向量的长度D。其中, K和D属于模型的超参数。

1.2　记忆的读取和更新

整个框架的核心部分在于记忆向量的读取和更新, 下面分别介绍 2 个模块。

1.2.1　记忆读取模块

读取模块主要负责将记忆矩阵中的记忆, 即用户的长期兴趣, 按照一定的规则读取出来, 用于指导会话阶段的训练。具体来说, 设$\boldsymbol{p}^u$是用户u的兴趣向量, $\boldsymbol{p}^u$是根据当前会话的交互项目v_i作为输入去读取$\boldsymbol{M}^u$而得到的。$\boldsymbol{p}^u$可以表示为

$$\boldsymbol{p}^u=\mathrm{READ}(\boldsymbol{M}^u,\boldsymbol{v}_i)$$

式中: $\boldsymbol{v}_i$是当前会话中的第i个交互项目的嵌入向量。直观上, 之前的记忆向量会对当前的兴趣有不同的影响, 因此在此引入了注意力机制[18], 对不同的记忆向量赋予不同的权重值。READ(•)操作的具体过程为

$$w_{i,k}=v_i\cdot m_k^u \tag{1}$$

$$z_{ik}=\frac{\exp(\beta w_{ik})}{\sum_j\exp(\beta w_{ij})} \tag{2}$$

$$p^u=\sum_{k=1}^{K}z_{ik}\cdot m_k^u \tag{3}$$

式中: β是一个强度参数, 可以放大或者减小聚焦的程度, β=1 时就是一个标准的 softmax[19]。z_{ik}作为注意力权重, 利用它推导出用户u的兴趣向量$\boldsymbol{p}^u$, 因此, 可以根据用户历史行为对当前项目的影响, 来访问用户历史行为。

1.2.2　记忆写入模块

写入模块负责将一个时间步结束后的 GRU 隐状态更新到记忆矩阵中, 本质上属于一个状态更新问题[20]。神经图灵机参考了 LSTM[21] 的更新门的思想: 先用输入门决定需要添加的信息, 再用遗忘门决定要丢弃的信息, 最后用更新门加上增加的信息并减去丢弃的信息。具体来说, 神经图灵机会生成一个擦除向量和一个添加向量, 向量中每个元素的值大小范围为 0~1, 表示要增加或者删除的信息。由于整个过程都是矩阵的加减乘除, 所有的读写操作都是可微分的, 因此可以用梯度下降法训练整个参数模型。对于擦除向量$\boldsymbol{e}_i^{\mathrm{rase}}$:

$$\boldsymbol{e}_i^{\mathrm{rase}}=\sigma(\boldsymbol{E}^{\mathrm{T}}\boldsymbol{h}_i+\boldsymbol{b}_e)$$

式中: $\sigma(\cdot)$是 sigmoid 函数, $\boldsymbol{E}$和$\boldsymbol{b}$是需要学习的擦除参数; $\boldsymbol{h}_i$是用户当前的隐状态。给定注意权重和擦除向量, 对特征偏好记忆进行更新:

$$m_k^u\leftarrow m_k^u\odot(1-z_{ik}\cdot\boldsymbol{e}_i^{\mathrm{rase}})$$

其中z_{ik}是注意力权重。擦除后, 利用添加向量$\boldsymbol{a}_i^{\mathrm{dd}}$更新特征偏好存储器, 也就是记忆矩阵:

$$\boldsymbol{a}_i^{\mathrm{dd}}=\tanh(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{h}_i+\boldsymbol{b}_a)$$

$$m_k^u\leftarrow m_k^u+z_{ik}.\boldsymbol{a}_i^{\boldsymbol{dd}}$$

其中$\boldsymbol{A}$和$\boldsymbol{b}_a$是添加操作中需要学习的参数。这种擦除−添加更新的策略允许遗忘和加强学习过程中对用户偏好嵌入向量的学习, 模型可以通过自动学习擦除和添加参数来决定哪些信号需要减弱, 哪些信号需要加强。

1.3　损失函数

经典贝叶斯个性化排序(Bayesian personalized ranking ,BPR)[22] 是一种利用成对排序损失的矩阵因子分解方法, BPR 比较的是一个正例和一个负例的得分。在损失迭代计算过程中, 将正例项目的得分与同一个 batch 中其他会话的下一个项目的得分进行比较, 并使用他们的平均值作为损失。在某一会话的某一时刻的损失定义为

$$L_s=-\frac{1}{N_s}\sum_{j=1}^{N_s}\ln(\sigma(\hat{r}_{s,i}-\hat{r}_{s,j})) \tag{4}$$

式中: N_s是表示采样的负样本数; $\hat{r}_{s,i}$是正样本的分数; $\hat{r}_{s,j}$是负样本的分数; $\hat{r}_{s,i}$和$\hat{r}_{s,j}$都是 GRU 的输出经过 LeakyReLU 激活函数得到的; σ是 sigmoid 函数。

1.4　算法流程

1) 将会话按用户进行分组, 将每个用户的会话按照时间排列顺序。同一会话内用户与项目交

互序列是以时间为序的。

2) 在同一个用户的训练中，将不同的会话横向拼接起来送入会话级的 GRU 中。

3) 记忆读取模块根据用户当前会话的 GRU 隐状态对记忆矩阵进行读取记忆，将读取得到的记忆作为用户偏好向量初始化 GRU 的隐藏层单元。

4) 当 GRU 的一个时间步结束时，记忆写入模块将该会话的最后状态写入到记忆网络中，更新记忆矩阵，进行用户级上的训练。HNUM 推荐执行过程伪代码如算法 1 所示。

算法 1　HNUM 算法

输入　元祖<UserId,SessionId,ItemId>

输出　预测评分$y=\{y_1,y_2,\cdots,y_m\}$

1) 按用户对会话进行分组

2) 初始化记忆矩阵 $\boldsymbol{M}$

3) For i in epoch:

4) For j in epoch:

5) 通过 reader 读取记忆矩阵 $\boldsymbol{M}$

6) if 新的会话

7) 初始化 GRU 的隐状态

8) 通过 reader 将状态写入记忆矩阵 $\boldsymbol{M}$

9) 根据式(4)计算损失

10) end for

11) end for

2　实验分析

2.1　实验数据集

本文所用数据集详细信息如表 1 所示。

表 1　数据集详细信息

Table 1　Details of the dataset

数据集	MovieLens-25M	Adressa
用户数	9879	26777
项目数	27810	37405
会话数	105193	292483
平均会话长度	24.3	22.5

2.1.1　Movie Lens-25M

Movie Lens-25M(以下简称 Movie Lens) 是美国 Minnesota 大学 GroupLens 小组开发的 Movie Lens 站点所提供的数据集。本文所用的版本公布于 2019 年 12 月，是一个在电影推荐中广泛使用的公开数据集。数据集包含了 Movie Lens 网站上大约 2 500 万条的评分记录。为了适合本文提出的算法，首先为每个用户的评分数据按照时间排序，又因为 Movie Lens 数据集没有划分会话的标志，因此本文按照天对数据进行划分。去掉了长度小于 5 的会话，同时去掉了会话个数小于 6 个的用户。对于每个用户，其 80% 的会话作为训练集数据，20% 作为测试集。

2.1.2　Adressa

Adressa[23] 是 RecTech 项目中发布的一个新闻数据集，数据集包含了用户的上下文信息和新闻的标题、内容等详细信息。本文的实验需要获得用户的长期历史行为信息，因此只能选择数据集中已注册的用户作为实验数据。数据集提供了用户所用设备类型、用户所在位置等信息。数据集中有会话的开始和停止符号，可以据此分割会话。该数据集有 2 个版本，一个是 Adresseavisen 新闻门户网站上 10 周流量的包含 2 000 万阅读行为的大型数据集，另一个是只有 1 周流量的包含 200 万阅读行为的小型数据集。本文使用的是包含 2 000 万阅读行为的大型数据集，筛选出其中会话个数不少于 5 个且会话长度不少于 6 的用户，将其中 80% 作为训练集，20% 作为测试集。

2.2　评价指标

Recall@K：本文的第一个评估指标是 Recall@K，表示所有测试用例中前 K 个项目中拥有所需项目的比例。召回不考虑项目的实际排名。这很好地模拟了某些实际场景，在这些场景中没有突出的建议，绝对顺序也不重要。召回率 Recall 在传统意义上的计算公式为

$$R^{\mathrm{Recall}}=\frac{N^{\mathrm{TP}}}{N^{\mathrm{TP}}+N^{\mathrm{FN}}}$$

式中：N^{TP} 表示正样例被预测为正样例的数量；N^{FN} 表示正样例被预测为负样例的数量；Recall 度量有多个正例被分为正例。在推荐系统的个性化排序任务中，召回率的计算被定义为

$$R^{\mathrm{Recall}}=\frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|T(u)|} \tag{5}$$

式中：$R(u)$指的是为用户u推荐的N个物品的列表；$T(u)$指的是用户u在测试集上喜欢的物品的集合。本文采用文献 [6] 所使用的计算 Recall 的方法，该方法将会话推荐任务视为一个逐项推荐的任务，即会话当前阶段的目标项目只有一个。在式(5)中，表示$T(u)$的长度为 1，当推荐列表的长度取值为 20 时，目标项目的得分如果排在前 20，召回率的值就记为 1，否则为 0。最后召回率的得分是所有用户的平均值。

MRR@K：实验中使用的第 2 个度量是平均

倒数排名 (mean reciprocal rank, MRR) 这是所需项目的倒数排名的平均值。如果排名大于K, 则倒数排名被设为 0。MRR 考虑项目的排序, 这在注重推荐顺序的情况下是很重要的。计算公式为

$$R^{\mathrm{MR}}=\frac{1}{Q}\sum_{i=1}^{|Q|}\frac{1}{r_i}$$

式中: $|Q|$表示用户感兴趣的项目数量, r_i表示用户感兴趣的项目在推荐列表中的排名。当真实值的排名大于设置的截断值时, 排名的倒数设置为 0。MRR 更能反映在排序问题中推荐的质量, 因为人们往往更关注推荐列表中的前几项, 当真实值的排名十分靠后的时候, 即使真实值在推荐列表中, 也不能算高质量的推荐结果[24]。

2.3　实验设计

首先介绍实验所用的软件和硬件平台。本文使用 Tensor flow 框架来完成模型的搭建。并在硬件平台 TESLA P100 上进行实验。训练过程中, 使用 RMSProp 优化器优化模型, 将 batch_size 设置为 128。实验中发现, 对于本文实验环境, 当 batch_size 为 128 时更能兼顾性能于效率。模型的参数通过正态分布进行初始化, 正态分布的均值为 0, 标准差为 0.01; 初始学习率为 0.000 15, 学习率衰减系数为 0.96; 为防止过拟合问题, Dropout 的 keep_prob 参数为 0.8; GRU 单元数量为 100 个。实验中发现, 由于网络结构较为复杂, 在使用 Tanh 作为激活函数的时候, 经常会出现激活函数的饱和现象, 导致实验结果虚高, 因此, 在 GRU 单元的输出层之后使用 LeakyReLU 作为激活函数。LeakyReLU 函数的公式为

$$y_i=\begin{cases}x_i, & x_i\geqslant 0\\ a_i x_i, & x_i<0\end{cases}$$

其中$a_i\in(0,1)$。LeakyReLU 函数不会产生饱和, 也避免了神经元死亡。记忆矩阵中, 记忆向量的个数 K 在 [2, 15] 之内确定, 记忆向量的长度 D 设置为 100。所有的超参数的选择都是根据实验结果进行调整得到的最优值。

2.4　实验结果分析

2.4.1　模型对比

为探究在 HNUM 模型的推荐性能, 将 HNUM 模型与 HGRU 模型 (hierarchical recurrent neural networks) 和 GRU4REC 模型进行对比实验。

GRU4REC 模型[6], 该模型是比较经典的基于深度学习的会话推荐模型, 模型使用 GRU 捕捉用户在会话中的兴趣, 进而根据兴趣生成推荐列表。该模型是会话推荐领域常用的基线算法模型。

HGRU 模型[8] 是一种层次化的会话推荐模型, 模型的两层都使用 GRU 单元捕捉用户兴趣, 该模型在跨会话的 RNN 端点上进化出潜在的隐藏状态, 利用 GRU 的隐状态表示用户的历史兴趣。在刻画用户长期兴趣方面, 模型受限于 GRU 单元的记忆能力。

短期注意力/记忆优先级模型 (short term attention/memory priority, STAMP) 是一种[25] 使用了注意力层来替代以往的所有 RNN 编码器, 来捕获当前会话中用户的一般兴趣和用户最后一次点击的短期兴趣。

对比实验的实验参数设置, HNUM 模型, 将记忆向量的个数设置为 20。GRU4REC 的模型使用 100 个 GRU 单元, batch_size 为 128。HGRU 模型, 会话层的 GRU 和用户层的 GRU 单元个数都设置为 100。表 2~5 为 HNUM 模型在 Adressa 和 MovieLens-25M 上的 Recall@K 和 MRR@K 的结果。

表 2　Adressa 数据集上会话个数为 5 的 Recall@K 结果
Table 2　Results of Recall@K on the Adressa dataset with the number of sessions of five

模型	Recall@5	Recall@10	Recall@20
GRU4REC	0.1023	0.1854	0.3074
HGRU	0.1607	0.2897	0.4529
STAMP	0.1623	0.2882	0.4388
HNUM	0.1638	0.2935	0.4550

表 3　MovieLens-25M 数据集上的 Recall@K 的结果
Table 3　Results of Recall@K on the MovieLens-25M dataset

模型	Recall@5	Recall@10	Recall@20
GRU4REC	0.0213	0.0450	0.0831
HGRU	0.0247	0.0571	0.0958
STAMP	0.0257	0.0523	0.0792
HNUM	0.0286	0.0623	0.1034

表 4　Adressa 数据集上会话个数为 5 的 MRR@K 结果
Table 4　Results of MRR@K on the Adressa dataset with the number of sessions of five

模型	MRR@5	MRR@10	MRR@20
GRU4REC	0.0620	0.0761	0.0846
HGRU	0.0906	0.1104	0.1224
STAMP	0.0921	0.1088	0.1192
HNUM	0.0931	0.1129	0.1249

表 5　MovieLens-25M 数据集上的 MRR@K 的结果
Table 5　Results of MRR@K on the MovieLens-25M dataset with the number of sessions of five

模型	MRR@5	MRR@10	MRR@20
GRU4REC	0.0090	0.0146	0.0173
HGRU	0.0112	0.0191	0.0298
STAMP	0.0121	0.0162	0.0186
HNUM	0.0134	0.0227	0.0326

从表 2~5 可以看出，在同样是基于会话的推荐算法中，HGRU、STAMP 和 HNUM 模型的推荐效果总体上要好于 GRU4REC 模型。GRU4REC 模型没有考虑用户的历史行为信息，只捕捉了用户在当前会话的兴趣，而 HGRU 模型和 HNUM 模型利用了用户的历史行为，因此推荐性能更佳。在同样的数据集下 HGRU 模型和 STAMP 模型在 Recall 上的表现也弱于 HNUM。因为 HGRU 模型在刻画用户长期兴趣的时候，将用户的兴趣压缩为 GRU 单元的隐状态，这种方式不利于对历史状态的动态提取和选择，STAMP 模型在排名截断值为 5 时表现优于 HGRU，因为 STAMP 除了捕捉用户的长期兴趣外，还考虑到用户的短期兴趣，从用户最后一次的操作去考虑用户的当前兴趣。而使用记忆网络的 HNUM 模型避免了这种情况，在记忆网络中存储用户的长期兴趣和短期兴趣。在实验结果中可以看出，各算法在 Movie Lens 上的表现都比在 Adressa 数据集上要差。Movie Lens 不是针对会话推荐的数据集，数据集并没有表明评分时间顺序与观影顺序有关，因此基于会话的推荐算法在 Movie Lens 数据集上表现并不理想。

相比基线算法 GRU4REC、HGRU 和 STAMP，本文所提出的算法 HNUM 在 Recall 和 MRR 上都有更好的表现，验证了所提算法的有效性。

2.4.2　长期记忆能力探究

为了探寻模型对用户长期兴趣的记忆能力，设计实验对比了模型在会话个数大于 10 个和会话个数为 5 个时候的不同表现。两组数据分为会话个数为 5 个的数据集和会话个数为 10 个的数据集。实验对比了 HGRU 模型和 HNUM 模型在面对会话个数增加的时候性能的提升比例。比如，对于 HGRU 模型，会话个数为 10 的时候，Recall@20 为 0.4693，会话个数为 5 时，Recall@20 为 0.4529，性能提升了 3.6%。具体结果如表 6 和表 7 所示。

将表 6 和表 7 与表 2 和表 3 中的数据对比，可以看到，当选取会话个数为 10 的数据时，HGRU 和 HNUM 在 Recall 和 MRR 上均有提升，但是提升程度不同。图 3 和图 4 对比了分别 2 个模型使用 10 个会话相对于使用 5 个会话的性能提升率。

表 6　Adressa 数据集会话个数为 10 时 Recall@K 的结果
Table 6　Rsults of Recall@K on the Adressa dataset with the number of sessions of ten

模型	Recall@5	Recall@10	Recall@15	Recall@20
HGRU	0.1682	0.3027	0.3976	0.4693
HNUM	0.1775	0.3146	0.4096	0.4831

表 7　Adressa 数据集会话个数为 10 时 MRR@K 的结果
Table 7　Results of MRR@K on the Adressa dataset with the number of sessions of ten

模型	MRR@5	MRR@10	MRR@15	MRR@20
HGRU	0.0954	0.1159	0.1240	0.1271
HNUM	0.1018	0.1225	0.1307	0.1353

为了探究长会话对 HNUM 模型的提升，设计在不同 Top-K 下实验，具体结果如图 2~3 所示。

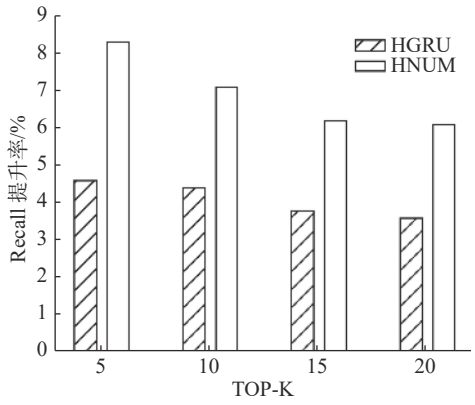

图 2　长会话对 Recall 提升的对比
Fig. 2　Comparison of long sessions on Recall boost

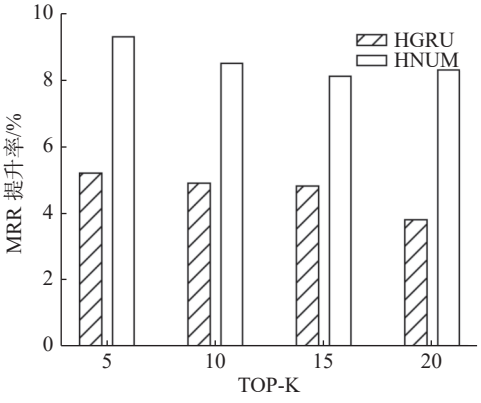

图 3　长会话对 MRR 提升对比
Fig. 3　Comparison of long sessions on MRR boost

由图 2 和图 3 可知，在面对更长的会话个数时 HNUM 模型可以获得推荐效果的提升，原因是

记忆网络具有存储长序列信息的能力, 更多的会话可以带来更多的用户信息, 而这些用户信息都可以被存储在记忆网络中。

3 结束语

为解决传统的循环神经网络,如 LSTM 和 GRU,记忆能力有限而且记忆数据的存取不够灵活, 对较长序列的特征捕捉有着先天的不足的问题, 本文提出了具有用户记忆矩阵的层次化网络 HNUM, 使用记忆矩阵存储每个时间步的隐状态, 用于刻画用户的长期兴趣, 设计了相应的读写模块读取和更新记忆向量。实验结果表明 HNUM 在用户会话个数较多的时候, 相比基线算法有更好的推荐性能提升。在会话个数为 10 相对于会话个数为 5 的性能提升中, 本文所提算法 HNUM 在 Recall 和 MRR 的提升度上相对于基线都有 4 个百分点的增加。

Transformer 在推荐系统中已表现出巨大潜力, 相比传统的 RNN、CNN 模型, Transformer 可以将用户的短期行为序列进行建模, 通过自注意力机制计算某一时刻的项目与所有时刻项目的相关性。下一步工作可以对 Transformer 的融合进一步的探索。

参考文献:

[1] 何鹏, 吴浩, 曾诚, 等. Truser: 一种基于可信用户的服务推荐方法 [J]. 计算机学报, 2019, 42(4): 851–863.
HE Peng, WU Hao, ZENG Cheng, et al. Truser: an approach to service recommendation based on trusted users[J]. Chinese journal of computers, 2019, 42(4): 851–863.

[2] WANG SHOUJIN, HU LIANG, WANG YAN, et al. Sequential recommender systems: challenges, progress and prospects[EB/OL]. (2019−12−28)[2021−10−05].https://arxiv.org/abs/2001.04830.

[3] WANG SHOUJIN, CAO LONGBING, WANG YAN, et al. A survey on session-based recommender systems [EB/OL]. (2019−02−13)[2021−10−05].https://arxiv.org/abs/1902.04864.

[4] 黄立威, 江碧涛, 吕守业, 等. 基于深度学习的推荐系统研究综述 [J]. 计算机学报, 2018, 41(7): 1619–1647.
HUANG Liwei, JIANG Bitao, LYU Shouye, et al. Survey on deep learning based recommender systems[J]. Chinese journal of computers, 2018, 41(7): 1619–1647.

[5] 吕刚, 张伟. 基于深度学习的推荐系统应用综述 [J]. 软件工程, 2020, 23(2): 5–8.
LYU Gang, ZHANG Wei. Survey of deep learning applied in recommendation system[J]. Software engineering, 2020, 23(2): 5–8.

[6] HIDASI B, KARATZOGLOU A, BALTRUNAS L, et al. Session-based recommendations with recurrent neural networks[EB/OL]. (2015−11−21)[2021−10−05].https://arxiv.org/abs/1511.06939.

[7] 高茂庭, 徐彬源. 基于循环神经网络的推荐算法 [J]. 计算机工程, 2019, 45(8): 198–202,209.
GAO Maoting, XU Binyuan. Recommendation algorithm based on recurrent neural network[J]. Computer engineering, 2019, 45(8): 198–202,209.

[8] QUADRANA M, KARATZOGLOU A, HIDASI B, et al. Personalizing session-based recommendations with hierarchical recurrent neural networks[C]//Proceedings of the Eleventh ACM Conference on Recommender Systems. New York: ACM, 2017: 130–137.

[9] LI Jing, REN Pengjie, CHEN Zhumin, et al. Neural attentive session-based recommendation[C]//Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. New York: ACM, 2017: 1419–1428.

[10] 夏永生, 王晓蕊, 白鹏, 等. 基于时序和距离的门控循环单元兴趣点推荐算法 [J]. 计算机工程, 2020, 46(1): 52–59.
XIA Yongsheng, WANG Xiaorui, BAI Peng, et al. Point of interest recommendation algorithm of gated recurrent unit based on time series and distance[J]. Computer engineering, 2020, 46(1): 52–59.

[11] 牛耀强, 孟昱煜, 牛全福. 基于异质注意力循环神经网络的文本推荐 [J]. 计算机工程, 2020, 46(10): 52–59.
NIU Yaoqiang, MENG Yuyu, NIU Quanfu. Text recommendation based on heterogeneous attention recurrent neural network[J]. Computer engineering, 2020, 46(10): 52–59.

[12] 刘建伟, 王园方, 罗雄麟. 深度记忆网络研究进展 [J]. 计算机学报, 2021, 44(8): 1549–1589.
LIU Jianwei, WANG Yuanfang, LUO Xionglin. Research and development on deep memory network[J]. Chinese journal of computers, 2021, 44(8): 1549–1589.

[13] WESTON J, CHOPRA S, BORDES A. Memory networks[EB/OL]. (2014−10−15)[2021−10−05].https://arxiv.org/abs/1410.3916.

[14] GRAVES A, WAYNE G, DANIHELKA I. Neural Turing machines[EB/OL]. (2014−10−20)[2021−10−05].https://arxiv.org/abs/1410.5401.

[15] SUKHBAATAR S, SZLAM A, WESTON J, et al. End-to-end memory networks[EB/OL]. (2015−03−31)[2021−10−05].https://arxiv.org/abs/1503.08895.

[16] TAN Qiaoyu, ZHANG Jianwei, LIU Ninghao, et al. Dynamic memory based attention network for se-

quential recommendation[EB/OL]. (2021−02−18)[2021−10−05].https://arxiv.org/abs/2102.09269.

[17] CHEN Xu, XU Hongteng, ZHANG Yongfeng, et al. Sequential recommendation with user memory networks[C]// Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. Marina Del Rey: ACM, 2018: 108–116.

[18] TAO Ye, WANG Can, YAO Lina, et al. TRec: sequential recommender based on latent item trend information[C]// 2020 International Joint Conference on Neural Networks. Glasgow: IEEE, 2020: 1−8.

[19] SUN Yang, YUAN Fajie, YANG Min, et al. A generic network compression framework for sequential recommender systems[C]//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2020: 1299–1308.

[20] JI Wendi, WANG Keqiang, WANG Xiaoling, et al. Sequential recommender via time-aware attentive memory network[C]//Proceedings of the 29th ACM International Conference on Information & Knowledge Management. Virtual Event Ireland. New York: ACM, 2020: 565–574.

[21] KALATIAN A, FAROOQ B. A context-aware pedestrian trajectory prediction framework for automated vehicles[J]. Transportation research part C:emerging technologies, 2022, 134: 103453.

[22] MA Chen, KANG Peng, LIU Xue. Hierarchical gating networks for sequential recommendation[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Anchorage: ACM, 2019: 825–833.

[23] GULLA J A, ZHANG Lemei, LIU Peng, et al. The Adressa dataset for news recommendation[C]//Proceedings of the International Conference on Web Intelligence. New York: ACM, 2017: 1042–1048.

[24] DONG Xinzhou, JIN Beihong, ZHUO Wei, et al. Improving sequential recommendation with attribute-augmented graph neural networks[C]//Pacific-Asia Conference on Knowledge Discovery and Data Mining. Cham: Springer, 2021: 373−385.

[25] LIU Qiao, ZENG Yifu, MOKHOSI R, et al. STAMP: short-term attention/memory priority model for session-based recommendation[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London: ACM, 2018: 1831−1839.

作者简介：

鹿祥志，硕士研究生，主要研究方向为推荐系统。

孙福振，副教授，博士，主要研究方向为数据挖掘、智能信息处理。发表学术论文 30 余篇。

王绍卿，副教授，博士，主要研究方向为推荐系统、数据挖掘。