

基于注意力机制的用户动态兴趣推荐算法

刘星宇, 谢颖华

东华大学信息科学与技术学院, 上海

收稿日期: 2022年1月8日; 录用日期: 2022年2月4日; 发布日期: 2022年2月14日

摘要

传统推荐算法由于缺乏对用户兴趣的精准捕捉, 会导致算法性能不佳等问题。随着深度学习在推荐系统领域应用的不断深入, 针对该问题有了许多很好的解决方法。为实现对用户动态兴趣的捕捉与行为的预测, 提出一种基于注意力机制的用户动态兴趣推荐算法(Attention Mechanical Recommendation Algorithm Based on Long Short-Term, AM_LST)。该算法通过潜在因子模型与门控循环单元, 分别实现对用户长短期兴趣的动态捕捉; 并使用注意力机制将用户长短期兴趣有机结合, 来改善传统推荐算法因用户兴趣的动态变化而导致推荐效果下降的问题。最终通过对比实验, 证明了本文提出的模型在性能上的提升。

关键词

推荐算法, 注意力机制, 潜在因子模型, 循环神经网络, 门控循环单元

User Dynamic Interest Recommendation Algorithm Based on Attention Mechanism

Xingyu Liu, Yinghua Xie

College of Information Science and Technology, Donghua University, Shanghai

Received: Jan. 8th, 2022; accepted: Feb. 4th, 2022; published: Feb. 14th, 2022

Abstract

Due to the lack of accurate capture of user interest, the traditional recommendation algorithm will result in poor performance. With the application of deep learning in the field of recommendation system, there are many good solutions to this problem. In order to realize the capture of user's dynamic interest and behavior prediction, a user's dynamic interest recommendation algorithm based on attention mechanism (Attention Mechanical Recommendation Algorithm Based on Long Short-Term, AM_LST) is proposed. In this algorithm, latent factor model and gated recurrent unit

are used to capture the users' long short-term interests. In addition, attention mechanism is used to organically combine users' long and short interests to improve the problem that the traditional recommendation algorithm's recommendation effect decreases due to the dynamic change of users' interests. Finally, the performance improvement of the proposed model is proved through comparative experiments.

Keywords

Recommendation Algorithm, Attention Mechanism, Latent Factor Model, Recurrent Neural Network, Gated Recurrent Unit

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

当前,越来越多的互联网场景开始使用推荐算法为用户定制个性化推荐界面,而推荐算法的准确率与用户对该应用的依赖度明显成正比关系,因此只有充分提高应用的推荐算法性能才能有效提高该应用的使用率,这样一方面利于用户检索匹配自身需求的信息,另一方面利于“商品”的有效推广。传统的推荐算法往往只单独关注了用户的长期或是短期兴趣,即根据用户的属性以及长期以来关注的更多的信息进行推荐,或是仅根据用户近几次会话进行即时推荐。而在实际应用中,用户的兴趣并不是固定不变的,而是会随着时间的推移动态变化,因此基于时间的序列推荐被提出,该类推荐致力于挖掘用户的动态兴趣,以实现对用户兴趣的更精准捕捉。早期解决序列推荐问题的方法是使用马尔可夫链[1] [2] (Markov Chain, MC)等。后来,循环神经网络(Recurrent Neural Network, RNN)、卷积神经网络(Convolutional Neural Network, CNN)等被成功地应用于序列推荐[3] [4]。再到后面注意力机制(Attention Mechanism, AM)在深度学习模型上的应用取得了巨大成果[5] [6] [7],也大大改善了序列推荐的预测能力。

注意力机制是源自人类视觉研究的一种信号处理机制。人类在关注某件事物时,其注意习惯往往具有选择性,并不会对该事物的各部分平均分配注意力,而是会选择性地将主要注意力投放至指定区域,而对其他区域分配较少或是不分配注意力。基于该现象提出的注意力机制能够有效优化模型的建模效果,提高模型的建模效益,其本质是让模型自己学习如何分配注意力,为输入信号加权,在强调感兴趣的区域同时,抑制不相关的背景区域,以突出重要特征对结果的影响。目前,注意力机制已被广泛地应用于深度学习的各个领域,并从 2017 年开始应用于推荐领域,AFM [8]、DIN [5]等模型中都引入了注意力机制。

本文提出一种利用注意力机制融合用户长短期兴趣的推荐算法——基于长短期的注意力机制推荐算法(Attention Mechanical Recommendation Algorithm Based on Long Short-Term, AM_LST),使用注意力机制对用户的长短期兴趣特征按照学习到的权重相融合,以实现用户长短期兴趣的自动捕获与长短期权重的自动分配,有效提高推荐效果。

2. 用户动态兴趣捕捉

本文在融合潜在因子模型与门控循环单元的长短期推荐算法(Recommendation Algorithm Based on Long Short-Term, RA_LST)的基础上引入注意力机制,以提高模型的推荐性能。RA_LST 模型分别通过潜

在因子模型(Latent Factor Model, LFM)与门控循环单元(Gated Recurrent Unit, GRU)来捕捉用户的长期兴趣与短期兴趣, 然后采用随机梯度优化(Stochastic Gradient Descent, SGD)的方式来将两部分学习结果融合, 以达到融合用户长短期兴趣的效果。

2.1. 潜在因子模型 LFM

潜在因子模型是一种潜在的语义分析技术, 其目的是发现潜在的主题或类别。它源于文本挖掘领域的研究, 近年来已被应用于推荐系统等领域[9]。例如, 在电影推荐系统中, 它可以根据用户的行为自动对推荐项目进行聚类, 即根据用户兴趣将推荐项目划分为不同的类别或主题, 然后给用户一个合理的电影推荐列表。

潜在因子模型是一种通过矩阵分解对评级矩阵进行处理, 然后输出特定用户的电影推荐列表的模型[10]。具体过程如下: 首先, 给定用户对电影的稀疏评级矩阵 \mathbf{R} (如表 1 所示)。然后, 对该评级矩阵进行矩阵分解, 得到用户 - 潜在因子矩阵 \mathbf{P} (如表 2 所示)和潜在因子 - 电影矩阵 \mathbf{Q} (如表 3 所示)。接着, 将矩阵 \mathbf{P} 和矩阵 \mathbf{Q} 相乘, 可以得到一个用户对电影的预测评分矩阵 $\hat{\mathbf{R}}$ (如表 4 所示)。它填补了原始稀疏评级矩阵中的空白元素, 填补的元素(黑体)即代表用户的预测电影评分, 最后我们可以由此得到对指定用户的推荐列表。

Table 1. User-movie sparse rating matrix \mathbf{R}

表 1. 用户 - 电影稀疏评分矩阵 \mathbf{R}

	电影 1	电影 2	电影 3	电影 4
用户 1	5	0	1	0
用户 2	0	3	0	0
用户 3	0	0	0	2

Table 2. User-latent factor matrix \mathbf{P}

表 2. 用户 - 潜在因子矩阵 \mathbf{P}

	潜在因子 1	潜在因子 2	潜在因子 3
用户 1	1.7647	1.0422	0.7777
用户 2	1.2321	1.0759	0.6078
用户 3	0.9873	1.0438	0.9718

Table 3. Latent factor-movie matrix \mathbf{Q}

表 3. 潜在因子 - 电影矩阵 \mathbf{Q}

	电影 1	电影 2	电影 3	电影 4
潜在因子 1	1.8941	1.3417	0.2862	0.6497
潜在因子 2	0.9913	0.8771	0.0297	0.4239
潜在因子 3	0.7840	0.6471	0.5931	0.9334

Table 4. User-movie prediction score matrix $\hat{\mathbf{R}}$

表 4. 用户 - 电影预测评分矩阵 $\hat{\mathbf{R}}$

	电影 1	电影 2	电影 3	电影 4
用户 1	4.9853	3.7850	0.9973	2.3142
用户 2	3.8767	2.9900	0.7451	1.8239
用户 3	3.6666	2.8690	0.8900	1.9910

以上的矩阵分解过程可以由式(1)表示:

$$R_{m \times n} \approx P_{m \times k} \times Q_{k \times n} = \hat{R}_{m \times n} \quad (1)$$

其中 k 代表分粒度, 即潜在因子数, 在本文中, 我们将 LFM 的分粒度设置为 10。

假设用户预测评分矩阵 \hat{R} 中的第 i 行 j 列元素为 \hat{r}_{ij} , 代表用户 i 对电影 j 的评分, 则其值为矩阵 P 第 i 行元素与矩阵 Q 第 j 列元素的乘积和。

$$\hat{r}_{ij} = p_i^T q_j = \sum_{k=1}^K p_{ik} q_{kj} \quad (2)$$

在得到预测评分矩阵 \hat{R} 后, 通过损失函数来衡量矩阵分解结果, 并在损失函数中加入正则 L2 范数, 以防止过拟合。

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 = \left(r_{ij} - \sum_{k=1}^K p_{ik} q_{kj} \right)^2 + \frac{\beta}{2} \sum_{k=1}^K (p_{ik}^2 + q_{kj}^2) \quad (3)$$

接着使用梯度下降法, 求解损失函数的负梯度, 以获得分解矩阵 P 和 Q 。

$$\frac{\partial}{\partial p_{ik}} e_{ij}^2 = -2 \left(r_{ij} - \sum_{k=1}^K p_{ik} q_{kj} \right) q_{kj} + \beta p_{ik} = -2e_{ij} q_{kj} + \beta p_{ik} \quad (4)$$

$$\frac{\partial}{\partial q_{kj}} e_{ij}^2 = -2 \left(r_{ij} - \sum_{k=1}^K p_{ik} q_{kj} \right) p_{ik} + \beta q_{kj} = -2e_{ij} p_{ik} + \beta q_{kj} \quad (5)$$

最后根据负梯度方向不断地更新变量, 通过迭代计算直至收敛, 算法结束。

$$p'_{ik} = p_{ik} - \alpha \left(\frac{\partial}{\partial p_{ik}} e_{ij}^2 + \beta p_{ik} \right) = p_{ik} + \alpha (2e_{ij} q_{kj} - \beta p_{ik}) \quad (6)$$

$$q'_{kj} = q_{kj} - \alpha \left(\frac{\partial}{\partial q_{kj}} e_{ij}^2 + \beta q_{kj} \right) = q_{kj} + \alpha (2e_{ij} p_{ik} - \beta q_{kj}) \quad (7)$$

2.2. 门控循环单元 GRU

门控循环单元是用于分析序列数据的标准循环神经网络(Recurrent Neural Network, RNN)最有效的改进版本之一[11]。它既解决了 RNN 的梯度爆炸与梯度消失问题, 又相比于长短期记忆(Long Short-Term Memory, LSTM)构造更精简, 训练速度更快。

如图 1 所示, 门控循环单元使用两个门: 重置门 r_t 与更新门 u_t 来决定信息是否被传递至输出。其中, 重置门 r_t 用来去除与当前时间节点 t 的输入数据预测无关的上一时间节点 $t-1$ 信息, 而更新门 u_t 则用来保留与当前时间节点 t 的输入数据预测有关的上一时间节点 $t-1$ 信息。除了重置门与更新门以外, 门控循环单元还由三个节点组成: 输入节点 x , 隐藏状态 h 和输出节点 y 。

重置门 r_t 与更新门 u_t 均由当前时间节点输入 x_t 与上一时间节点传递下来的隐藏状态 h_{t-1} 定义, 公式如式(8) (9)所示。

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (8)$$

$$u_t = \sigma(W_u x_t + U_u h_{t-1}) \quad (9)$$

重置门控制上一节点隐藏状态 h_{t-1} 对当前隐藏状态 \hat{h}_t 的贡献。

$$\hat{h}_t = \tanh(W x_t + U (r_t \cdot h_{t-1})) \quad (10)$$

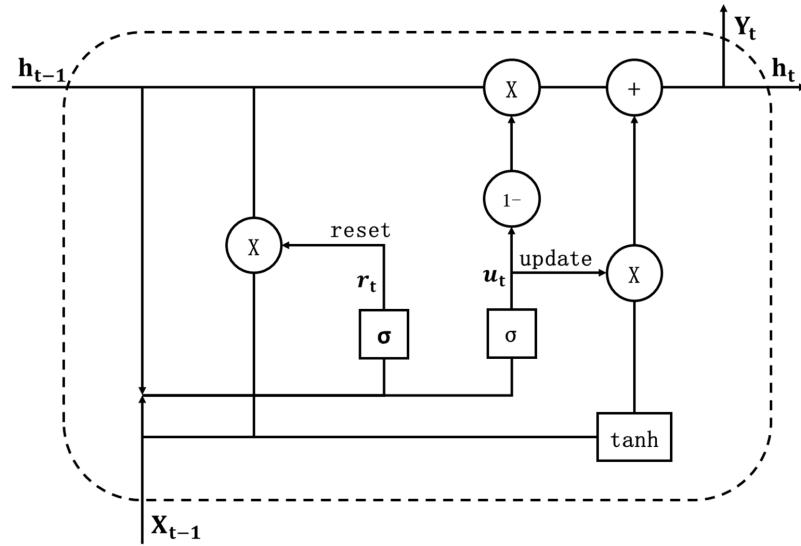


Figure 1. GRU architecture schematic diagram

图 1. GRU 架构原理图

更新门控制上一节点隐藏状态 h_{t-1} 与当前隐藏状态 \hat{h}_t 的混合比重, 对原本的隐藏状态 h_{t-1} 选择性遗忘, 对包含当前节点信息的隐藏状态 \hat{h}_t 选择性记忆, 最终组成传递给下一节点的隐藏状态 h_t 。

$$h_t = (1 - z_t) h_{t-1} + z_t \hat{h}_t \quad (11)$$

在使用 GRU 进行批量训练时需要统一输入数据的长度, 但由于每个用户的交互项数目有所不同, 导致每个用户的输入数据长度不一, 从个位数到上千不等。该问题通常的解决方法是将其裁剪成相对较短的等长片段, 但在序列问题中, 该方法会削减时间特征, 因此我们使用基于会话序列推荐的循环神经网络(Gated Recurrent Unit for Recurrent, GRU4Rec)中的并行小批量方法来进行训练[12]。如图 2 所示, GRU4Rec 输入三个并行的 t 时刻交互项目, 模型预测序列 $t+1$ 时刻的交互项目, 并在一个用户的序列结束后自动补充未训练的用户序列, 该方法既可以保证输入序列的连续性, 又可以保证模型输入长度的一致性, 最大限度地提高了并行训练效率。

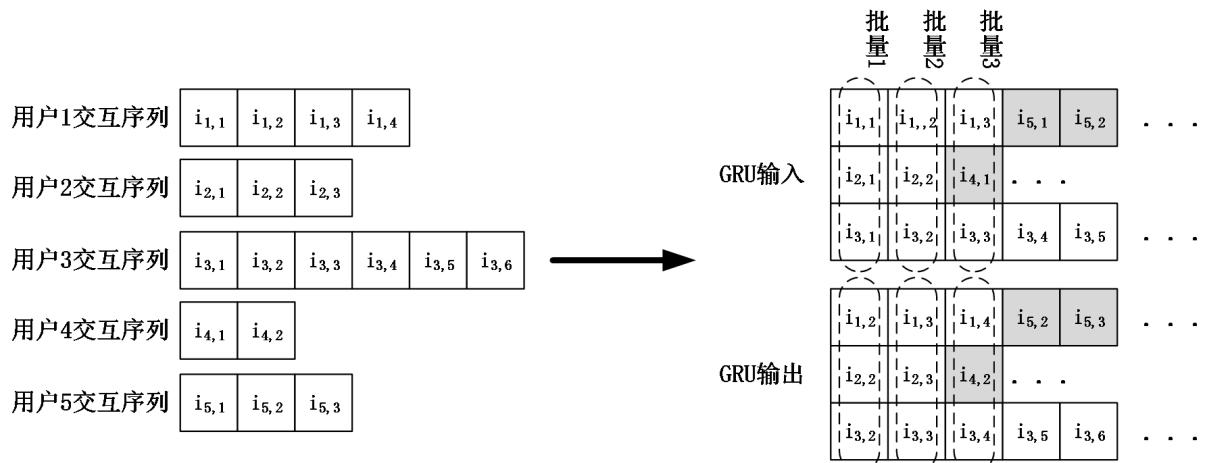


Figure 2. GRU4Rec Batch Training

图 2. GRU4Rec 批量训练

3. 基于注意力机制的推荐算法

通过 GRU 与 LFM 两个模型的学习, 我们分别提取到了用户的长期兴趣与短期兴趣这两项特征评分, 维度为参评电影个数。其中, 用户的长期兴趣源自用户的属性或是长期以来保持不变的兴趣状态, 例如在用户甲的全部交互项中, 悬疑类型的电影占有很大比重, 那么可以表明用户甲对悬疑类电影表现出浓厚的兴趣, 代表着其长期兴趣; 而用户的短期兴趣往往出现在用户的短期会话当中, 而我们要提取的是时间最近的一段短期会话, 来代表用户的短期兴趣, 该类特征通常持续时间较为短暂, 具有阶段性, 例如用户乙在不同的时间阶段会集中观看不同演员主演的电影, 表现出对不同电影演员的喜爱, 代表着该用户的短期兴趣。

接下来, 我们使用注意力机制, 通过长短期两类兴趣特征之间的内在关系, 自动学习它们各自对推荐结果的重要性, 来获得长短期兴趣特征评分的相应权重, 即注意力值(Attention 值), 以刻画长期兴趣与短期兴趣之间的权重关系, 判断两种特征的参考重要性, 然后对其进行加权平均, 获得最终预测评分。

3.1. 注意力机制

注意力机制模型的训练过程即为注意力分配过程, 它相当于一个加权过程, 包括注意力权重学习与加权平均两个部分。

在进行训练前, 首先要将特征评分输入到嵌入层, 得到嵌入向量, 再将其作为注意力机制模型的输入, 此时模型输入的形状为(电影数量 \times 特征评分项 \times 嵌入维度)。

本文的注意力部分是注意力机制中的乘性注意力机制[13], 公式定义如下。

$$a'_i = h^T \text{Relu}(Wx_i + b) \quad (12)$$

$$a_i = \frac{\exp(a'_i)}{\sum_i \exp(a'_i)} \quad (13)$$

a_i 为经过注意力模型训练后得到的长短期偏好注意力权重, 即注意力得分, 使用各项权重参数表示, 其中包括特征交叉层到注意力网络全连接层的权重矩阵 W , 偏置向量 b , 全连接层到 softmax 输出层的权重向量 h 。 x_i 为用户长期与短期偏好特征评分。

最后将注意力权重 a_i 与特征评分 x_i 相乘, 再经过一个输出大小为 1 的全连接层即可得到最终的输出——预测评分 \hat{y} 。

$$\hat{y} = P^T \sum_i a_i x_i \quad (14)$$

在以上的注意力模型中, 我们使用单层全连接网络进行参数学习, 以防止特征交叉数据稀疏导致权重参数难以收敛的问题。

3.2. 注意力对长短期兴趣的融合

本文提出的模型是在“长短期特征评分预测层”与“输出层”之间引入注意力网络, 通过学习, 对长期与短期两项兴趣特征评分结果赋予一个权重, 来表示对不同特征的关注度。注意力网络层通过学习输入特征向量各维度之间的内在关系来给予各维度适当的权重[14], 以实现用户的长期兴趣特征与短期兴趣特征的融合, 更精准地表征用户动态兴趣, 更精确地为用户推荐项目。根据以上, 在本文提出的算法中, 输入特征向量分为长期兴趣与短期兴趣这两个维度, 算法流程如图 3 所示。

在输入层将用户稀疏评分矩阵输入到模型中, 在模型的神经网络层分别通过 LFM 与 GRU 的长短期兴趣预测, 得到用户长期兴趣预测评分矩阵与用户短期兴趣预测评分矩阵。

在长期兴趣预测 LFM 部分, 首先通过高斯分布随机生成初始化用户与电影两个矩阵, 然后采用随机梯度下降迭代优化, 直至生成最优的用户与电影潜在因子矩阵, 进行内积以获得 LFM 部分的预测评分矩阵。

在短期兴趣预测 GRU 部分, 首先对输入进行 one-hot 编码(用户交互项置为 1, 未交互项置为 0), 然后在嵌入层使用 \tanh 激活函数对其做嵌入处理, 优化其编码程度, 最后通过 GRU 层获得 GRU 部分的预测评分矩阵。

在注意力网络层, 将以上两部分的预测评分矩阵输入到注意力网络中, 分别学习长短期兴趣的权重 α 与 β 。然后使用学习到的 α 与 β 权重进行加权求和, 将长短期兴趣预测评分矩阵合并为最终的预测评分矩阵, 即用户动态兴趣预测评分矩阵, 将其在输出层输出。

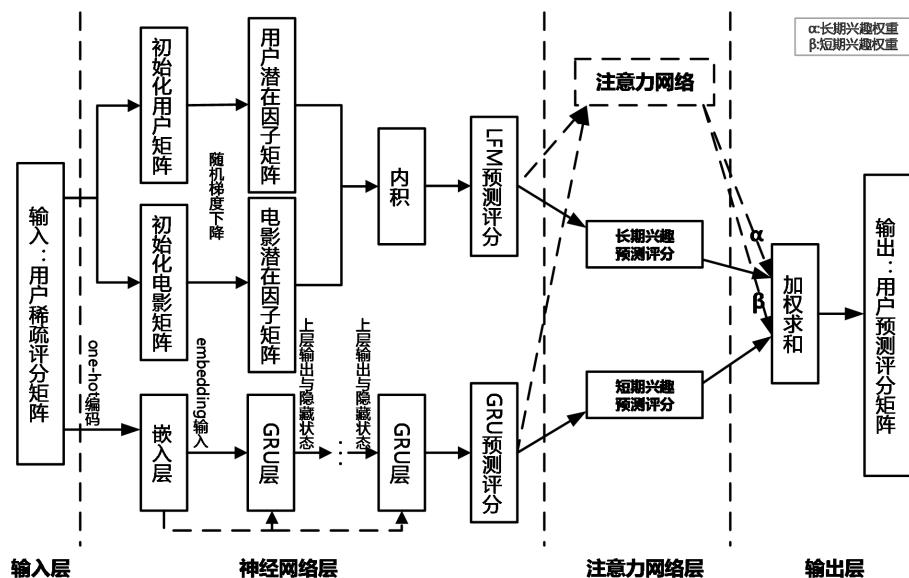


Figure 3. Flow chart of AM_LST algorithm

图 3. AM_LST 算法流程图

以上的注意力网络部分由可视化工具输出如图 4 所示, 该结构图即为 3.1 注意力机制理论部分的模型实现。在通过嵌入层得到嵌入向量后, 再依次通过注意力连接层与预测连接层获得注意力权重, 最后通过全连接层, 按照注意力权重对长短期特征评分进行加权求和, 得到最终的预测评分输出。

4. 实验分析

4.1. 实验数据集

本文提出的模型仿真基于 MovieLens 与 Netflix 两个开源数据集。MovieLens 原始数据集是推荐系统的经典测试数据集, 本文所使用的是 1 M 版本, 包含 6000 多用户对 3000 多电影的评级数据。Netflix 原始数据集作为推荐系统的另一个常用数据集, 包含 48 万用户对 1.7 万电影的评分数据, 数据信息包括用户 ID、电影 ID、用户评分、评价时间等相关信息。由于两个数据集的评分信息均带有时间戳, 因此适用于本模型等基于时间序列的算法评估。

在数据集的划分上, 本文选择使用留一法(Leave-One-Out), 对每个用户的交互项按时间顺序升序排列, 将前 90% 交互项作为训练数据, 后 10% 交互项作为测试数据[15]。最终获得分别以用户 ID 与电影 ID 作为行索引与列索引的评分表格。根据以上模型需求, 对 MovieLens 与 Netflix 两个数据集进行预处理, 得到如表 5 所示的实验数据集。

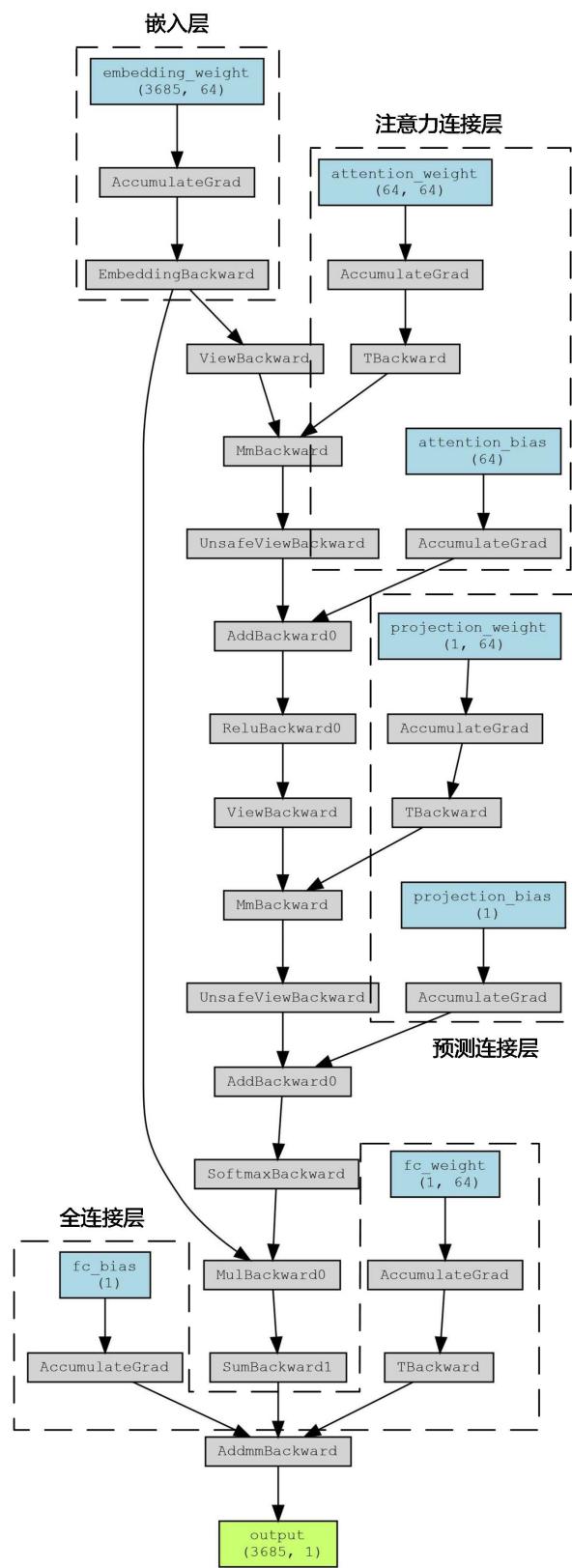


Figure 4. Structure diagram of attention network model
图 4. 注意力网络模型结构图

Table 5. Experimental data set
表 5. 实验数据集

	训练集用户	训练集电影	训练集实例
MovieLens	6040	3685	902826
Netflix	8551	6711	896864
	测试集用户	测试集电影	测试集实例
MovieLens	6040	3416	97357
Netflix	8551	4922	95193

4.2. 实验评估指标

为了评估模型的推荐性能, 本文采用归一化折损累计增益(Normalized Discounted Cumulative Gain, NDCG)与均方根误差(Root Mean Square Error, RMSE)作为实验评估指标。其中, NDCG 评价推荐列表与真实交互列表之间的误差, 而 RMSE 衡量用户预测评分与用户真实评分之间的误差。

归一化折损累计增益, 顾名思义是在累计增益之后对其进行折损, 再进行归一化处理所得到的评估结果, 分数越高表示性能越好。其中, 增益 *Gain* 等于相关性分数, 定义为式(15)。

$$Gain = rel(i) = 2^i - 1 \quad (15)$$

当预测结果在测试集时, r_i 的值取 1; 当预测结果不在测试集, r_i 的值取 0。

累计增益 CG_k 即对增益求和。

$$CG_k = \sum_{i=1}^k rel(i) \quad (16)$$

接着考虑排序因素, 对累计增益进行折损得到折损累计增益 DCG_k , 以增加排名靠前项目的影响, 减小排名靠后项目的影响。

$$DCG_k = \sum_{i=1}^k \frac{rel(i)}{\log_2(i+1)} \quad (17)$$

最后进行归一化后得到的归一化折损累计增益 $NDCG$ 。

$$NDCG = \frac{DCG}{IDCG} \quad (18)$$

其中 $IDCG$ 为根据 $rel(i)$ 降序排列得到最好的 DCG , 即理想折损累计增益(ideal DCG)。

均方根误差是均方误差(Mean Square Error, MSE)的平方根, 分数越低代表性能越好。均方误差 MSE 是预测值与真实值之间差值的平方和平均值, 公式定义为式(19)。

$$MSE = \frac{1}{N} \sum_i^N (\hat{y} - y)^2 \quad (19)$$

将均方误差 MSE 开方即可得均方根误差 $RMSE$:

$$RMSE = \sqrt{\frac{1}{N} \sum_i^N (\hat{y} - y)^2} \quad (20)$$

4.3. 实验结果

本文提出的模型仿真在 PyTorch 深度学习框架下实现, 模型训练的嵌入维度设置为 64。我们将

AM_LST 的仿真结果与 LFM、GRU4Rec、RA_LST 三种用于稀疏数据预测的同类竞争方法进行对比实验。并且为了验证模型性能的普遍性, 实验分别在 MovieLens(ML)与 Netflix(NF)两个数据集上进行, 结果如表 6 所示。

Table 6. Comparison of simulation results of each model

表 6. 各模型仿真结果对比

	LFM	GRU4Rec	RA_LST	AM_LST
ML_NDCG@10	0.3669	0.4346	0.4501	0.4708
ML_RMSE	0.3029	0.6352	0.1867	0.1064
NF_NDCG@10	0.3635	0.4596	0.4732	0.4939
NF_RMSE	0.3068	0.6450	0.2889	0.1714

表 6 总结了每种方法在不同次的迭代(epoch)训练下获得的平均性能, 可以看到, 本文提出的基于注意力机制的用户动态兴趣推荐算法 AM_LST 在 NDCG、RMSE 两个指标上均表现出一定优势。在 MovieLens 数据集中, AM_LST 的平均 NDCG 比 RA_LST 提高了 0.02, 平均 RMSE 降低了 0.08; 在 Netflix 数据集中, AM_LST 的 NDCG 比 RA_LST 提高了 0.02, RMSE 降低了 0.11。

该实验结果充分证明了本文提出的 AM_LST 模型相较于单一考虑用户长期或短期兴趣的模型 LFM 与 GRU 在推荐性能上的提高, 一定程度上解决了一般推荐算法对用户兴趣的单一考虑、缺乏对兴趣的动态捕捉以及出现兴趣漂移等问题。

另外, 相比于引入注意力机制前的模型 RA_LST, 本文提出的 AM_LST 模型在提高预测排名的准确性与降低预测分数的误差方面均表现出色, 表明该模型中的注意力机制起到了一定的作用, 能够有效地将用户的长短期兴趣融合, 以达到对用户兴趣精准捕捉的目的。

图 5 对比了 AM_LST 与 LFM、GRU4Rec、RA_LST 四种模型在 MovieLens 数据集中随 epoch 变化的排名评估与测试误差曲线图(NDCG\RMSE)。可以看到, AM_LST 在不同 epoch 下的 NDCG 均高于其他三种模型, 可以更好地拟合推荐排名, 并且 RMSE 也均低于其他三种模型, 预测结果更加准确。

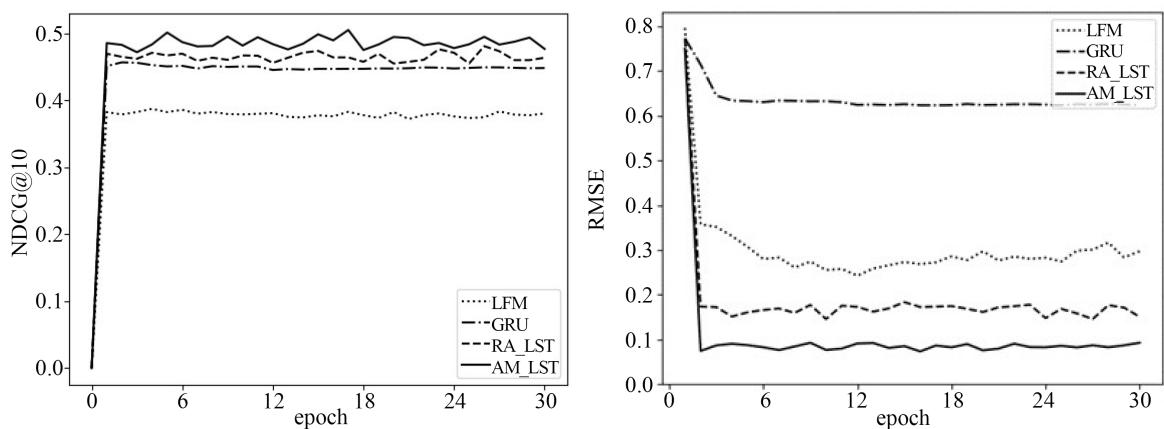


Figure 5. Comparison diagram of test errors index of each model

图 5. 各模型测试误差指标对比图

5. 结束语

本文提出一种基于时间序列的推荐算法, 使用注意力机制将通过潜在因子模型与门控循环单元模型

提取到的用户长短期兴趣进行融合, 以更精准地分析用户的动态兴趣, 提高推荐模型的推荐性能。通过与同类模型的对比实验表明, 该 RA_LST 模型在 MovieLens 与 Netflix 两个数据集上都能够有效地提高预测排名的准确性, 降低预测分数的误差。

参考文献

- [1] Rendle, S., Freudenthaler, C. and Schmidt-Thieme, L. (2010) Factorizing Personalized Markov Chains for Next-Basket Recommendation. *Proceedings of the 19th International Conference on World Wide Web*, Raleigh, 26-30 April 2010, 811-820. <https://doi.org/10.1145/1772690.1772773>
- [2] He, R. and McAuley, J. (2016) Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Barcelona, 12-15 December 2016, 191-200. <https://doi.org/10.1109/ICDM.2016.0030>
- [3] Xu, C., Zhao, P., Liu, Y., Xu, J., Sheng, S., Cui, Z., et al. (2019) Recurrent Convolutional Neural Network for Sequential Recommendation. *The World Wide Web Conference*, San Francisco, 13-17 May 2019, 3398-3404. <https://doi.org/10.1145/3308558.3313408>
- [4] Tang, J. and Wang, K. (2018) Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, Marina del Rey, 5-9 February 2018, 565-573. <https://doi.org/10.1145/3159652.3159656>
- [5] Zhou, G., Zhu, X., Song, C., Zhu, H., Ma, X., Yan, Y., et al. (2018) Deep Interest Network for Click-Through Rate Prediction. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, 19-23 August 2018, 1059-1068. <https://doi.org/10.1145/3219819.3219823>
- [6] Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., et al. (2019) Deep Interest Evolution Network for Click-Through Rate Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 5941-5948.
- [7] Feng, Y., Lv, F., Shen, W., Wang, M., Sun, F., Zhu, Y., et al. (2019) Deep Session Interest Network for Click-Through Rate Prediction. *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, Macao (China), 10-16 August 2019, 2301-2307. <https://doi.org/10.24963/ijcai.2019/319>
- [8] Xiao, J., Ye, H., He, X., Zhang, H., Wu, F. and Chua, T.-S. (2017) Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Melbourne, 19-25 August 2017, 3119-3125. <https://doi.org/10.24963/ijcai.2017/435>
- [9] 王升升, 赵海燕, 陈庆奎, 曹健. 个性化推荐中的隐语义模型[J]. 小型微型计算机系统, 2016, 37(5): 881-889.
- [10] Liu, W., Wang, B. and Wang, D. (2018) Improved Latent Factor Model in Movie Recommendation System. *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)*, Singapore, 1-3 March 2018, 101-104. <https://doi.org/10.1109/ICoIAS.2018.8494074>
- [11] Hidasi, B., Karatzoglou, A., Baltrunas, L. and Tikk, D. (2015) Session-Based Recommendations with Recurrent Neural Networks.
- [12] Chen, J. and Abdul, A. (2019) A Session-Based Customer Preference Learning Method by Using the Gated Recurrent Units with Attention Function. *IEEE Access*, 7, 17750-17759. <https://doi.org/10.1109/ACCESS.2019.2895647>
- [13] Cui, R., Wang, J. and Wang, Z. (2021) Multiplicative Attention Mechanism for Multi-Horizon Time Series Forecasting. *2021 International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, 18-22 July 2021, 1-6. <https://doi.org/10.1109/IJCNN52387.2021.9533598>
- [14] 谢恩宁, 何灵敏, 王修晖. 基于注意力机制的深度协同过滤模型[J]. 中国计量大学学报, 2019, 30(2): 219-225+242.
- [15] He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T.-S. (2017) Neural Collaborative Filtering. *Proceedings of the 26th International Conference on World Wide Web*, Perth, 3-7 April 2017, 173-182. <https://doi.org/10.1145/3038912.3052569>