

文章编号: 1003-0077(2020)05-0097-09

一种基于 CW-RNN 的多时间尺度序列建模推荐算法

袁 涛^{1,2}, 牛树梓², 李会元²

- (1. 中国科学院大学, 北京 100049;
2. 中国科学院 软件研究所, 北京 100190)

摘 要: 序列化推荐试图利用用户与物品的历史交互序列, 预测下次即将交互的物品。针对序列化推荐中推荐物品依赖于用户的长时间全局兴趣、中时间兴趣还是短时间局部兴趣的不确定性, 该文提出了一种基于 CW-RNN 的多时间尺度序列建模推荐算法。首先, 该算法引入 CW-RNN 层, 从用户与物品的历史交互序列中抽取多个时间尺度的用户兴趣特征。然后, 通过尺度维卷积来建模对不同时间尺度的用户兴趣特征的依赖, 生成多时间尺度用户兴趣特征的统一表示。最后, 利用全连接层建模统一的多尺度用户兴趣特征和隐式物品特征的交互关系。在 MovieLens-1M 和 Amazon Movies and TV 两个公开数据集上的实验结果表明, 相比于现有最优的序列推荐算法, 该文提出的算法在准确率上分别提升了 3.80% 和 8.63%。

关键词: 序列推荐; 多时间尺度; 动态建模
中图分类号: TP391 **文献标识码:** A

A Multi-Scale Temporal Dynamic Model for Sequential Recommendation with Clockwork RNN

YUAN Tao^{1,2}, NIU Shuzi², LI Huiyuan²

- (1. University of Chinese Academy of Sciences, Beijing 100049, China;
2. Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Sequential recommendation attempts to use the historical interaction sequence between users and items to predict the next item to interact with. A multi-scale temporal dynamic model for sequential recommendation with Clockwork RNN is proposed to solve the uncertainty of recommended items by the on user's long-term global interest, medium time interest or short time local interest. Firstly, the CW-RNN layer is introduced to extract user's multi-scale temporal interest features from the historical interaction sequence between users and items. The convolution with CNN on the time scale dimension is then used to learn the user's interest dependency on different time scales, and generate the user's unified interest representations. Finally, it uses the fully connected layer to model the interaction between the unified multi-scale user interest representations and item's embedding representations. Experiments are carried out on MovieLens-1M and Amazon Movies and TV, two public datasets. The results show that our proposed model improves the accuracy by 3.80% and 8.63% respectively compared with the current optimal sequential recommendation algorithms.

Keywords: sequential recommendation; multi-temporal scale; dynamic model

0 引言

随着信息技术和互联网的发展, 人们逐渐从信息匮乏的时代走入了信息过载 (information over-

load) 的时代^[1]。推荐系统, 作为一种信息过滤的重要技术和手段, 被认为是当前可解决信息超载问题的非常有效的工具。目前大型的电子商务网站, 包括 Amazon、eBay、淘宝、京东等, 均采用了推荐系统作为创收的重要手段。各种提供个性化服务的多媒

体信息网站,诸如今日头条、网易新闻、爱奇艺、豆瓣等,均不同程度地采用了各种形式的推荐技术来为用户推荐信息物品^[2]。

与传统的推荐不同,序列化推荐将用户与物品的交互历史视为序列数据,预测下一次用户会与哪些物品交互。根据用户与物品的交互历史序列的利用尺度不同,典型的序列化推荐算法可以分为长期依赖(如 GRU4REC^[3])、短期依赖(如 FPMC^[4], STAMP^[5]等)、长短期混合依赖算法(如 HRM^[6]等)等。

基于深度学习的推荐算法成为目前的研究热点。其中,典型的代表是基于循环神经网络 RNN (recurrent neural network) 的算法。传统的 RNN 采用随时间反向传播算法(back propagation through time, BPTT)来进行梯度求导,会导致梯度消失(vanishing gradients)问题,而梯度消失的直接后果是无法获取过去的长时间依赖信息^[7]。为了解决这个问题,长短期记忆网络(long short term memory, LSTM)与门控循环单元(gated recurrent unit, GRU)等相继提出。LSTM 是对传统 RNN 的改进,通过引入 Cell State 来保留过去的记忆信息,避免 BPTT 导致的梯度消失问题。但在实际推荐应用中,对于下次交互物品的预测真正起作用的并不都是非长即短的依赖,也有可能是处于长短之间的依赖。

本文以电商平台上某用户购买序列...A...B...A...C...B...为例,来看这种非长即短的依赖建模是否合适。其中,A、B、C 是不同品牌的同一种物品。该用户于 t_1 时刻购买了 A 品牌的物品,用完之后于 t_2 时刻购买了 B 品牌的同一物品,发现 B 不如 A 好用,于是又继续于 t_3 时刻购买了 A 品牌的。用完之后于 t_4 时刻购买 C 品牌的同一物品,发现 C 没有 A 与 B 品牌的物品好用。因此不等 C 用完便又来购买该物品,发现没有 A 品牌的物品,便于 t_5 时刻购买了 B 品牌的物品。对于该用户的历史购买序列 $A \rightarrow B \rightarrow A \rightarrow C \rightarrow B$ 来说,A 是该用户长期购买过的品牌,B 是中间曾经购买过的品牌,C 是短期购买过的品牌,要预测用户在 t_5 时刻购买的物品则是用户在品牌 A、B 和 C 三者之间不断比较的结果。其既不完全依赖于长期因素,也不完全依赖于短期因素,而是依赖于长期、短期、中期因素等多个时期的因素的复杂交互。因此,如何建模多尺度依赖成为一个重要的挑战。

为了解决序列化推荐中的多尺度依赖问题,

本文提出了一种基于 CW-RNN 的多时间尺度序列建模推荐算法 multi-scale CW-RNN,简记为 MC-RNN。MC-RNN 算法首先引入带时钟频率的循环神经元层(即 CW-RNN 层),通过对隐层神经元的不同分区更新频率不同,产生用户的多个时间尺度的兴趣表示。然后,通过 MC-RNN 的尺度维卷积层,建模对不同时间尺度的用户兴趣特征表示的依赖,生成多尺度的用户兴趣特征的统一表示。最后,MC-RNN 采用全连接层建模统一的多尺度用户兴趣特征表示与物品的隐式特征之间的关系,通过 softmax 函数预测用户对物品的喜好程度。

本文在 MovieLens-1M 和 Amazon Movies and TV 两个公开数据集上进行实验。实验结果表明,相比于现有最优的序列推荐算法,本文提出的 MC-RNN 算法在准确率上分别提升了 3.80% 和 8.63%。

1 相关研究工作

目前,传统的推荐算法主要关注用户—物品的二元矩阵,并对它们的二元关系进行建模。在这种情况下往往只能利用用户对物品有限的显性反馈数据,而用户对物品的隐性反馈行为以及行为的时效性往往会被忽略。而序列化推荐算法则不同,其将用户与物品的交互历史视为时间序列数据,预测下一次用户会与哪些物品交互。根据用户与物品的交互历史序列的利用尺度不同,序列化推荐算法大致可以分为三种:长期依赖推荐算法、短期依赖推荐算法和长短期混合依赖推荐算法。

1.1 长期依赖推荐算法

长期依赖推荐算法通过建模用户对物品的交互历史全序列,尽可能地考虑用户行为序列的全局信息,因而能够捕获到用户的长期偏好。

Hidasi 等人^[3]在基于会话的推荐系统中提出了基于循环神经网络的序列推荐(GRU4REC)模型。该模型采用 GRU 捕捉会话中用户点击行为之间的依赖关系。每一个时间步,模型的输入是当前用户点击物品的独热编码,然后通过一个嵌入层压缩为一个低维特征,中间是多个基本 GRU 层,最后是一个前向层,输出层利用 softmax 方法计算每个物品用户点击的概率。Wu 等人^[8]针对 Top-N 推荐提出了协同去噪自动编码器推荐(CDAE)模型。该模型通过使用去噪自动编码器形成用户物品反馈

数据,学习用户和物品的分布式表示。类似相关的研究工作还有,He 等人^[9]提出了神经网络协同过滤(NeuCF)模型。Travis 等人^[10]提出了协同记忆网络(CMN)模型。这些研究工作利用神经网络模型能较好地捕获用户与物品交互的深层全局兴趣,取得了较好的推荐效果。

1.2 短期依赖推荐算法

短期依赖推荐算法考虑到用户偏好的时间敏感性的特点,认为与用户下一次行为的时间间隔越近,对用户的行为影响越大,因而更关注用户近期的行为序列,能够很好地捕获到用户的短期偏好。

Liu 等人^[5]在基于会话的推荐系统中提出了基于记忆优先级的序列推荐(STAMP)模型。从常理来推断,和用户当前希望浏览的商品最相关的是用户上一个浏览的商品,并且不同的时间浏览过商品的重要性不同,该模型将最后一个浏览的商品重点考虑,同时将之前浏览过的所有商品用 attention 建模。对全局商品用 attention 建模得到一个全局表达,输入多层感知机(MLP),然后将最后一个商品直接输入多层感知机建模得到最后一个商品的表达,最后将这两个表达组合起来,用 softmax 来预测未来可能会点击的 n 个商品。类似相关的研究工作还有,Feng 等人^[11]在下一个购物篮推荐系统中提出了动态循环序列推荐(DREAM)模型。Massimo 等人^[12]在基于会话的推荐系统中提出了层次循环神经网络序列推荐(HRNN)模型。Steffen 等人^[4]提出了一种分解个性化马尔科夫链模型(factorizing personalized markov chain, FPMC)。这些研究工作能够较好地捕获用户的局部兴趣,在短时推荐领域取得了较好的推荐效果。

1.3 长短期混合依赖推荐算法

长短期混合依赖推荐算法将用户的长期依赖和短期依赖结合起来,利用深度神经网络同时建模用户的长期兴趣和短期兴趣。该模型既能对用户的短期时序行为建模,也能捕获用户的长期偏好,因此和单一尺度的依赖推荐算法相比,其具有更好的推荐效果。

Wang 等人^[6]提出了一种层次表达模型(hierarchical representation model, HRM),很好地借鉴

了当前自然语言处理领域中词向量的思想,使用了一种深度网络结构训练用户和商品的特征向量表达,把用户和商品表达成一个连续的低维向量,来进行预测用户在下一个时刻对商品是否发生购买行为,在预测时综合考虑了用户的交易记录中的所有商品和用户的偏好向量,能够捕获长期偏好和短期序列行为。Zhao 等人^[13]在电影推荐系统中提出了结合长短期信息的序列推荐(LSIC)模型,该模型采用了生成对抗网络(GAN)框架,将 MF 矩阵分解和 RNN 循环神经网络的模型融合,同时捕获用户长期偏好和短期会话信息,从而最大限度地提高推荐系统的最终性能。类似相关的研究工作还有 Lu 等人^[14]在购物篮推荐系统中提出了同时建模长短期用户偏好的多阶注意力排序序列推荐(MARank)模型。

2 多时间尺度序列推荐算法 MC-RNN

序列化推荐算法用交互历史序列预测即将发生的交互行为。已有的基于深度学习的序列推荐算法都是建模对历史序列的长期、短期依赖。针对前言中提到的例子,这种粗粒度的建模方式并不能抓住真实的依赖。因此本文提出从多个层次建模预测目标对历史序列的依赖问题。为了解决这个问题,本文提出多尺度序列推荐算法(multi-scale clockwork RNN),简记为 MC-RNN。

2.1 形式化定义

假设用户集合 $U = \{u_1, u_2, \dots, u_m\}$, 物品集合 $I = \{i_1, i_2, \dots, i_n\}$, 其中 $|U| = m$, $|I| = n$ 。用户 u 在 t 时刻之前交互的物品序列记为 $L_u^{<t} = I_u^1, I_u^2, \dots, I_u^{t-1}$, 其中, $I_u^{t-1} \subseteq I$ 表示物品集合的子集。给定所有用户交互物品序列集合 $\{L_u^{<t}\}_{u \in U}$, 序列推荐任务的目标在于预测用户 u 在下一个时刻 t 会交互的物品。

2.2 MC-RNN 体系结构

MC-RNN 模型的体系结构如图 1 所示,主要由以下三个部分构成:①带时钟频率的循环神经元层(clockwork recurrent neural networks):抽取多个时间尺度的用户兴趣表示;②尺度维的卷积层:建模多个时间尺度之间的关系,获取多尺度用户兴趣的统一表示;③全连接层:建模统一的多尺度用户兴趣表示与物品的隐式表示的交互关系。

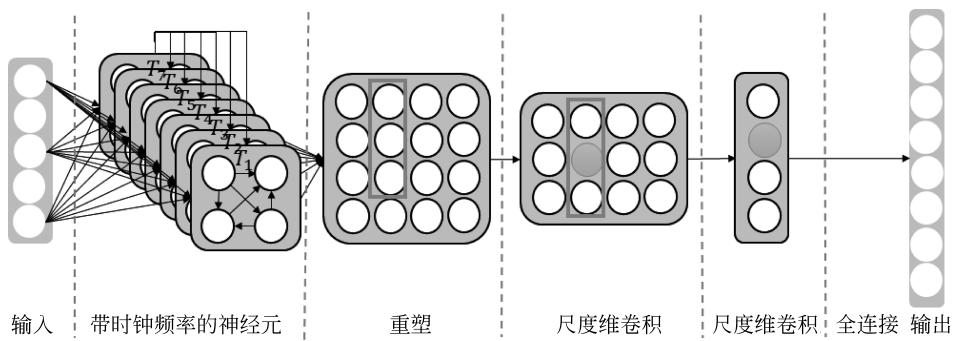


图 1 MC-RNN 体系结构

2.3 带时钟频率的循环神经元

带时钟频率的循环神经网络 clockwork RNN (CW-RNN)^[15] 将隐藏层神经元节点分为 g 个大小为 p 的组, 每个组都分配了一个时钟周期 $T_k, k = 1, 2, \dots, g$. 时钟周期可以采用线性 ($T_k = k$)、指数 ($T_k = 2^{k-1}$) 或斐波拉杰数列 ($T_k = T_{k-1} + T_{k-2}, T_2 = 2, T_1 = 1$). 该网络中每一个 T_k 内部的节点都是全连接的, 若周期 T_k 小于 T_l , 则从组 l 到组 k 存在循环连接。循环连接从右到左, 从慢向快传播。

仅当 $t \% T_k = 0$ 时, 对 $h_t^u(k)$ 中对应的部分用式(1)进行更新; 否则, $h_t^u(k)$ 中对应的部分保留原值。

$$h_t^u(k) = f_H(W_H(k) \cdot h_{t-1}^u(k) + W_I(k) \cdot I_t^u) \quad (1)$$

周期较大的节点组, 更新频率低, 能够保留长期的依赖信息。周期较小的节点组, 更新频率高, 能够捕捉局部的高频信息。依据更新频率不同, 该层的输出 $\{o_t^u(k)\}_{k=1}^g$, 如式(2)所示, 表征了用户在不同时间尺度的兴趣表示。

$$o_t^u(k) = f_o(W_o(k) \cdot h_t^u(k)) \quad (2)$$

2.4 尺度维卷积

如何由多个不同时间尺度的用户兴趣表示 $\{o_t^u(k)\}_{k=1}^g$ 得到统一的用户兴趣表示 o_t^u ? 一个直观的做法是线性加权: $o_t^u = \sum_{k=1}^g W_D(k) \cdot o_t^u(k)$ 。但是线性加权模型是否能完全建模不同时间尺度的兴趣表示之间的关系? 实验结果表明, 线性加权模型并不能。如前所述, 更新周期大的节点组, 捕捉到的是长期(全局)特征; 更新周期小的节点组, 捕捉到的是短期(局部)特征。通过更新的过程可以发现, 这些不同尺度的特征之间形成某种包含关系。线性加权模型并不适合建模这种关系。

本文将多个时间尺度的 $1 \times p$ 维的特征向量

$\{o_t^u(k)\}_{k=1}^g$ 重塑成一个大小为 $g \times p$ 的矩阵 M_t^u , 如式(3)所示, 从而这种时间维的特征关系转换为空间的层次关系。

$$M_t^u = (M_t^u(i, j))_{g \times p}, M_t^u(i, \bullet) = o_t^u(i) \quad (3)$$

而卷积操作比较擅长提取空间中的从简单到复杂、从局部到全局的特征。因此, MC-RNN 引入尺度维的卷积层来抽取这种层次之间的关系。

假设卷积层 $1, 2, \dots, L$, 则抽取到的层次关系如式(4)所示。

$$o_t^u = W_D^L \otimes \dots \otimes W_D^1 \otimes M_t^u \quad (4)$$

其中, \otimes 表示卷积操作。

第 1 层卷积核为 W_D^1 , 大小为 2×1 卷积输入核, 实现相邻尺度间特征的关系的抽取, 将尺度特征矩阵压缩为大小为 $(g-1) \times p$ 的矩阵。第 2 层卷积核为 W_D^2 , 大小为 2×1 , 将尺度特征矩阵进一步压缩为大小为 $(g-2) \times p$ 的矩阵, 如图 2 所示。依此类推, 最后一层输出为大小为 $1 \times p$ 的向量, 记为 o_t^u , 作为统一的用户多尺度兴趣的表示。

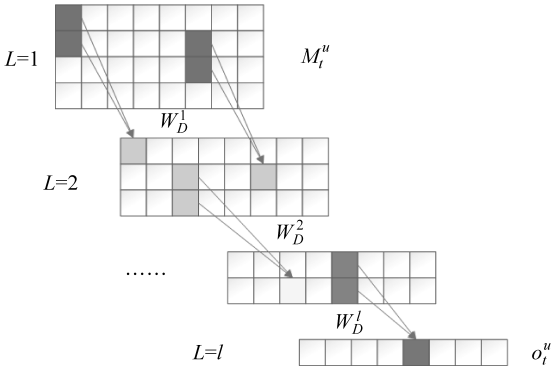


图 2 逐层卷积方式

2.5 全连接层

本文将下一个物品预测问题归约为分类的任务, 因此输出层的神经元个数为 n , 与物品个数相同。本文用全连接网络建模物品与用户之间的交互

关系,其中隐层输出 o_i^u 代表统一的用户多尺度兴趣的表示,那么隐层与输出层之间的连接权重可以用来作为物品的隐式特征表示记为 E ,大小为 $n \times p$ 。该全连接层的激活函数为 softmax,用于预测用户下一个即将购买或浏览的物品 \hat{I}_u^t ,如式(5)所示。

$$\hat{I}_u^t = \text{softmax}(E \cdot o_i^u + b) \quad (5)$$

2.6 优化算法

从多分类的角度来看,本文采用交叉熵作为损失函数,如式(6)所示,其中 I_u^t 表示用户 u 在 t 时刻真实交互的物品集合, λ 为正则化因子, W_D 为卷积层权重。本文采用 Adam 算法来优化该目标函数,如式(6)所示。

$$L = - \sum I_u^t \cdot \log(\hat{I}_u^t) + \lambda \cdot \|E\| + \lambda \cdot \|W_D\| \quad (6)$$

3 实验与分析

本文首先分析了 MC-RNN 在两个基准数据集上的性能,并进一步分析了 MC-RNN 的结构参数对性能的影响。

3.1 实验设置

3.1.1 数据集

本文在 MovieLens-1M 和 Amazon Movies and TV 两个推荐系统性能测试常用的公开数据集上进行实验。①MovieLen-1M 是 MovieLens 数据集的一个子集,MovieLens 数据集包含多个用户对多部电影的评分数据,包括用户 ID、电影 ID、评分和时间信息。其一共有 6 040 个用户、3 706 部电影和 1 000 209 个评分。②Amazon Movies and TV 为 Amazon Product 数据集的子集,包含了 200 多万用户 20 多万部电影电视类别的评分信息。本文从 Amazon Movies and TV 数据集中选取了评分数量大于 20 的用户,最终选取的数据集一共有 16 141 个用户、116 565 部电影和 953 682 个评分。

3.1.2 算法

本文选取了传统推荐算法和序列推荐算法两类基准算法进行对比。①传统推荐算法: POP、UKNN^[4] 和 BPRMF^[15]。POP 方法算出每一部电影的被评分次数,把电影被评分次数作为评分规则,次数越高,排名越高,也就是向用户推荐最热门的物品。UKNN 方法计算用户之间的欧氏距离,找

到和查询用户距离最近的近邻用户,将近邻用户的物品推荐给查询用户。BPRMF 从贝叶斯分析的角度来获取最大化后验估计,并将该算法应用在矩阵分解模型上。②序列推荐算法: FPMC^[4], GRU4REC^[8], MARank^[14] 以及 HRM^[6]。FPMC 用马尔科夫链建模用户的短期兴趣。基于循环神经网络的序列推荐算法 GRU4REC 使用 GRU-LSTM 建模长期的依赖关系,解决基于会话的序列推荐问题。MARank 和 HRM 将长短期混合起来建模用户。

本文采用 Top-N 推荐评价指标,包括准确率 (P)、召回率 (R)、NDCG (normalized discounted cumulative gain)^[16] 和 MRR (mean reciprocal rank)。设 $S(u)$ 为用户 u 的推荐列表, $T(u)$ 为测试集中用户的真实行为数据记录,则上述评价指标的定义分别表示为式(7)~式(10)。

$$P = \frac{1}{m} \sum_{u \in U} \frac{|S(u) \cap T(u)|}{|S(u)|} \quad (7)$$

$$S = \frac{1}{m} \sum_{u \in U} \frac{|S(u) \cap T(u)|}{|T(u)|} \quad (8)$$

$$\text{NDCG} = \frac{1}{m} \sum_{u \in U} \frac{\text{DCG}(u)}{\text{IDCG}(u)}, \quad (9)$$

$$\text{DCG}(u) = \sum_{i=0}^k \frac{2^{\text{rel}(i)} - 1}{\log_2(i+1)}$$

$$\text{MRR} = \frac{1}{m} \sum_{i=0}^m \frac{1}{\text{rank}(i)} \quad (10)$$

其中, $\text{rank}(i)$ 为对于第 i 个用户,推荐列表中第一个在测试集中的物品所在的排列位置。 m 为用户的个数, k 为推荐列表的个数。

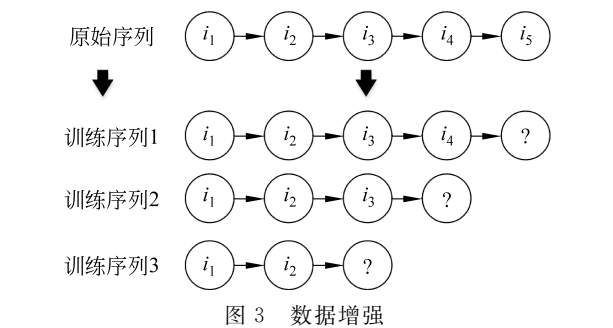
3.1.3 参数

对于每个数据集,我们将用户浏览过的物品序列根据时间排序,然后将每个用户前 80% 的物品作为训练数据,后 20% 的物品作为测试数据。实验中选择训练集上的性能最佳的参数配置作为测试集上的参数配置。各模型超参数分别选取如下配置: UKNN 的近邻个数取值为 20;BPRMF 的隐层个数取值为 64,学习率为 0.005,正则化系数为 0.001; FPMC 和 HRM 的隐层个数取值均为 32,学习率分别为 0.001 和 0.002,正则化系数均为 0.001,批处理大小均为 256,迭代次数均为 30;GRU4REC 和 MC-RNN,训练序列最短长度取值为 32,隐层个数取值均为 16,学习率和正则化系数取值为 0.001。

3.2 数据增强

为了提高训练序列的数量,充分训练网络模型,

本文采用了如图 3 所示的数据增强方式,将一个长度为 L 的原始序列,拆分成 N 个训练序列。为了便于对序列进行多时间尺度研究,限定生成的训练序列最小长度为 M ,则 $N = L - M$ 。假设一条长度为 5 的原始序列 $\{i_1, i_2, i_3, i_4, i_5\}$,最小训练序列长度为 2,则可将其拆分成 3 条训练序列,即 $\{(i_1, i_2, i_3, i_4), i_5\}$, $\{(i_1, i_2, i_3), i_4\}$ 和 $\{(i_1, i_2), i_3\}$ 。



以 MovieLens-1M 为例,图 4 为数据增强前训练序列长度直方图,从中可知大部分用户的物品序列长度为 0~500。

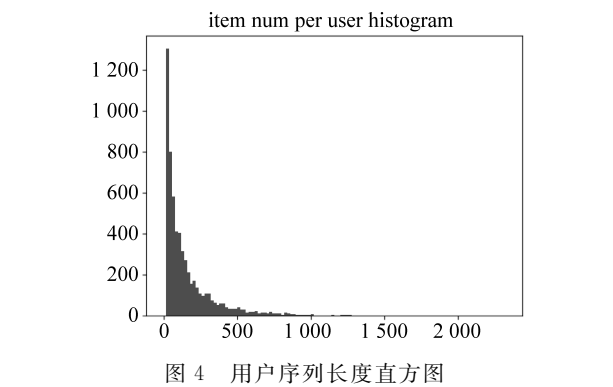
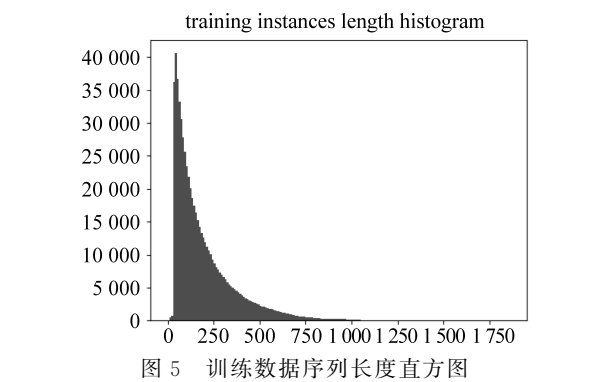


图 5 为数据增强后的训练序列长度直方图,训练实例个数由原来的 6 040 增加到 622 067。



3.3 性能分析

MC-RNN 与四类基准算法在 MovieLens-1M

与 Amazon Movies and TV 上的对比结果如表 1、表 2 所示。在 MovieLen-1M 数据集上 MC-RNN 的时间尺度为三尺度,采取指数方式采样,在 Amazon Movies and TV 数据集上 MC-RNN 的时间尺度为五尺度,采取斐波拉杰方式采样。

表 1 MovieLens-1M 实验结果

算法	Pre@10	Rec@10	NDCG@10	MRR@10
POP	0.101 0	0.037 5	0.050 2	0.214 7
UKNN	0.125 5	0.065 6	0.074 9	0.260 9
BPRMF	0.123 3	0.060 2	0.073 0	0.267 2
FPMC	0.172 3	0.100 6	0.118 1	0.375 5
HRM	0.164 2	0.092 4	0.107 4	0.347 9
GRU4REC	0.197 6	0.105 9	0.128 4	0.399 7
MARank	0.178 3	0.092 7	0.112 8	0.376 4
MC-RNN	0.205 1	0.118 1	0.139 4	0.413 5

表 2 Amazon Movies and TV 实验结果

算法	Pre@10	Rec@10	NDCG@10	MRR@10
POP	0.008 6	0.007 8	0.007 2	0.021 6
UKNN	0.022 6	0.026 3	0.027 9	0.066 9
BPRMF	0.020 4	0.024 8	0.022 7	0.057 8
FPMC	0.025 5	0.026 6	0.028 8	0.071 8
HRM	0.022 8	0.026 5	0.024 6	0.064 2
GRU4REC	0.023 9	0.021 6	0.021 0	0.061 6
MARank	0.021 2	0.019 8	0.019 8	0.058 3
MC-RNN	0.027 7	0.027 5	0.029 2	0.072 7

传统推荐算法 v. s. MC-RNN 和传统的推荐算法 POP、UKNN 和 BPRMF 相比,在 MovieLens-1M 数据集上,MC-RNN 算法的准确率比传统算法中准确率最高的 UKNN 算法提升了 63.43%,在 Amazon Movies and TV 数据集上,MC-RNN 算法的准确率比传统算法中准确率最高的 UKNN 算法提升了 22.57%。与传统的推荐算法相比,序列推荐算法考虑了更多的信息,如物品间的时序关系,因此多尺度序列推荐算法在两个数据集上的性能优于传统算法。

短期依赖序列推荐算法 v. s. MC-RNN 与短期依赖序列推荐算法 FPMC 相比,MC-RNN 算法的 Pre@10 在 MovieLens 1M 上提升了 19.04%,在 Amazon Movies and TV 上提升了 8.63%。仅考虑

短期依赖,只建模局部影响因素,会导致预测结果不稳定,因此多尺度序列推荐算法在基准数据集上的性能优于短期依赖序列推荐算法。

长期依赖序列推荐算法 v. s. MC-RNN 与长期依赖算法 GRU4REC 相比,MC-RNN 的 Pre@10 在 MovieLens-1M 上提升了 3.80%,在 Amazon Moves and TV 上提升了 15.90%。仅考虑长期依赖,只建模全局影响因素,会导致预测模型不灵活,故而多尺度序列推荐算法在基准数据集上的性能优于长期依赖推荐算法。

混合依赖序列推荐算法 v. s. MC-RNN 与长短期混合依赖算法 HRM 相比,MC-RNN 的 MRR@10 在 MovieLens-1M 上提升了 18.86%,在 Amazon Movies and TV 上提升了 13.24%。与 MARank 相比,MC-RNN 的 MRR@10 在 MovieLens-1M 上提升了 9.85%,在 Amazon Movies and TV 上提升了 24.69%。从时序关系来看,HRM 和 MARank 算法依旧归类于长短期混合依赖推荐算法。因为其仍旧是从长期与短期依赖两个角度分别建模,最后通过残差网络将两者融合起来做预测。长短期混合依赖算法的固有问题是短期的定义,因为影响预测结果的物品并不一定能恰好落入定义的所谓短期的区间内。如果采样多个区间,影响预测结果的物品命中率大大提高,这便是本文引入多尺度的意义。因此,多尺度序列推荐算法在基准数据集上的性能优于长短期混合依赖推荐算法。

通过上述对比可知,在 MovieLens-1M 和 Amazon Movies and TV 两个基准数据集上,MC-RNN 算法的四个性能评价指标都要优于以上四类基准算法。四类基准算法在不同的数据集上表现不同。在 MovieLens-1M 上,长期依赖序列推荐算法 GRU4Rec 在四类对比算法中表现最优;在 Amazon Movies and TV 上,短期依赖推荐算法 FPMC 在四类对比算法中表现最优。这个现象也从侧面验证了本文的假设:单一的时间依赖假设不可靠。而长短期混合依赖算法虽然打破了单一时间依赖的假设,但短期依赖的定义不够灵活,导致效果不佳。引入多尺度可以有效解决以上问题。

3.4 网络结构参数的影响分析

影响 MC-RNN 结构的有两个关键因素:带时钟频率的神经元的分组数 g 以及时钟频率的定义。分组数 g 决定了从多少个时间尺度来抽取序列的关联信息。时钟频率的定义决定了从哪些时间尺度

来抽取序列的关联信息。

为了研究 MC-RNN 中尺度数目与定义对模型性能的影响,实验中选取 $g \in \{2,3,4,5\}$,时钟频率的定义采用前述章节中提到的指数、线性以及斐波拉杰作为候选。MC-RNN 在 MovieLens-1M 与 Amazon Movies and TV 上的性能如表 3、表 4 所示。

表 3 MovieLens-1M 尺度采样实验结果

尺度	采样	Pre@10	Rec@10	MRR@10
2	指数/线性/斐波拉杰	0.208 2	0.119 1	0.419 6
3	指数	0.205 7	0.117 6	0.415 6
	线性/斐波拉杰	0.199 0	0.112 4	0.410 3
4	指数	0.205 1	0.118 1	0.413 5
	线性	0.204 4	0.117 3	0.408 2
	斐波拉杰	0.202 7	0.115 2	0.411 6
5	指数	0.194 9	0.106 6	0.389 6
	线性	0.199 2	0.110 1	0.398 6
	斐波拉杰	0.199 5	0.111 0	0.404 1

表 4 Amazon Movies and TV 尺度采样实验结果

尺度	采样	Pre@10	Rec@10	MRR@10
2	指数/线性/斐波拉杰	0.027 2	0.026 2	0.070 9
3	指数	0.025 1	0.023 4	0.066 5
	线性/斐波拉杰	0.027 4	0.026 0	0.071 4
4	指数	0.026 3	0.024 9	0.070 3
	线性	0.027 3	0.025 3	0.070 4
	斐波拉杰	0.027 2	0.026 9	0.071 7
5	指数	0.024 5	0.022 5	0.064 0
	线性	0.025 8	0.024 2	0.069 9
	斐波拉杰	0.027 7	0.027 5	0.072 7

从 MC-RNN 的尺度数 g 来看, g 越大,提取的序列信息越多,理论上模型的性能会越高,但实验结果并非如此。以 MovieLens 上指数时钟频率定义为例,Pre@10 随着 g 的增大而变小,当 $g = 2$ 时,Pre@10 最大。同样,Rec@10 以及 MRR@10 也是这种变化趋势。在 MovieLens 上的结论告诉我们两尺度足够好。Amazon Movies and TV 上采用斐波拉杰时钟频率定义时,Pre@10 随着尺度增大,不断波动,在 $g = 5$ 时达到最大。Rec@10 以及 MRR@10 表现出同样的变化趋势。Amazon Movies and

TV 上的性能随尺度数目 g 的变化趋势说明仅两个尺度是不够的。总之,虽然不同数据集对于多尺度的依赖程度不同,但多尺度建模对于序列化推荐问题是必要的。

从 MC-RNN 的时钟频率定义来看,数据集不同,合适的时钟频率定义也不同。在 MovieLens-1M 上,指数定义在大部分时候是最好的选择。尤其是当 $g \leq 4$ 时,无论是从哪个评价指标来看,指数定义都是最好的。在 Amazon Movies and TV 上,从三个评价指标来看,采用斐波拉杰定义时性能是最好的。与线性定义相比,指数定义与斐波拉杰定义都会使得不同组之间时钟频率差距较大。采用这种时钟频率差距大的定义能从更加不同的层面抽取序列的特征。

4 总结与展望

在序列推荐系统中,对于未来的预测真正起作用的是长时间的依赖(全局)还是短时间依赖(局部)是未知的。现有基于深度学习推荐算法依据长短期依赖来设计推荐算法,在实际应用中不灵活。为了在时间维度上通过不同尺度更精细地对用户兴趣进行动态建模,本文提出了一种基于 CW-RNN 的多时间尺度序列建模推荐算法 MC-RNN。该算法引入 CW-RNN 层,从用户与物品的历史交互序列中抽取多个时间尺度的用户兴趣特征。然后,通过尺度维卷积来建模对不同尺度的用户兴趣特征的依赖,生成多尺度用户兴趣特征的统一表示。最后,利用全连接层建模统一的多尺度用户兴趣特征和隐式物品特征的交互关系。在 MovieLens-1M 和 Amazon Movies and TV 两个公开数据集上的实验结果表明,MC-RNN 优于当前最先进的序列推荐算法。

下一步的工作包括两个方面:其一,在用户与物品的交互历史信息之外,融合更多的物品内容信息,如物品的评价及描述信息等,以获得更准确的推荐;其二,尝试采用注意力机制等其他方式来进行用户兴趣多尺度特征融合方面的研究。

参考文献

[1] 陈洁敏,汤庸,李建国,等. 个性化推荐算法研究[J]. 华南师范大学学报(自然科学版), 2014, 46(5): 8-15.

[2] 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.

[3] Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-based recommendations with recurrent neural networks [C]//Proceedings of the International Conference on Learning Representations, 2016.

[4] Rendle S, Freudenthaler C, Schmidt Thieme L. Factorizing personalized Markov chains for next-basket recommendation[C]//Proceedings of the 19th International Conference on World Wide Web, 2010: 811-820.

[5] Liu Q, Zeng Y, 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, 2018: 1831-1839.

[6] Wang P, Guo J, Lan Y, et al. Learning hierarchical representation model for next basket recommendation [C]//Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015: 403-412.

[7] Hihi S E, Bengio Y. Hierarchical recurrent neural networks for long-term dependencies [C]//Proceedings of Neural Information Processing Systems, 1995: 493-499.

[8] Wu Y, Dubois C, Zheng A X, et al. Collaborative denoising auto-encoders for Top-N recommender systems [C]//Proceedings of Web Search and Data Mining, 2016: 153-162.

[9] He X, Liao L, Zhang H, et al. Neural collaborative filtering [C]//Proceedings of the 26th International Conference on World Wide Web, 2017: 173-182.

[10] Ebesu T, Shen B, Fang Y. Collaborative memory network for recommendation systems [C]//Proceedings of the the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018: 515-524.

[11] Yu F, Liu Q, Wu S, et al. A dynamic recurrent model for next basket recommendation [C]//Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, 2016: 729-732.

[12] Quadrana M, Karatzoglou A, Hidasi B, et al. Personalizing session-based recommendations with hierarchical recurrent neural networks [C]//Proceedings of the Conference on Recommender Systems, 2017: 130-137.

[13]

Zhao W, Wang B, Ye J, et al. Leveraging long and short-term information in content-aware movie recommendation[J]. CoRR, 2017: abs/1712.09059.

[14]

Yu L, Zhang C, Liang S, et al. Multi-order attentive ranking model for sequential recommendation[C]//Proceedings of the National Conference on Artificial Intelligence, 2019.

[15]

Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[J]. Uncertainty in Artificial Intelligence, 2009: 452-461.

[16]

Jarvelin K, Kekalainen J. Cumulated gain-based evaluation of IR techniques[J]. Acm Transactions on Information Systems, 2002, 20(4): 422-446.

[17]

Lops P, De Gmmis M, Semeraro G. Content-based recommender systems: State of the art and trends [M]. Recommender Systems Handbook. Boston, MA, Springer US. 2011: 73-105.

[18]


Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms [C]//Proceedings of the 10th International Conference on World Wide Web, 2001: 285-295.

[19]

Burke R. Hybrid Recommender Systems: Survey and Experiments[J]. User Modeling and User-Adapted Interaction, 2002, 12(4): 331-370.


[20]

Koutnik J, Greff K, Gomez F J, et al. A Clockwork RNN[C]//Proceedings of the ternational Conference on Machine Learning, 2014: 1863-1871.




袁涛(1987—),通信作者,博士研究生,工程师,主要研究领域为数据挖掘、机器学习及推荐系统。

E-mail: yuantao@iscas.ac.cn



牛树梓(1985—),博士,副研究员,主要研究领域为信息检索、数据挖掘和自然语言处理等。

E-mail: shuzi@iscas.ac.cn



李会元(1973—),博士,研究员,博士生导师,主要研究领域为大型异构系统下的高性能科学计算与数学软件、数值偏微分方程的谱方法与谱元素法、特征值问题的高性能计算方法等。

E-mail: huiyuan@iscas.ac.cn


(上接第 96 页)

[30]

Chen X, Hao F, Lin T Y, et al. Microsoft COCO Captions: Data collection and evaluation server[J]. arXiv preprint arXiv: 1504.00325, 2015.


[31]

Seo M, Kembhavi A, Farhadi A, et al. Bidirectional attention flow for machine comprehension[J]. arXiv preprint arXiv, : 1611.01603,2017.




张兆滨(1996—),硕士研究生,主要研究领域为自然语言处理。

E-mail: 610047214@qq.com



王素格(1964—),通信作者,教授,主要研究领域为自然语言处理、情感分析。

E-mail: wsg@sxu.edu.cn



陈鑫(1992—),博士研究生,主要研究领域为情感分析。

E-mail: 1315614497@qq.com