

基于循环时间卷积网络的序列流推荐算法



李太松^{1,2} 贺泽宇^{1,2} 王冰^{1,2} 颜永红^{1,2,3} 唐向红⁴

1 中国科学院声学研究所语言声学与内容理解重点实验室 北京 100190

2 中国科学院大学电子电气与通信工程学院 北京 100190

3 中国科学院新疆理化技术研究所新疆民族语音语言信息处理重点实验室 乌鲁木齐 830011

4 贵州大学现代制造技术教育部重点实验室 贵阳 550025

(tythonlee@163.com)

摘要 针对循环神经网络(Recurrent Neural Network,RNN)模型在序列流推荐中只能从宏观上捕捉序列的演变模式,忽略了物品(Item)间内部的微观联系,无法长程建模序列数据的变化规律的问题,提出了多维度序列建模算法循环时间卷积网络(Recurrent Temporal Convolutional Network,RTCN)。首先,将每个物品表示成定长向量,采用多层因果卷积和扩张卷积操作扩大感受野范围,建立序列元素间的长程依赖关系。利用残差连接网络提取不同层次的特征信息,解决反向传播中梯度衰减甚至消失的问题。综合设计时间卷积网络(Temporal Convolutional Network,TCN)提取序列流中前后物品间的局部特征,将物品信息映射到隐藏空间,得到细粒度的特征向量。为进一步建立元素间的宏观联系,将特征向量依次输入门限循环单元(Gated Recurrent Unit,GRU),迭代更新现有隐藏状态并预测下一时刻的输出。RTCN通过时间卷积网络,从输入序列流提取出长时间、多维度、细粒度的局部关联特征;经过门限循环网络,建模序列间的长距离依赖关系,捕捉序列元素的演变模式,并预测下一个出现的物品。利用网站、手机应用和音乐3个不同场景中的数据对模型进行了实验。实验结果显示,RTCN模型在召回率(Recall)和平均排序倒数(MRR)两个指标上比RNN模型高出6%~13%,比传统推荐算法高出9%~59%。通过对比不同的损失函数,模型在交叉熵损失函数下表现最优。此外,由于TCN中的卷积层具有多通道的结构,当数据维度丰富时,该模型对物品和用户的上下文信息具有很强的综合能力。

关键词: 推荐系统;深度学习;序列流推荐;时间卷积网络;循环神经网络

中图法分类号 TP183

Session-based Recommendation Algorithm Based on Recurrent Temporal Convolutional Network

LI Tai-song^{1,2}, HE Ze-yu^{1,2}, WANG Bing^{1,2}, YAN Yong-hong^{1,2,3} and TANG Xiang-hong⁴

1 Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China

2 School of Electronic, Electrical and Communication Engineering, University of Chinese Academy Sciences, Beijing 100190, China

3 Xinjiang Key Laboratory of Minority Speech and Language Information Processing, Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumchi 830011, China

4 Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education, Guizhou University, Guiyang 550025, China

Abstract Since the Recurrent Neural Network (RNN) generally models transition patterns, ignores the inner connection of items and can't model the long-term evolving patterns of sequential data in session-based recommendations. A Recurrent Temporal Convolutional Network (RTCN) was proposed. Firstly, each item in the sequence is embedded as a vector, the multi-layer causal convolutions and dilated convolutions are applied so that the receptive field is improved and the long-term connections are established. A residual network is stacked to extract features from different layers. Therefore, the gradient vanishing or even disappearing in back propagation can be solved. With above operations, a well-designed Temporal Convolutional Network (TCN) is estab-

收稿日期:2019-05-31 返修日期:2019-08-27 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(11590770-4,11722437,61650202,U1536117,61671442,11674352,11504406,61601453);国家重点研发计划(2016YFB0801203,2016YFC0800503,2017YFB1002803);新疆维吾尔自治区重大科技专项(2016A03007-1);贵州省留学回国人员科技活动择优资助项目(2018.0002)

This work was supported by the National Natural Science Foundation of China (11590770-4, 11722437, 61650202, U1536117, 61671442, 11674352, 11504406, 61601453), National Key Research and Development Program (2016YFB0801203, 2016YFC0800503, 2017YFB1002803), Key Science and Technology Project of the Xinjiang Uygur Autonomous Region (2016A03007-1) and Project of Guizhou High-level Study Abroad Talents Innovation and Entrepreneurship(2018.0002).

通信作者:王冰(ariel.bingwang@outlook.com)

lished. It extracts local features from sequence items, maps item information into latent space and generates fine-grained feature vectors as results. To further explore the connections between items in macroscopic way, the feature vectors are feed into Gated Recurrent Unit (GRU). After multiple iterations and updates to hidden states, the model can make a prediction of the next item. RTCN can extract long-time, multi-dimension, fine-grained local features from inputs by adapting temporal convolutional network. It also models the long-distance connections between items, captures the transition patterns and infers the next items by using GRU networks. The experimental results demonstrate that the RTCN model outperforms 6% ~ 13% than RNN-based model and 9% ~ 59% than other traditional recommendation methods under the metrics of Recall and Mean Reciprocal Rank (MRR). By comparing different definitions of loss, RTCN performs best under the cross entropy loss function. Meanwhile, due to the TCN multi-channel structure, the proposed model has a high potential capacity to embedding context features of items and users when the dataset information is rich.

Keywords Recommendation system, Deep learning, Session-based recommendation, Temporal convolutional network, Recurrent neural network

1 引言

大数据时代的到来,使得人们逐渐被淹没在商品的海洋中。在充满各种选择的场景下,为用户提供其想要并有用的物品将是推荐系统的重要任务。现实场景中,很多网站并没有保存足够的用户信息;同时,由于隐私保护,很多用户的个人数据不能被使用。在数据单一或信息缺失的条件下,传统的推荐算法将受到考验。因此,根据用户单一的行为信息预测其下一时刻将要点击的物品变得至关重要。该研究被定义为序列流推荐(Session-based Recommendations),近年来逐渐成为推荐系统领域的研究热点。

基于循环神经网络的序列流推荐^[1]于2016年被提出并广泛应用于推荐领域。该模型采用门限循环单元捕捉用户点击物品的顺序特征,并对网站点击数据进行序列化建模。然而,基于RNN的推荐模型只能量化向量之间的关联^[2],并没有捕捉相邻向量内部元素之间的关系。图像领域的研究表明,卷积神经网络(Convolutional Neural Network, CNN)具有很好的局部特征提取能力。通过不同卷积核的卷积操作,可以将向量元素映射到不同维度,从而提取丰富的局部特征。为此,本文提出采用时间卷积网络提取微观特征,然后使用循环神经网络拟合全局序列的演变,由微观到宏观、从低维到高维进行序列化建模。所提模型具有广泛的应用场景,如电商的商品推荐、手机用户的位置预测以及音乐数据流推荐,对广大商家和用户具有重要的现实意义。

2 相关工作

在推荐系统领域,传统的算法主要有两类:基于因子分解的机器学习算法^[3-5]和基于协同过滤的相似度算法^[6-7]。协同过滤方法主要依据物品(用户)共同出现的频率来计算物品(用户)之间的相似度,进而推荐相似的物品。该方法由于计算快且数据需求量小,在序列流推荐中被大量使用,如亚马逊和阿里巴巴等电商的首页推荐。然而,协同过滤算法仅计算物品(用户)之间的相似度,没有考虑用户兴趣的变化和物品序列的演变规律,在实际应用中取得的效果有限。基于因子分解的方法通过分解用户-物品的交互矩阵,得到每个用户和物品的低维向量。推荐系统通过计算分解后向量的内积,填充矩阵中缺失的元素,重构用户-物品交互矩阵。由于序列流

推荐中往往缺失用户的个人信息和交互信息,因此因子分解方法将不适合用于流数据的推荐。

近些年,基于人工神经网络的深度学习方法在计算机视觉、语音识别和自然语言处理等多个领域取得了优异的性能。在处理图像和语音等非结构化数据时,神经网络通过卷积层、循环神经单元和全连接网络,从原始数据中提取局部特征并拟合时序演变模式,将输入信号转变和压缩为低维度向量,从而完成现实需求下的各种任务。随着深度学习在图像、语音和文本等数据中的成功应用,越来越多的研究者将其应用到推荐系统领域。作为推荐领域的一个重要分支,序列流推荐逐渐成为了深度学习方法的重要应用之一。

2016年,Hidasi等^[1]首先提出了GRU4Rec模型,利用循环神经网络学习点击序列之间的逻辑关系,进行用户行为预测。不同于传统的基于物品相似度的方法,Hidasi首次将序列流推荐问题转化成序列元素变化规律预测的问题。同年,Hidasi等^[8]在推荐系统大会上进一步丰富了RNN模型的输入,将物品的文本、图像信息作为上下文信息并行输入到RNN模型中,取得了比单一数据更好的效果。2017年,基于RNN的序列流推荐被进一步细化。Bogina等^[9]针对序列中前后序列元素的间隔时长进行切分处理,使得间隔时间较长的序列元素被多次输入RNN,符合现实生活中人们对某个物品感兴趣时停留时间较长的应用情景。Quadrana等^[10]则将用户信息嵌入RNN模型,通过两个并行的RNN单元(一个捕捉序列元素的演变规律,另一个描述用户的兴趣变化),丰富了输入信息的维度,改善了序列流推荐的效果。目前,基于RNN的深度学习方法在序列流推荐任务上取得了显著的成果,其主要思路是从深度和广度两方面来扩充GRU4Rec算法的边界,从深度上丰富输入序列的上下文信息,从广度上增加输入数据的维度,最终使用循环神经网络及其变种进行序列化建模。然而,面对数据缺失或数据单一的情况(如只有购买记录,没有用户特征、商品属性以及购买时间等信息)时,上述方法将无法有效提升传统GRU4Rec算法的效果。此外,基于RNN的方法在处理序列流元素之间的向量关系时,采用矩阵相乘和累加的粗粒度操作,如何捕捉不同向量各元素之间的局部关联是该方法所忽略的细节。

3 相关工作

为实现面向序列数据的推荐,本文提出了基于时间卷积

网络的序列预测模型 RTCN。该模型可在数据单一的情况下建立物品之间不同粒度的关联关系,捕捉用户行为的变化规律,从而进行更精准的推荐。

为实现面向序列数据的推荐,本文提出了一种融合循环神经网络和时间卷积网络的模型结构 RTCN。该模型首先使用时间卷积网络提取序列的局部关联特征,建立表示序列特征的矩阵;然后将其作为循环卷积网络的输入;最后使用反向传播算法进行端到端的训练,生成最终的模型。该模型可以提取序列数据中更有效的特征,通过序列建模对序列元素进行预测。

3.1 局部关联特征的提取

时间卷积网络是基于卷积神经网络的时序模型框架^[11]。与通常将 CNN 作为图片或文本的特征提取模型不同^[12-14],时间卷积网络主要用于提取时序数据特征。当模型输入序列为 $\mathbf{X} = \{x_1, x_2, \dots, x_T\}$ 时,输出可以是相应的预测序列 $\mathbf{Y} = \{y_1, y_2, \dots, y_T\}$,也可以是中间层的特征表示 $\mathbf{H} = \{h_1, h_2, \dots, h_T\}$ 。所有的输出满足因果条件限制,即当前的输出 y_t 只与 $\{x_1, x_2, \dots, x_t\}$ 有关,与“未来”的输入 $\{x_{t+1}, x_{t+2}, \dots, x_{t+T}\}$ 无关。这也符合现实生活中大多数的应用场景,即在只知道历史数据的情况下预测未来的状态。输入和输出的映射关系可以正式表示为:

$$y_1, y_2, \dots, y_T = f(x_1, x_2, \dots, x_T) \quad (1)$$

映射函数 f 表示 TCN 网络的作用,即通过学习和调整神经网络的参数,使得每一个序列的输入对应到相应的输出。

时间卷积网络主要由 3 部分组成:因果卷积层(Causal Convolutions),扩张卷积层(Dilated Convolutions),残差连接层(Residual Connections)。根据实际的数据类型及分布,本文设计了符合实验效果的网络结构,如图 1 所示。

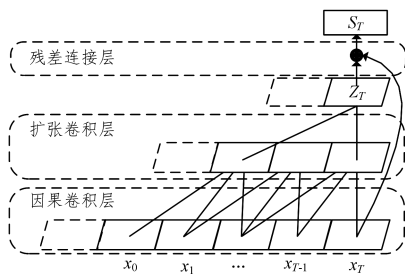


图 1 TCN 网络结构

Fig.1 Structure of temporal convolutional network

(1) 因果卷积层

时间卷积网络的设计基于两个准则:1)网络的输出大小等于输入数据的大小;2)当前的输出只与当前和历史的输入有关,与未来的输入无关。为满足第一个准则,TCN 的第一层使用一维全卷积网络(Fully-Convolutional Network, FCN)^[15]。FCN 在卷积的过程中,设置步长(stride)为 1,填充值(zero-padding)为 $kernel_size - 1$ ($kernel_size$ 为卷积核大小),输入矩阵经过全卷积网络映射到隐藏层后,数据大小不变。同时,为保证输入与输出的因果关系,TCN 只对当前时刻 t 及 t 以前时刻的输入进行卷积操作。由于其要求输入与输出数据的大小一致,在面对大数据量、长程历史连接的数据时,需要很深的网络结构或者巨大的卷积核,这将对模型造成很大的计算负担。为此,在因果卷积层的基础

上,TCN 引入了扩张卷积层。

(2) 扩张卷积层

扩张卷积与普通卷积最大的不同是其在普通卷积的感受野(Receptive Field)中注入了空洞,以此来增加感受野的范围。扩张卷积的超参数被称为扩张率(Dilation Rate),用于表示卷积核的间隔数量。为避免网格效应(Griding Effect),通常将扩张率设为 2 的指数形式 $(1, 2, 4, 8, \dots, 2^i)$ 。扩张率值为 1 表示正常的卷积操作;当扩张率的值增大时,感受野的范围也增大,卷积的输出将与更长的历史输入相关。在处理序列数据时,扩张卷积层的结构如图 2 所示。第一层是因果卷积层,卷积后数据的大小等于模型输入数据的大小。随着扩张卷积层数的增加,序列当前的隐藏层状态将与历史输入的长度呈指数级关联;同时,隐藏层的容量大大减小,输入数据将得到更高维度的表示。

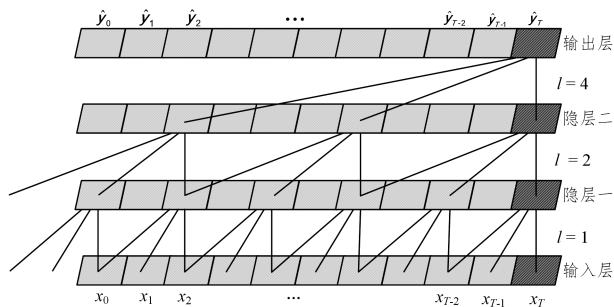


图 2 一维扩张卷积网络

Fig.2 1-D dilated convolutional network

(3) 残差连接层

残差网络用于解决深度条件下卷积神经网络性能退化的问题,分类性能表现出色^[16]。在序列流推荐任务中,因果卷积层和扩张卷积层的叠加使得神经网络逐渐变深。为了减少网络反向传播中梯度衰减甚至消失的现象,在模型的输出层引入残差连接,将模型输入 \mathbf{X} 加权融合到卷积网络的输出中。其表达式如下:

$$\mathbf{O} = \text{Activation}(\mathbf{X} + \mathbf{F}(\mathbf{X})) \quad (2)$$

其中, $\mathbf{F}(\mathbf{X})$ 表示卷积层的输出, $\text{Activation}(\cdot)$ 表示激活函数。残差连接层的引入使得模型成为深层网络和浅层网络的集成。在提取高维卷积特征后,残差将大大提高模型的泛化能力。

融合残差连接层的 TCN 如图 1 所示。模型中,由扩张因果卷积网络(Dilated Casual Conv)、权值归一化层(Weighted-Norm)^[17]和非线性激活单元组成卷积模块,再通过残差连接层与模型输入联合组成时间卷积网络的输出。TCN 作为本文模型的特征提取部分,相比传统 CNN 具有以下 3 个优点:1)通过扩张卷积结构,解决了序列预测中的长程依赖问题;2)通过堆叠式扩张卷积层和参数共享机制,大大减小了模型的计算量;3)通过残差连接结构,解决了深度卷积模型的性能退化问题,使得模型可以提取更高维度和更丰富的特征。

3.2 RTCN 模型

为实现序列数据的推荐,本文提出了一种融合循环神经网络和时间卷积网络的深度学习模型。该模型首先将序列数据中的各元素映射成随机分布的低维实数向量,建立表示元素初始特征的序列矩阵,然后将其作为 RTCN 模型的输入,

最后使用反向传播算法进行端到端的训练。模型训练后,可对序列数据进行输入与输出的预测,即输入一个序列元素,模型预测出最有可能出现的下一个元素。本节将详细介绍循环时间卷积网络(RTCN)的模型结构,分析模型的融合机制,最后介绍模型的训练预测方法。

3.2.1 循环时间卷积网络的结构

循环时间卷积网络主要由时间卷积网络和门控循环单元组成。TCN提取长程历史数据的局部特征;GRU单元将提取的特征进行序列化建模,捕捉序列的演化模式。融合TCN和GRU的循环时间卷积网络可分为3层,即输入层、卷积特征提取层和序列层,如图3所示。

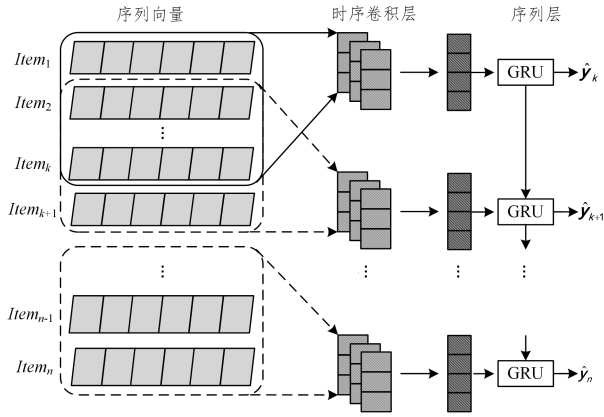


图3 RTCN的结构

Fig. 3 Structure of recurrent temporal convolutional network

第一层是输入层。定义输入序列的长度为 n ,每个输入序列中的元素(item)被映射成 d 维随机分布的实数向量,然后纵向拼接成二维数据矩阵 $\mathbf{X} \in \mathbf{R}^{d \times n}$,具体表示为:

$$\mathbf{X} = \mathbf{X}_1 \oplus \mathbf{X}_2 \oplus \dots \oplus \mathbf{X}_n \quad (3)$$

其中, \oplus 为拼接运算符, $\mathbf{X}_i \in \mathbf{R}^d$ 为序列中第 i 个元素对应的实数向量。对于输入长度为 n 的序列,为保证每个元素拥有长度为 k 的历史信息,模型从序列的第 k ($k < n$)个元素开始计算损失函数并更新参数。将 k 之后的元素输入到模型中时,通过向前移动历史元素窗口,来保证每个输入元素有相同长度的历史序列。

第二层是卷积特征提取层。卷积层通过卷积操作提取局部关联特征,实现对序列流数据的局部感知。根据TCN中扩张因果卷积层的结构,卷积层首先对输入序列矩阵 \mathbf{X} 进行因果卷积操作,在对序列第 i 个元素进行卷积时,只取该元素之前的元素信息,不涉及 i 之后的序列元素。随后,使用扩张卷积核对上一层卷积后的输出进行卷积操作,得到包含历史局部信息的高维特征表示;同时,数据量将大大减小。

给定序列输入矩阵 $\mathbf{X} \in \mathbf{R}^{d \times n}$,利用窗口大小为 k 的滤波器对序列中的元素 s 进行扩张卷积:

$$F(x) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{X}_{s-h \cdot i} \quad (4)$$

其中, h 表示扩张率, $s-h \cdot i$ 指定卷积的方向, $f(\cdot)$ 表示卷积核函数。当卷积核 f 与 \mathbf{X} 中 s 之前的元素进行扩张卷积后,得到一个特征向量 \mathbf{C}_i 。当输入序列中的所有元素经过卷积操作后,就得到一张特征映射图:

$$\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{n-k}\} \quad (5)$$

之后,通过权值归一化层和激活函数得到特征矩阵:

$$\hat{\mathbf{C}} = \text{Activation}(\text{WeightedNorm}(\mathbf{C})) \quad (6)$$

其中, $\text{Activation}(\cdot)$ 表示线性整流函数(Rectified Linear Unit, ReLU)。对特征映射图进行归一化可以加速模型收敛,而非线性化操作使得模型具有更强的泛化能力。

利用卷积层中的不同卷积核可以提取不同的局部特征。本文设置 m 种卷积核结构,对每一个序列元素提取 m 个特征矩阵,充分考虑了序列元素之间的上下逻辑关系。利用所有类型的卷积核提取的特征矩阵,通过求和池化(Sum Pooling)操作,来得到卷积特征矩阵 $\mathbf{Z} \in \mathbf{R}^{(n-k) \times m}$ 。

$$\mathbf{Z} = \{\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_{n-k}\} \quad (7)$$

其中,矩阵元素 \mathbf{Z}_{ij} 表示序列中第 i 个元素经过第 j 个类型的卷积核产生的特征值。最后,通过残差连接网络使得卷积特征矩阵 \mathbf{Z} 和输入矩阵 \mathbf{X} 融合,得到序列特征矩阵 $\mathbf{S} \in \mathbf{R}^{(n-k) \times m}$:

$$\mathbf{S} = \mathbf{Z} + \mathbf{X}_{k:n} \quad (8)$$

其中,为了使卷积特征矩阵 \mathbf{Z} 和输入序列矩阵 \mathbf{X} 的列数一致,特别设定卷积核的个数 m 等于 \mathbf{X} 中元素向量的长度。同时,考虑到元素的历史长度一致性,只取 \mathbf{X} 中后 $n-k$ 个元素参与损失函数的计算。

第三层为序列层。本文采用门限循环单元作为序列层的建模单元。选择GRU集成序列的特征向量具有以下优势:1)GRU的时间传递特性与人们选择物品时兴趣的变化相吻合;2)解决了RNN在长序列建模时梯度消失的问题,能够捕捉序列元素之间的长程依赖关系;3)GRU与LSTM的性能接近,且计算消耗更少,模型收敛更快^[18]。因此,在序列数据推荐任务中,大多数模型采用GRU作为建模的基本单元^[1,8-10]。

GRU通过门控单元实现了长序列信息的传递和保存。在每一个时间点,GRU接收一个序列元素的输入特征向量 s_t ,并结合前一时刻的隐藏状态 \mathbf{h}_{t-1} ,加权更新新时刻的隐藏状态 \mathbf{h}_t 。迭代和更新公式如下:

$$\begin{cases} \mathbf{z}_t = \sigma(\mathbf{W}_z s_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z) \\ \mathbf{r}_t = \sigma(\mathbf{W}_r s_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r) \\ \hat{\mathbf{h}}_t = \tanh(\mathbf{W}_h s_t) + \mathbf{r}_t \cdot \mathbf{U}_h \mathbf{h}_{t-1} + \mathbf{b}_h \\ \mathbf{h}_t = (1 - \mathbf{z}_t) \cdot \mathbf{h}_{t-1} + \mathbf{z}_t \cdot \hat{\mathbf{h}}_t \end{cases} \quad (9)$$

其中, \cdot 是点乘运算;重置门限 \mathbf{r}_t 和更新门限 \mathbf{z}_t 控制每个隐含层的信息更新; \mathbf{W} 和 \mathbf{U} 表示系数矩阵, \mathbf{b} 表示偏置向量,用于自适应地选择和丢弃构建当前序列元素的历史信息。给定序列特征矩阵 $\mathbf{S} = \{s_1, s_2, \dots, s_{n-k}\}$,依次输入GRU进行迭代计算,得到各隐层向量 $\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{n-k}\}$ 。每一个隐层向量被认为包含了下一个序列元素的逻辑信息,通过输入全连接网络,得到每一个元素出现的概率分布。

循环时间卷积网络通过输入层和卷积特征提取层,从输入序列流中提取出长时间、多维度和细粒度的局部关联特征。经过门限循环网络,建模序列间的长距离依赖关系,捕捉序列元素的演变模式,预测下一时刻的元素分布概率。

3.2.2 损失函数的定义

循环时间卷积网络的预测输出是所有序列元素对应的概

率值,对于推荐系统而言,这是一个学习排序(Learning to Rank, LTR)的问题,即给定当前元素,预测下一时刻所有序列元素的排列顺序。

本文使用多个损失函数来优化模型的预测性能,包括交叉熵、BPR(Bayesian Personalized Ranking)和 TOP1。其中,交叉熵是单文档排序(Point Wise)方法的损失函数,BPR 和 TOP1 则是文档对方法(Pair Wise)的损失函数。损失函数定义如下:

$$\begin{cases} CE_Loss = \sum_{i=1}^n y_i \cdot \ln(\hat{y}_i) \\ BPR_{Loss} = -\frac{1}{N_s} \sum_{j=1}^{N_s} \log(\sigma(\hat{r}_{s,i} - \hat{r}_{s,j})) \\ TOP1_{Loss} = \frac{1}{N_s} \sum_{j=1}^{N_s} \sigma(\hat{r}_{s,j} - \hat{r}_{s,i}) + \sigma(\hat{r}_{s,i}^2) \end{cases} \quad (10)$$

其中,在交叉熵损失函数中, y_i 和 \hat{y}_i 分别表示元素 i 的实际标签值和预测概率值, n 表示序列元素的类别个数;在 BPR 和 TOP1 损失函数中, N_s 表示采样生成的负样本数, $\hat{r}_{s,i}$ 表示命中样本 i 的排名, $\hat{r}_{s,j}$ 表示负样本的排名, $\sigma(\cdot)$ 表示 sigmoid 函数。BPR 和 TOP1 都强调了正样本排序的优先特性,不同的是, TOP1 加入了正则项 $\sigma(\hat{r}_{s,i}^2)$,用于防止参数优化过程中负样本损失值过大,使得优化过程更为稳定^[1]。

3.2.3 循环时间卷积网络的训练

本文提出的循环时间卷积网络在训练过程中采用的是有监督学习方式,用输入序列中每个元素对应的下一个元素表示真实概率分布,将模型输出的元素概率分布作为预测值。利用不同损失函数计算真实值与预测值之间的损失,以交叉熵为例:

$$Loss(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = \frac{1}{n-k} \sum_{k=k}^n y_i \cdot \ln(\hat{y}_i) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|^2 \quad (11)$$

其中, \mathbf{x} 表示输入序列; $\boldsymbol{\theta} = \{ \mathbf{W}, \mathbf{U}, \mathbf{b} \}$ 表示模型参数集合,也是模型更新和优化的目标参数;参数 i 表示输入序列的第 i 个元素,为保证元素的历史序列长度为 k ,模型只计算第 k 个至第 n 个元素的交叉熵作为损失函数;为防止过拟合,增强模型的泛化能力,在损失函数中加入了 L_2 正则项, λ 为正则项的偏置。在训练过程中,采用随机梯度下降法(Stochastic Gradient Descend, SGD)最小化目标函数,通过反向传播算法进行参数更新。

4 实验及结果分析

4.1 实验数据

本文实验使用的数据来自 3 个领域:网络电商、手机应用和在线音乐。3 个数据集分别表示用户的在线商品点击流、打卡位置序列和在线音乐播放序列,分别定义为 RSC15, BK 和 LastFM。数据集的统计特征如表 1 所列。

表 1 数据集的统计特征

Table 1 Statistics of datasets

Dataset	Users	Items	Events
RSC15	7966257	37483	31637239
BK	1679	23243	599618
LastFM	954	48188	4320170

(1)RSC15。在线电商数据来源于 2015 年的推荐系统挑

战赛^[19],其中包含 Yoochoose 网络购物平台在 6 个月的点击流数据。经过数据预处理和筛选后,提取 7966257 个记录窗口,共计 31637239 次点击和 37483 个商品(item)作为实验数据。根据时间的先后,划分出 15324 个窗口,共 71222 次点击事件作为测试数据集,其余作为训练数据。

(2)BK。手机应用数据集,来自社交网站 Brightkite^[20]。该数据集包含用户使用网站移动端 App 时的打卡数据(check in)。每一条记录包含用户编号、打卡地点和打卡时间。数据收集于 2011 年,包含 1679 位用户、23243 个打卡地点和 599618 条打卡记录。本文使用数据中后 2.5% 的数据作为测试集,其余数据作为训练集,用训练集训练好模型后,预测测试集中每个用户在下一时刻打卡的地点。

(3)LastFM。在线音乐数据集,来自音乐播放网站 LastFM^[21]。该数据集采集了用户的听歌记录以及相应的音乐专辑信息。本文将用户所听音乐的歌手作为序列元素,预测用户下一时刻所听歌曲的歌手。该音乐数据包含 954 位用户、48188 位音乐人和 4320170 条播放记录。与 Brightkite 数据类似,本文使用数据中后 2.5% 的数据作为测试集,其余数据作为训练集。

4.2 评价指标

面向序列流数据的预测多数限定在推荐场景中,由于推荐系统每次给用户推荐的商品有限,而用户在多数情况下只对推荐列表中的少数商品感兴趣,因此需要评测推荐列表中的元素是否齐全并且排名是否靠前。选择召回率(Recall)和平均排序倒数(Mean Reciprocal Rank, MRR)作为模型的评测指标。受推荐列表长度的限制,一般只取前 10 个商品作为推荐展示,两个指标分别表示为 Recall@10 和 MRR@10。

两种评价方式从召回和排名两个方面来衡量推荐模型的性能表现,在序列流推荐中被广泛使用^[1,8-10]。其计算公式如下:

$$\begin{aligned} Recall &= \frac{TP}{TP+FN} \\ MRR &= \frac{1}{Q} \sum_{i=1}^Q \frac{1}{rank_i} \end{aligned} \quad (12)$$

其中, TP 表示用户感兴趣的物品在推荐列表中的数量, FN 表示用户感兴趣的物品不在推荐列表中的数量; $|Q|$ 表示用户感兴趣的物品数量, $rank_i$ 表示用户感兴趣的物品在推荐列表中的排名。

4.3 实验结果

4.3.1 实验设置

为了验证 RTCN 算法的有效性,将其与序列流推荐中广泛应用的几种算法进行对比。

(1)POP:最受欢迎推荐算法,根据训练集中出现次数最多的 item 进行推荐。该方法是许多领域的经典推荐算法。

(2)S-POP:根据每位用户在训练集中点击最多的 item 进行推荐。不同于 POP 方法,该算法考虑了各用户的不同喜好。

(3)Item-KNN:基于商品相似度的 KNN 聚类算法。商品间的相似度根据商品在不同用户选择列表中共现的频率来求取。该算法基于用户喜欢相似物品的特性进行推荐。

(4)GRU4Rec:Hidasi 等^[1]提出的序列流推荐的经典算法。将序列元素使用 one-hot 表示后,输入 GRU 单元进行序列化建模。

利用以上算法进行实验时,将每一个模型的参数调到最优,用最好的预测结果进行对比实验。需要说明的是,由于实验数据限定为单一的序列流数据,不存在元素的上下文和具体时间等信息,因此文献[8-10]中提出的方法不适用于该实验,本文将不对其进行实验对比。RTCN 模型取得最优性能的参数设定如表 2 所列,其中学习率初始设置为 4,并随着迭代次数的增加及性能的提升而减小;不同数据集的迭代次数的设定不同,通常在 50~100 之间收敛完毕。

表 2 RTCN 模型的超参数设置

Table 2 Hyper-parameters setting of RTCN model

超参数	说明	参数值
n	序列流长度	20
k	历史长度	10
d	元素向量长度	100
$kernel_size$	卷积核大小	3
m	卷积核个数	100
lr	学习率	4
l	扩张卷积层数	2
$nhid$	隐层节点个数	180

实验在 Tesla K80 GPU (12 GB GDDR4 显存),E5-2698v3 @2.3GHz CPU 环境中运行,以迭代 100 次为标准,3 个数据集 (RSC15, BK, LastFM) 的计算时间分别约为 149.51 h, 1.76 h 和 78.11 h。RTCN 通过随机梯度下降法进行参数更新和优化,用训练集训练好模型后,对测试集序列数据进行预测。

4.3.2 实验结果

序列推荐的实验结果如表 3 和表 4 所列。

表 3 不同模型在 3 个数据集上的召回率(Recall@10)结果

Table 3 Recall@10 results of different model on three datasets

模型	Recall@10		
	RSC15	BK	LastFM
POP	0.0042	0.0625	0.0375
S-POP	0.2344	0.2633	0.1123
Item-KNN	0.4165	0.4851	0.4292
GRU4Rec	0.4599	0.5665	0.4689
RTCN	0.5956	0.6531	0.5688

表 4 不同模型在 3 个数据集上的召回率(MRR@10)结果

Table 4 MRR@10 results of different model on three datasets

模型	MRR@10		
	RSC15	BK	LastFM
POP	0.0006	0.0019	0.0201
S-POP	0.1217	0.1498	0.0724
Item-KNN	0.1848	0.4163	0.4008
GRU4Rec	0.2194	0.4387	0.4251
RTCN	0.3082	0.5013	0.5056

从表 3 和表 4 可以得出以下结论:

(1)基于时序建模的推荐算法 GRU4Rec 和 RTCN 大幅优于基于流行度的算法。GRU4Rec 和 RTCN 在召回率和平均排序倒数方面都比 POP 和 S-POP 高出几十个百分点。这是因为基于流行度的方法 POP 和 S-POP 只考虑了物品的个体点击特征,没有关注物品之间的关联信息。基于时序建模的算法在建模物品之间的逻辑关系时,点击次数多的物品由

于多次出现,也能被模型捕捉到,因此 GRU4Rec 和 RTCN 具有更强的拟合能力。

(2)基于时序建模的推荐算法 GRU4Rec 和 RTCN 也优于基于相似度的算法。Item-KNN 的召回率比 POP 和 S-POP 分别高出 41 和 18 个百分点,但低于 GRU4Rec 和 RTCN。由于 Item-KNN 的推荐机制是寻求高频共现的相似物品,当输入一个物品时,通过聚类算法找出相似物品进行推荐。该方法可以捕捉到序列中共现程度高的物品,相比单纯计算物品流行度的方法增加了物品之间的关联信息,因此优于 POP 和 S-POP 模型。然而,面对物品种类多且每个物品点击量小的情况,共同出现的物品变得稀疏而多样,模型将无法准确预测下一个点击的物品。GRU4Rec 和 RTCN 将序列输入隐射到隐层表示,可以捕捉不同种类物品之间的共性,因此在面对稀疏数据时具有更好的表达能力。

(3)本文算法 RTCN 优于传统时序建模算法 GRU4Rec。相比 GRU4Rec,本文提出的 RTCN 的召回率 $Recall@10$ 在 3 个数据集上分别提升了 13.57%,8.66%,9.99%;平均排序倒数 $MRR@10$ 分别提升了 8.88%,6.26%,8.05%。对于不同的评价指标,不同数据有着显著而普遍的提升效果。实验结果表明,TCN 模型采用多层结构可提取高维度特征;残差连接层的引入,使得模型可以有更大的深度;初始元素向量与高维卷积特征的加权融合,也使得序列元素的特征具有更好的表达能力;同时,GRU 具有的长程依赖建模能力,可以进一步优化序列特征的依赖关系,由向量元素到向量整体、从微观到宏观地捕捉序列的变化模式。

4.3.3 损失函数的分析

为进一步对比不同损失函数对模型预测能力的影响,本文在 3 个数据集中分别采用不同的损失函数进行了序列预测实验。如表 5 所列,交叉熵损失函数在不同评价指标 (Recall@10 和 MRR@10) 和不同数据集 (RSC15, BK 和 LastFM) 下,都比基于文档对方法的 BPR 和 TOP1 表现更优。其原因在于:负样本的空间过大,针对不同数据集和不同批量大小的输入难以找到最合适的负样本,因此影响到正负样本的损失函数计算。文献[1]提到了交叉熵不稳定特性,由于在输入 GRU 层之前进行了特征的细粒度提取操作,使得输入 GRU 的元素向量分布更为均匀;正则项的引入,也使得模型优化的稳定性得到提升。此外,交叉熵损失函数不需要元素对之间的采样操作,大大降低了计算复杂度,有助于模型的快速收敛。

表 5 不同损失函数的实验结果

Table 5 Experimental results with different loss functions

Loss	RSC15		BK		LastFM	
	Recall@10	MRR@10	Recall@10	MRR@10	Recall@10	MRR@10
CE	0.5956	0.3082	0.6531	0.5013	0.5688	0.5056
BPR	0.3112	0.1306	0.0618	0.0254	0.4006	0.3168
TOP1	0.0646	0.0258	0.5247	0.3489	0.3259	0.2443

结束语 针对主流面向时序数据的推荐算法难以捕捉序列流的局部特征演化的问题,本文提出基于时序卷积网络的特征提取算法,从微观的角度提取向量元素之间的联系;同时引入门限循环单元,用于捕捉特征向量之间的逻辑关系,强化

了模型预测序列的演化能力。实验结果表明,本文方法在网站商品推荐、用户位置推荐和音乐流推荐任务方面的表现都优于传统方法,有效地提高了推荐预测的精准度。本文还进一步分析了不同损失函数对模型性能的影响,在3个数据集中验证了交叉熵是当前模型和任务的最佳损失函数。此外,基于卷积网络的深度学习模型在处理图片、文本时具有优异的特征提取能力,可以将物品的照片、评价等额外信息很好地融入特征向量。综上,所提模型对物品和用户的上下文信息具有很强的潜在综合能力。

参 考 文 献

- [1] HIDASI B, KARATZOGLOU A, BALTRUNAS L, et al. Session-based recommendations with recurrent neural networks [J]. arXiv:1511.06939, 2015.
- [2] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back-propagating errors [J]. Nature, 1986, 323(6088):399-421.
- [3] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems [J]. Computer, 2009, 42(8):30-37.
- [4] WEIMER M, KARATZOGLOU A. Cofi rank-maximum margin matrix factorization for collaborative ranking [C] // Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, 2008:1593-1600.
- [5] HIDASI B, TIKK D. Fast ALS-Based tensor factorization for context-aware recommendation from implicit feedback [C] // Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Berlin: Springer, 2012:67-82.
- [6] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms [C] // International Conference on World Wide Web, ACM, 2001:285-295.
- [7] KOREN Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model [C] // ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008:426-434.
- [8] HIDASI B, QUADRANA M, TIKK D. Parallel recurrent neural network architectures for feature-rich session-based recommendations [C] // ACM Conference on Recommender Systems, ACM, 2016:241-248.
- [9] BOGINA V, KUFLIK T. Incorporating dwell time in session-based recommendations with recurrent Neural networks [C] // CEUR Workshop Proceedings, 2017:57-59.
- [10] QUADRANA M, KARATZOGLOU A, HIDASI B, et al. Personalizing session-based recommendations with hierarchical recurrent neural networks [C] // Eleventh ACM Conference on Recommender Systems, ACM, 2017:130-137.
- [11] BAI S, KOLTER J Z, KOLTUN V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling [J]. arXiv:1803.01271, 2018.
- [12] LIANG M, HU X. Recurrent convolutional neural network for object recognition [C] // Computer Vision and Pattern Recognition. IEEE, 2015:3367-3375.
- [13] PINHEIRO P H O, COLLOBERT R. Recurrent convolutional neural networks for scene labeling [C] // International Conference on International Conference on Machine Learning, 2014:82-90.
- [14] LÉCUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.
- [15] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation [C] // IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2015:3431-3440.
- [16] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C] // IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2016:770-778.
- [17] SALIMANS T, KINGMA D P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks [C] // Advances in Neural Information Processing Systems, 2016:901-909.
- [18] CHUNG J, GULCEHRE C, CHO K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling [J]. arXiv:1412.3555, 2014.
- [19] BEN-SHIMON D, TSIKINOVSKY A, FRIEDMANN M, et al. Recsys challenge 2015 and the yoochoose dataset [C] // RecSys'15: Proceedings of the 9th ACM Conference on Recommender Systems, New York: ACM, 2015:357-358.
- [20] CHO E, MYERS S A, LESKOVEC J. Friendship and mobility: user movement in locationbased social networks [C] // Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011:1082-1090.
- [21] CELMA O. Music Recommendation and Discovery in the Long Tail [M]. Springer, 2010.



LI Tai-song, born in 1990, doctoral student. His main research interests include recommendation system and data mining.



WANG Bing, born in 1984, Ph.D, associate professor. Her main research interests include recommendation system and data mining.