

基于嵌入学习的用户动态偏好预测

温 雯¹ 林泽钊¹ 蔡瑞初¹ 郝志峰^{1,2} 王丽娟¹

(广东工业大学计算机学院 广州 510000)¹ (佛山科学技术学院数学与大数据学院 广东 佛山 528000)²

摘 要 传统的刻画用户偏好的方法主要着眼于用户的长期兴趣,然而在现实应用中,用户兴趣随着时间迁移而不断变化,如何挖掘用户在时序上的动态偏好仍然面临挑战。为此,文中提出了一种基于嵌入学习的动态行为预测方法。首先,利用改进的词嵌入模型从用户的点击行为序列中学习获得每一个点击项的低维向量表示;然后,基于所学习的向量表示,结合用户近期点击行为推断用户的动态偏好,进而预测其下一步的点击行为。在两个真实数据集上将提出的方法与近年出现的其他基准方法进行比较,结果表明,所提方法在预测准确率上具有明显的优势。

关键词 用户兴趣,时序行为,行为预测,嵌入,word2vec

中图分类号 TP181 **文献标识码** A **DOI** 10.11896/jsjcx.180901801

Predicting User's Dynamic Preference Based on Embedding Learning

WEN Wen¹ LIN Ze-tian¹ CAI Rui-chu¹ HAO Zhi-feng^{1,2} WANG Li-juan¹

(School of Computers, Guangdong University of Technology, Guangzhou 510000, China)¹

(School of Mathematics and Big Data, Foshan University, Foshan, Guangdong 528000, China)²

Abstract Traditional methods for capturing user preferences mainly focus on user's long-term preferences. However, user interests always change over time in real-world applications. As a result, how to capture user's dynamic preferences still remains a big challenge. This paper proposed an embedding-based approach for predicting user's dynamic preferences. Firstly, an improved embedding method is used for learning the low-dimensional vector representations of items from user's click sequences. Then, based on the learned item vectors and user's short-term click behaviors, user's dynamic preferences are obtained and used for predicting the next click. Experiments were conducted on two real-world datasets and the proposed method was compared with state-of-the-art methods. The results demonstrate the significant superiority of the proposed method in prediction accuracy compared with other algorithms.

Keywords User preferences, Temporal behaviors, Behavior prediction, Embedding, Word2vec

1 引言

近年来,随着信息技术的发展,推荐系统的应用越来越广泛,各大社交网络和电商平台都倾向于对用户进行个性化推荐。在这些平台中,许多用户行为拥有时序信息,且用户的点击行为在时间上呈现出一定的相关性。因此,通过分析用户的历史行为来准确地刻画用户的行为模式,以对用户未来的行为进行预测,该方法已成为推荐系统领域的一个研究范式^[1-2]。

用户的行为模式受多种因素影响,且不同用户的行为模式不同,比如,在时间维度上,用户的行为不仅与远期行为相关,更可能与近期行为相关^[3]。因此,如何通过用户不同时间的行为来刻画用户的兴趣偏好,是一个巨大的挑战。

传统的协同过滤方法主要通过用户对用户远期的行为进行建

模,来获得对未来决策的预测^[4]。然而,这类方法没有考虑到不同时期的用户行为对用户未来行为的影响程度是不同的;换言之,用户行为偏好的改变并没有被考虑。然而,在真实场景中,用户的兴趣会随着时间而改变,从直觉上看,一个用户近期的行为应该比远期的行为对用户兴趣产生更大的影响^[5]。由于用户近期行为随着时间不断变化,因此在模型中融合近期行为信息对用户偏好进行动态刻画并非是一件轻而易举的事情。

另外,随着词嵌入模型在自然语言处理领域取得成功,该模型被越来越多地应用于其他领域。在推荐系统中,用户行为的日志是根据时间排序的,一段时间内的用户行为构成一个行为序列,可以将序列类比为自然语言处理的文档,序列中每个行为类比为自然语言处理的词,因此将神经网络词嵌入模型应用于用户点击行为具有很大的可行性。神经网络词嵌

到稿日期:2018-09-27 返修日期:2019-03-14 本文受国家自然科学基金(61472089),NSFC-广东联合基金(U1501254)资助。

温 雯(1981-),女,博士,副教授,CCF会员,主要研究方向为机器学习、模式识别、信息检索;林泽钊(1993-),男,硕士生,主要研究方向为数据挖掘、机器学习、信息检索,E-mail:906268746@qq.com(通信作者);蔡瑞初(1983-),男,博士,教授,CCF高级会员,主要研究方向为数据挖掘、机器学习、信息检索;郝志峰(1968-),男,博士,教授,CCF会员,主要研究方向为机器学习、人工智能;王丽娟(1978-),女,博士,副教授,主要研究方向为机器学习、高维数据聚类分析。

入模型的基本思想是将节点映射到一个稠密而低维的向量空间中,并在映射过程中尽量保留网络信息,其中包括每个节目之间的结构特征,从而解决了用户行为的高维度和稀疏性等问题^[6-7]。

本文主要关注如何融合近期行为信息对用户偏好进行建模和预测这一问题,提出了一种基于嵌入学习的用户动态点击行为预测方法。该方法在神经网络语言模型的基础上进行改进,结合用户的历史行为将节目特征投影到低维稠密向量中,不仅解决了节目特征维度过高的问题,也考虑了用户偏好动态变化的问题。在对用户行为进行学习之后,得到了每个节目的向量表示,并根据节目向量的特征相似度对用户的近期行为进行预测。在两个数据集上的实验验证了本文所提出的方法优于现有方法。

本文第 2 节介绍了相关工作,包括基于历史行为挖掘用户偏好的传统方法和基于神经网络的方法;第 3 节详细介绍了本文提出的基于嵌入学习的用户动态点击行为预测方法,并给出了复杂度分析;第 4 节在两个数据集上对本文所提算法的有效性进行了验证,并报告了相关实验结果;最后对本文的工作进行了总结和展望。

2 相关工作

本节首先介绍用户偏好挖掘的相关工作,然后简述基于嵌入学习的相关方法及近期该类方法在推荐领域的代表性工作。

2.1 用户偏好挖掘方法

备受关注的用户偏好挖掘方法主要是基于时间动态性的。在传统的推荐系统中,如音乐、电影、新闻等场景,时间因素被广泛使用。在现有的研究成果中,有学者假设了用户兴趣会随着时间递减,即近期行为对用户未来行为的影响较大^[5,8]。Ding 等^[5]提出了一种基于协同过滤方法的时间加权方法,通过使用一个衰减因子来减小对旧数据的权重,在进行推荐时,用户的目标项与该用户先前评分过的项目之间的相似度随着时间距离的增加而降低。Xiang 等^[3]认为最新的数据并不是唯一重要的,用户的行为也可能受旧数据的影响,因此提出了一种基于会话的时间图模型(STG),该图模型结合了用户的长期和短期偏好进行建模;基于 STG 模型框架,他们又提出了一种注入偏好融合(IPF)推荐算法,并扩展了个性化的随机游走以进行推荐。随后,Yu 等^[9]通过引入主题信息将 STG 模型扩展为一个名为 Topic-STG 的混合模型,并使用该模型来刻画用户偏好以进行推特推荐。最近,Nzeko' O 等^[10]提出了一种基于内容的 STG,STG 框架默认随机游走到达任何一个会话的权重是相同的,但在现实场景中,一周前的行为应该比一年前的行为的影响力更大,因此作者为每一条边设置一个根据时间衰减的权重,即为最近的行为对应的边赋予更大的权重,而为久远的行为对应的边赋予更低的权重,该模型取得了比以往研究更好的推荐效果。

除了时间加权方法和结合长短期用户偏好之外,许多成功的时间协同过滤方法主要基于潜在因子模型。例如,Koren^[11]通过矩阵分解对时间动态进行了建模,提出了能够有效挖掘用户偏好的局部变化的 timeSVD++方法,其准确地

预测了 Netflix 的电影评分。Liu 等^[12]提出了一种在线进化协同过滤方法来追踪用户的兴趣随着时间改变的情况,以达到实时推荐的目的。该方法通过结合时间信息对基于邻域的算法进行了扩展,并提出了一种使用新数据更新邻域相似性的增量算法。然而,现有的大多数时间推荐算法用于评分预测任务而不是 top-N 推荐任务,但在推荐系统中,我们往往更希望能够得到最满意项目的一个排序,即 top-N 推荐。

在推荐系统中,主题模型也常被用来挖掘用户兴趣。主题模型是一类从大量文档中挖掘主题结构的有效方法,其中的主题是潜在的变量,类似于用户行为推荐中的用户兴趣,因此其同样适用于用户行为推荐系统。传统的主题模型无法处理时间信息,为了结合时间信息来挖掘用户偏好,许多考虑主题随着时间演变的模型被提出。例如,文献^[13-14]通过对比连续时间片中建模的主题来研究文本中主题的演变。随后,其他的跟随时间变化的主题模型纷纷被提出,例如动态主题模型(DTM)^[15]、MAP-PLSA^[16]、在线 LDA^[17]等。2012 年,Diao 等^[18]提出了 TimeUserLDA 模型,并利用该模型从微博中寻找热门话题。虽然该模型假设了用户行为受到用户兴趣和热门主题共同的影响,但是在 TimeUserLDA 中只有一个共享的基础主题,未区分基于面向用户的主题和面向时间的主题,混淆了用户兴趣和时间上下文。为了解决这个问题,Yin 等^[19]提出了一种针对社交媒体系统用户评分行为的 TCAM 模型,但该模型认为用户的兴趣是固定的。然而,在现实中用户的兴趣并不总是固定的,而是可能会随着时间的推移而改变。随后,他们将 TCAM 模型扩展为一个动态版本(DTCAM)^[20],该模型能够刻画用户的动态兴趣。

2.2 基于嵌入学习的用户行为分析方法

在许多自然语言处理的应用场景中,包括信息检索、词性标注等应用,特定的目标都可以被概括为求一序列单词的概率的任务,语言模型就是在该背景下被提出的^[21-22]。传统的语言模型使用 one-hot 表示——每个单词用一个维度大小与词典大小相同的向量表示,其中与单词相关的一个维度为 1,其余维度为 0。然而,这种方法存在很大的缺点——高维度的向量和严重的数据稀疏性。为了解决这个问题,神经语言模型被提出。神经语言模型使用低维、稠密的向量来代替高维、稀疏的 one-hot 向量,不仅大大降低了维度,而且还能够通过不同词在低维空间中的距离来刻画词之间的相似性^[23-24]。然而,即使基于神经网络的模型解决了高维度的问题,但是其训练代价仍然十分庞大,因为在实际任务中,词汇量可能会达到数百万甚至更多。为此,Mikolov 等^[25-26]提出了具有高度可扩展的 CBOW 模型和 Skip-gram 模型,这两种模型完美地解决了大词汇量的问题,并且可以挖掘文本中单词之间的语法和语义关系,在大量的 NLP 任务中取得了良好的结果。

近年来,随着神经网络词嵌入模型在自然语言领域取得了巨大的成功,越来越多的学者将其迁移到推荐算法中。Barkun 等^[6]通过将 SGNS 的算法思路迁移到基于物品的协同过滤上,把商品的共现性作为自然语言中的上下文关系,构建了神经网络,学习出了商品的低维特征向量表示。Grbovic 等^[7]则提出了一种 Prod2Vec 算法,该算法类比于自然语言的

句子,使用滑动窗口获取每个目标物品的上下文,通过模型同时训练每一个目标物品及其上下文的向量表示。通过训练每一对(目标物品,上下文)来学习各自的向量表示。在这个模型中,作者假设了目标物品后面的点击行为也是已知的,但一般来说,用户的点击行为具有时间顺序,因此发生在目标行为后面的点击行为在预测阶段其实是无法获取的。

3 基于嵌入学习的动态推荐方法

本文提出了一种基于嵌入学习的动态推荐方法(Dynamic Recommendation based on Embedding Learning, DREL),该方法能够利用用户的短期行为对用户偏好进行建模。为了更好地阐述所提出的模型及算法,我们首先给出了问题的形式化描述及相关符号约定,如表1所列。

表1 符号及其定义

Table 1 Notations and their definitions

符号	定义
T	节目集合, $ T $ 表示节目的数量
t_i	节目集中的第 i 个节目, $t_i \in T$
S	用户点击行为序列集
s	点击序列集的单个序列, $s \in S$
C_i	节目 t_i 的上下文
d	向量的维度
Φ_i	t_i 作为目标项的向量表示, $\Phi_i \in R^d$
Ψ_i	t_i 作为上下文的向量表示, $\Psi_i \in R^d$

嵌入学习的目标是:给定一个点击序列,通过对序列中每个节目 t_i 及其上下文 C_i 进行学习,将每个节目 t_i 映射为一个低维的特征向量 $t_i \in R^d$,其中 $d \ll |T|$,且要求如果两个节目是相似的,那么它们被映射成向量后的距离也是接近的。

3.1 模型描述

用户点击行为预测的目的是根据用户的历史点击行为预测下一个点击行为,然而用户往往拥有着大量的历史行为,因此研究的重点在于挖掘用户的下一个行为与哪些历史行为的相关程度较高。由于用户的点击行为在时间上的分布是不均匀的,因此本文舍弃了以往完全使用时间作为用户偏好变化影响因子的做法,而采用行为窗口(即一个行为窗口中包含若干个行为)来对动态点击行为进行建模,这意味着我们不用考虑点击行为的具体时间,只须考虑下一个点击行为的前面若干个行为。问题定义如下:

给定一个播放列表 $s = \{t_1, t_2, \dots, t_n\}$,其中 $s \in S, t_i \in T$ 为在列表 s 中根据时间排序的第 i 个节目。对于序列中的每一个节目 t_i ,使用窗口 w 来获取其上下文 $C_i = \{t_{i-1}, t_{i-2}, \dots, t_{i-w}\}$ 。本文假设越靠近目标节目的点击行为与目标节目的相关性越大。记 $\Phi_i \in R^d$ 为 t_i 的低维向量表示,类似于自然语言处理中的 CBOW 模型,可以通过优化每一个 $\langle C_i, t_i \rangle$ 样本的联合概率(式(1))获得 Φ_i 的表达:

$$\arg \max_{\Phi_i, \Psi_{C_i}} \prod_{s \in S} \prod_{t_i \in s} p(t_i | C_i) \quad (1)$$

我们使用 softmax 函数来刻画 $p(t_i | C_i)$,如式(2)所示:

$$p(t_i | C_i) = \frac{\exp(\Psi_{C_i}^T \cdot \Phi_i)}{\sum_{t_j \in T} \exp(\Psi_{C_i}^T \cdot \Phi_j)} \quad (2)$$

其中, Ψ_{C_i} 为上下文权重特征,因此:

$$p(t_i | C_i) \propto \exp(\Psi_{C_i} \cdot \Phi_i) \quad (3)$$

针对 C_i ,我们分别设计了两种方案来获取其对应的输入。

方案1(DREL-1):如图1所示,对于每个样本的上下文中的每一个节目 t_j ,将其映射到对应的输入向量表示 Ψ_j ,对所有 Ψ_j 进行求和后将其作为上下文的整体向量。对目标行为的向量 Φ_i 进行训练,如式(4)所示:

$$p(t_i | C_i) \propto \exp\left(\left(\sum_{t_j \in C_i} \Psi_j\right)^T \cdot \Phi_i\right) \quad (4)$$

直观的解释如下:目标节目的特征与其上下文整体具有相关性,对于每一次预测,上下文对目标节目的影响是整体性的。

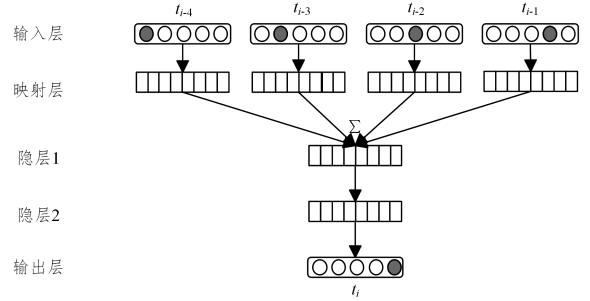


图1 DREL-1网络图

Fig. 1 Network diagram of DREL-1

方案2(DREL-2):如图2所示,与DREL-1类似,首先将每个 $t_j \in C_i$ 映射到对应的输入向量表示 Ψ_j ,随后分别使用每个 Ψ_j 单独训练目标节目向量 Φ_i ,其中 $p(t_i | C_i)$ 如式(5)所示:

$$p(t_i | C_i) \propto \prod_{t_j \in C_i} \exp(\Psi_j^T \cdot \Phi_i) \quad (5)$$

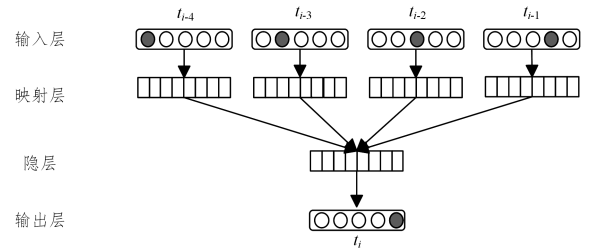


图2 DREL-2网络图

Fig. 2 Network diagram of DREL-2

直观的解释如下:上下文中各个节目独立地影响目标节目,可以通过将上下文中每个节目对目标节目的预测概率相乘来获得预测结果。

由于式(2)中的计算归一化项的时间复杂度太高,本文利用文献[19]提出的负采样方法进行优化,即通过最大化如下转换函数来学习节目的嵌入表示向量:

$$\Delta_i = \log \sigma(\Psi_{C_i}^T \cdot \Phi_i) + \sum_{k=1}^s E_{t_r \sim P_n(t)} [\log \sigma(\Psi_{C_i}^T \cdot \Phi_r)] \quad (6)$$

其中, $\sigma(x) = \frac{1}{1 + \exp(-x)}$ 为 sigmoid 函数, $t_r \sim P_n(t)$ 表示依据分布 $P_n(t)$ 来采样上下文 Ψ_{C_i} 的负样本 t_r 。本文定义 $P_n(t) \propto \text{counter}(t)^{0.75}$,其中 $\text{counter}(t)$ 为节目 t 在样本集中出现的频数,每个上下文均采样 s 个负样本。为此,我们将模型的目标函数定义为:

$$L = - \sum_{s \in S} \sum_{t_i \in s} \Delta_i \quad (7)$$

为了简化模型,对式(7)进行改写:

$$L = - \sum_{(t_j, C_i, \mu) \in D} \log \sigma(\mu \Psi_{C_i}^T \cdot \Phi_j) \quad (8)$$

其中,样本集合 D 包含了所有的正负样本; μ 为标志符号,当 $\mu=1$ 时, t_j 是 C_i 的正样本,当 $\mu=-1$ 时, t_j 是 C_i 的负样本,即通过负采样得到的,每个正样本都对对应着 s 个负样本。

通过该目标函数,可以针对每一个到来的样本进行训练,从而学习得到一个能够精准预测的模型。

3.2 用户行为预测

如上文所述,本文模型可以从用户的点播序列数据中学习每一个节目的向量表示,这些向量适用于建模用户对节目的上下文偏好。在进行推荐时,给定一组上下文序列 $C_i = \{t_1, t_2, \dots, t_w\}$,对应 3.1 节中的模型,我们依旧使用两种方案进行推荐。当训练方案为 DREL-1 时,将 $\Psi_{C_i} = \sum_{t_j \in C_i} \Psi_j$ 作为上下文的输入权重,并使用 softmax 计算每一个候选项的概率:

$$p(t_i | C_i) = \frac{\exp(\Psi_{C_i}^T \cdot \Phi_i)}{\sum_{t_k \in T} \exp(\Psi_{C_i}^T \cdot \Phi_k)} \quad (9)$$

计算每一个 $t_i \in T$ 的概率,并对其进行排序,最后得到概率最高的 top-N 个节目作为预测列表进行推荐。

当训练方案为 DREL-2 时,每一个候选项的概率为:

$$p(t_i | C_i) \propto \prod_{t_j \in C_i} \exp(\Psi_j^T \cdot \Phi_i) \quad (10)$$

由于我们的目标是得到概率最高的 top-N 个节目,为了计算简便,可以使用对数对其进行转换。

$$p(t_i | C_i) \propto \sum_{t_j \in C_i} \log \exp(\Psi_j^T \cdot \Phi_i) = \sum_{t_j \in C_i} \Psi_j^T \cdot \Phi_i \quad (11)$$

因此,我们直接使用 $\sum_{t_j \in C_i} \Psi_j^T \cdot \Phi_i$ 对每一个 $t_i \in T$ 进行排序,最终得到概率最高的 Top-N 个节目进行推荐。

3.3 负采样及目标函数优化

本节将着重介绍 DREL 算法的细节和具体实现。针对 3.1 节所述的模型,DREL 算法的设计主要包括 3 个主要模块:数据预处理、负采样和目标函数优化。DREL 算法的整体流程如下:首先对数据集进行预处理,主要是对数据进行清洗,并将数据集划分为训练集与测试集;然后利用负采样方法^[27]获得式(6)所涉及的正负样本集,进而计算目标函数(式(8));最后利用随机梯度下降方法对目标函数进行优化,从而得到训练模型。下文将详细阐述负采样和目标函数优化的过程。

3.3.1 负采样

由于式(2)中归一化项的时间复杂度很高,本文利用负采样方法对其进行优化。首先,对训练集中的每一个序列进行遍历,在序列中遍历每一个节目,将遍历到的当前节目作为目标行为。对于每个目标行为,均向前回溯生成长度为 θ_1 的上下文。对于方案 DREL-1,将上下文作为一个整体与目标节目组成正例样本加入正样本集合 D_+ ;而方案 DREL-2 则将上下文的每一项分别与目标节目组成正例样本加入正样本集合 D_+ 。然后,针对 D_+ 中的每一个正样本 $(C_i, t_i, 1)$,采用负采样的方法从节目集中采样 θ_2 个节目与 C_i 组成负样本 $(C_i, t_r, -1)$,其中 t_r 服从 P_n 分布且 $t_r \neq t_i$,并将其加入负样本集合 D_- 。最后,我们得到了整个训练样本集合: $D = D_+ \cup D_-$ 。利用负采样方法生成目标函数(式(8))中样本集合 D 的过程如算法 1 所示。

算法 1 负采样

Input:序列集 S ,滑动窗口大小 θ_1 ,每个正样本对应的负样本数量 θ_2

Output:训练数据集 D

Step1 $D_+ \leftarrow \emptyset; D_- \leftarrow \emptyset; D \leftarrow \emptyset;$

Step2 $i \leftarrow 0;$

Step3 for $s \in S$ do:

Step4 for $i < |s|$ do:

Step5 根据上下文滑动窗口大小 θ_1 ,得到 t_i 的上下文 C_i

Step6 将正样本 $(C_i, t_i, 1)$ 加入 D_+ 中(DREL-1) or 将正样本集合 $\{(t_j, t_i, 1) | t_j \in C_i\}$ 加入 D_+ 中(DREL-2)

Step7 end for

Step8 end for

Step9 for $(C_i, t_i, 1) \in D_+$ do:

Step10 将负采样集合 $\{(C_i, t_r, -1) | t_r \sim P_n, t_r \neq t_i\}^{\theta_2}$ 加入 D_- 中

Step11 end for

Step12 return $D = D_+ \cup D_-$.

3.3.2 目标函数优化

本文使用随机梯度下降方法对目标函数(式(8))进行优化^[26],其中 Ψ_{C_i} 与 Φ_i 是待优化参数,需要同时更新。其梯度计算如式(12)和(13)所示:

$$\frac{\partial L}{\partial \Psi_{C_i}} = \frac{\partial \log \sigma(\mu \Psi_{C_i}^T \Phi_j)}{\partial \Psi_{C_i}} = \mu \Phi_j (1 - \sigma(\mu \Psi_{C_i}^T \Phi_j)) \quad (12)$$

$$\frac{\partial L}{\partial \Phi_j} = \frac{\partial \log \sigma(\mu \Psi_{C_i}^T \Phi_j)}{\partial \Phi_j} = \mu \Psi_{C_i} (1 - \sigma(\mu \Psi_{C_i}^T \Phi_j)) \quad (13)$$

最终,当目标函数达到收敛时就得到了 Ψ 和 Φ 两组向量。目标优化过程如算法 2 所示。首先随机初始化,然后使用训练样本集合 D 来更新 Ψ 和 Φ 中的向量,直到算法收敛。

算法 2 目标函数优化

Input:训练数据集 D ,节目向量维度 d ,梯度更新步长 μ

Output:上下文表示向量 $\Psi \in \mathbb{R}^{|\mathcal{T}| \times d}$,目标行为表示向量 $\Phi \in \mathbb{R}^{|\mathcal{T}| \times d}$

Step1 随机初始化上下文表示向量 Ψ ,零化目标行为表示向量 Φ

Step2 $i \leftarrow 0$

Step3 while 不收敛 do:

Step4 for each sample (C_i, t_j, μ) in D :

Step5 $\Phi_i \leftarrow \Phi_i + \mu \frac{\partial L}{\partial \Phi_i}$

Step6 $\Psi_k \leftarrow \Psi_k + \mu \frac{\partial L}{\partial \Psi_k}, t_k \in C_i$

Step7 end for

Step8 end while

3.4 时间复杂度分析

算法 1 假设每个序列的平均长度为 l 。对于每个目标行为,若是方案 DREL-1,则产生 1 个正样本,若是方案 DREL-2,则产生 θ_1 个正样本,由于针对不同的目标行为,其动态窗口大小不同,因此对于每个正样本,均采样 θ_2 个负样本加入训练集。由于生成每个正样本和负样本的时间复杂度为 $O(1)$,因此方案 DREL-1 生成所有正样本的时间复杂度为 $O((1 + \theta_2) * l * |S|)$,方案 DREL-2 生成所有正样本的时间复杂度为 $O((1 + \theta_2) * \theta_1 * l * |S|)$ 。以上时间复杂度正比于用户点击行为总数 $l * |S|$ 。

算法 2 的输入为算法 1 生成的所有样本点,对于每次迭代,均需要计算每个样本点的梯度。计算梯度的时间复杂度为 $O(1)$,假设对训练集 D 最多迭代 θ_3 次,则方案 DREL-1 计算梯度的时间复杂度为 $O((1 + \theta_2) * \theta_3 * l * |S|)$,方案 DREL-2 计算梯度的时间复杂度为 $O((1 + \theta_2) * \theta_3 * \theta_1 * l * |S|)$ 。

|S|)。因此,方案 DREL-2 的时间复杂度为方案 DREL-1 的时间复杂度的 θ_1 倍。

4 实验

4.1 实验设定

本文使用两个数据集来验证提出的假设,分别是来自 Last.fm 的公开数据集¹⁾和国内某运营商网络电视数据集(下文简称“TV”)。数据集信息如下:Last.fm 数据集的结构是用户生成的播放列表,其中一个播放列表为一个用户一天的点击日志。由于 Last.fm 数据跨度较大,为了得到用户活跃度较大的数据,我们进一步从数据中选择了 2009 年 1 月 1 日到 2009 年 4 月 31 日的的数据,得到了包含 759 个用户、80 051 首歌、40 416 个播放列表,总共 1 539 426 个用户点击项的数据。我们采用统一的条件对 IPTV 从 2015 年 10 月 7 日到 2015 年 11 月 9 日的点播数据集进行筛选,最终得到了包含 9 935 个用户、49 871 个节目、156 819 个播放列表,总共 2 774 118 个点击项的数据集。两个数据集的统计信息如表 2 所列。

表 2 数据集统计信息

Table 2 Statistical information of dataset

数据集	#用户	#节目	#点播放列表	#点击项总数
Last.fm	759	80 051	40 416	1 539 426
TV	9 935	49 871	156 819	2 774 118

4.2 评价指标及对比算法

4.2.1 评价指标

我们使用 Hit Ratio at N(HR@N)和 Mean Reciprocal Rank at N(MRR@N)^[28]作为预测模型的评价指标。

1)HR@N

$$HR@N = \frac{1}{|D|} \sum_{d \in D} I(d, R(d)) \quad (14)$$

$$I(d, R(d)) = \begin{cases} 1, & d \in R(d) \\ 0, & d \notin R(d) \end{cases} \quad (15)$$

其中, D 为测试样本集; $R(d)$ 表示针对 d 的 top- K 预测结果; $I(d, R(d))$ 表示预测正确则置 1,否则置 0。其直观含义为:如果目标词出现在 top- K 推荐列表中,则认为预测正确,否则认为预测错误。

2)MRR@K

$$MRR@K = \frac{1}{|D|} \sum_{d \in D} RR_d \quad (16)$$

$$RR_d = \begin{cases} \frac{1}{rank_d}, & d \in R(d) \\ 0, & d \notin R(d) \end{cases} \quad (17)$$

其中, D 为测试样本集; $rank_d$ 表示样本 d 对应的结果出现在 top- K 推荐列表 $R(d)$ 中的排列位置; RR_d 表示样本 d 的评分,如果 d 出现在 $R(d)$ 中,则其评分为 $1/rank_d$,否则评分为 0。

与 HR@K 相比,MRR@K 更关注查询结果在推荐列表中的排序位置。

4.2.2 对比算法

本文采用了以下多种方法进行对比。

1)ItemKNN:根据目标项目与邻居项目的相似度进行推

荐,同样是推荐 top- N ^[29]。

2)Item2Vec:一种通过远期行为刻画用户偏好的方法^[6]。

3)Prod2Vec:一种基于 SGNS 算法的节目向量学习方法,其窗口取的是目标节点的前后点击行为,不仅包含了历史行为,还包括将未发生的点击行为^[7]。

ItemKNN 是最典型的一种推荐算法,大多数文献将 ItemKNN 作为 baseline 算法。Item2Vec 及 Prod2Vec 算法是近几年出现的与本文所提方法较为相近的算法。其中,Prod2Vec 是基于 Skip gram 的推荐算法,主要应用于商品推荐,并将购物序列的商品类比为“词”。因此,选择该算法作为本文的对比算法。

4.3 结果及分析

本文使用 4.1 节的两个数据集来验证本文所提模型的可行性。对于 Last.fm 数据集,将前 3 个月的数据作为训练集,将第 4 个月的数据作为测试集。对于 TV 数据,将前 26 天的数据作为训练集,将后面 8 天的数据作为测试集。在模型训练的过程中,使用随机梯度下降法来训练模型参数。针对每个正样本,我们都采样 $n(n=10)$ 个负样本加入训练集。对参数进行网络搜索寻优后发现,对于 Last.fm 和 TV,最优向量维度 d 均为 128,梯度下降学习率均为 0.1。具体参数设置如表 3 所列。

表 3 参数设定

Table 3 Parameters setting

数据集	n	d	μ
Last.fm	10	128	0.1
TV	10	128	0.1

为了寻找最优的历史行为窗口,分别使用了多组固定窗口进行实验。我们将使用 HR@K 和 MRR@K 指标分别从预测准确率和预测的排名质量两个角度来验证本文方法。

表 4—表 11 列出了在 Last.fm 和 TV 数据集上的 HR 值和 MMR 值(表中,粗体表示所有方法中得分最高的值,斜体表示对比方法中得分最高的值)。实验结果显示,本文提出的方法比所有对比算法的效果均更好,证明了短期动态窗口在刻画用户动态偏好上的有效性。

表 4 Last.fm 数据集固定窗口的 HR@10

Table 4 HR@10 of Last.fm dataset with fixed window

方法	1	2	3	5
ItemKNN	0.2872	0.2906	0.2836	0.2624
Item2Vec	0.2213	0.2284	0.2233	0.2081
Prod2Vec	0.3153	0.3054	0.3259	0.2853
DREL-1	0.3160	0.3287	0.3324	0.2872
DREL-2	0.3160	0.3312	0.3394	0.2999

表 5 Last.fm 数据集不同窗口的 HR@20

Table 5 HR@20 of Last.fm dataset with fixed window

方法	1	2	3	5
ItemKNN	0.3358	0.3553	0.3584	0.3222
Item2Vec	0.2911	0.3065	0.3023	0.2888
Prod2Vec	0.3450	0.3603	0.3649	0.3432
DREL-1	0.3504	0.3670	0.3655	0.3448
DREL-2	0.3504	0.3807	0.3807	0.3646

¹⁾ <https://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-1K.html>

表 6 TV 数据集不同窗口的 $HR@10$ Table 6 $HR@10$ of TV dataset with fixed window

方法	1	2	3	5
ItemKNN	0.8355	0.8476	0.8458	0.8154
Item2Vec	0.6175	0.5998	0.5671	0.4987
Prod2Vec	0.8387	0.8423	0.8475	0.8454
DREL-1	0.8423	0.8482	0.8524	0.8476
DREL-2	0.8423	0.8574	0.8569	0.8537

表 7 TV 数据集固定窗口的 $HR@20$ Table 7 $HR@20$ of TV dataset with fixed window

方法	1	2	3	5
ItemKNN	0.8520	0.8599	0.8605	0.8474
Item2Vec	0.7455	0.7469	0.7277	0.6790
Prod2Vec	0.8581	0.8614	0.8708	0.8637
DREL-1	0.8628	0.8618	0.8716	0.8706
DREL-2	0.8628	0.8739	0.8740	0.8772

表 8 Last. fm 数据集固定窗口的 $MRR@10$ Table 8 $MRR@10$ of Last. fm dataset with fixed window

方法	1	2	3	5
ItemKNN	0.1284	0.1291	0.1286	0.1119
Item2Vec	0.0783	0.0770	0.0726	0.0656
Prod2Vec	0.1469	0.1199	0.1038	0.0835
DREL-1	0.2347	0.1933	0.1610	0.1160
DREL-2	0.2347	0.2065	0.1696	0.1201

表 9 Last. fm 数据集固定窗口的 $MRR@20$ Table 9 $MRR@20$ of Last. fm dataset with fixed window

方法	1	2	3	5
ItemKNN	0.1319	0.1338	0.1333	0.1162
Item2Vec	0.0834	0.0826	0.0785	0.0715
Prod2Vec	0.1501	0.1237	0.1081	0.0894
DREL-1	0.2366	0.1961	0.1640	0.1200
DREL-2	0.2366	0.2093	0.1725	0.1237

表 10 TV 数据集固定窗口的 $MRR@10$ Table 10 $MRR@10$ of TV dataset with fixed window

方法	1	2	3	5
ItemKNN	0.3991	0.3774	0.3752	0.3380
Item2Vec	0.2315	0.1862	0.1692	0.1557
Prod2Vec	0.3521	0.3269	0.3017	0.2700
DREL-1	0.4514	0.4215	0.4223	0.4036
DREL-2	0.4514	0.4103	0.3932	0.3534

表 11 TV 数据集固定窗口的 $MRR@20$ Table 11 $MRR@20$ of TV dataset with fixed window

方法	1	2	3	5
ItemKNN	0.4015	0.3821	0.3842	0.3417
Item2Vec	0.2404	0.1891	0.1811	0.1646
Prod2Vec	0.3579	0.3316	0.3067	0.2782
DREL-1	0.4579	0.4260	0.4299	0.4073
DREL-2	0.4579	0.4149	0.3975	0.3591

根据表 4—表 11 可以得出以下结论:

1) 在 Last. fm 数据集上, DREL-2 模型的性能最好。以 HR 为评价标准, DREL-2 在窗口大小为 3 时的 HR 值最高, $HR@10$ 和 $HR@20$ 分别比最好的基准方法提高了 4.14% 和 4.33%。以 MRR 为评价标准, 得分最高的窗口为 1, 分别比最好的基准方法提高了 59.77% 和 57.63%。

2) 在 TV 数据集上, 以 HR 为评价标准, DREL-2 模型的

性能最好。 $HR@10$ 和 $HR@20$ 分别比最好的基准方法提高了 1.32% 和 1.45%。以 MRR 为评价标准, DREL-1 模型的性能最好, 得分最高的窗口为 1, 分别比最好的基准方法提高了 13.10% 和 14.51%。

3) 从整体来看, DREL 模型在预测准确率方面有着不错的提升效果, 且在排序质量上有着明显的优势。

4) 对于 DREL-1 和 DREL-2 模型, 不论是 Last. fm 还是 TV 数据集, 以 HR 为评价标准时, 均在窗口为 3 时性能最优; 以 MRR 为评价指标时, 均在窗口为 1 时取得最高的得分, 且在取得最优值后, 窗口越大, 得分越低。这说明通过短期历史行为预测用户的兴趣偏好是可行的。

结束语 本文提出了一种基于嵌入学习的用户动态点击行为预测方法, 与以往的推荐方法不同, 该方法舍弃了以往完全使用时间作为用户偏好变化影响因子的做法, 而采用行为窗口来对动态点击行为进行建模。首先, 利用词嵌入模型从点击行为序列中学习每一个节目的低维向量表示。本文采用两种方案学习节目的低维向量表示: 1) DREL-1 将上下文作为整体学习目标节点的表示向量; 2) DREL-2 使用上下文中的每个节目单独学习目标节目的表示向量。实验结果表明, 这两种方案的预测准确率和排序质量都比基准方法好。在未来的工作中, 我们将进一步考虑基于嵌入表示学习的更高效的刻画用户行为偏好的方法, 如通过用户的历史点击行为刻画用户的时间周期模式, 挖掘更详细的用户画像, 并执行更丰富的任务, 如用户聚类、个性化推荐等。

参 考 文 献

- [1] XU H L, WU X, LI X D, et al. Comparison Study of Internet Recommendation System[J]. Journal of Software, 2009, 20(2): 350-362. (in Chinese)
许海玲, 吴潇, 李晓东, 等. 互联网推荐系统比较研究[J]. 软件学报, 2009, 20(2): 350-362.
- [2] ADOMAVICIUS G, TUZHILIN A. Context-Aware recommender systems [C]//Proceedings of the RecSys 2008. New York: ACM Press, 2008: 335-336.
- [3] XIANG L, YUAN Q, ZHAO S, et al. Temporal recommendation on graphs via long-and short-term preference fusion[C]//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2010: 723-732.
- [4] LINDEN G, SMITH B, YORK J. Amazon. com recommendations: Item-to-item collaborative filtering [J]. IEEE Internet Computing, 2003, 7(1): 76-80.
- [5] DING Y, LI X. Time weight collaborative filtering [C]//Proceedings of the 14th ACM international conference on Information and knowledge management. ACM, 2005: 485-492.
- [6] BARKAN O, KOENIGSTEIN N. Item2vec: neural item embedding for collaborative filtering [C]//2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2016: 1-6.
- [7] GRBOVIC M, RADOSAVLJEVIC V, DJURIC N, et al. E-commerce in your inbox: Product recommendations at scale [C]//Proceedings of the 21th ACM SIGKDD International Conference

- on Knowledge Discovery and Data Mining. ACM, 2015; 1809-1818.
- [8] LIU J, DENG G. Link prediction in a user-object network based on time-weighted resource allocation[J]. *Physica A: Statistical Mechanics and its Applications*, 2009, 388(17): 3643-3650.
- [9] YU J, SHEN Y, YANG Z. Topic-STG: Extending the session-based temporal graph approach for personalized tweet recommendation[C] // *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 2014; 413-414.
- [10] NZEKO'O A J N, TCHUENTE M, LATAPY M. Time Weight Content-Based Extensions of Temporal Graphs for Personalized Recommendation[C] // *WEBIST 2017-13th International Conference on Web Information Systems and Technologies*, 2017.
- [11] KOREN Y. Collaborative filtering with temporal dynamics [C] // *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2009; 447-456.
- [12] LIU N N, ZHAO M, XIANG E, et al. Online evolutionary collaborative filtering[C] // *Proceedings of the fourth ACM Conference on Recommender Systems*. ACM, 2010; 95-102.
- [13] MEI Q, ZHAI C X. Discovering evolutionary theme patterns from text: an exploration of temporal text mining[C] // *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. ACM, 2005; 198-207.
- [14] WANG X, ZHAI C X, HU X, et al. Mining correlated bursty topic patterns from coordinated text streams[C] // *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2007; 784-793.
- [15] BLEI D M, LAFFERTY J D. Dynamic topic models[C] // *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 2006; 113-120.
- [16] GOHR A, HINNEBURG A, SCHULT R, et al. Topic evolution in a stream of documents[C] // *Proceedings of the 2009 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2009; 859-870.
- [17] ALSUMAIT L, BARBARA D, DOMENICONI C. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking[C] // *Eighth IEEE International Conference on Data Mining, 2008(ICDM'08)*. IEEE, 2008; 3-12.
- [18] DIAO Q, JIANG J, ZHU F, et al. Finding bursty topics from microblogs[C] // *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012; 536-544.
- [19] YIN H, CUI B, CHEN L, et al. A temporal context-aware model for user behavior modeling in social media systems[C] // *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. ACM, 2014; 1543-1554.
- [20] YIN H, CUI B, CHEN L, et al. Dynamic user modeling in social media systems[J]. *ACM Transactions on Information Systems (TOIS)*, 2015, 33(3): 10.
- [21] BAEZA-YATES R, RIBEIRO-NETO B. *Modern information retrieval*[M]. New York: ACM press, 1999.
- [22] LAVRENKO V, CROFT W B. Relevance-based language models[J]. *ACM SIGIR Forum*, 2017, 51(2): 260-267.
- [23] TURIAN J, RATINOV L, BENGIO Y. Word representations: a simple and general method for semi-supervised learning[C] // *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2010; 384-394.
- [24] COLLOBERT R, WESTON J, BOTTOU L, et al. Natural language processing (almost) from scratch[J]. *Journal of Machine Learning Research*, 2011, 12(8): 2493-2537.
- [25] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[J]. *arXiv: 1301.3781*, 2013.
- [26] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality [M] // *Advances in Neural Information Processing Systems*. Berlin: Springer, 2013; 3111-3119.
- [27] GOLDBERG Y, LEVY O. word2vec Explained: deriving Mikolov et al. 's negative-sampling word-embedding method [J]. *arXiv: 1402.3722*, 2014.
- [28] RADEV D R, QI H, WU H, et al. Evaluating web-based question answering systems[C] // *Proceedings of the 3rd International Conference on Language Resources and Evaluation*. 2002.
- [29] CHOI K, SUH Y. A new similarity function for selecting neighbors for each target item in collaborative filtering [J]. *Knowledge-Based Systems*, 2013, 37(1): 146-153.