

一种融合注意力和记忆网络的序列推荐算法^{*}

陈细玉, 林 穗

(广东工业大学 计算机学院, 广东 广州 510006)

摘 要:传统基于协同过滤和矩阵分解的静态表示推荐算法,不能很好地体现用户的动态兴趣。循环神经网络能够进行序列推荐,但存在序列之间的长距离依赖性差、各项目的区分度差等问题。由此提出一种融合注意力和记忆网络的序列推荐算法,根据 Word2vec 算法,引申 item2vec 和 user2vec,初始化用户和项目的固定表示嵌入矩阵,通过结合注意力机制和长短期记忆网络(Long Short-Term Memory, LSTM)解决序列之间的长距离依赖性差和区分度差问题。利用记忆网络获取用户的动态邻居,加强用户的动态表示,实现更准确的推荐。通过在 MovieLens 数据集上的实验结果表明,所提出的算法相比其他算法推荐效果显著提高。

关键词:注意力;记忆网络;LSTM;序列推荐

中图分类号:TP391

文献标识码:A

DOI: 10.19358/j.issn.2096-5133.2020.05.008

引用格式:陈细玉,林穗.一种融合注意力和记忆网络的序列推荐算法[J].信息技术与网络安全,2020,39(5):37-41.

A sequence recommendation algorithm combining attention and memory networks

Chen Xiyu, Lin Sui

(School of Computer, Guangdong University of Technology, Guangzhou 510006, China)

Abstract:The traditional static representation recommendation algorithm based on collaborative filtering and matrix decomposition can not well reflect the dynamic interest of users. The recurrent neural network can recommend sequences, but there are some problems such as poor long-distance dependence between sequences, poor discrimination of items between sequences and so on. Therefore, this paper proposes a sequence recommendation algorithm combining attention and memory networks. According to the word2vec algorithm, item2vec and user2vec are extended to initialize the fixed representation embedding matrix of users and items. The problems of poor long-distance dependence between sequences and poor discrimination of items between sequences are resolved by combining attention mechanisms and the Long Short-Term Memory (LSTM). The memory network is used to obtain the user's dynamic neighbors, which can strengthen the user's dynamic representation, and achieve more accurate recommendations. The experimental results on the MovieLens dataset show that the proposed algorithm significantly improves the recommendation effect compared to other algorithms.

Key words: attention; memory network; long short-term memory (LSTM); sequential recommendation

0 引言

协同过滤(Collaborative Filtering, CF)^[1],分为基于用户的协同过滤(UBCF)、基于项目的协同过滤(IBCF)和基于模型的协同过滤。基于用户的协同过滤是根据用户的历史行为计算出用户之间的相似度,将相似度高的用户的历史项目推荐给该用户;基于项目的协同过滤是计算项目之间的相似度,推荐与该用户历史项目相似度高且历史行为中

没有的项目。基于模型的协同过滤如矩阵分解(Matrix Factorization, MF)^[2]通过用户和项目的低维特征向量预测评分进行推荐;EBESU T^[3]等人提出协同记忆网络,使用记忆网络学习用户-项目对的特定邻居,用户和项目记忆共同利用邻居来产生排名分数。这些模型通常对静态的用户和项目交互进行建模,所表示的用户偏好是静态的。

但在现实生活中,用户的历史行为是动态的,人们的兴趣是随着时间变化的。早先的基于马尔科夫链的序列推荐^[4]将项目之间的转移矩阵分解为两个低秩矩阵,此序列推荐取得了良好的效果,

^{*} 基金项目:广州市科技计划项目(2017010160012);广州市科技计划项目(201902020016)

然而 k 阶马尔科夫链只能根据有限的前 k 个行为预测下一个行为。近年来基于循环神经网络 (Recurrent Neural Network, RNN) 的序列推荐能够依赖更多用户历史行为。FU H 等人^[5]使用双向长短期记忆网络 (Bi-LSTM) 捕获项目的序列特征,取得了更好的推荐效果。下一个项目的预测并不取决于用户所有的历史项目,只与部分有关联,且对当前的预测影响是不一致的。LUO A 等人^[6]提出了自适应注意感知门控循环单元 (3AGRU),采用注意力机制来适应用户顺序偏好的表示。LV F 等人^[7]提出了一种新的顺序深度匹配 (SDM) 模型,使用门控融合模块结合长期和短期偏好特征,来获取用户的动态偏好。LIN X 等人^[8]将 K 最近邻算法 (K -Nearest Neighbor, KNN) 与 RNN 相结合,利用动态邻居序列优化用户序列表示。

item2vec^[9]是 BARKAN O 等人借鉴自然语言处理中的 Word2vec 词向量表示方法,学习项目的向量表示并计算项目的相似度进行推荐。

RNN 存在长距离依赖问题,且项目之间的影响主要取决于相对位置而不是项目的属性。LSTM 增加遗忘门和更新门优化了长距离依赖问题,但仍旧没有彻底解决。注意力机制能从全局模式上既解决长距离依赖问题,又能解决序列中各项目的区分度差问题。实际生活中,短期偏好的用户会向资深固定偏好的用户靠拢,兴趣变化也会向这些用户的兴趣发展。

由此提出融合注意力和记忆网络的序列推荐算法,使用 Word2vec 学习并初始化用户和项目的固定表示,自注意力机制结合 LSTM 解决长距离依赖和区分度问题,学习用户的初级动态表示,并利用记忆网络获取动态邻居学习用户的终级动态表示,拼接固定表示和终级动态表示作为用户最终表示并进行推荐,提高用户的推荐效果并体现出可解释性。

1 相关工作

1.1 Word2vec

MIKOLOV T 等人^[10]提出了两种用于训练词向量表示的模型,分别是 Continuous Bag-of-Words 和 Continuous Skip-gram。前者是基于上下文来预测当前单词,后者是预测当前单词的一定范围内上下文单词。后续此作者^[11]使用分层 softmax、负采样和频繁词的子抽样对 Skip-gram 模型进行扩展,提高了模

型的训练速度和词表示的准确度,广泛应用于问答系统和文本摘要等领域。

1.2 注意力机制

注意力机制的提出借鉴于人类的特性,人们在观察或思考时会注意特定的东西而忽略其他无关的。广泛应用在图像处理、自然语言处理,而后也有应用于推荐。VASWANI A 等人^[12]提出完全基于注意力机制的 Transformer 模型,在机器翻译上取得了良好的效果。神经网络拥有黑盒特性,结合注意力机制能够有效地提高其可解释性。

1.3 记忆网络

记忆网络^[13]是 Facebook 成员 WESTON J 等人提出的一种新模型,主要针对 RNN 记忆太小不能完成需要长期记忆的任务,能够对大量的记忆进行读取和更新,缺点是需要监督。而后 SUKHBAATAR S 等人^[14]在这基础上进行了改进,利用软关注机制使其能够端到端地训练,减少了训练时的监督,是一种新颖的具有注意力机制的循环神经网络。

2 融合注意力和记忆网络的序列推荐算法

2.1 用户和项目固定表示

用户的兴趣变化由用户的历史项目体现出来。假设 $U = \{u_1, u_2, u_3, \dots, u_M\}$ 是用户集合, $H = \{v_1, v_2, v_3, \dots, v_N\}$ 是项目集合。对于每个用户 u , 其浏览的项目序列 $S_u = \{s_u^1, s_u^2, \dots, s_u^{t-1}, s_u^t\}$, 其中 s_u^t 是用户 u 在时间 t 时浏览的项目。序列预测问题是为每个用户 u 预测项目 s_u^{t+1} 。使用 skip-gram 模型学习所有项目的向量,用来初始化项目的固定表示嵌入矩阵 $B = [b_1, b_2, b_3, \dots, b_N]$ 。对于每个项目 v , 浏览过项目 v 的用户序列 $S_v = \{s_v^1, s_v^2, \dots, s_v^{t-1}, s_v^t\}$, 其中 s_v^t 是在时间 t 时浏览项目 v 的用户。使用 skip-gram 模型学习所有用户的向量,用来初始化用户的固定表示嵌入矩阵 $A = [a_1, a_2, a_3, \dots, a_M]$ 。部分项目由于没有用户浏览无法学习到向量表示,导致项目冷启动问题。根据项目的标签计算相似度,随机选取相似度高的项目放在冷启动项目前后,以此学习冷启动项目的向量表示。冷启动项目 i 和项目 j 的相似度计算公式如式 (1) 所示。

$$sl_{i,j} = e_i^T e_j \sqrt{h_j} \quad (1)$$

式中 e_i 和 e_j 是项目标签向量, h_j 是项目 j 的浏览用户数目。使用以上方法学习到的向量作为用户嵌入矩阵和项目嵌入矩阵的初始化。

2.2 用户的动态表示

首先使用自注意力机制和长短期记忆网络学习用户的初级动态表示。然后使用记忆网络存储所有用户的固定表示,某一用户的动态表示作为查询,自适应注意邻居,学习用户的终级动态表示。

对于每个用户 u , 新的序列表示是浏览的项目向量集 $\mathbf{X}_u = [\mathbf{x}_u^1, \mathbf{x}_u^2, \dots, \mathbf{x}_u^t, \mathbf{x}_u^{t-1}]$, 其中 $\mathbf{x}_u^t$ 的初始化是用户 u 在时间 t 时浏览的项目经 skip-gram 学习的向量, 即由项目固定表示嵌入矩阵 $\mathbf{B}$ 得到。

项目表示向量 $\mathbf{x}_u^t$ 加入位置嵌入, 公式如下所示:

$$\mathbf{X}'_u = \mathbf{X}_u + \mathbf{PE}(\mathbf{X}_u) \quad (2)$$

式中位置嵌入矩阵 $\mathbf{PE}$ 的第 po 行第 i 列表示为:

$$\mathbf{PE}_{(po, 2i)} = \sin(po / 10\,000^{2i/em}) \quad (3)$$

$$\mathbf{PE}_{(po, 2i+1)} = \cos(po / 10\,000^{2i/em}) \quad (4)$$

式中, em 为位置嵌入矩阵 $\mathbf{PE}$ 的维度。

自注意力机制解决序列之间的长距离依赖性差和区分度差问题, 自注意力时 $\mathbf{Q} = \mathbf{K} = \mathbf{V}$ 且项目 $\mathbf{x}_u^t$ 只能注意 t 时刻之前的项目。使用残差网络得到新的序列输入表示, 公式如下所示:

$$\tilde{\mathbf{X}}_u = \mathbf{X}'_u + \text{Attention}(\mathbf{X}'_u, \mathbf{X}'_u, \mathbf{X}'_u) \quad (5)$$

其中注意力计算公式如下所示:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{(\mathbf{Q}\mathbf{W}^Q + \mathbf{b}^Q)(\mathbf{K}\mathbf{W}^K + \mathbf{b}^K)}{\sqrt{em}}\right)(\mathbf{V}\mathbf{W}^V + \mathbf{b}^V) \quad (6)$$

式中 $\mathbf{W}^Q, \mathbf{b}^Q, \mathbf{W}^K, \mathbf{b}^K, \mathbf{W}^V, \mathbf{b}^V$ 为全连接参数矩阵。

将新的序列输入表示输入到 LSTM 模型中, 得到每个用户的初级动态表示, 公式如下:

$$\mathbf{O}_u = \text{LSTM}(\tilde{\mathbf{X}}_u) \quad (7)$$

使用单跳记忆网络, 将所有用户的固定表示存储在记忆模块中, 每个用户的初级动态表示作为查询, 获取用户的终级动态表示, 将原记忆网络中的点积注意力改为 Transformer 模型中的注意力, 值得注意的是用户的两个嵌入矩阵依旧按照记忆网络中设置为不同的, 即 $\mathbf{K} \neq \mathbf{V}$ 。 $\mathbf{Q}$ 为用户的初级动态表示。公式如下所示:

$$\tilde{\mathbf{O}}_u = \mathbf{O}_u + \text{Attention}(\mathbf{O}_u, \mathbf{A}, \mathbf{C}) \quad (8)$$

其中 $\mathbf{A}, \mathbf{C}$ 为所有用户的固定表示嵌入矩阵。

2.3 预测项目

拼接用户固定表示和用户终级动态表示作为

用户最终表示:

$$\mathbf{Y}_u = \text{Concat}[\tilde{\mathbf{O}}_u, \mathbf{A}_u] \quad (9)$$

对用户推荐项目时, 计算用户 u 与项目 i 的得分:

$$z_{u,v} = \mathbf{Y}_u^\top \mathbf{D}_v + \mathbf{b} \quad (10)$$

其中 $\mathbf{D}$ 为项目固定表示嵌入矩阵, 推荐得分高的项目。

3 实验

3.1 数据集

使用 100 K 的 MovieLens 数据集, 用户数 943, 项目数 1 682, 评论数 100 000。

本次实验将数据集中每个用户的历史项目按照时间从前往后排序, 每个用户的最后一个项目组成测试集, 倒数第二个项目组成验证集, 剩下的项目按照一定长度切割成多个样本组成训练集。

3.2 模型相关环境和参数设置

使用 TensorFlow 深度学习框架, python 3.5 编程语言。训练迭代次数为 30; 每个用户输入序列长度为 5; 批量大小为 256; 学习率为 0.002, 衰减速度为 400, 衰减系数为 0.98, 并限制最小值为 0.000 05; 嵌入矩阵 $\mathbf{A}, \mathbf{B}, \mathbf{C}$ 和 LSTM 单元输出维度均为 100, 嵌入矩阵 $\mathbf{D}$ 维度为 200; LSTM 单元激活函数为 tanh; 梯度裁剪的二范式约束值为 8; dropout 层丢失率均为 0.5; 使用交叉熵损失函数, Adam 优化器进行训练, 学习嵌入矩阵 $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{b}$, 所有嵌入矩阵的初始化均使用 skip-gram 学习的向量。

3.3 评价指标

实验采用主流的序列推荐评价指标命中率 (Hit ratio, HR), 平均倒数排名 (mean reciprocal rank, MRR) 和归一化折损累计增益 (normalized discounted cumulative gain, NDCG)。HR 是一种常用的衡量召回率的指标, 其定义如下式所示:

$$\text{HR@K} = \frac{\text{Number of Hits@K}}{|\mathbf{GT}|} \quad (11)$$

其中分子是每个用户前 K 个推荐项目中属于测试集合的个数的总和, 分母是所有用户的测试集合项目总数。

MRR 是用于对搜索算法进行评价的指标, 现也用于评价推荐系统的效果, 公式如下:

$$\text{MRR} = \frac{1}{|\mathbf{U}|} \sum_{u=1}^{|\mathbf{U}|} \frac{1}{ra_u} \quad (12)$$

其中, ra_u 是用户 u 推荐列表中第一个属于测试集的

项目的排序位置, $|U|$ 是所有待评价的用户个数。

NDCG 是常用的信息检索指标, 基本思想是: 返回列表中, 与查询文档相关性越强的文档排名越靠前。近来已经被广泛应用于推荐系统的评价指标。其定义如下:

$$DCG_u = \sum_{k=1}^K \frac{2^{re_k} - 1}{\log_2(k + 1)} \quad (13)$$

DCG 是折损累计增益, k 是推荐项目的排序位置, re_k 是排序为 k 的推荐项目的相关性等级。在本文中, 如果用户 u 的第 k 个推荐项目属于测试集合, 则 $re_k = 1$, 否则 $re_k = 0$ 。

$$NDCG_u = \frac{DCG_u}{IDCG_u} \quad (14)$$

其中 $IDCG_u$ 是最理想的推荐列表, 即所有项目都按照用户的喜爱程度排序的 DCG_u 值, 本文中使用用户 u 的测试集项目排序计算此值。NDCG 是所有待评估用户 $NDCG_u$ 值的平均值。

3.4 对比算法

为了验证本文提出的融合注意力和记忆网络的序列推荐算法 (AM-LSTM), 将选用 3 种算法来进行对比实验。

FPMC^[4]: 结合矩阵分解和马尔科夫链, 为每个用户建立基于马尔科夫链的项目-项目转移概率矩阵, 学习用户-项目偏好和建模用户顺序行为, 通过线性方式将它们组合起来, 用于下一个篮子推荐。

Caser^[15]: 将用户序列嵌入到时间和潜在空间的“图像”中, 并使用水平和垂直卷积滤波器学习点级、联合级和跳跃行为的序列模式, 捕获用户一般偏好和顺序模式。

GRU4Rec^[16]: 利用 RNN 对基于会话的序列进行建模, 并使用基于排名的损失函数训练模型。

3.5 实验结果分析

嵌入矩阵维度大小对 HR@50 的影响实验结果如图 1 所示, 对 MRR@50 的影响实验结果如图 2 所示。随着维度的增大, 本文算法 AM-LSTM 的 HR 逐渐升高; FPMC 和 Caser 的 HR 均随着维度增大而升高, 在维度为 100 时达到顶峰, 而后显著降低。AM-LSTM 和 Caser 的 MRR 均在维度为 75 时达到峰值, 而 GRU4Rec 跌到最低。模型的嵌入矩阵维度越大, 更能表达用户和项目之间的关系, 模型更加拟合, 推荐效果也更好。但当矩阵维度增大到一定程度时, 对训练集中用户和项目的关系表达太过细致,

使得模型过拟合, 且训练时间更长, 推荐效果变差。在此后的实验中, 将嵌入矩阵维度设为 100。

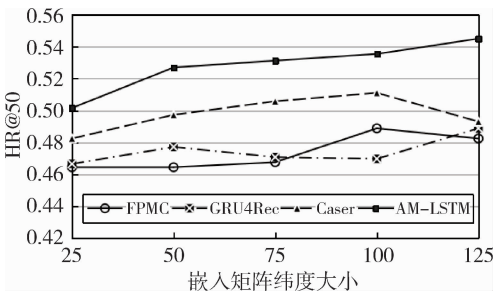


图 1 嵌入矩阵维度大小对 HR@50 的影响

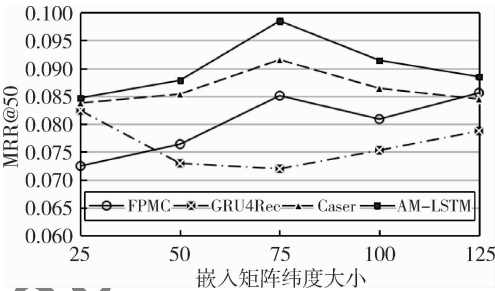


图 2 嵌入矩阵维度大小对 MRR@50 的影响

表 1 显示了 4 种算法在嵌入矩阵维度为 100, 推荐项目个数为 50 时的 HR、MRR、NDCG, 4 种算法的 3 种评价指标高低趋势是一致的, 从指标低到高的算法依次是 GRU4Rec、FPMC、Caser、AM-LSTM。指标越高说明效果越好。本文提出的算法 AM-LSTM 相比较 GRU4Rec, 在 HR 上提高了 13.98%, 在 MRR 上提高了 21.38%, 在 NDCG 上提高了 17.02%。实验结果表明, 加入注意力机制和记忆网络能显著提高推荐的效果。

表 1 4 种算法的 HR@50、MRR@50 和 NDCG@50 比较

算法	HR@50	MRR@50	NDCG@50
GRU4Rec	0.469 8	0.075 3	0.152 2
FPMC	0.488 9	0.080 9	0.159 9
Caser	0.511 1	0.086 4	0.169 1
AM-LSTM	0.535 5	0.091 4	0.178 1

4 结论

本文针对传统推荐系统无法很好地表示用户动态兴趣的演变, 基于循环神经网络的推荐算法存在长距离依赖性差、区分度差、可解释性差的问题, 提出融合注意力和记忆网络的序列推荐算法, 本文

先介绍了传统推荐算法和序列推荐算法,重点介绍基于循环神经网络序列推荐算法的现状,然后介绍了 Word2vec、注意力机制和记忆网络。提出融合注意力和记忆网络的序列推荐算法,使用 Word2vec 能更好地表示用户和项目潜在特征,加快收敛速度;使用自注意力机制解决用户序列的长距离依赖性差和区分度差问题;使用记忆网络获得用户的动态邻居,更准确地表示用户的动态兴趣。最后实验结果表明此算法提高了推荐的准确度,同时也增强了可解释性。

但是,本文的算法还有不足的地方,在使用 item2vec 学习项目向量表示时,少数项目不在训练集中,即冷启动项目,直接从 MovieLens 数据集中获取项目的标签,添加相同标签的项目来学习这些项目。对于没有标签的项目,还需进一步处理才能适用,可以从项目名称和内容描述中提取特征。现实生活中存在大量冷项目,以及 RNN 的输入序列是在时间轴上位置相邻,没有考虑两项目之间时间间隔的影响。今后将在这些方面展开研究。

参考文献

- [1] BREESE J S, HECKERMAN D, KADIE C. Empirical analysis of predictive algorithms for collaborative filtering[C]. the 14th Conference on Uncertainty in Artificial Intelligence, 1998; 43-52.
- [2] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [3] EBESU T, SHEN B, FANG Y. Collaborative memory network for recommendation systems[C]. SIGIR'18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018.
- [4] RENDLE S, FREUDENTHALER C, THIEME L S. Factorizing personalized markov chains for next-basket recommendation[C]. Proceedings of the 19th International Conference on World Wide Web, 2010; 811-820.
- [5] FU H, LI J, CHEN J, et al. Sequence-based recommendation with bidirectional LSTM network[M]. Lecture Notes in Computer Science, 2018, 11166: 428-438.
- [6] LUO A, ZHAO P, LIU Y, et al. Adaptive attention-aware gated recurrent unit for sequential recommendation[M]. Lecture Notes in Computer Science, 2019, 11447: 317-332.
- [7] LV F, JIN T, YU C, et al. SDM: sequential deep matching model for online large-scale recommender system[C]. The 28th ACM International Conference on Information and Knowledge Management (CIKM'19), 2019.
- [8] LIN X, NIU S, WANG Y, et al. K-plet recurrent neural networks for sequential recommendation[C]. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018; 1057-1060.
- [9] BARKAN O, KOENIGSTEIN N. Item2vec: neural item embedding for collaborative filtering[C]. 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), 2016.
- [10] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[C]. Proceedings of the International Conference on Learning Representations, 2013.
- [11] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C]. Advances in Neural Information Processing Systems, 2013.
- [12] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]. Advances in Neural Information Processing Systems, 2017.
- [13] WESTON J, CHOPRA S, BORDES A. Memory networks[C]. Proceedings of the International Conference on Learning Representations, 2015.
- [14] SUKHBATAR S, SZLAM A, WESTON J, et al. End-to-end memory networks[C]. Advances In Neural Information Processing systems, 2015.
- [15] TANG J, WANG K. Personalized top-n sequential recommendation via convolutional sequence embedding[C]. Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018; 565-573.
- [16] HIDASI B, KARATZOGLOU A, BALTRUNAS L, et al. Session-based recommendations with recurrent neural networks[C]. Proceedings of ICLR, 2016.

(收稿日期:2020-03-19)

作者简介:

陈细玉(1993-),女,硕士研究生,主要研究方向:推荐系统。

林穗(1972-),通信作者,女,副教授,主要研究方向:云计算、云存储、操作系统、数据挖掘。E-mail: Linsui@gdut.edu.cn。

版权声明

经作者授权，本论文版权和信息网络传播权归属于《信息技术与网络安全》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《信息技术与网络安全》编辑部
中国电子信息产业集团有限公司第六研究所