



刻画长短期用户兴趣的基于会话的推荐系统

王鸿伟, 过敏意*

上海交通大学新兴并行计算研究中心, 上海 200240

* 通信作者. E-mail: guo-my@cs.sjtu.edu.cn

收稿日期: 2019-08-22; 修回日期: 2019-10-02; 接受日期: 2020-01-07; 网络出版日期: 2020-11-20

国家自然科学基金 (批准号: 61832006, 61872240) 资助项目

摘要 在很多在线推荐系统场景中, 用户和系统的交互通常都会按照时间间隔划分成一系列的会话. 本文研究了如何在基于会话的推荐系统中对用户兴趣进行建模. 现有的工作或者假设会话之间相互独立, 忽略了历史会话中包含的长期用户兴趣信息; 或者将用户在一个会话中的短期兴趣视为静态不变. 这两者都无法充分刻画实际场景中的用户兴趣和行为. 我们提出了循环记忆网络 RMN, 一种刻画了用户长期和短期兴趣的基于会话的推荐系统. RMN 基于循环神经网络模型, 其核心部分是一个存储了用户长期偏好的兴趣记忆模块. 另外, 我们在 RMN 中设计了一个会话内的记忆读取单元和一个会话间的记忆写入单元, 这两个单元分别用于刻画短期 (即一个会话内) 的用户兴趣波动和长期 (即跨越多个会话) 的用户兴趣转移. 我们在真实的电影推荐和职位推荐数据集上进行了实验, 实验结果表明 RMN 相比于现有方法而言取得了显著的性能提升.

关键词 推荐系统, 会话, 长短期用户兴趣, 循环神经网络, 记忆网络

1 引言

在互联网高速发展的今天, 大量的在线应用和服务提供商都使用推荐系统技术去发掘用户兴趣, 提升用户对产品和服务的参与程度, 例如电子商务^[1]、音乐^[2]、新闻^[3]、职位^[4]、在线投票^[5,6]等. 在这些互联网应用的推荐系统中, 用户和系统的交互通常会被按照时间间隔划分为一系列的会话^[7]. 一个会话 (session) 是指在一个特定时间间隔 (例如几小时、几天、几周等) 内, 用户和系统之间发生的一组交互行为. 在一个会话内, 用户的意图通常会表现出高度的相关性和一致性. 例如, 一个用户在一次购物中会同时购买尿布和奶粉; 一个用户在一段时间内会喜欢听相同风格的音乐. 根据会话的信息来刻画用户在局部时间内的行为规律对于提升推荐系统的性能大有裨益, 因此, 基于会话的推荐最近也成为了研究热点^[7~14].

引用格式: 王鸿伟, 过敏意. 刻画长短期用户兴趣的基于会话的推荐系统. 中国科学: 信息科学, 2020, 50: 1867–1881, doi: 10.1360/SSI-2019-0177
Wang H W, Guo M Y. Recurrent memory networks: modeling long short-term user preferences for session-based recommendation (in Chinese). Sci Sin Inform, 2020, 50: 1867–1881, doi: 10.1360/SSI-2019-0177

一般而言, 基于会话的推荐面临如下两个挑战^[15]. 一方面, 用户的兴趣会随着会话的变化而发生显著或缓慢的转变^[16]. 例如, 一个用户会在夏天购买短袖衬衫, 而在冬天购买围巾; 一个用户喜欢的音乐风格也可能在几年间从摇滚转移到嘻哈. 因此, 准确地对用户的长期 (也叫做会话间, inter-session) 兴趣转移进行建模是基于会话的推荐系统的一个重要目标. 另一方面, 尽管一个用户在一个会话内的行为表现出高度的一致性^[13], 这些行为意图依然会在种类、品牌、风格上表现出不同. 例如, 用户在购买完手机之后会更倾向于浏览手机保护壳或手机挂坠等商品, 此时再度给该用户推荐手机会显得冗余而低效. 用户在一个会话内的兴趣也会发生变化, 我们称之为短期 (也叫做会话内, intra-session) 用户兴趣.

尽管近些年有很多基于会话的推荐系统被提出, 但是他们都没有显式地解决以上两点问题. 大部分已有工作都假设会话之间相互独立且用户是匿名的, 因此, 他们只对用户在一个特定会话内的短期行为进行建模, 而忽略了来自历史会话的信息^[7, 9, 10, 12, 14]. 然而, 在大部分互联网应用中, 用户通常都是已登录状态, 或者系统拥有其他可用的用户识别信息例如浏览器缓存 (cookie) 等^[13]. 因此, 我们可以合理地假设系统能够追踪到用户在过去会话中的行为. 用户的历史行为会给推荐系统提供宝贵的信息, 然而这些信息在这些已有工作中无法被利用. 值得指出的是, 有两个现有工作考虑了对用户在会话间的兴趣演变进行建模: II-RNN^[17] 使用了一个会话内循环神经网络 (recurrent neural networks, RNN) 对用户单个会话内的行为进行建模; 同时设计了另一个会话间 RNN, 用以读取用户的历史会话, 并为会话内的 RNN 提供初始输入. 但是在 II-RNN 中, 用户的历史信息只被用于为会话内 RNN 的初始状态 (而不是会话内 RNN 的每个状态) 提供输入, 这使得用户的历史信息只能为当前会话提供有限的监督信号. 类似地, HRNN^[13] 也增加了一个额外的用户层级的 RNN 来建模用户的长期兴趣转移. 但是 HRNN 将用户在一个会话内的兴趣视为是静态不变的, 这种做法缺乏足够的刻画用户短期内变化的兴趣.

为了解决已有工作的不足之处, 本文提出了循环记忆网络 (recurrent memory networks, RMN), 是一种在基于会话的推荐系统中对用户长期和短期兴趣进行联合建模的方法. 与现有方法的不同之处在于, RMN 显式地刻画了用户的长期和短期兴趣. RMN 主要有如下 3 个模块: (1) 通常而言, RNN 只能存储短期记忆而很难存储长期记忆^[18]. 受到最近在提升神经网络记忆机制方面的工作^[19~21] 的启发, 我们设计了一个外部的兴趣记忆模块 (preference memory) 来储存每个用户的长期兴趣, 用以增强 RNN 的记忆能力. (2) 我们在两个相邻的会话中间设计了一个会话间记忆写入单元 (inter-session memory writer) 用以对用户的长期兴趣转移进行建模. 该记忆写入单元会学习用户长期兴趣发生的变化, 并在每个会话结束后更新用户的记忆模块的内容. (3) 为了获得用户短期兴趣的动态表征, 我们设计了一个会话内记忆读取单元 (intra-session memory reader), 该单元会将用户当前状态与他的记忆模块中的每条记录进行匹配, 并且从记忆模块中按照不同的权重抽取相应的记忆内容. 抽取出的短期兴趣会被当作 RNN 的一个额外输入, 用以辅助预测用户在下一个时刻的行为.

我们将 RMN 应用到了两个真实的数据集上: 关于电影推荐的 MovieLens-20M 数据集和关于职位推荐的 XING 数据集. 实验结果表明, 相比于最好的对比方法, RMN 在 MRR@10 指标上分别实现了 6.8% 和 20.6% 的性能提升. 实验结果也验证了在 RMN 中对用户长期和短期兴趣进行建模的有效性和必要性.

2 问题描述

本文将基于会话的推荐问题定义如下: 我们用 $\mathcal{U} = \{u_1, u_2, \dots\}$ 和 $\mathcal{V} = \{v_1, v_2, \dots\}$ 分别表示用户

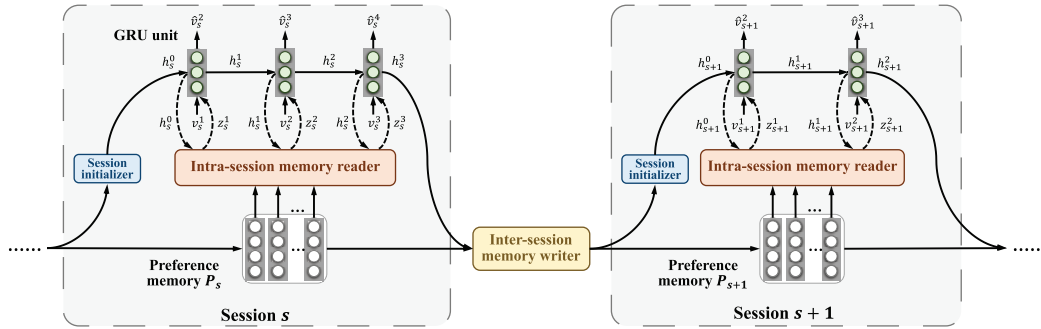


图 1 (网络版彩图) 循环记忆网络 RMN 在两个连续会话中的模型结构
 Figure 1 (Color online) The overall framework of RMN for two consecutive sessions

集合和商品集合. 我们根据用户对商品的隐式反馈定义一个用户和商品之间的交互矩阵 $Y = \{y_{uv} | u \in \mathcal{U}, v \in \mathcal{V}\}$, 即如果用户 u 点击过商品 v , 那么 $y_{uv} = 1$, 否则 $y_{uv} = 0$. 下文会在没有引起歧义的情况下省去记号中的用户标记 u , 因为我们的模型描述只针对某个给定的用户. 用户和商品之间的每条交互都关联了一个会话编号 s . 我们使用 \mathcal{V}_s 来表示一个给定用户在第 s 个会话内交互过的所有商品的集合. 给定一个用户的历史会话信息 $\mathcal{V}_1, \dots, \mathcal{V}_{s-1}$ 以及在当前会话内的所有交互信息 $\mathcal{V}_s = \{v_s^1, \dots, v_s^{m-1}\}$, 我们的目标是预测该用户下一个可能会交互的商品 \hat{v}_s^m .

3 循环记忆网络

本节详细介绍提出的 RMN 模型. 我们首先介绍模型的整体结构, 然后分别阐述其中的每个模块. 我们最后会讨论 RMN 和现有模型相比的优点.

3.1 模型结构

图 1 展示了 RMN 模型在两个连续会话中的整体结构. 在第 s 个会话内, 一个用户在 RMN 中具有以下信息: (1) 一个兴趣记忆模块 (preference memory) P_s , 其记录了该用户的长期兴趣; (2) 一系列的隐状态 (hidden state) $h_s^i (i = 0, 1, \dots, N_s, N_s$ 是该用户在第 s 个会话内交互行为的数量).

在该会话的第 i 个阶段, 会话内记忆读取单元 (intra-session memory reader) 使用隐状态 h_s^i 和兴趣记忆模块 P_s 作为输入, 然后输出一个表示该用户在当前阶段的短期兴趣的向量 z_s^i . 随后, 该用户的隐状态 h_s^i 、短期兴趣 z_s^i 和当前阶段点击的商品 v_s^i 被输入到一个门控循环神经网络 (gated recurrent unit, GRU)¹⁾ 中, 用以输出预测该用户在下一个阶段点击的商品 \hat{v}_s^{i+1} 以及下一个阶段的隐状态 h_s^{i+1} . 在当前会话结束后, 会话间记忆写入单元 (inter-session memory writer) 会根据该用户在该会话内的最后一个隐状态 $h_s^{N_s}$ 的值, 对兴趣记忆模块 P_s 进行写入和更新. 新的记忆 P_{s+1} 理论上保存了该用户所有的历史信息, 并作为下一个会话的初始输入.

3.2 模型组成

RMN 由以下 5 个模块组成.

1) 这里也可以使用长短期记忆神经网络 (long-short term memory networks, LSTM) 作为 RNN 的实现. 我们选择 GRU 的原因是, 这两者在性能上没有显著区别, 但是 GRU 比 LSTM 运行时间更短, 效率更高.

3.2.1 用户兴趣记忆模块

在 RMN 中, 我们为每个用户在第 s 个会话内定义一个兴趣记忆模块 P_s , 用以刻画该用户对商品的偏好. 兴趣记忆模块 P_s 为一个用户存储了 M 个隐偏好向量, 即

$$P_s = \{p_s^1, p_s^2, \dots, p_s^M\}, \quad (1)$$

其中 $p_s^i \in \mathbb{R}^D$ 是该用户在第 s 个会话中的第 i 个偏好向量. 一般而言, P_s 可以按照以下 3 种方式实现.

- 聚类层级 (cluster-level) 的记忆包含了用户点击历史的 M 个聚类的中心, 其综合衡量了该用户在过往会话中的信息;
- 商品层 (item-level) 的记忆包含了该用户最近点击过的 M 个商品的表征向量, 其更关注于对用户近期的行为进行建模^[18];
- 特征层级 (feature-level) 的记忆包含了 M 个隐兴趣向量, 用来刻画用户对于商品特征的兴趣^[22].

本文使用特征层级的记忆作为 P_s 的实现, 因为实验结果表明这种实现可以赋予 RMN 更多的灵活性和表达能力. 但是值得注意的是另外两种记忆的实现也可以很容易地部署到 RMN 中. 与 HRNN^[13] 将用户的兴趣定义为单个向量的方式不同, RMN 将用户的兴趣记忆定义为一个矩阵, 这样使得 RMN 在会话中的每个阶段都具有了更加细粒度的用户兴趣建模的能力. 超参数 M 和 D 分别控制了用户兴趣的数量和维度, 它们会在后续实验中通过交叉验证来确定.

在 RMN 中引入用户兴趣记忆有如下两个目的: (1) 在一个会话的每个阶段, 兴趣记忆可以帮助提取用户的短期兴趣 (通过记忆读取单元), 该短期兴趣向量会和用户的当前隐状态结合起来, 用以更好地刻画用户的即时行为; (2) 在一个会话结束后, 兴趣记忆会吸收和包含用户在该会话中的信息, 并刻画用户兴趣在跨会话时的迁移 (通过记忆写入单元).

3.2.2 会话初始化单元

在一个新的会话开始时, 会话初始化单元基于用户的兴趣记忆, 为该会话提供了初始输入信息. 一个拥有足够信息的 GRU 单元的初始状态可以更好地对用户行为进行建模. 在 RMN 中, GRU 的初始隐状态由一个多层感知机 (multi-layer perceptron, MLP) 初始化, 该 MLP 使用用户的兴趣记忆向量作为输入:

$$h_s^0 = f_{\text{init}} \left(\frac{1}{M} \sum_{i=1}^M p_s^i \right). \quad (2)$$

通过这种形式, 用户在历史会话中的兴趣信息被转移进入了当前的会话. 我们会在实验部分验证会话初始化单元的效果.

3.2.3 会话内记忆读取单元

给定一个用户在第 s 个会话内的兴趣记忆 $P_s = \{p_s^1, p_s^2, \dots, p_s^M\}$, 我们的目标是在该会话内的每一个阶段刻画用户的短期兴趣. 我们将该操作抽象成为读取 (READ) 操作:

$$z_s^j \leftarrow \text{READ} (\{p_s^1, p_s^2, \dots, p_s^M\}, h_s^j), \quad (3)$$

其中 z_s^j 是该用户在第 s 个会话中的第 j 个阶段的短期兴趣. 一个直观的做法是简单地将该用户所有的兴趣向量求平均值:

$$z_s^j \leftarrow \frac{1}{M} \sum_{i=1}^M p_s^i, \quad j = 1, 2, \dots, N_s. \quad (4)$$

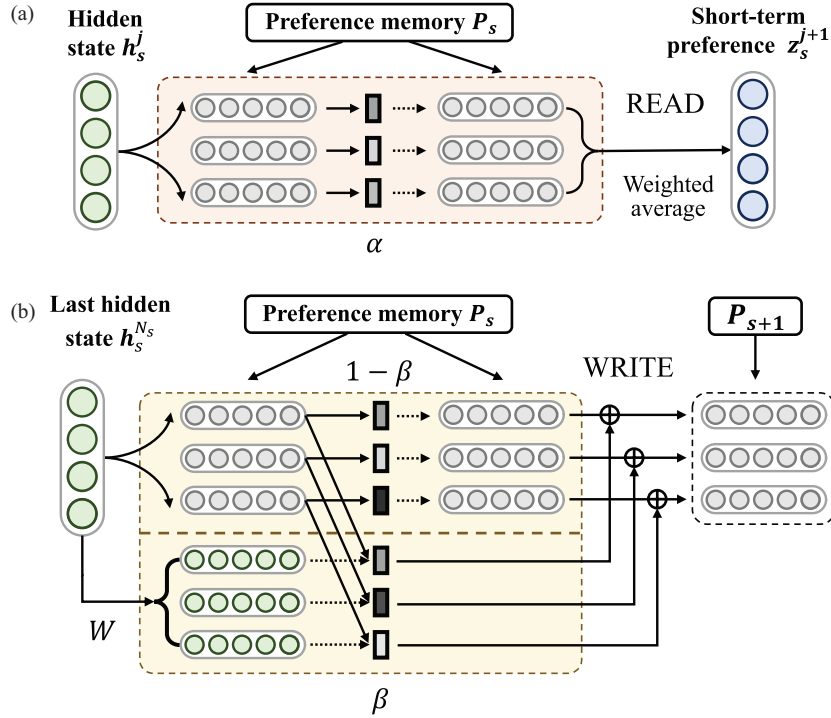


图 2 (网络版彩图) RMN 中的 (a) 会话内记忆读取单元和 (b) 会话间记忆写入单元

Figure 2 (Color online) Illustration of (a) intra-session memory reader and (b) inter-session memory writer in RMN

注意到这种做法将用户的短期兴趣在整个会话内视为是静态不变的. 然而, 正如我们在引言中提到的, 用户的兴趣在一个会话的时间内也可能会发生变化. 在考虑用户的短期行为时, 该用户的不同的兴趣记忆向量也应该对其当前的短期兴趣有着不同的影响. 为了更细粒度地刻画用户的短期兴趣, 我们在 RMN 中设计了一个注意力记忆读取单元. 该单元的结构如图 2(a) 所示. 具体地说, 对于用户的兴趣记忆向量 p_s^i 及其当前的隐状态 h_s^j , 我们首先将这两者连接起来, 然后使用 $f_{att}(\cdot)$ 作为注意力函数, 以及归一化指数函数 (softmax) 来计算归一化的影响力值:

$$\alpha_s^{i,j} = \text{softmax}(p_s^i, h_s^j) = \frac{\exp(f_{att}(p_s^i, h_s^j))}{\sum_{i=1}^K \exp(f_{att}(p_s^i, h_s^j))}. \quad (5)$$

注意力函数 $f_{att}(\cdot)$ (例如内积或 MLP) 读取一个用户兴趣记忆向量 p_s^i 和一个 GRU 的隐状态 h_s^j 作为输入, 并输出它们之间的影响力值. 一个大的 α 值表明该条兴趣在当前阶段对用户会产生主要的影响. 最后, 用户的短期兴趣可以表示为他的兴趣记忆向量按照 α 值的加权平均:

$$z_s^j \leftarrow \sum_{i=1}^M \alpha_s^{i,j} p_s^i. \quad (6)$$

我们会在实验部分验证会话内记忆读取单元的效果.

3.2.4 GRU 单元

在 RMN 中, 我们选择 GRU 作为 RNN 的实现, 对用户在一个会话的点击序列进行建模. 由于其独特的门单元设计, GRU [23] 是一种可以从拥有长距离依赖关系的时间序列中高效抽取信息的 RNN

结构. 具体地说, 我们定义在输入序列的第 j 个位置 ($j = 1, \dots, N_s$) 的 GRU 单元为: 商品输入 v_s^j , 短期用户兴趣输入 z_s^j , 中间隐状态 \tilde{h}_s^j , 隐状态 h_s^j , 更新门 p_s^j , 重置门 q_s^j , 输出向量 o_s^j . 门向量 p_s^j 和 q_s^j 的取值区间为 $[0, 1]$. GRU 单元的转移方程如下所示:

$$p_s^j = \sigma(W_p[h_s^{j-1}, v_s^j, z_s^j]), \quad (7)$$

$$q_s^j = \sigma(W_q[h_s^{j-1}, v_s^j, z_s^j]), \quad (8)$$

$$\tilde{h}_s^j = \tanh(W_h[q_s^j \circ h_s^{j-1}, v_s^j, z_s^j]), \quad (9)$$

$$h_s^j = (1 - p_s^j) \circ h_s^{j-1} + p_s^j \circ \tilde{h}_s^j, \quad (10)$$

$$o_s^j = W_1 h_s^{j-1} + W_2 v_s^j + W_3 z_s^j, \quad (11)$$

其中 W_p, W_q, W_h 是 GRU 单元的转换矩阵, W_1, W_2 和 W_3 是用来计算输出值的矩阵, $\sigma(\cdot)$ 是 sigmoid 函数, \circ 表示按位相乘操作.

RMN 中的 GRU 单元和传统的 GRU 单元有 3 个不同: (1) GRU 中的初始状态 h_s^0 是由会话初始单元赋值的, 它包含了用户在上一个会话的兴趣信息, 因此可以减轻传统 GRU 中的冷启动问题; (2) 每个 GRU 单元都有一个额外的短期用户兴趣向量作为额外输入, 因此用户的即时兴趣为 GRU 提供了一个额外的输入监督信号; (3) 和传统 GRU 将隐状态直接作为输出不同, 我们额外使用了 v_s^j 和 z_s^j 来构建输出向量. 根据我们的实验结果, 这样可以有效地提高模型的性能.

在得到第 j 个阶段的输出值 o_s^j 之后, 我们使用一个 softmax 函数层来计算所有候选商品的预测概率值:

$$\left[\hat{r}_{s,j}^1, \dots, \hat{r}_{s,j}^{|\mathcal{V}|} \right] = \text{softmax}(W o_s^j), \quad (12)$$

其中 $\hat{r}_{s,j}^i$ 是商品 i 的预测的被点击概率.

3.2.5 会话间记忆写入单元

如我们在引言中所述, 一个用户的兴趣通常会在不同的会话间发生变化. 为了刻画这种兴趣转移现象, 我们在 RMN 中设计了一个会话间记忆写入单元来为下一个会话更新用户的兴趣记忆. 记忆写入单元的操作被抽象为 WRITE:

$$P_{s+1} \leftarrow \text{WRITE}(P_s, h_s^{N_s}), \quad (13)$$

其中 $h_s^{N_s}$ 是第 s 个会话中最后一个隐状态. 受文献 [18] 的启发, 我们计算权值 β 并用其来控制用户兴趣记忆中的每个向量需要被更新的比例. 对于每个向量 p_s^i , 权值 β_s^i 的计算方式为

$$\beta_s^i = \sigma(f_{\text{att}}(p_s^i, h_s^{N_s})), \quad (14)$$

其中 $f_{\text{att}}(\cdot)$ 也是一个注意力函数. 然后, 用户兴趣记忆中的每个向量会根据该权值来进行更新:

$$p_{s+1}^i \leftarrow (1 - \beta_s^i) \cdot p_s^i + \beta_s^i \cdot W^i h_s^{N_s}, \quad i = 1, \dots, M, \quad (15)$$

其中 W^i 是向量 i 的转移矩阵. 记忆写入单元如图 2(b) 所示.

由于兴趣记忆模块 P_s 存储了第 s 个会话之前的所有信息, 且当前会话的最后一个隐状态 $h_s^{N_s}$ 存储了第 s 个会话中的所有信息, 因此, P_{s+1} 可以完全刻画用户所有的历史信息. 我们会在实验部分验证会话间记忆写入单元的效果.

3.3 学习算法

Top- K 推荐通常比较关注候选商品的排列顺序^[24]. 在 RMN 中, 和文献 [7] 类似, 我们选择 TOP-1 排序损失作为我们模型中损失函数的实现. RMN 中对于一个给定用户的完整损失函数如下所示:

$$\mathcal{L} = \sum_{s=1}^S \sum_{j=1}^{N_s-1} \sum_{i=1}^T \left(\sigma(\hat{r}_{s,j}^i - \hat{r}_{s,j}^k) + (\hat{r}_s^i)^2 \right) + \lambda \mathcal{L}_{\text{reg}}. \quad (16)$$

在式 (16) 中, 第 1 项遍历了所有会话中的所有阶段. 对于第 s 个会话中的第 j 个阶段, 我们随机采样了 T 个负样本. 上标 k 表示正样本 (即下一个点击的商品), 上标 i 表示负样本. $\sigma(\hat{r}_{s,j}^i - \hat{r}_{s,j}^k)$ 这一项会鼓励模型给正样本 k 分配更大的预测概率值 (相对于负样本而言). $(\hat{r}_s^i)^2$ 这一项会促使负样本的预测概率值接近 0, 这有利于模型的稳定性. 损失函数的第 2 项是一个 L2 正则项, 用来避免模型出现过拟合现象. 其中 λ 为 \mathcal{L}_{reg} 项的权重.

我们使用了随机梯度下降算法来迭代地优化损失函数. 在每个训练回合, 我们首先根据文献 [13] 中描述的策略, 采样出一个小批量的训练样本. 然后, 我们计算损失函数 \mathcal{L} 关于模型参数的梯度, 并根据采样出来的小批量样本通过反向传播更新所有的参数值.

4 实验

本节分别在电影推荐和职位推荐数据集上验证 RMN 模型的性能.

4.1 数据集

本文使用如下两个数据集.

- MovieLens-20M²⁾ 是一个在电影推荐中广泛使用的基准数据集. 它包含了 MovieLens 网站上大约两千万条的评分记录 (分值从 1 到 5). 我们把大于等于 4 分的记录作为标签为 1 的隐式反馈, 并随机采样了一些用户没有看过的电影作为标签为 0 的隐式反馈. 我们从原始数据集中摘取了 2005 年到 2015 年之间的数据, 并将会话的长度设置为一天. 最终的数据集包含了 118145 个用户, 14847 部电影和 6747655 条交互记录.

- XING³⁾ 数据集发布于 2016 年 Recsys 挑战赛. 它包含了 XING 网站上 80 天时间内 77 万用户和职位招聘帖的交互数据. 每个交互都有一个时间戳信息和一种类型 (点击、收藏、回复、删除). 和文献 [13] 类似, 我们舍弃了“删除”类型的交互, 然后将所有的交互按照 30 分钟的长度划分成不同的会话. 我们也过滤掉了过于稀疏的数据, 包括少于 20 个交互的帖子, 少于 3 个交互的会话和少于 5 个会话的用户. 最终的数据集包含了 11479 个用户, 59297 个帖子, 546862 条交互和 89591 个会话.

4.2 对比方法

我们将 RMN 和如下方法进行比较.

- M-Pop (most popular) 为用户推荐过去最热门的商品.
- S-Pop (session popular) 为用户推荐当前会话中最热门的商品.
- Item-KNN^[12] 基于商品在会话中的共同出现的频次来计算商品之间的相似度, 然后给用户推荐和他最后一次点击的商品最相似的商品.

2) <https://grouplens.org/datasets/movielens/>.

3) <http://2016.recsyschallenge.com/>.

- Session-KNN^[25] 首先确定 k 个和当前会话最相似的历史会话, 然后推荐在这些历史会话中出现频次最高的商品.
- RUM^[18] 是一种使用外部记忆的序列推荐模型 (这意味着它忽略了会话的信息). 本文使用特征层级 (feature-level) 的 RUM.
 - NARM^[14] 使用了一种基于注意力机制的混合编码器来刻画用户在一个会话中的序列行为.
 - RNN^[7] 假设会话之间相互独立, 并使用 RNN 来预测用户的下一个点击商品.
 - II-RNN^[17] 在 RNN 方法的基础上增加了一个 RNN, 用以从最近的会话中获取信息并将其作为当前会话中的 RNN 的初始状态. 本文使用平均池化 (average pooling) 操作.
 - HRNN^[13] 也增加了一个用户层级的 RNN, 用以对用户的长期兴趣进行建模, 但是其假设用户的短期兴趣保持不变.
- GNN^[26] 将用户和物品的交互视为一个二分图, 然后使用图神经网络的方法来学习用户和物品的表征. 得到表征后, 我们使用内积来预测点击概率.

4.3 实验设置

在 RMN 中, 对于 MovieLens-20M 数据集, 我们将用户兴趣向量的数量和维度设置为 $M = 32$ 和 $D = 32$; 对于 XING 数据集, 我们将用户兴趣向量的数量和维度设置为 $M = 64$ 和 $D = 128$. 我们会在后续实验中进一步研究 M 和 D 对实验结果的影响. GRU 中的商品表征维度和隐状态维度分别是 256 和 512. 我们使用 Adam^[27] 作为优化算法的实现, 学习率为 5×10^{-4} , 批量大小为 512. 商品表征和隐状态的 dropout 概率为 0.05. f_{init} 和 f_{att} 的实现都是带有一个隐藏层的 MLP. RMN 和所有对比方法的超参数都是通过在一个验证集上最优化 MRR@2 指标而确定的. 另外, 在预训练中, 我们也使用了和文献 [28] 相似的软标签技术.

对比方法的重要超参数设置如下. 对于 Session-KNN, 我们使用从 500 个最近的候选会话中选择了 200 个最近邻. 对于 RUM, 记忆向量的数量为 32, 维度是 64. 对于 RNN 和 HRNN, 批量大小为 128. 对于 II-RNN, 表征维度为 64. 对于 GNN, 用户和物品的表征维度为 32. 剩余的超参数为默认值或相应文献中汇报的值.

对于每个用户, 我们使用其最后一个会话中的交互作为测试集, 剩余的会话作为训练集. 每个实验被重复 5 次, 然后我们报告 5 次实验的平均值. 我们验证 RMN 和对比方法在 Top- K 推荐中的性能, 我们使用 Precision@ K , Recall@ K , F1@ K 和 Mean Reciprocal Rank@ K (MRR@ K) 作为指标, 其中 $K \in \{2, 5, 10, 20, 50\}$.

4.4 实验结果

RMN 和对比方法的 Precision@ K , Recall@ K , F1@ K 和 MRR@ K 结果如表 1 和图 3 所示. 我们有如下的观察结果.

- S-pop 比 M-pop 表现更好. 我们将其归因于用户 - 商品的交互在一个会话内会多次重复地出现. 但是 S-Pop 并不能为用户进行个性化推荐.
- Session-KNN 比 Item-KNN 表现更好, 这又一次说明考虑会话的信息对于推荐非常重要.
- GNN 的表现在所有对比方法中并不理想, 这是因为 GNN 更适合于节点有原始属性的情形, 而在我们构造的用户 - 物品二分图中, 用户和物品的特征都是随机初始化的, 并没有原始特征作为输入.
- RUM 在所有基于 RNN 的方法中表现最差. 这可能是因为 RUM 是一个通用的序列推荐模型, 它没有考虑会话的信息.

表 1 MovieLens-20M 和 XING 数据集上的 MRR@K 的结果

Table 1 Results of MRR@K on MovieLens-20M and XING dataset

	MovieLens-20M					XING				
	MRR@2	MRR@5	MRR@10	MRR@20	MRR@50	MRR@2	MRR@5	MRR@10	MRR@20	MRR@50
RMN	0.2737	0.2892	0.2940	0.2981	0.3055	0.0922	0.1085	0.1158	0.1188	0.1211
M-Pop	0.0045	0.0073	0.0096	0.0130	0.0142	0.0003	0.0006	0.0010	0.0011	0.0014
S-Pop	0.1077	0.1251	0.1430	0.1612	0.1727	0.0693	0.0923	0.0953	0.0954	0.0960
Item-KNN	0.1855	0.2076	0.2196	0.2231	0.2249	0.0244	0.0380	0.0427	0.0453	0.0469
Session-KNN	0.2014	0.2175	0.2216	0.2360	0.2442	0.0457	0.0525	0.0688	0.0742	0.0863
RUM	0.0754	0.0846	0.1021	0.1068	0.1124	0.0205	0.0368	0.0425	0.0436	0.0477
NARM	0.2512	0.2687	0.2745	0.2895	0.2911	0.0657	0.0884	0.0951	0.0977	0.0995
RNN	0.2241	0.2389	0.2412	0.2426	0.2467	0.0512	0.0770	0.0867	0.0891	0.0904
II-RNN	0.2354	0.2546	0.2753	0.2778	0.2810	0.0815	0.0867	0.0916	0.0974	0.1058
HRNN	0.2310	0.2487	0.2502	0.2593	0.2644	0.0603	0.0883	0.0960	0.0992	0.1009
GNN	0.2134	0.2271	0.2429	0.2533	0.2631	0.0512	0.0721	0.0767	0.0836	0.0882

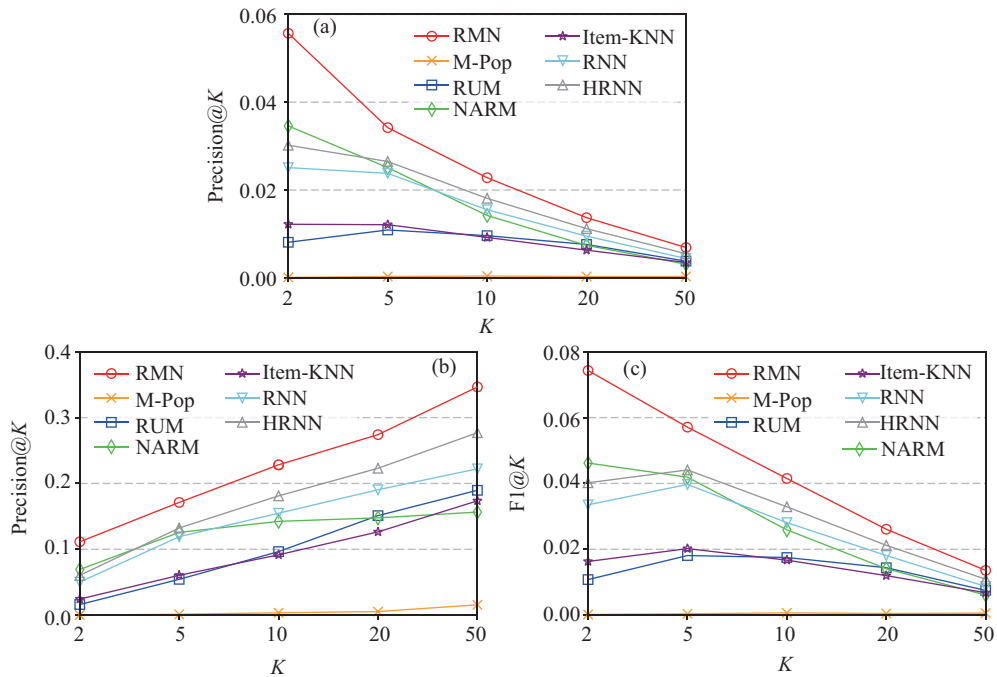


图 3 (网络版彩图) XING 数据集上的 (a) Precision@K, (b) Recall@K 和 (c) F1@K 的结果. 出于清晰的目的, 一些对比方法的结果没有在图中展示

Figure 3 (Color online) Results of (a) Precision@K, (b) Recall@K, and (c) F1@K on XING dataset. Some of the baselines are not shown in this figure for the sake of clarity

• II-RNN 和 HRNN 比 RNN 表现更好. 这表明对用户的长期兴趣进行建模 (通过增加一层 RNN 的方式) 有利于基于会话的推荐系统.

• RMN 在所有方法中表现最好. 例如, RMN 在 XING 数据集上, 相比于最好的对比方法 HRNN 而

表 2 RMN 及其变种 RMN-short (除去会话初始化单元和记忆写入单元), RMN-long (除去记忆读取单元) 在 XING 数据集上的 F1@K 性能对比

Table 2 The comparison of RMN with its short-term-only (removing session initializer and inter-session memory writer) and long-term-only (removing intra-session memory reader) variants on XING dataset evaluated by F1@K

	F1@2	F1@5	F1@10	F1@20	F1@50
RMN	0.0743	0.0571	0.0415	0.0261	0.0136
RMN-short	0.0345 (-53.6%)	0.0408 (-28.5%)	0.0291 (-29.9%)	0.0190 (-27.2%)	0.0088 (-35.3%)
RMN-long	0.0411 (-44.7%)	0.0455 (-20.3%)	0.0331 (-20.2%)	0.0220 (-15.7%)	0.0114 (-16.2%)

言, 分别实现了 26.1% 的 F1@10 性能提升和 20.6% 的 MRR@10 性能提升. 这个结果表明, 对用户的短期兴趣进行细粒度建模可以进一步提升基于会话的推荐系统的性能.

4.4.1 对用户长短期兴趣进行建模的效果

为了更清楚地展示对用户长期和短期兴趣进行建模的效果, 我们设计了两个 RMN 的变种方法: (1) RMN-short 移除了会话初始单元和记忆写入单元, 因此它忽略了用户的历史记录, 只刻画了用户的短期行为; (2) RMN-long 移除了记忆读取单元, 因此它将用户的短期兴趣视为静态不变的. F1@K 的结果如表 2 所示, 从中我们可以清楚地观察到, 在短期或长期模块缺失的情况下, RMN 的性能会有所退化. 平均而言, RMN-short 的 F1@K 指标平均比 RMN 低 34.9%, RMN-long 的 F1@K 指标平均比 RMN 低 23.4%. 这个结果验证了对用户的长期和短期兴趣进行建模所带来的性能的额外提升. 而且, 当 K 较小时 ($K = 2$), RMN-short/long 与 RMN 的性能差距相对更大, 这也说明 RMN-short/long 在真实推荐场景下无法取得满意的效果, 因为在真实场景中用户一般只会浏览头部的少量推荐结果. 另外, 我们也注意到 RMN-long 的性能比 RMN-short 的性能更好. 由于 RMN-long 只移除了记忆读取单元, 其本质上更接近传统的基于会话的推荐方法 HRNN; 而 RMN-short 移除了会话初始单元和记忆写入单元, 这样就割裂了不同会话之间的联系, 使得 RMN-short 更难刻画会话的特性. 因此, 这个结果也说明了基于会话进行物品推荐的重要性.

4.4.2 预测时使用的历史会话的数量对性能的影响

在预测用户的下一个点击商品时, 我们改变了该用户的历史会话的数量来研究性能会如何变化. 具体地说, 对于一个给定的用户, 我们使用其全部的数据来训练 RMN, 然后使用训练好的模型在以下 3 种情况下来预测该用户在最后一个会话中点击的商品: (1) 使用其最近的 1 个会话的信息; (2) 使用其最近的 4 个会话的信息; (3) 使用其所有历史会话的信息. 在 XING 数据集上的结果如图 4 所示. 该结果表明, 随着历史会话的长度从 1 提升到 4, RMN 的性能在逐步提升; 但是当所有的历史会话都被使用时, 性能发生了微弱的下降. 这是因为当预测用户在当前会话中的行为时, 更多的历史会话可以提供更多有用的信息. 但是随着历史会话长度的增加, 这种有用信息带来的边际效应会逐步减小. 当使用的历史会话过多时, 性能反而会开始下降, 这是因为太过久远的会话和当前会话的相似度和相关性一般都会很小, 因而带来了更多的噪声. 因此, 在预测用户在当前会话中的行为时, 需要合理地选择使用的历史会话的数量. 为了结果的清晰, 我们省去了历史会话长度等于中间其他值的结果, 真实的实验结果表明当历史会话的长度取 4 时, RMN 模型的性能达到了最优.

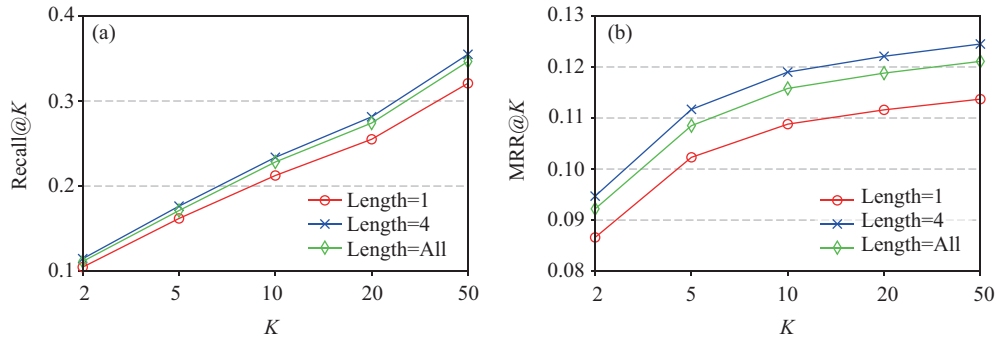


图 4 (网络版彩图) 预测时使用的历史会话的数量对性能的影响

Figure 4 (Color online) The impact of the length of users' historical sessions used in prediction. (a) Recall@K; (b) MRR@K

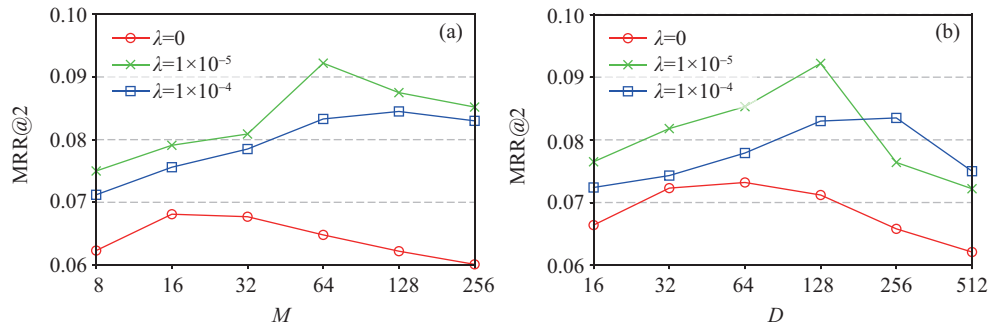


Figure 5 (网络版彩图) RMN 模型对超参数的敏感性. (a) 用户兴趣记忆向量的数量 M ; (b) 用户兴趣记忆向量的维度 D

Figure 5 (Color online) Parameters sensitivity of RMN. (a) The number of slots of preference memory; (b) the dimension of preference memory

4.4.3 超参数敏感性

我们研究了超参数 M 和 D , 即 RMN 中用户兴趣记忆向量的数量和维度对性能的影响. 我们将 M 和 D 的值分别从 8 变化到 256 和从 16 变化到 512, 然后记录 MRR@2 结果的变化. XING 数据集上的实验结果如图 5 所示, 其中每一条曲线对应了 RMN 模型的损失函数 (16) 中正则化损失项 \mathcal{L}_{reg} 的权重 λ 的取值. 我们从图 5(a) 观察到, 随着 M 的增加, 模型性能一开始也在提升, 这是因为更多的兴趣记忆向量可以存储更多有用的用户信息. 然而, 当 M 取值过大之后性能开始下降, 这是由于模型参数的大量增加而导致模型的能力过强, 最终产生了过拟合. 我们也可以从 λ 变化的角度来分析实验结果. 我们发现, 当 λ 取值为 0 时, RMN 模型的性能从 $M=16$ 之后就开始下降, 这是因为此时模型没有正则化项的约束, 对训练数据的拟合能力非常强, 从而在 M 值很小时就出现了过拟合现象. 而当 λ 取值较大 ($\lambda=1\text{E}-4$) 时, 由于正则化项的约束过大, RMN 模型有欠拟合的趋势, 因此这时在较大范围内 ($8 \sim 128$) 提升 M 的值对性能一直都有提高的作用. D 变化时, RMN 模型性能的变化趋势和分析结论与 M 类似, 如图 5(b) 所示.

5 相关工作

RMN 可以被视为现有两种基于会话的推荐系统 HRNN^[13] 和 RNN^[7] 的泛化和扩展. HRNN 是一个基于 RNN 的框架, 它具有一个额外的用户层级的 GRU, 用以模拟跨会话时的用户活动. 如果我们将 RMN 中的用户兴趣记忆模块替换为一个单个的兴趣向量 (等价于移除掉 RMN 中的记忆读取单元), 那么 RMN 模型就退化成了 HRNN 模型. 进一步地, 如果同时移除掉 RMN 中的记忆读取单元和记忆写入单元, RMN 模型会退化成会话独立的情形, 即文献 [7] 中提出的 RNN 模型. 一般而言, 兴趣记忆模块及其相关联的记忆读取和记忆写入单元使得 RMN 模型可以更好地刻画用户的长短期记忆, 并取得更好的实际性能.

基于会话的推荐系统也和流推荐系统 (streaming recommender systems)^[29,30] 相关. 流推荐系统的目的是基于用户的历史记录为用户推荐下一个可能点击的物品. 一般而言, 流推荐系统假设用户数据的数量更多, 产生速度更快, 算法也需要实时运行, 输出结果的延迟要更低. 文献 [29] 提出了一种基于记忆网络的流推荐系统模型, 并使用了生成对抗网络进行负样本采样. 文献 [30] 使用了基于池 (reservoir) 的流推荐系统方法进行主动样本采样, 并且使用了基于矩阵分解的注意力机制方法来对用户的历史记录进行建模. 但是这些方法都没有考虑不同会话之间的用户兴趣的差异, 而我们的方法使用了会话间的记忆写入单元对用户长期兴趣的转变进行了显式地刻画.

RMN 模型也和记忆网络的工作相关^[19~21]. 记忆网络 (memory networks) 是一种基于注意力的模型, 它们使用一个外部的记忆模块存储序列中的知识, 对记忆模块中的知识进行读取可以帮助记忆网络从序列中抽取长距离的依赖关系. 记忆网络被用到了很多任务中, 诸如情感分类^[31]、推荐系统^[18,22,32,33] 等. 我们的工作和现有工作的区别在于, 我们将记忆模块和 GRU 单元紧密地联系起来, 这种模型是专门为基于会话的推荐而设计的, 比传统的记忆网络模型更加精细.

RMN 模型从时间维度来考虑用户和物品的交互行为. 从另一个角度而言, 如果我们将用户和物品的交互看作是一个二分图, 那么我们可以从空间维度来对用户和物品的交互行为进行建模^[34,35]. 近年来, 图神经网络^[26,36] 是一种新兴的对图进行表征学习的方法. 也有一些相关工作尝试将图神经网络应用到推荐系统领域. 例如, 文献 [37] 在 Pinterest 中使用图神经网络对 pin-board 二分图进行表征学习; 文献 [38,39] 使用图神经网络对知识图谱进行表征学习, 然后将其应用到了推荐系统中. 基于这些工作, 将会话信息和二分图信息综合考虑并提出一种时间 - 空间联合模型也是可行的方向.

6 总结和未来工作

本文提出了循环记忆网络 RMN, 一种对用户长期和短期兴趣进行同时建模的基于会话的推荐系统. RMN 为每个用户分配了一个记忆模块, 用以存储其长期兴趣. 同时, 在 RMN 中我们设计了一个会话间记忆写入单元和会话内记忆读取单元, 分别用于刻画用户长期兴趣的转移和用户短期兴趣的动态变化. 本文在电影和职位推荐数据集上进行了实验, 实验结果表明 RMN 相对于现有方法取得了显著的性能提升. 实验结果也证实了在 RMN 中刻画用户长期和短期兴趣的必要性.

从实验结果观察到, 不同的用户在一个会话中点击过的商品具有高度的相关性. 在未来工作中, 我们打算把在一个会话中出现过的商品作为先验信息进行考虑, 以更好地提高推荐系统模型的性能.

参考文献

- 1 Zhou G, Zhu X, Song C, et al. Deep interest network for click-through rate prediction. In: Proceedings of the 24th

- ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, 2018. 1059–1068
- 2 van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. In: Proceedings of Advances in Neural Information Processing Systems, Lake Tahoe, 2013. 2643–2651
 - 3 Wang H, Zhang F, Xie X, et al. DKN: deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide Web Conference, Lyon, 2018. 1835–1844
 - 4 Lian J, Zhang F, Hou M, et al. Practical lessons for job recommendations in the cold-start scenario. In: Proceedings of the Recommender Systems Challenge 2017, Como, 2017. 1–6
 - 5 Yang X, Liang C, Zhao M, et al. Collaborative filtering-based recommendation of online social voting. *IEEE Trans Comput Soc Syst*, 2017, 4: 1–13
 - 6 Wang H, Wang J, Zhao M, et al. Joint topic-semantic-aware social recommendation for online voting. In: Proceedings of the 2017 ACM Conference on Information and Knowledge Management, Singapore, 2017. 347–356
 - 7 Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-based recommendations with recurrent neural networks. In: Proceedings of the 4th International Conference on Learning Representations, San Juan, 2016
 - 8 Shani G, Heckerman D, Brafman R I. An MDP-based recommender system. *J Mach Learn Res*, 2005, 6: 1265–1295
 - 9 Park S E, Lee S, Lee S G. Session-based collaborative filtering for predicting the next song. In: Proceedings of 2011 1st ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering, Jeju, 2011. 353–358
 - 10 Hidasi B, Quadrana M, Karatzoglou A, et al. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, Boston, 2016. 241–248
 - 11 Tan Y K, Xu X, Liu Y. Improved recurrent neural networks for session-based recommendations. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, 2016. 17–22
 - 12 Greenstein-Messica A, Rokach L, Friedman M. Session-based recommendations using item embedding. In: Proceedings of the 22nd International Conference on Intelligent User Interfaces, Limassol, 2017. 629–633
 - 13 Quadrana M, Karatzoglou A, Hidasi B, et al. Personalizing session-based recommendations with hierarchical recurrent neural networks. In: Proceedings of the 11th ACM Conference on Recommender Systems, Como, 2017. 130–137
 - 14 Li J, Ren P, Chen Z, et al. Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM Conference on Information and Knowledge Management, Singapore, 2017. 1419–1428
 - 15 Jannach D, Lerche L, Jugovac M. Adaptation and evaluation of recommendations for short-term shopping goals. In: Proceedings of the 9th ACM Conference on Recommender Systems, Vienna, 2015. 211–218
 - 16 Liu Q, Wu S, Wang L. Deepstyle: learning user preferences for visual recommendation. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, 2017. 841–844
 - 17 Ruocco M, Skrede O S L, Langseth H. Inter-session modeling for session-based recommendation. In: Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems, Como, 2017. 24–31
 - 18 Chen X, Xu H, Zhang Y, et al. Sequential recommendation with user memory networks. In: Proceedings of the 11th ACM International Conference on Web Search and Data Mining, Los Angeles, 2018. 108–116
 - 19 Weston J, Chopra S, Bordes A. Memory networks. In: Proceedings of the 3rd International Conference on Learning Representations, San Diego, 2015
 - 20 Sukhbaatar S, Weston J, Fergus R, et al. End-to-end memory networks. In: Proceedings of Advances in Neural Information Processing Systems, Montreal, 2015. 2440–2448
 - 21 Miller A, Fisch A, Dodge J, et al. Key-value memory networks for directly reading documents. 2016. ArXiv: 1606.03126
 - 22 Huang J, Zhao W X, Dou H, et al. Improving sequential recommendation with knowledge-enhanced memory networks. In: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, 2018. 505–514
 - 23 Cho K, van Merriënboer B, Gulcehre C, et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation. 2014. ArXiv: 1406.1078
 - 24 Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback. In: Pro-

- ceedings of the 25th Conference on Uncertainty in Artificial Intelligence, Montreal, 2009. 452–461
- 25 Jannach D, Ludewig M. When recurrent neural networks meet the neighborhood for session-based recommendation. In: Proceedings of the 11th ACM Conference on Recommender Systems, Como, 2017. 306–310
 - 26 Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. In: Proceedings of Advances in Neural Information Processing Systems, Long Beach, 2017. 1024–1034
 - 27 Kingma D P, Ba J. ADAM: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations, San Diego, 2015
 - 28 Tang Z, Wang D, Zhang Z. Recurrent neural network training with dark knowledge transfer. In: Proceedings of 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, 2016. 5900–5904
 - 29 Wang Q, Yin H, Hu Z, et al. Neural memory streaming recommender networks with adversarial training. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, 2018. 2467–2475
 - 30 Guo L, Yin H, Wang Q, et al. Streaming session-based recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, 2019. 1569–1577
 - 31 Li Z, Zhang Y, Wei Y, et al. End-to-end adversarial memory network for cross-domain sentiment classification. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, 2017. 2237–2243
 - 32 Wang H, Zhang F, Wang J, et al. Ripplenet: propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Turin, 2018. 417–426
 - 33 Wang H, Zhang F, Wang J, et al. Exploring high-order user preference on the knowledge graph for recommender systems. *ACM Trans Inf Syst*, 2019, 37: 1–26
 - 34 Wang H, Wang J, Wang J, et al. Graphgan: graph representation learning with generative adversarial nets. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, 2018. 2508–2515
 - 35 Wang H, Zhang F, Hou M, et al. Shine: signed heterogeneous information network embedding for sentiment link prediction. In: Proceedings of the 11th ACM International Conference on Web Search and Data Mining, Los Angeles, 2018. 592–600
 - 36 Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations, Toulon, 2017
 - 37 Ying R, He R, Chen K, et al. Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, 2018. 974–983
 - 38 Wang H, Zhao M, Xie X, et al. Knowledge graph convolutional networks for recommender systems. In: Proceedings of the World Wide Web Conference, San Francisco, 2019. 3307–3313
 - 39 Wang H, Zhang F, Zhang M, et al. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, 2019. 968–977

Recurrent memory networks: modeling long short-term user preferences for session-based recommendation

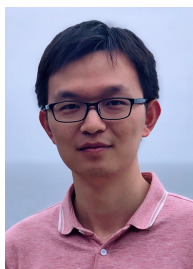
Hongwei WANG & Minyi GUO*

Emerging Parallel Computing Center, Shanghai Jiao Tong University, Shanghai 200240, China

* Corresponding author. E-mail: guo-my@cs.sjtu.edu.cn

Abstract In many online systems where recommendations are applied, interactions between a user and the system are usually organized into sessions. This paper studies how to model user preferences in session-based recommendation systems. Existing studies have either assumed that sessions are independent from each other and ignore long-term information from historical sessions, or treat the user's short-term preferences in a session as static, which cannot fully characterize user behavior in practical scenarios. Thus, we propose the recurrent memory network (RMN), which is an RNN-based framework that unifies the users' long-term and short-term preferences in session-based recommendation. The key component of the proposed RMN is preference memory, which stores a user's long-term interests. In addition, in the RMN, we design an intra-session memory reader and inter-session memory writer to facilitate explicit characterization of short-term (i.e., within a session) user preferences variation and long-term (i.e., cross-session) user preference transfer, respectively. The results obtained in extensive experiments on real-world datasets for movie and job recommendations demonstrate that the proposed RMN achieves substantial gains over state-of-the-art baselines.

Keywords recommender systems, session, long-short term user preferences, recurrent neural networks, memory networks



Hongwei WANG was born in 1992. He received his B.E. and Ph.D. degrees from Shanghai Jiao Tong University, China in 2014 and 2018, respectively. He is currently is a postdoctoral researcher at Stanford University. His research interests include machine learning and data mining, particularly in graph representation learning mechanisms, algorithms, and their applications in real-world data mining scenarios, e.g., recommender systems, knowledge graphs, social networks, and sentiment analysis.



Minyi GUO was born in 1962. He received his B.S. and M.E. degrees from Nanjing University, China in 1982 and 1986, respectively. He received his Ph.D. degree from University of Tsukuba, Japan in 1998. He is currently a professor at Shanghai Jiao Tong University. His research interests include automatic parallelization and data-parallel languages, bioinformatics, compiler optimization, high-performance computing, and pervasive computing.