

基于方向性偏好的个性化序列推荐模型

胡雨萱

东北大学理学院, 辽宁 沈阳

收稿日期: 2021年11月15日; 录用日期: 2021年12月10日; 发布日期: 2021年12月17日

摘要

目前推荐系统主要基于用户偏好和物品的相似度等标量进行推荐, 忽略了用户的偏好方向, 增加了不相关推荐的风险。基于此, 本文提出基于方向性偏好的个性化序列推荐模型, 通过推荐符合用户偏好方向的物品, 综合用户的偏好和需求进行推荐从而提高了推荐的准确性。本文以电影评论数据集为背景, 使用选择器多头自注意力机制和改进胶囊网络分别提取用户的长期偏好和短期需求, 然后根据用户长期偏好和短期需求构成的偏好向量进行推荐。实验结果表明, 模型的推荐准确性指标相比原模型提升了52%。

关键词

序列推荐, 偏好方向, 胶囊网络, 多头自注意力机制, 短期需求

Personalized Sequence Recommendation Model Based on Directional Preference

Yuxuan Hu

College of Science, Northeastern University, Shenyang Liaoning

Received: Nov. 15th, 2021; accepted: Dec. 10th, 2021; published: Dec. 17th, 2021

Abstract

The current recommendation system is mainly based on scalar variables such as similarity of user preferences and items, ignoring the user's preference direction, and increasing the risk of irrelevant recommendations. Based on this, this paper proposes a personalized sequence recommendation model based on directional preference, which improves the accuracy of recommendation by recommending items that conform to the user's preference direction and integrating the user's

preferences and needs. Based on the movie review data set, this paper uses the selector multi-head self-attention mechanism and the capsule network to extract the user's long-term preferences and short-term needs, and then recommends based on the user's long-term preferences and short-term needs. The experimental results show that the recommended accuracy index of the model is improved by 52% compared with the original model.

Keywords

Sequence Recommendation, Preference Direction, Capsule Network, Multi-Head Self-Attention, Short-Term Demand

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

传统推荐系统方法主要包含基于内容的推荐方法、基于人口统计学的方法和协同过滤方法[1] [2] [3], 其中运用和后续发展最多的是协同过滤方法。原始协同过滤方法通过研究用户对项目的评分矩阵, 分解出用户特征和项目特征, 然后基于用户特征和项目特征为用户推荐相似项目或者为用户推荐与其相似的用户喜欢过的物品。近年来随着深度学习的兴起, 深度学习也被运用在了推荐系统中并且大大提升了推荐准确度。深度学习在推荐系统中的运用主要分为三类, 分别是 DeepFM 类模型, 卷积神经网络和循环神经网络[4] [5] [6], 其中多层感知机和卷积神经网络主要用来学习输入数据的特征[7], 学习特征之间的相互关系, 而循环神经网络主要用来处理用户行为序列[8], 从序列中学习用户的特征演化。

序列推荐是推荐系统的一个研究方向, 序列推荐[9]通过研究用户按照时间产生项目交互序列, 从序列之间的相关性出发研究用户兴趣随时间变化的特征, 从而做出推荐。

序列推荐的挑战主要有两点: 1) 用户的偏好和需求是随时间动态变化的[10], 并且用户与项目的交互通常发生在连续的上下文中; 2) 如何从长序列中学习长期序列之间的依赖关系和高阶顺序依赖关系[11]。对于这两个挑战, 目前的解决方法是高阶马尔可夫模型和循环神经网络类模型。高阶马尔可夫模型[12]可以学习序列数据之间的高阶依赖关系, 但其模型需要估计的参数随着阶数的增长呈指数型增加, 不适用于长序列数据。传统的循环神经网络[13]如 RNN 除了可以学习序列数据之间的依赖关系, 还能学习数据的顺序关系, 但当序列过长时, RNN 会出现梯度消失/爆炸的问题。后来出现的 LSTM 模型通过引入门控单元一定程度解决了 RNN 的长距离依赖问题, 但是 RNN, LSTM 等循环神经网络都对序列的次序有很强的假设: 1) 任何临近的商品在序列中都是高阶依赖关系; 2) 输入的数据有严格的次序相关性。从这两个假设可知, 当输入的数据不是严格符合次序相关性时, 循环神经网络类模型的表达效果不佳。

从上面可以看出, 目前主流的两种方法都有着一定的限制性。为了更准确地学习用户序列的特征, 另一个突破点是结合多个方法的混合模型。例如, Tang 等人[14]提出了分别使用 RNN 和注意力对不同距离关系的序列进行特征捕捉, 从而达到了对序列的长距离依赖关系的学习。Li 等人[15]则使用 GRU 预测出用户对电影的体裁偏好, 然后结合改进的协同过滤进行电影推荐。Li 等人[16]还提出了 DESR 的模型, 分别使用混合高斯模型和胶囊网络学习用户的偏好和需求。这些模型都将用户数据分段, 然后根据数据的特点分别使用不同的模型, 这样做的好处是可以根据数据的特点选择更适合的模型。

然而以上模型的推荐准则都是使用例如用户偏好和物品相似度等标量进行推荐, 用户偏好和物品的

相似度越高，则该物品越容易被推荐。但是单一标量准则忽略了用户的偏好方向，只能表达标量所表达的信息，容易为用户推荐相似度高但是不符合用户偏好/需求的物品。根据这个情况，DESR模型首次提出了偏好方向的概念，偏好方向定义为短期需求指向长期偏好，使用偏好方向进行推荐的优点是：综合了用户的长期偏好和近期的需求从而提高了推荐的准确性，减少了不相关推荐的风险。

尽管使用偏好方向的推荐综合考虑了用户的长期偏好和短期需求使得推荐更准确，但是DESR模型对于用户的长期偏好只是固定聚类表达了其单一的长期偏好，并没有考虑其长期偏好之间的依赖性和顺序演化，同样对于用户的短期需求也没有考虑其短期序列之间的依赖性。综上所述，可以建立模型来模拟用户长期和短期序列之间的依赖性和顺序演化，提升推荐的准确性。本文的贡献：

- 1) 根据用户行为习惯个性化划分用户序列。根据每个用户不同的行为习惯特征，自适应地划分用户的长期和短期序列；
- 2) 使用带选择器的多头自注意力机制进行用户潜在长期偏好提取并学习长期偏好之间的依赖性和顺序演化，增加用户本质长期偏好的权重，减小噪声行为对长期偏好提取的影响；
- 3) 使用带注意力的胶囊网络表示用户短期需求，通过增加模型对用户近期需求的关注度来抽取用户短期需求序列的潜在特征；
- 4) 基于短期需求指向长期偏好的用户偏好向量的方向进行推荐，使得模型专注于用户偏好和短期需求，提升模型推荐的准确性。

2. 相关工作

2.1. 带选择器的多头自注意力机制

多头自注意力机制在自然语言处理的任务中获得了很好的运用，其优点主要在于将输入序列映射到多个不同的特征子空间从而更好的学习输入元素之间的依赖关系。结合了位置向量的多头自注意力机制克服了RNN类模型中的长序列数据导致的梯度消失/爆炸问题，还可以并行运算，增加了模型的工程实现能力。在本文中，使用结合选择器的多头自注意力机制，可以很好地提取用户长期序列中潜在的长期偏好，减少噪声行为的影响。

2.1.1. 多头自注意力机制

假定有一个 h 个头的多头自注意力模型，输入序列为 $X \in R^{N \times d_k}$ ，其中第 i 个头的计算过程如下：

$$head_i = ATT(Q_i, K_i, V_i) \quad (1)$$

$$ATT(Q_i, K_i, V_i) = softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \quad (2)$$

其中 $Q_i = XW_i^Q, K_i = XW_i^K, V_i = XW_i^V$ ， $\sqrt{d_k}$ 是放缩因子，作用是防止 $softmax$ 函数由于内层的内积过大而造成梯度爆炸/消失， d_k 为嵌入的维度。则多头自注意力的计算公式为：

$$MultiHead(Q, K, V) = concat(head_1, head_2, \dots, head_h)W^O \quad (3)$$

其中 W^O 为待学习参数矩阵， h 个头的注意力权重计算除了使用点积，还可以使用拼接或者感知机等方式进行结合，本文使用点积。

多头自注意力还加入了位置向量来学习序列之间的顺序关系。位置向量加在注意力权重计算之前，加入的形式为直接与输入序列的编码相加，位置向量构造公式为：

$$\begin{cases} PE_{2i}(p) = \sin\left(\frac{p}{10000^{\frac{2i}{d_{pos}}}}\right) \\ PE_{2i+1}(p) = \cos\left(\frac{p}{10000^{\frac{2i}{d_{pos}}}}\right) \end{cases} \quad (4)$$

即将 ID 为 p 的位置映射为一个 d_{pos} 维的向量，其第 i 个元素的值为 $PE_i(p)$ 。

2.1.2. 选择器模块

传统自注意力网络模型考虑所有输入的元素，然而自注意力模型计算每个元素表示的时候仅仅只是将所有的输入的元素考虑在内，而没有考虑其对于当前元素的相关性。使用基于选择机制的自注意力可以针对每个计算表示的元素，动态地选择与之具有相关性的子集，然后作为输入进行后续自注意力计算。

选择器注意力的计算过程为：

$$ATT_k^S(A, Q, K) = \text{softmax}\left(A \frac{QK^T}{\sqrt{d}}\right)V \quad (5)$$

$$A = \text{sigmoid}(QK^T) \quad (6)$$

选择器模块对初始注意力权重进行 *sigmoid* 操作，保留大于阈值的权重，去掉小于阈值(本文选取阈值为 0.5)的权重进而达到减小噪声影响，保留影响大的数据的作用。

2.2. 胶囊网络

胶囊网络是由 Hinton [17]提出的一种可解释性的神经网络。胶囊网络最初用于图像识别领域，相比于传统 CNN 模型，胶囊网络可以学习特征之间的位置关系并且可解释性更强，其结构见图 1：

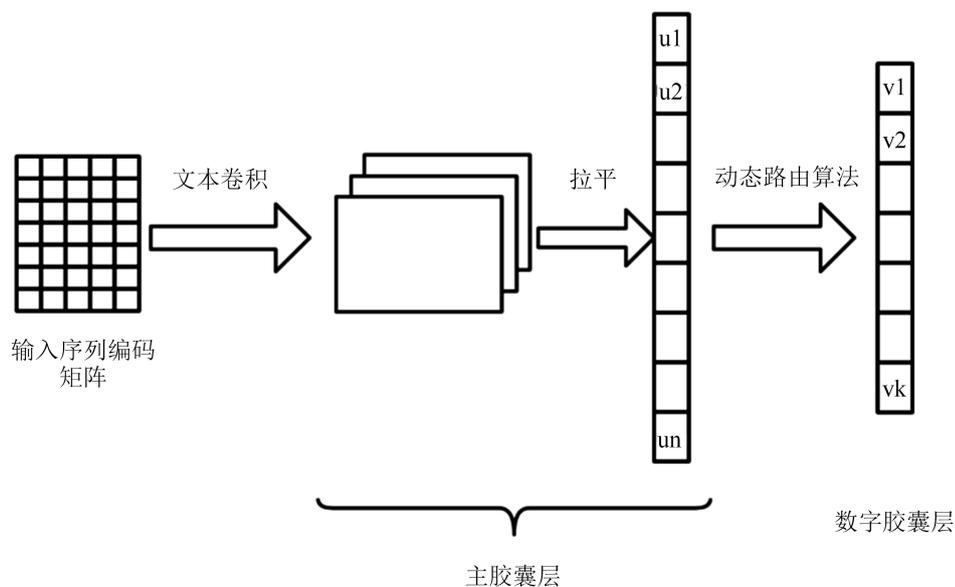


Figure 1. Capsule network structure

图 1. 胶囊网络结构图

图 1 中，胶囊网络的通过一种迭代的“动态路由”算法抽取输入数据的特征，通过动态路由算法，多次迭代之后输出的向量会聚集成可以代表之前输入向量高层特征。假设有底层输入特征 $u = (u^1, u^2, \dots, u^n)$ ，经过胶囊网络提取高层特征 $v = (v^1, v^2, \dots, v^k)$ 的计算过程如下：

$$v^i = W^i u^i \quad (7)$$

$$s = \sum_{i=1}^n c_i v^i \quad (8)$$

$$v = \text{Squash}(s) \quad (9)$$

$$\text{Squash}(s) = \frac{\|s\|^2}{1 + \|s\|^2} \frac{s}{\|s\|} \quad (10)$$

其中 W^i 由反向传播确定， c_i 的值由动态路由算法确定。 $\text{Squash}()$ 称为挤压操作，用来“挤压”向量的模且不变向量方向。

动态路由算法确定 c_i 值的伪代码如下：

```

For all capsules i in layer l and for all capsules j in layer l+1:  $b_{ij} \leftarrow 0$ .
For  $\gamma$  iterations do:
  For all capsule i in layer l:  $c_i \leftarrow \text{softmax}(b_i)$ 
  For all capsule j in layer (l+1):  $s_j \leftarrow \sum_i c_{ij} \hat{\mu}_{ji}$ 
  For all capsule j in layer (l+1):  $v_j \leftarrow \text{Squash}(s_j)$ 
  For all capsule i in layer l and capsule j in layer (l+1):  $b_{ij} \leftarrow \hat{\mu}_{ji} v_j$ 
Return  $v_j$ 

```

3. 基于方向性偏好的个性化序列推荐模型

3.1. 模型结构

本文提出的基于方向性偏好的个性化序列推荐模型 SDESR 的框架如图 2 所示，其主要分为四部分：1) 输入层：输入用户的历史行为序列并自适应地划分为长期序列和短期序列，然后进行编码；2) 长期偏好提取层：使用带选择器的多头自注意力机制基于长期序列提取用户的长期偏好；3) 短期需求推断层：使用基于注意力的胶囊网络提取高层用户短期需求特征；4) 方向性偏好推荐层：根据前两步求出的用户长期偏好和短期需求向量构建用户的偏好向量，最后根据偏好向量推荐符合用户短期需求和长期偏好的物品加入用户的推荐列表。

3.2. 自适应划分历史行为序列

由于每个用户的行为习惯不同，其偏好的形成和演化过程也不同，所以需要根据用户的行为习惯来划分用户的行为序列，于是本文提出习惯因子特征 $habit_factor$ ，根据习惯因子特征自适应地划分用户短期行为序列会话。

假定有用户 u 的行为序列嵌入 $H = (h^1, h^2, \dots, h^n)$ ，每个行为发生的时间戳与上一次行为的时间间隔序列为 $T^{H_n} = (t^1, t^2, \dots, t^{n-1})$ ，对于一个包含 $l (< n)$ 个行为的序列片段 H_l ，其习惯因子特征为：

$$habit_factor_{H_i} = \lambda D(T^{H_i}) + (1 - \lambda) D(H_i) \quad (11)$$

其中 $D()$ 表示方差。习惯因子特征刻画了该段行为序列中每个行为发生的时间与前一行为发生的时间紧密性以及该段行为序列内部联系的紧密性。给定阈值 hf (由数据的分布特点决定), 本文将 $habit_factor_{H_i} < hf$ 的行为划分成同一段短期行为序列。

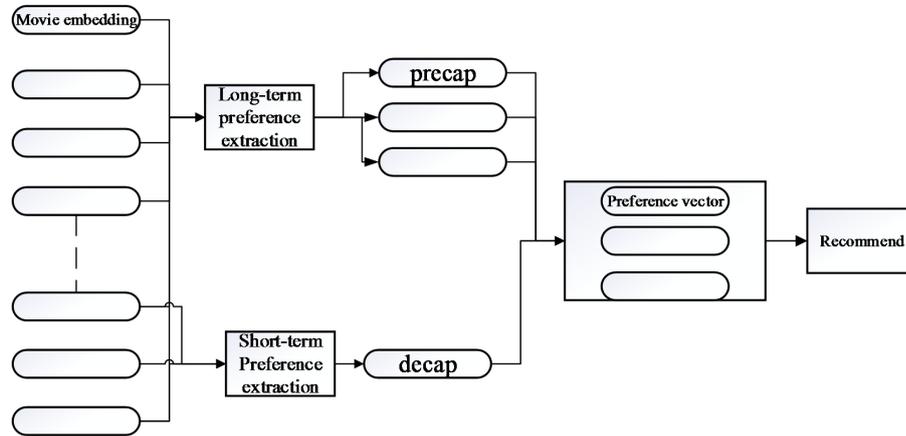


Figure 2. SDES model structure
图 2. SDES 模型框架

3.3. 长期偏好提取

由于每个人的偏好是不断演化的, 并且受到之前的行为影响会丢弃或者新生成一些长期偏好, 所以在用户的序列数据中存在一些噪声行为, 根据此考虑在学习用户长序列数据特征时“去伪存真”, 使用选择器自注意力保留用户本质的偏好, 去除不相关的噪声行为的影响。

长期偏好提取部分以用户长期序列编码矩阵作为输入, 输出用户的长期偏好向量, 具体结构框架如图 3 所示:

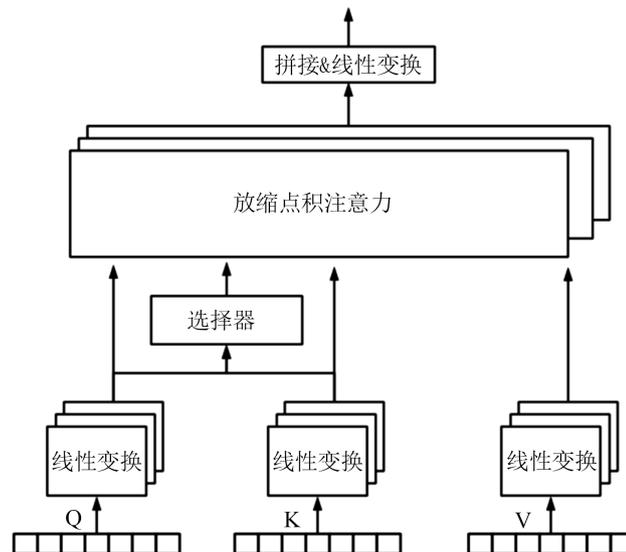


Figure 3. Selector multi-head attention network framework
图 3. 选择器多头注意力网络框架

给定用户长期序列行为的编码矩阵 $X_{long}^{n \times d_k}$ ，首先输入带 h 个头的选择器多头自注意力网络学习用户行在不同特征子空间的依赖和顺序关系，其中第 i 个头的计算过程为：

$$head_i = ATT_K^S(A_i, K_i, V_i) \quad (12)$$

$$ATT_K^S(A_i, K_i, V_i) = softmax\left(A_i \frac{Q_i K_i^T}{\sqrt{d_k}}\right) V \quad (13)$$

$$A_i = sigmoid(Q_i K_i^T) \quad (14)$$

其中 $Q_i = W_i^Q X_p$ ， $K_i = W_i^K X_p$ ， $V_i = W_i^V X_p$ ， X_p 为经过位置编码后的序列矩阵， $X_p = X_{long}^{n \times d_k} + PE(X_{long}^{n \times d_k})$ 。最后将所有头的输出拼接并线性映射成 k 个长期偏好表示 $precap^{k \times d_k}$ ：

$$Head = Concat(head_1, head_2, \dots, head_h) W^C \quad (15)$$

$$precap = Head W^O \quad (16)$$

其中 W^C 和 W^O 为待学习的参数。

选择器多头自注意力模型的目标函数为：

$$MSE(precap, y_{long}) = \frac{1}{k} \sum_{i=1}^k (precap_i - y_{long})^2 \quad (17)$$

其中 y_{long} 为用户目标推荐列表中所有标签的均值。

3.4. 短期需求提取

用户当前的需求主要受近期的行为影响，并且影响程度随着时间的前溯减弱。基于此，在胶囊网络中引入注意力机制，对胶囊网络的输出进行注意力加权来对用户短期需求进行提取。

模型输入用户的短期行为序列 $X_{short}^{n \times d_k}$ ，输出提取的用户短期需求特征 $\widehat{decap}^{d \times d_k}$ 。模型训练时，第 i 段短期行为序列的下一行为嵌入 $decap_i$ 作为其标签。

预测需求特征 \widehat{decap} 的计算过程如下：假定胶囊网络有 d 个输出胶囊，对一组含有 M 条记录的样本，输出其 d 个胶囊分别学习到的 d 个特征 $cap = (cap_1, \dots, cap_d)$ ，将第 M 条记录嵌入作为注意力的查询 Q_m 分别对 d 个特征进行注意力加权：

$$ATT(Q_m, K_{cap}) = softmax(Q_m K_{cap}^T) \quad (18)$$

$$decap = ATT(Q_m, K_{cap}) V_{cap} \quad (19)$$

其中 $Q_m = X_{short, M} W^Q$ ， $K_{cap} = cap W^K$ ， $V_{cap} = cap W^V$ 。

虽然用户最近一次行为最能代表其近期需求，但是推荐列表不是一次性的推荐，而是对用户未来一段时间的预测，需要考虑用户需求的演化性，所以为了更好地预测用户未来一段时间的行为，增加 L_a 作为辅助损失，自定义损失函数 Loss 如下：

$$L = \frac{1}{n} \sum_{j=1}^N \sum_{i=1}^d (decap_j - \widehat{decap}_{ji})^2 \quad (20)$$

$$L_a = \sum_{j=1}^N (\widehat{decap}_{m_j} - decap_{s_j})^2 \quad (21)$$

$$Loss = L + \alpha L_a \quad (22)$$

其中, $decap_j$ 是第 j 组短期序列的标签, \widehat{decap}_{ji} 是第 j 组短期序列的第 i 个预测需求特征。 \widehat{decap}_{m_j} 是第 j 组短期序列的 d 个预测特征的均值, $decap_{s_j}$ 是第 j 组短期序列的下 s 条标签的均值, α 是辅助损失的权重。增加了辅助损失的模型更能综合表达用户未来一段时间的综合需求。

3.5. 偏好向量计算及推荐

给定目标用户的 k 个长期偏好向量 $precap = \{precap_1, \dots, precap_k\}$, 短期需求向量 $decap$, 则用户的偏好向量一共有 k 个, 其中第 i 个偏好向量 $scap_i$ 为:

$$scap_i = precap_i - decap \quad (23)$$

给定候选物品嵌入 $T^{t \times d_k}$, 模型为每个用户推荐固定个数 LR 个物品, 根据角度矩阵 $Angle$ 和距离矩阵 Dis 构成的得分矩阵 $Score$ 给出推荐:

$$Score = (1 - m_base) Angle + m_base(1 - Dis) \quad (24)$$

$$Angle = \cos(T, scap) \quad (25)$$

$$Dis = dis(T, decap) \quad (26)$$

其中 $\cos(x, y) = \frac{x^T y}{\|x\| \|y\|}$, $dis(x, y) = \frac{(x - y)^2}{dis_{max}}$, dis_{max} 为 Dis 的最大值。

根据推荐准则 $Score$ 可以看出, 对于候选物品集合中的每一个物品, 若其方向与偏好向量方向夹角越小, 其距离短期需求向量的距离越小, 则其被推荐的概率就越大。

4. 实验结果与分析

4.1. 数据集及实验设置

4.1.1. 数据集

本文在真实电影数据集 `movielens-1m` 上进行实验, 该数据集包含了电影推荐服务[movielens](<http://movielens.org>)的5星评分和自由文本标记活动的信息。其中有9742部电影的100,836个评级和3683个标签。

4.1.2. 实验设置

本文仿真实验在 `macOS 11.4` 环境下进行, 使用 `python3.8`, 基于 `pytorch` 框架进行开发实验。

本文将数据集中用户的电影历史记录按照时间戳排序, 将用户按照 8:2 的比例划分数据集。首先基于电影标签数据使用 `Struc2Vec` 图嵌入方法构建每部电影节点与其对应标签节点的图并计算电影节点的嵌入表示。根据 `Struc2Vec` 算法原理, 拥有相同或者相似的标签的电影节点其嵌入表示向量之间的距离也越近。

4.2. 评价指标

给定目标用户的预测推荐列表 $\{t_1, \dots, t_{LR}\}$ 和标签电影列表 $\{y_1, \dots, y_{label_len}\}$, 基于方向偏好的评价指标定义如下:

1) 推荐列表准确度

$$accuracy = 1 - \frac{1}{LR} \sum_{i=1}^{LR} \min(\{dis(t_i, y_1), dis(t_i, y_2), \dots, dis(t_i, y_{label_len})\}) \quad (27)$$

$$dis(t_i, t_j) = \frac{\sqrt{dMSE(t_i, t_j)}}{dis_{max}} \quad (28)$$

其中 t_i 为推荐列表中的待推荐电影嵌入, dis_{max} 为距离归一化参数, 取所有电影嵌入之间距离的最大值。

2) 长期偏好准确度

$$acc_pre = 1 - \frac{1}{LR} \sum_{i=1}^{LR} \min(\{dis(t_i, precap_1), dis(t_i, precap), \dots, dis(t_i, precap_k)\}) \quad (29)$$

acc_pre 是衡量推荐列表中待推荐电影与目标用户的长期偏好的准确度, acc_pre 值越大, 则表示推荐列表越符合目标用户的长期偏好。

3) 短期需求准确度

$$acc_de = 1 - \frac{1}{LR} \sum_{i=1}^{LR} dis(t_i, decap) \quad (30)$$

acc_de 是衡量待推荐电影符合目标用户短期需求的准确度, acc_de 值越大, 则推荐列表越符合目标用户短期需求。

4.3. 对比模型

为了验证本文提出的 SDES R 模型的有效性, 选取以下模型作为对比模型:

- 1) RAND: 基于随机的方法, 基于本文的数据集是从用户电影历史记录中没有的电影中随机生成推荐。
- 2) LSTM: 长短时记忆网络, 可以学习序列之间的顺序性和长距离依赖关系, 相比于 RNN 其能在更长的序列中有更好的表现。
- 3) KFN: 原理与 KNN 相似, 即根据邻居的行为提供推荐。
- 4) DESR: 首次提出使用方向性偏好进行推荐的模型。

4.4. 实验结果与分析

4.4.1. 模型训练超参数分析

本节主要分析 SDES R 模型中两个主要模型的超参数影响。首先是带选择器的多头自注意力网络的超参数 $precap_num$, 其表示多头自注意力网络输出长期偏好的个数, 实验结果见图 4:

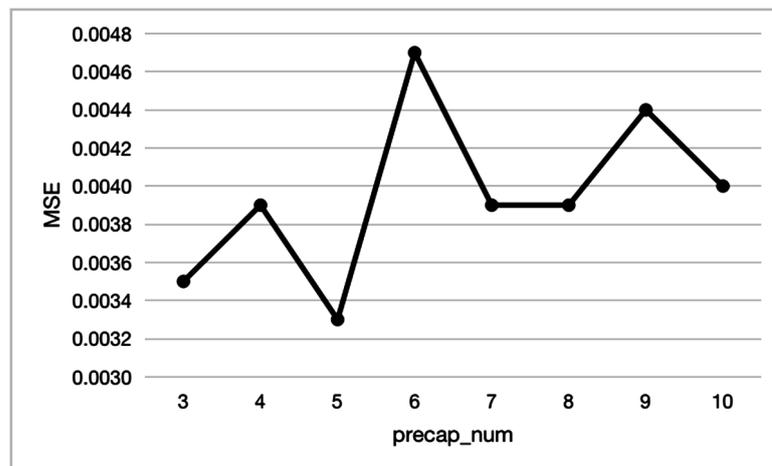
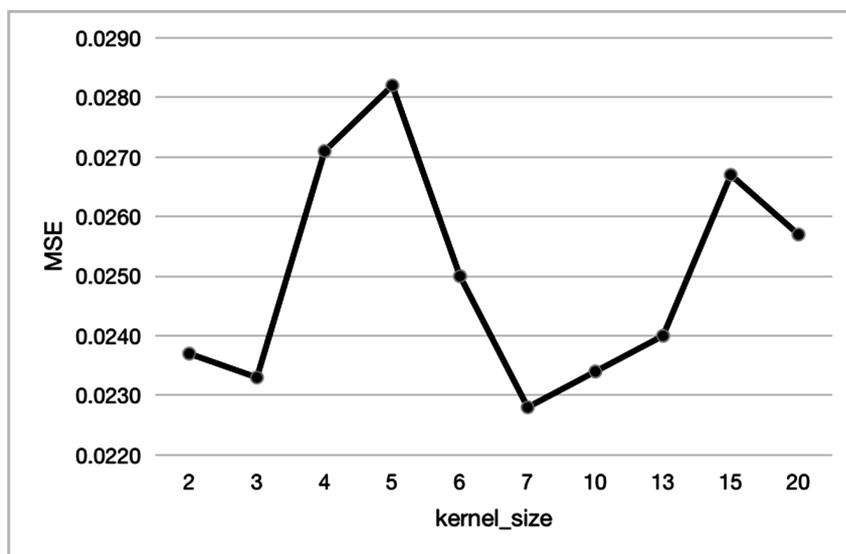


Figure 4. The influence of $precap_num$ on the MSE

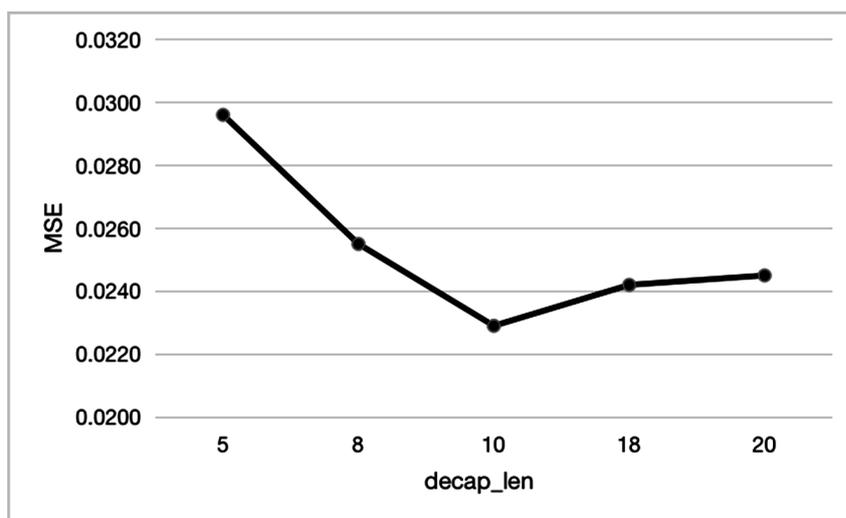
图 4. 自注意力网络输出长期偏好个数对模型精度影响

从图 4 可以看出, 当 $precap_num = 5$ 时, 模型的 MSE 最小, 对于用户的预测偏好最准确, 于是选取输出长期偏好个数 $k = 5$ 。

其次是带注意力的胶囊网络训练超参数: $decap_len$ 和 $kernel_size$, 分别表示胶囊网络损失函数的标签个数和胶囊网络中每个胶囊抽取序列的窗口长度。实验结果如图 5:



(a) 胶囊窗口长度对模型结果影响



(b) 标签个数对模型结果影响

Figure 5. Capsule network model hyper parameters

图 5. 胶囊网络模型超参数

从图 5 可以看出, 当窗口长度 $kernel_size = 7$, 标签个数 $decap_len = 10$ 时, 模型效果最好。

最后是模型的推荐超参数 m_base , 其表示得分矩阵中短期需求距离矩阵的权重, 当 m_base 值越大时, 符合用户短期需求的物品越会被推荐, 实验的结果见图 6。

从图 6 可以看出, 随着 m_base 值增加, 模型的长期偏好准确性上升幅度很小, 短期需求准确性上升幅度最大, 总体准确性上升幅度也随着短期需求上升而上升。根据实验结果, 本文选取 $m_base = 0.9$ 。

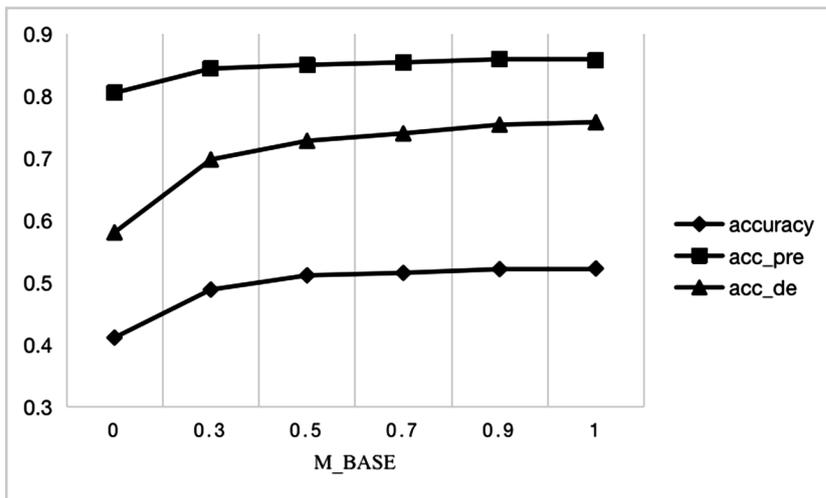


Figure 6. The influence of m_base on the model recommendation effect
 图 6. 得分矩阵超参数对模型推荐效果影响

4.4.2. 模型结果分析

基于上一节确定的超参数值，在为用户推荐 $LR = 10$ 个电影的情况下，对比了本文提出的模型和对比模型之间的效果，对比结果见表 1:

Table 1. Comparison results of different models
 表 1. 不同模型对比结果

模型	accuracy	acc_pre	acc_de
RAND	0.150	0.129	0.291
KFN	0.225	0.268	0.349
LSTM	0.454	0.420	0.533
DESR	0.344	0.668	0.429
SDESR	0.523	0.857	0.758

从表 1 可以看出，本文提出的 SDESR 模型在三个指标下的表现效果均为最佳，其中总体准确度 $accuracy$ 相比 DESR 模型提升了 52%，相比传统序列推荐模型 LSTM 提升了 15%，相比 KFN 和 RAND 分别提升了超过 80%。在长期偏好准确度 acc_pre 方面，SDESR 模型比 DESR 模型提升了 28%。短期需求准确度 acc_de 方面，SDESR 模型相比 DESR 提升了 76%。综上所述，SDESR 模型在长期偏好和短期需求表达方面表现均优于对比模型，相比基于标量的推荐，基于偏好向量的推荐更加符合用户偏好。

4.4.3. 模型意外性推荐效果分析

一个好的推荐系统不仅要推荐符合用户偏好的项目，还要有开拓用户视野的能力，本节探究了 m_base 参数对模型多样化和差异化推荐的影响，定义模型多样性指标 div 和差异性指标 $difference$:

$$div = \frac{1}{LR^2} \sum_{i=1}^{LR} \sum_{j=1}^{LR} dis(t_i, t_j) \quad (31)$$

$$difference = \frac{1}{LR * LH} \sum_{i=1}^{LR} \sum_{j=1}^{LH} dis(t_i, t_j) \quad (32)$$

其中 LH 为目标用户历史记录中的电影数, div 反映了推荐列表中各推荐项目之间的不同程度, $difference$ 反映了推荐列表与用户历史记录的差异程度。实验结果见图 7:

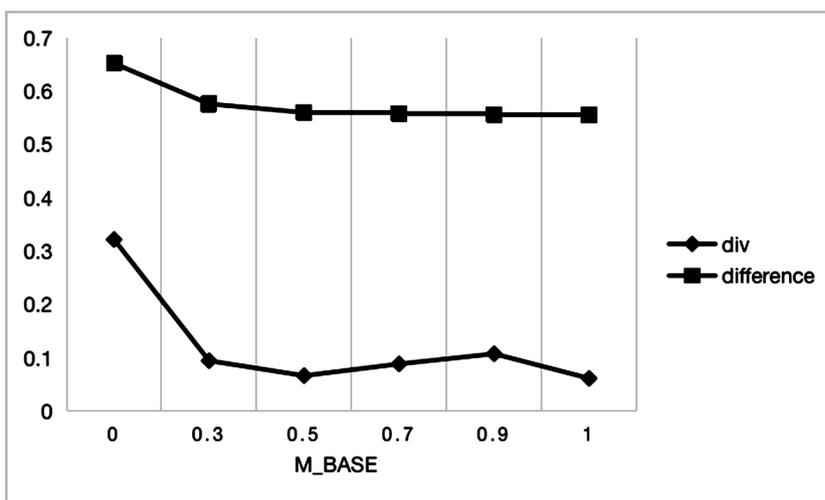


Figure 7. The influence of m_base on model diversity and differentiated recommendation

图 7. m_base 对模型多样性和差异化推荐的影响

从图 7 可以看出, 随着 m_base 值增加, 即当模型越来越关注用户短期需求时, 模型的多样性和差异性效果均下降。其中模型推荐差异性随着 m_base 值增加下降幅度较小, 而多样性下降和波动幅度较大, 说明 m_base 值主要影响模型的推荐多样性, 即模型越关注用户的短期需求, 模型推荐的物品越单一化。

5. 结束语

目前大部分推荐模型主要基于用户偏好与项目之间的相似度等标量进行推荐, 忽略了用户的偏好方向, 本文基于方向性偏好, 综合用户长期偏好和短期需求为用户推荐既符合用户偏好又专注用户需求的物品。本文提出的基于方向性偏好的个性化序列推荐模型, 利用用户的序列数据, 使用带选择器的多头自注意力机制学习长序列数据中的顺序关系和依赖关系, 保留重要特征, 去除噪声信息, 有效地提取了用户的长期偏好, 并且专注用户的近期需求, 通过提高模型对于用户近期的需求信息的关注来提高用户的短期需求预测准确度从而提升模型的推荐效果。实验结果表明, 本文提出的模型相比于原模型在长期偏好和短期需求方面的推断准确性有很大的提升, 使得本文模型的推荐效果进一步提升, 但本文提出的模型在多样性推荐和差异性推荐方面还未有很有有效的解决方法, 希望在未来的工作中提供更加多样化的推荐。

参考文献

- [1] 马宏伟, 张光卫, 李鹏. 协同过滤推荐算法综述[J]. 小型微型计算机系统, 2009, 30(7): 1282-1288.
- [2] Folajimi, Y. and Olowofoyeku, K. (2014) Web Items Recommendation Using Hybridized Content-Based and Collaborative Filtering Techniques. *Journal of Computer Science and Its Application*, **21**, 64-72.
- [3] Liang, Z., Peng, L.-F. and Phelan, C.A. (2014) Novel Recommendation of User-Based Collaborative Filtering. *Journal of Digital Information Management*, **12**, 165-175.
- [4] Zhou, T.-X., Jiang, Z.-B., Liu, X.-J., et al. (2020) Research on the Long-Term and Short-Term Forecasts of Navigable River's Water-Level Fluctuation Based on the Adaptive Multilayer Perceptron. *Journal of Hydrology*, **591**, Article ID: 125285. <https://doi.org/10.1016/j.jhydrol.2020.125285>

-
- [5] 周飞燕, 金林鹏, 董军. 卷积神经网络研究综述[J]. 计算机学报, 2017, 40(6): 1229-1251.
- [6] Yuan, W.-H., Hong, W., Yu, X.-M., *et al.* (2020) Attention-Based Context-Aware Sequential Recommendation Model. *Information Sciences*, **510**, 122-134. <https://doi.org/10.1016/j.ins.2019.09.007>
- [7] 宗春梅, 张月琴, 赵青杉. 等. 可视化支持下 CNN 在个性化推荐算法中的应用[J]. 计算机系统应用, 2020, 29(6): 204-210.
- [8] Wafa, S. and Byun, Y.C. (2020) A Context-Aware Location Recommendation System for Tourists Using Hierarchical LSTM Model. *Sustainability*, **12**, Article No. 4107. <https://doi.org/10.3390/su12104107>
- [9] 沈学利, 杜志伟. 融合自注意力机制与长短期偏好的序列推荐模型[J]. 计算机应用研究, 2020, 38(5): 1371-1375+1380.
- [10] Wang, S.-J., Hu, L. and Wang, Y. (2019) Sequential Recommender Systems: Challenges, Progress and Prospects. *Proceedings of the 28th International Conference on Artificial Intelligence, Macao (China)*, 10-16 August 2019, 6332-6338. <https://doi.org/10.24963/ijcai.2019/883>
- [11] Wu, C.-Y., Amr, A., Alex, B., Smola, A.J. and Jing, H. (2017) Recurrent Recommender Networks. *Proceedings of the 10th ACM International Conference on Web Search and Data Mining, Cambridge*, 6-10 February 2017, 495-503. <https://doi.org/10.1145/3018661.3018689>
- [12] Yang, Y., Jang, H.J. and Kim, B. (2020) A Hybrid Recommender System for Sequential Recommendation: Combining Similarity Models with Markov Chains. *IEEE Access*, **8**, 190136-190146. <https://doi.org/10.1109/ACCESS.2020.3027380>
- [13] 夏瑜潞. 循环神经网络的发展综述[J]. 电脑知识与技术, 2019, 15(21): 182-184.
- [14] Tang, J.-X., Belletti, F., Jain, S., *et al.* (2019) Towards Neural Mixture Recommender for Long Range Dependent User Sequences. *Proceedings of the World Wide Web Conference, San Francisco*, May 2019, 1782-1793. <https://doi.org/10.1145/3308558.3313650>
- [15] Li, X.-Q., Jiang, W.-J., Chen, W.G., *et al.* (2019) HAES: A New Hybrid Approach for Movie Recommendation with Elastic Serendipity. *Proceedings of the 28th ACM International Conference, Beijing*, November 2019, 1503-1512. <https://doi.org/10.1145/3357384.3357868>
- [16] Li, X.-Q., Jiang, W.-J., Chen, W.-G., *et al.* (2020) Directional and Explainable Serendipity Recommendation. *Proceedings of the Web Conference, Taipei*, April 2020, 122-132. <https://doi.org/10.1145/3366423.3380100>
- [17] Hinton, G.E. (2017) Dynamic Routing between Capsules. 31st Conference on Neural Information Processing Systems, Long Beach, 4-9 December 2017, 1-11.