

基于元学习的响应式推荐系统^①



武多才, 张 谧

(复旦大学 软件学院, 上海 201438)

通讯作者: 武多才, E-mail: duocaiw@qq.com

摘 要: 在信息时代, 面对浩繁的信息, 用户急需高效的推荐系统为他们选择最感兴趣的内容. 我们面临的一个重要问题是用户的兴趣会随时间发生变化并且不断有新用户使用系统. 因此, 一个好的推荐系统需要根据用户最新的少量交互信息及时响应用户兴趣的变化和快速捕捉到新用户的兴趣以更好地满足用户的需求. 但是, 根据我们的调研, 现有的推荐系统方法还没有很好地满足响应性 (responsiveness) 的需求. 为了解决这个问题, 我们提出了一套基于元学习 (Meta Learning) 的响应式推荐系统设计. 该方法能够及时响应用户最新的交互信息, 同时提高对老用户和新用户的推荐质量. 总的来说, 我们利用元学习从大量历史交互信息中挖掘出先验知识, 使得模型仅仅通过少量新的交互信息快速地学习出用户兴趣, 从而满足响应性的需求. 我们在 MovieLens 和 Nertflix 两个推荐系统中常用的公开数据集上验证了我们方法的有效性.

关键词: 推荐系统; 元学习; 矩阵分解; 响应性

引用格式: 武多才, 张谧. 基于元学习的响应式推荐系统. 计算机系统应用, 2021, 30(8):60-66. <http://www.c-s-a.org.cn/1003-3254/8015.html>

Responsive Recommender System Based on Meta Learning

WU Duo-Cai, ZHANG Mi

(School of Software, Fudan University, Shanghai 201438, China)

Abstract: In the era of information explosion, most users urgently need timely and effective recommendation service. As a result, the number of interactions, which a recommender system requires to recognize the drifted interests of existing users or the unknown preference of new users, would largely determine the application's survival rate in the highly competitive consumer market. However, for the best of our knowledge, this responsiveness aspect of recommender system is far from well-studied. To bridge this gap, we propose a task-based meta learning approach towards responsive recommender system, which helps improve the recommendation quality for both existing and new users after the system only observes a limited number of incoming interactions. Basically, the leverage of meta learning contributes to the fast adaption to the optimum of the underlying model with few interactions to satisfy the responsiveness requirement. Extensive experiments on MovieLens and Netflix datasets highly demonstrate the responsiveness of the proposed method.

Key words: recommender system; meta learning; matrix factorization; responsiveness

推荐系统能够帮助用户从浩繁的信息快捷地选出他们感兴趣的内容, 但是我们面临一个重要问题是用户的兴趣会随着时间发生变化^[1,2], 此外新用户也会不断到来. 根据最新的调研, 电商平台的销售量和其推荐

服务对用户的响应性呈现显著的正相关关系^[3]. 在这样的背景下, 我们提出响应性作为一个在线推荐系统需要满足的重要指标. 具体来说, 当系统收集到用户少量的最新交互信息之后, 一个响应式推荐系统需要满足

① 收稿时间: 2020-11-11; 修改时间: 2020-12-21; 采用时间: 2020-12-25; csa 在线出版时间: 2021-07-31

如下需求:

(1) 对老用户的响应性. 响应式推荐系统需要及时有效地捕捉到老用户兴趣的变化. 以图1中的上半部分为例, 对于一个老用户, 他之前都是比较喜欢科幻类型的电影. 但是最近, 该用户的兴趣发生了变化, 展现出了对情感类型电影的兴趣. 这种变化可能是受朋友影响或者个人经历变化等因素.

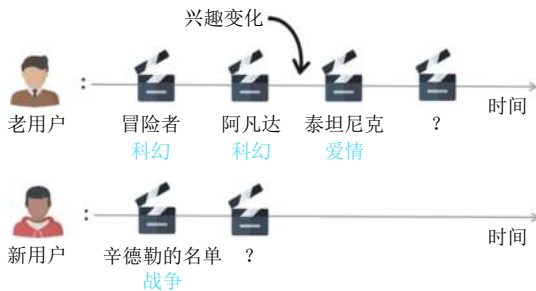


图1 两个典型的响应式推荐系统需求场景示意图

(2) 对新用户地响应性. 响应式推荐系统需要准确地学习新用户兴趣. 以图1中下半部分所示为例, 该用户为系统中新用户且只有只看了一部战争类型地电影.

接下来, 我们将这两点响应性需求统称为学习用户近期的兴趣. 根据我们的调研, 尽管响应性是推荐系统中一个重要的需求, 现有的方法还没有很好地满足该需求. 一方面, 传统的推荐系统方法采用线下训练线上预测的模式^[4,5]. 这些方法很难满足响应性的需求, 因为模型训练好之后就不再改变. 重新训练模型又十分耗时难以满足时间需求. 另一方面, 最近的一些研究工作提出了一些增量更新的在线推荐算法^[1,6-10]. 相比于重新训练模型, 这些方法很好地提升了模型更新的时间效率. 这些方法主要分为两类, 一类是在线下训练好的模型的基础上, 只使用新的交互信息微调 (fine-tuning) 模型^[8-10]; 另一类方法会缓存一部分交互信息到一个池子 (reservoir) 中, 然后定时更新模型, 更新模型是使用池子中的交互信息和新的交互信息同时微调模型^[1,7].

虽然这些基于微调的更新方法可以解决模型更新的时间效率问题, 他们面临着新的交互信息量太少的问题. 传统的机器学习模型需要大量的数据才能准确学习样本的表征, 以 MNIST 手写数字识别数据集为例, 该数据集有 6 万张图片, 平均每个数字有 6 千张图

片. 为了应对新的交互信息量太少的问题, 我们提出了一套基于元学习^[11]的增量式模型更新方案. 该方案能够从之前的利用大量的历史交互数据训练的过程中, 学习出一些解决训练数据量少问题的先验知识, 然后将这些先验知识应用于最近的利用少量新的交互信息的更新过程中, 从而指导模型更好地学习用户近期的兴趣.

元学习在许多小样本学习 (few shot learning) 的任务中都取得了超过了微调的方法^[11-13]. 其设计理念借鉴于人类的学习过程. 人类在学习新的种类的分类任务时, 往往只需要一两张样例图片就可以了, 这是因为人类在分类未见过的种类时, 会使用到之前学习到的先验知识, 而不是像传统的机器学习算法一样从头开始学习.

从元学习的角度看, 在我们的推荐场景中, 响应式推荐的要求是根据某个用户少量的最新的交互信息为其推荐下一个物品. 我们将这个需求定义为一个任务. 为了能够利用少量的样本学习出用户在新交互物品上的兴趣. 借鉴于元学习的思想, 我们需要为该任务准备先验知识. 接下来的问题是如何学习这样的先验知识. 为了学习先验知识, 我们首先利用大量的历史交互数据构建与当前任务相似的任务, 即使用用户相连的少量交互去预测紧接着的下一个交互. 从历史数据中, 我们可以构建大量的这样的任务. 然后我们使用模型的初始参数作为先验知识. 该初始参数能够使得模型在该任务上仅使用少量的样本就能准确学习出用户近期的兴趣. 为了学习到这样的初始参数, 借鉴于元学习的想法, 我们将该需求建模成线下训练过程中监督损失函数, 即线下训练的任务要在使用该初始参数的基础上, 经过少量交互的更新, 都能准确预测其任务中对应的下一个交互物品. 这样, 在新的任务中, 模型就可以在之前的相似任务中学习到的初始参数的基础上更好地学习出用户的兴趣.

从传统方法的角度看, 相比于传统的线下训练好模型然后线上微调模型的策略. 我们的方法的优势是在线下训练模型的时候考虑到了之后线上更新只会有少量的交互信息, 并将其设计在了线下更新模型参数的损失函数中. 这样学习到的模型参数能够在线上更新的过程中仅使用少量的新的交互信息就能更准确地学习出用户近期的兴趣.

后文组织如下: 第1节介绍我们提出的基于元学

习的响应式推荐算法;第2节介绍实验设置和实验结果分析;第3节进行总结和展望未来工作。

1 基于元学习的响应式推荐系统

1.1 构建线下训练任务

我们将 (u, v) 定义为用户 u 和物品 v 之间的一个交互。交互的集合记为 $R = \{\dots, (u, v), \dots\}$ 。从元学习的角度,为了对应线上使用少量最新交互信息准确更新用户兴趣表征的需求,我们将线下的大量的历史交互信息重组为大量的如图2所示的响应式推荐任务。每个任务就是使用用户少量的 k 个连续交互信息去预测该用户紧接着的下一个交互物品。因为当前用户的 k 个交互数量太少,我们使用邻居的交互信息来丰富当前任务的数据量。我们将邻居定义为与当前用户的 k 个交互物品至少存在一个交互的其他用户。如图2所示的用户 i 和 j 为用户1的邻居。然后我们从邻居的交互信息中随机选取 $K-k$ 个交互信息补充到当前的训练任务中。然后一个线下训练任务 t 定义为用这 K 个交互信息(记为集合 R_t^K)去预测当前用户的下一个交互(记为集合 R_t^1)。从历史的交互信息中,我们可以构建大量的这样的任务,记为集合 $T = \{t_1, t_2, \dots, t_H\}$,其中 H 为线下任务数量。

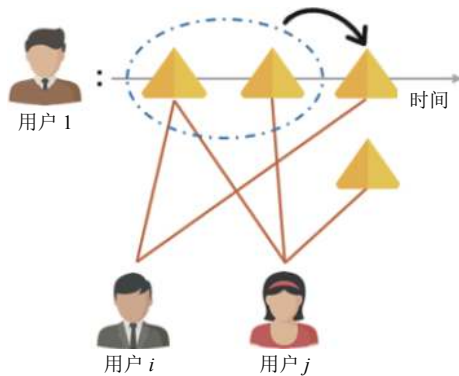


图2 一个带有邻居信息的线下训练任务

1.2 利用元学习提取先验知识

我们将推荐模型记为 f ,所有任务共享相同的推荐模型结构,然后在自己的任务上调整其参数以满足任务需求。借鉴于MAML^[11]的想法,我们将先验知识定义为模型 f 的初始参数,记为 θ 。然后对于任意一个任务 $t \in T$,响应式推荐的目的是使用交互信息 R_t^K 来更新模型,使得模型能够准确预测任务对应用户的下一个

交互 R_t^1 。模型更新过程我们就使用简单的一步或者多步梯度下降更新方法。以一步梯度下降更新为例,模型参数的更新规则如下:

$$\theta_t = \theta - \nabla_{\theta} L_t^K(\theta)$$

其中, $L_t^K(\cdot)$ 为模型在任务 t 的 R_t^K 交互集合上的损失函数, α 为学习率。然后我们对先验知识 θ 的要求是模型在这样的初始参数上经过少量样本的更新就能准确的预测对应用户的下一个交互信息,根据该要求我们设计如下优化目标来学习所有任务公用的初始参数 θ :

$$\min_{\theta} \sum_t L_t^1(\theta_t) = \min_{\theta} \sum_t L_t^1(\theta - \alpha \nabla_{\theta} L_t^K(\theta))$$

其中, $L_t^1(\cdot)$ 为模型在任务 t 的 R_t^1 交互集合上的损失函数。值得注意的是,在我们的方法中,我们在梯度更新计算 θ_t 的过程中会保存所有中间状态,所以 θ_t 可以视为关于参数 θ 的一个函数,然后最后的学习目标是对参数 θ 求导数,然后更新参数 θ 。直观地说,我们的优化目标是学习到初始参数 θ ,使其在线下训练的过程中,对于每个任务都可以经过少量交互的更新,然后准确预测对应用户的下一个交互物品。之后,我们可以将参数 θ 作为先验知识,对于线上的更新需求,其正好与我们线下训练的任务相似,所以在线上更新的过程中,使用该参数作为初始参数更新模型,也能够少量样本上准确地学习到用户的近期兴趣,以准确地预测用户的下一个交互。

在我们的场景中,我们使用矩阵分解模型^[3]来实现推荐模型 f ,模型的参数 $\theta = \{P, Q\}$,其中 P 为用户的特征矩阵, p_i 为矩阵 P 的第 i 列,表示用户 u_i 的特征; Q 为物品的特征矩阵, q_j 为 Q 的第 j 列,表示物品 v_j 的特征。根据矩阵分解模型,我们使用特征的内积表示用户 u_i 对物品 v_j 的喜好程度,记为 $r_{ij} = p_i^T q_j$ 。有了喜好程度的评分之后,我们可以根据评分为用户推荐喜欢的物品。

然后我们基于BPR (Bayesian Personalized Ranking)^[14]的方法来设计优化目标 L_t^1 。BPR的优化目标是优化排序,要求模型给出的评分将交互过的物品排在为交互过的物品之前。为了满足该要求,给定一个交互集合 R ,首先需要如下定义BPR中的训练数据集,记为 D_R 。

$$D_R = \left\{ (u_i, v_j) \mid (u_i, v_j) \in R \cap (u_i, v_k) \notin R \right\}$$

然后结合矩阵分解给出的评分,对于用户 u_i ,其交互过的物品 v_j 和为交互过的物品 v_k 之间的排序关系定

义为:

$$\hat{x}_{ijk} = r_{ij} - r_{ik} = p_i^T q_j - p_i^T q_k$$

因为我们需要将交互过的物品排在为交互过的物品之前,所以需要最大化 \hat{x}_{ijk} .加上正则化项之后,最终的BPR损失函数定义如下:

$$L_{ijk} = -\log \sigma(\hat{x}_{ijk}) + \lambda_P \|p_i\|^2 + \lambda_Q^+ \|q_j\|^2 + \lambda_Q^- \|q_k\|^2$$

其中, $\lambda_P, \lambda_Q^+, \lambda_Q^-$ 是正则化项的权重超参数.

有了BPR损失函数的定义之后,对于某个任务 $t \in T$,我们定义该任务的损失函数 L_t^T 如下:

$$L_t^T = \sum_{(u_i, v_j, v_k) \in D_{R_t^K}} L_{ijk}$$

其中, $D_{R_t^K}$ 是以交互集合 R_t^K 构建的BPR训练数据集.对于 L_t^T ,我们也使用BPR损失函数,与 L_t^T 不同的是 L_t^I 是在交互集合 R_t^I 上构建的损失函数.

值得注意的是在任务 t 上经过 L 次更新之后得到的参数,记为 θ_t^L .在我们的设计中, θ_t^L 相当于关于初始参数 θ 的一个 L 层的网络.这样当 L 增加时,模型在训练过程中容易发生梯度爆炸的现象^[15].为了解决这个问题,我们再加上浅层的BPR损失函数来限制参数的空间以保证模型的稳定性.最终的优化目标如下:

$$\min_{\theta} \sum_t L_t^K(\theta) + \lambda_{ML} L_t^I(\theta_t^L)$$

其中, λ_{ML} 是基于元学习的损失函数的权重超参数.

1.3 响应式推荐

学好了先验知识 θ 之后,我们可以利用该先验知识在线上的推荐任务中实现响应式推荐.对于一个用户 u ,我们首先利用其最近的 k 个交互构建响应式推荐任务中的交互集合 R_t^K ,然后我们以之前学习到的先验知识 θ 为基础,利用该任务上的损失函数 L_t^K 做 L 次参数更新得到模型参数 θ_t^L .在先验知识 θ 的帮助下, L 次参数更新得到的模型能够更好地捕捉到用户最近的兴趣.

1.4 线上增量更新

最后,我们的模型非常容扩展线上增量更新.给定一个新的交互 (u, v) ,如果用户 u 或者物品 v 是新用户或新物品,我们照如下公式初始化他们的特征:

$$p_{M+1} = \frac{1}{M} \sum_{u_m \in U} p_m + \epsilon_p$$

$$q_{N+1} = \frac{1}{N} \sum_{v_n \in V} q_n + \epsilon_q$$

其中, U 和 V 表示现有用户和物品的集合; M 和 N 是现有用户和物品的数目; ϵ_p 和 ϵ_q 是采样自标准多维高斯分布 $N(\mu, \sigma^2 I)$ 的小噪声.通过这种方法,新用户的表征可以利用到模型学习到的老用户的表征,使得将新用户加入后,模型也可以快速调整,学习出新用户的表征.

如果用户 u 不是新用户,则我们可以为该新交互构建一个任务,其中待预测交互集合 $R_t^I = \{(u, v)\}$,然后我们可以使用该用户之前最近的 k 个历史交互信息构建任务训练交互集 R_t^K .如果该用户的历史交互信息少于 k 个,我们就重复采样直至凑够 k 个历史交互.有了新任务 t 的交互集合 R_t^K 和 R_t^I 之后,我们就可以前面的优化目标更新先验知识 θ .

2 实验设置和结果分析

2.1 数据集介绍

我们使用了 MovieLens-1M 和 Netflix 这两个公开数据集做了验证实验.两个数据集都是电影评分数据集.其中, MovieLens-1M 数据集大约包含 6 千用户, 4 千部电影和 1 百万个评分,后面简称 MovieLens; Netflix 数据集大约包含 1.8 万用户, 45 万部电影和 1 亿个评分.评分的范围是从 1 到 5.所有的评分都有一个时间戳.在我们的实验中,类似于 BPR^[14] 的设置,我们将评分大于等于 3 的电影视为正样本,其它的都视为为交互过的样本.

2.2 对比模型介绍

我们选取如下流行的线上推荐模型作为对比:

(1) RKMF^[5]: 该方法使用核函数建模用户与物品特征向量间的关系.在迭代更新的过程中,该方法交替地更新用户特征或者物品特征.

(2) sRec^[9]: 该方法使用概率模型建模用户特征和物品特征的演变过程,并且使用现有特征来初始化新用户或者新物品特征.

(3) SPMF^[10]: 该方法在迭代增量的更新过程中,同时会使用缓存的之前的一部分老的交互信息.

2.3 评估方法介绍

我们首先随机选出 10% 的用户作为新用户,将他们的交互信息从数据集中移除.为了验证推荐模型的响应性,对于这些新用户,我们为模型提供这些用户的 n 个新的交互,然后检测模型是否能够准确预测这些用户的下一个交互.在实验中,设置 $n = 1, 5, 10$ 三种情况.

然后将剩下的交互信息按照时间顺序划分为90%和10%两部分. 对于所有的模型, 我们将90%的部分用于线下训练模型, 然后后面10%的部分用于线上迭代更新模型并检验模型对于老用户兴趣变化的响应性. 在这个过程中, 对于后面10%的交互信息, 按照时间顺序, 每当来一批交互信息的时候, 我们先验证模型是否能够准确预测这些交互信息, 然后再用这些新的交互信息更新模型.

根据之前的工作, 我们采取 leave-one-out 的评估模式^[16,17]. 我们的目标是预测用户的下一个交互的物品, 为了评估模型同时提升评估效率, 我们从为交互过的物品中随机采样100个物品作为负样本. 然后我们根据模型对这101个物品的评分计算如下3个评测指标:

- (1) HR@K(Hit Ratio): 评估正样本出现在排序前K个之中的比例.
- (2) AUC: ROC 曲线下的面积值. 简单来说, 该方法综合评估了正样本排在负样本前面的概率.
- (3) NDCG@K (Normalized Discounted Cumulative Gain): 该方法要求正样本排在前K个的同时还会根据排序位置进行惩罚. 排序越靠后, 惩罚越多.

总的来说, 这3个指标都是得分越高, 模型性能越好.

2.4 对老用户的响应性对比

图3展示了不同模型在更新更新过程中的性能变化曲线. 从图中可以看出, RKMF和sRec模型的性能在线上更新的过程中迅速下降. 因为这些模型在线上更新的过程中忽略了历史数据, 从而导致模型在少量新样本上容易过拟合, 出现模型不稳定现象. 与之不同, 我们的模型在自己相对独立的任务中, 基于共有的先验知识, 独立地优化自己的参数, 导致模型不会受到少量新交互信息的剧烈扰动, 从而表现出更高的稳定性. 同时, 由于使用的历史信息先验知识更利于模型在少量样本上快速收敛, 我们的模型在响应式推荐的任务中性能表现也更好.

2.5 对新用户的响应性对比

表1展示了不同模型在学习新用户兴趣任务上的表现, 其中n为模型当前观测到的新用户交互的数量. 从表中可以看出我们的模型在学习新用户兴趣上的显著优势. 举例来说, 在Netflix数据集上, 当n=1时, 我们的模型在NDCG@10的指标上超出了对比模型13%.

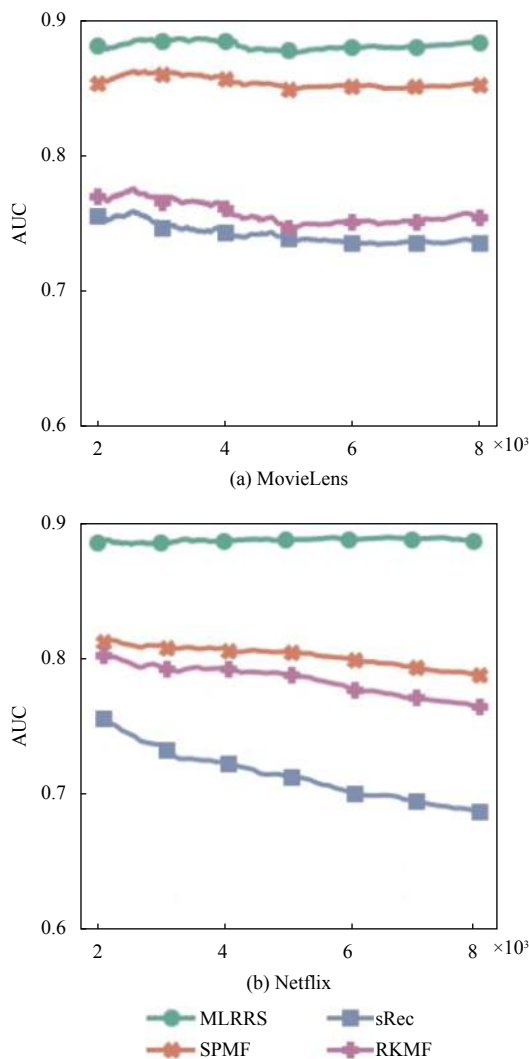


图3 不同模型对老用户响应性的AUC值对比

表1 不同模型对新用户响应性对比 (HR@10和NDCG@10简记为HR和NDCG)

n	模型	MovieLens		Netflix	
		HR	NDCG	HR	NDCG
1	RKMF	0.487	0.346	0.466	0.340
	sRec	0.530	0.392	0.498	0.365
	SPMF	0.543	0.444	0.541	0.368
	Ours	0.599	0.512	0.617	0.499
5	RKMF	0.493	0.350	0.456	0.335
	sRec	0.535	0.395	0.533	0.387
	SPMF	0.559	0.456	0.564	0.408
	Ours	0.666	0.590	0.631	0.523
10	RKMF	0.490	0.350	0.484	0.360
	sRec	0.533	0.399	0.565	0.420
	SPMF	0.571	0.467	0.609	0.445
	Ours	0.679	0.601	0.657	0.551

此外, 相对于HR@10评价指标, 我们的模型在NDCG@10评价指标上取得了更高的提升. 举例来说,

在 MovieLens 数据集上, 当 $n = 5$ 的时候, 我们的模型在 HR@10 指标上取得了 10.7% 的提升, 在 NDCG@10 指标上取得了 13.4% 的提升. 在 Netflix 的数据集上也有相同的现象. 由于 NDCG 指标会对排序位置进行惩罚, 这说明了我们的模型可以将正样本排得更靠前.

2.6 对元学习损失函数权重的实验分析

我们细化研究了元学习损失函数对我们模型性能的影响. 图 4 展示了实验结果, 从图中我们可以分

析得出如下结论:

(1) 使用适当的权重, 我们的元学习损失函数能够使得模型学习出更好的性能

(2) 随着元学习损失函数权重 λ_{ML} 的增加, 模型的性能先提高后降低. 同时值得注意的是, 当 λ_{ML} 超过一定值之后, 模型会发生梯度爆炸现象. 由前面的方法部分介绍可知, 这是由于元学习损失函数相当于建立在多层的神经网络之上, 容易过拟合, 甚至发生梯度爆炸现象.

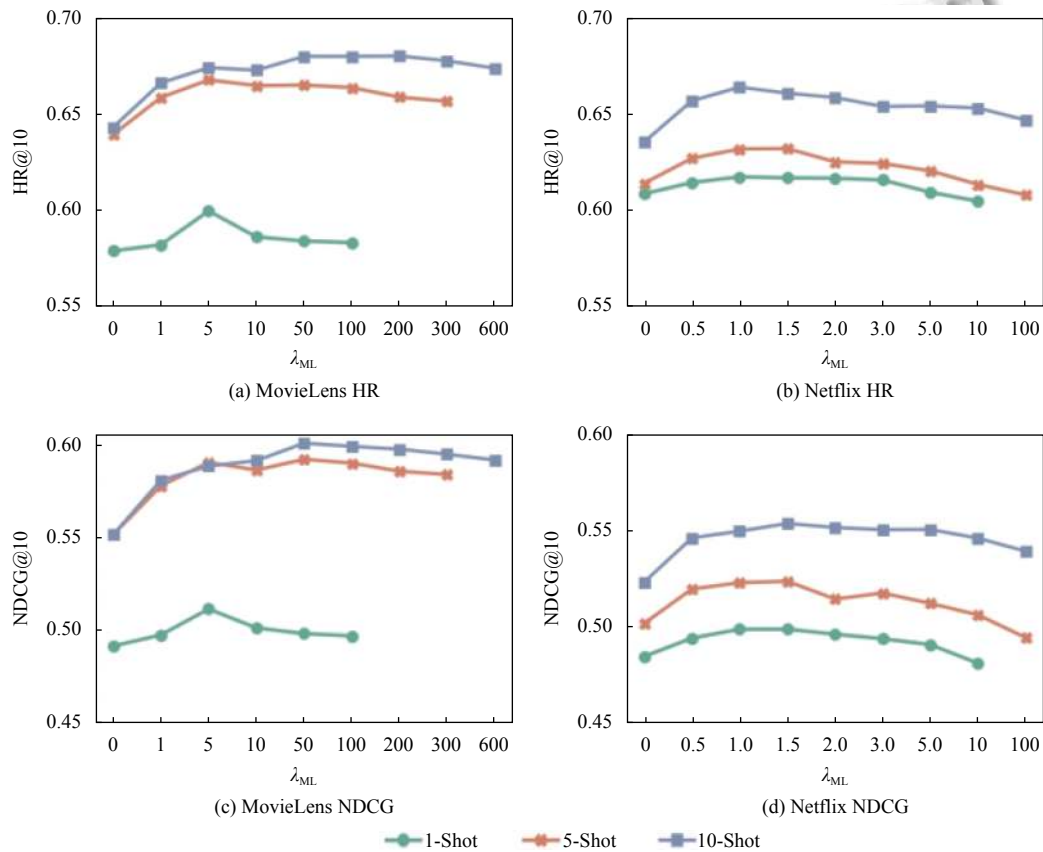


图 4 元学习损失函数权重 λ_{ML} 对模型性能的影响

3 结论和展望

在本文中, 我们提出了一套基于元学习的响应式在线推荐系统设计. 在线上更新的过程中, 由于新的交互信息数量少, 导致模型很难根据少量新的交互信息准确快速地学习出用户最近的兴趣. 在我们的方法中, 我们通过基于元学习的方案, 从大量的历史交互信息中提取出先验知识, 然后使用该先验知识指导模型的线上更新过程, 从而使得模型能够更加准确地学习出用户最近的兴趣. 我们通过大量的实验证明了我们方案的有效性.

在未来的工作中, 我考虑使用社交网络的信息来增强对用户兴趣特征的建模^[18-20]; 同时我们希望通过元学习的方法更加细致地建模分析用户的兴趣随时间变化的模式.

参考文献

- 1 Chen H. Modeling and detecting drift in user interest based on hierarchical classification. Proceedings of the 2019 IEEE 4th International Conference on Big Data Analytics. Suzhou, China. 2019. 246-251.
- 2 Wang WQ, Yin HZ, Huang Z, *et al.* Streaming ranking based

- recommender systems. Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. Ann Arbor, MI, USA. 2018. 525–534.
- 3 Chavosh A, Halimi A B, Soheilrad S, *et al.* Customer responsiveness and export performance of selected electronic equipment export companies in malaysia. Proceedings of 2011 International Conference on Social Science and Humanity-Section1. Singapore. 2011. 124–127.
 - 4 Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Las Vegas, NV, USA. 2008. 426–434.
 - 5 Salakhutdinov R, Mnih A. Probabilistic matrix factorization. Proceedings of the 20th International Conference on Neural Information Processing Systems. Vancouver, BC, Canada. 2007. 1257–1264.
 - 6 Rendle S, Schmidt-Thieme L. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. Proceedings of the 2008 ACM Conference on Recommender Systems. Lausanne, Swiss. 2008. 251–258.
 - 7 Diaz-Aviles E, Drumond L, Schmidt-Thieme L, *et al.* Real-time top-n recommendation in social streams. Proceedings of the 6th ACM Conference on Recommender Systems. Dublin, Ireland. 2012. 59–66.
 - 8 Song Q, Cheng J, Lu HQ. Incremental matrix factorization via feature space re-learning for recommender system. Proceedings of the 9th ACM Conference on Recommender Systems. Vienna, Austria. 2015. 277–280.
 - 9 He XN, Zhang HW, Kan MY, *et al.* Fast matrix factorization for online recommendation with implicit feedback. Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. Pisa, Italy. 2016. 549–558.
 - 10 Chang SY, Zhang Y, Tang JL, *et al.* Streaming recommender systems. Proceedings of the 26th International Conference on World Wide Web. Perth, France. 2017. 381–389.
 - 11 Finn C, Abbeel P, Le S. Model-agnostic meta-learning for fast adaptation of deep networks. Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia. 2017. 1126–1135.
 - 12 Vinyals O, Blundell C, Lillicrap T, *et al.* Matching networks for one shot learning. Proceedings of the 30th International Conference on Neural Information Processing Systems. Barcelona, Spain. 2016. 3637–3645.
 - 13 Yoon J, Kim T, Dia O, *et al.* Bayesian model-agnostic meta-learning. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montreal, QC, Canada. 2018. 7343–7353.
 - 14 Rendle S, Freudenthaler C, Gantner Z, *et al.* Bpr: Bayesian personalized ranking from implicit feedback. Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence. Montreal, QC, Canada. 2009. 452–461.
 - 15 Pascanu R, Mikolov T, Bengio Y. Understanding the exploding gradient problem. arXiv: 1211.5063.
 - 16 Huang J, Zhao WX, Dou HJ, *et al.* Improving sequential recommendation with knowledge-enhanced memory networks. Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. Ann Arbor, MI, USA. 2018. 505–514.
 - 17 Cao D, He XN, Miao LH, *et al.* Attentive group recommendation. Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. Ann Arbor, MI, USA. 2018. 645–654.
 - 18 Li XQ, Yan XM. Analyzing of personalized recommendation model of social network users based on big data. Proceedings of the 2019 6th International Conference on Dependable Systems and Their Applications. Harbin, China. 2020. 135–142.
 - 19 Guo YN, Cao JX, Lin WG. Social network influence analysis. Proceedings of the 2019 6th International Conference on Dependable Systems and Their Applications. 2020. Harbin, China. 517–518.
 - 20 Firdaus MF, Baizal ZKA, Bratawisnu MK, *et al.* Pattern recognition of dynamic social network. Proceedings of the 2020 3rd International Conference on Intelligent Autonomous Systems. Singapore. 2020. 56–61.