

看了很多负采样的论文，最后我选择不采样了。

原创 一元 炼丹笔记 1月4日

收录于话题

#必读论文系列 9 #搜索推荐前沿算法 59

↑↑↑关注后"星标"炼丹笔记

炼丹笔记干货

作者：一元，四品炼丹师

Efficient Heterogeneous Collaborative Filtering without Negative Sampling for Recommendation(AAAI20)

背景

最近推荐相关的研究主要集中在探索神经网络的结构等，然后采用负采样对模型进行高效的学习。然而，这么做会导致有两方面的问题没有被考虑仔细：

- 负采样会带来较大的波动；基于采样的方法很难获得最优的排序结果；
- 尽管heterogeneous的反馈在许多在线系统中被广泛使用，(例如view,click,purchase),大多数现有的方案只会使用用户的一种反馈，例如购买等；

本文提出了一个新的**非采样的迁移学习解决方案**，我们称其为 Efficient Heterogeneous Collaborative Filtering (EHCF) 的Top-N推荐。它不仅可以对细粒度的用户项关系进行建模，而且能够以**较低的时间复杂度**从整个异构数据（包括所有未标记数据）中高效地学习模型参数。

在三个实际数据集上的大量实验表明，EHCF在传统（单一行为）和异构场景中都显著优于最先进的推荐方法。此外，EHCF在训练效率上有显著提高，使其更适用于实际的大型系统。

在之前的许多工作当中，有两个重要的问题还没有充分考虑，

1. 很多神经网络为了提升模型的训练效率，采用负采样，但是这么做会出现模型训练不稳定，对于采样的分布以及采样的比例较为敏感。
2. 之前的工作缺乏用户行为之间的关系的深度探索，每个行为都会有自己对应的上下文以及非常强的迁移关系。

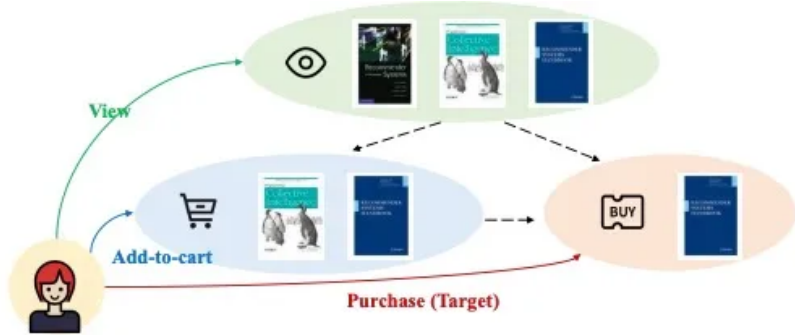


Figure 1: An example of multiple types of user feedback (not limited to these behaviors). Generally there are strong transfer relations among different behaviors (dotted lines).

Efficient Heterogeneous Collaborative Filtering

问题定义

假设我们有用户 U ,商品 V ,我们使用 u 来表示用户, v 表示一个商品, $\{R_{(1)}, R_{(2)}, \dots, R_{(K)}\}$ 表示用户和商品的对应 K 种不同行为的交互矩阵, 其中 $R_{(k)} = [R_{(k)uv}]_{|U| \times |V|} \in \{0, 1\}$ 表示用户 u 是否和商品 v 在某个行为 k 下有交互。

广义上, heterogeneous collaborative filtering有一个目标需要优化, 我们将其表示为 $R_{(k)}$, 而Heterogeneous CF就是为了估计 $\bar{R}_{(K)uv}$ 的概率, 商品会以分数 $\bar{R}_{(K)uv}$ 下降的顺序进行排序, 从而提供top—N的商品推荐列表。

模型一览

模型整体框架如下:

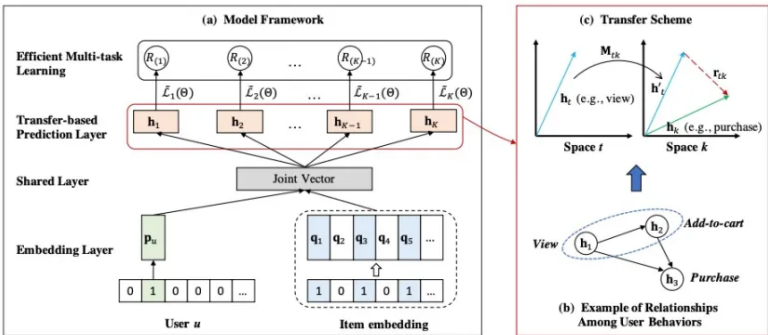


Figure 2: An overview of EHCF model. (a) Illustration of the model framework. (b) An example of the relationships among behaviors, where h_1, h_2 , and h_3 denotes the prediction functions of behaviors: view, add-to-cart, and purchase, respectively (note that EHCF is not limited to these examples). (c) Illustration of the transfer scheme of two relational behaviors.

用户和商品会先被投影为一个dense向量表示, user-item的对 (u, v) , 我们使用用户以及他对应的商品交互作为输入, 此外, 这边我们不进行采样, 使用一种高效的优化方法。

对于每一个user-item的实例 (u, v) ,映射函数可以定义为:

$$\phi(p_u, p_v) = p_u \odot q_v$$

其中 $p_u \in R^d, q_v \in R^d$ 是用户 u 和商品 v 的潜藏向量, d 表示embedding的大小。 \odot 表示element-wise相乘。

$$\bar{R}_{(k)uv} = h_k^T(p_u \odot q_v) = \sum_{i=1}^d h_{k,i} p_{u,i} q_{v,i}$$

1. Transfer-based Multi-behavior Prediction

不同的行为之间是存在转化关系的, 两个行为之间的关系可以通过下面的方式来定义:

$$f_{h_t \rightarrow h_k} = h_t M_{tk} + r_{tk}$$

其中, $M_{tk} \in R^{d \times d}$ 是一个迁移矩阵, 它将 h_t 从地 t 个行为空间迁移到来第 k 个行为空间。 $r_{tk} \in R^d$ 是一个bias向量, 所以第 k 个行为的预测层可以通过下面的形式计算:

$$h_k = \sum_t (f_{h_t \rightarrow h_k}) = \sum_t (h_t M_{tk} + r_{tk})$$

其中 t 表示第 k 个行为之前的行为, 初始的行为是随机初始化的(在推荐里面就是view)

2. Efficient Optimization without Sampling

此处我们使用加权的regression loss

$$L_{\Theta} = \sum_{u \in B} \sum_{v \in V} c_{uv}^k (R_{(k)uv} - \bar{R}_{(k)uv})^2 = \sum_{u \in B} \sum_{v \in V} c_{uv}^k (R_{(k)uv}^2 - 2R_{(k)uv} \bar{R}_{(k)uv} + \bar{R}_{(k)uv}^2)$$

其中 c_{uv}^k 表示 $R_{(k)uv}$ 的权重。计算该loss的时间复杂度为 $O(|B||V|d)$, 这在实践中是可以接受的。

在只有正样本的数据中, non-observed的实例通常会被设定为一个标签 $R_{(k)uv} = 0$, 未观测到的 $R_{(k)uv}$ 可以被简化该等式。

$$\bar{L}_{\Theta} = -2 \sum_{u \in B} \sum_{v \in V^{k+}} c_{uv}^{k+} (R_{(k)uv} \bar{R}_{(k)uv}) + \sum_{u \in B} \sum_{v \in V^k} c_{uv}^k \bar{R}_{(k)uv}^2 = \overbrace{\sum_{u \in B} \sum_{v \in V^{k+}} ((c_{uv}^{K+} - c_{uv}^{K-}) \bar{R}_{(k)uv}^2 - 2c_{uv}^{k+} R_{(k)uv} \bar{R}_{(k)uv})}^{L_k^P(\Theta)}$$

- 关于 Θ 的常数值已经被省略了。

其中 $\bar{L}_k^P(\Theta)$ 表示正样本的损失, 其中 $\bar{L}_k^A(\Theta)$ 表示对所有样本的损失, $L_k^A(\Theta)$ 表示对所有正样本的和所有样本的损失。

基于内积乘法操作的decouple的操作,

$$\bar{R}_{(k)uv}^2 = \sum_{i=1}^d h_{k,i} p_{u,i} q_{v,i} \sum_{j=1}^d h_{k,j} p_{u,j} q_{v,j} = \sum_{i=1}^d \sum_{j=1}^d (h_{k,i} h_{k,j}) (p_{u,i} p_{u,j}) (q_{v,i} q_{v,j})$$

我们将上式代入得到:

- 因为 c_{uv}^{k-} 是均匀的; 参数 c_v^{k-} 是商品相关的;
- $p_{u,i}$ 和 $q_{v,i}$ 可以被很好地分开, $\sum_{v \in V} c_v^{k-} q_{v,i} q_{v,j}$ 的优化以及 $\sum_{u \in B} p_{u,i} p_{u,j}$ 是相互独立的, 我们可以通过预先计算这两项来加速。

最终对于单个第 k 个行为的整个数据集的loss就是:

$$\bar{L}_k(\Theta) = L_k^P(\Theta) + \sum_{i=1}^d \sum_{j=1}^d (h_{k,i} h_{k,j}) \left(\sum_{u \in B} p_{u,i} p_{u,j} \right) \left(\sum_{v \in V} c_v^{k-} q_{v,i} q_{v,j} \right)$$

所以关于 $L_k^A(\Theta)$ 的复杂度从 $O(|B||V|d)$ 降为 $O((|B| + |V|)d^2)$ 。

3. Multi-Task Learning

$$L(\Theta) = \sum_{k=1}^K \lambda_k \bar{L}_k(\Theta)$$

其中 K 为用户行为的个数, λ_k 用来控制第 k 个需变更为 u 的行为的影响, 此处我们令 $\sum_{k=1}^K \lambda_k = 1$ 。

讨论

在第 k 个行为的batch更新的时间复杂度为: $O((|B| + |V|)d^2 + |R_B^k|d)$, 其中 R_B^k 表示该batch中用户和商品在 k 的行为下的正向交互次数。在MTL更新 K 类行为时, 我一个batch需要耗费 $O(K(|B| + |V|)d^2 + \sum_{k=1}^K |R_B^k|d)$ 。

对于原始的回归损失, 会需要耗费 $O(K|B||V|d)$ 次, 在实践中, 因为 $|R_B^k| \ll |B||V|$ 以及 $d \ll |B|$, 本文的模型复杂度得到了大大的降低。

实验

1. 效果比较

Table 2: Performance of different models on three datasets. * and ** denotes the statistical significance for $p < 0.05$ and $p < 0.01$, respectively, compared to the best baseline.

<i>Movielens-1M</i>		HR@10	HR@50	HR@100	HR@200	NDCG@10	NDCG@50	NDCG@100	NDCG@200
Single	BPR	0.0822	0.2637	0.4048	0.5710	0.0418	0.0757	0.0986	0.1217
	ExpoMF	0.0892	0.2808	0.4253	0.5864	0.0457	0.0764	0.1053	0.1309
	NCF	0.0936	0.2902	0.4316	0.6023	0.0481	0.0837	0.1097	0.1324
	EHCF-Sin	0.1025**	0.3104**	0.4597**	0.6177**	0.0508**	0.0934**	0.1169**	0.1392**
<i>Beibei</i>		HR@10	HR@50	HR@100	HR@200	NDCG@10	NDCG@50	NDCG@100	NDCG@200
Single	BPR	0.0437	0.1246	0.2192	0.3057	0.0213	0.0407	0.0539	0.0689
	ExpoMF	0.0452	0.1465	0.2246	0.3282	0.0227	0.0426	0.0553	0.0723
	NCF	0.0441	0.1562	0.2343	0.3583	0.0225	0.0445	0.0584	0.0757
	EHCF-Sin	0.0464**	0.1637**	0.2586**	0.3743**	0.0247**	0.0484**	0.0639**	0.0799**
Heterogeneous	CMF	0.0482	0.1582	0.2843	0.4288	0.0251	0.0462	0.0661	0.0852
	MC-BPR	0.0504	0.1743	0.2755	0.3862	0.0254	0.0503	0.0653	0.0796
	NMTR	0.0524	0.2047	0.3189	0.4735	0.0285	0.0609	0.0764	0.0968
	EHCF	0.0608**	0.3316**	0.4312**	0.5460**	0.0325**	0.1213**	0.1374**	0.1535**
<i>Taobao</i>		HR@10	HR@50	HR@100	HR@200	NDCG@10	NDCG@50	NDCG@100	NDCG@200
Single	BPR	0.0376	0.0708	0.0871	0.1035	0.0227	0.0269	0.0305	0.0329
	ExpoMF	0.0386	0.0713	0.0911	0.1068	0.0238	0.0270	0.0302	0.0334
	NCF	0.0391	0.0728	0.0897	0.1072	0.0233	0.0281	0.0321	0.0345
	EHCF-Sin	0.0398*	0.0743**	0.0936**	0.1141**	0.0244*	0.0298**	0.0339**	0.0372**
Heterogeneous	CMF	0.0483	0.0774	0.1185	0.1563	0.0252	0.0293	0.0357	0.0379
	MC-BPR	0.0547	0.0791	0.1264	0.1597	0.0263	0.0297	0.0361	0.0397
	NMTR	0.0585	0.0942	0.1368	0.1868	0.0278	0.0334	0.0394	0.0537
	EHCF	0.0717**	0.1618**	0.2211**	0.2921**	0.0403**	0.0594**	0.0690**	0.0789**

- 使用异质反馈的方法通常优于仅利用购买行为的方法，这表明了用户异质反馈的互补性；
- 使用全数据学习策略的方法通常比基于样本的方法表现更好；

2. 解耦实验

Table 3: Performance of variants of EHCF on Taobao dataset.								
Taobao		HR@10	HR@50	HR@100	HR@200	NDCG@10	NDCG@50	NDCG@100
Data Ablation	Purchase	0.0398	0.0743	0.0936	0.1141	0.0244	0.0298	0.0339
	Purchase&Add-to-cart	0.0612	0.1187	0.1569	0.2008	0.0297	0.0465	0.0527
	Purchase&View	0.0673	0.1438	0.2061	0.2824	0.0384	0.0493	0.0593
Model Ablation	Without Transferring	0.0633	0.1112	0.1525	0.2088	0.0312	0.0389	0.0457
	Without MTL	0.0611	0.1171	0.1552	0.2003	0.0304	0.0448	0.0509
Full Model	EHCF	0.0717**	0.1618**	0.2211**	0.2921**	0.0403**	0.0594**	0.0690**

- 将View数据和购物车数据添加到我们的模型中都会带来改进，从而验证了辅助行为对用户偏好建模的有效性。
- 我们的方法的显著改进也表明了应用非抽样策略从异质反馈中学习的必要性。此外，没有迁移和没有MTL的变体的性能都比完整的EHCF模型差，这验证了所提出的基于迁移的预测层和多任务训练组件的有效性；

3. 效率分析

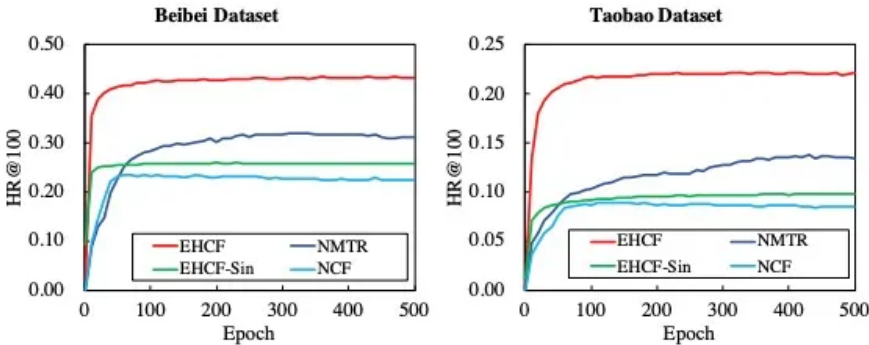


Figure 3: Performance curves of NCF (best single-behavior baseline), NMTR (best heterogeneous baseline), EHCF-Sin, and EHCF.

Table 4: Comparisons of runtime (second/minute [s/m]). “S”, “I”, and “T” represent the training time for a single iteration, the number of iterations to converge, and the total training time, respectively.

Model	Movielens-1M			Beibei			Taobao		
	S	I	T	S	I	T	S	I	T
NCF	91s	100	152m	62s	100	104m	115s	100	192m
EHCF-Sin	4.5s	100	8m	3.2s	100	6m	6s	100	10m
NMTR	—	—	—	165s	200	550m	180s	200	600m
EHCF-Original	—	—	—	62s	200	207m	192s	200	640m
EHCF	—	—	—	7s	200	24m	16s	200	54m

- EHCF比NCF和NMTR在单个行为数据以及heterogeneous数据上的效果都好；

4. 解决数据稀疏问题

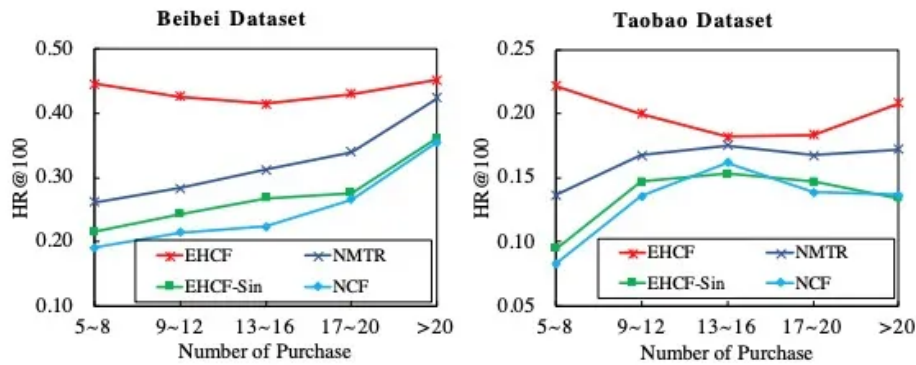


Figure 4: Performances of NCF, NMTR, EHCf-Sin, and EHCf on users with different number of purchase records.

- 使用其它的信息可以缓解数据标签稀疏的问题（例如购买等）；

5. 参数的影响

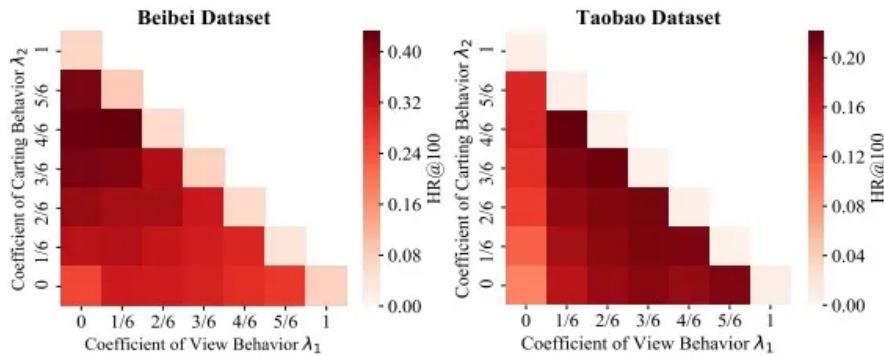


Figure 5: Performance of EHCf with different loss coefficient.

- 不同的数据集对应的参数λ是不一样的。

小结



本文提出了一种新的端到端推荐模型EHCf。所提出的EHCf具有两个关键特性：

- 采用新设计的优化方法进行高效的基于全数据的模型学习；
- 通过传递方式关联每个行为的预测，捕捉不同行为之间的复杂关系。

在三个真实数据集上的大量实验表明，EHCf不仅比现有的推荐模型有很大的提高，而且训练过程也相当快。这项工作补充了主流的基于样本的神经网络推荐模型和隐式反馈，为神经推荐模型的研究开辟了一条新的途径。所设计的高效的基于整体数据的策略有可能使许多人受益只观察到正面数据的任务。未来的工作包

括在网络嵌入和多标签分类等其他相关任务中探索我们的EHCF模型。我们还将尝试扩展我们的优化方法，使它适用于学习非线性模型。

参考文献

1. Efficient Heterogeneous Collaborative Filtering without Negative Sampling for Recommendation:
<http://www.thuir.cn/group/~mzhang/publications/AAAI2020-Chenchong.pdf>
2. <https://github.com/chenchongthu/EHCF/blob/master/code/EHCF.py>

来一波三连关注，通往神奇的炼丹学习之旅



收录于话题 #搜索推荐前沿算法·59个

上一篇

小学生才做选择题，“塔”们，我全要！多兴趣召回！

下一篇

2020年度「炼丹笔记」干货集锦