



 所谓 (/users/845813) 2020-10-26 09:23:20 61 阅读 0 收藏 (/articles/184912/mark/)

知识体系: 机器学习/深度学习 (/articles/?kid=941) 数据挖掘 (/articles/?kid=944) 搜索推荐 (/articles/?kid=945) 人工智能 (/articles/?kid=951) 数据处理 (/articles/?kid=955) 数据建模 (/articles/?kid=957)  修改知识体系

文章标签: 算法 (/search?q=算法&type=ARTICLE)  修改标签  标签历史 (/articles/184912/tags/history)

附加属性: 作者原创

# 多任务学习12:Why I like it\_Multi-task Learning for Recommendation and Explanation(RecSys2018)

- 1 Why I like it: Multi-task Learning for Recommendation and Explanation(RecSys18)
  - 1.1 简介
  - 1.2 模型
    - 1.2.1 Adversarial Sequence-to-Sequence Learning
      - 1.2.1.1 Recurrent Review Generator 递归评论生成器
      - 1.2.1.2 卷积评论判别器
      - 1.2.1.3 reinforce的评论生成对抗训练
    - 1.2.2 生成个性化的解释
    - 1.2.3 评分预测的上下文-aware的矩阵分解
    - 1.2.4 算法优化
  - 1.3 实验
    - 1.3.1 模型效果
    - 1.3.2 解释质量
    - 1.3.3 判别器效果
  - 1.4 小结
- 2 参考文献

## Why I like it: Multi-task Learning for Recommendation and Explanation(RecSys18)

### 简介

这是比较早的一篇使用多任务学习来赋能推荐任务可解释性的文章,可解释性和有效性是构建推荐系统的两大关键成份,本文提出了一种新的多任务推荐模型, 通过结合矩阵分解和对抗性序列对序列学习来进行评分预测和推荐解释。改进了评级预测性能, 同时产生了有效的个性化解释。

近期很多工作会尝试利用用户的反馈来赋予模型可解释性。

### 模型

模型的整体框架如下:

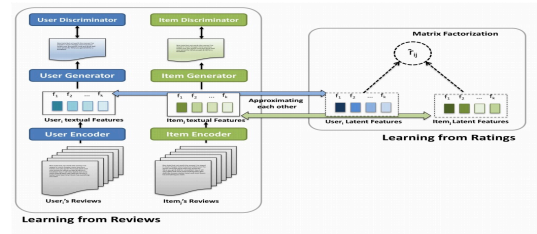


Figure 1: System architecture of multi-task learning model.

## Adversarial Sequence-to-Sequence Learning

此处我们提出了encoder-decoder的框架,他可以基于评论的文档生成相关的评论。

- *user review document*: 用户*i*的所有评论,我们称其为*user review document*,  $d_{u,i}$
- *item review document*: 关于商品*j*写的所有评论我们称之为*item review document*,  $d_{v,j}$
- $d_{u,i} \cap d_{v,j} \neq \emptyset$ : 当且仅当用户*i*关于商品*j*写过评论。

优化seq2seq模型的传统方法是在训练过程中的每个时间步输入真实性注释。这种优化策略会导致暴露偏差(exposure bias)的问题,即在测试时,模型会暴露在自己的预测中,因此误差会随着时间累积。



为了解决这个问题,我们采用了一种对抗性的训练过程,在这个过程中我们同时训练一个生成器网络和一个判别器网络。前者学会了创造虚假评论来愚弄判别者,而后者则旨在检测这些对抗性样本。

### Recurrent Review Generator 递归评论生成器

给定用户的评论文档 $d_{u,i}$ ,我们希望用户评论生成器能生成类似于用户*i*写的文档,此处,我们使用seq2seq的学习框架:

- 用户文档编码器;
- 用户评论生成器;

对于用户*i*,用户文档编码器先将评论文档 $d_{u,i}$ 映射到 $D$ 维的向量 $\bar{U}_i$ ,然后基于 $\bar{U}_i$ 生成和评论文档类似的评论。此处用户文档编码器和生成器都是基于GRU的。

细节:

- 对每个词进行embedding,然后过BiGRU,得到前向和后向的向量 $h_T = [\vec{h}_T, \overleftarrow{h}_T]$ ;
- 我们对用户*i*所有生成的文本向量进行求均值,得到 $\bar{U}_i$ ;
- 然后使用decoder学习特定用户的评论,通过估计下面的条件概率:

$$P(y_{i,1}, \dots, y_{i,T'} | \bar{U}_i) = \prod_{t=1}^{T'} p(y_{i,t} | \bar{U}_i, y_{i,1}, \dots, y_{i,t-1})$$

其中 $(y_{i,1}, \dots, y_{i,T'})$ 是预测长度为 $T'$ 的评论。

### 卷积评论判别器

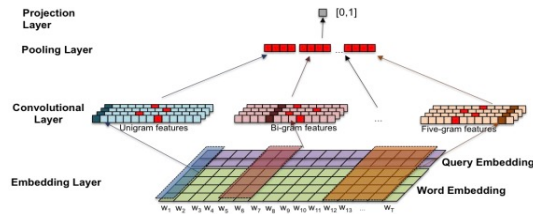


Figure 2: Convolutional Review Discriminator.

用户的评论生成器是希望生成和用户相近的评论,而评论判别器则希望能将生成的样本和真实的评论分开。本文采用卷积的形式来进行此操作,具体的框架如上图所示。

### reinforce的评论生成对抗训练

我们假设生成器 $G_\theta$ 是一个虚拟的agent,它的目的是在每次尝试中尽可能获得多的reward(由判别器的置信度给出)。也就是说生成器的目的是尽可能的制造能骗过判别器的生成样本。

此处我们判别器的训练目标为(最大化真实样本,最小化假的生成样本):

$$\max_{\phi} E_{Y \sim P_{data}} [\log D_{\phi}(Y)] + E_{Y' \sim G_{\theta}} [\log(1 - D_{\phi}(Y'))]$$

其中 $Y \sim P_{data}$ 表示从真实评论中采样的样本, $Y' \sim G_{\theta}$ 表示生成器生成的样本。

因为序列生成是一个离散化采样的过程,所以梯度不能直接从判别器传到生成器。我们此处使用policy gradient算法来估计目标函数 $J(\theta)$ 的梯度。

$$\begin{aligned} \nabla J(\theta) &= E_{Y' \sim G_{\theta}} \nabla_{\theta} \log G_{\theta}(Y') \cdot [R(Y') - b(Y')] \\ &\approx \frac{1}{K} \sum_{i=1}^K \nabla_{\theta} \log G_{\theta}(Y'_i) \cdot [R(Y'_i) - b(Y'_i)] \end{aligned}$$

其中 $R(Y'_i)$ 是有判别器给第*i*个生成序列 $Y'_i$ 的reward,  $b(Y'_i)$ 表示降低方差的baseline。

为了方便后续多任务学习,我们对文本特征学习得到的用户系数矩阵和矩阵分解得到的矩阵进行正则,这样我们的设定可以使我们的seq2seq模型更好地探索从协同过滤方法中得到的用户偏好。

- $L(\theta) = -J(\theta) + \lambda ||U - \bar{U}||_F$

其中 $U$ 是通过矩阵分解得到的用户稀疏矩阵,  $\bar{U}$ 是从用户文档编码器中学到的用户文本特征。

生成个性化的解释

我们先抽取用户和商品的文本特征 $\bar{U}_i$ 和 $\bar{V}_j$ , 我们使用单个解码器生成当前用户商品对的评论, 在每一步, 我们先将 $\bar{U}_i$ 和 $\bar{V}_j$ 拼接, 然后再输入到decoder中。为了进行对抗性训练, 我们还训练了一个评论判别器和评论生成器。评论判别器的目的是判断评论是不是用户 $i$ 是否给商品 $j$ 上编写的。



评分预测的上下文-aware的矩阵分解

我们有一个用户商品的评分矩阵 $R \in R^{N \times M}$ ,  $N$ 和 $M$ 分别是用户和商品的个数。我们假设所有的评分带有噪音(均值为0, 方差为 $\sigma^2$ 的高斯噪音), 所以我们的评分为:

- $p(R|U, V, \sigma^2) = \prod_{i=1}^N \prod_{j=1}^M [N(R_{ij}|u_i^T \cdot v_j, \sigma^2)]^{I_{ij}}$

如果用户 $i$ 对商品 $j$ 进行评分, 那么 $I_{ij}$ 为1, 否则为0.

为了解决稀疏性的问题, 此处我们假设评论文档中抽取的文本特征可以用作隐特征的信息性指标, 所以我们有:

- $p(U|\bar{U}, \sigma_U^2) = \prod_i^N N(U_i|\bar{U}_i, \sigma_U^2 I)$
- $p(V|\bar{V}, \sigma_V^2) = \prod_i^N N(V_i|\bar{V}_i, \sigma_V^2 I)$

通过在潜在变量的先验分布中引入文本特征, 上下文感知矩阵分解模型可以有效地利用辅助信息, 从而解决传统协同过滤方法的稀疏性问题。

算法优化

此处我们采用EM算法对模型进行优化。

Algorithm 1: Optimization Algorithm

```
for epoch ← 1 to T do
  for i ← 1 to N do
    Update  $U_i$  with least square approximation:
       $U_i \leftarrow (V I_i V^T + \lambda_U I_K)^{-1} (V R_i + \lambda_U \bar{U}_i)$ 
  end
  for j ← 1 to M do
    Update  $V_j$  with least square approximation:
       $V_j \leftarrow (U I_j U^T + \lambda_V I_K)^{-1} (U R_j + \lambda_V \bar{V}_j)$ 
  end
  for iteration ← 1 to T - step do
    Update  $\theta_U$  and  $\theta_V$  via teacher forcing.
  end
  for iteration ← 1 to G - step do
    Update  $\theta_U$  and  $\theta_V$  via policy gradient.
  end
  for iteration ← 1 to D - step do
    Update  $\phi_U$  and  $\phi_V$  via gradient ascent.
  end
end
```

实验

模型效果

Model	Y13	Y14	AE	AV	AG
PMF	0.985	1.053	1.311	1.297	1.251
HFT	0.977	1.029	1.159	1.152	1.121
CTR	0.975	1.013	1.184	1.147	1.139
JMARS	0.970	0.998	1.128	1.133	1.114
ConvMF+	0.917	0.954	1.117	1.092	1.084
MT - encoder	0.874	0.922	1.071	1.065	1.037
MT - decoder	0.871	0.926	1.074	1.069	1.038
MT	<b>0.861</b>	<b>0.896</b>	<b>1.018</b>	<b>1.026</b>	<b>1.004</b>

Table 2: Recommendation performance in terms of MSE

- MT模型比所有的baseline模型的效果都要很好多;

解释质量

Model	Y13	Y14	AE	AV	AG
N-gram	89.3	82.1	68.5	73.7	79.6
Skip-gram	85.2	79.8	65.2	67.2	74.2
LSTM	79.7	71.5	64.3	65.7	70.8
GCN-U	73.3	63.6	57.7	64.1	65.3
GCN-I	69.3	61.3	54.8	59.8	63.2
GCN-P	67.9	60.7	52.9	57.6	62.1
MT-U	66.1	62.5	53.6	59.2	60.8
MT-I	65.6	60.2	52.8	54.1	58.3
MT-P	<b>65.2</b>	<b>59.4</b>	<b>51.4</b>	<b>53.7</b>	<b>57.6</b>

Table 3: Explanation performance in terms of perplexity

Model	Y13	Y14	AE	AV	AG
N-gram	0.007	0.009	0.005	0.009	0.011
Skip-gram	0.009	0.014	0.007	0.011	0.015
LSTM	0.019	0.022	0.014	0.017	0.019
Opinosis	0.029	0.031	0.029	0.025	0.027
GCN-U	0.042	0.040	0.039	0.041	0.038
GCN-I	0.048	0.044	0.045	0.045	0.046
GCN-P	0.051	0.047	0.045	0.047	0.046
MT-U	0.048	0.043	0.042	0.044	0.047
MT-I	0.051	0.045	0.046	0.049	0.049
MT-P	0.053	0.052	0.049	0.052	0.051

Table 4: Explanation performance in terms of tf-idf

- 个性化的MT-P方法倾向于在所有测试数据集中执行所有MT方法中的最佳性能，并且在每种情况下，MT结果始终优于所有基线；



判别器效果

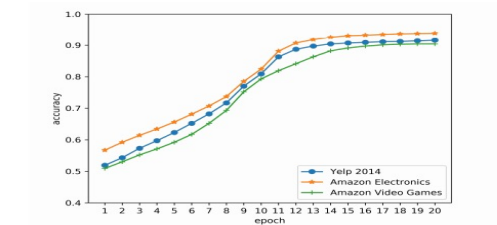


Figure 3: Learning curve of the discriminator network

- 判别器逐渐学习有助于区分敌方样本和真实样本的关键因素。
- 在Amazon Electronics数据集上,用户判别器可以拿到79.4%的准确性,商品判别器可以拿到85.8...%的准确性, 这说明判别器确实了解了用户和物品的重要特征。

小结

本文提出了一种新的多任务学习框架，它可以同时学习进行评分预测和推荐解释。通过两个任务的联合训练还可以提升效果。与当前最新技术相比，所有测试数据集的评测效果都有所提高。与使用更传统的方法相比，所产生的解释性也更接近于实际情况。

参考文献

1. Why I like it: Multi-task Learning for Recommendation and Explanation:<https://dl.acm.org/doi/pdf/10.1145/3240323.3240365>  
(<https://dl.acm.org/doi/pdf/10.1145/3240323.3240365>)

2. <https://www.youtube.com/watch?v=2Oz43b3GaP8> (<https://www.youtube.com/watch?v=2Oz43b3GaP8>)

评论文章 (0)

 1 (/articles/184912/voteup)



他们赞过该文章

所谓 (/users/845813)

上一篇：多任务学习11:Jointly Learning Explainable Rule...

下一篇：多任务学习13: 多任务学习(MTL)在搜索推荐场景...

写下你的评论...



评论