TransNets: Learning to Transform for Recommendation

本文是 DeepCoNN 模型的改进，RecSys'2017，论文链接：https://arxiv.org/abs/1704.02298

本文摘要

深度学习及相关模型在推荐系统任务中，比传统方法都要更优秀。目前推荐系统中，会融合很多的外部信息，如用户评论等，而 NLP 相关技术的发展更是受到了神经网络的大增益，DeepCoNN 模型借助神经网络从评论文本中获得用户、物品的表示。但是 DeepCoNN 模型在预测用户给目标商品的打分时，更多的是利用了用户的评论信息，本质上是一个根据评论文本预测打分的“情感分类”的问题，而不是推荐问题。本文提出的 TransNets 模型，扩展了 DeepCoNN，引入了一个额外的隐藏层来表示用户、物品，使之与训练集的评论文本类似，从而获得更为出色的推荐性能。

本文介绍

在推荐系统中使用用户的评论信息，可以大幅提升推荐系统的性能，因为相比传统的协同方法，除了历史打分记录，评论文本可以作为很好的辅助和补充信息。因此，现今有很多的模型开始关注用户和物品的“上下文”信息，从而构建用户和物品的表示。对用户来说，这些信息包括人口学、社会经济学等宽泛意义上的概念，可以从侧面反映出用户的偏好；对于商品来说，它们的价格、外形、实用性、质量、服务等属性，也体现出这件商品区别与其他的特征。在协同过滤模型中，这些额外的信息被综合利用，影响最终的预测结果。

评论信息，与一般的内容不同，它不是单单属于用户或者商品的，而是体现出两者之间交互的信息。最近的模型 DeepCoNN 使用了神经网络从评论中生成一个用户或者是物品的表示，再通过一个回归模型来预测用户对预测商品的打分。仔细考虑 DeepCoNN 模型及其实现方式，在训练时，预测是有一个目标用户、目标商品、对应的评论这些输入，而在测试时，是不含有评论作为输入的。因此本文设计了一种在训练时使用评论数据，测试时生成一个近似替代，用来计算最终的得分。这个近似的替代生成过程，在训练中加入正则处理，使它能够与训练集中用户对物品的真是评论数据尽可能相似。

模型方法

CNN 文本处理

与 DeepCoNN 类似，通过经典的 CNN 网络，将评论文本抽象成一个 n 维向量表示，即 $\Gamma : [w_1, w_2, \cdots, x_T] \rightarrow \mathbb{R}^n$。

DeepCoNN 模型

上一篇笔记 已经详细介绍了，不再赘述。重点指出 DeepCoNN 的局限：DeepCoNN 模型能够比其他的方法表现更好的情况下，是因为在预测的时候，DeepCoNN 使用了待测目标用户对目标商品写下的评论信息，这就将推荐问题转化为一个情感分析的问题了，这显然是不符合实际情况的，所以在本文的设计中，验证集和测试集中的用户和物品对，相关的评论文本都是不使用的。

TransNets 模型

通常认为，用户 A 对于物品 B 的评论 rev_{AB} 是包含了情感信息的，在预测打分 r_{AB} 是一个很重要的参考。这在训练时可以获取，而测试时是不可见的。为了解决这一困境，本文提出的 TransNets 包含了两个网络：一个目标网络来处理目标评论 rev_{AB}，另一个是源网络，处理用户 A 和物品 B 的评论信息（不包括 rev_{AB}）。

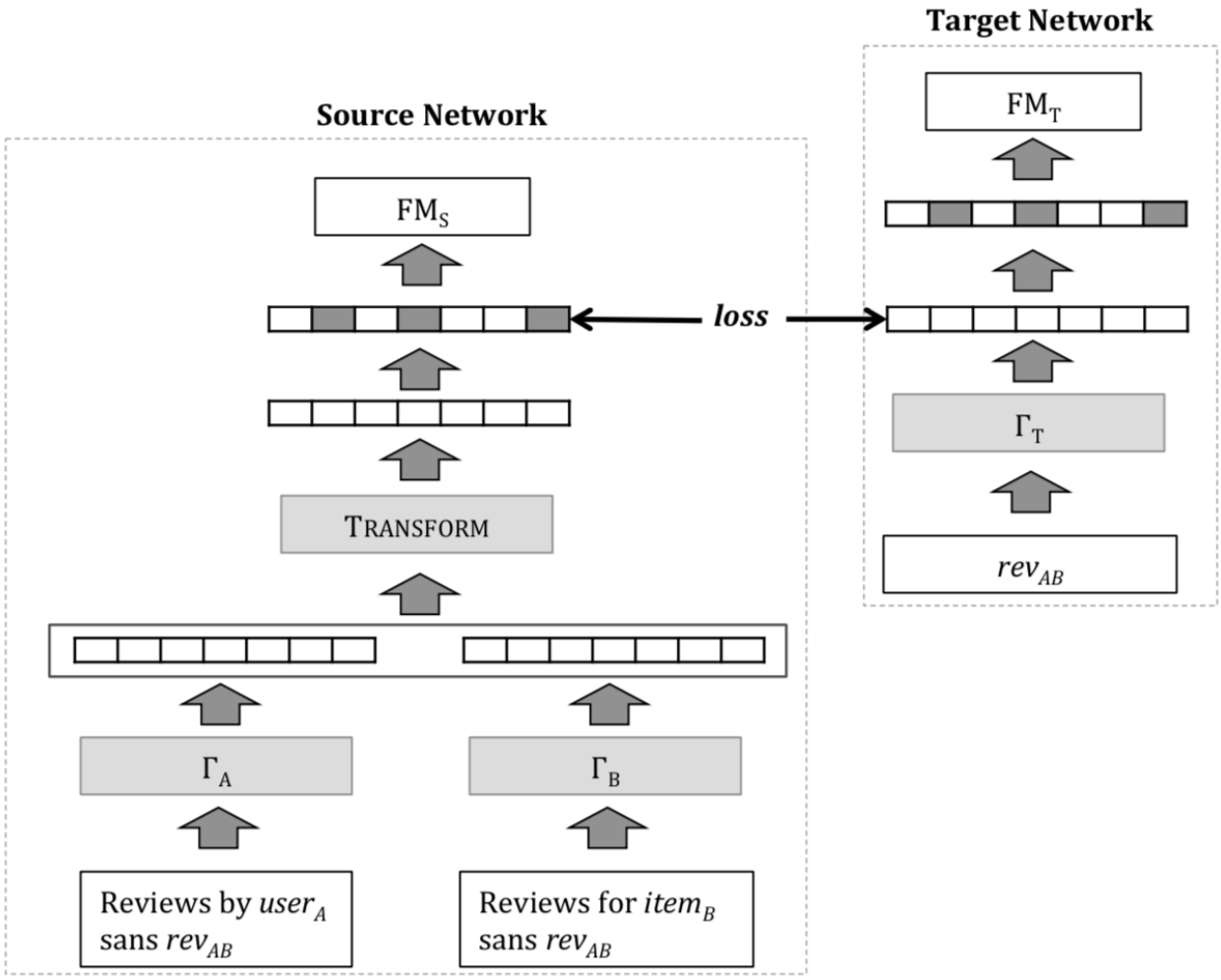

Figure 3: The TransNet architecture

目标网络通过一个 CNN 处理模块 Γ_T， 把评论 rev_{AB} 转换成向量表示 x_T，再通过 Dropout 层得到 $\bar{x}_T = \delta(x_T)$，最终的得分由 Factorization Machine FM_T 给出：$\hat{r}_T = FM_T(\bar{x}_T)$，这里使用的是用户 A 对物品 B 的真实评论文本，类似一个情感分析的任务。

源网络通过两个 CNN 模块，分别对用户 A 的所有评论（除去 rev_{AB}）和物品 B 的所有相关评论（除去 rev_{AB}），得到各自的基于评论文本的表示向量 x_A 和 x_B，令 $z_0 = [x_A x_B]$，即两个表示进行拼接，通过一个 Transform 网络（本质上就是一个 MLP）得到一个最终层的表示 z_L，之后连接一个 Dropout 层，$\bar{z} = \delta(z_L)$，最终的打分也由一个 FM_S 给出，即 $\hat{r}_S = FM_S(\bar{z}_L)$。

TransNets 训练

为了教会源网络在没有对应目标评论数据时能够得到一个近似的表示，要求最小化 $\bar{z}_L$ 和真实的评论表示 x_T 之间的 L_2 损失，因此这个子过程只更新与 $\bar{z}_L$ 有关的参数 $\theta_{trans} = \{\Gamma_A, \Gamma_B, W_l, g_l\}$，（$W_l, g_l$ 是 Transform 中每一层的权重和偏置）。最后，通过最小化真实打分 r_{AB} 和 源网络的预测打分 $\hat{r}_S$ 之间的 L_1 损失来更新源网络中的参数 θ_S，即 FM_S 中的参数。具体过程如下：

Algorithm 1 Training TransNet

procedure TRAIN(D_{train})
  while not converged **do**
    for $(text_A, text_B, rev_{AB}, r_{AB}) \in D_{train}$ **do**
      #Step 1: Train Target Network on the actual review
      $x_T = \Gamma_T(rev_{AB})$
      $\hat{r}_T = FM_T(\delta(x_T))$
      $loss_T = |r_{AB} - \hat{r}_T|$
      update θ_T to minimize $loss_T$
      #Step 2: Learn to Transform
      $x_A = \Gamma_A(text_A)$
      $x_B = \Gamma_B(text_B)$
      $z_0 = [x_A x_B]$
      $z_L = \text{TRANSFORM}(z_0)$
      $\bar{z}_L = \delta(z_L)$
      $loss_{trans} = ||\bar{z}_L - x_T||_2$
      update θ_{trans} to minimize $loss_{trans}$
      #Step 3: Train a predictor on the transformed input
      $\hat{r}_S = FM_S(\bar{z}_L)$
      $loss_S = |r_{AB} - \hat{r}_S|$
      update θ_S to minimize $loss_S$
  return θ_{trans}, θ_S

其中的 Step 2 则是呼应了 Trans 的含义。而在测试阶段，TransNets 只会通过源网络进行预测。

模型变体

在设计 TransNets 中，还有很多种选择。

- 联合学习还是分步学习。联合学习通过最小化总体损失，即 $loss_{total} = loss_T + loss_{trans} + loss_S$，这样的设计可能会使整体模型向目标网络最优的方向进行，会进入其他损失的局部最优情况，而希望的则是在源网络上有更小的损失。因此需要把目标网络部分从中分离出来，使它在有明确的打分信息下，抽象出一个更好的、与打分有关的表示。
- 独立训练，即首先训练目标网络至收敛，再开始训练源网络。在单独训练目标网络时，无法评估参数的好坏，因为难以知道收敛时是网络参数有一个比较好的泛化特征还是过拟合的情况，必须要通过数据验证或测试，这是在源网络中完成的。最终的选择是两个网络同时学习。
- 使用相同的 CNN 模块。考虑到对于用户和物品的评论中，需要关注的内容不同，一个侧重感受，一个侧重描述，而如果使用目标网络的 Γ_T 处理的话，会得到一个较为泛化的表示，这也是不那么合理的，需要分开用户、物品的表示过程。
- Dropout，为了防止过拟合。

进一步扩展

只使用评论信息是不够的，考虑到传统推荐算法中的分解模型，额外引入与评论文本无关的用户、物品表示 ω_A 和 ω_B，使得 FM 的输入从单一的 $\bar{z}_L$ 扩展为 $\bar{z} = [\omega_A, \omega_B, \bar{z}_L]$，最终的打分预测仍旧使用分解机 FM_{SE}，即 $\hat{r}_{SE} = FM_{SE}(\bar{z})$。

实验与结果

本文中进行模型测试的数据集有 Yelp17 和 Amazon 中带有评论文本的 Elec、CSJ 和 Mov 三个类目的数据。实验的设置基本与 DeepCoNN 一致。评价指标采用 MSE。相关的对比试验有三个：

- DeepCoNN，在测试用户 P 对物品 Q 的打分时，把评论 rev_{PQ} 从中剔除；
- DeepCoNN-rev_{AB}，与上一种不同之处在于训练用户 A 和物品 B 时，也不会使用评论 rev_{AB}；
- MF 模型，仅仅使用打分信息进行训练。

Table 2: Performance comparison using MSE metric

Dataset	DeepCoNN + Test Reviews	MF	DeepCoNN	DeepCoNN-rev$_{AB}$	TransNet	TransNet-Ext
Yelp17	1.2106	1.8661	1.8984	1.7045	1.6387	1.5913
AZ-Elec	0.9791	1.8898	1.9704	2.0774	1.8380	1.7781
AZ-CSJ	0.7747	1.5212	1.5487	1.7044	1.4487	1.4780
AZ-Mov	0.9392	1.4324	1.3611	1.5276	1.3599	1.2691

可见，TransNets 及其扩展取得了最好的结果，DeepCoNN + Review 的方式应该是该种模型下达到的最优结果下界。

参数分析

实验发现，Transform 中的深度网络层数以 2 层和 4 层为最佳，过多很容易在训练集上过拟合，同时学习的参数规模也会很多，学习效果差。事实上，2 个非线性层已经可以模拟出比较复杂的异或操作了。

评论分析

虽然本文的主要目的是得到更为精确的打分，但是评估模型学到的用户、物品的表示也是很重要的。为了验证 TransNets 如何找到与不可见的评论最相似的表示，并且用这个近似表示来预测打分，进行了可视化分析。一个关注服务质量和等待时间的用户和一个对价格敏感的用户，他们的评论风格、关注内容是不同的，因此，他们的表示也是不同的。以测试时的目标用户 P 和物品 Q 为例，通过源网络得到表示 z_L，作为评论 rev_{PQ} 的近似。在训练集中，把物品 Q 的所有评论 re_{CQ} 分别通过目标网络作用，得到 $x_{CQ} = \Gamma_T(rev_{CQ})$，如果 x_{CQ} 与实际用户 P 对物品 Q 的评论相似，那么 x_{CQ} 也应该与 z_L 相似，体现在欧氏距离比较近上面。

在 Yelp17 数据中，给出了一些实例：

Table 3: Original review vs. Predicted most helpful review

Original Review	Predicted Review
my laptop flat lined and i did n't know why , just one day it did n't turn on . i cam here based on the yelp reviews and happy i did . although my laptop could n't be revived due to the fried motherboard , they did give me a full explanation about what they found and my best options . i was grateful they did n't charge me for looking into the problem , other places would have . i will definitely be coming back if needed . .	my hard drive crashed and i had to buy a new computer . the store where i bought my computer could n't get data off my old hard drive . neither could a tech friend of mine . works could ! they did n't charge for the diagnosis and only charged $ 100 for the transfer . very happy .
excellent quality korean restaurant . it 's a tiny place but never too busy , and quite possibly the best korean dumplings i 've had to date .	for those who live near by islington station you must visit this new korean restaurant that just opened up . the food too good to explain . i will just say i havent had a chance to take picture since the food was too grat .
this place is so cool . the outdoor area is n't as big as the fillmore location , but they make up for it with live music . i really like the atmosphere and the food is pretty spot on . the sweet potato fry dip is really something special . the vig was highly recommended to me , and i 'm passing that recommendation on to all who read this .	like going on monday 's . happy hour for drinks and apps then at 6pm their burger special . sundays are cool too , when they have live music on their patio .
i have attempted at coming here before but i have never been able to make it in because it 's always so packed with people wanting to eat . i finally came here at a good time around 6ish ... and not packed but by the time i left , it was packed ! the miso ramen was delicious . you can choose from add on 's on your soup but they charge you , i dont think they should , they should just treat them as condiments . at other ramen places that i have been too i get the egg , bamboo shoot , fire ball add on 's free . so i am not sure what their deal is .	hands down top three ramen spots on the west coast , right up there with , and the line can be just as long .

右侧是根据 z_L 找到的最相似的评论，可见与用户实际的评论也有很多相似的地方。

其他相关工作

主要介绍与 TransNets 模型设计相关的一些工作。

Student-Teacher Model

包含两个网络，Teacher 网络较为庞大、复杂，能力强；而 Student 网络尽可能去模拟 Teacher 网络的输出，但是规模会比较小。与 TransNets 不同之处在于，

- TransNets 中的两个网络输入不同，而 Student-Teacher 网络的输入是相同的；
- TransNets 的源网络类比 Student 网络，反而是复杂的；目标网络类比 Teacher 网络，反而是要精简一些的；
- TransNets 中的两个网络是同时学习优化的，而 Student-Teacher 网络，Teacher 部分是预训练好的。

GANs

TransNets 与 GANs 都是能够产生与实际不可观测到的数据类似的数据，但是两者依然显著不同：

- GAN 的 Generator 部分产生数据是随机的，而 TransNets 的源网络部分生成近似表示是有确定的输入的；
- GAN 的 Discriminator 鉴别生成数据是否真实，而 TransNets 则保证生成的近似与实际尽可能相似，不需要区分真与假；
- GAN 需要明确数据的真与假，从而生成一个更为宽泛的表示，而 TransNets 总是有一个确定的事实数据，不需要学习一个一般的真实数据表示。

总结与点评

本文分析了 DeepCoNN 模型的不足，并且抓住了其中的局限性进行了分析与改进，同时还借鉴了其他成功模型，进行了扩展。虽然改动不是很大，但是分析得头头是道，而且每一步都是有理有据，文章写得也很有逻辑性。

我们总是想到的关于 embedding 之间的相似，通过某种对齐的方式把数据融合，这在之前想要改进 HFT 等类似的方法也想到过，比如通过 RNN 或者 CNN 从用户的评论中得到一个用户的表示和商品的表示，作为额外的输入信息放到一个深度模型中，丰富特征输入以求一个更好的结果。可能受到数据集质量的问题，并未达到预期。作者说的联合训练和分布训练也都是尝试过的，只能一声叹息，没有好好调参，找到一些比较好的数据集。