2016 Byte Cup国际机器学习竞赛解决方案

**天穹队**

钱乾  
寻仟信息科技（上海）有限公司

308456341@qq.com

陈靖  
中国科学技术大学

1094852860@qq.com

郭鹏博  
純規則研究中心

ji1tian1xi4@163.com

# 简介

头条问答是一个新兴的移动社交问答平台，基于头条5亿+用户及精准分发技术优势，在移动端以问答为形式进行的碎片化创作及互动。头条问答将信息和人精准匹配，为问题找到合适的回答者；为回答找到合适的阅读者。从而实现“让所有人问所有人，所有人答所有人”。

目前，头条问答每天已有数万用户参与答题，带来的优质回答每天有数千万的阅读。因此一个重要问题就是如何为每个热门问题找到愿意回答的专家用户并将问题推送给他们。如果问题推送策略准确度不高，为了保证问题有足够的高质量回答数，只能尽量扩大推送覆盖面，可能给部分不适合回答问题的专家用户带来打扰。为了更好的解决这个问题，今日头条联合IEEE中国分会举办了本次比赛。

赛题给定若干问题，参赛者需要预测哪些专家更有可能回答这些问题。具体的，针对每个问题和一位候选专家，参赛者需要根据计算该专家回答问题的概率。实际运营中，系统会优先向回答概率高的候选专家发送这个问题的回答邀请，直到收到的回答数量达到指定阈值。评估方面，给定一个问题，我们会按照预测概率把候选专家排序，并分别评估排序结果的NDCG@5，NDCG@10最后评分公式为 NDCG@5 \* 0.5 + NDCG@10 \* 0.5

# Models

赛题给出专家与相应的问题，需要预测被邀请专家是否回答该问题，解题思路可以有：1、将其作为分类问题，类别为未回答（0），回答（1）两类，选用合适的分类模型；2、回归问题，将target 01作为专家回答问题的概率，选用回归模型；3、因子分解，与回归类似，将专家-问题概率矩阵通过分解机进行分解，在拟合已知值得基础上预测未知值。

我们队综合了以上三种解题思路，尝试了多种模型并在最终做了融合。下面将对所使用的模型分别介绍。

## SVD

测试集需要预测专家回答问题的概率，一个很自然的想法是，将专家回答问题的概率写成等式，为专家回答问题的概率，为问题被回答的概率。将该等式推广，把p扩展成一个向量，向量的每一个分量都代表一个属性，

如果将专家E-问题Q的历史数据构成一个01概率矩阵P，则上式可写成矩阵分解的形式

其中，为专家个数，为问题个数，K为专家和问题的特征属性长度。于是可以用SVD[1]来解决这个问题，我们队使用的是加入偏置项的SVD算法，具体公式如下：

采用SGD最小化以下目标函数

为正则项。

## SVD++

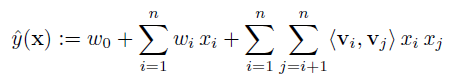
SVD++是在SVD基础上，考虑到专家的兴趣爱好对结果的影响，在SVD等式中加入专家被邀请历史项y

同样用SGD去拟合原始矩阵，这里除了更新,,外，还需要更新，Inv(e)为专家e被邀请过的问题集合，SVD++在线上NDCG的表现要比SVD好1个百分点。

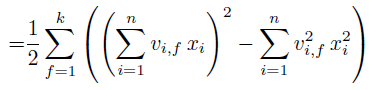
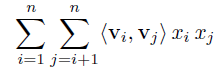
## LibFM

在验证了SVD对这个问题的效果之后，我们开始尝试用一个比较通用的fm来处理这个矩阵的分解问题，我们队一开始使用的是常用的fm库LibFM[2]，在比赛后期为了满足自定义优化需求，重新实现了fm并在优化方法上做了修正，使得预测效果得到很大的提升。

fm不仅考虑了单个特征，还考虑了特征之间的两两内积对结果的影响。其等式为

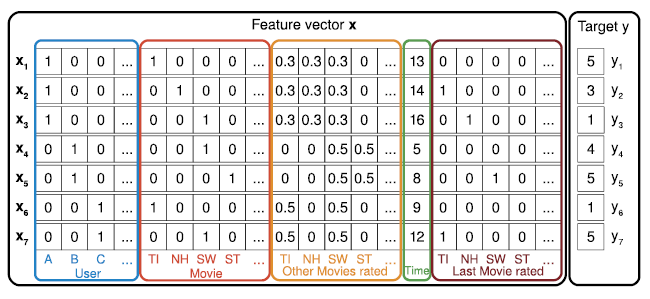


为了减小做两两内积的时间复杂度，libfm对等式中的两两交叉项做了变换



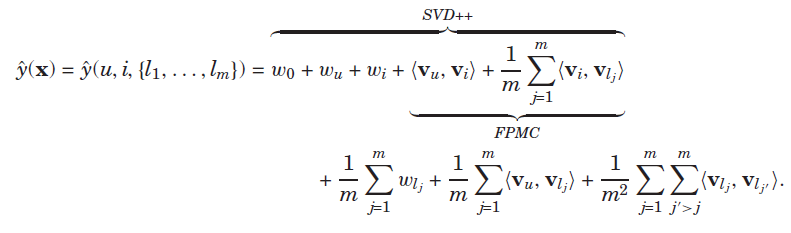
在可扩展性上，fm比SVD要强，可以灵活地增加特征属性

将这个问题类比电影推荐里的评分预测问题，其fm的特征矩阵构造形如下图：



从左往右第一个矩阵为专家E的ID编码矩阵，即专家自身的特征矩阵，第二个为问题Q的ID编码矩阵，前两个矩阵为基本矩阵，后面的矩阵可视为扩展的内容，第三个矩阵为专家被邀请过的问题矩阵，第四个矩阵为问题邀请过的专家矩阵，其特征值可以设置为1/m，m为邀请次数。

实际上，当第三个矩阵的特征值设置为时，fm的等式中就包含了SVD++



为了能更好的利用专家（问题）的邀请记录去反映专家（问题）的主要特点，我们采用TF-IDF去构造特征。我们尝试过使用提供的分词数据去构造TF-IDF特征，但是效果不显著，于是我们利用邀请数据去构造TF-IDF特征，算出来的特征值替换上述的1/m，得到很大的提升。以专家特征为例，具体做法如下：

专家ID: 邀请的问题列表

E1: Q1 Q3 Q8 … Qm

E2: Q1 Q4 Q11 … Qn

…

EN: Q2 Q4 Q13 … Qk

在这里将专家E看成doc，把问题Q看成term，即可套用TF-IDF的公式算出每个Q的TF-IDF值，将其作为第三个fm矩阵的特征值

同样的过程对问题Q进行一次，算出每个E的TF-IDF值作为第四个fm矩阵的特征值。

为了满足自定义优化需求，我们重新用python实现了fm，使用Adam[3]去优化，不仅收敛速度得到改善，效果也有明显提升。

## XGBoost

由于社交图与语言模型存在一定的相似性，我们通过相同用户或者问题的邀请看做是连接节点的边

，将<E,Q>对抽象成无向图，将这个推荐问题转化为社交图中个体节点的表征。

我们采用Deepwalk[4]算法，其算法理念为：

以任意节点为起点构建随机游走路径作为document，其中随机游走的节点可抽象的认为是word，生成大量语料训练集，使用word2vec算法训练得到节点的词向量表征。

我们将<E,Q>对转换成词向量，用来训练XGBoost[5]模型

## LR

采用与fm模型相同的特征训练一个logistic regression模型。

## DNN

采用与fm模型相同的特征训练一个1 hidden layer neural netowrk模型，使用relu作为激活函数，使用dropout率为0.75，并且使用adam优化模型

## 协同建模法

对于给定待预测的问题甲和用户子，基于对问题的思考和理解以及对数据的观察和分析，我们有如下三个思路：

1. 若用户子曾经接受过大部分的邀请，则他很有可能也会接受对于问题甲的邀请；即用户子的回答率（回答数与被邀请数的比值）越大，他回答问题甲的可能性也越大；

2. 若有用户丑和用户子被邀请回答多个相同的问题，则他们可能擅长相同的领域；此时若已知用户丑回答了问题甲，则可以推测用户子很有可能也回答了问题甲；并且这样的用户丑越多（占所有被邀请回答问题甲的用户的比例越大），用户子回答问题甲的可能性越大；

3. 若有问题乙和问题甲被邀请多个相同的用户回答，则它们可能涉及相同的领域；此时若已知问题乙被用户子回答，则可以推测问题甲很有可能也被用户子回答；并且这样的问题乙越多（占所有被邀请用户子回答的问题的比例越大），问题甲被用户子回答的可能性越大。

经过线下测试，我们设计了如下的形式来实现上述思路

将K2、K3融合得

式中W子表示用户子被邀请回答的问题集合，Wd子表示用户子回答的问题集合，Y甲表示被邀请回答问题甲的用户集合，Yd子表示回答问题甲的用户集合；α为参数，经线下测试可以取α = 2。

　　将K1、K2、K3、K4加权求和，即

式中β、γ、δ、ε为权重。经线下测试可取

K甲,子即为对用户子回答问题甲可能性的预测。

# Ensemble

对于所有机器学习模型，我们采用stacking方式。

我们构建一个2 layer ensemble，第一层使用fm,lr,xgboost,nn作为基模型，对特征切分cv fold，训练并预测每一个fold的结果，合并后生成meta feature。

第二层使用xgboost作为stacker，使用第一层fm模型的特征分别提取SVD，TSNE[6]，NMF[7]降维特征，并且合并第一层得到的meta feature训练xgboost并预测结果。

最终我们将stacking结果与协同建模法结果按权重融合，融合权重为2:1

# 总结

感谢今日头条、IEEE中国分会共同举办这次比赛，让我们有机会能够接触到真实的数据以及模型的应用场景。在比赛过程中，我们也尝试了各种各样的方法，虽然并不是总能带来提升，但是在对算法的理解上又更近了一步，对自身的能力的提升带来了很大的帮助，期望主办方今后还能举办更多有趣有挑战的竞赛。

# 参考

1. Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]//Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008: 426-434.
2. Rendle S. Factorization machines with libfm[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2012, 3(3): 57.
3. http://www.cnblogs.com/xinchrome/p/4964930.html
4. B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. arXiv preprint arXiv:1403.6652, 2014.
5. Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system. arXiv preprint arXiv:1603.02754 .
6. Nicola Pezzotti, Boudewijn P.F. Lelieveldt, Laurens van der Maaten, Thomas Höllt, Elmar Eisemann, Anna Vilanova. Approximated and User Steerable tSNE for Progressive Visual Analytics
7. P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics, 5(2):111–126, 1994

Columns on Last Page Should Be Made As Close As Possible to Equal Length