网络数据挖掘课设作业

——新闻推荐系统报告

冯子朋 郭超旭 贺翔宇 张继元 赵敏钧

{fengzipeng2017,guochaoxu2017,hexiangyu2017,zhangjiyuan2017}@mails.ia.ac.cn

zhaominjun@mails.ucas.ac.cn

1. Introduction

广义上的推荐系统可以理解为是主动向用户推荐物品的系统，所推荐的物品可以是音乐、书籍、商品、新闻条目等等。由于不同用户在兴趣爱好、关注领域、个人经历等方面的不同，以满足不同用户的不同推荐需求为目的、不同人可以获得不同推荐为重要特征的个性化推荐系统应运而生。目前所说的推荐系统一般指个性化推荐系统。

本次大作业，我们针对给定的新闻数据，分别运用 NMF 协同过滤和 SVDFeature 两种算法实现了新闻推荐系统，并搭建了线下系统用于展示推荐结果。

2. Related Work

(1) 基于协同过滤的推荐

基于协同过滤的推荐始自 1994 年明尼苏达大学 GroupLens 研究组推出的 GroupLens 系统，它引领了之后推荐系统在今后十几年的发展方向。基于协同过滤的推荐算法主要有三种：基于用户的协同过滤推荐算法(User-based Collaborative Filtering Algorithms)，基于物品的协同过滤算法(Item-based Collaborative Filtering Algorithms)，以及基于矩阵分解的协同过滤算法(SVD-based/NMF-based, etc.)。

基于用户的协同过滤推荐的基本原理是：根据所有用户对物品或者信息的偏好，发现与当前用户口味和偏好相似的 K 个“邻居”用户群；然后，基于这 K 个邻居的历史偏好信息，为当前用户进行推荐。

基于物品的协同过滤推荐的基本原理也是类似的，只是它使用所有用户对物品或者信息的偏好，发现物品和物品之间的相似度，然后根据用户的历史偏好信息，将类似的物品推荐给用户。

由于数据规模巨大，因此每个用户的评分是一个高维并且稀疏的向量，计算这样的向量之间的相似性效率并不令人满意。通过降维可以删除不相关的特征并降低噪声，从而大大减少计算量，同时可以减小稀疏性带来的影响，这就是基于矩阵分解的协同过滤算法的主要思想。

非负矩阵分解（Non-negative matrix factorization，NMF）是将一个非负矩阵 V 转换成两个正交的非负矩阵 W 和 H 的乘积：

$$V = W \cdot H$$

其中矩阵 V 的大小为 n * m，矩阵 W 和 H 的大小分别为 n * r 和 r * m，而 r 远小于 n 和 m。原矩阵 V 中的元素可以解释为对左矩阵 W 中对应行向量中的元素的加权和，而权重系数为右矩阵 H 中对应列向量中的元素。NMF 产生的矩阵中所有元素都是非负的，相比其他矩阵分解的解释可以更有实际意义。

具体到推荐问题中，我们把用户评分数据用矩阵 $V_{n\times m}$来表示，$V_{n\times m}$中的元素 V_{ij} 表示

用户 i 对物品 j 的评分。然后对矩阵 V 进行 NMF 分解为矩阵 $W_{n\times r}$和 $H_{r\times m}$，分别为特征矩阵和权重矩阵，则 r 可以解释为特征的数量。特征矩阵 W 中，每一行对应一个用户，每一列对应一个特征，矩阵中的元素 $W_{i,k}$ 可以解释为用户 i 对特征 k 的偏爱程度；权重矩阵 H 中，每一行对应一个特征，每一列对应一件物品，矩阵中的元素 $H_{k,j}$ 可以解释为物品 j 拥有特征 k 的程度。这样，特征矩阵和权重矩阵的乘积 W*H，就构造了一个稠密的评分矩阵

$$V_{ij}^{*}=\sum_{k=1}^{r}W_{ik}H_{kj}$$

矩阵 V^{*}中的元素就代表用户 i 对电影 j 的预测评分。

(2) 基于特征的推荐

基于特征的推荐是一种综合考虑内容过滤和协同过滤的推荐方法，能克服内容过滤中用户与用户、文档与文档之间的相似关系刻画不足的问题，也能有效解决协同过滤中文档内容信息刻画不足的问题。

SVD feature，是一种基于特征的协同过滤机器学习库。SVD feature 能够快速求解基于特征的矩阵分解，只要提供给足够多的特征信息，SVD feature 能够建立起一个融合各种信息比如阶级信息、时间动态等，能够进行评分和排序。在推荐系统任务中，最常用的是基于用户的特征进行推荐，而协同过滤正是推荐系统最常用的算法，同时矩阵分解也是协同过滤算法的一个主要途径，通过将已有的数据信息进行低秩分解，从而得到对用户和准备推荐的物品的隐特征信息，这种方式能够让我们更方便地对用户进行推荐等工作。我们选择 SVD feature，是因为其更能处理大型数据集，而且我们只需要做特征工程方面的工作，设置模型的各种参数，而不需要自己去实现模型的具体训练细节。

在大多数协同过滤算法中，都有涉及到三个因素：表征用户兴趣的信息，物品性质和其他一些会影响用户喜好的特征信息。SVD feature 将这三种信息都融合在模型里面，其对一个用户对一种物品的评分表示为：

$$\hat{y}(\alpha,\beta,\gamma)=(\sum_{j=1}^{s}\gamma_j b_j^{(g)}+\sum_{j=1}^{n}\alpha_j b_j^{(u)}+\sum_{j=1}^{m}\beta_j b_j^{(i)})+(\sum_{j=1}^{n}\alpha_j p_j)^{T}(\sum_{j=1}^{m}\beta_j q_j)$$

其中，模型的参数有$\theta=\{b_j^{g},b_j^{u},b_j^{i},p,q\}$，而$p_j,q_j\in R^{d}$，是一个 d 维隐向量，与相应的用户特征和物品特征相联系。$\{\alpha,\beta,\gamma\}$则分别是用户特征、物品特征和一些能影响用户喜好的其他特征信息。这些参数是通过随机梯度下降 SGD 进行优化的，而优化的损失函数有多种选择包括线性函数,Sigmoid 函数和平滑的 Hinge-loss。因为我们这里将推荐看作每一个特征对的二分类问题，所以我们选自 Sigmoid 函数和 Log-Likelihood 损失函数：

$$Loss=r ln\tilde{r}+(1+r)\,ln(1+\tilde{r})+R$$

$$\tilde{r}=f(y)=\frac{1}{1+e^{-y}}$$

3. Approach

我们小组共使用了两种方法进行系统设计——基于 NMF 的协同过滤推荐和基于特征的推荐。下面分别对两种推荐方法的实现过程进行详述。

1. 基于 NMF 的协同过滤推荐

- 数据处理

来源 一万名国内某著名财经新闻网站得用户一个月的全部浏览记录

数据格式 共有五个域：用户编号、新闻编号、访问页面的时间(Unix 时间戳)、新闻标题、新闻正文，例如:

user_id	news_id	read_time	news_title	news_content
5218791	100648598	1394463264	消失前的马航 370	【财新网】（实习记者葛菁）据新华社消息，马来西亚航空公司表示...
5218791	100648802	1394463205	马航代表与乘客家属见面	3 月 9 日，马来西亚航空公司代表在北京与马航客机失联事件的乘客家属见面。

提取给定数据集中的用户编号、新闻编号和访问时间的信息。因为新闻标题和正文信息在协同过滤中用处不大，所以不做提取。经过数据统计，浏览记录共 **116,224** 条，用户数共 **10,000 个**，出现新闻数共 **6183** 条。根据访问时间的先后，以 3 月 20 号为界限，前 20 天的数据(**83209** 条)作为训练数据， 后 10 天的数据(**18995** 条)作为测试数据。，并按照“用户-新闻-访问时间”的格式进行保存。其中，为方便实现，“用户”和“新闻”都从 0 开始按顺序进行了重新编号。

- 构建“user-item”矩阵

user-item 矩阵的大小为用户数*新闻数，为 0-1 稀疏矩阵。在训练集中，若用户 i 点击了新闻 j，则$(i，j)$位置为 1，其他为 0。

- 非负矩阵分解

对单个用户来说，访问的新闻是极少的。这样造成了“user-item”矩阵含有大量的空值，数据极为稀疏。矩阵分解的核心思想认为用户的兴趣只受少数几个因素的影响，因此将稀疏且高维的“user-item”矩阵分解为两个低维矩阵，即通过 User、Item 评分信息来学习到的用户特征矩阵 P 和新闻特征矩阵 Q。通过重构的低维矩阵预测用户对新闻的评分。

假设输入评分矩阵为 R 为 M×N 维矩阵，通过非负矩阵分解得到用户特征矩阵$P_{M\times K}$和物品特征矩阵$Q_{K\times N}$，其中$K \ll M, N$。

- 生成推荐列表

用户 u 对物品 i 的评分预测方法为：

$$r_{uj} = q_j^T p_u$$

其中p_u和q_j分别为用户 u 和物品 i 的特征向量，两者的内积即为所要预测的评分。用户与新闻相关度越高，则预测的评分越高。

去除掉目标用户在训练集中点击的新闻，计算其他每条新闻的分值，按照分值排序，选择前 l 个新闻作为推荐。

- 算法验证

由于验证数据集的特殊性，为测试推荐的性能，选用了如下两个常用指标进行验证。

① P@K：设置一个排序位置 K，计算前 K 个位置中正确推荐的新闻所占百分比。（若测试集中实际点击的新闻出现在推荐列表中，即认为被正确推荐）

② MAP：考虑出现过正确推荐的新闻的位置 $K_1, K_2, \ldots K_R$，分别计算位置 $K_1, K_2, \ldots K_R$ 的 P@K，并取平均值。它充分考虑了推荐顺序对用户的影响。

2. 基于 SVD feature 的推荐

（1）特征构造

我们将推荐任务变成一个二分类问题，对每一个用户，我们在训练集寻找他阅读过的新闻和没有阅读的新闻（在这里我们假设用户没阅读的新闻是他不感兴趣的），然后对应每一个用户阅读过的新闻，我们随机挑选一个或多个其没阅读过的新闻，然后构成一个或多个特征对，再送入判别模型进行训练。对于选取特征的方式，这里涉及到用户的特征和新闻的特征。对于新闻的特征，我们是计算每一篇新闻里面的关键词（高频词），综合全部新闻的关键词然后将其构成一个字典，相应地，每篇新闻就有对应字典的一个词向量，将对应的词向量作为新闻的特征向量；而对于用户的特征，我们用其看过的所有新闻的特征作为其特征，具体地，我们用其看过的所有新闻的关键词组成词向量，并作为用户的特征向量。这里我们的特征向量没有使用加权组合的方式，对词向量中的每个词也是没有相应的权重系数，即权重均为 1。

（2）特征格式

我们构造的特征格式以以下形式提供给模型：

评分 全局特征的数目 用户特征数目 物品特征数目 用户特征 物品特征

以文本形式存储训练特征，文本文件的每一行均以这种格式表示，摘取某一行特征如下：

```
1     0     62    14    16873:1 21760:1 258:1   6699:1  6533:1  6663:1  1544:1  139:1   5772:1  1165:1  3726:1  6927:1  9489:1  10258:1 6723:1  12692:1 9070:1
1604:1  7835:1  25244:1 23198:1 25631:1 6304:1  7585:1  977:1   7843:1  2085:1  13225:1 11307:1 22447:1 9480:1  8435:1  7606:1  11963:1 19903:1 7235:1  7620:1  6305:1
3275:1  8529:1  7633:1  9938:1  7510:1  6615:1  8536:1  22749:1 25822:1 22752:1 5091:1  2661:1  6993:1  22121:1 21100:1 9709:1  5358:1  9203:1  19985:1 9077:1  23415:1
1658:1  13375:1 12028:1 6565:1  21877:1 3770:1  7096:1  8618:1  14878:1 9708:1  2187:1  18495:1 20552:1 677:1   20331:1 3659:1  13084:1
```

（3）模型训练参数

对于模型训练我们只需要选择激活函数类型 active_type，我们选择 Sigmoid 函数，即 active_type=3。输入数据形式 Input_type，因为 SVD feature 支持输入数据以文本形式或者二进制文件形式，这里我们为了数据训练效率，将输入数据转成二进制形式，因此选择二进制输入，即 Input_type=1。因为这里我们使用 pair-rank 的思想，所以模型选择 model_type 选择为排序模型，即 model_type=1。最关键的参数是我们模型的学习率，实验中我们尝试了不同的学习率，0.001，0.0005 和 0.0001。

4. Experiments

分别对这两种实现方法进行结果分析：

1. NMF 协同过滤：

在实验中，我们基于用户的相关性及文档之间的相似度，对朴素的 NMF 算法进行了扩展，得到了如下目标函数形式：

$$\left||V-WH|\right|_F^2+\alpha*L_1*(\left||vec(W)|\right|_1+\left||vec(H)|\right|_1)+0.5*\alpha*(1-L_1)*(\left||W|\right|_F^2+(\left||H|\right|_F^2)$$

其中，超参数α及L_1用于调节对于参数的惩罚强度，这里我们对 2 范数及 1 范数进行了折中处理。由于用户文档矩阵的高度稀疏性，我们对正则项选取较大权重(α=3, L_1=0.3)，以保证重构矩阵的局部平滑。

进一步的，考虑重构后的矩阵$V \in R^{m\times n}$，左乘 user-user 余弦相似度矩阵$S_u \in R^{m\times m}$，

$$V^* = S_u V$$

等价于通过考虑其他用户对该条新闻的浏览情况，重构了用户 u 对文档 i 进行阅读的可能性。即通过 Pearson 系数进行了加权求和：

$$V_{u,i}^* = \frac{1}{const}\sum_{k=1}^{m} V_{k,i}S_{u,k}$$

对于与该用户阅读习惯相似的用户如果阅读了某些文章，那么我们将相应提升该用户阅读

某些文章的概率。类似的，通过右乘 item-item 相似度矩阵$S_I \in R^{m\times m}$，

$$V' = VS_I$$

等价于通过考虑文档之间的相似度(选取每篇文档 Top20 的关键词计算 Tf-idf 的余弦相似度)，重构用户 u 对文档 i 进行阅读的可能性。对于与用户阅读过的文章相似的文章，我们认为用户也有较大几率选择去阅读。最终我们得到的重构矩阵为：

$$\tilde{V} = \lambda(S_u WH) + (1-\lambda)(WHS_I)$$

通过对数据的可视化分析(如图 1)，我们注意到原始的矩阵 V 是高度稀疏化的，训练样本中有大量用户的阅读的新闻条目不足 10 条，却需要进行同样数量级的新闻推荐，这意味着我们需要尝试对用户未体现出兴趣的新的领域进行推荐。因此，S_u和S_I的加入将提供 NMF 之前无法参考的新闻自身属性的信息和用户之间的相关度的信息。

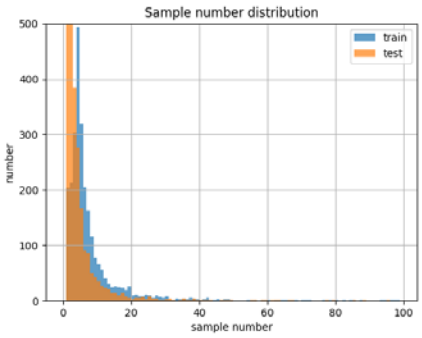

图 1.

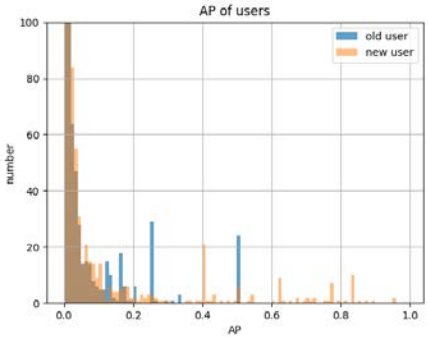

图 2.

此外，通过对目标函数收敛判别条件的重构，我们极大放宽$\left||V-WH|\right|_F^2 < \epsilon$的阈值限制，以期能够解决冷启动的问题。在实验中，我们令$\epsilon = 0.5$，最终得到较为稠密的重构矩阵$\tilde{V}$。实验结果表明，通过S_u和S_I的修正，冷启动的问题可以得到一定的缓解，但是可解释性不强。无法对重构出的非零元素的位置和大小进行准确的估计。

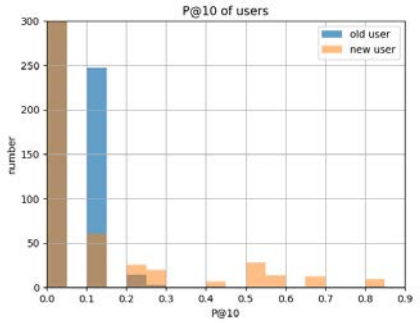

图 3.

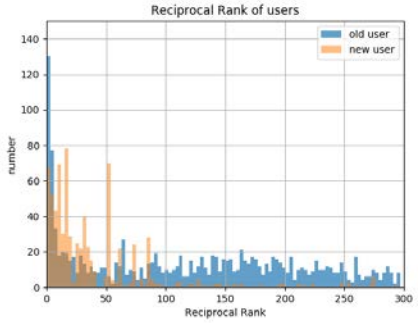

图 4.

实验分析：

在具体的实验中，我们采用了三种结果的度量标准(mAP, P@K, Reciprocal Rank)，并分别对冷启动的样本及曾出现过的用户进行了测试。如图 2,3,4 所示，冷启动的样本结果要略优于有训练样本的测试用户。我们分析认为：

1) 冷启动的样本的测试样例偏少，而曾出现在训练样本中的测试数据推荐难度大且测试样例较多造成了该问题。
2) 可以看到绝大多数测试样例推荐效果不佳，但如图 4 所示，仍存在大量的精准推荐的情况，即 Reciprocal Rank 接近于 1 的情况。考虑到原始用户文档矩阵的稀疏度小于 0.1%，我们认为 NMF 并不能很好的解决 0-1 矩阵过于稀疏的问题，且二值信息也不

足以体现只出现少量记录的用户的偏好性。

测试数据结果如下：

	precision@5	precision@10	mAP	Reciprocal Rank
New User	**9.1%**	**11.3%**	**0.11**	**447**
Old User	1.67%	1.14%	0.02	675
All User	3.47%	3.58%	0.03	619
Best Result	100%	100%	1.00	1

2. SVD-feature:

我们实验过程中发现有 700 多个用户属于冷启动状态，所以在构造用户特征的时候我们使用在训练集中频繁被阅读的新闻的关键词特征作为这些冷启动用户的特征。进行测试时，我们尝试不同的学习率和训练迭代次数，对测试集中存在的 4000 多篇新闻进行打分，然后对于每个用户，我们取相应的 TopK 分数的新闻作为其推荐结果。以下为实验结果：

Learning Rate	Iterations	precision@5	precison@10	Reciprocal Rank
0.001	100	--	--	--
0.0005	100	1.18%	1.08%	**535**
0.0005	50	1.50%	1.11%	**535**
0.0001	100	1.28%	**1.13%**	**535**
0.0001	50	**1.62%**	1.12%	536

其中学习率 0.001 的情况是学习率太大导致模型不收敛，对新闻的打分均为 nan。

实验分析：

从实验结果中可看出，我们的测试比较低，Precision@k 只保持在 1.3%左右。原因可能是我们在构造特征的时候存在以下问题：

1） 构造用户和新闻特征的时候没有对特征向量没有进行归一化

2） 简单地将新闻高频词作为新闻的特征，权重均为 1，没有更好的权重方式

3） 构造用户特征的时候没有对新闻特征给一个权重，而是简单组合。

4） 对于推荐任务，因为对于不同的测试用户，可能存在不同的点击率，统一地使用 Precision@k 这种评价方式可能会因为数据集中的用户数据不均衡而造成这种准确率衡量方式不是很客观。

5. Discussion

（1） 与其它任务相比的异同点

新闻非常讲求时效性，更新速度很快，相比较用户特性更为稳定；并且一般情况下新闻条数远大于用户数量，因此基于用户的协同过滤在新闻推荐系统中的使用效果更好。而电商平台等的推荐，用户众多，商品特性更为稳定，因此基于物品的协同过滤使用效果更好。在不同的应用场合，还需要综合考虑，以选用更为合适的推荐算法。

（2） 优缺点分析

NMF 协同过滤

基于协同过滤的推荐机制是现今应用最为广泛的推荐机制，它有以下几个显著的优点：

① 容易理解，计算简单。

② 它不需要对物品或者用户进行严格的建模，而且不要求物品的描述是机器可理解的，所以这种方法是领域无关的。

③ 这种方法计算出来的推荐是开放的，可以共用他人的经验，很好的支持用户发现潜在的兴趣偏好。

而它也存在以下几个问题：

① 方法的核心是基于历史数据，所以对新物品和新用户都有“冷启动”的问题。

② 推荐的效果依赖于用户历史偏好数据的多少和准确性。

③ 对于一些特殊品味的用户不能给予很好的推荐。

SVD-feature

优点：

① SVD feature 简化了构造一个推荐系统的工作，我们只需要清洗特征和构造相应的用户特征、物品特征和一些其他的特征就好，相应的，我们可以根据任务需要而随时加入新的特征而不需要过多繁琐的工作。同时，SVD feature 做预测的时候能够直接输出分数，让我们对推荐结果有一个比较直观的认识，能够对模型进行调整。

② SVD feature 时由于我们采用二进制输入方式，训练 1 个 epoch 平均时间为 15 秒，而对测试集中所有用户，每个用户 4000 多篇新闻进行测试也只耗时 120 秒左右，速度上具有更大的优势。

缺点：

SVD feature 同时限制住了使用的模型的形式和规范，总体灵活度不够高，如果我们想做一些模型的调整，就只能在源码上进行修改。

（3） 可能的改进方向

从结果分析来看，单一的推荐算法都没有达到理想的效果。可以考虑将几种不同的推荐按照一定权重组合起来，通过在测试数据集上反复实验获取具体权重的值，从而达到最好的推荐效果。