豆瓣电影个性化推荐

—— 期末小组 project 报告

覃可欣(17307110487)：数据爬取、数据可视化、pre 视频录制

樊可(17307110287)：传统推荐算法实现，相关部分 PPT 制作

员司雨(17307110448)：深度学习算法实现，相关部分 PPT 制作

摘要： 本项目以爬取的豆瓣电影和用户数据为研究对象，通过对数据进行可视化分析，发现数据呈现社区化特征。基于数据的社区化特征，我们建立了 Bias-SVD 和 FM 传统推荐模型，并基于文本卷积神经网络搭建了 Movie-Net 神经网络推荐模型。通过模型比较看出，Movie-Net 神经网络模型效果最为优秀。我们最终基于 Movie-Net 神经网络模型，实现了合理的豆瓣电影个性化推荐。

1、 问题背景

中国电影市场近年来快速发展，从 2010 年起票房正式迈入“百亿时代”，到 2018 年，中国电影市场总票房突破 600 亿元，成为仅次于美国的全球第二大电影市场。同时，互联网入局电影市场，不仅带来了买票方式的改变，也改变了人们的观影习惯，越来越多的人通过视频网站观看电影。在这样的背景下，用户如何在海量电影中寻找自己喜欢的电影，网站又如何精确地向用户投放合适的电影？基于上述问题，我们根据豆瓣电影数据实现了推荐系统。

豆瓣电影是国内最权威、最有公信力的电影打分平台, 收录了大量国内外的电影数据，用户可以自行添加电影信息，并可以对收录的电影进行收藏、评论、打分等。由于豆瓣电影平台不同于烂番茄网站的专业影评人评分机制，每个用户的数据高度个性化，可以利用用户的历史数据挖掘用户特征，根据用户的个人喜好来向用户推荐电影。

2、 豆瓣爬虫与数据集

为了优化推荐系统表现、提升预测准确性，并根据用户关系建立豆瓣社交网络，我们通过爬虫和数据清洗重构了维度更多、体量更大的数据集。

2.1 豆瓣爬虫设计

在本实验中，我们实现了对于豆瓣电影网站 [1]的多线程自动爬虫。为了构建关系更紧密的用户关系网络，我们通过获取原有用户的关注用户和粉丝用户，并将其中的新用户加入扩展队列（深度优先搜索）这一方式来扩大用户数据集。在爬取了用户数据后，我们进一步爬取了所有用户看过或想看的电影信息。

豆瓣用户的关注信息和粉丝信息需要通过登录获得，在本实验中，为了避免模拟登录带来的验证码问题，我们采用浏览器 cookies 自动登录的方式来爬取信息。另外，爬虫主要运行时间消耗是请求网页时的 IO 阻塞，所以开启多线程，让不同请求的等待同时进行，可以大大提高爬虫运行效率。在实验中我们利用

[1] 豆瓣电影网址：https://movie.douban.com/

threading 包实现多线程，利用 *Queue* 包实现多线程编程的先进先出，在生产者消费者线程之间安全地传递信息。

2.2 数据集介绍

爬取数据完毕后，为了提高推荐系统的预测效率和可解释性，我们对数据进行了清洗，去除了观看电影部数少于 5 部的用户和没有评分信息的电影，并将信息储存在 csv 文件中。数据集信息如下：

数据集	用户	电影	评分条数
small	5000	41785	921462
big	24044	114275	4719011

表 1：爬取前后数据集对比

除扩大数据集条数外，我们在爬取时增加了数据集的维度，从更多角度挖掘电影的相似性和用户偏好，其中电影信息增加了：年份、国家/地区、时长；用户信息增加了：想看的电影、关注用户、粉丝用户。

3、 描述性分析

在构建推荐系统之前，首先对豆瓣关系网络进行描述性分析，初步刻画豆瓣电影用户的关系网络特征。

3.1 网络统计性质

从统计结果可以看出，在豆瓣社交网络中，由于关注关系具有较高的传递性，相比于 web（平均聚类系数 0.081）而言，网络具有较高的平均聚类系数。两个节点间的平均路径长度也较短，呈现“小世界现象”。图密度较小，整体网络稀疏。

平均度	12.431
最大直径	7
图密度	0.015
平均聚类系数	0.205
平均路径长度	3.976

表 2：豆瓣关系网络统计性质

3.2 可视化算法

3.2.1 社区发现

社区发现的目的是使得划分后的社区内部连接较为紧密，而在社区之间连接较为稀疏。为了更清楚地展现豆瓣关系网络的结构，我们使用 fast unfolding 算法进行基于模块度的社区发现，将豆瓣中的用户划分为不同的社区，并在可视化中展示为不同的颜色，从而更清楚地展现用户之间的关系。

3.2.2 力导向算法

在可视化过程中，如果随机分配节点位置，整个网络会显得较为杂乱，节点连接没有规律，不能突出网络特点。力导向算法模拟物理世界中的作用力，施加在节点上，并迭代计算以达到合理放置节点，使图更紧凑，可读性强，能充分展现网络的整体结构及其自同构特征。

3.3 豆瓣关系网络展示

我们随机抽取 10 个用户作为种子节点，扩展构建 1000 名用户豆瓣关系网络，通过 *PyEcharts* 和 *Gephi* 进行可视化，并刻画了相关指标统计图的幂律分布形态。为了更清晰地呈现网络结构，在实验中还选取了一个小社区进行两种力导向布局的可视化实践。

3.3.1 整体网络图

图 1：html 网络结构展示

图 2：html 节点关系展示

图中不同颜色的区分是社区发现算法的结果，不同社区被赋予不同颜色，节点的大小由度决定。在 html 文件中，鼠标移动到节点上方会显示相应节点的名称，移动到边上方会显示连接的两个节点名称和边的方向。在可视化结果中国，可以明显看到超级节点的存在，这一特点符合 Zipf 分布，即少数重要节点连接了很多节点，说明豆瓣关系网络符合真实世界无标度网络的特征。

另外，我们观察到相互关注的节点占比不大，关系大多为单边。可见豆瓣网络中熟人关系不多，更多是普通用户对于大 V 的关注，整个网络接近于优先链接模型；另外，图中的三元闭包也相对较少，这可能也是由于熟人不多，用户之间的社交压力较小。上述特点符合豆瓣电影平台的特征，即不是社交网站，更多是信息交流平台。

3.3.2 指标统计图

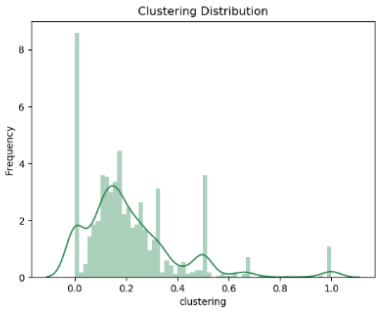

图 3：度分布图

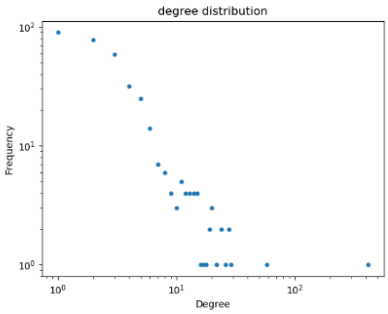

图 4：聚类系数分布

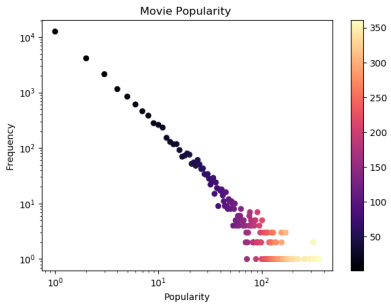

图 5：电影流行度分布

上面三图分别为1000个用户关系网络中的电影流行度分布，用户度分布和节点的聚类系数分布。三个图都大致符合幂律分布，并都呈现出“重尾”的特征。其中，度分布和聚类系数分布的重尾由网络中的大型中心节点形成。而电影流行度的“重尾”从某种程度上体现了马太效应和社区的羊群效应，即用户倾向于看

流行的电影和邻居节点看过的电影。

3.3.3 小社区可视化

为了更清晰地展示网络关系结构，我们选择了网络中的一个社区在*Gephi*中进行具象化，主要对比了两种力导向算法。颜色代表不同社区，点的大小展示节点的度。Fruchterman Reingold 布局各向趋同，社区关系和节点的中枢作用不明显；相对而言，Force Atlas布局效果更优，同一社区分布更集中，清晰地展示出网络的超级节点、中枢节点和边缘节点。

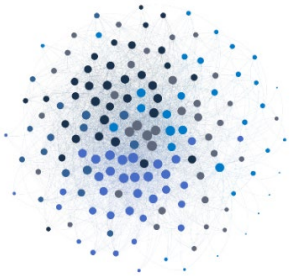

图6：Fruchterman Reingold 力引导布局

图7：Force Atlas 力引导布局

4、 传统推荐算法

4.1 FunkSVD 算法与 BiasSVD 算法

在推荐系统中，按照每一行代表一个用户，每一列代表一个物品，在用户对应的行与物品对应的列交叉的位置打上用户对物品的评分值，用户没有打分的物品设置为缺失的模式，可以将数据整理成一个评分矩阵。矩阵分解方法是推荐系统中一类有效的传统算法。

	Top Burgers	La Parisienne	Cheap Eats	Wings 'n' Things
Emily	4	-	-	3
Bob	5	2	5	-
Jane	-	5	1	-

=

	Low Cost	White Tablecloths	Polite Serving Staff
Emily	?	?	?
Bob	?	?	?
Jane	?	?	?

x

	Top Burgers	La Parisienne	Cheap Eats	Wings 'n' Things
Low Cost	?	?	?	?
White Tablecloths	?	?	?	?
Polite Serving Staff	?	?	?	?

图 8：SVD 算法因子示意图

FunkSVD，也称隐语义模型，能够较好的解决矩阵分解的稀疏问题。考虑 m 个用户 n 个物品的用户评分矩阵$R_{m\times n}$，将矩阵分解为 2 个矩阵$P_{m\times k}$和$Q_{k\times n}$，满足$R_{m\times n}=P_{m\times k}\times Q_{k\times n}$。这等价于把用户和物品都映射到一个 k 维空间中，k 维空间每个维度对应着一个隐因子（如图 8 所示），表征了用户和物品的一种特征，使用用户因子与物品因子的内积来预测用户评分，即$r_{ui}\approx q_i^T p_u$，求解优化问题：

$$\min_{q_i,p_u}\sum_{(u,i)\in K}(r_{ui}-q_i^T p_u)^2+\lambda(\|P\|_F^2+\|Q\|_F^2)$$

损失函数一部分由已知的用户评分与预测评分的偏差决定，控制预测误差；另一部分是正则化项目，由用户和物品的因子大小决定，减少过拟合，提高泛化能力。

BiasSVD 在 FunkSVD 的基础上加上了偏离偏移项，偏移项衡量了用户评分与因子无关的部分。如优秀的电影广受大众喜爱，而拙劣的电影往往评分较低；有批判性的用户对电影评分普遍较低，而抱有欣赏性眼光的用户则评分普遍偏高。为每个用户增加偏移项$b_{ij} = \mu + b_i + b_j'$，$b_i$表示用户的平均评分，$b_j'$表示电影的平均评分，$\mu$代表豆瓣这个网站的平均评分。最小化 x 修正后的损失函数：

$$\underset{p_j,q_j,\mu,b_i,b_j'}{argmin} \sum_{(i,j)\in K} (r_{ij} - \mu - b_i - b_j' - q_j^T p_i)^2 + \lambda(\| P \|_F^2 + \| Q \|_F^2 + \| b \|_2^2 + \| b' \|_2^2)$$

4.2 因子分解机模型

受到广义线性模型和矩阵分解的启发，Steffen Rendle 提出了因子分解机(Factorization Machine)，旨在处理线性模型的特征组合问题。

user						movie (item)					time	rating	
$x^{(i)}$	u_1	u_2	u_3	u_4	...	i_1	i_2	i_3	i_3	...	t	r	$y^{(i)}$
$x^{(1)}$	1	0	0	0	...	1	0	0	0	...	2	5	$y^{(1)}$
$x^{(2)}$	0	1	0	0	...	0	0	1	0	...	18	1	$y^{(2)}$
$x^{(3)}$	0	1	0	0	...	0	0	0	1	...	6	2	$y^{(3)}$
$x^{(4)}$	0	0	1	0	...	0	1	0	0	...	12	3	$y^{(4)}$
$x^{(5)}$	1	0	0	0	...	0	0	1	0	...	3	5	$y^{(5)}$
:	:	:	:	:	:	:	:	:	:	:	:	:	:
$x^{(m)}$	0	0	0	1	...	0	1	0	0	...	9	4	$y^{(m)}$

图 9：FM 算法中独热编码示意图

在线性模型中，为了考察每个特征对评分的影响，可以对每个特征独热编码，然后进行回归。但是独热编码后，数据高度稀疏，且一阶特征难以有效预测评分，可以通过引入二阶项的方式对线性模型进行拓展，将$\hat{y} = w_0 + \sum_{i=1}^n w_i x_i$修改为$\hat{y} = w_0 + \sum_{i=1}^n w_i x_i + \sum_{0<i<j<=n} w_{ij} x_i x_j$。

由于特征高度稀疏，组合权重w_{ij}难以得到有效学习，考虑到二阶特征之间不独立，可以使用隐因子关联，即把所有的w_{ij}看成一个对称矩阵W_{n*n}，进行矩阵分解$W_{n*n} = V_{n*k} * V_{k*n}^T$，$V_{n*k}$是因子矩阵，而$w_{ij} = \langle \vec{v_i}, \vec{v_j} \rangle$为因子的内积，线性模型简化为$\hat{y} = f(\vec{x}) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{0<i<j<=n} \langle \vec{v_i}, \vec{v_j} \rangle x_i x_j$。一方面，二阶项参数个数从$O(n^2)$降低到$O(kn)$,降低了计算量，减少过拟合，另一方面，即使$x_i x_j = 0$，二阶参数也能得到有效学习，提高了学习效率。

5、 深度学习模型

除了上述介绍的传统推荐算法以外，近些年来深度学习被广泛应用于推荐系统当中。在基于内容的过滤中，深度学习技术主要用于提取特征，以从异构数据源生成基于内容的用户/项目文件。在混合推荐系统中，利用深度学习方法从辅助信息中提取特征，并将其集成到推荐过程中。深度学习技术被用来处理推荐系统的稀疏性和冷启动问题，方法是从辅助信息中提取特征并将它们集成到用户项目偏好中。此外，基于深度学习的方法被用于将高级和稀疏特征的维数降低为低级和密集特征。

现存的文献表明，基于深度学习的方法比传统的推荐算法(如基于矩阵分解和最近邻的方法)提供更准确的推荐。出现这种情况的主要原因是深度学习算法提供了用户偏好的非线性表示，能够发现意想不到或不可理解的行为。

5.1 文本卷积神经网络

本项目，考虑到电影名属于短文本信息，我们将文本卷积神经网络应用到推荐系统当中。文本卷积神经网络结构如下图 10 所示。

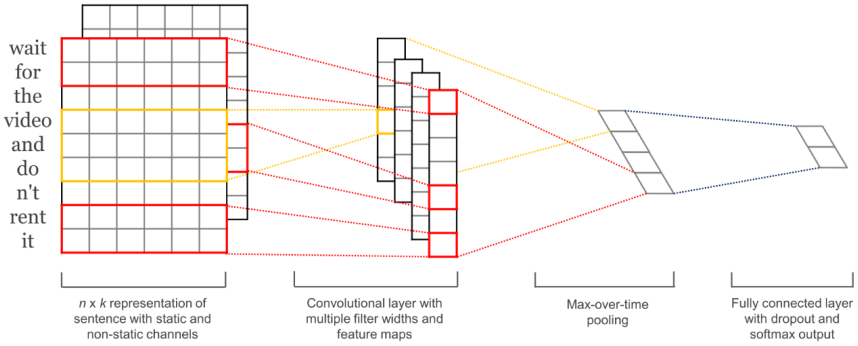

图 10：文本卷积神经网络结构图

网络的第一层是词嵌入层，由每一个单词的嵌入向量组成的嵌入矩阵。下一层使用多个不同尺寸（窗口大小）的卷积核在嵌入矩阵上做卷积，窗口大小指的是每次卷积覆盖几个单词。这里跟对图像做卷积不太一样，图像的卷积通常用 3x3、5x5 之类的尺寸，而文本卷积要覆盖整个单词的嵌入向量，所以尺寸是（单词数，向量维度），比如每次滑动 4 个或者 5 个单词。第三层网络是通过 max pooling 得到一个长向量，最后使用 drop out 做正则化，得到电影 Title 的特征。

5.2 Movie-Net 神经网络

结合上述的文本卷积神经网络，我们构建豆瓣电影推荐神经网络 Movie-Net 神经网络结构如图 11 所示

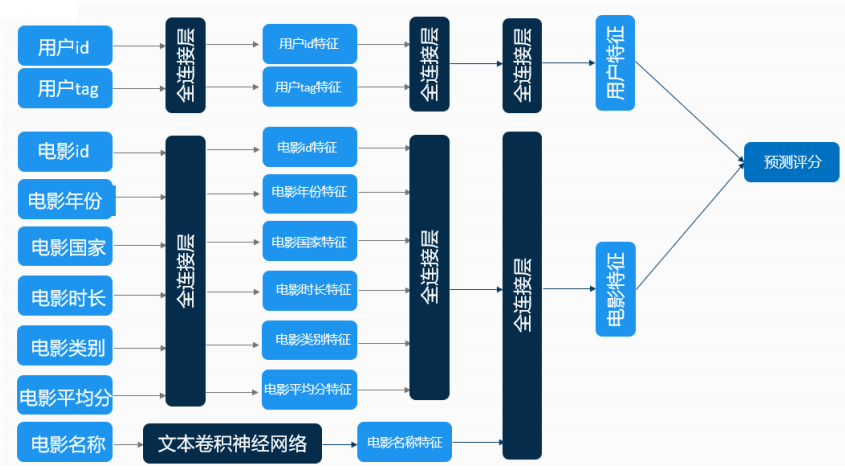

图 11：Movie-Net 神经网络结构图

我们通过 embedding 层和多层感知机，分别提取用户特征和电影特征，并做点乘后降维得到对每一部电

影的预测评分。我们使用 Adam 算法作为优化方式、MSE 作为损失函数来优化整个 Movie-Net 网络。

5.3 模型部分小结

传统推荐系统算法与深度学习算法结果对比如表 3 所示。

模型	MSE	MAE	NDCG
FM	0.58	0.60	0.82
SVD	0.62	0.61	0.80
Movie-Net	0.58	0.56	**0.99**

表 3：传统推荐系统算法与深度学习算法结果对比

可以看出，无论是 MSE 和 MAE，深度学习算法略优于传统算法，但是深度学习的 NDCG 明显优于传统算法的 NDCG。因此我们可以得出，深度学习算法在推荐豆瓣电影这一应用上是优于传统推荐系统算法的。

6、 豆瓣电影个性化推荐

利用预测效果最好的 Movie-Net 神经网络生成的用户特征矩阵和电影特征矩阵做豆瓣电影个性化推荐。主要有两种推荐方式，第一种是推荐同类型的电影，即计算当前看的电影特征向量与整个电影特征矩阵的余弦相似度，取相似度最大的 5 个作为推荐结果；第二种是推荐用户喜欢的电影，使用用户特征向量与电影特征矩阵计算所有电影的评分，取评分最高的 5 个作为推荐结果。

6.1 推荐同类型的电影

《北京青年》主要讲述土生土长在北京的四个堂兄弟，何东、何北、何西、何南，这四个北京青年，为了各自的理想而努力工作，经历爱情考验和生活洗礼的励志故事。推荐与《北京青年》同类型的电影结果如表 4 所示，

推荐电影	简介
璀璨的婚礼	该片讲述保罗和刘纯恩在家人和朋友的帮助下克服困难，选择以一场简单而温馨的婚礼阐释对爱的定义。
戚继光英雄传	真实讲述戚继光一生的英雄传奇，实地探访戚继光成长、战斗和生活过的地方
太空熊猫英雄归来	该影片讲述了熊猫族王子泰隆带领族人寻找新家园的故事。
风口青春	该片讲述了主人公在一个曾经分离又重逢的初恋情人和一个陪他走过创业最艰难时期、温柔贤惠的职场丽人之间感情抉择的故事。
纯真时代	本片以 1398 年发生在朝鲜李氏王朝的"戊寅靖社"事件为背景，讲述了三个男人的欲望和野心。

表 4：推荐与《北京青年》同类型的电影结果

可以看到，推荐的五部电影的内容在某些方面均与《北京青年》相似。相对而言看上去差距较大的《戚继光英雄传》与《北京青年》同年上映，都有“2012”、“励志”、“国产”的标签，在类型上同属于“剧情”，可见推荐结果仍然是较为合理的。

6.2 推荐用户喜欢的电影

用户 id 为 90 的用户对应的频率最高的两个标签为“感人”和“励志”，为其推荐的电影如表 5 所示，

推荐电影	简介
阿甘正传	描绘了先天智障的小镇男孩福瑞斯特·甘自强不息，最终“傻人有傻福”地得到上天眷顾，在多个领域创造奇迹的励志故事。
风口青春	该片讲述了主人公在一个曾经分离又重逢的初恋情人和一个陪他走过创业最艰难时期、温柔贤惠的职场丽人之间感情抉择的故事。
璀璨的婚礼	该片讲述保罗和刘纯恩在家人和朋友的帮助下克服困难，选择以一场简单而温馨的婚礼阐释对爱的定义。
雷锋的微笑	该片从毛主席回忆雷锋生前一张张可敬可爱的微笑照片开始，讲述雷锋从一名普通战士成长为全军全国人民共同学习的榜样历程。
纯真时代	本片以 1398 年发生在朝鲜李氏王朝的"戊寅靖社"事件为背景，讲述了三个男人的欲望和野心。

表 5：为用户 id 为 90 的用户推荐的电影

可以看出，推荐的五部电影均为励志或者感人电影，与用户喜好高度相符，推荐结果是合理的。

7、 总结与建议

本次实验，我们通过对豆瓣电影和用户数据进行可视化分析和模型建立，实现了豆瓣电影个性化推荐。对比传统推荐算法和深度学习算法，我们得出深度学习算法在豆瓣电影个性化推荐这一任务中更有优势。我们的数据集和模型仍然可以进一步优化，例如，在今后的研究中，可以增加爬取豆瓣电影内容简介信息和用户短评信息，将其加入到我们的文本卷积神经网络中，以获得更为精准的推荐结果。

参考文献

[1] Kim Y . Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.

[2] S. Rendle, Factorization machines, In ICDM, 2010.

[3] Funk, Simon, https://sifter.org/~simon/journal/20061211.html

[4] Noack, Andreas, Modularity clustering is force-directed layout, 2009.

[5] Broder, A.Z., et al., Graph structure in the Web. Comput. Networks, 2000. 33(1-6): p. 309-320.