电影评论分析

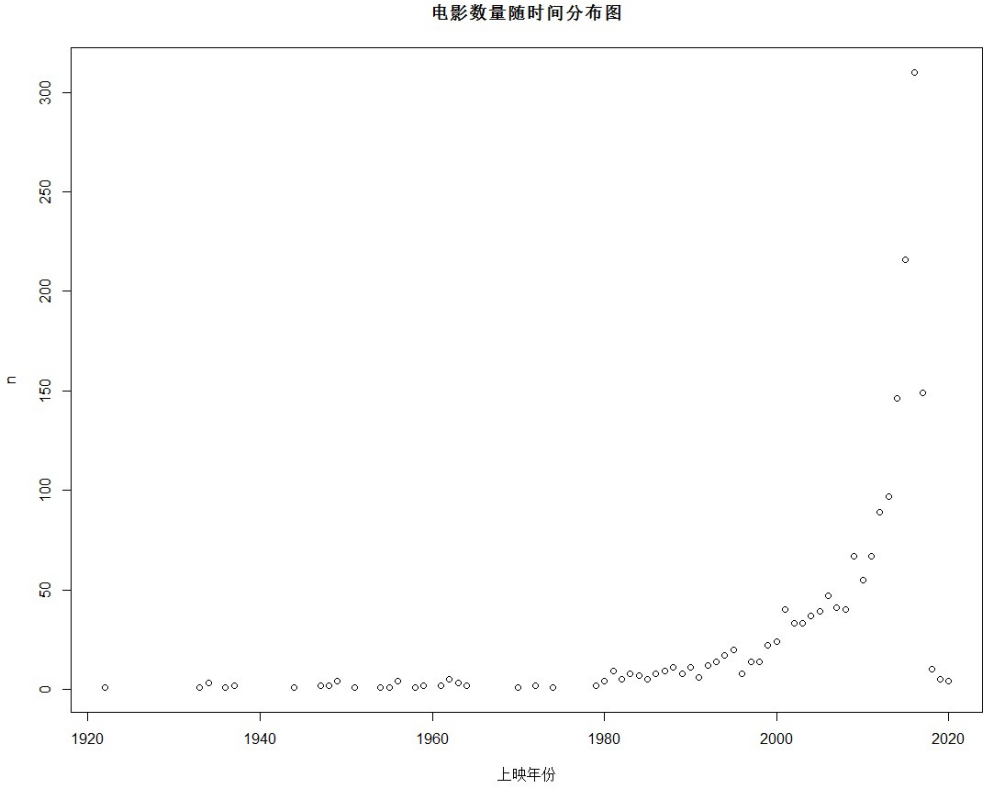
2017 数据科学训练营作业 第三题 陈轶伦 数理统计

1.相关分析

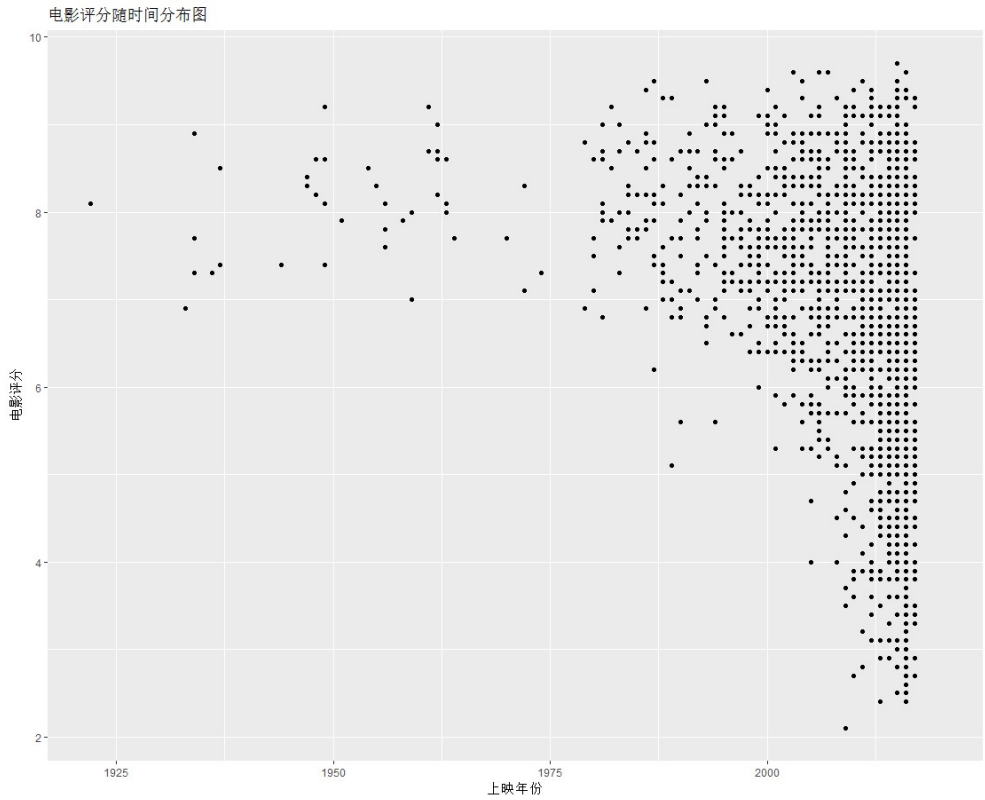
数据集是豆瓣上抓取的中国电影。时间跨度从 1922 年至 2020 年，部分电影未上映，但在豆瓣上也有人给出了评价。这部分主要关注数据集中的电影数目，上映时间，评论数目，电影评分这几个变量的关系。

1.1 时间因素

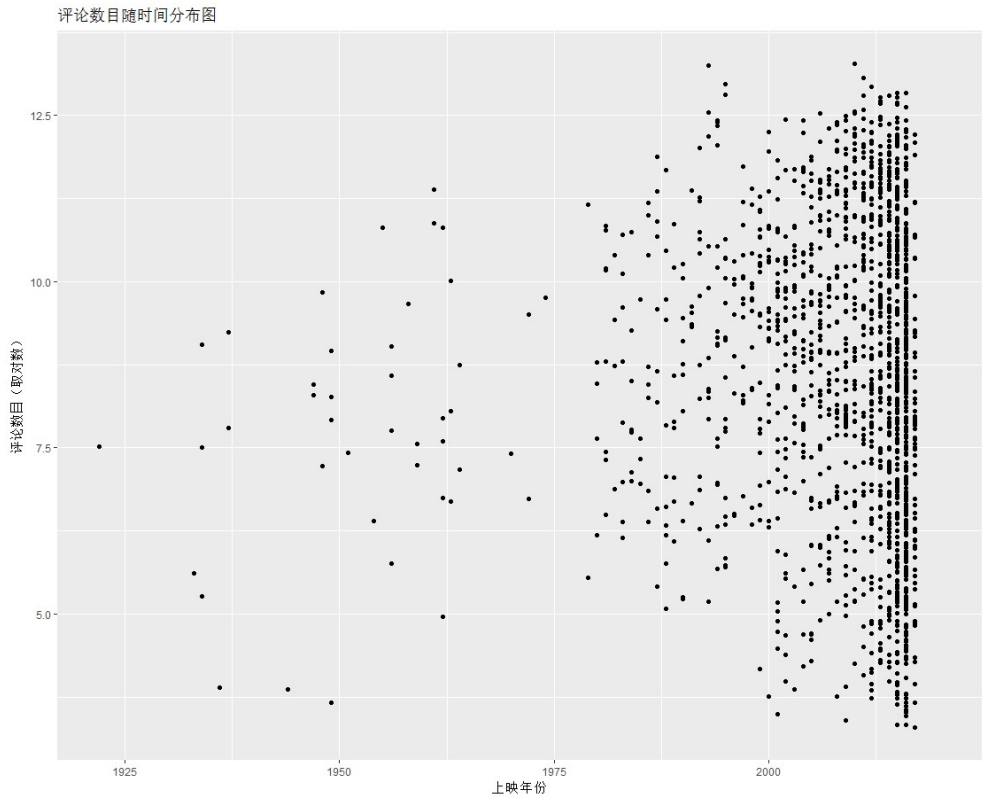
总体来说，中国的电影行业发展态势迅猛。从下图可以看出，除去部分未上映就放出消息的电影，自 2000 年以来，中国电影每年的产量都持续上升，并在近十年间有了指数型的爆发式增长，虽然这一发展态势无法一直保持，但可以预计接下来的电影行业仍然是一个热门产业。



豆瓣电影网站自上线以来，一直是在中国人们最权威的电影评价平台。从下图电影评分随时间的变化图可以看出，2000 年左右以及之前的电影，人们的评价普遍较好。原因可能是因为电影早期数量少质量高，另外人们看电影的成本高，对看电影一事多半伴随着较好的体验。而近十几年来，随着电影的门槛不断降低，越来越多的电影进入了市场，人们看电影也变得更为频繁。不断的观影体验提高了人们对电影质量的要求，而还未成熟的电影环境使得许多烂片进入人们的视野，在此影响下电影评分的跨度便越来越大。

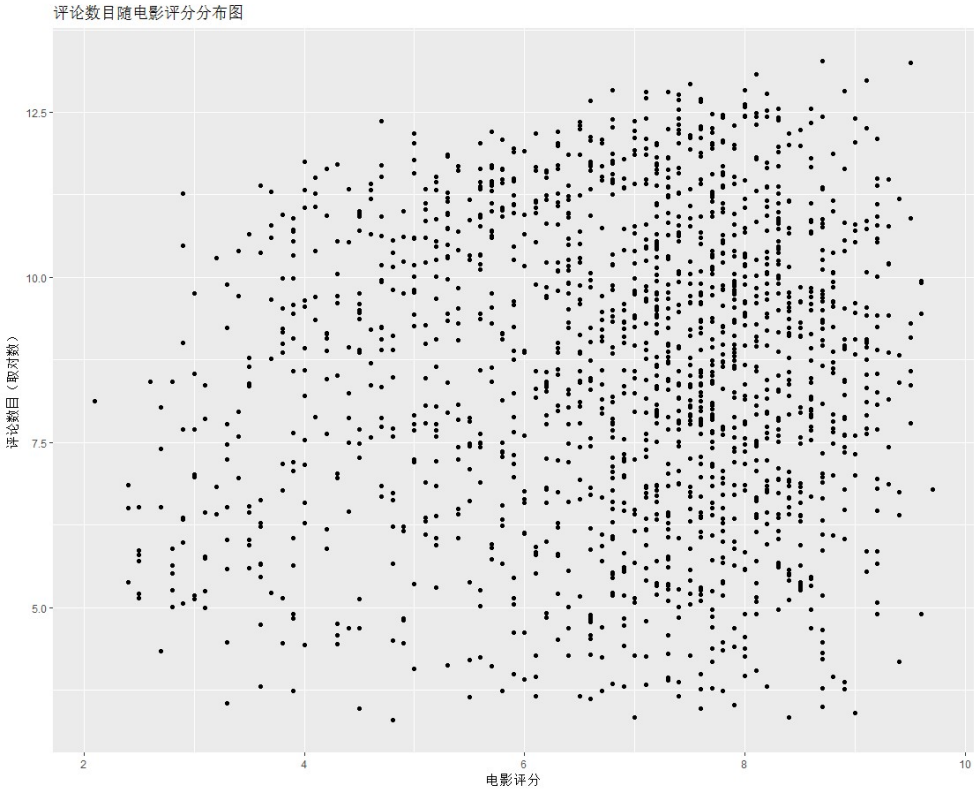


此外，通过下图评论数目随时间的变化图我们也可以看出，年代较为久远的电影因为影片质量落后，播放的平台少，人们对其的评价也就不好，而到了近三十年来，人们对电影的评论近似是对数正态分布，是一个右偏的分布。



1.2 热门程度

目前社会上存在一种说法是一个好的电影会获得社会更高的认可度，更高的认可度会反应在这不电影的热门程度，也就是评论数目上。针对这一说法，通过下图散点图的绘制可以看出并非如此。可以看出，在低分电影中，评论数会稍微低一些，但是这也有可能是因为数量少的原因，在高分电影中，评论数目可以认为是均匀分布。对于差的电影，人们一样会在上面发表负面评论，同时因为目前存在幕后团队刷评论数目和评分的因素，因此电影评分与评论数目的关系，在豆瓣平台上，并没有什么显著关系。

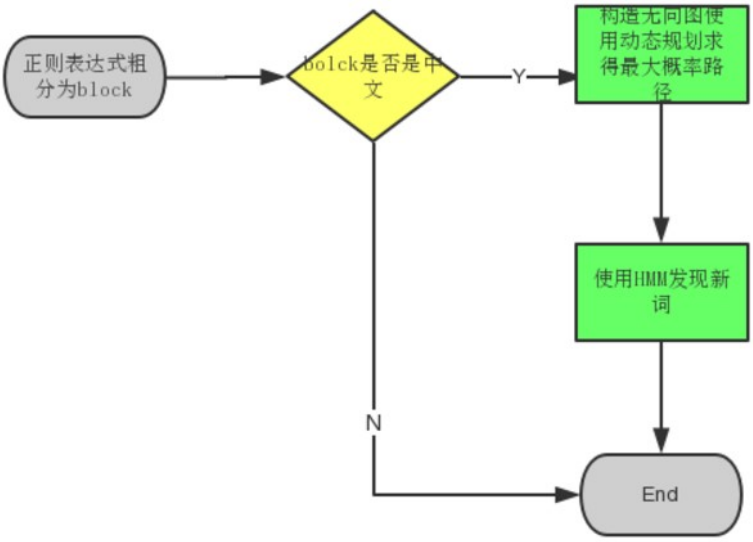


2.文本分析

该部分侧重数据集中的电影类型以及剧情简介两部分，希望借助模型对各个剧情简介通过非监督和监督的方式提取出一些有用的信息。

2.1 关键词

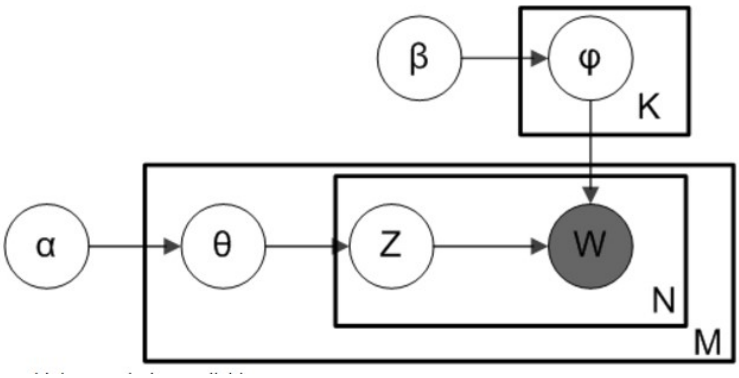
首先需要对剧情简介进行分析，本文使用的分词方式原理如下图，既先使用正则表达式初略分词，然后通过 DAG 模型对已经在词库中的词区分，对于新词则使用 HMM 模型估计。



分词结束后使用修改后的停用词词典进行去停用词，统计词频后绘制了相应的高频词云：

2.2 主题模型

在获得词频的基础上，建立 BOW 然后构造语料库，通过非监督学习的 LDA 模型，使用自适应的浓度参数 alpha 和 gibbs 抽样估计计算语料库中的隐藏主题分布。LDA 算法的图模型如下。



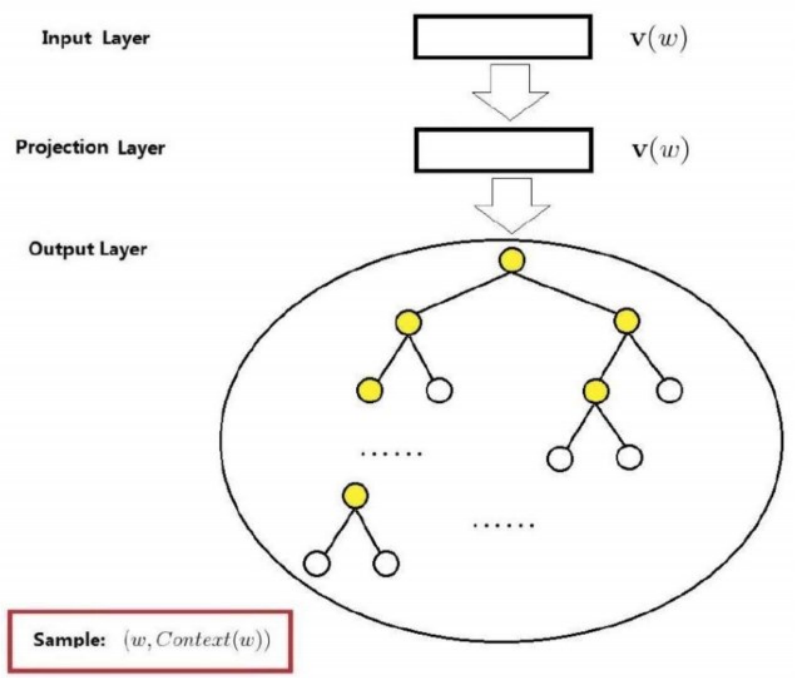
alpha 为语料库 DN 生成第 M 篇文章主题参数 theta，theta 为第 N 个位置生成主题 Z，beta 为单词 DN 生成主题参数 phi，结合 Z 和 phi 生成单词 W。
模型运行的结果，alpha 取 0.1，生成的部分主题 theta 组合如下。

主题一	生活	工作	神秘	意外	爱情
主题二	故事	城市	人生	爱情	一场
主题三	母亲	父亲	家庭	孩子	妻子
主题四	发现	故事	生活	历史	情感
主题五	纪录片	故事	栏目	讲述	拍摄

可以看到，主题一与探险类题材相关，主题二更像是都市爱情剧，主题三家庭类无疑，主题四似乎是古装戏，主题五为纪录片性质。

2.3 词向量

在获得了语料库的基础上，使用 skip-gram 和 hierarchical softmax 建立神经网络计算词向量。本文计算的向量维度为 100，计算 word2vec 使用的方法原理图如下。假定词向量已知，通过神经网络全连接并通过 softmax 映射出一列向量，该向量与 one-hot 编码对应从而进行监督学习，计算梯度时使用 hierarchical softmax 进行估计，并反向训练出词向量以及全连接层的权重。

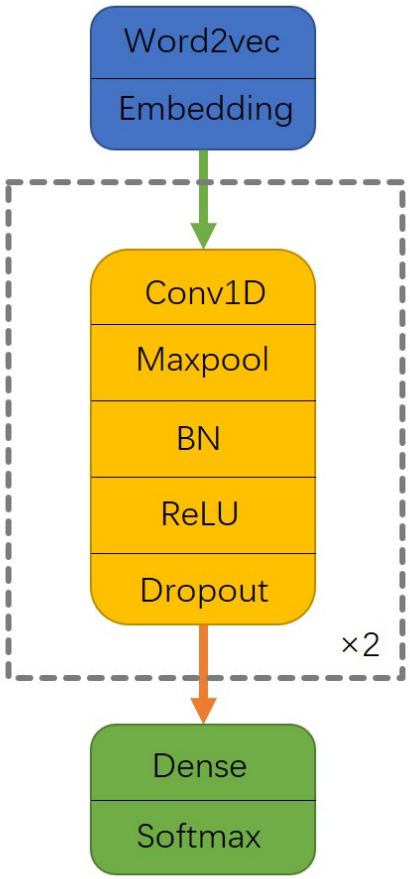


2.4 模型预测

在获得词向量的基础上，希望使用电影简介对电影类型进行预测。数据集中一共有 32 个电影类型，如喜剧，古装，冒险，悬疑等。每部电影有不止一种类型，部分电影缺失电影简介或者电影类型。
在模型预处理上，先将含有缺失信息的 10 来部电影删除，留下 1700 多个完整的电影数据，同时对每个电影的类型进行拆分，比如某一部电影类型为剧情，爱情，喜剧。则复制相应份数的电影简介，每一个电影简介对应某一个电影类型并使用 one-hot 编码。同时因为不同类型的数目不均衡将对训练造成影响，故用重抽样的办法使每个类型数量相等，然后再补全电影的类型。重抽样后样本数量为 30000 左右。
同时需要填充词向量，每个词的词向量维度为 100，1700 多部电影中，因为电影简介最长的词有 134 个，因此将词向量按简介的顺序叠成一个 134*100 的矩阵，空余部分用 0 填补。因此输入数据为 30000*134*100，输出数据为 30000*32。因为输出数据的编码中含有多个 1，倘若根据编码的唯一性做类，分类数目将超过 1000，效果不好。若直接使用输出数据编码，考虑到无法让监督模型正确的区分到底属于哪一类，因此在模型训练上使用 mae 作为代价函数，训练三个 NN 模型效果如下

Model	Detail	MAE
CNN	filter=100, kernel=3, dropout=0.5	0.0365
LSTM	dim=100, dropout=0.5	0.0313
CNN+LSTM	filter=100, kernel=3, dim=40, dropout=0.5	0.0215
CNN+LSTM+BN	filter_1=100, filter_2=60, kernel=3, dim=40, dropout=0.5	0.0192

另外，倘若对于每一个电影，随机的选取其中一个类型作为其唯一的因变量，在损失一定信息的情况下，NN 模型就可以使用 categorical-crossentropy 作为代价函数对类别进行分类。经过尝试，采取如下框架。



Info	Value
Shuffle	True
Early Stop	True
GPU	True
Cross Validation	0.7
Train vs Test	0.66
Epochs	104
Batch Size	32
Filter	100
Kernel	3
Maxpool	3
Dropout	0.5
Rate	1e-4
Decay	1e-6
Training Time	832s
Train Accuracy	0.98
Test Accuracy	0.93

在 NN 框架中，采取了两个卷积层和一个全连接层，在卷积层中为了避免过拟合而采取了正交化处理和随机断开，在训练过程中为了防止过拟合，一旦交叉验证的准确率在几个循环后不再增加，训练便会提前停止。本 NN 框架中一共进行了 104 次循环，历时约 14 分钟，利用电影简介信息训练机器区分 32 类电影类型的正确率可达 93%。

但是，该模型还存在很多不足。问题一，在因变量的处理上为了方便而忽略了多类型的可能性。问题二，为了平衡因变量类型而进行的重抽样使得模型有很大的过拟合风险。改进一，修改 NN 模型中的损失函数使其适应多类型的因变量输出，本文尝试的 MAE 办法在预测效果上并不佳，而且无法告知预测结果应该有多少类型；通过嵌套模型的办法将训练得出的因变量作为新模型比如 GMM 的自变量继续训练。改进二，利用 SMOTE 等办法进行抽样，但方法是否好依赖于词向量的构造是否合理。也可以采用聚类方法切割出均衡样本，分别训练以后用 boost 方法将预测结果集成起来。当然最好的办法就是获得更多的数据，可以将中文转成英文然后和外国电影数据集一起训练。

3.参考资料

Jieba 分词原理：http://midday.me/article/003023dc3f814bc493b37c50b2a9ee71
LDA 模型：http://blog.csdn.net/v_july_v/article/details/41209515
Word2vec 数学原理：http://blog.csdn.net/itplus/article/details/37969519
DL 学习：http://www.deeplearningbook.org/
NN 框架参考：https://arxiv.org/abs/1512.03385