电影推荐

1 问题阐述

提供电影打分数据，根据打分数据，设计一套完整的推荐算法，通过十折交叉验证给出算法的预测准确性：

表 1　数据集描述

数据集类型	推荐系统
训练集用户数	6040
训练集电影数	3952
每个用户打分的电影数量	>20
有无缺失值	有

任务为：基于有监督学习或者矩阵分解算法进行推荐。统一采用基于打分数据的十折交叉验证进行效果评估，评估指标可以使用均方误差、ROC 曲线并计算出 AUC 值、或者 PR（Precision-Recall）曲线并计算出 AUPR 值。

2 技术原理

1)　TensorFlow

TensorFlow 是一个采用数据流图（data flow graphs），用于数值计算的开源软件库。节点（Nodes）在图中表示数学操作，图中的线（edges）则表示在节点间相互联系的多维数据数组，即张量（tensor）。Tensor（张量）意味着 N 维数组，Flow（流）意味着基于数据流图的计算，TensorFlow 为张量从流图的一端流动到另一端计算过程。TensorFlow 是将复杂的数据结构传输至人工智能神经网中进行分析和处理过程的系统。

2)　文本卷积网络

本文采用了一个卷积神经网络语句分类的模型，所用模型来源于 Kim Yoon 的论文：Convolutional Neural Networks for Sentence Classification[5]。

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。图像的卷积通常用 2x2、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 3 个，4 个或者 5 个单词。第三层网络是 max pooling 得到一个长向量，最后使用 dropout 做正则化，最终得到电影 Title 的特征。

3)　十折交叉验证

十折交叉验证，即 10-fold cross-validation，用来测试算法的准确性。十折交叉验证是常用的测试方法，其原理如图 1 所示。主要方法是将数据集分成十份，轮流将其中 9 份作为训练数据，1 份作为测试数据，进行试验。每次试验都会得到相应的正确率，通过一次或多次的十折交叉验证，求取均值即可得出算法的准确性。

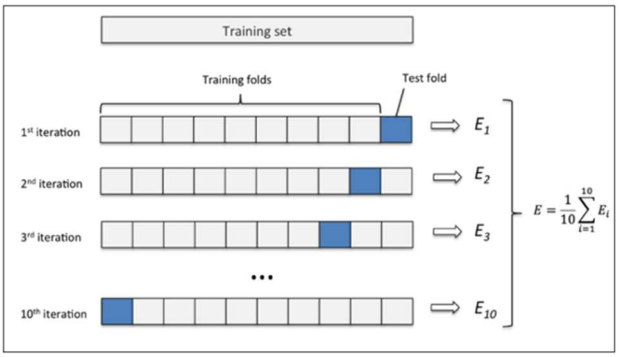

图 1 十折交叉验证

4)　ROC 曲线及 AUC

ROC 曲线[6]是以真阳性率 TPR 为纵坐标，假阳性率 FPR 为横坐标绘制的曲线，能够较容易地查出任意界限值时的对类别的识别能力。其中，TPR = TP/P，FPR = FP/N。ROC 曲线下方的面积即 AUC 是模型准确率的度量。

ROC 曲线越靠近左上角,试验的准确性就越高。最靠近左上角的 ROC 曲线的点是错误最少的最好阈值，其假阳性和假阴性的总数最少。

通常使用 sklearn.metrics 模块中的 roc_curve、auc 结合 matplotlib.pyplot 绘制 ROC 曲线并计算出 AUC 的值。

3 实验方法

1) 加载数据并进行预处理

在 movielens 数据集内，分为三个文件即用户数据 users.dat，电影数据 movies.dat 和评分数据 ratings.dat。

用户数据 user.dat 中，各列信息为 UserID、Gender、Age、Occupation、Zip-code。其中 Gender 以‘M’代表男性，‘F’代表女性，Age 根据年龄段分为<18、18-24、25-34、35-44、45-49、50-55、56+，Occupation 划分成 21 类，包括律师、医生、作家、学生等。

电影数据 movies.dat 中，各列信息分别代表 MovieID、Title、Genres。其中 Genres 是指电影类型，包括动作、喜剧、战争等 18 类。

评分数据 ratings.dat 中，各列信息为 UserID、MovieID、Rating、Timestamp。

首先是加载数据，采用 pandas 模块可以读取 dat 文件格式，该模块还能实现特征和标签的数据分割。

其次对用户数据进行预处理，将 Gender 中的‘F’和‘M’转换为数字‘0’和‘1’。将 Age 中的 7 个年龄段转换成连续的数字即 0，1，2，3，4，5，6。Zip-code 信息用不到，删除。

然后对电影数据进行预处理。通过正则表达式将 Title 中的年份删除，并将 Title 转换为不重复的连续数字。将 Genres 中的字段转换为数字列表，同个电影可能是多种类型的组合。

最后是评分数据，UserID、MovieID、Rating 是有用信息，而 Timestamp 用不到，所以删除该列的数据信息。

2) 训练神经网络

训练神经网络首先是要训练出用户特征和电影特征，以便在推荐系统中可以使用。获取用户数据、电影数据中的信息，根据用户的嵌入向量得到用户特征，根据电影 ID、电影类型、电影名的嵌入向量得到电影的特征。

首先是获取用户的 ID、性别、年龄、职业作为参数传入嵌入层，得到用户 ID 特征、性别特征、年龄特征、职业特征。将各个特征传入全连接层，变换成 1×200 大小，得到用户特征矩阵。

将电影 ID 和电影类型传入嵌入层，得到电影 ID 特征和电影类型特征，再将特征传入全连接层，得到 1×64 的矩阵。将电影名通过文本卷积网络得出 1×32 的电影名特征。最后将电影 ID 和电影类型到的全连接层和电影特征一起传入全连接层，得到 1×200 的电影特征矩阵。

将用户特征和电影特征进行相加求和操作，得出预测评分。通过和真实评分的对比，采用十折交叉验证方法，计算得出均方误差 MSE，采用 MSE 优化损失。

3) 利用训练好的模型进行电影推荐

根据电影类型进行推荐。选择一部电影，根据该部电影的类型推荐相似电影。计算当前电影的特征向量与整个电影特征矩阵的余弦相似度，随机选取同类型电影进行推荐。

根据用户信息进行推荐。指定某一用户，根据用户的特征向量与电影特征矩阵计算所有电影的评分，选择评分最高的电影进行推荐。

根据看过某一电影的几个用户来推荐可能喜欢的电影。首先指定一部电影，根据电影挑选出喜欢该电影的一些用户，得到用户特征矩阵。然后计算这些用户对所有电影的评分，选取每个人评分最高的电影进行推荐。

指定用户和电影进行评分。由于预测评分不一定准确，所以在该模块引入了原数据集的数据。如果在原数据集即 ratings.dat 文件中存在指定用户与电影的打分情况，则输出数据集中的打分。如果原数据集内不存在，则输出预测评分。

4 实验结果

本文采用有监督学习的方法进行电影推荐，并采用基于打分数据的十折交叉验证进行效果评估。对训练网络进行 10 次迭代，采用 MSE 优化损失，记录训练损失和测试损失。

```
Training 100: Batch 3196/3516   train_loss = 0.752
Training 100: Batch 3216/3516   train_loss = 0.575
Training 100: Batch 3236/3516   train_loss = 0.371
Training 100: Batch 3256/3516   train_loss = 0.556
Training 100: Batch 3276/3516   train_loss = 0.491
Training 100: Batch 3296/3516   train_loss = 0.691
Training 100: Batch 3316/3516   train_loss = 0.498
Training 100: Batch 3336/3516   train_loss = 0.533
Training 100: Batch 3356/3516   train_loss = 0.912
Training 100: Batch 3376/3516   train_loss = 0.585
Training 100: Batch 3396/3516   train_loss = 0.641
Training 100: Batch 3416/3516   train_loss = 0.539
Training 100: Batch 3436/3516   train_loss = 0.975
Training 100: Batch 3456/3516   train_loss = 0.700
Training 100: Batch 3476/3516   train_loss = 0.860
Training 100: Batch 3496/3516   train_loss = 0.850
Test 100 : Batch   10/390   test_loss = 0.846
Test 100 : Batch   30/390   test_loss = 0.582
Test 100 : Batch   50/390   test_loss = 0.687
Test 100 : Batch   70/390   test_loss = 0.533
Test 100 : Batch   90/390   test_loss = 0.544
Test 100 : Batch  110/390   test_loss = 0.863
Test 100 : Batch  130/390   test_loss = 0.778
Test 100 : Batch  150/390   test_loss = 1.019
Test 100 : Batch  170/390   test_loss = 0.785
Test 100 : Batch  190/390   test_loss = 0.900
Test 100 : Batch  210/390   test_loss = 0.556
Test 100 : Batch  230/390   test_loss = 0.825
Test 100 : Batch  250/390   test_loss = 0.573
Test 100 : Batch  270/390   test_loss = 0.861
Test 100 : Batch  290/390   test_loss = 0.657
Test 100 : Batch  310/390   test_loss = 0.708
Test 100 : Batch  330/390   test_loss = 0.588
Test 100 : Batch  350/390   test_loss = 0.797
Test 100 : Batch  370/390   test_loss = 0.812
```

图 2 训练损失和测试损失

如图 2 所示，经过 10 次迭代调整后，均方误差 MSE 基本低于 1.0。最后得到的测试损失 MSE 均值为 0.733。

图 3 是经过 351600 步后的训练损失变化图，最后在较低范围 0.8 左右波动。图 4 是十折交叉验证的测试损失变化图，可以看出，测试损失图有 10 个波峰，分别对应十折交叉验证的十次验证过程。随着模型训练步数的增加，测试损失的均方误差不断下降，最后都在 1.0 左右波动或者小于 1.0。

图 5 是 ROC 曲线及 AUC 值的计算结果。其中，AUC 值即 ROC 曲线下方的面积为 0.49。

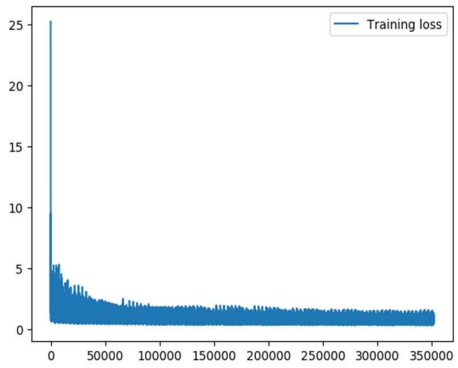

图 3 训练损失 MSE

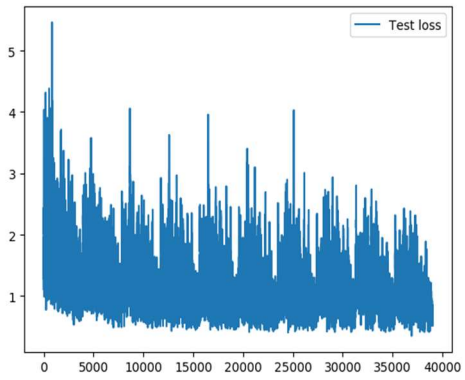

图 4 测试损失 MSE

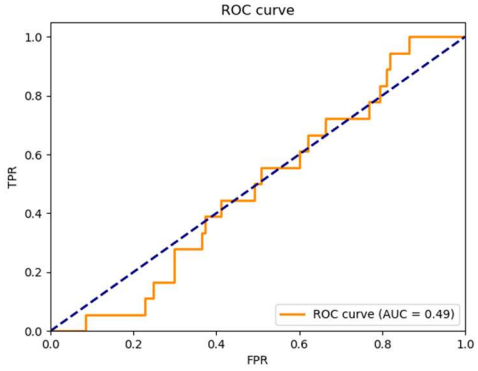

图 5 ROC 曲线及 AUC

基于可视化交互功能的设计，制作了推荐系统的 UI 界面，以实现多次推荐的功能，界面如图 6 所示。

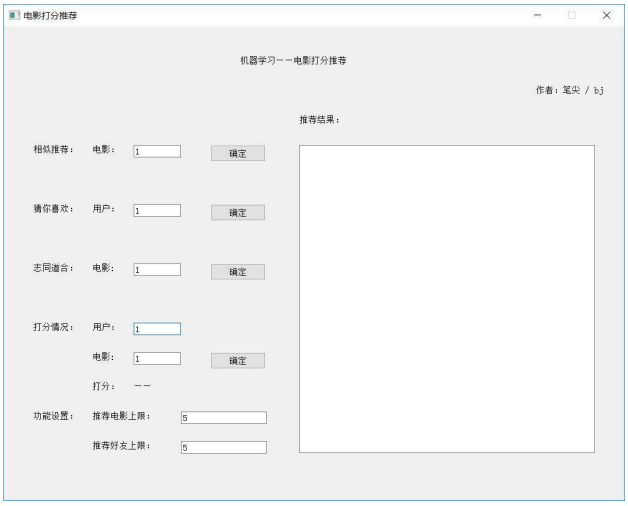

图 6 电影打分推荐系统 UI 界面

根据电影类型进行推荐，即“相似推荐”功能。以第一部电影为例，推荐电影上限设置为 3，即推荐 3 部相似类型电影，结果如图 7 所示。

图 7 相似推荐功能

根据用户推荐电影，即“猜你喜欢”功能模块。任选用户 ID，以用户 20 和用户为例，根据他们的用户信息进行电影推荐，如图 8 和图 9 所示。

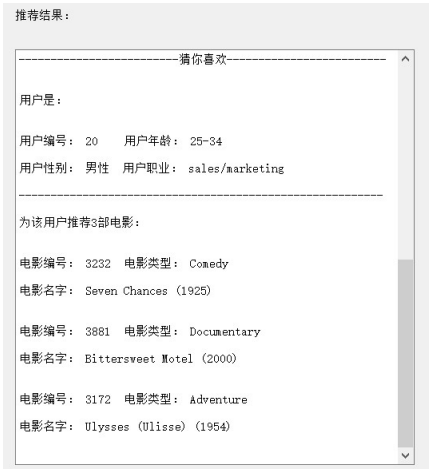

图 8 为用户 20 推荐电影

图 9 为用户 1000 推荐电影

根据看过某一电影的几个用户来推荐可能喜欢的电影，同时推荐这些用户作为好友推荐，即“志同道合”功能模块。以电影 2 为例，设置推荐电影上限为 5，推荐好友上限为 5，推荐好友和推荐电影的结果分别如图 10 和图 11 所示。

图 10 志同道合--推荐好友

图 11 志同道合--推荐电影

指定用户和电影进行评分。图 12 的方框区域中，选择用户 1、电影 1193 进行评分，由于是原数据集中已有数据，故输出对应评分即 5。图 13 的方框区域中，选择用户 1、电影 10 进行评分，由于原数据集中没有该项，所以输出预测评分 3。

图 12 评分(用户 1、电影 1193)

图 13 评分(用户 1，电影 10)