

基于深度学习和矩阵分解的推荐算法

张玉瑶,程学林,尹天鹤
(浙江大学 软件学院,浙江 杭州 310058)

摘要:推荐系统可以有效地缓解信息过载,帮助用户从海量数据中筛选其偏好的内容。目前应用最普遍的推荐算法,如协同过滤,普遍存在着数据稀疏、冷启动、特征提取不充分等问题。把电影推荐作为研究对象,提出了融合深度学习和矩阵分解的LM-SVD推荐算法。以多层感知机MLP和长短期记忆网络LSTM的组合模型学习用户、电影属性数据及文本数据,获取用户和电影的深层特征表示。接着以BiasSVD矩阵分解模型学习用户电影评分数据,获取用户和电影的潜在隐特征向量,并与深度学习阶段获得的深层特征向量相融合。改进矩阵分解的预测评分和损失函数计算方式,缓解评分矩阵的稀疏性,使得特征提取更充分。在两个MovieLens数据集上进行的算法对比实验表明,LM-SVD算法有效提升了电影评分预测准确度,使得推荐性能提高。

关键词:深度学习;矩阵分解;电影推荐;多层感知机;长短期记忆网络

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2021)07-0021-07

doi:10.3969/j.issn.1673-629X.2021.07.004

A Recommendation Algorithm Based on Deep Learning and Matrix Factorization

ZHANG Yu-yao, CHENG Xue-lin, YIN Tian-he
(School of Software, Zhejiang University, Hangzhou 310058, China)

Abstract: Recommendation systems can effectively alleviate information overload and help users filter their preferred content from vast amounts of data. At present, the most common recommendation algorithms, such as collaborative filtering, generally have some problems, such as sparse data, cold starting and insufficient feature extraction. Taking movie recommendation as the research object, the LM-SVD recommendation algorithm is proposed by integrating deep learning and matrix factorization. The combined model of MLP and LSTM is used to learn user, movie attribute data and text data, so as to obtain the deep feature representation of user and movie. Then the BiasSVD matrix decomposition model is used to learn the user's movie rating data, to obtain the latent implicit vectors of the user and the movie, and to integrate with the deep feature vectors obtained in the deep learning stage. The prediction scoring and loss function calculation methods of matrix decomposition are improved to alleviate the sparsity of scoring matrix and make feature extraction more sufficient. Algorithm comparison experiments on two MovieLens data sets show that the LM-SVD algorithm effectively improves the prediction accuracy of movie scores and improves the recommendation performance.

Key words: deep learning; matrix factorization; movie recommendation; multilayer perceptron; long short-term memory

0 引言

随着大数据时代的到来,每时每刻都有海量数据被制造和发布到网络上。为了应对数据量的激增导致的“信息过载”问题,推荐系统应运而生。在电影推荐领域,用户可以方便地通过移动终端接触各式各样的电影,对电影的选择越来越依赖于电影评分和智能化电影推荐^[1]。推荐系统的核心在于推荐算法,推荐算法的准确度直接决定着推荐系统的效果。

协同过滤推荐算法能够充分利用用户行为数据,不受物品内容的限制,成为应用最普遍的推荐算法之一^[2]。矩阵分解(matrix factorization, MF)是一种基于模型的协同过滤算法,模型简单且擅长评分预测,在电影推荐等领域得到了广泛的应用。文献[3]提出的隐语义模型(latent factor model, LFM)是一种典型的矩阵分解,引入用户和物品评分隐因子,之后该算法又被进一步优化。文献[4]将用户隐式反馈信息及用户和

收稿日期:2020-08-02

修回日期:2020-12-04

基金项目:2020年中央教改-浙江大学“十三五”教改项目第二批一期(zdjg19072);浙江省基础公益研究计划项目(2014C31076)

作者简介:张玉瑶(1996-),女,硕士研究生,研究方向为深度学习、推荐算法;程学林,博士,高级工程师,CCF会员(C1107M),研究方向为数据可视分析、软件工程。

项目偏好随时间的变化纳入考虑,提出 TimeSVD++算法,使得推荐准确性进一步提升。然而,矩阵分解过度依赖于用户对物品的行为数据,导致了数据稀疏性、冷启动以及特征提取不充分等问题^[5],对推荐准确度造成了严重影响。

随着人工智能浪潮的兴起,一些学者开始将深度学习学习方法运用到推荐算法研究中,文献[6]对深度学习在推荐算法中的应用进行了综述。文献[7]提出了深广学习模型(wide & deep learning),能够对包括项目特征、情景特征在内的多源异构数据加以综合利用,使得训练得到的模型同时获得记忆和泛化能力。文献[8]提出一种基于长短期记忆网络 LSTM 的上下文感知模型,该模型能够捕捉用户的长短期兴趣偏好,使得个性化上下文感知引用推荐成为可能。也有一些研究将深度学习算法和协同过滤融合起来,构建混合推荐模型。将深度学习算法提取到的特征作为辅助信息,有利于缓解数据稀疏问题,提高推荐准确度。文献[9]提出 CNN-LFM 模型,利用隐语义模型 LFM 挖掘评分数据,获得用户和项目的潜在特征,其中项目的潜在特征会在卷积神经网络(convolutional neural networks, CNN)提取的图像特征的约束下不断完善,很好地提高了推荐的性能。文献[10]提出了 ConvMF 卷积矩阵分解模型,将 CNN 集成到概率矩阵分解(probabilistic matrix factorization, PMF)中,通过融入项目文本特征提升推荐准确度。然而,现有的混合算法在特征提取上比较单一,往往忽视了用户特征,对项目特征的挖掘则集中于文本、图像等某一方面,并不能充分体现项目特征。

该文将电影推荐作为研究对象,提出融合深度学习和矩阵分解的 LM-SVD 推荐算法。利用深度学习长于特征提取和矩阵分解善于预测评分的优点,提高电影评分预测准确度。首先以多层感知机(multilayer perceptron, MLP)和长短期记忆网络(long short-term memory, LSTM)的组合模型学习用户、电影属性数据及文本数据,以获取用户和电影的深层特征表示。接着以 BiasSVD 矩阵分解模型学习用户电影评分数据,获取用户、电影的潜在隐向量,并与深度学习阶段获得的深层向量相融合。最后结合评分偏置信息进行电影评分预测,改进 SVD 的预测评分和损失函数计算方式,缓解评分矩阵的数据稀疏性,使得特征提取更加充分。

该算法的特点在于:(1)改进了矩阵分解推荐算法的评分预测方式与损失函数。通过深度学习挖掘用户和电影的深层次特征,融入矩阵分解模型中,提升电影评分预测准确度。(2)通过 MLP 学习用户、电影属性特征, LSTM 学习电影文本特征, BiasSVD 矩阵分解

学习用户电影潜在特征,对用户特征和电影特征进行了深入挖掘,充分利用了辅助信息。即使当用户电影评分数据较为稀疏时, LM-SVD 依然可以准确地进行预测。

1 相关工作

1.1 多层感知机

多层感知机包含输入层、隐藏层和输出层,又称人工神经网络。除了输入输出层,中间隐藏层的数量可以根据需求任意添加。MLP 不同层之间是全连接的,即上一层的任意一个神经元与下一层的所有神经元之间都存在连接。

1.2 长短期记忆网络

长短期记忆网络^[11]是循环神经网络(recurrent neural network, RNN)^[12]的一个重要变种。RNN 很擅长进行序列化建模,而 LSTM 对 RNN 单元的结构进行了修改,通过设计门结构和记忆单元来解决梯度爆炸或梯度消失问题,能够对输入长度较长的时间序列问题进行建模。每一个时刻 t 的 LSTM 单元由输入门 i_t 、遗忘门 f_t 、输出门 o_t 、细胞记忆单元 c_t 和隐层状态 h_t 组成。

遗忘门 f_t 用来控制从细胞记忆单元中丢弃信息,决定上一时刻历史信息的流入量,见式(1)。

$$f_t = \text{sigmoid}(W_{if}X_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \quad (1)$$

其中, h_{t-1} 表示上一个时刻的输出, X_t 表示当前时刻的输入。

输入门 i_t 用于控制当前时刻信息的流入。首先,以一个 sigmoid 层决定哪些信息需要更新,见式(2)。接着,以一个 tanh 层生成当前时刻信息的备选更新值 g_t ,由当前时刻的输入值 X_t 和上一时刻的输出值 h_{t-1} 所决定,见式(3)。最后,将这两部分结合起来,实现对细胞记忆单元的更新,将 c_{t-1} 更新为 c_t ,见式(4)。

$$i_t = \text{sigmoid}(W_{ii}X_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \quad (2)$$

$$g_t = \tanh(W_{ig}X_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \quad (3)$$

$$c_t = f_t \times c_{t-1} + i_t \times g_t \quad (4)$$

输出门 o_t 主要用于控制当前时刻信息的流出,以决定最终输出的值。如式(5)所示,先通过 sigmoid 层来决定输出细胞记忆单元的哪些部分。

$$o_t = \text{sigmoid}(W_{io}X_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \quad (5)$$

接着,把细胞记忆单元 c_t 通过 tanh 进行处理,并将它和 o_t 相乘筛选出最终的输出结果,见式(6)。

$$h_t = o_t \times \tanh(c_t) \quad (6)$$

以上公式解释了信息在 LSTM 隐藏层的处理过程。LSTM 通过设置三道控制门实现了信息选择,可以很好地对长时间序列进行建模。

1.3 加入评分偏置项的矩阵分解

矩阵分解是一种基于模型的协同过滤算法^[13],在推荐领域应用广泛。该文采用的矩阵分解算法是求解用户、物品潜在特征隐向量,考虑评分偏置的 BiasSVD 算法^[3]。

设 R 代表评分矩阵,现有 m 个用户, n 个物品。在矩阵分解阶段,寻找用户特征矩阵 P 和物品特征矩阵 Q ,使得两矩阵的乘积近似等于 R 。 k 表示用户隐向量和物品隐向量的维度:

$$R_{m \times n} = P_{m \times k}^T \cdot Q_{k \times n} \approx R \quad (7)$$

用户潜在特征隐向量 p_u ,表示用户 u 的兴趣;物品潜在特征隐向量 q_i ,表示物品 i 的特点。两特征向量的内积代表用户 u 对物品 i 的预测评分 r_{ui} :

$$r_{ui} = p_u \times q_i^T + \mu + b_u + b_i \quad (8)$$

在实际应用中,用户对物品的评分可能存在偏见,因此在预测评分中加入评分偏置项,其中 μ 为整体评分偏差,是所有评分的全局平均值; b_u 为用户评分偏置, b_i 为物品评分偏置。

算法的目的是最小化真实评分 r_{ui} 与预测评分 r_{ui} 之间的误差,再加入防止过拟合的正则化项,模型的损失函数为:

$$\min; \text{Loss} = \sum_{r_{ui} \neq 0} (r_{ui} - r_{ui})^2 + \lambda (\|p_u\|^2 + \|q_i\|^2 + \|b_u\|^2 + \|b_i\|^2) \quad (9)$$

2 LM-SVD 推荐算法

2.1 算法设计思想

矩阵分解具有易于建模、擅长评分预测的特点,是

目前应用最为广泛的电影推荐算法之一。然而,矩阵分解电影推荐算法存在着以下问题:

(1)数据稀疏性问题是用户电影评分矩阵是极为稀疏的。用户只对极少部分的电影做出过评分等行为,稀疏的评分矩阵使得矩阵分解的效果不佳。

(2)冷启动问题是一种极端情况下的数据稀疏性问题。当一个新用户或者一部新电影刚刚进入系统时,由于还没有产生用户行为数据,也就无法做出推荐。

(3)特征提取不充分问题是指矩阵分解只利用了用户行为数据,而没有将用户属性数据和电影属性数据考虑在内。这些辅助信息中很可能蕴含着用户和电影的重要特征。

为应对上述问题,该文将深度学习和矩阵分解算法融合设计混合推荐算法 LM-SVD,使得电影评分预测准确度提高,从而提升推荐质量。首先利用多层感知机(MLP)学习用户、电影属性特征,利用长短期记忆网络(LSTM)学习电影文本特征,获取用户和电影的深层次特征向量。之后将习得的深层特征向量融入 BiasSVD 矩阵分解模型中,改进 BiasSVD 的评分预测方式和损失函数,分解用户电影评分矩阵从而预测评分。将深度学习提取到的特征作为辅助信息,可以有效地缓解用户电影评分矩阵的数据稀疏性和冷启动问题。

2.2 LM-SVD 模型结构

该文提出融合深度学习算法和 BiasSVD 矩阵分解模型的 LM-SVD 推荐算法。算法整体架构如图 1 所示。

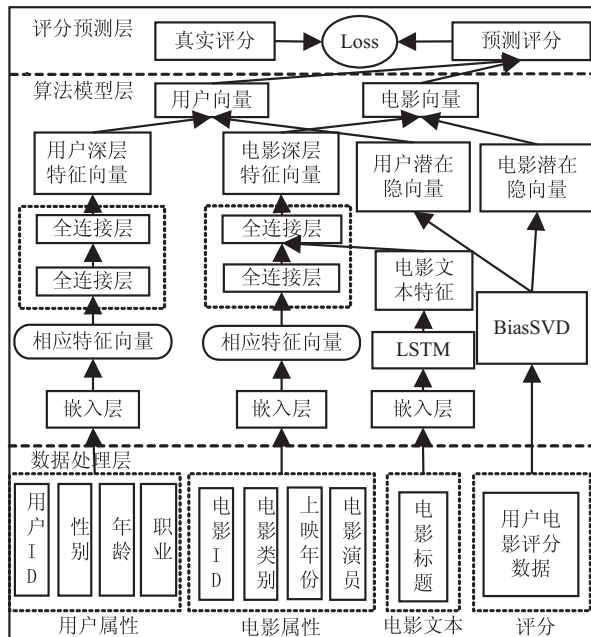


图 1 LM-SVD 算法模型

算法整体分为数据处理层、算法模型层和评分预测层。数据处理层进行数据预处理和特征工程。其中用户属性、电影属性及电影文本数据作为辅助信息,用于深度学习模型训练;用户电影评分行为数据用于矩阵分解模型训练。

算法模型层包含深度学习模块和矩阵分解模块。深度学习模块包含两个并行的网络 Net_u 和 Net_i 。 Net_u 负责通过 MLP 学习用户属性数据,输出用户深层特征向量 \mathbf{P}_u ; Net_i 首先以 MLP 学习电影属性数据,并通过 LSTM 从电影标题数据中学习电影文本特征,之后输出电影深层特征向量 \mathbf{Q}_i 。BiasSVD 矩阵分解模块负责从用户电影评分矩阵中分解出用户潜在隐向量 \mathbf{p}_u 和电影潜在隐向量 \mathbf{q}_i ,这属于潜在特征向量。将深度学习得到的深层特征向量 \mathbf{P}_u 和 \mathbf{Q}_i 与矩阵分解得到的潜在特征向量 \mathbf{p}_u 和 \mathbf{q}_i 进行融合,并结合用户评分偏置,对 SVD 算法的损失函数进行改进。

评分预测层通过最小化损失函数训练模型。可利用训练好的模型进行评分预测,从而为每个用户形成其电影推荐列表。

2.3 深度学习提取深层特征向量

(1) 学习用户深层特征向量。

将用户属性数据输入到模型中以获取用户深层特征向量。首先将用户属性输入到嵌入层,从而将各特征转化为稠密向量。设经过嵌入层后用户属性特征集为 $\mathbf{x}; \{x_1, x_2, x_3, x_4\}$, 用户 ID、年龄、性别、职业特征向量的维度均为 8。之后将各用户特征向量输入到多层感知机 MLP, MLP 包含两层全连接层。如式(10)所示,经过第一层全连接层后,得到用户属性特征向量 $\bar{\mathbf{x}}$ 。第一层包含 32 个神经元,激活函数为 relu。如式(11)所示,利用 concatenate(\cdot) 函数将用户各个属性特征进行融合,输入第二层全连接层,最终输出用户深层特征向量 \mathbf{P}_u 。 \mathbf{P}_u 的维度为 k , 是待确定的参数。第二层包含 k 个神经元,激活函数为 tanh。

$$\bar{\mathbf{x}} = \text{relu}(w_1 \cdot \mathbf{x} + b_1) \quad (10)$$

$$\mathbf{P}_u = \tanh(w_2 \cdot \text{concatenate}(\bar{\mathbf{x}}) + b_2) \quad (11)$$

式(10)、(11)中, w_1 和 w_2 表示全连接层权重, b_1 和 b_2 表示偏置。

(2) 学习电影深层特征向量

将电影属性和电影文本数据输入到模型中以获取电影深层特征向量。首先将电影属性输入到嵌入层,从而将各特征转化为稠密向量。经过嵌入层后,电影 ID、电影类别、上映年份、电影演员特征向量的维度均为 8。由于每部电影有多名演员,需要对每部电影对应的多个演员向量做加和,以代表该部电影的演员向量,类别特征同理。

将电影标题文本数据输入到 LSTM 模型中。这部分模型包括嵌入层、LSTM 层和 dropout 层。嵌入层将电影标题文档转换为词向量矩阵。在数据预处理阶段,所有电影标题句都被映射为长度为 15 的等长数字列表,对应 15 个单词,标题长度不足 15 的用填充符所对应的数字补齐。则每个电影标题通过嵌入层后,都会被编码为形如 $\mathbf{D} = \{a_1, a_2, \dots, a_{15}\}$ 的词向量矩阵, a_i 是第 i 个单词的词向量,维数为 8。之后通过一层 LSTM 层处理词向量矩阵。隐藏层神经元的个数为 8, 激活函数采用 tanh。dropout 层以预先设置的概率停止神经元的输出,意味着每次的网络训练只有一部分数据特征在参与,从而达到防止神经网络过拟合的目的。这里 dropout 设置为 0.5。

经过上述处理, LSTM 将电影标题的行文档作为输入,输出每一行标题文本的特征向量 \mathbf{Q}_{il} :

$$\mathbf{Q}_{il} = \text{LSTM}(W, y_i) \quad (12)$$

其中, W 代表所有的权重和偏置,对于每个初始电影标题行文档 y_i , 得到一个维数为 8 的文本特征向量 \mathbf{Q}_{il} 。

将嵌入层输出的各电影属性特征向量输入到 MLP, 包含一层全连接层,共 32 个神经元,激活函数为 relu。设第一个 MLP 输出的电影属性特征向量为 \mathbf{Q}_{i2} , 将 \mathbf{Q}_{i2} 和 LSTM 输出的电影文本特征向量 \mathbf{Q}_{il} 输入到第二个 MLP 中。如式(13)所示,最终输出电影深层特征向量 \mathbf{Q}_i 。 \mathbf{Q}_i 的维度为 k , 是待确定的参数。concatenate(\cdot) 函数用于将 \mathbf{Q}_{il} , \mathbf{Q}_{i2} 两个特征向量进行串联, w 为全连接层权重, b 为偏置。该 MLP 包含一层全连接层,共 k 个神经元,激活函数为 tanh。

$$\mathbf{Q}_i = \tanh(w \cdot \text{concatenate}(\mathbf{Q}_{il}, \mathbf{Q}_{i2}) + b) \quad (13)$$

2.4 添加深层特征向量的矩阵分解模型

在 1.3 节中,阐述了加入评分偏置项的 BiasSVD 矩阵分解算法原理和主要计算公式。如图 1 所示, LM-SVD 算法将深度学习模型训练得出的用户深层特征向量 \mathbf{P}_u 和电影深层特征向量 \mathbf{Q}_i 融合到矩阵分解模型中,与矩阵分解获得的用户潜在隐特征向量 \mathbf{p}_u 和电影潜在隐特征向量 \mathbf{q}_i 进行合并,对 BiasSVD 的损失函数进行修改。

现需要将 \mathbf{p}_u 和 \mathbf{P}_u , \mathbf{q}_i 和 \mathbf{Q}_i 分别进行融合。为了融和深层特征向量和潜在特征向量,文献[14]提出将两向量对应维度相加成为一个向量可取得较好的推荐性能。进行向量相加后,用户 u 对电影 i 的预测评分 r_{ui} 计算公式由式(8)修改为式(14)。

$$r_{ui} = (\mathbf{P}_u + \mathbf{p}_u) \times (\mathbf{Q}_i + \mathbf{q}_i)^T + \mu + b_u + b_i \quad (14)$$

其中, \mathbf{p}_u 、 \mathbf{P}_u 、 \mathbf{q}_i 、 \mathbf{Q}_i 均为 k 维向量; μ 为整体评分偏差,是所有评分的全局平均值; b_u 为用户评分偏置, b_i 为电影评分偏置。

LM-SVD 模型的损失函数在式(9)的基础上进行改进,最终如式(15)所示:

$$\min; \text{Loss} = \sum_{r_{ui} \neq 0} (r_{ui} - r_{ui})^2 + \lambda (\| \mathbf{P}_u \|^2 + \| \mathbf{Q}_i \|^2 + \| \mathbf{p}_u \|^2 + \| \mathbf{q}_i \|^2 + \| \mathbf{b}_u \|^2 + \| \mathbf{b}_i \|^2) \quad (15)$$

2.5 算法流程

LM-SVD 算法训练和参数优化的过程总结如下,通过最小化损失函数训练模型。

算法:LM-SVD 推荐算法。

输入:用户电影评分矩阵 \mathbf{R} , 用户特征数据 \mathbf{M} , 电影特征数据 \mathbf{N} , 用户和电影特征向量维度为 k 。

步骤 1:划分数据集为训练集、验证集及测试集。

步骤 2:训练提取用户深层特征向量 \mathbf{P}_u 。

(1)用户特征数据预处理,并通过嵌入层转化为稠密向量。

(2)嵌入层后输入 MLP 中,需要确定的参数有激活函数、隐藏层层数、隐藏层神经元数等。训练得 \mathbf{P}_u 。

步骤 3:训练提取电影深层特征向量 \mathbf{Q}_i 。

(1)电影特征数据预处理,并通过嵌入层转化为稠密向量。

(2)嵌入层后输入 MLP 中,需要确定的参数有激活函数、隐藏层层数、隐藏层神经元数等。训练得电影属性特征向量。

(3)将电影标题文本数据输入 LSTM 中,需要确定的参数有 LSTM 层数、隐藏层神经元数等。训练得电影文本特征向量。

(4)拼接属性特征和文本特征,再通过一层全连接层,得电影深层特征向量 \mathbf{Q}_i 。

步骤 4:将 \mathbf{P}_u , \mathbf{Q}_i 融入 BiasSVD 模型,计算预测评分,最小化损失函数。需要调整的参数有正则化系数 λ 、学习率 α 等。

(1)根据式(14)计算预测评分 r_{ui} 。

(2)根据式(15)计算损失函数 Loss。

(3)根据式(16)计算均方根误差 RMSE。

(4)迭代训练模型直到达到最大迭代次数或者 RMSE 值小于阈值。

3 算法实验

3.1 实验数据集与评价指标

(1)实验数据集。

该文采用的实验数据集是电影推荐研究领域常用的公开数据集 MovieLens-1M 和 MovieLens-100K,来源于 (<https://grouplens.org/datasets/movielens>)。另通过爬虫从 IMDB (<https://www.imdb.com/>) 获取电影上映年份、电影演员等信息对数据集进行补充。补

充之后两个数据集中都包含了用户电影评分数据、电影信息数据以及用户信息数据。两数据集的基本信息如表 1 所示。

表 1 两实验数据集基本信息

数据集	用户数	电影数	评分数
MovieLens-1M	6 040	3 883	1 000 209
MovieLens-100K	943	1 682	100 000

需对原始数据集进行数据预处理,从三个数据集中获取需要的字段,并将各非数值特征字段转换为数字。具体处理方式如下:

电影信息数据需要用到电影 ID、电影类别、上映年份、电影演员、电影标题五个字段。上映年份字段需把各年份区间映射为 0 到 7 的数字。电影类别字段需要将全部的电影类别组成一个字符串到数字的映射字典,之后再每部电影类别字段转换成等长的数字列表。电影标题字段首先创建标题中词汇文本到数字的映射字典,然后将标题句子转换成词汇对应的等长数字列表。电影演员字段的处理方式同电影类别。用户信息数据需要用到用户 ID、性别、年龄、职业四个字段。性别字段需把两种性别字符映射为 0 和 1,年龄字段则要把各年龄区间映射为 0 到 6 的数字。用户电影评分数据需要用到用户 ID、电影 ID、评分三个字段,这三个字段无需特殊的处理。

(2)实验评价指标。

推荐算法的质量主要体现在其推荐准确性。该文主要度量推荐算法预测评分的准确度,以均方根误差 (root mean squared error, RMSE) 为评估指标。RMSE 衡量的是预测评分值与实际评分值之间的偏差。其值越小,则评分预测准确度越高,推荐算法质量越好。RMSE 的计算公式如式(16)所示:

$$\text{RMSE} = \sqrt{\frac{\sum_{r_{ui} \in \text{Test}} (r_{ui} - r_{ui})^2}{|T|}} \quad (16)$$

其中,Test 表示测试集, r_{ui} 表示 Test 中用户 u 对电影 i 的实际评分, r_{ui} 表示 Test 中用户 u 对电影 i 的预测评分, T 为测试集中评分的数量。

3.2 实验及结果分析

3.2.1 参数实验

将数据集随机分为 3 个部分,分别为训练集 (80%)、验证集 (10%) 和测试集 (10%)。采用五折交叉验证法进行参数选择。由于模型涉及的参数较多,这里只展示一些主要参数的调优实验结果。实验关注的参数主要有:用户、电影特征向量维度 k ; 正则化项系数 λ 。

经过反复实验,其余参数设置如下:迭代次数

epoch=10, 学习率 $\alpha=0.001$, 训练迭代的批次样本数量 Batch_Size=256。其他深度学习阶段参数设置在 2.3 节中阐述。

(1) 用户、电影特征向量维度 k 。

如式(14), 深度学习阶段获得的用户、电影深层特征向量 P_u 、 Q_i , 以及矩阵分解阶段获得的用户、电影潜在特征向量 p_u 、 q_i , 维度均为 k 。当其他参数固定时, 调整特征向量维度 k , 取值范围为 [40, 50, 60, 70, 80]。由表 2 可知, 随着 k 的增加, LM-SVD 算法的 RMSE 值在两数据集上均呈现先减小后增大的趋势。当 k 增加时, 模型学习到了更多用户和电影的隐藏特征, 使得预测准确度提高。但当 k 超过一定范围时, 模型的泛化能力有所下降, 使得 RMSE 值增加。在 MovieLens-1M 数据集上, 当 k 为 50 时, 算法的 RMSE 值最小, 为 0.847 9。在 MovieLens-100K 数据集上, 当 k 为 60 时, 取得最小 RMSE 值, 为 0.907 2。

表 2 特征向量维度 k 对算法 RMSE 值的影响

特征向量维度 k	MovieLens-1M	MovieLens-100K
40	0.848 1	0.908 2
50	0.847 9	0.907 6
60	0.848 0	0.907 2
70	0.848 5	0.908 0
80	0.848 6	0.908 4

(2) 正则化项系数 λ 。

正则化项系数 λ 的取值范围为 [0.02, 0.04, 0.06, 0.08, 0.10]。由表 3 可知, 随着正则化项系数 λ 的增加, LM-SVD 算法的 RMSE 值在两数据集上呈现先减小后增大的趋势。当 λ 为 0.04 时, 算法在两数据集上的 RMSE 值均达到最小。MovieLens-1M 的 RMSE 为 0.847 8, MovieLens-100K 的 RMSE 为 0.907。

表 3 正则化项系数 λ 对算法 RMSE 值的影响

正则化项系数 λ	MovieLens-1M	MovieLens-100K
0.02	0.852 6	0.910 6
0.04	0.847 8	0.907 0
0.06	0.848 3	0.908 2
0.08	0.849 8	0.909 2
0.10	0.851 5	0.913 2

根据上述参数实验结果, 在 MovieLens-1M 数据集上, LM-SVD 算法的最优参数组合为: $k=50$, $\lambda=0.04$; 在 MovieLens-100K 数据集上, LM-SVD 算法的最优参数组合为: $k=60$, $\lambda=0.04$ 。

3.2.2 对比实验

为了客观验证文中算法的推荐效果, 将保持最优参数组合条件下的 LM-SVD 算法与表 4 中的 3 个算法进行对比。

表 4 对比算法

对比算法	特点
BiasSVD 算法 ^[3]	加入评分偏置的矩阵分解推荐, 擅长电影评分预测
DN-CBR 算法 ^[15]	深度神经网络推荐, 利用 MLP 和 CNN 提取用户、物品属性和文本特征
ConvMF 算法 ^[10]	深度矩阵分解推荐, 结合 CNN 和概率矩阵分解 PMF, 将物品文本信息融入矩阵分解

接着, 在 MovieLens-1M, MovieLens-100K 数据集上对四种算法的评分预测准确度进行测试。以 RMSE 作为评测指标, 对比结果如表 5 所示。

表 5 对比实验结果

对比算法	MovieLens-1M	MovieLens-100K
DN-CBR	0.870 9	0.936 1
BiasSVD	0.862 5	0.918 7
ConvMF	0.853 1	0.913 2
LM-SVD	0.847 6	0.906 8

如表 5 所示, 文中提出的 LM-SVD 算法在两个数据集上表现出的电影评分预测准确度均优于三种对比算法。在 MovieLens-1M 数据集上, LM-SVD 算法的 RMSE 值比 DN-CBR、BiasSVD、ConvMF 分别下降 2.68%, 1.73%, 0.64%。在 MovieLens-100K 数据集上, LM-SVD 算法的 RMSE 值比 DN-CBR、BiasSVD、ConvMF 分别下降 3.13%, 1.29%, 0.70%。由上述实验结果可得出以下结论:

LM-SVD 算法的表现优于 BiasSVD 矩阵分解、DN-CBR 深度神经网络和 ConvMF 深度矩阵分解, 证明将深度学习和矩阵分解进行融合, 可以通过融入用户、电影属性特征等辅助信息来缓解原有评分矩阵的数据稀疏性问题, 从而提升电影评分预测准确度。且效果好于单纯的深度学习评分预测模型。相对于仅仅利用电影描述等文本信息作为辅助信息的 ConvMF 算法, 文中提出的算法在用户和电影特征提取方面更加充分, 优化了评分预测效果。

4 结束语

文中将电影推荐作为研究对象, 针对矩阵分解推荐算法存在的数据稀疏性、冷启动和特征提取不充分等问题, 将深度学习和矩阵分解相融合, 提出 LM-SVD 推荐算法。首先利用 MLP 和 LSTM 的组合模型学习用户和电影的深层特征表示。之后将深层特征融入到 BiasSVD 矩阵分解算法中, 改进 SVD 的损失函数计算方式, 缓解了评分矩阵的数据稀疏性, 使得特征提取更加充分。在 MovieLens 的两个数据集上进行的算法对比实验表明, 该算法在电影评分预测方面的表现优于三种对比算法, 表现出较好的推荐性能。模型充

分利用了用户和电影的属性信息、文本信息以及用户电影评分信息,有效地缓解了传统矩阵分解中存在的
数据稀疏性和冷启动问题。但该模型没有考虑用户兴
趣偏好随时间的变化,下一步的研究可以考虑加入时
间因子。

参考文献:

- [1] 王晓通. 大数据背景下电影智能推送的“算法”实现及其潜在问题[J]. 当代电影, 2019(5): 64-70.
- [2] 翁小兰, 王志坚. 协同过滤推荐算法研究进展[J]. 计算机工程与应用, 2018, 54(1): 25-31.
- [3] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [4] WEI J, HE J, CHEN K, et al. Collaborative filtering and deep learning based recommendation system for cold start items [J]. Expert Systems With Applications, 2017, 69: 29-39.
- [5] 刘君良, 李晓光. 个性化推荐系统技术进展[J]. 计算机科学, 2020, 47(7): 47-55.
- [6] 黄立威, 江碧涛, 吕守业, 等. 基于深度学习的推荐系统研究综述[J]. 计算机学报, 2018, 41(7): 1619-1647.
- [7] CHENG H T, KOC L, HARMSSEN J, et al. Wide & deep learning for recommender systems[C]//Proceedings of the 1st workshop on deep learning for recommender systems. Boston: ACM, 2016: 7-10.
- [8] YANG L B, ZHENG Y, CAI X Y, et al. A LSTM based model for personalized context-aware citation recommendation[J]. IEEE Access, 2018, 6: 59618-59627.
- [9] 梁昌勇, 范汝鑫, 陆文星, 等. 基于 CNN-LFM 模型的个性化推荐[J]. 计算机仿真, 2020, 37(3): 399-404.
- [10] KIM D, PARK C, OH J, et al. Convolutional matrix factorization for document context-aware recommendation[C]//Proceedings of the 10th ACM conference on recommender systems. Boston: ACM, 2016: 233-240.
- [11] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.
- [12] WILLIAMS D, HINTON G. Learning representations by back propagating errors[J]. Nature, 1986, 323(6088): 533-538.
- [13] 王运, 倪静, 马刚. 基于 FunkSVD 矩阵分解和相似度矩阵的推荐算法[J]. 计算机应用与软件, 2019, 36(12): 245-250.
- [14] CHEN C, ZHANG M, LIU Y, et al. Neural attentional rating regression with review-level explanations[C]//Proceedings of the 2018 world wide web conference. Lyon: ACM, 2018: 1583-1592.
- [15] 樊海玮, 史双, 张博敏, 等. 基于 MLP 改进型深度神经网络学习资源推荐算法[J]. 计算机应用研究, 2020, 37(9): 2629-2633.
- [16] AN Jiyong, ZHOU Yong, ZHAO Yujun, et al. An efficient feature extraction technique based on local coding PSSM and multifeatures fusion for predicting protein-protein interactions [J]. Evolutionary Bioinformatics, 2019 (15): 117693431987992.
- [17] HUANG Y A, YOU Z H, GAO X, et al. Using weighted sparse representation model combined with discrete cosine transformation to predict protein-protein interactions from protein sequence[J]. Biomed Res Int, 2015, 2015: 902198.
- [18] HUANG Y A, YOU Z H, CHEN X, et al. Sequence-based prediction of protein-protein interactions using weighted sparse representation model combined with global encoding [J]. BMC Bioinformatics, 2016, 17(1): 1-11.
- [19] GAO Z G, WANG L, XIA S X, et al. Ens-PPI: a novel ensemble classifier for predicting the interactions of proteins using auto covariance transformation from PSSM[J]. Biomed Research International, 2016, 2016(4): 1-8.
- [20] NANINI L. An ensemble of K-local hyperplanes for predicting protein-protein interactions[J]. Bioinformatics, 2006, 22(10): 1207-1210.

(上接第 12 页)

- [8] HWANG S, JEONG M K. Robust relevance vector machine for classification with variational inference [J]. Annals of Operations Research, 2015, 263(1-2): 21-43.
- [9] GAN G, NG K P. Subspace clustering using affinity propagation[J]. Pattern Recognition, 2015, 48(4): 1455-1464.
- [10] GUO T, LI D, WANG K, et al. Adaptive affinity propagation clustering [J]. Acta Automatica Sinica, 2012, 33: 1242-1246.
- [11] TAO Z, RENBIAO W. Affinity propagation clustering of measurements for multiple extended target tracking[J]. Sensors, 2015, 15(9): 22646-22659.
- [12] MAYER C, BACHLER M, HOLZINGER A, et al. The effect of threshold values and weighting factors on the association between entropy measures and mortality after myocardial infarction in the cardiac arrhythmia suppression trial (CAST) [J]. Entropy, 2016, 18(4): 129.
- [13] BEKIROU S, NGUYEN D K, JUNIOR L S, et al. Information diffusion, cluster formation and entropy-based network dynamics in equity and commodity markets[J]. European Journal of Operational Research, 2017, 256(3): 945-961.