

doi: 10.3969/j.issn.0490-6756.2019.06.008

基于深度神经网络和概率矩阵分解的混合推荐算法

胡思才^{1,2}, 孙界平¹, 琚生根¹, 王霞¹

(1. 四川大学计算机学院, 成都 610065; 2. 解放军 61920 部队, 成都 610505)

摘要: 针对个性化推荐中用户和项目描述信息未充分利用, 用户评分矩阵数据集极端稀疏的情况, 提出了基于深度神经网络和概率矩阵分解(PMF)的混合推荐算法. 首先, 对用户和项目描述信息进行预处理, 形成包含用户偏好特征的用户和项目特征集, 再将各特征输入深度神经网络模型中进行训练. 同时, 利用概率矩阵分解模型, 根据用户评分矩阵通过最大后验估计优化得到潜在特征向量; 然后, 通过对概率矩阵分解模型的用户和项目潜在特征向量以及深度神经网络模型的真实特征向量进行迭代更新, 收敛得到融合用户和项目真实信息的潜在特征向量; 最后, 利用该特征向量对用户进行个性化推荐. 实验证明, 本文算法较经典推荐算法以及前人算法在均方误差与平均绝对误差指标上均有改善, 说明本文算法的有效性.

关键词: 混合推荐; 矩阵分解; 神经网络; 特征向量; 卷积

中图分类号: TP391

文献标识码: A

文章编号: 0490-6756(2019)06-1033-09

Hybrid recommendation algorithm based on deep neural network and probabilistic matrix factorization

HU Si-Cai^{1,2}, SUN Jie-Ping¹, JU Sheng-Gen¹, WANG Xia¹

(1. College of Computer Science, Sichuan University, Chengdu 610065, China;

2. Troops 61920 of PLA, Chengdu 610505, China)

Abstract: Aiming at the facts that user and project description information is not fully utilized in personalized recommendation and user score matrix data set is extremely sparse, a hybrid recommendation algorithm based on deep neural network and probabilistic matrix factorization (PMF) is proposed. Firstly, user and item description information is preprocessed to form user and item feature sets containing user preference, and then each feature is fed into the deep neural network model for training. At the same time, the probabilistic matrix decomposition model is used to optimize the potential eigenvectors based on the maximum posterior estimation of the user score matrix. Then the potential feature vectors of the probabilistic matrix model and the real feature vectors of the deep neural network model are iteratively updated to converge to the potential feature vectors that fuse the real information of the user and the item. Finally, this feature vector is used to make personalized recommendation to users. Experiments show that the proposed algorithm is better than the classical recommendation algorithm and previous algorithms in term of the mean square error and mean absolute error index, which shows the effectiveness of the proposed algorithm.

Keywords: Hybrid recommendation; Matrix factorization; Neural network; Eigenvectors; Convolution

收稿日期: 2019-04-24

基金项目: 四川省重点研发项目(2018GZ0182)

作者简介: 胡思才(1987-), 男, 硕士生, 研究方向为数据科学.

通讯作者: 琚生根. E-mail: jsg@scu.edu.cn

1 引言

随着互联网技术的快速发展,社会已呈现出数据信息井喷的状态,人们逐渐从信息匮乏的时代迈入到信息爆炸的大数据时代.为了解决信息繁多冗余带来的信息低可用性问题,使得用户能够有效地获得自己需要的信息,推荐系统应运而生.个性化推荐系统是根据用户购买消费记录、用户和项目特征以及对项目的评价等各类信息,为用户推荐用户感兴趣的信息和项目.在众多的推荐系统中,协同过滤推荐算法^[1]无疑是最受欢迎的推荐算法,但是用户评分矩阵的稀疏性导致了传统的协同过滤推荐算法在评分预测精度上的下降^[2],影响了实际的推荐效果,从而降低了推荐的质量.

国内外许多研究人员根据推荐系统不同的应用场景和存在的一些突出问题提出了很多优化改进算法. Raghavan 等人^[3]提出了一种两阶段模型,该模型首先使用公式来估计评审质量分数,然后再基于质量分数进行加权评分,建立了概率协同过滤模型,这种两阶段算法能够在一定程度上提高推荐系统的整体性能. Patra 等人^[4]将 Bhattacharyya 测度用于相似度计算方法中,能够综合利用用户评级信息,在稀疏的评级数据集中定位活跃用户的有用邻居,可以为稀疏数据中的活动用户提供相对可靠的项目推荐. Li 等人^[5]通过将概率矩阵分解与边缘去噪的堆叠式自编码器相结合,能够从用户项评级和侧面信息中发掘有效的潜在因素,取得了较好的性能. Paradarami 等人^[6]提出了一种利用人工神经网络框架来预测用户评分的新算法,该算法能够有效融合用户评分矩阵和项目的重要信息,通过对特征空间的拟合训练,形成一个评级分类模型. Kim 等人^[7]提出了一种基于上下文感知的卷积矩阵分解模型,通过文档建模的方法有效利用项目评论信息,从而能够捕获文档的上下文信息,有效地缓解了数据稀疏性问题. 杨苏雁^[8]提出了一种基于矩阵分解及神经网络的推荐排序学习方法,通过对用户评分信息进行矩阵分解获得用户和项目的潜在特征向量;再通过对该潜在特征向量进行聚类操作,形成基于用户行为数据的类别信息,并与用户和项目的基础信息进行交叉处理,丰富了模型的特征输入;最后将所有的特征输入到神经网络中进行处理,获得用户评分信息及用户推荐项目,该方法有效提升了推荐系统的精准度和用户体验.

上述算法均在一定程度上提高了个性化推荐

的效果,但是这些算法基本上都忽略了用户的文档描述信息及用户偏好特征情况.在推荐过程中,用户的文档描述信息包含了很多相关的重要内容,能够有效地体现出用户的消费层次及用户偏好等潜在特征,因此用户的文档描述信息对个性化推荐能够起到很好的辅助作用.本文不仅考虑了用户评分矩阵和项目的属性信息,同时对用户的文档描述信息和用户的偏好特征也加以利用,能够进一步缓解用户评分矩阵稀疏性所带来的推荐质量不高的问题.本文采用经典数据集 MovieLen 电影网站的数据集来进行相关的实验.因此数据的预处理则以 MovieLen(1 M)数据为例进行说明,该算法也同样适用于其他相关数据集.

2 基础知识

2.1 矩阵分解模型

基于矩阵分解的协同过滤推荐算法可以考虑成对用户评分矩阵进行分解来完成用户对项目的预测评分任务:给定一个矩阵 $R = [r_{ij}] \in R^{m \times n}$, 行代表用户,列代表项目,非 0 值就是已知评分,在一个 k 维因子模型中,需要找到两个潜在特征矩阵 $U \in R^{m \times k}$, $V \in R^{n \times k}$,使得 $R \approx UV$.其中,用户 i 的潜在特征用 $u_i \in R^k$ 表示,项目 j 的潜在特征用 $v_j \in R^k$ 表示.系统中 v_j 的每个元素就代表项目 j 的特征,例如电影的类型、名称等,而 u_i 中的每个元素代表了用户的特征.用户对项目的爱好取决于用户和项目的这几个因子匹配情况,最后评分结果就是用户 i 和物品 j 这些因子的线性组合,由式(1)表示.

$$r_{ij} = \sum_{l=1}^k u_{il}v_{jl} = u_i^T v_j \quad (1)$$

为了找到最优 U, V ,利用式(2)的目标函数进行优化解决,其中, λ 是正则参数且 $\lambda > 0$.

$$\min_{U \in R^{m \times k}, V \in R^{n \times k}} \frac{\lambda}{2} (\|U\|_F^2 + \|V\|_F^2) + \sum_{ij \in S} (r_{ij} - u_i^T v_j)^2 \quad (2)$$

2.2 神经网络

神经网络由输入层、隐藏层和输出层三部分组成.应用于推荐算法的神经网络模型的输入层用来输入用户和项目的特征数据,隐藏层为用户对这些项目特征的喜好程度数据,输出层输出评分值.对于待推荐项目,在提取其项目特征后,将项目特征向量作为输入,经过训练好的用户偏好模型,可以得到用户对该项目的偏好程度输出(即用户对项目的评分).神经网络是一种优秀的建模工具,用于近

似任何非线性关系并在数据中找到模式. 用式(3)表示隐藏层中神经元的矢量化方式.

$$h^{(l)} = \sigma(W^{(l-1)T}h^{(l-1)} + b^{(l)}) \quad (3)$$

每个神经网络是一个对输入特征 $h^{(l-1)}$ 附上权重参数的非线性组合. 激活函数在特征空间中具有识别关系的能力, 对神经网络的复杂度和性能有很大的影响.

3 改进的混合推荐算法

3.1 混合推荐算法的总体框架

在个性化推荐中, 概率矩阵分解模型 (PMF) 具有缓解数据稀疏的作用, 它能够将用户和项目进行特征分解, 生成用户、项目的潜在特征, 能够有效提高个性化推荐的效果^[9-13]. 但是概率矩阵分解模型得到的潜在特征向量只是通过用户评分矩阵获得的潜在特征, 并非真正代表用户和项目的真实特征. 同时用户和项目描述语句中除了用户评分矩阵外, 还有大量的信息可以利用, 如用户年龄、职业、用户偏好、项目名称、属性等文档描述信息. 深度神经网络在特征训练方面有着其他算法无法比拟的优势, 因此通过深度神经网络对这些用户和项目的真实信息进行特征训练, 能够很好的将这些真实特征和潜在特征向量结合起来. 基于特征训练这个特点, 本文将概率矩阵分解和深度神经网络这两种算法在训练用户和项目潜在特征这方面进行融合, 能够有效的发挥两种算法的优势, 将用户和项目的潜在特征向量与其真实特征进行关联, 为潜在特征向量提供与其对应的现实意义, 能够为后续的用户推荐发挥重要意义.

概率矩阵分解模型是在正则化矩阵分解的基础上, 引入了概率模型来进一步优化. 假设用户 U 和用户 V 的特征矩阵均服从高斯分布, 通过用户评分矩阵已知的值, 得到 U 和 V 潜在特征向量, 然后利用该特征向量去预测评分矩阵中的未知值. 观测噪声是指预测评分矩阵和真实评分矩阵的差值, 其为高斯分布, 由式(4)表示.

$$P(R_{ij} - U_i^T V_j | 0, \sigma^2) \Rightarrow P(R_{ij} | U_i^T V_j, \sigma^2) \quad (4)$$

评分矩阵 R 的条件概率由式(5)表示. 其中, U 和 V 是参数; $N(x | \mu, \sigma^2)$ 是具有均值 μ 和方差 σ^2 的高斯正态分布的概率密度函数; I_{ij} 是指标函数, 如果训练集中有用户 i 对项目 j 的评分, 则 I_{ij} 为 1, 否则为 0.

$$P(R | U, V, \sigma^2) =$$

$$\prod_{i=1}^N \prod_{j=1}^M [N(r_{ij} | u_i v_j, \sigma^2)] I_{ij} \quad (5)$$

用户和项目真实特征集通过深度神经网络进行迭代训练, 以优化后的潜在特征向量 U 和 V 为目标值, 训练网络中的各个权值参数. 以 δ_u 和 δ_v 变量作为概率矩阵分解模型训练的潜在特征和深度神经网络训练的文档真实特征的误差情况, 将其作为零均值高斯噪声, 进一步优化特征模型. 模型中潜在特征表示如式(6).

$$\begin{cases} U = DL(W_u, X_u) + \delta_u, \delta_u \sim N(0, \sigma_u^2), \\ V = DL(W_v, X_v) + \delta_v, \delta_v \sim N(0, \sigma_v^2). \end{cases} \quad (6)$$

其中, X_u 和 X_v 是用户和项目的真实特征集合. 使用从深度神经网络模型获得的真实特征向量作为高斯分布的平均值, 并将该用户和项目的高斯噪声用作高斯分布的方差, 作为深度神经网络和概率矩阵分解模型之间的关联点, 有助于分析用户项目的文档描述信息和用户评分矩阵的关系.

对于用户和项目特征训练的每个权重 W_u 、 W_v , 我们使用零均值高斯分布, 如式(7)所示.

$$\begin{cases} P(W_u | 0, \sigma_{w_u}^2) = \prod N(w_u | 0, \sigma_{w_u}^2), \\ P(W_v | 0, \sigma_{w_v}^2) = \prod N(w_v | 0, \sigma_{w_v}^2). \end{cases} \quad (7)$$

基于这两种算法融合改进的总体框架图 1 所示.

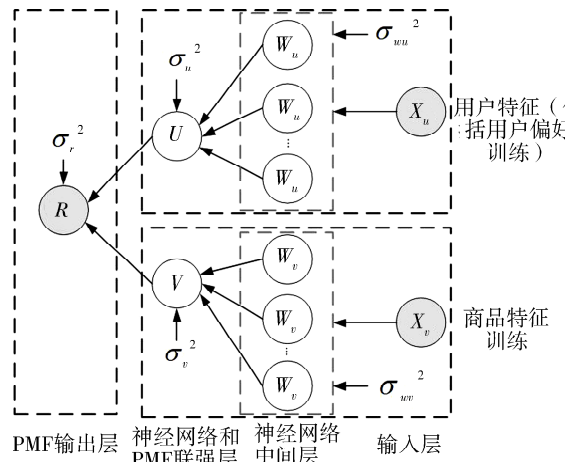


图 1 算法总体框架

Fig. 1 Algorithm framework

图 1 显示了基于深度神经网络的概率矩阵分解模型的结构, 它将深度神经网络集成到 PMF 中. 例如文件中含有 N 个用户和 M 个项目, 则其用户评分矩阵由 $R \in R^{N \times M}$ 的矩阵表示. 该模型的目标是找到包含深度神经网络拟合训练的用户和项目真实信息的潜在特征向量 U 和 V (其中 $U \in R^{K \times N}$ 和 $V \in R^{K \times M}$), 通过其乘积 ($U^T V$) 重建评级矩阵 R , 完成对用户的推荐.

3.2 基于用户偏好的深度神经网络模型

3.2.1 用户和项目信息预处理 深度神经网络的训练需要大量的用户和项目的信息进行建模.

本文在深度神经网络训练之前,充分利用用户项目的文档描述信息,在对其进行预处理(包括特征编码)后,根据用户评分矩阵以及项目的属性特点,提取用户偏好特征,再将预处理后的各类特征输入到网络模型中,能够有效地减少网络训练的时间,同时也能够合理地利用用户的偏好信息.

用户偏好是指用户喜欢或者是重点关注项目的特征类型.例如,某用户的观看记录是《星球大战》、《星际迷航》及《天外来客》,则可将该用户定位成科幻片爱好者,而最近的一部影片《复仇者联盟3:无限战争》根据其剧情内容会被打上“科幻”的标签,则在特征匹配时,模型会有很大概率将这部影片推荐给该用户.

用户偏好特征可以根据对用户已评分项目的属性信息进行分析,定量分析用户偏好属性值,从而为用户推荐提供参考数据.用户 U 对项目特征属性的兴趣程度可用矩阵 UP 描述.其中, up_{ij} 在区间 $[-1, 1]$ 上取值, 1 表示喜欢的程度最高; -1 表示喜欢的程度最低.

$$UP = \begin{bmatrix} up_{11} & up_{12} & \cdots & up_{1n} \\ up_{21} & up_{22} & \cdots & up_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ up_{m1} & up_{m2} & \cdots & up_{mn} \end{bmatrix} \quad (8)$$

用户已评分项目代表用户已观看的项目,但由于看过不代表喜欢看,用户根据自己对项目的喜好程度以评分来表示,分值越高代表喜欢的程度越高,分值越低代表用户不喜欢的程度就越高.因此将用户喜欢的项目(评分为 3、4、5 分的项目)定义为积极项目,用户不喜欢的项目(评分为 1、2 分的项目)定义为不积极项目.针对积极项目和不积极项目,分别计算用户对相应项目的每个属性的偏好值,将积极项目和不积极项目的各属性偏好值进行归一化处理,再将对应的属性偏好值进行求差处理,最终确定属性偏好值较大的属性作为每个用户的偏好特征;用户 U 对项目的第 i 个特征的喜爱程度 up_{ij} 定义如式(9)所示.其中, i 表示用户编号; j 表示项目属性编号; n_1 表示用户 i 的积极项目的数量; n_2 表示用户 i 的不积极项目的数量.

$$up_{ij} = \frac{1}{n_1} \sum_{k=1}^{n_1} I_{jk}^{\text{top}} - \frac{1}{n_2} \sum_{k=1}^{n_2} I_{jk}^{\text{bottom}} \quad (9)$$

对用户偏好特征通过 One-Hot 编码方式进行

操作,采用状态寄存器来对用户多个偏好特征状态进行编码,每个状态都有一个独立的寄存器位,形成一个多元特征向量.

3.2.2 深度神经网络训练 在深度神经网络中,首先对用户特征、用户偏好特征和项目特征通过嵌入层对特征进行编码处理.其中,项目名称是一列单词组成的词系列,对其利用文本卷积网络进行处理,网络的第一层是词嵌入层,下一层利用不同窗口大小(窗口大小指每次卷积覆盖的单词数量)的卷积核在嵌入矩阵上做卷积,再通过池化层利用 max pooling 得到一个长向量,最后使用 dropout 做正则化,即可得到项目名称的特征.然后将各特征传入全连接层,最终输出用户项目真实特征向量,将输出值回归到概率矩阵分解模型生成的潜在特征向量中,采用 MSE 优化损失.整个深度神经网络训练的流程框架如图 2 所示.

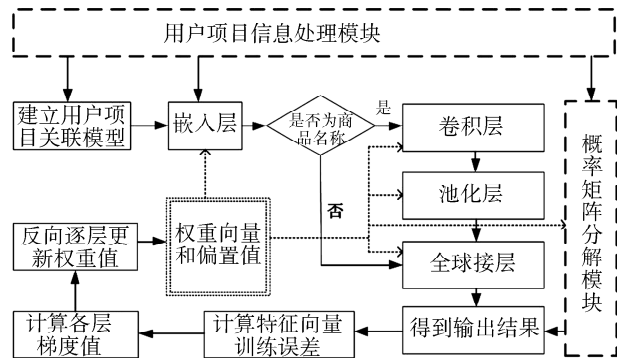


图 2 深度神经网络训练流程图
Fig. 2 Deep neural network training flow chart

深度神经网络训练模型的目的是根据训练集找到合适的权重矩阵 W 以及它的偏置 b . 在训练的过程中,本文采用的是块数据(batch)训练模式,即并不是所有数据训练完才进行模型的更新,而是对数据进行分块训练,在块数据训练完成后即可进行模型更新.本文对所有训练数据进行多次迭代,在每次迭代的过程中都对训练数据做随机打乱,然后从打乱的数据中依次选出块大小的数据做训练,直到所有数据都被遍历完毕,再进行下一次的迭代训练.

在迭代过程中,样本集 x_k 经过一轮运算后,会得到一个深度神经网络模型的各个预测特征分量的特征值 $dl_{w,b}(x_k)$,它总体的损失函数由式(10)表示.

$$L(W_u, b) = \frac{1}{n} \sum_{k=1}^{|X|} \sum_{i=1}^n (dl_{w,b}(x_k) - u_i)^2 + \frac{\lambda}{2} \sum_{i=1}^{|W_u|} w_i^2$$

$$L(W_v, b) = \frac{1}{n} \sum_{k=1}^{|X|} \sum_{j=1}^m (dl_{w,b}(x_k) - v_j)^2 + \frac{\lambda}{2} \sum_{j=1}^{|W_v|} \omega_j^2 \quad (10)$$

其中, u_i 和 v_j 是通过概率矩阵分解模型优化后得到的用户和项目的潜在特征, 损失函数的值越小就意味着预测得越准确. 式(10)中的第一项是一个均方差项; 第二项是规则化项, 这个规则化项所起的主要作用就是防止过拟合. 最后利用梯度下降法反向调整误差, 并更新模型参数. 优化目的就是寻找合适的参数组合 (W, b) 使得 $L(W, b)$ 最小.

3.2.3 嵌入层 嵌入层可以用来处理推荐系统中的稀疏矩阵问题. 用户和项目特征通过特征编码后输入到嵌入层中, 由嵌入层来嵌入用户和项目的文档描述信息: 首先, 对用户和项目真实特征(如用户年龄、职业、偏好、项目名称及属性等)分配索引, 用这些索引构建用户和项目的各类别的属性序列; 再将该索引转换为具有固定大小的向量. 初始向量值为 0 到 1 之间的正态分布随机变量, 每一个特征根据编码值获取该特征对应的一列随机变量值. 在训练神经网络的过程中, 每个嵌入的向量都会得到更新. 因此能够在多维空间中探索用户和项目的潜在特征. 嵌入矩阵需要明确词向量的长度 l , 即每个索引需要分配多少个潜在因子. 通常情况下, 嵌入矩阵词向量长度设置为 32 或者 50, 本文考虑到所处理的用户和项目特征数目较多, 因此采用较小的向量长度 32 对特征进行嵌入.

3.2.4 卷积层和池化层 卷积层中的特征图由经过嵌入层输出的嵌入矩阵和卷积核通过卷积计算得到, 卷积核又被称为滤波器. 在通过嵌入层处理后, 项目名称嵌入矩阵为 $s \times l$ (其中, s 为项目名称长度, l 为嵌入矩阵词向量长度), 本文将 s 设置为项目名称的最大单词数 18. 利用特征窗口 $win = \{2, 3, \dots, m, \dots, t\}$ 对项目名称嵌入矩阵做卷积操作, 其中每个特征窗口有 m 个项目名称的嵌入向量, 卷积核可表示为一个 $m \times l$ 的矩阵 c_m , 通过卷积操作后, 特征图能够得到 $s - m + 1$ 个特征, 即特征图 $f = [f_1, f_2, \dots, f_{s-m+1}]$, 其中, 第 i 个特征 f_i 计算方法如式(11)所示.

$$f_i = g(c_m \otimes x_{i_i} + b) \quad (11)$$

其中, m 表示该卷积核内的特征向量中第 m 个分量; b 表示卷积偏置项; $g(x)$ 表示非线性激活函数.

池化层又被称为下采样层, 通过对卷积运算得到的特征图按照一定规则进行采样缩放, 有利于减

低网络的复杂度. 本文采用最大池化操作对卷积层得到的每一个特征图进行处理, 获得项目名称中最主要的信息. 在卷积层中采用了几个不同窗口长度的卷积核, 其中每个卷积核都可提取出一个特征. 卷积核矩阵为 $m \times l$ 的池化操作由式(12)表示.

$$pool_feature_m = \max(f_1, f_2, \dots, f_{s-m+1}) \quad (12)$$

网络模型中卷积层和池化层的结构如图 3 所示. 图中句子长度设置为训练集中项目名称最大单词数, 通过卷积池化后获得项目名称的一列特征向量. 池化操作不需要偏置, 经过池化后将该特征向量进行 dropout 处理, 以缓解训练过拟合的情况, 最后得到全连接层的输入特征.

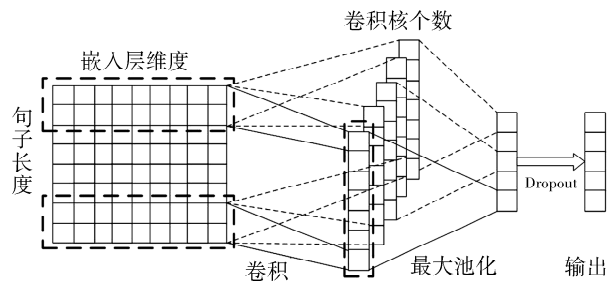


图 3 卷积层和池化层结构

Fig. 3 Convolution layer and pooling layer structure

3.2.5 全连接层 全连接层是对所有特征进行加权处理后获得用户和项目的潜在特征值. 本文在全连接层中, 将用户和项目的真实特征(包括用户偏好特征)分别进行全连接处理, 获得用户和项目的真实特征向量, 再将网络模型训练的真实特征和概率矩阵分解模型优化得到的用户和项目的潜在特征进行融合, 得到关于两个模型输出结果的均方差的损失函数, 通过对损失函数进行梯度下降逐步地遍历特征空间, 进而优化网络模型.

3.3 基于概率矩阵分解的特征优化算法

概率矩阵分解模型通过将概率函数融入矩阵分解模型中, 能够通过贝叶斯公式变换得到最大后验(maximum a posteriori)估计, 逐步迭代训练获得用户和项目的潜在特征向量, 再利用用户和项目的潜在特征向量对用户未评分项目进行评分预测, 最终向用户进行推荐. 概率矩阵分解模型能够有效地缓解用户评分矩阵稀疏的问题, 在一定程度上提高了个性化推荐准确率.

用户和项目的真实特征是通过深度神经网络进行优化后所获得的, 其与基于 PMF 输出的潜在特征的条件分布如式(13), 其中, X_u 、 X_v 是用户和

项目的真实特征集. 用户和项目之间的特征是独立存在的, 因此本文对神经网络中用户和项目这两部分损失函数分别进行优化.

$$\begin{cases} P(U|W_u, X_u, \sigma_u^2) = P(U|DL(W_u, X_u), \sigma_u^2) = \prod N(u_i | dl(\omega_u, x_u), \sigma_u^2) \\ P(V|W_v, X_v, \sigma_v^2) = P(V|DL(W_v, X_v), \sigma_v^2) = \prod N(v_j | dl(\omega_v, x_v), \sigma_v^2) \end{cases} \quad (13)$$

为了优化用户和项目的潜在特征向量、神经网络权重和偏差变量等变量, 本文使用最大后验估计(MAP)进行优化, 如式(14)所示. 该公式通过朴素贝叶斯转换为计算最大似然估计和参数的先验概率的乘积.

$$\frac{P(U, V, W_u, W_v | R) = P(R|U, V, W_u, W_v) \times P(U, V, W_u, W_v)}{P(R)} \propto P(R|U, V; W_u, W_v) \times P(U, V, W_u, W_v) \quad (14)$$

假设 U 和 V 相互独立, 可以通过贝叶斯公式得到 R, U, V 的联合分布: 用户 U 的特征矩阵满足均值为 $dl(\omega_u, x_u)$, 方差为 σ_u^2 的高斯分布, 项目 V 的特征向量满足均值为 $dl(\omega_v, x_v)$, 方差为 σ_v^2 的高斯分布, 如式(15)所示.

$$\begin{aligned} P(U, V, W_u, W_v | R) &= P(R|U, V) \times P(U, W_u) \times P(V, W_v) = \\ &P(R|U, V, \sigma_r^2) \times P(U|W_u, X_u, \sigma_u^2) \times \\ &P(W_u | 0, \sigma_{wu}^2) \times P(V|W_v, X_v, \sigma_v^2) \times \\ &P(W_v | 0, \sigma_{wv}^2) = \\ &\prod \prod [^N(R_{ij} | u_i v_j, \sigma_r^2)] I_{ij} \times \end{aligned}$$

$$\prod N(u_i | dl(\omega_u, x_u), \sigma_u^2) \times \prod N(\omega_u | 0, \sigma_{wu}^2) \times \prod N(v_j | dl(\omega_v, x_v), \sigma_v^2) \times \prod N(\omega_v | 0, \sigma_{wv}^2) \quad (15)$$

计算最大后验估计, 通常对公式两边同时取对数(对数似然), 以便将乘法转换为加法, 再根据正态分布的联合概率密度函数求导算法, 得到用户评分矩阵 R 的条件概率公式以及用户和项目的潜在特征的条件分布如式(16).

$$\begin{cases} \ln P(R | U, V, \sigma_r^2) = \sum_i \sum_j \frac{I_{ij}}{2} (r_{ij} - u_i v_j)^2, \\ \ln P(U | W_u, X_u, \sigma_u^2) = \frac{\lambda_u}{2} \sum_i [u_i - dl(\omega_u, x_u)]^2, \\ \ln P(V | W_v, X_v, \sigma_v^2) = \frac{\lambda_v}{2} \sum_j [v_j - dl(\omega_v, x_v)]^2. \end{cases} \quad (16)$$

将式(16)代入 $\ln P(U, V, W_u, W_v | R)$ 即可得到带有网络权值的 PMF 模型的损失函数, 如式(17)所示.

$$\begin{aligned} L(U, V, W_u, W_v) &= \sum_i \sum_j \frac{I_{ij}}{2} (r_{ij} - u_i v_j)^2 + \\ &\frac{\lambda_u}{2} \sum_i [u_i - dl(\omega_u, x_u)]^2 + \frac{\lambda_{wu}}{2} \sum_i^{|\omega_u|} \omega_u^T \cdot \omega_u + \frac{\lambda_v}{2} \sum_j [v_j - dl(\omega_v, x_v)]^2 + \\ &\frac{\lambda_{wv}}{2} \sum_i^{|\omega_v|} \omega_v^T \cdot \omega_v \end{aligned} \quad (17)$$

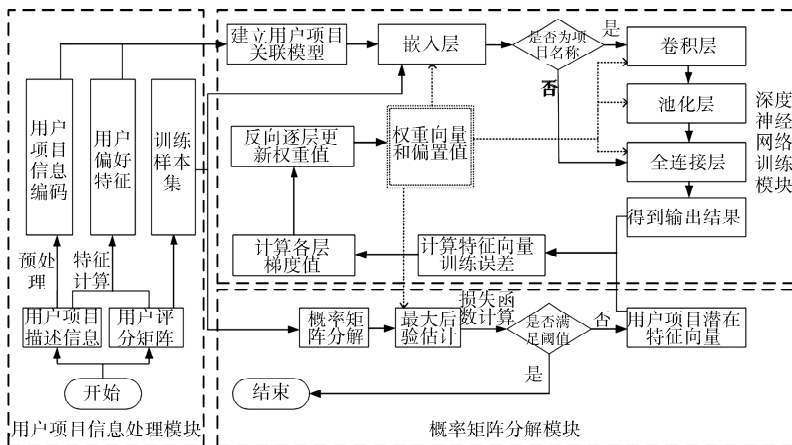


图 4 整体算法训练流程图
Fig. 4 Overall algorithm training flow chart

通过该模型的损失函数的偏导数计算, 即可得到损失函数针对用户和项目特征向量的偏导数, 如

式(18)所示. 再利用梯度下降方法求解 U, V 中的每一个元素.

$$\begin{cases} \frac{\partial L}{\partial U} = -2I_{ij}(R-U^T V)V + \lambda_u[U - DL(W_u, X_u)], \\ \frac{\partial L}{\partial V} = -2I_{ij}(R-U^T V)U + \lambda_v[V - DL(W_v, X_v)]. \end{cases} \quad (18)$$

深度神经网络模型中的权重系数 W 与网络训练的特征密切相关,如嵌入层、卷积层、池化层、全连接层等训练.当 U 和 V 恒定时, l 可以解释为具有 L2 正则项的均方误差函数.上一节主要就网络模型中权重系数 W 利用反向传播算法训练优化进行了详细介绍,这里就不再赘述了.本文算法的整体训练流程如图 4 所示.

在图 4 中,训练样本集是由用户评分矩阵生成多批次 batch 数据集,一次训练一个 batch 数据集.在数据集中将用户评分矩阵以及用户和项目描述信息经过预处理整合后共同输入到神经网络训练模块中进行训练,同时该 batch 数据集也输送到概率矩阵分解模块进行处理,通过最大后验估计得到新的用户潜在特征向量 U 和项目潜在特征向量 V .将该潜在特征向量作为神经网络训练模块的输出参考值,通过对神经网络训练模块的输出结果与潜在特征向量进行误差计算,得到深度神经网络模型的损失函数,最后利用梯度下降法更新深度神经网络模块的权重向量 W_u 、 W_v 和偏置值.通过多批次 batch 数据集的训练迭代,替代地更新 U 、 V 和 W_u 、 W_v ,直到这四个参数达到最优值.通过优化的 U 、 V 和 W_u 、 W_v ,即可预测用户对项目的未知评分.

4 实验验证

4.1 实验数据

实验采用 MovieLen 电影网站的数据集 MovieLen (1 M) (<http://grouplens.org/datasets/movielens/>).数据集描述如下所示.

表 1 MovieLens(1 M)数据描述

Tab.1 Data description of MovieLens(1 M)

用户数	物品数	评分数
6 040	3 883	1 000 209

由于本文算法能够有效地缓解用户评分矩阵极度稀疏的问题,因此本文将数据集根据用户评分稀疏情况,将数据集分为以下三部分进行实验,最后对整个数据集进行实验验证,以表明本文算法的有效性.

表 2 实验数据集

Tab.2 Experimental data set

数据集	用户评分量 N	数据集大小/条	稀疏度/%
data1	$N > 450$	329 668	81.08
data2	$200 \leq N \leq 450$	327 313	91.09
data3	$N < 200$	343 228	97.81
data4	全部	1 000 209	95.73

4.2 评价指标

为了衡量算法的预测准确度,本实验采用均方误差(MSE)和平均绝对误差(MAE)来验证本文算法所得预测结果与用户真实评分的差距.均方误差(MSE)计算如式(19),平均绝对误差(MAE)计算如式(20).

$$MSE = \frac{1}{n} \sum_{i=1}^n (r_i - pre_i)^2 \quad (19)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |r_i - pre_i| \quad (20)$$

其中, r_i 表示用户对物品的真实评分; pre_i 表示用户对物品的预测评分; n 表示测试集大小. MSE 和 MAE 值越小,说明预测值与实际值越近,预测结果的准确率越高.

4.3 实验结果

将本文算法与前人算法文献[6]、文献[7]以及基于用户的协同过滤推荐算法(CF)和概率矩阵分解算法(PMF)进行比较,以验证本文算法的有效性.

为了评估本文算法在个性化推荐中的泛化能力和鲁棒性,对每个数据集进行五折交叉验证:将数据集随机划分为 5 等份,其中,4 份用作训练数据,另外一份用作测试数据,按这个方式对数据集进行训练和测试.实验结果如下所示.

(1) 针对数据集 data1 的均方误差和平均绝对误差情况如图 5 所示.

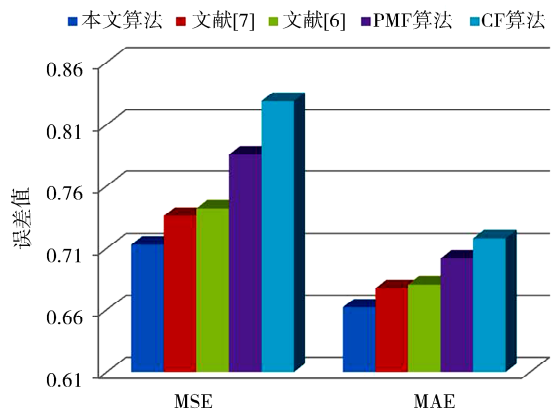


图 5 对数据集 data1 的预测误差对比实验
Fig.5 The experiment on prediction error of the data1

(2) 针对数据集 data2 的均方误差和平均绝对误差情况如图 6 所示。

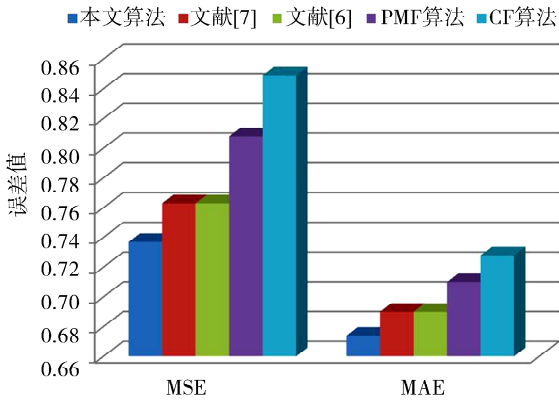


图 6 对数据集 data2 的预测误差对比实验

Fig. 6 The experiment on prediction error of the data2

(3) 针对数据集 data3 的均方误差和平均绝对误差情况如图 7 所示。

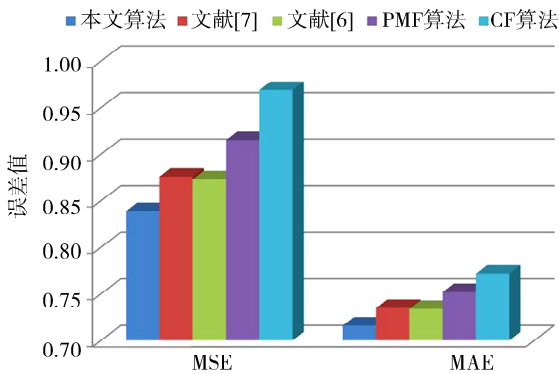


图 7 对数据集 data3 的预测误差对比实验

Fig. 7 The experiment on prediction error of the data3

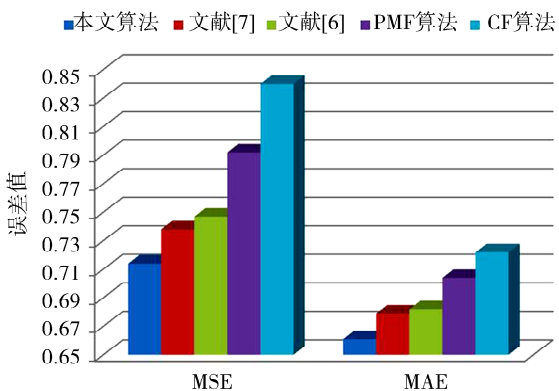


图 8 对数据集 data4 的预测误差对比实验

Fig. 8 The experiment on prediction error of the data4

分稀疏情况所拆分的三个数据集进行实验验证,本文算法无论是对稀疏度高的数据还是稀疏度低的数据,相较于传统的 CF 推荐和基础的概率矩阵分解算法都有较大的优势,同时较前人算法也有所改进。

(4) 针对数据集 data4 的均方误差和平均绝对误差情况如图 8 所示。

由图 8 看出,本文算法对整个数据集在 MSE、MAE 方面均优于前人算法以及经典算法,说明了本文算法的有效性。

5 结 论

本文在前人研究的基础上,不仅充分利用了用户评分矩阵数据,同时将用户描述信息和项目描述信息利用深度神经网络进行特征提取,生成包括用户偏好特征在内的用户和项目真实特征集,再利用最大后验估计对概率矩阵模型和深度神经网络生成的特征进行迭代优化处理,进一步降低了推荐的误差,提高了个性化推荐的准确率。未来可以继续探索将项目评价内容等更多的信息加入到这个模型中,进一步增强潜在特征向量和现实描述信息的关联度,提高个性化推荐的效果。

参考文献:

- [1] Liu H F, Hu Z, Mian A, *et al.* A new user similarity model to improve the accuracy of collaborative filtering [J]. *Knowl-Based Syst*, 2014, 56: 156.
- [2] 王洋, 骆力明. 一种解决协同过滤数据稀疏性问题的方法[J]. *首都师范大学学报:自然科学版*, 2012, 33: 1.
- [3] Raghavan S, Gunasekar S, Ghosh J. Review quality aware collaborative filtering [C]//*Proceedings of the sixth ACM conference on Recommender systems*. Dublin, Ireland: ACM, 2012.
- [4] Patra B K, Launonen R, Ollikainen V, *et al.* A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data[J]. *Knowl-Based Syst*, 2015, 82: 163.
- [5] Li S, Kawale J, Fu Y. Deep collaborative filtering via marginalized denoising auto-encoder [C]//*Proceedings of the 24th ACM International*. [S. l.]: ACM, 2015.
- [6] Paradarami T K, Bastian N D, Wightman J L. A hybrid recommender system using artificial neural networks [J]. *Expert Syst Appl*, 2017, 83: 300.
- [7] Kim D, Park C, Oh J, *et al.* Convolutional matrix

从图 5~图 7 可以看出,将数据集按照用户评

- factorization for document context-aware recommendation [C]//Proceedings of the ACM Conference on Recommender Systems. [S. l.]: ACM, 2016.
- [8] 杨苏雁. 基于矩阵分解及深度神经网络的推荐排序学习方法研究[D]. 北京: 北京交通大学, 2018.
- [9] Salakhutdinov R. Probabilistic matrix factorization [C]//Proceedings of the International Conference on Neural Information Processing Systems. Curran Associates Inc. [S. l.]: [s. n.], 2007.
- [10] 张昇. 基于概率矩阵分解的推荐算法[J]. 西安航空学院学报, 2017, 35: 78.
- [11] Tan F, Li L, Zhang Z, *et al.* A multi-attribute probabilistic matrix factorization model for personalized recommendation [J]. *Pattern Anal Appl*, 2016, 19: 857.
- [12] Huang L, Tan W, Sun Y. Collaborative recommendation algorithm based on probabilistic matrix factorization in probabilistic latent semantic analysis [J]. *Multimed Tool Appl*, 2018: 1.
- [13] Akulwar P, Pardeshi S. Bayesian probabilistic matrix factorization—a dive towards recommendation [C]// Proceedings of the International Conference on Inventive Computation Technologies. [S. l.]: IEEE, 2017.

引用本文格式:

中文: 胡思才, 孙界平, 琚生根, 等. 基于深度神经网络和概率矩阵分解的混合推荐算法[J]. 四川大学学报: 自然科学版, 2019, 56: 1033.

英文: Hu S C, Sun J P, Ju S G, *et al.* Hybrid recommendation algorithm based on deep neural network and probabilistic matrix factorization [J]. *J Sichuan Univ: Nat Sci Ed*, 2019, 56: 1033.