

一种混合推荐算法的Mahout实现

唐 科

(电子科技大学成都学院计算机学院, 四川 成都 611731)

✉kevindictor@aliyun.com



摘 要: 推荐算法作为推荐引擎实现的核心而得到广泛研究。在各类推荐算法中, 大部分对于用户行为特征属性、用户人口属性、物品特征属性, 以及用户—物品关联特征属性等参数的应用方式存在局限性。它们一般采用相似度计算、或模型计算等方法, 其特征提取及参数的调优依赖于事前定义, 存在参数优化效率低的问题。本文结合机器学习技术, 提出一种混合推荐算法, 即(MMLHC算法), 以多层神经网络作为参数优化计算的模型, 应用Mahout库实现算法, 实验结果显示算法能有效去除原始输入数据的噪声、奇异点, 在模型的各层之间优化权重参数与偏差, 输出数据去噪平滑, 正常拟合。相似度与精确度的计算指标良好。

关键词: 推荐算法; 机器学习; 多层神经网络; 隐藏层; 可视层

中图分类号: TP312 **文献标识码:** A

Implementation of a Hybrid Recommendation Algorithm Based on Mahout

TANK Ke

(School of Computer Science, Chengdu College of University of Electronic Science and Technology, Chengdu 611731, China)

✉kevindictor@aliyun.com

Abstract: Recommendation algorithms are widely studied as the core of recommendation engine implementation, but most types of the current recommendation algorithms have limitations on the application of user behavior characteristic attribute, user population attribute, item characteristic attribute and user-item relation characteristic attribute. They generally use similarity-based calculation, or model-based calculation methods, and their feature extraction and parameter tuning depends on prior definition, with low effectiveness of parameter optimization. This paper proposes a hybrid recommendation algorithm of Multilayer Machine Learning Hybrid Compute (MMLHC). Multi-layer neural networks are considered as model of parameter optimization calculation and Apache Mahout implements the algorithms. The experimental results show that the algorithm can effectively remove the noise and singular points of the original input data, optimize the weight parameters and deviations between the layers of the model, and smooth the output data denoising and normal fitting. The index of similarity and accuracy is improved.

Keywords: recommendation algorithm; machine learning; multi-layer neural network; hidden layer; visual layer

1 引言(Introduction)

围绕互联网及其应用的各类信息技术持续地迅猛发展, 各类电子商务交易系统平台、网络学习平台、音乐社交网络应用、新闻网站、视频网站等层出不穷, 这些应用往往通过建立推荐引擎来为用户提供令其满意的个性化优质服务, 而推荐引擎的核心就是推荐算法的设计实现。目前大多数的推荐算法是基于内容的推荐算法, 基于协同过滤的推荐算法, 基于关联规则知识的推荐算法等。在这些算法中, 完成推荐

的方式各不相同。有的是通过向量化特征计算相似度, 有的是通过建立用户—物品矩阵并分解矩阵, 计算得出物品的特征向量, 还有的是通过挖掘关联规则, 由频繁项集生成强关联规则, 并计算其置信度。在这些算法的研究实现中, 大部分对于用户行为特征属性、用户人口属性、物品特征属性, 以及用户—物品关联特征属性等参数的应用方式采用了相似度计算、或模型计算等, 当与机器学习技术结合后, 学习过程中的特征提取, 以及参数的调优都依赖于事前的手工定

义,所以,存在参数优化的效率不高这一问题。因此,本文在这些算法的工作基础上,设计了一种可在学习过程中持续自动优化参数的模型,并基于此模型,提出了一种混合推荐算法(MMLHC算法)。该算法通过综合应用各个特征参数,在模型各层中以交叉关联的方式进行学习,即多次的输入与输出的关联,初次的参数输入经过学习后,产生了特征的输出,而这些输出又作为第二次的输入,继续进行学习优化,一直持续下去,直到获取最优参数为止。算法的设计实现则利用Apache开源的Mahout库中的类和自定义类,并分别在各自相同的数据集Movielens-ml-20M中运行后,计算平均绝对误差、均方根误差、准确率、召回率等指标,MMLHC算法达到了较好的推荐效果。

2 推荐算法的相关研究工作(Related studies of recommendation algorithms)

通过应用矩阵分解技术进行重构评级矩阵,并与自动编码方法结合实现预测,从而构建出推荐系统^[1]。估计分布算法^[2]则应用了真实用户信息和音频数据高级表示的推荐算法设计,从用户感兴趣的歌曲特征中捕捉个人的听力行为,实现了音乐推荐。袁正午等学者提出一种基于多层次混合相似度的协同过滤推荐算法^[3],改进了数据稀疏的性能缺陷和单一度量问题。黄震华等学者综述了排序技术与推荐算法融合的各类算法技术,对关键技术和应用进展等进行概括、比较和分析^[4],并提出未来的发展趋势。杨佳莉等学者提出一种自适应混合协同推荐算法,该算法基于张量分解计算物品间的相似度,通过短路径枚举叠加生成预测结果^[5]。曹占伟等学者提出了结合LDA主题模型的矩阵分解推荐算法^[6],解决了冷启动及数据稀疏问题。杨丰瑞等学者提出一种结合特征传递和概率矩阵分解的社交网络推荐算法^[7],实现概率矩阵因式分解,缓解冷启动并提高了算法的精确度。刘晴晴等学者提出基于SVD填充的混合推荐算法^[8],采用奇异值分解技术分解项目评分矩阵^[8],获得了优化的评分结果。

3 混合推荐算法设计与模型(Design and model of the hybrid recommendation algorithm)

传统的推荐算法中的参数调优,以及学习过程中的特征提取依赖于事前的手动定义,参数优化的效率不高,针对参数优化问题,本文提出一种多层神经网络模型,结合算法中的迭代计算解决该问题。

3.1 模型与算法设计

在机器学习领域,主要的理论基础是概率论、数理统计、线性代数和数学分析,其核心技术是数据、算法和模型的综合应用。本文所提出的MMLHC算法涉及其中的逻辑回归与神经网络,逻辑回归是数据分析中的常用算法,它输出概率估算值,并将此值用Sigmoid函数映射到[0, 1],以实现

样本分类。神经网络则是基于历史数据构建的模型,随新数据的产生进行动态优化,并应用新数据重新训练模型,以及调整参数值。

MMLHC算法所用的模型是基于多层神经网络的模型,其结构如图1所示。当该模型处于工作状态时,其权重参数是需要进行调整的,之后可以使各个单元的状态发生改变,最终让整个网络的误差函数最小化,由此可以去除输入数据的噪声,并将其作为整体来提取特征。

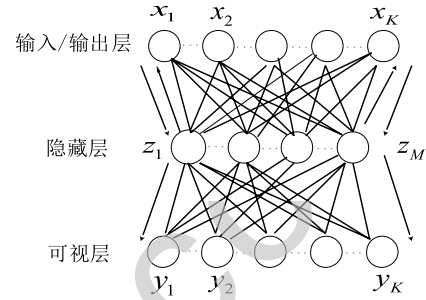


图1 MMLHC算法模型结构

Fig.1 Model structure of MMLHC algorithm

在输入数据中存在不完整、被破坏的情况,这是带有噪声的输入数据。设输入数据(含中间输入,即输出再次作为输入)单元集合 $X = \{x_1, x_2, \dots, x_K\}$,模型的最终输出(即可视层)单元集合 $Y = \{y_1, y_2, \dots, y_K\}$ 。隐藏层单元集合 $Z = \{z_1, z_2, \dots, z_M\}$ 。在输入数据中, $\tilde{x}$ 用于表示带噪声的输入数据,则隐藏层单元的输出,可视层单元的输出分别表示如下:

$$z_i = \Omega\left(\sum_{j=1}^M w_{ij} \tilde{x}_j + c_i\right) \quad (1)$$

$$y_i = \Omega\left(\sum_{j=1}^K w_{ij} z_j + b_i\right) \quad (2)$$

式(1)中, c_i 代表隐藏层的偏差, w_{ij} 代表隐藏层的权重,式(2)中, b_i 代表可视层的偏差, w_{ij} 代表可视层的权重。 Ω 是逻辑回归函数。根据算法的模型结构,输入层到隐藏层有一个映射过程,称输入带噪声的数据并映射到隐藏层的过程为编码。反之,将编码后的数据反向再变为原始输入数据的过程称为解码。则估计函数可以表示为一个对数似然函数,这个对数似然函数是关于 x_i 和 y_i 的。

$$E = -\sum_{i=1}^K \{x_i \ln y_i + (1 - x_i) \ln(1 - y_i)\} = -\ln L(\theta) \quad (3)$$

式(3)中,模型的参数由 θ 表示,它包括了可视层与隐藏层的偏差和权重。通过计算出参数的梯度,并依次求导,在此过程中,应用逻辑回归函数求导,由这些求导计算得出各个参数的更新,即权重的更新以及偏差的更新,同时还确定了参数优化迭代的次数,最终完成模型参数的计算。

为便于求导计算,应用换元法,引入两个中间变量 f_i 和 p_i ,分别替换式(1)和式(2)中的 $\Omega()$ 函数中的自变量表达式,

如下所示：

$$f_i = \sum_{l=1}^M w_{il} \tilde{x}_i + c_l \quad (4)$$

$$p_i = \sum_{l=1}^K w_{il} z_l + b_l \quad (5)$$

根据上述两个式子，原式(1)与式(2)可改写如下：

$$z_l = \Omega(f_l) \quad (6)$$

$$y_i = \Omega(p_i) \quad (7)$$

在估计函数 E 中，分别对权重、偏差这些参数进行求偏导数的运算，即求得了这些参数各自的梯度。运算过程如下：

$$\frac{\partial E}{\partial w_{il}} = \frac{\partial E}{\partial f_l} \frac{\partial f_l}{\partial w_{il}} + \frac{\partial E}{\partial p_i} \frac{\partial p_i}{\partial w_{il}} \quad (8)$$

$$\frac{\partial E}{\partial w_{il}} = \frac{\partial E}{\partial f_l} \tilde{x}_i + \frac{\partial E}{\partial p_i} z_l \quad (9)$$

$$\frac{\partial E}{\partial b_l} = \frac{\partial E}{\partial p_i} \frac{\partial p_i}{\partial b_l} = \frac{\partial E}{\partial p_i} \quad (10)$$

$$\frac{\partial E}{\partial c_l} = \frac{\partial E}{\partial f_l} \frac{\partial f_l}{\partial c_l} = \frac{\partial E}{\partial f_l} \quad (11)$$

求得各参数的梯度后，再根据 $\Omega()$ 函数的求导运算法则，如下所示：

$$\frac{d}{dx} \Omega(x) = \Omega(x)(1 - \Omega(x)) \quad (12)$$

由式(6)和式(11)联立，对 f_l 偏导数求得如下：

$$\frac{\partial E}{\partial f_l} = \frac{\partial E}{\partial z_l} \frac{\partial z_l}{\partial f_l} = \frac{\partial E}{\partial z_l} z_l(1 - z_l) \quad (13)$$

$$\frac{\partial E}{\partial z_l} = \sum_{i=1}^M \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial z_l} = \sum_{i=1}^M w_{il} (x_i - y_i) \quad (14)$$

将式(14)代入式(13)，则得到下式：

$$\frac{\partial E}{\partial f_l} = \left(\sum_{i=1}^M w_{il} (x_i - y_i) \right) z_l(1 - z_l) \quad (15)$$

同理，还可求得对 p_i 的偏导数，如下：

$$\frac{\partial E}{\partial p_i} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial p_i} = x_i - y_i \quad (16)$$

经过上述计算过程，分别求得权重参数、可视层的偏差参数、隐藏层的偏差参数等这三大参数的更新计算结果如下：

$$w_{il}^{(k+1)} = w_{il}^k + \eta \left[\left(\sum_{l=1}^M w_{il}^k (x_i - y_i) \right) z_l(1 - z_l) \tilde{x}_i + (x_i - y_i) z_l \right] \quad (17)$$

$$b_l^{(k+1)} = b_l^k + \eta (x_i - y_i) \quad (18)$$

$$c_l^{(k+1)} = c_l^k + \eta \left(\sum_{i=1}^M w_{il}^k (x_i - y_i) \right) z_l(1 - z_l) \quad (19)$$

其中， k 代表迭代的次数， η 代表学习率。

在推荐算法的设计中，对基于内容的过滤与协同过滤方法的综合应用，形成了混合推荐算法。模型的参数包含了用户评分、用户—物品关联特征、用户之间的相似度、项之间的相似度等。

MMLHC算法的实现依赖于模型的逐层训练，并在此过

程中进行参数的优化调节。逐层训练分为两个阶段，第一个阶段是预训练，第二个阶段则是精确调校。在整个网络模型中，权重参数与偏差在模型的各个层之间都需要进行调整。因此，模型中的前面层向后传递的值会成为当前层的输入，输入输出的角色动态地交替。由模型的输入层输入数据，经过隐藏层后再回到输入层，而此时的输入层就会承担输出层的作用，但同时它也依然不会抛弃输入的职责。在这个过程中，模型算法会调整网络的权重参数，同时消除误差。那么，前层学习到的特征会成为当前层的输入，所以，在整个模型中，输入数据会从逐层学习到新特征。

在模型的预训练阶段，通过权重参数的调节以更好地匹配通用特征，同时减少噪声部分的权重参数。在模型的精确调校阶段，进行有监督的机器学习，即监督训练，以解决分类问题，完成目标的推荐。MMLHC算法结合Apache Mahout库中的推荐引擎实现了混合推荐的指标计算，即相似度参数、预测准确度、均方根误差、准确率和召回率。

3.2 实验验证

算法的实验验证基于单机运行方式下完成，硬件配置如下：CPU是Intel Core i7-4790@3.60GHz，内存32GB，操作系统为Windows7-64位。软件配置，基于Maven插件的Eclipse4.11.0，配置Apache Mahout 0.13.0，单机运行，无Hadoop配置。Windows7操作系统下安装JDK12.0.1，并配置好相应的环境变量。训练与测试数据集应用Movielens-ml-20M。

在Eclipse4.11.0中运行各个自定义实现类，经过迭代优化后，调用相应的方法获取输出数据。在实验中，模型初始参数设定为隐藏层单元数20，输入/输出层单元数200，可视层单元数200，预训练阶段迭代次数最大1000，精确调校阶段迭代次数最大1000，预训练阶段学习率20%，精确调校阶段学习率15%。分别给出四组原始输入数据(即含噪声的输入数据)，运行并采集优化后的输出数据。得出图2、图3、图4、图5的四组对比，原始输入明显存在奇异点，这是包含损毁噪声的数据，而输出的结果则显示去除噪声后的平滑特征，并且与原始数据正常拟合，未有欠拟合及过拟合现象。实验结果验证了算法模型具有较好的参数优化能力。

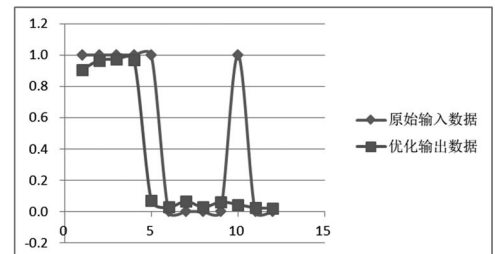


图2 数据优化对比一组

Fig.2 Data optimization and comparison (Group One)

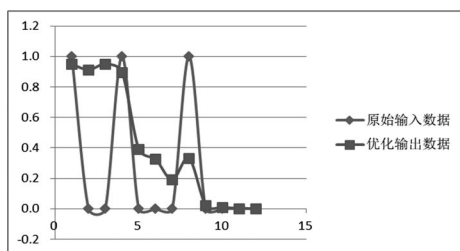


图3 数据优化对比二组

Fig.3 Data optimization and comparison (Group Two)

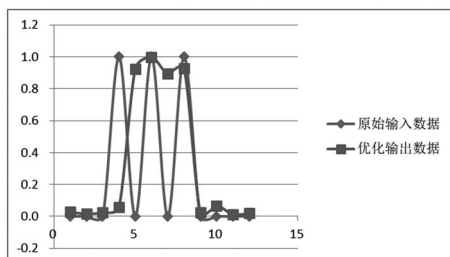


图4 数据优化对比三组

Fig.4 Data optimization and comparison (Group Three)

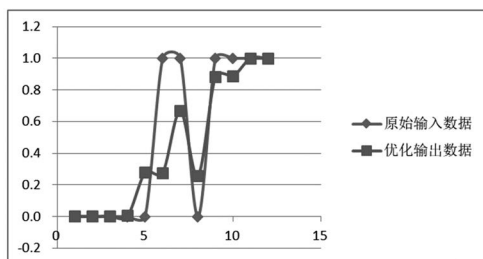


图5 数据优化对比四组

Fig.5 Data optimization and comparison (Group Four)

推荐算法的评测指标可在不同的方面反映出算法的性能。在实验中,通过相似度的计算来标明各个属性特征之间的量化相关性,并作为推荐结果的评价依据。表1分别比较了MMLHC算法、ECFA算法^[1]、CFRA算法^[3]各自的皮尔逊相关系数、余弦向量相似度、对数似然相似度的计算结果。从表中可以看出,它们的相似度指标近乎一致,至多相差1%—2%,因此,在相似度指标的评价中,这几个算法具有相同的性质。

表1 相似度指标计算

Tab.1 Similarity index calculation

算法及相似度指标(%)	皮尔逊相关系数	余弦向量相似度(%)	对数似然相似度(%)
ECFA	85.21	81.32	85.04
CFRA	84.86	83.64	84.58
MMLHC	85.31	86.16	85.52

除相似度外,均方根误差、准确率及召回率的计算比较如表2所示。表中反映出MMLHC算法相比于其他算法,均方根误差略低,准确率与召回率高出3%,性能上的提高得益于

算法模型中对参数的迭代优化,最终使得属性特征的提取数据达到了最优状态,验证了算法的较好精确度。

表2 精确度指标计算

Tab.2 Accuracy index calculation

算法及精确度指标(%)	均方根误差	准确率(%)	召回率(%)
ECFA	84.96	82.51	75.27
CFRA	85.07	83.67	77.03
MMLHC	84.35	86.37	79.62

4 结论(Conclusion)

基于MMLHC算法模型结构与Apache Mahout库结合,实现了一种混合推荐算法。算法模型建立在多层神经网络之上,通过对权重参数的迭代优化计算,排除了损毁数据对属性特征提取的干扰,降低了整个模型网络的误差,输出最优参数。通过实验验证了该算法的相关度以及精确度,达到了较好的状态。

今后的研究将进一步着眼于提高算法的性能,充分利用Mahout库支持分布式系统的特征。将Mahout库配置于Hadoop之上,可获得分布式计算的能力,在此基础上,继续改进算法的性能,并使之能满足大容量特征数据的处理需求。

参考文献(References)

- [1] F.STRUB, J. MARY, R. GAUDEL. Hybrid recommender system based on autoencoders [C]. in Proc. of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016), Boston, 2016: 11–16.
- [2] P. CHILIGUANO, G. FAZEKAS. Hybrid music recommender using content-based and social information[C]. in Proc. of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP2016), Shanghai, China, 2016: 28–38.
- [3] 袁正午,陈然.基于多层次混合相似度的协同过滤推荐算法[J].计算机应用,2018,38(3):633–638.
- [4] 黄震华,张佳雯,田春岐,等.基于排序学习的推荐算法研究综述[J].软件学报,2016,27(3):691–713.
- [5] 杨佳莉,李直旭,许佳捷,等.一种自适应的混合协同过滤推荐算法[J].计算机工程,2019,45(7):222–228.
- [6] 曹占伟,胡晓鹏.一种结合主题模型的推荐算法[J].计算机应用研究,2019,36(6):1638–1642.
- [7] 杨丰瑞,郑云俊,张昌.结合概率矩阵分解的混合型推荐算法[J].计算机应用,2018,38(3):644–649.
- [8] 刘晴晴,罗永龙,汪逸飞.基于SVD 填充的混合推荐算法[J].计算机科学,2019,46(6A):468–472.

作者简介:

唐 科(1969—),男,硕士,讲师/经济师.研究领域:移动互联应用,JavaEE软件开发,机器学习。