DOI:10.19981/j.CN23-1581/G3.2024.11.012

融合协同过滤的 XGBoost 在音乐推送上的应用研究

王方圆,张国华*
(湖南工业大学 理学院,湖南 株洲 412000)

摘 要:该文研究一种基于融合协同过滤和 XGBoost 的音乐推荐算法。首先,使用协同过滤算法计算用户之间或物品之间的相似度,从而得到初始的推荐列表,作为召回集。考虑到协同过滤法产生的音乐推荐列表还存在计算量大、稀疏性等问题,导致推荐列表并没有那么准确。接下来,对推荐列表中的每个项目进行特征提取和特征工程,并使用 XGBoost 算法对其进行预测,得到最终的推荐列表。该研究的贡献在于提出一种新的音乐推送算法,融合协同过滤和 XGBoost 算法的优点,可以得到更精准的音乐推荐列表。

关键词:XGBoost;协同过滤;推荐;应用;音乐推送

中图分类号:TP391.41 **文献标志码**:A **文章编号**:2095-2945(2024)11-0049-04

Abstract: This paper studies a music recommendation algorithm based on fusion collaborative filtering and XGBoost. First, we used a collaborative filtering algorithm to calculate the similarity between users or items and obtained an initial list of recommendations as a recall set. Considering that the music recommendation list generated by collaborative filtering method still has some problems such as large computation and sparsity, the recommendation list is not so accurate. Next, we carried out feature extraction and feature engineering for each item in the recommendation list, and used XGBoost algorithm to predict it and got the final recommendation list. The contribution of this study is to propose a new music recommendation algorithm, which combines the advantages of collaborative filtering and XGBoost algorithm to get more accurate music recommendation list.

Keywords: XGBoost; collaborative filtering; recommendation; application; music push

迄今,我国同时在线音乐活跃用户数已超 7.7 亿,网络用户大量增长,音乐作品与日俱增,音乐类别日益多元化[1]。在如此信息爆炸的时代,大量新领域下的推荐需求应运而生,音乐正是非常合适的个性化推荐产品。面对庞大的音乐库和用户多样化的偏好,如何准确地向用户推荐其可能感兴趣的音乐仍然是一个具有挑战性的问题。协同过滤算法通过分析用户历史行为和兴趣推荐相似的音乐给用户,但难以应对冷启动和稀疏数据的挑战。XGBoost 是梯度下降树的一种[2],具有处理大规模数据和高维特征的能力,并且能够建模非线性关系。本文将这 2 种算法结合起来,可以利用协同过滤的协同性和 XGBoost 的学习能力,提高音乐推送的准确性和个性化程度。

1 相关工作

1.1 协同过滤

协同过滤推荐算法使用用户和产品的交互行为[2],其主要思想是通过计算用户之间的相似度,为用户推荐与其兴趣相似的其他用户所喜欢的物品。协同过滤推荐算法可以分为基于用户的协同过滤和基于物品的协同过滤 2 种。本篇论文使用基于用户的协同过滤算法,下面介绍基于用户的协同过滤算法在音乐推荐上的应用思想。

首先,计算听众之间的相似度,可以使用 Jaccard 相似度算法,然后为目标听众推荐与其相似度较高的其他听众所喜欢的歌曲。

将每个听众的兴趣或偏好转化为文本形式。对于每个听众,对其兴趣或偏好进行分词处理,并形成一个词条集合。

根据词条集合,计算每对用户之间的 Jaccard 相似系数。具体而言,对于 2 个用户 A 和 B,计算公式为

$$J(A,B)=\frac{(A\cap B)}{(A\cup B)},$$

第一作者简介:王方圆(2001-),女,硕士研究生。研究方向为应用数学。
* 通信作者:张国华(1970-),男,博士,教授,硕士研究生导师。研究方向为计算数学与大数据方面的教学与研究。

式中：A 和 B 分别表示 2 个用户的词条集合，$A \cap B$ 表示 A 和 B 共同出现过的词条集合，$A \cup B$ 表示 A 和 B 出现过的所有不重复的词条集合。

根据计算得到的 Jaccard 相似系数，可以构造一份用户评分矩阵，其中每个元素表示对应用户之间的相似度值。

Jaccard 算法的计算过程简单明确，其结果较为容易理解和解释。相似度值的范围在 0 到 1 之间，数值越大表示用户之间的相似度越高。但是，只应用协同过滤为听众生成音乐推荐列表仍然存在数据稀疏和冷启动等问题[3]。下面将介绍 XGBoost 算法，该算法可以通过特征提取和选择，解决数据稀疏问题，将其与协同过滤相融合建立一个音乐推荐模型，可以提高模型的推荐准确度。

1.2 XGBoost 算法原理

XGBoost 针对梯度提升决策树算法的改进型集成学习算法[4]，基于历史数据，XGBoost 可以对不同的用户做不同的预测。其可以根据用户的历史进行和项目目标的特征来预测用户对其他项目的喜爱度，其特有的树结构具备从原始变量中通过组合获取隐含信息的特性[5]。

1.2.1 特征提取

通过分析用户历史播放记录、搜索记录、收藏记录等数据，提取出与用户兴趣相关的特征，例如歌曲的音乐风格、歌手信息、发布时间等。

1.2.2 构建训练集和测试集

将提取出的特征与用户对歌曲的评分作为标签，构建训练集和测试集。

1.2.3 训练模型

使用 XGBoost 算法训练模型，通过迭代和优化参数来提高预测准确率。

首先假设数据集 $T=\{(x_i,y_i)\}$，其中 $x_i \in R^m, y_i \in R^m, i=1,2,3,4,\cdots,n$。$x_j$ 为特征数据集，特征个数为 m，y_i 为标签值即目标变量，假定 XGBoost 模型中共有 k 棵树，则模型可以简单定义为

$$\hat{y}_i = F_k(x_i) = F_{k-1}(x_i) + f_k(x_i) \quad , \tag{1}$$

式中：$f_k(x_i)$ 代表第 k 棵树的预测，不难看出预测值 $\hat{y}_i$ 是模型中所有树预测的求和。

下面定义 XGBoost 的目标损失函数为

$$Obj(t) = \sum_{j=1}^{n} l\left(y_j, \hat{y}^{(t)}\right) + \Omega(f_t) \ , \tag{2}$$

式中：$\sum_{j=1}^{n} l\left(y_j, \hat{y}^{(t)}\right)$ 为第 t 次预测值下的误差函数，$\hat{y}^{(t)}$ 为第 t 次计算后的预测值，$\Omega(f_t)$ 为正则项。

树的复杂函数为

$$\Omega(f_t) = \gamma^T + \frac{1}{2}\lambda\sum_{k=1}^{T}\omega_k^{\ 2} \ , \tag{3}$$

式中：T 代表的是叶子节点的个数，ω_k 表示为决策树第 k 个叶子节点时候的输出值，$\sum_{k=1}^{T}\omega_k^2$ 则表示叶子节点上面输出分数的模平方，γ 和 λ 表示的是最终模型公式中的控制比重。

目标损失函数中的 $\hat{y}^{(t)}$ 对应的表达式为

$$\hat{y}_i^{(t)} = \sum_{k=1}^{k} f_k(x_j) = \hat{y}_j^{(t-1)} + f_t(x_j) \ , \tag{4}$$

式中：k 为决策树输出叶子节点的个数，$\hat{y}_j^{(t-1)}$ 则为 $t-1$ 次的预测值，$f_t(x_j)$ 为当前时刻预测值。

将式(3)与式(4)代入式(2)中并加以改写可以得到

$$\begin{aligned} Obj(t) &\cong \sum_{i=1}^{n}\left(g_i f_t(x_i) + \frac{1}{2}h_i f_t^{\,2}(x_i)\right) + \Omega(f_t(x_i)) \\ &= \sum_{j=1}^{T}\left[\left(\left(\sum_{i\in I_j} g_i\right)\omega_k + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)\right)\omega_k^{\,2}\right] + \lambda^T \end{aligned} \ 。 \tag{5}$$

令 $G_j=g_i$、$H_j=h_i$，此时改写后的目标函数(5)包含了 T 个独立的单变量二次函数，如下

$$Obj(t) \cong \sum_{j=1}^{T} G_j\left[\omega_k + \frac{1}{2}\left(H_j + \lambda\right)\omega_k^2\right] + \lambda^T \ , \tag{6}$$

现要求得最优解 ω_k，令 $\omega_k^{'}=0$ 可得目标函数表达式为

$$Obj(t) = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^{\ 2}}{H_j + \lambda} + \gamma^T \ , \tag{7}$$

式中：T 为叶子节点的个数，γ 和 λ 为比重系数。

1.2.4 预测结果

根据训练后的模型，对未听过的歌曲进行预测，得出推荐结果。

2 融合协同过滤的 XGBoost 音乐推荐算法的模型构建

本模型使用协同过滤算法与 XGBoost 相融合的算法进行音乐个性化推荐。具体算法流程如图 1 所示。

2.1 阶段 1：基于协同过滤的内容召回

输入：听众历史交互记录。

输出:部分听众的召回集。

2.2 阶段 2:XGBoost 模型训练

输入:听众的历史交互记录,听众的特征数据,歌曲的特征数据。

输出:XGBoost 模型。

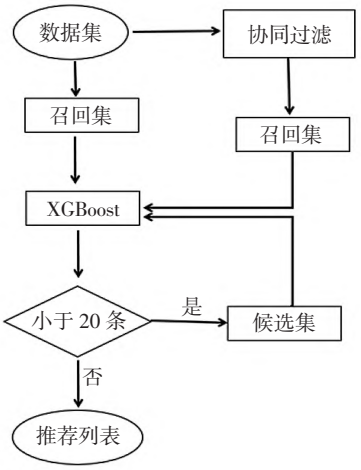

图 1 融合协同过滤的 XGBoost 算法流程

数据预处理:①提取听众交互数据,将存在点击记录的歌曲视为 1,不存在点击记录的项目视为 0。②对数据进行均衡化,使得已点击歌曲与未点击歌曲的比例为 1:1。

特征工程:提取听众的特征,并对其进行预处理,生成用户—歌曲偏好的文件。

标签定义:根据用户对音乐的播放频率,给音乐分别打上喜欢(2)、中立(1)、不喜欢(0)这 3 个标签,再标签这个字段加载到用户—歌曲偏好文件。

模型训练:使用 XGBoost 进行模型训练,并对参数进行调整,得到最终的模型。

2.3 阶段 3:产生推荐列表

输入:部分听众的召回集,原始候选集,XGBoost模型。

输出:推荐列表。

使用 XGBoost 模型对阶段一中输出的部分听众的召回集进行预测,将预测值比阈值大的歌曲作为推荐列表中的歌曲。

假若得到的推送歌曲列表少于 20 首歌曲,那么再从原始候选集中选用数据使用 XGBoost 进行预测,并将预测值比阈值大的歌曲项目作为推荐歌曲。

3 融合协同过滤的 XGBoost 推送模型在音乐推送上的实验

3.1 数据集

本文的原始数据集分为 2 部分,第一部分收集于网上的一些在线音乐播放平台,大约 223 首歌曲的基本信息。加工后的歌曲信息表包括的字段信息包括歌曲编号(song_id)、歌手(artist)、风格(style)、情感(emotion)、发布时间(publish_time)和时长(duratio)。

另一部分是用机器模拟生成的用户—歌曲交互信息,模拟了 30 组数据,每一组代表一个用户,每组 60~80 条记录,一共 2 132 条。为了贴合实际,具有普适性进而得到有效的实验结果,利用程序有意地控制了每组数据即每个用户的播放次数的占比(play_count 1 ~80 次 (20%)、80 ~220 次 (30%)、220 ~300 次 (50%))。用户歌曲交互信息表包含的字段信息包括用户编号 (user_id)、歌曲编号 (song_id)、播放次数 (play_count)。

得到原始的数据集之后,对用户歌曲信息表的播放次数(play_count)字段进行处理,把其转化为用户对歌曲的评分,得到用户—歌曲—评分的数据。用户—歌曲交互信息表这部分数据集,用于基于用户的协同过滤算法,将得到的结果从歌曲信息表中抽取出来,形成一个最初的歌曲推荐列表 A_u。

将歌曲基本信息表和处理后的用户—歌曲交互信息表拼接到一起(通过歌曲编号 song_id)得到完整的用户—歌曲信息表,将其按 4:1 的比例划分为训练集和测试集,分别用于 XGBoost 模型的训练和测试。

3.2 评价指标

本文选取精确率 Precision 作为 XGBoost 模型的评价指标,精确率越高,就表明改进融合后的推荐算法预测的音乐推荐列表可以更符合目标用户的需求。

Precision 计算公式如下

$$Precision=\frac{(C_u\cup D_u\cup E_u)}{B_u},$$

式中:B_u 为参与预测的所有音乐,C_u 为预测的喜欢的音乐集合,D_u 为预测正确的中立的音乐,E_u 为代表预测正确的不喜欢的音乐。

3.3 实验结果及分析

利用基于用户的协同过滤推荐算法得到初始目标用户的推荐列表 A_u,将 A_u 输入到训练好的 XGBoost 模型进行预测,按喜好度的高低排序后得到给目标用户的最终推荐列表 F_u。

本实验一共有 30 组数据,每组数据训练后都会得到对应某个用户特定的模型,利用 Precision 计算公

式,计算得到的实验结果如图 2 所示。

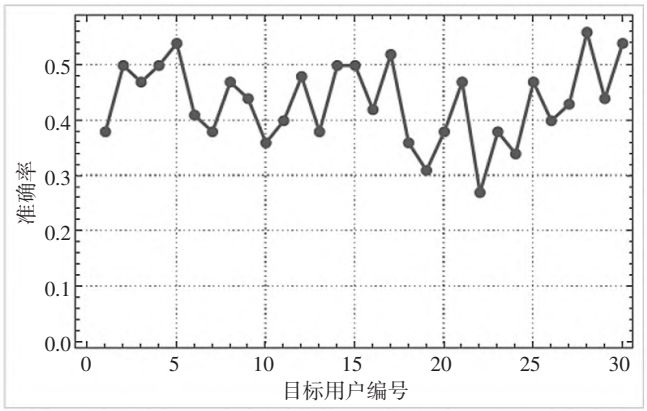

图 2 XGBoost 模型的 Precision 值

图 2 中横轴表示的是目标用户的编号，纵轴表示的是 XGBoost 模型的准确度,图中结果显示,模型的准确率在 0.25~0.60 之间波动,计算其平均准确度为 4.3。因为最终的精准度理论上由协同过滤算法叠加 XGBoost，一般来说协同过滤算法推荐精准度可以达到 40%~50%,再综合 XGBoost 大约能达到 60%~70%,可以说模型的准确度已经属于比较高的存在。

4 融合协同过滤的 XGBoost 在音乐推送上应用的创新

4.1 增加推荐准确度

协同过滤算法和 XGBoost 算法的结合可以提高模型的准确性和泛化能力,特别是当数据集比较稠密时,可以大幅提高推荐准确度。

4.2 解决数据稀疏问题

协同过滤算法在数据稀疏的情况下，推送效果会变得不稳定，而 XGBoost 算法可以通过音乐特征提取和选择,解决数据稀疏问题。

4.3 提高个性化推荐效果

协同过滤算法可以发现相似听众之间的隐藏关系,而 XGBoost 算法可以提供更准确的预测能力,从而实现更加个性化、更精准的音乐推送列表。

4.4 提高实时推荐效率

XGBoost 算法具有高效的预测能力，可以实现实时推荐,而协同过滤算法的训练和预测时间较长,结合 XGBoost 算法可以提高算法的实时推荐效率。

总之，协同过滤算法与 XGBoost 算法结合应用于音乐推荐当中，提高音乐推荐算法的准确性和泛化能力,从而实现更好的个性化推荐效果,提高听众对音乐推荐界面音乐的满意度。

5 结论

现在，音乐对于越来越多的人来说已经成为不可或缺的部分,但随着音乐库的日益增加,对于许多的听众来说,短时间内找到自己真正想要的音乐很难[6]。融合协同过滤和 XGBoost 的模型在音乐推荐中取得了显著的性能提升。

首先,与单一的协同过滤算法作为推荐算法相比,融合 XGBoost 的音乐推送模型能够更准确地捕捉用户的喜好和音乐的特征,提供更个性化的音乐推荐。通过对用户的历史行为数据和音乐的各种特征进行深入挖掘和学习,XGBoost 模型能够有效地捕捉到音乐的隐藏关联,提升推荐的准确性和多样性。

其次，融合协同过滤和 XGBoost 的模型能够有效解决冷启动和稀疏性问题。由于协同过滤算法在面对新用户或新音乐时存在不足，通过融合 XGBoost 的模型可以利用音乐的内容特征和用户的个人特征，弥补协同过滤的不足,提供更准确的推荐结果。

总之,融合协同过滤的 XGBoost 模型在音乐推荐中具有较高的推荐准确性和个性化程度，能够有效解决传统协同过滤算法的不足。然而,该模型的性能仍然与数据的质量和数量密切相关,因此,在实际应用中需要综合考虑数据收集和模型优化的问题。

参考文献:

[1] 张如琳,王海龙,柳林,等.音乐自动标注分类方法研究综述[J].计算机科学与探索,2023,17(6):1225-1248.
[2] 陈雷.面向用户交互行为挖掘的协同过滤推荐算法研究[D].合肥:合肥工业大学,2022.
[3] 陈垲冰.基于协同过滤的个性化推荐算法研究及应用[D].江门:五邑大学,2019.
[4] 谢冬青,周成骥.基于 Bagging 策略的 XGBoost 算法在商品购买预测中的应用[J].现代信息科技,2017,1(6):80-82.
[5] 邓晴元.基于 XGBoost 算法的债券违约风险预测研究[J].投资与创业,2023,34(2):1-3.
[6] 让冉,邢林林,张龙波,等.面向新领域的推荐系统综述[J].智能计算机与应用,2023,13(5):1-8,17.