

# 基于深度学习的个性化音乐推荐算法研究

王晓燕

(集宁师范学院, 内蒙古 乌兰察布 012000)

**摘要** 对于音乐推荐来讲, 将相似音乐推荐给受众仅属于一部分内容, 为用户提供多样性推荐服务, 让用户产生丰富的体验才是关键内容。对此, 本文对 Apriori 算法进行优化, 以此为基础对音乐信息关联性进行计算, 以准确推荐音乐, 希望能够为相关单位与人员提供参考。

**关键词** 深度学习 个性化音乐 推荐算法

中图分类号: TP393.09

文献标识码: A

文章编号: 1007-0745(2022)11-0097-03

现阶段, 个性化推荐已经获得广泛认可, 同时充分融入民众生活。音乐是一种人们喜闻乐见的艺术形式, 可以给民众带来娱乐, 然而在海量音乐中准确寻找符合用户要求的作品, 需要通过个性化推荐算法结合用户行为进行有效筛选, 满足用户基于当时情景的要求, 进而实现众口可调目标<sup>[1]</sup>。

## 1 优化的 Apriori 算法

### 1.1 Apriori 算法

设定音乐项目集合为  $M=\{M_1, M_2, M_3, \dots, M_n\}$ ,  $D$  (字段集合) 为项集, 子集  $T$  为  $\{t_1, t_2, t_3, \dots, t_n\}$ ,  $t_i \in I$  对各个事务标注为  $T_{id}$ ,  $D$  中项目数量是  $k$ 。

$T$  在  $D$  中出现次数在  $D$  中的比重为  $T$  支持度, 见下式:

$$Support(T_i) = \frac{|T_i \in I_1, I_1 \subseteq D|}{|D|} \quad (1)$$

支持度代表规则之中事务出现频率, 一般会设定最小阈值支持度, 如果项集支持度比阈值大, 那么变成频繁项集。

设定  $M_1 \cap M_2 = \emptyset$ , 那么称  $D$  中涵盖  $T_1, T_2, T_2$  是项集置信度, 见下式:

$$Congidence(T_1 \rightarrow T_2) = \frac{Support(T_1 \cup T_2)}{Support(T_1)} \quad (2)$$

$D$  事务矩阵如下:

$$H = \begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_m \end{pmatrix} = \begin{pmatrix} M_1 & M_2 & & M_3 \\ d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & & \vdots \\ d_{m1} & d_{m2} & \dots & d_{mn} \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_m \end{pmatrix} \quad (3)$$

进行计算时, 扫描  $D$  过程中, 若是支持度与置信度均比各自阈值大, 则是强关联规则<sup>[2]</sup>。

在运用 Apriori 算法开展关联规则计算工作时, 主要涵盖以下步骤: 首先, 数据库的扫描, 寻找项集中涵盖的频繁项集。其次, 按照频繁项集进行最小置信度构造。

### 1.2 优化 Apriori 算法

Apriori 算法优化核心内容是扫描数据库, 形成矩阵储存事务信息, 之后按照权值对所有事务支持度进行计算, 同时根据支持度计算结果对事物矩阵展开排序处理, 和阈值之间逐条对比, 若是事务支持度比阈值小, 则进行删除处理, 之后分解事务矩阵, 最后开展形成频繁项集操作<sup>[3]</sup>。

扫描音乐数据库形成  $D$  (事物矩阵) 从, 每行记录一个事务与相应权重, 之后进行排序编码处理, 将低于阈值事项删除, 使行向量  $[d_1, d_2, d_3, \dots, d_j]$  变为 0 与 1, 进而能够按照频繁项集升序对  $H$  进行分解, 变为子矩阵, 数量为  $P$  个, 即  $H_1, H_2, H_3, \dots, H_p$ 。

扫描  $H$  第  $p$  个列向量, 即  $D_p = (d_{1p}, d_{2p}, d_{3p}, \dots, d_{qp})$ , 对  $d_{ip}$  依次判断, 如果值为 1, 那么选取  $H$  中前  $p$  项, 产生  $H_p$ 。

$$H = \begin{matrix} M_1 & M_2 & M_3 & M_4 & M_5 & M_6 \\ \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \end{pmatrix} \end{matrix} \quad (4)$$

借助简单例子对具体过程中进行说明, 将支持度

★基金项目: 基于深度学习的音频数据分析定制化推荐研究, 课题编号: jsky2021026。

阈值设定为2, 音乐事务如下:  $T_1: M_1, M_2, M_6$ ;  $T_2: M_2, M_3$ ;  $T_3: M_1, M_2, M_4$ ;  $T_4: M_1, M_3, M_5, M_6$ ;  $T_5: M_2, M_5, M_6$ ;  $T_6: M_1, M_2, M_5, M_6$ ;  $T_7: M_5, M_6$ 。

按照相应公式, 根据权值能够获得  $T_i$  子集支持度,

$$\overline{\text{Support}}(T_i) = \frac{1}{k} \left( \sum_{j \in T_i} w_j \right) \times \text{Support}(T_i) \quad (5)$$

其中,  $k$  代表  $T_i$  长度。

扫描数据库, 获得事务矩阵:

$$H = \begin{matrix} & M_1 & M_2 & M_3 & M_4 & M_5 & M_6 \\ \begin{matrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix} \quad (6)$$

借助计算获得行向量与列向量, 分别为 {3,2,3,4,3,4,2}、{4,5,2,1,4,5}, 进行降序排列处理, 获得  $H'$  :

$$H' = \begin{matrix} & M_1 & M_2 & M_3 & M_4 & M_5 & M_6 \\ \text{Sup}^5 & 5 & 4 & 4 & 2 & 1 \\ \begin{matrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix} \quad (7)$$

## 2 音乐标签的信息处理

### 2.1 以用户为基础的音乐标签

音乐标签多样性、语义性特点良好, 良好音乐的标签可以充分代表音乐内容, 将音乐关键特征充分体现出来, 专家标注方式已经无法满足多样化、海量音乐内容, 当前, 一般选择用户打标签模式, 然而此种方式也存在一定问题, 即在用户数量增长过程中, 同时标准不够统一, 用户能够选择任意标签对同一音乐表达理解, 所以, 导致标签信息数量持续增长, 而有用信息降低。

针对此类音乐标签, 其可以将受众对于音乐最直接反映体现出来, 为推荐算法提供良好的依据, 但是也存在信息杂乱、不统一等问题。

### 2.2 音乐标签

因为受众标注音乐时, 存在不规范、不相关以及其他问题, 使用此类标签过程中, 需要将噪声信息删除, 保留有效信息, 可以借助多维对应方法将标签噪声删除。

对于用户日志, 一般借助听歌记录并不能够对其标注信息进行获取, 可以借助 MBID 接口获取标注信息。

开展数据采集工作时, 若是同一标签对不同音乐进行标注, 则应该自动干预, 此种现象表明不同音乐具有一定相似性, 可以借助 jaccard 相似系数对各个音乐相似度进行计算, 见下式:

$$CO_{sy}(t_i, t_j) = \frac{|t_i \cap t_j|}{|t_i \cup t_j|} \quad (8)$$

其中,  $t_i$  代表集合中的标注标签。

处理音乐标签时, 主要目的就是删除被标注标签的噪声, 对有效标签进行保留。借助多维对应分析法开展分类数据分析工作, 对底层结构数据集进行检测以及表示, 标签数据能够以分类数据角度分析, 然而对各类标签若是看作一类标签, 则会由于数据库中信息维度过高, 影响数据计算效果, 因此应该开展数据降维处理, 标签点按照降维数进行映射处理, 因此可以选择余弦相关性对 2 个维度相似性进行计算。

借助  $A_1$  代表音乐标注第一个信息,  $A_2$  代表第二个信息, 最终计算结果显示, 相应向量夹角即  $A_1$  相似度与  $A_2$  相似度, 最终能够删除弱相关性的标签。

处理音乐标签时, 借助删除无效内容, 并对标签信息中高频出现标签信息进行记录, 进而得到有效用户标注与音乐标签, 此种方式能够客观反馈音乐信息, 也是用户兴趣点。

## 3 推荐方法

受众兴趣度涵盖以下类型: 首先, 显性类型, 受众主动提供给系统, 体现出受众主观意愿。其次, 隐性类型, 借助分析受众听歌行为获取, 可以将用户听歌行为习惯体现出来。因为显性兴趣度能够直接获取, 所以主要分析隐性兴趣度。

数字音乐属于被推荐商品, 其含有商品共性, 另外个性较强, 特别对于推荐系统, 此种个性导致无法根据普遍推荐算法开展音乐推荐工作。以用户行为层面分析, 受众可以通过多样化方式操作音乐, 比如受众对于一首歌曲听的时间可以体现出受众对于该歌曲的喜爱程度, 这与推荐时借助用户购买记录获取相似商品存在一定差异。对普通受众兴趣度计算展开合理优化, 开展计算工作前, 进行几点假设:

第一, 受众会点击收听感兴趣音乐。

第二, 受众对于歌曲的操作行为体现出其在音乐方面的兴趣度。例如, 受众听取一首音乐的时长体现出其兴趣度, 若是其听完该音乐, 代表其对于此首音乐比较感兴趣; 若是收藏此首音乐, 则体现出受众非

常感兴趣。

第三, 同类受众在未知音乐中具有相同音乐度。

## 4 实验评估

### 4.1 Last.fm 数据集

Last.fm 支持受众自定义对音乐标签进行处理, 同时对受众收听信息进行记录。现阶段, 公布的信息中用户记录达到 36 万个, 随机挑选 4128 个信息, 其中音乐为 231453 首, 音乐标签为 13246 个。

### 4.2 评价标准

评价推荐结果过程中, 以 Recall (召回率) 与 Precision (准确率) 为标准, 见下式:

$$\text{Recall} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (9)$$

$$\text{Precision} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (10)$$

其中,  $T(u)$  代表原始数据集受众感兴趣音乐几何,  $R(u)$  系统推荐给受众的集合。

另外, 为了对推荐结果多样性进行验证, 选择相似度计算方法进行验证, 见下式:

$$\text{Sim}(u,r) = \sum_{t_i \in M(r)} \frac{M(t_i)}{1 + \text{count}(r \in M(t_i))} \quad (11)$$

其中,  $1 + \text{count}(r \in M(t_i))$  代表  $t_i$  音乐标签使用数,  $M(t_i)$  代表  $t_i$  所标注音乐集,  $t_i$  代表音乐标签。

### 4.3 实验结果

选择 UserCF (协同过滤算法)、聚类 UserCF 和本文方法展开召回率、准确率对比, 结果如下: 聚类 UserCF: Precision 为 72%, Recall 为 29%。UserCF: Precision 为 64%, Recall 为 20%。Apriori+: Precision 为 79%, Recall 为 26%。

通过实验结果能够发现, 准确率对比方面优于所对比方法, 比聚类 UserCF 方法召回率略低。

对于推荐算法, Playcount 算法主要是对受众潜在兴趣度着重挖掘, 采用 Playcount 方法和各项结果展开对比。

对于推荐系统, 音乐结合应该适量, 若是推荐结果较多则会导致针对性不足, 进而导致受众在推荐结果方面丧失耐心, 若是结果较少则会导致受众感兴趣内容遗漏, 所以对于推荐集合音乐数量进行设定: 25、23、20、18、15、13、10、8、5, 结果准确率, Playcount 为:

0.851、0.852、0.857、0.861、0.865、0.876、0.881、0.898、0.910。Apriori+ 为: 0.879、0.882、0.882、0.866、0.873、0.876、0.885、0.910。

结果召回率分别如下: Playcount 为: 0.26、0.25、0.24、0.23、0.18、0.16、0.13、0.08、0.03。Apriori+ 为: 0.28、0.27、0.26、0.24、0.20、0.17、0.14、0.09、0.03。

通过以上实验结果能够发现, 对于准确率, 推荐次数在 18 次以内时, 两种方式并无较大差异, 然而超出 18 次时, 本文方法推荐效果比 Playcount 方法效果突出, 体现出基于推荐数量相同条件下, 本文方法形成的推荐集合受众感兴趣音乐占比优于 Playcount。对于召回率, 本文方法优于 Playcount, 体现出本文方法形成的推荐集受众感兴趣的音乐数量比 Playcount 高。

相似度如下: Playcount 为: 0.96、0.94、0.92、0.91、0.90、0.88、0.87、0.86、0.85。Apriori+ 为: 0.83、0.83、0.82、0.80、0.80、0.79、0.78、0.76、0.75。

通过上述数据能够发现, Playcount 方法具有较高相似度, 极易出现同质化问题。但是本文方法具有较低相似度, 不仅对受众相似兴趣进行充分考虑, 同时体现出推荐多样性。

## 5 结语

综上所述, 根据受众兴趣度计算, 对推荐结果多样性进行考虑, 选择优化 Apriori 算法对目标音乐和音乐库之间相似度, 充分保证准确率, 并且提高推荐结果多样性。对音乐标签开展噪声处理, 留下有效标签。并基于受众兴趣度, 设计推荐系统流程, 有效提高推荐效果。实验选择 Last.fm 进行仿真实验, 最终结果可以满足使用要求。

## 参考文献:

- [1] 徐健雄, 何先波, 何清玉. 基于用户群体兴趣变化的个性化推荐算法研究 [J]. 西华师范大学学报 (自然科学版), 2022, 43(01): 102-107.
- [2] 欧如月, 陶宏才. 一种融合 PageRank 和 Personal Rank 的多层个性化推荐算法 [J]. 成都信息工程大学学报, 2021, 36(03): 305-310.
- [3] 沈杰, 乔少杰, 韩楠, 等. 融合多信息的个性化推荐模型 [J]. 重庆理工大学学报 (自然科学), 2021, 35(03): 128-138.