计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(11):28−35 [doi: 10.15888/j.cnki.csa.006049]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

大规模隐式反馈的词向量音乐推荐模型①

于　帅, 林宣雄, 邱媛媛

(西安交通大学 软件学院, 西安 710049)

摘　要: 现有音乐推荐系统在大规模隐式反馈场景下存在推荐困难的问题, 提出大规模隐式反馈的词向量音乐推荐模型 (Word-Embedding Based Implicit Music Recommender). 本模型借鉴了自然语言处理领域的 Word2Vec 技术, 通过学习用户音乐收藏播放记录里的歌曲共现信息, 获得用户、音乐在分布式空间的低维、紧致的向量表示, 从而得到用户、音乐之间的相似度进行推荐, 并且在理论上论述了 Word2Vec 技术应用在推荐系统上的正确性. 该模型在保证准确率和召回率几乎不变的同时, 收敛速度快, 占用内存小, 试验结果表明该模型有效的解决了大规模隐性反馈场景下音乐推荐困难的问题.

关键词: 词向量; 协同过滤; 推荐算法; Word2Vec; 深度学习

引用格式: 于帅,林宣雄,邱媛媛.大规模隐式反馈的词向量音乐推荐模型.计算机系统应用,2017,26(11):28–35. http://www.c-s-a.org.cn/1003-3254/6049.html

Implicit Music Recommender Based on Large Scale Word-Embedding

YU Shuai, LIN Xuan-Xiong, QIU Yuan-Yuan

(School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A large scale word-embedding based implicit music recommender is proposed to address the problems that most of current recommendation systems cannot work in the scenario of large scale implicit feedback recommendation. This model employs the Word2Vec technique which is popular in Natural Language Processing in recent years. By learning the songs co-occurrences in the users' history collections, we can get the distributed representation of users and songs with a low-dimension and dense vector. In this way, we can get the similarities of users and songs which could be used for the recommendation and we also analyze the correctness of application of Word2Vec technique in recommendation. This model can effectively solve the problem mentioned above with the accuracy remaining the same. In addition, this model can converge faster and take less memory than those of traditional methods.

Key words: word-embedding; collaborative filtering; recommendation algorithm; Word2Vec; deep learning

近年来, 用户越来越倾向于通过在线或流服务的方式欣赏音乐. 为了应对海量的音乐对用户造成的信息过载, 个性化音乐推荐系统正在日益增多. 传统的推荐系统主要是依据用户对商品的显式反馈进行推荐, 基于显式反馈的推荐系统的研究已经相对成熟[3,4,7]. 一方面基于隐式反馈的推荐系统的研究目前相对较少, 主要包括基于邻域的推荐和基于矩阵分解的推荐, 另一方面在音乐推荐领域隐式反馈数据容易收集, 不易引起用户的反感. 但隐式反馈数据在推荐过程中也面临样本不平衡、高维稀疏与噪声、规模大等挑战[1]. 在实际的音乐推荐系统中, 曲库的数量通常在 10^6-10^8 级别, 传统的基于邻域的推荐模型[5,6]使用稀疏的 0、1 向量表示用户、音乐, 然后求用户、音乐相似度进行推荐, 在如此大的音乐数量下, 音乐两两之间的相似度计

① 收稿时间: 2017-02-20; 修改时间: 2017-03-09; 采用时间: 2017-03-16

算需要消耗巨大的时间, 相似度的存储同样需要巨大的存储空间, 因此在实际应用场景中应用基于邻域的模型进行推荐是不现实的. 而基于矩阵分解的模型通过将用户、音乐表示为一个低维的向量, 使用用户和音乐向量的内积来表示用户对某首音乐的喜爱程度. 虽然可以在一定程度上可以缓解基于邻域的模型消耗巨大时间空间的问题, 但在这种方法的时间复杂度正比于向量维度大小的三次方, Hu 等人进一步把矩阵分解的复杂度优化到向量维度大小的平方[2], 但计算量依然很大. 除此之外, 传统的矩阵分解模型统一将用户没有选择的商品当成负反馈[2,10]来对待是不科学的, 因为用户没有选择一件商品的原因可能是多样的, 可能是用户不喜欢该商品, 也可能是用户完全不知道有那件商品存在, 因此把用户所有没选择的商品都当成是用户不喜欢的商品会引入噪声, 降低推荐系统的准确度.

面对上述问题与挑战, 本文提出词向量的音乐推荐模型对用户和音乐进行建模, 通过学习用户、音乐的共现信息, 得到用户、音乐在分布式空间的低维、紧致的向量表示, 我们可以计算出用户-音乐、用户-用户、音乐-音乐之间的相似度, 解决了数据稀疏无法推荐的问题; 利用音乐出现的频率进行负采样, 极大地排除了噪声的干扰, 解决了样本不平衡问题; 由于本模型不需要像矩阵分解模型一样对矩阵进行分解, 所以向量维度大小不会造成时间复杂度的大幅增加, 解决了数据规模大无法推荐的问题.

本模型和传统方法相比, 在保持原有准确度不变的情况下, 还具有解释性好, 覆盖率高, 收敛速度快, 内存占用小的优点. 有效的解决了大规模隐式反馈场景下音乐推荐困难的问题.

1 相关工作

根据推荐方法的不同, 推荐系统通常分为 3 类: 基于内容的推荐、协同过滤推荐和混合推荐. 基于内容的推荐是根据用户过去的行为推断用户的偏好. 协同过滤推荐是根据具有相似喜好的其他用户的行为, 推断这一类用户的使用偏好. 混合推荐是综合使用内容推荐和协同推荐两种方式[3]. 本文使用的方法属于协同过滤的推荐方法, 它的好处是模型简单, 扩展型好, 可以和使用内容的推荐的方法综合使用, 提高推荐的准确度.

1.1 基于邻域的隐式反馈模型

目前广泛研究和应用的推荐系统技术绝大多数都是基于邻域的协同过滤方法拓展或融合而来[2]. 最原始的形式是面向用户的邻域模型, 文献[4]很好的分析了这种方法. 随后基于商品的协同过滤模型[5,6]变得流行起来, 这种方法相比面向用户的邻域模型准确度高、解释性强, 使得这种方法得到了更广泛的应用. 下面以基于商品的协同过滤模型为例进行介绍.

对于大多数基于商品的协同过滤模型, 关键是获得商品之间的相似度. 在隐式反馈数据上, 使用用户来表示商品, 从而获取商品之间的相似度. 一般地我们使用余弦相似度来度量.

$$sim_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|i\| * \|j\|} \quad (1)$$

文献[6]提出了一种预先计算所有商品之间的相似度得分的模型, 但是对于 n 个商品的情形这需要 $O(n^2)$ 的存储空间. 事实上, 只需要一小部分相似的商品来进行推荐, 因此对于每一个商品 j, 只保留前 k 个最相似的商品, 其中 $k<<n$. 在预测用户 u 对商品 j 的倾向程度时, 只需要找出与 j 最相似的前 k 个商品, 然后根据用户实际购买这 k 个商品的情况, 得出用户 i 对商品 j 的预测.

这种模型是目前最常见的隐式反馈场景下的音乐推荐模型. 但这种模型使用用户来表示商品, 如果用户的数量非常大, 将产生高维向量计算, 这意味着高昂的计算开销, 在大规模隐式反馈场景下变得十分困难. 本文提出的模型将商品表示成低维向量, 有效的解决了这一问题.

1.2 基于矩阵分解的隐式反馈模型

近年来, 基于矩阵分解的模型已经逐渐取代传统的依赖基于邻域的模型而成为研究热点和主流模型. 基于矩阵分解的模型有较高的预测准确率和较好的可扩展性[1]. 不像基于邻域的模型需要计算用户-用户、商品-商品之间的相似性, 一些基于矩阵分解的模型, 如奇异值分解 (SVD) 将用户-商品评分矩阵分解为两个低维矩阵进行推荐. 矩阵分解形式化表示为: $R \approx UV^T$, 其中 R 表示用户-商品评分矩阵, U 表示用户隐语义矩阵, V 表示商品隐语义矩阵. 因此基于矩阵分解的模型具有如下优点: 利用矩阵分解降维, 处理稀疏数据提高准确率; 可以对分解的向量维度进行参数调整, 模型复杂度不会随着用户或商品数的增加线性增长.

但是传统的矩阵分解模型并不能在隐式反馈场景

下进行推荐, 对此 Hu 等人[2]将未选择的产品设置一相对小的置信度, 通过交替最小二乘法 (ALS, alternative least squares) 来拟合用户、商品向量, 从而获得推荐得分. 这种方法的时间复杂度为$O((M+N)F^3+|R|F^2)$. 其中 F 代表用户、产品向量的维度, M 代表用户的数量, N 表示音乐的数量, $|R|$表示用户收藏记录的数量. 在大数据隐式反馈推荐场景下, F 需要设置很大才能得到很好的效果, 因为在大数据集下 F 越大才能描述足够的信息. 所以 F 的阶数很高在实际推荐场景下是不现实的.

本文提出的模型受矩阵分解降维技术的启发, 将用户、音乐表示成低维的向量, 在大数据推荐场景下将模型的时间复杂度优化到$O(|R|F)$的数量级, 使大数据集推荐变得可能.

1.3 基于深度学习的推荐模型

值得一提的是, 近年来深度学习在图像识别和语音处理方面取得了显著的成果. 深度学习的本质目标就是学习到对象的特征表示, 而推荐系统的核心就是生成用户兴趣和物品的表示, 因此他们的目标是一致的. 可以预见深度学习推荐领域也会得到重要应用. 目前在推荐领域内应用深度学习还不多见. 文献[11]利用 DNN 网络预测语音信号的隐语义信息进行推荐, 文献[12]利用 DNN 网络对商品本身信息辅助协同过滤模型来提高推荐结果. 这些模型都在推荐上表现出很好的效果, 但是这些模型只是依赖数据额外的信息, 比如声音信号特征、文本评论特征、图像特征, 但是这些模型在进行推荐时还是依赖于矩阵分解的模型进行推荐. 而基于矩阵分解的推荐模型存在计算复杂度高的缺点, 很难进行大规模推荐.

本文提出的词向量音乐推荐模型使用浅层神经网络, 极大的降低了计算复杂度, 解决了数据稀疏问题.

1.4 分布式空间 Word2Vec 技术

词是承载语义的最基本的单元, 而传统的独热表示 (one-hot representation) 仅仅将词符号化, 不包含任何语义信息. 如何将语义融入到词表示中 Harris 在 1954 年提出的分布假说 (distributional hypothesis) 为这一设想提供了理论基础: 上下文相似的词, 其语义也相似. Firth 在 1957 年对分布假说进行了进一步阐述和明确: 词的语义由其上下文决定 (a word is characterized by the company it keeps) 二十世纪 90 年代初期, 统计方法在自然语言处理中逐渐成为主流, 分布假说也再次被人关注. Dagan 和 Schütze 等人总结完善了利用上下文分布表示词义的方法, 并将这种表示用于词义消歧等任务这类方法在当时被成为词空间模型 (word space model). 在此后的发展中, 这类方法逐渐演化成为基于矩阵的分布表示方法, 期间的十多年时间里, 这类方法得到的词表示都被直接称为分布表示 (distributional representation). 1992 年 Brown 等人同样基于分布假说, 构造了一个上下文聚类模型, 开创了基于聚类的分布表示方法. 2006 年之后, 随着硬件性能的提升以及优化算法的突破, 神经网络模型逐渐在各个领域中发挥出自己的优势, 使用神经网络构造词表示的方法可以更灵活地对上下文进行建模, 这类方法开始逐渐成为了词分布表示的主流方法[13].

本文提出的词向量模型借鉴了自然语言处理领域的 Word2Vec 技术[8,9], 该技术在学习文本相似度上取得了显著的成果. Word2Vec 技术主要包括 CBOW 和 Skip-gram 模型, CBOW 模型去除了上下文各词的词序信息, 使用上下文各词词向量的和或者平均值来预测上下文中的一个词, 而 Skip-gram 模型尽可能的让一个词只和它上下文相关 (距离较近) 的词相似而不和上下文不想关 (距离较远) 的词相似, 这样 Skip-gram 模型通常可以通过一个词预测上下文的单词, 通过上下文中的一个词对候选词打分, 分数即为该候选词出现的概率. CBOW 模型和 Skip-gram 模型见图 1.

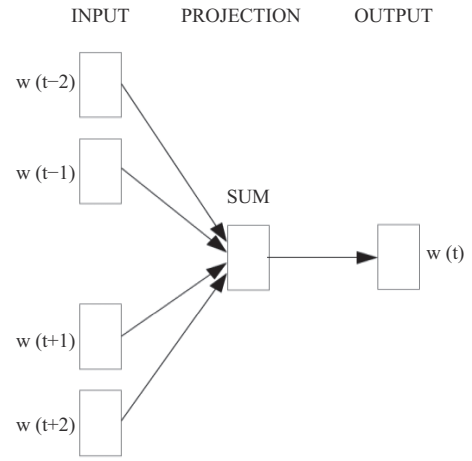
CBOW 模型和 Skip-gram 模型由于使用了较为浅层的神经网络结构, 没有隐藏层. 因此可以极大地提高训练速度从而使大数据集推荐成为可能. 本文提出的词向量音乐推荐模型是基于 Skip-gram 模型的基础之上进行推荐.

2 词向量音乐推荐模型

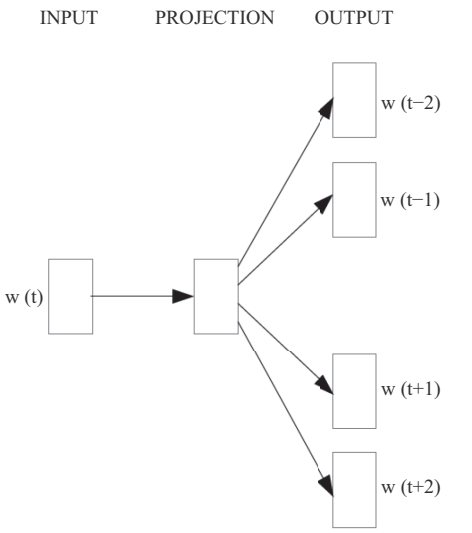
2.1 使用词向量模型对用户和音乐进行建模

受 Word2Vec 技术启发, 我们把每个用户的收藏记录类比为一个句子, 而每首音乐类比为一个词. 假设用户的收藏的音乐之间是有一定联系的可以反映出用户喜好的倾向. 而 Skip-gram 模型恰好可以让用户收藏的音乐只和它距离较近的音乐 (即用户收藏记录内的音乐) 相似, 而和其它距离较远的音乐 (即其他用户的收藏记录内且不包含此音乐) 不相似. 通过学习用户收藏记录的共现信息, 可以得到用户、音乐在低维空间的向量表示, 这些用户、音乐在分布式空间的表示也

反应出不同用户之间的对不同音乐喜好的关联. 因此我们得到的推荐结果是可以解释的, 即: 喜欢听音乐 A 的人也喜欢听音乐 B.



(a) CBOW模型



(b) Skip-gram 模型

图 1 CBOW 模型和 Skip-gram 模型

在隐式反馈推荐场景中, 输入数据可以表示为: $s:\{s_1,s_2,s_3\cdots s_n\}$, 其中 u 表示用户, s 表示用户收藏的音乐. 在词向量模型中, 我们把用户向量初始化为 $u\in R^f$, 音乐向量初始化为: $s\in R^f$, f 通常的取值为 100 到 300. 我们把初始化的向量输入到模型中, 通过训练得到的向量表示在分布式空间中如图 2 所示.

从图 2 中可以看出用户 u_1 距离 s_3 很近, u_2 距离 s_4、s_5、s_6 很近. 因此我们计算相似度的方式有三种: 用户-用户之间的相似度、用户-音乐之间相似度、音乐-音乐之间的相似度, 在 2.3 节中详细介绍.

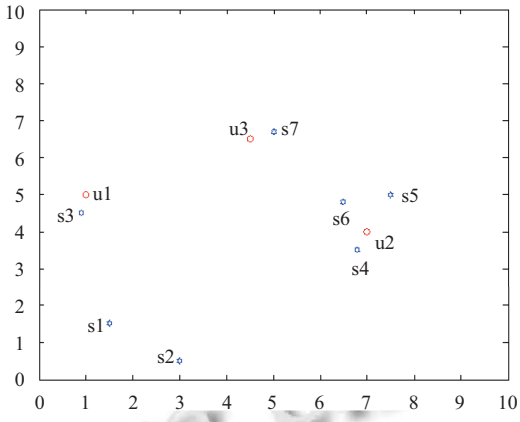


图 2 用户、音乐在高维空间的向量示意图

2.2 词向量推荐模型的理论推导

令所有音乐的集合为 M, 输入一组用户收藏音乐 $s:\{s_1,s_2,s_3\cdots s_n\}$, 由推荐系统给出一组推荐的音乐 μ. 我们的目标是最大化 G:

$$G=\prod_{m\in M} g(m) \tag{2}$$

记 $window(m)$ 表示距离 m 较近的音乐, 对于一个给定的 $(window(m), m)$, 我们希望最大化属于 $window(m)$ 的音乐概率, 同时最小化那些不属于 $window(m)$ 的音乐的概率, 即最大化下式:

$$g(m)=\prod_{n\in window(m)} p(n|m)\prod_{n\notin window(m)} 1-p(n|m) \tag{3}$$

$p(n|m)$的概率可以用以下 softmax 公式表示:

$$p(n|m)=\frac{\exp(v_n'^T v_m)}{\sum_{l=1}^{M}\exp(v_l'^T v_m)} \tag{4}$$

其中$v(.)$表示音乐的向量表示. 由于隐式反馈场景下, 只有正反馈而缺少负反馈, 把全体不属于 s 的音乐作为负反馈是不切实际的, 因为一个成熟的音乐推荐系统, 音乐数量通常是 10^6-10^8 级别. 如果要计算式 (4) 计算开销太大, 因此我们在给出正反馈的同时需要选择一定数量的音乐作为负反馈, 我们选择和窗口大小相等并且出现频率很高的音乐作为负反馈的集合记作 $neg(m)$, 那么$n\in(window(m)\cup neg(m))$.

对条件概率$p(n|m)$, 令$L^m(n)$表示给定音乐 m, n 是否为正反馈. 当 n 为正反馈时, $L^m(n)=1$, $p(n|m)$的概率为: $p(n|m)=\sigma(v(m)\theta^n)$. 同样地, n 为负反馈时, $L^m(n)=0$, $p(n|m)$ 概率为: $p(n|m)=1-\sigma(v(m)\theta^n)$, 其中 $v(m)$ 表示音乐 m 代表的词向量, σ 为 $sigmoid$ 函数, 将 $p(n|m)$整理成一个表达式:

$$p(n|m)=[\sigma(v(m)\theta^n]^{L^m(n)}[1-\sigma(v(m)\theta^n]^{1-L^m(n)} \quad (5)$$

把式 (5) 代入到式 (2) 中, 对 G 取 log 得到以下似然函数:

$$\log G=\sum_{m\in M}\sum_{n\in neg(m)\cup window(m))}\{L^m(n)\log[\sigma(v(m)\theta^n)]+(1-L^m(n))log[1-\sigma(v(m)\theta^n)]\} \quad (6)$$

将式 (6) 花括号内的内容简记为$L(m,n)$, 然后我们使用随机梯度上升法求得$L(m,n)$对于 θ^n 的梯度, 于是 θ^n 的更新公式可以写成:

$$\theta^n:=\theta^n+\alpha[L^m(n)-\sigma(v(m)\theta^n)]v(m) \quad (7)$$

接下来我们计算$L(m,n)$对 $v(m)$ 的梯度, 于是 $v(m)$ 的更新公式为:

$$v(m):=v(m)+\alpha\sum_{n\in neg(m)\cup window(m)}[L^m(n)-\sigma(v(m)\theta^n)]\theta^n \quad (8)$$

下面我们给出词向量音乐推荐算法伪代码.

算法开始:

error=0

//对窗口中的正例和负例进行迭代

For $n\in neg(m)\cup window(m)$ DO

{

$g:=\alpha[L^m(n)-\sigma(v(m)\theta^n)]$

$error:=error+g\theta^n$ //累积 error

$\theta^n:=\theta^n+gv(m)$

}

$v(m):=v(m)+\alpha*error$

算法结束.

2.3 基于词向量模型的三种推荐方法

通过上面的介绍, 我们可以训练出用户、音乐在分布式空间的向量表示. 然后我们利用训练出的向量进行音乐推荐. 基于词向量模型的推荐主要有三种方式: 基于 K 近邻用户进行推荐, 基于 K 近邻音乐进行推荐, 基于 K 近邻用户-音乐进行推荐.

基于 K 近邻的用户推荐: 这种推荐方法我们把协同过滤方法应用在训练得到的向量表示上. 首先我们选择与目标用户最相近的用户, 然后把相邻用户以前喜欢的音乐推荐给用户. 具体的步骤是首先我们选择通过向量表示计算出与目标用户最相似的 N 个用户, 然后收集 N 个用户以前喜欢的音乐, 并且过滤掉目标用户已经听过的音乐, 最后对这些音乐进行投票, 选择前 K 首被邻居用户听的最多的音乐. 推荐给目标用户.

基于 K 近邻的音乐推荐: 这种推荐方法受基于内容的推荐的方法的启发. 通过训练出的音乐向量表示, 计算出与目标用户与音乐的余弦相似度, 然后选出前 K 首最相似的音乐推荐给用户.

基于 K 近邻的用户-音乐推荐: 这种推荐方法结合上面两种推荐方式. 首先我们首先查找和目标用户最相似的 N 个用户, 然后我们查找与目标用户和最相似的 N 个用户最相似的 K 首音乐, 最后我们把这 K 首音乐推荐给用户.

2.4 模型复杂度

记音乐收藏记录的数量为$|M|$, 窗口大小为 C, 音乐向量维度为 D, 用户数量为$|U|$, 音乐数量为$|V|$, 对于每一条记录需要计算 $2\times C\times D$ 次, 对于全部音乐收藏记录需要计算 $2\times C\times D\times M$ 次, 所以模型的时间复杂度为 $O(CD|M|)$, 可以看出模型的复杂度和音乐的收藏记录的大小呈线性相关, 在实际试验中, 窗口大小增大, 推荐的准确度会增加, 但会增加计算的次数. 对于整个模型, 整体需要的空间为 $M\times D$, 模型的时间复杂度为 $O(M\times D)$. 对于大规模稀疏的数据集, $|M|>>C\times D$, 因此模型的复杂度主要和音乐收藏记录数量有关.

3 试验评估

3.1 数据集和实验方法描述

本文实验采用两个数据集, 一个是 lastfm-dataset-1k(http://www.last.fm) 数据集, 一个是我们自行抓取的虾米音乐 (http://www.xiami.com) 数据集. 我们把数据分成训练集和测试集两个部分, 两个数据集的具体描述见表 1.

表 1 数据集描述

		用户	音乐	用户-音乐
lastfm	训练集	892	121, 5104	17, 243, 881
	测试集	99	325, 192	1, 915, 086
虾米音乐	训练集	1, 980, 000	1, 215, 379	108, 437, 692
	测试集	220, 000	346, 804	12, 263, 427

为了模拟真实场景, 我们把测试集中的每个用户的历史记录按 7:3 分成两部分, 前 70% 作为用户的历史记录, 后 30% 作为正确答案. 经过模型推荐, 推荐的结果按照一定的标准排序, 作为候选集推荐给用户. 然后我们可以将推荐列表与正确答案进行对比.

3.2 实验标准

本文采用通用的 F1 分数、覆盖率以及 NDCG

(Normalized Discounted Cumulative Gain) 来描述推荐的性能指标. NDCG 是信息检索领域常用指标. DCG 可以表示为:

$$DCG_K = \sum_{k=1}^{k} \frac{rel_k}{\log_2(k+1)}$$

其中 rel_k 代表第 k 个推荐结果的质量, 我们以 0 或 1 为标准, 0 表示未命中, 1 表示命中. DCG 要惩罚排序在后面的推荐结果, 因为排序越靠后的结果贡献越小. NDCG 可以表示为:

$$NDCG_k = \frac{DCG_k}{IDCG_k} \quad (9)$$

其中 IDCG(Ideal Normalized Discounted Cumulative Gain) 表示推荐结果按理想的排序 (命中的推荐结果都排在前面) 的情况下求得的 DCG. NDCG 反映了推荐结果的质量.

3.3 实验对比算法

本文采用 2.3 节中提到的基于 K 近邻的音乐推荐算法和传统方法进行对比试验. 根据隐式反馈场景我们选择 3 种对比算法作为对比算法.

基于用户的协同过滤 User-CF: 把 User 看成一个行向量, 对于每一个 User, 我们计算其他所有 User 的行向量和他的相似度. 然后得到 K 个最相似的用户, 根据这 K 个用户的历史记录进行推荐.

基于物品的协同过滤 Item-CF: 和 User-CF 相似, 把每首音乐看成一个列向量, 我们计算其他所有音乐的列向量和他的相似度. 然后得到 K 个最相似的音乐进行推荐.

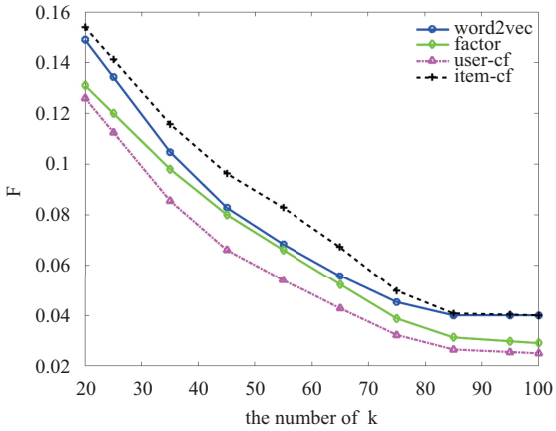
基于矩阵分解的隐式推荐模型 Factor: 把所有用户收藏的记录设置为 1, 用户未观察的设置为 0, 并按照音乐出现次数设置权重, 文献[2]将传统 0-1 矩阵分解进行推荐.

3.4 性能测试与分析

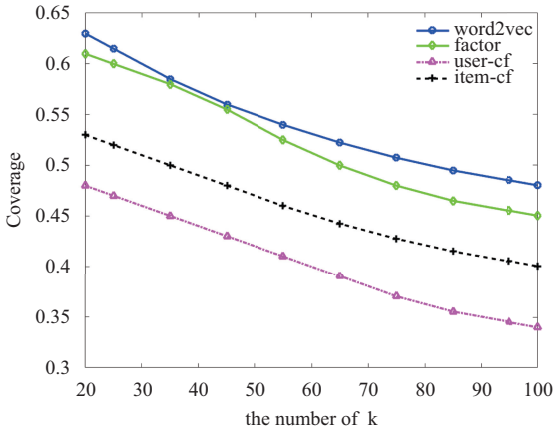
我们在一台 CPU 为 Intel E5-2650 V3, 内存为 24 GB 的服务器上进行性能测试.

实验一. 不同模型推荐性能对比.

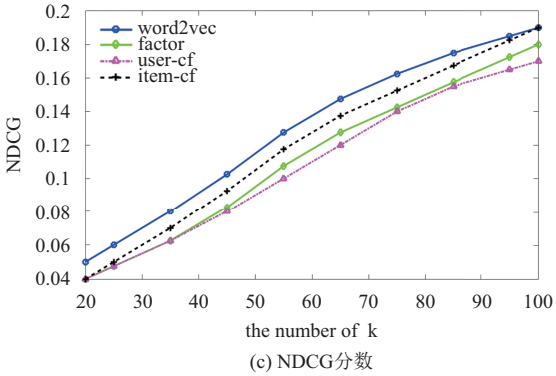
首先我们在 lastfm 数据集上测试不同算法在 K 取 20、40、60、80、100 时推荐的结果. 我们的模型参数取窗口大小为 100, 维度大小为 300, 实验对比了 User-CF、Item-CF 和 Factor(选取 F=40) 算法. 图 3(a) 横轴代表 K 值, 纵轴表示 F1 分数. 图 3(b) 横轴表示 K 值, 纵轴表示覆盖率.



(a) F1分数



(b) 覆盖率



(c) NDCG分数

图 3 4 种推荐算法在不同 k 值下的 F1 分数、覆盖率和 NDCG 分数

由图 3 可以看出 Item-CF 的效果最好, 模型和 Item-CF 的推荐效果差不多, MF 的效果优于 User-CF 的效果. 但基于邻域的模型覆盖率明显不如 MF 和词向量模型, 这是因为基于邻域的模型由于只寻找相似的用户或者音乐导致音乐重复率较高, 导致覆盖率有所下降.

实验二. 词向量模型在不同窗口大小和维度大小

下推荐性能比较.

我们在虾米音乐数据集上对比窗口大小和维度大小改变对推荐效果的影响. 图 4(a) 横轴表示值, 纵轴表示 F1 分数, 我们取窗口大小为 50、100、150、200, 维度大小为 300 做了实验, 图 4(b) 横、纵轴意义和图 4(a) 相同, 我们取维度为 100、200、300、500, 窗口大小为 150 做了实验.

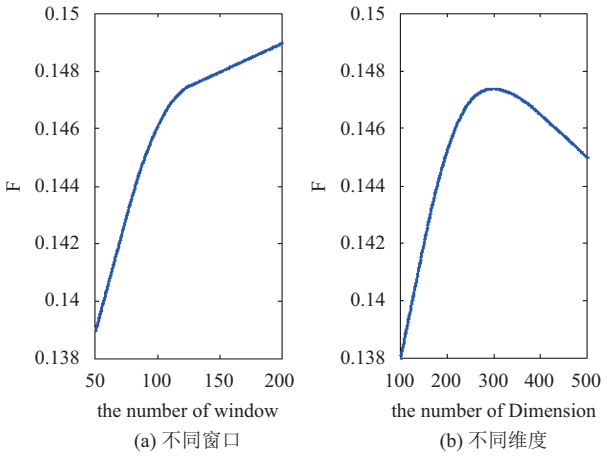


图 4 不同窗口和维度大小下 F1 分数的实验结果

由图 4(a) 可以看出当维度固定为 300 时, 随着窗口大小增大, F1 值有很大提升, 但随着窗口大小从 150 增加到 200, F1 增长趋缓. 这是因为随着窗口增大超过了用户的收藏记录的长度, 导致结果变差. 由图 4(b) 可以看出随着维度的增长 F1 分数提高, 和窗口大小不同的是当维度提高到 500, F1 分数出现了下降, 这是因为当维度过高时, 需要的训练样本的数据量也很大, 如果训练数据不足就会导致拟合效果差. 这也反映出本文提出的词向量推荐模型, 不需要很高的维度即可拟合出很好的效果.

实验三. 不同模型时间复杂度和空间复杂度对比.

在大规模隐式反馈场景下, 很多算法因为计算时间或空间巨大, 不能在实际中应用, 因此我们需要验证我们的模型在大数据集下是否有效. 我们在虾米音乐的训练数据集按照 1/5、2/5、3/5、4/5、1 的比例进行分割, 然后记录各模型在不同大小数据集上需要的时间和空间大小. 图 5(a) 横轴代表训练数据集大小, 纵轴代表训练所花时间, 图 5(b) 横轴代表训练数据及大小, 纵轴代表训练所占内存.

由图 4 可以看出词向量推荐模型训练速度明显要快于其他模型, 在内存空间使用上, 词向量模型占用的空间始终和训练数据呈线性增长, 而基于邻域的模型在数据集越来越大的情况下已经不能处理. 由图可以看出词向量模型在该数据集上训练时间是 factor 模型的 1/6, 空间大约是 factor 模型的 1/3.

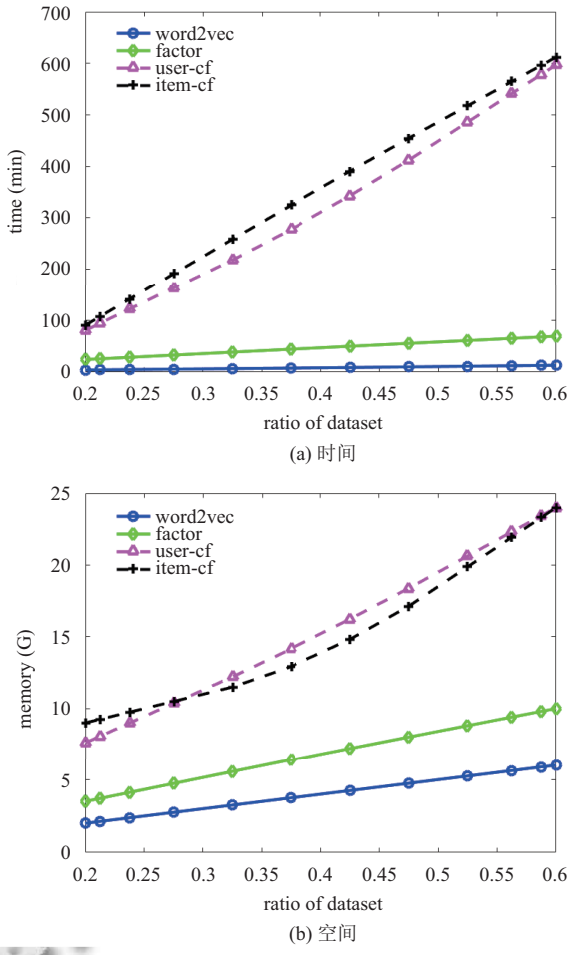


图 5 不同数据集大小下 4 种模型所需的时间和空间实验结果

实验四. 词向量模型在不同稀疏度数据集下的推荐性能.

我们将虾米音乐数据集取部分数据集, 形成一个稀疏度为 95% 的子数据集, 虾米音乐原数据集的稀疏度为 99.9%. 分别在这两个数据集上进行推荐性能测试. 在稀疏度 99.9% 的数据集上 F1 分数为 0.148, 在稀疏度为 95% 的数据集上 F1 分数为 0.14, F1 分数提高了 5%, 结果表明词向量推荐模型没有因为稀疏度提高而降低性能, 反而因为记录的增加提高了推荐性能.

4 结语

随着大数据时代的到来, 许多现有推荐模型不能

对海量数据进行推荐或者需要耗费较高的空间和时间进行推荐. 在实际的音乐推荐场景中也不可能要求用户对每首音乐进行评分, 因此传统的基于显式评分数据的推荐模型无法进行推荐. 本文提出的基于词向量的音乐推荐模型解决了大数据和隐式反馈推荐中存在的样本不平衡、高维稀疏与噪声、规模大的问题, 通过浅层神经网络学习歌曲之间的共现信息, 将用户、音乐表示为低维、紧致的向量进行推荐. 得到了与现有推荐模型基本持平的效果, 相比传统推荐模型, 词向量推荐模型极大的降低了时间复杂度和空间度使大规模隐式推荐变得可能.

本文提出的词向量音乐推荐模型由于本身采用神经网络的特性, 需要更多的训练数据才能得到更好的推荐效果, 也可以考虑和其他模型融合以期得到更好的推荐效果下一步我们将具体研究与其他模型融合时的模型. 选择、融合策略和实验效果等问题, 从而更好的提高推荐性能.

参考文献

1 印鉴, 王智圣, 李琪, 等. 基于大规模隐式反馈的个性推荐. 软件学报, 2014, 25(9): 1953–1966.

2 Hu YF, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. Proc. of the 8th IEEE International Conference on Data Mining. Pisa, Italy. 2008. 263–272.

3 彭飞, 邓浩江, 刘磊. 加入用户评分偏置的推荐系统排名模型. 西安交通大学学报, 2012, 46(6): 77–78, 86.

4 Herlocker JL, Konstan JA, Borchers A, *et al*. An algorithmic framework for performing collaborative filtering. Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Berkeley, California, USA. 1999. 230–237.

5 Linden G, Smith B, York J. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 2003, 7(1): 76–80. [doi: 10.1109/MIC.2003.1167344]

6 Sarwar B, Karypis G, Konstan J, *et al*. Item-based collaborative filtering recommendation algorithms. Proc. of the 10th International Conference on World Wide Web. Hong Kong, China. 2001. 285–295.

7 Salakhutdinov R, Mnih A. Probabilistic matrix factorization. Proc. of the 20th International Conference on Neural Information Processing Systems. Vancouver, British Columbia, Canada. 2007. 1257–1264.

8 Le Q, Mikolov T. Distributed representations of sentences and documents. Proc. of the 31st International Conference on Machine Learning. Beijing, China. 2014. 1188–1196.

9 Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. arXiv:1301.3781, 2013.

10 Lee DD, Seung HS. Algorithms for Non-negative Matrix Factorization. Proc. 13th International Conference on Neural Information Proc. Systems. Cambridge, MA, USA. 2000. 556–562.

11 Van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. Proc. 26th International Conference on Neural Information Proc. Systems. Lake Tahoe, NV, USA. 2013. 2643–2651.

12 Wang H, Wang NY, Yeung DY. Collaborative deep learning for recommender systems. Proc. of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Sydney, NSW, Australia. 2015. 1235–1244.

13 来斯惟. 基于神经网络的词和文档语义向量表示方法研究 [博士学位论文]. 北京: 中国科学院大学, 2016.