

基于社交上下文和神经网络的增量推荐算法

邹 锋

(广州商学院 信息技术与工程学院, 广东 广州 511363)

摘 要: 传统协同过滤推荐系统在数据变化剧烈的情况下需要重新训练全部数据集, 在大数据场景下产生巨大的计算开销。因此, 提出一种考虑社交上下文和神经网络的增量协同过滤推荐算法。在预处理阶段将用户的个人社交档案、显式反馈信息和隐式反馈信息建模为序列格式的上下文, 设计基于序列的相似性度量方法; 将词嵌入模型和协同过滤系统结合, 利用上下文预测用户偏爱的项目列表。实验结果表明, 该算法的推荐结果实现了较好的排列满意度、新颖性和多样性, 支持增量的系统更新处理。

关键词: 协同过滤; 词嵌入; 神经网络; 推荐系统; 相似性度量; 增量训练

中图分类号: TP391 **文献标识号:** A **文章编号:** 1000-7024 (2020) 10-2825-07

doi: 10.16208/j.issn1000-7024.2020.10.021

Incremental recommendation system based on sequence similarity measurement and neural networks

ZOU Feng

(School of Information Technology and Engineering, Guangzhou College of Commerce, Guangzhou 511363, China)

Abstract: Traditional collaborative filtering recommendation systems need to re-train the whole datasets when drastic changes happen, which leads to huge computational overhead in the big data scenario. In view of this, an incremental collaborative filtering recommendation system based on sequence similarity measurement and neural networks was proposed. In the offline pre-processing phase, the user profile, explicit feedback information and implicit feedback information were modeled as sequence formatted contexts, and a similarity measurement method between the sequences was designed. The word embedding model and collaborative filtering system were combined, the item lists preferred by target user were predicted according to the context, to improve the ranking quality, novelty and diversity of the recommendation lists. Experimental results indicate that the recommendation results of the proposed system achieve good ranking quality, novelty and diversity, and it supports incremental system updating process as well.

Key words: collaborative filtering; word embedding; neural networks; recommendation system; similarity measurement; incremental training

0 引 言

经典的协同过滤推荐系统通常使用 k -近邻 (k -nearest neighbor, k -NN) 技术^[1] 寻找和目标用户 (或项目) 最相似的 top- k 用户 (或项目), 然后为目标用户推荐其近邻用户偏爱的项目。矩阵分解^[2] 是一种推荐系统的经典方法, 大数据情况下矩阵运算的效率较低。基于内存的推荐系统又分为基于用户的推荐方式和基于项目的推荐方式^[3], 基于项目的推荐方式比基于用户推荐方式的准确率高, 前者倾向于向用户推荐相似的项目, 而这会引起严重的“马太

效应”^[4]。

为了在保证高推荐准确率的前提下, 降低推荐项目的马太效应, 许多学者设计了改进算法以提高推荐结果的多样性、新颖性以及推荐列表满意度。文献 [5] 需要迭代地寻找推荐列表的帕累托最优解, 虽然实现了理想的指标, 但是其计算效率较低。与文献 [5] 属于同一类型的算法还有: 基于多目标粒子群的系统^[6]、基于布谷鸟搜索的系统^[7]和基于狮群优化算法的系统^[8]等。另一方面, 通过结合成熟的聚类技术来提高推荐的多样性也是一种有效手段, 文献 [9] 设计了基于模拟退火的聚类技术对项目进行精准

收稿日期: 2019-11-21; 修订日期: 2020-01-15

作者简介: 邹锋 (1979-), 男, 湖南衡阳人, 硕士, 讲师, 研究方向为计算机软件与应用。E-mail: zouyudg@126.com

©1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

的分类处理,通过模拟退火的优化技术保证了聚类结果的多样性,由此提高推荐结果的多样性。与文献[9]属于同一类型的算法还有:基于去中心化聚类的推荐系统^[10]、基于增量回归模型的推荐系统^[11]、基于关联规则和聚类处理的推荐系统^[12]等。上述推荐系统均能够有效提高推荐效果,但计算时间较长,每当数据发生较大变动均需要重新训练全部数据集,无法满足推荐系统的实时性需求。

词嵌入模型在许多应用领域中表现出良好的性能,如词嵌入在文本情感分类方面的应用^[13]。神经网络语言模型(neural network language model, NNLM)是一种性能突出的词嵌入方法,能够精确地根据上下文预测出当前词。本文将用户个人档案、显式反馈信息和隐式反馈信息作为上下文,利用 NNLM 预测特定上下文偏爱的项目列表。将 NNLM 运用于推荐系统需要解决两个难题:首先,需要建模 NNLM 的上下文结构,本文设计了层次用户档案模型,并提出相应的快速相似性度量方法;再者,神经网络的训练计算量较大,为此本文设计了一种增量学习技术,加快神经网络的训练速度。

1 建立词嵌入的上下文

1.1 建立用户档案序列

每个用户表示为一个项目集和一个评分集,通过以下步骤为每个用户建立一个上下文序列:

步骤 1 分析用户评分项目的信息,将这些边际信息补充到用户的个人档案内。

为每个项目关联一个函数 e , 其返回值是该项目相关的元素集,表示为函数 $e: I \times R \rightarrow I \times T^k$, k 为每个项目相关的元素数量, T 表示元素。

步骤 2 将项目集转化为元组形式。

该步骤表示为函数 $t: I \times T^k \rightarrow I \times Z^k$ 。该步骤将所有数据转换为整型符号。

步骤 3 将整型符号排列成序列。

直接按照项目 ID 将项目排列成序列。该步骤表示为函数 $s: (\{i_j, (n_{jk})_k\}_j) = ((n_{jk})_k)_{j=1}^{|I|}$ 。

序列生成函数 f 视为上述 3 个函数的级联组合,表示为 $f = s \circ t \circ e$ 。例如:电影《让子弹飞》的 ID 为 1, 用户 u 对它评分 5 分。使用函数 e 获取该电影的类型信息,发现电影 ID=1 有两个类型:类型=7, 10, 函数 e 的输出为元组 $(1, \{\langle \text{剧情}, 5 \rangle, \langle \text{探险}, 5 \rangle\})$ 。使用函数 t 获得这些元组的符号序列,然后将电影类型转换成 ID 形式,产生新的元组 $(1, \{75, 105\})$ 。最终,仅需要分析用户 u 评分的项目元组,即 $(75, 105)$ 。

1.2 提取用户间相同子序列

不同用户的档案序列之间可能存在大量重复的元素,因此从 n 个给定的序列中提取出最长的相同子序列 $\sum =$

$(\sigma_1, \dots, \sigma_s)$ 。设序列 α 的子序列为 β , β 由 α 的元素 $[1, |\beta|]$ 构成。采用动态规划求解该问题,如算法 1 所示,该算法逐渐填充 $(m+1) \times (n+1)$ 矩阵的元素, m 和 n 分别为序列 x 和序列 y 的长度,其计算式为

$$L[i, j] = \begin{cases} 0, & \text{if } i = 0 \text{ 或 } j = 0 \\ L[i-1, j-1] + 1, & \text{if } i, j > 0 \text{ 或 } x_i = x_j \\ \max(L[i, j-1], L[i-1, j]), & \text{if } i, j > 0 \text{ 或 } x_i \neq x_j \end{cases} \quad (1)$$

式中: $L[m, n]$ 记录了两个序列间的最大相同子序列长度。原序列的时间复杂度和空间复杂度为 $O(mn)$, 提取子序列后的空间复杂度降至 $O(m)$ 或者 $O(n)$ 。根据两个序列的最长相同子序列评价两个用户的相似性。

算法 1: 基于序列的相似性度量算法(SSM)。

输入: 序列 x , 序列 y 。

输出: $L[m, n]$

(1) $L[0 \dots m, 0 \dots n] = 0$;

(2) for $i = \{1 \dots m\}$ do

(3) for $j = \{1 \dots n\}$ do

(4) if $x_i = y_j$ then

(5) $L[i, j] = L[i-1, j-1] + 1$;

(6) else

(7) $L[i, j] = \max(L[i, j-1], L[i-1, j])$;

(8) endif

(9) endfor

(10) endfor

为了运用 SSM 算法度量推荐系统的相似性,提出了基于 SSM 的上下文正则化算法,如算法 2 所示。算法 2 对算法 1 进行了两点修改:引入变换函数 f 和匹配阈值 δ 。其中, δ 越高说明相似用户的差异越大,该阈值根据具体应用场景设定;变换函数则为 1.1 小节介绍的 f 函数。

算法 2: 推荐系统的 SSM 计算算法。

输入: 用户 u , 用户 v , 变换函数 f , 阈值 δ

输出: $L[m, n]$

(1) $(x, y) \leftarrow (f(u), f(v))$

(2) $L[0 \dots m, 0 \dots n] \leftarrow 0$;

(3) for $i = \{1 \dots m\}$ do

(4) for $j = \{1 \dots n\}$ do

(5) if $\text{match}(x_i, y_j, \delta)$ then

(6) $L[i, j] = L[i-1, j-1] + 1$;

(7) else

(8) $L[i, j] = \max(L[i, j-1], L[i-1, j])$;

(9)

endif

(10) endfor

(11) endfor

1.3 相似性归一化处理

SSM相似性的范围为 $[0, \min(|f(u), f(v)|)]$, 但协同过滤系统的相似性指标取值范围一般为 $[-1, 1]$ 或 $[0, 1]$ 。通过式(3)的归一化技术处理SSM相似性值

$$\text{sim}^{f,\delta}(u,v) = \text{SSM}(u,v,f,\delta) \quad (2)$$

$$\text{sim}^{f,\delta}_{\text{norm}}(u,v) = \frac{\text{sim}^{f,\delta}(u,v)^2}{|f(u)| \cdot |f(v)|} \quad (3)$$

SSM算法的计算成本较高, 在推荐之前离线计算SSM相似性, 推荐系统可以透明地利用这些相似性值。

2 Top-N 推荐系统

2.1 Top-N 推荐系统的模型

假设 u 为用户集 U 中的一个用户, 推荐系统的目标是从项目集 I 中选出推荐给 u 的列表。协同过滤推荐系统利用用户-项目的交互数据, 如评分、评论和浏览时间等。将用户 u 对项目 i 的评分设为 r_{ui} , 未评分的情况 r_{ui} 记为0。 I_u 表示用户 u 评分的项目集, U_i 表示对项目 i 评分的用户集。Top-N推荐的目标是为用户提供 N 个项目的列表。

2.2 基于内存的推荐系统

协同过滤技术依赖相似的用户集或项目集, 常用 k -近邻(k -NN)技术寻找和目标用户(或项目)最相似的top- k 用户(或项目)。基于用户(或项目)推荐系统的输出定义为

$$\gamma_u = \sum_{v \in V_u} s_{uv} r_{uv} \quad (4)$$

$$\gamma_i = \sum_{j \in I_i} s_{ij} r_{ij} \quad (5)$$

式中: s 表示用户(或项目)之间的相似性, 即第1小节的SSM, r_{ui} 和 r_{uj} 分别表示用户 u 对项目 i 和 j 的评分。

3 基于词嵌入的推荐系统模型

3.1 词嵌入表示

prod2vec^[14]是一种基于项目嵌入的产品推荐模型, 该模型利用了矩阵分解技术, 无法运用于基于内存的推荐系统。本文设计了新的词嵌入推荐系统, 同一个用户评分的项目之间具有一定的相关性, 而对同一个项目评分的用户之间也存在一定的共性。下文以基于项目的协同过滤系统为例, 基于用户的推荐系统与之相似。本文将连续词袋模型(CBOW)^[14]和协同过滤top-N推荐系统结合, 将推荐系统项目作为CBOW的词, 用户个人档案(以及显式、隐式反馈信息)作为CBOW的上下文, CBOW根据上下文预测其偏好的词。利用CBOW模型主要出于两点考虑: ①其计算效率较高, 高于Skip-gram等模型; ②推荐系统存在稀疏性问题和冷启动问题, CBOW模型对输入上下文平均化处理, 产生平滑的概率估计, 能够缓解评分信息不足的情况。

图1所示是本文推荐系统的神经网络结构。网络隐层学习训练样本集, 每个训练样本是一个目标项目和上下文数据, 上下文是对该项目评分的用户相关信息。每个用户的训练样本数量设为 $|I_u|$, 输入层由上下文向量 $\{x_1, \dots, x_i, x_{i+1}, \dots, x_{|I_u|}\}$ 组成。

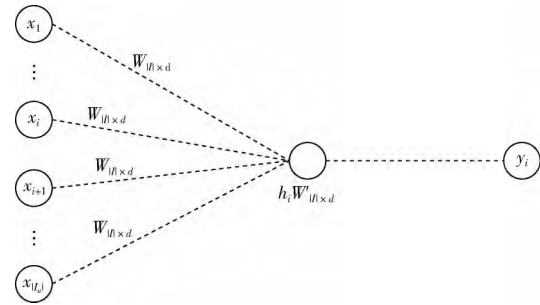


图1 嵌入协同过滤系统的主要结构

隐层神经元数量设为 d , 隐层采用线性激活函数。输入层和隐层间的连接权重表示为矩阵 $W \in \mathbb{R}^{|I| \times d}$, 矩阵每一行 v_j 表示项目 j 第 d 维的输入向量。项目 i 的隐层输出设为 h_i , h_i 的计算式为

$$h_i = \frac{W}{\sum_{j \in I_u} r_{uj}} \sum_{j \in I_u} r_{uj} x_j = \frac{\sum_{j \in I_u} r_{uj} v_j}{\sum_{j \in I_u} r_{uj}} \quad (6)$$

输出层的神经元数量为 $|I|$, 输出层采用softmax激活函数。隐层和输出层间的权重设为 $W' \in \mathbb{R}^{d \times |I|}$, 权重每一列 v'_i 是项目 i 的 d 维输出向量。输出层的输入为 $v'_i{}^T h$, 输出值 y 是目标上下文的后验分布。采用softmax激活函数计算后验分布的概率, softmax激活函数的计算式为

$$p(i | I_u) = y_i = \frac{\exp(v'_i{}^T h_i)}{\sum_{j \in I} \exp(v'_j{}^T h_i)} \quad (7)$$

通过最小化以下的损失函数来训练神经网络

$$\text{Loss} = - \sum_{v \in V} \sum_{i \in I_u} \log p(i | I_u) \quad (8)$$

网络训练程序通过后向传播学习矩阵 W 和 W' , 对于每个训练样本softmax函数均需遍历所有的项目, 所以大数据问题学习 W' 的计算开销十分巨大。采用负采样加快训练速度, 将输出词 i 的向量加上其它项目的少量随机样本, 这些样本称为负样本, 记为 \mathcal{N} 。采用负采样的损失函数为

$$p_{\text{ns}}(i | I_u) = y_i = \sigma(v'_i{}^T h_i) \prod_{j \in \mathcal{N}} \sigma(v'_j{}^T h_i) \quad (9)$$

式中: $\sigma(x) = 1/(1 + e^{-x})$ 为logistic函数, n 为 \mathcal{N} 的长度。式(9)的训练时间独立于项目的数量。

3.2 正则化处理

单隐层网络中失活正则比 L_2 正则的效果好, 因此采用失活处理解决过拟合问题。在训练阶段, 运用失活机制, 随机失活一定数量的单元, 失活概率为 p 。因为本文主要学习矩阵 W 的权重, 因此在测试阶段无需正则处理。

3.3 推荐系统的实现方法

(1) 失活机制在推荐系统的实现

原 CBOW 模型的输入是文本语料库, 输出是词向量, 窗口参数 w 控制上下文的词数量。本文建立一个上下文文档表示每个用户的信息, 为了将全部用户的信息作为上下文, 窗口 w 设为 w_{max} 来覆盖全部的用户信息。因为将全部用户档案的平均值作为输入, 所以无需考虑档案的顺序。

本文修改超参数 w , 在输入层增加失活处理。如果 w 小于用户档案序列的最大长度, 那么系统将一部分信息作为上下文, 将其它信息失活。该 w 参数和失活率 p 成比例关系。

(2) 推荐系统的相似性度量

使用式 (3) 计算用户 (或项目) 间的相似性, 采用 k -NN 寻找用户 (或项目) 的近邻。本文采用 NN-descent^[15] 代替传统的 k -NN 算法, NN-descent 的计算效率远高于经典的 k -NN 算法。

3.4 神经网络的增量训练方法

推荐系统首先使用后向传播技术随机初始化矩阵 W 和 W' , 学习这两个矩阵的参数。这不是凸优化问题, 因此通过梯度下降法寻找其局部最优解。实际应用中数据连续不断地发生变化, 随时可能加入新的评分信息、新的用户或者新的项目, 本文设计了增量的系统更新方法, 从而避免重新训练神经网络。当加入新项目的时候, 在 W 中增加行, 在 W' 中增加列, 随机初始化新加入的行和列, 因为矩阵内其它的向量接近局部最优解, 所以系统的收敛速度较快。

4 实验与结果分析

实验的硬件环境为: Intel 酷睿 i7 9700 K, CPU 主频为 3.6 GHz, 8 个核心, 16 GB 内存。软件环境为: Window 10 操作系统, 基于 Python 编程实现推荐算法的算法。

4.1 实验方法

(1) 实验数据集和预处理

采用从 GroupLens 网站获取的 MovieLens 数据集进行实验验证, Movielens_20M 数据集由 20 000 263 个评分构成, 评分范围从 0.5 星到 5 星。为了支持半星的评分, 本文将项目的 ID 乘以 100, 评分值乘以 10, 例如: 用户 u 对项目 1 的评分为 3.5 星, 那么语句预处理的结果为 $1 \times 100 + 3.5 \times 10 = 135$ 。表 1 所示是实验数据集的基本信息, 随机划分 80% 的子数据集作为训练集, 剩下的 20% 作为测试集, 分别独立划分 5 次, 完成 5 折交叉验证。推荐列表的数量设为 5, 即 top-5 推荐系统。

(2) 性能评价指标

本文考虑了多个指标全面地评价推荐系统多个方面的性能。

1) 排列质量和推荐准确率

采用 4 个指标评价推荐系统的推荐准确率, 分别为归一化折损累计增益 nDCG^[16]、精度 P^[16]、召回率 R^[15] 和平均精度均值 MAP^[16]。nDCG 用来衡量排序质量, 精度 P 和召回率 R 分别用来衡量推荐项目的命中率和查全率。

表 1 实验数据集的基本信息

数据集	用户数量	项目数量	评分数量	密度
MovieLens_20M	138 493	26 744	20 000 263	0.540%

均精度均值 MAP^[16]。nDCG 用来衡量排序质量, 精度 P 和召回率 R 分别用来衡量推荐项目的命中率和查全率。

2) 新颖性:

EPC (expected popularity complement)^[17]: 定义为期望的新出现推荐项目数量。

EPD (expected profile distance)^[17]: 推荐项目和用户偏爱项目间的距离。

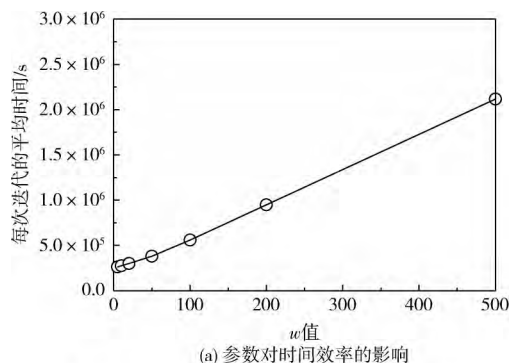
3) 多样性:

α -nDCG^[18]: 多样性感知的排列指标。

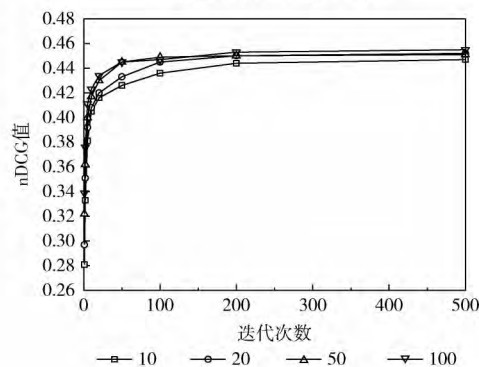
EILD (expected intra-list diversity)^[18]: 期望的列表内多样性。

4.2 神经网络超参数的选取

通过 5 折交叉验证训练本文神经网络模型, 训练以最大化 nDCG 为目标, 每组实验完成 100 次迭代。因为 w 是本文网络的关键超参数, 通过试错法调节 w , 将神经网络对 MovieLens_20M 数据集进行实验, 结果如图 2 所示。图中显示随着 w 值的升高, 每次迭代所需的时间升高, 且 nDCG 收敛曲线的质量也提高, 当 $w=50$ 之后, nDCG 收敛曲线质量的提升幅度较小, 但处理时间升高明显, 所以选取 $w=50$ 为最佳参数。



(a) 参数对时间效率的影响



(b) 参数对性能的影响

图 2 神经网络 w 参数的实验

最终获得神经网络对于 Movielens_20M 数据集的最佳超参数值, 见表 2。

表 2 神经网络超参数设置

参数	值
k	75
d	300
w	50
i	500
n	10

4.3 SSM 相似性验证实验

本文提出了新的 SSM 相似性度量指标, 一方面该指标的计算效率较快, 另一方面该指标的精度也较好。采用经典基于内存的协同过滤推荐算法和不同的相似性度量指标集成成为不同的推荐系统, 采用每个推荐系统独立地完成实验。本文 SSM 的参数 δ 设为 5 和 10, 将两种参数的指标分别简记为 SSM1 和 SSM2, γ 参数统一设为 10。选择 3 个常用相似性度量指标和本文方法比较, 分别为余弦相似性 Cosine、Jaccard 相关性和 JMSD^[19]。

图 3 所示是每个相似性度量指标对推荐系统的影响实验结果。图中可见余弦相似性和 Jaccard 的效果十分接近, 且明显好于 JMSD 指标。本文指标受所选取参数的影响较大, 总体而言, SSM2 的效果好于 SSM1, 且略好于 Cosine 指标和 Jaccard 指标, Cosine 指标和 Jaccard 指标均为广义的距离度量方法, 难以对推荐系统所需的细节信息做深度开发, 因此在推荐系统的应用效果差于本文的相似性度量方法。SSM2 引入了匹配阈值 δ , δ 越高说明相似用户的差异越大, 由于 Movielens_20M 数据集的数据量较大, $\delta=10$ 的效果好于 $\delta=5$ 。本文将选用 SSM2 作为相似性度量方法。

4.4 推荐系统的性能

测试本文推荐系统在静态完全训练下的推荐性能。将本文推荐系统与下面的算法进行比较, 评估各算法的性能:

SocialMF^[20]: 结合用户之间的信任关系, 学习用户和项目的潜在特征向量, 是一种基于矩阵分解的方法。

SocialSVD^[21]: 根据人际关系中的六度分隔理论计算用户之间信任度, 填充用户信任矩阵, 是一种基于奇异值分解的方法。

MIF^[22]: 根据互信息评价相似性, 是一种基于内存的协同过滤推荐算法。

RNNN^[23]: 通过循环神经网络根据过去的交互信息预测用户未来的偏好项目, 是一种基于会话相似性的协同过滤算法。

TIUI^[24]: 通过分析隐式反馈信息建立信任模型, 并且设计了信任传播机制, 是一种基于内存的协同过滤推荐算法。

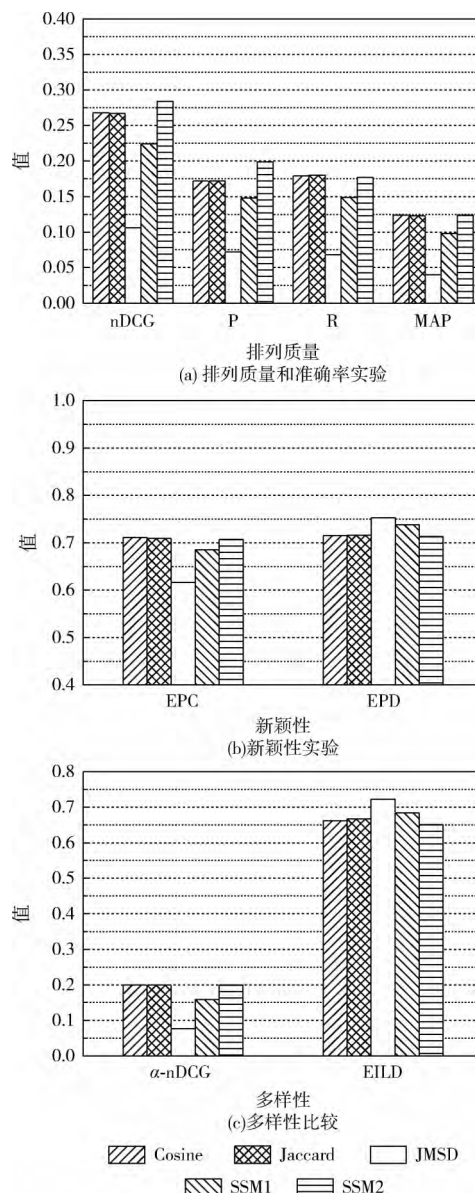


图 3 SSM 相似性实验

图 4 所示是各个推荐系统对于 Movielens_20M 数据集的实验结果。MIF 和 RNNN 均实现了较好的排列质量和推荐准确率, 本文算法略高于这两个算法。SocialMF 和 SocialSVD 是两个以分析社交关系为核心的推荐系统, 由于 Movielens_20M 数据集中包含的社交关系极少, 所以这两个算法并未获得理想的效果。观察新颖性实验结果和多样性实验结果, 本文算法的新颖性和多样性均实现了显著的提高, 将图 4 和图 3 比较, 可以发现本文词嵌入模型有效地提高了基于内存推荐系统的新颖性和多样性。

4.5 增量推荐实验

支持增量地更新神经网络模型是本文算法的一个特色, 因此通过实验评估本文增量推荐系统的性能。将数据集随机划分为 4 个大小相等的分区, 首先对第 1 个分区进行完

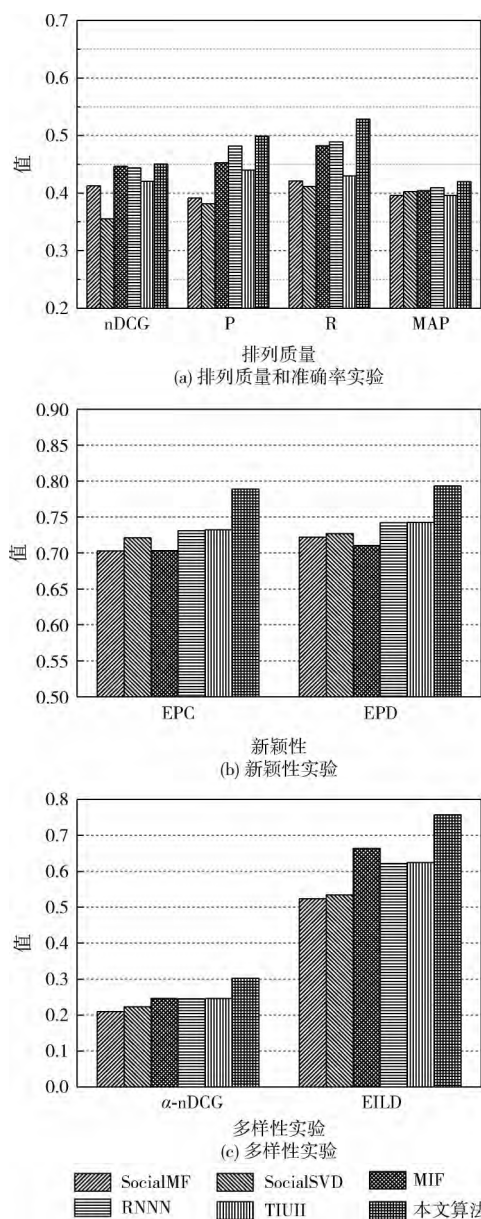


图4 静态推荐系统实验

全训练,并且对模型进行验证实验。然后将第2个分区和第1个分区结合,进行重新训练(重新训练的迭代次数为400次),该模型的测试结果记为“重新训练”模型;另外以第1个分区训练的网络为基础,采用增量训练技术训练第2个分区的模型,分别对增量学习20次、40次、60次迭代的网络进行了测试实验。第3个分区、第4个分区采用和第2个分区相似的方式完成增量训练实验,结果如图5所示。

图5中可看出,随着训练数据分区的增加,系统的性能逐渐提高;随着迭代次数的增加,神经网络的性能也明显提高。比较不同迭代次数的nDCG值,可发现增量训练的模型在40次迭代已经和重新训练的模型十分接近,而重

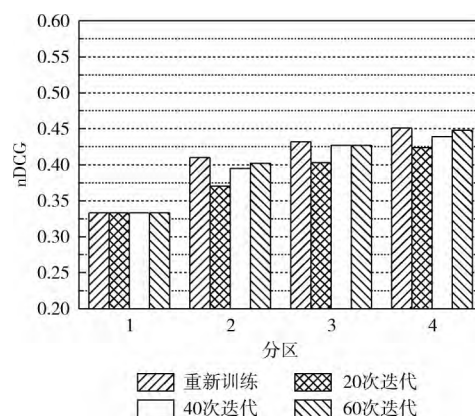


图5 增量推荐实验

新训练需要300次迭代,因此本文的增量训练方法能够节约巨大的时间开销。

5 结束语

本文在离线预处理阶段,将用户的个人档案、显式反馈信息和隐式反馈信息建模为序列格式的上下文,并设计了基于序列的相似性度量方法。将用户个人档案、显式反馈信息和隐式反馈信息作为上下文,利用NNLM预测特定上下文偏爱的项目列表。本文的增量训练能够通过较少的迭代次数即可达到局部最优解,在牺牲少量推荐性能的情况下,大幅度地降低了训练时间,能够较好地解决用户和项目变化剧烈的推荐问题。本文系统目前仅考虑了基本的反馈信息,未来将重点研究时域信息、社交关系信息等和本文推荐系统的集成方法。

参考文献:

- [1] GUO Yudong, GUO Zhigang, CHEN Gang, et al. Recommendation method based on k nearest neighbors using data dimensionality reduction and exact Euclidean locality-sensitive hashing [J]. Journal of Computer Applications, 2017, 37 (9): 2665-2670 (in Chinese). [郭喻栋, 郭志刚, 陈刚, 等. 基于数据降维与精确欧氏局部敏感哈希的k近邻推荐方法[J]. 计算机应用, 2017, 37 (9): 2665-2670.]
- [2] HAO Yaxian, SUN Yanrui. K-nearest neighbor matrix factorization for recommender systems [J]. Journal of Chinese Computer Systems, 2018, 39 (4): 755-758 (in Chinese). [郝雅娴, 孙艳蕊. K-近邻矩阵分解推荐系统算法[J]. 小型微型计算机系统, 2018, 39 (4): 755-758.]
- [3] Aditya P H, Budi I, Munajat Q. A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for E-commerce in Indonesia: A case study PT X [C] // International Conference on Advanced Computer Science & Information Systems, 2017: 1-6.
- [4] Hao W, Wang Z, Zhang W. Quantitative analysis of Matthew

- effect and sparsity problem of recommender systems [C] // IEEE International Conference on Cloud Computing & Big Data Analysis, 2018: 20-25.
- [5] Karabadi N E I, Beldjoudi S, Seridi H, et al. Improving memory-based user collaborative filtering with evolutionary multi-objective optimization [J]. *Expert Systems with Applications*, 2018, 98 (15): 153-165.
- [6] Priyanka Choudhary, Vibhor Kant, Pragya Dwivedi. A particle swarm optimization approach to multi criteria recommender system utilizing effective similarity measures [C] //The 9th International Conference. ACM, 2017: 63-71.
- [7] Han L Q, You Y H. Improving performance of free space optical communication using an adaptive gain blind optimization algorithm [J]. *Journal of Optoelectronics Laser*, 2015, 26 (5): 857-861.
- [8] Wang T, Manogaran G, Wang M. Framework for social tag recommendation using lion optimization algorithm and collaborative filtering techniques [J]. *Cluster Computing*, 2019, 11 (2): 1-11.
- [9] Mostafa A Shehata, Mohammad Nassef, Amr A Badr. Simulated annealing with levy distribution for fast matrix factorization-based collaborative filtering [J]. *International Journal of Advanced Computer Science & Applications*, 2017, 9 (4): 501-508.
- [10] Tran C, Kim J Y, Shin W Y, et al. Clustering-based collaborative filtering using an incentivized/penalized user model [J]. *IEEE Access*, 2019, 1 (99): 1-1.
- [11] Nilashi M, Jannach D, Ibrahim O B, et al. Clustering and regression-based multi-criteria collaborative filtering with incremental updates [J]. *Information Sciences*, 2015, 293 (1): 235-250.
- [12] Najafabadi M K, Mahrin M N, Chuprat S, et al. Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data [J]. *Computers in Human Behavior*, 2017, 67 (C): 113-128.
- [13] YU Haiyan, LU Huijuan, ZHENG Wenbin. Feature weighting method based on part of speech embedding for sentiment classification [J]. *Computer Engineering and Applications*, 2017, 53 (22): 121-125 (in Chinese). [于海燕, 陆慧娟, 郑文斌. 情感分类中基于词性嵌入的特征权重计算方法 [J]. *计算机工程与应用*, 2017, 53 (22): 121-125.]
- [14] ZHANG Jing, ZHU Guobin. Hot topic discovery research of Stack Overflow programming website based on CBOW-LDA topic model [J]. *Computer Science*, 2018, 45 (4): 208-214 (in Chinese). [张景, 朱国宾. 基于 CBOW-LDA 主题模型的 Stack Overflow 编程网站热点主题发现研究 [J]. *计算机科学*, 2018, 45 (4): 208-214.]
- [15] David C Anastasiu, George Karypis. Fast parallel cosine k-nearest neighbor graph construction [C] //6th Workshop on Irregular Applications: Architecture and Algorithms. IEEE Computer Society, 2016: 93-98.
- [16] ZHU Chunyang, GUO Xiaotong, SUN Haoran, et al. Many-objective optimization algorithm based on nadir point search and nondominated sorting [J]. *Journal of Chinese Computer Systems*, 2018, 39 (1): 32-36 (in Chinese). [朱春阳, 郭晓彤, 孙浩然, 等. 基于极值点搜索和非支配排序的高维多目标优化算法 [J]. *小型微型计算机系统*, 2018, 39 (1): 32-36.]
- [17] Gouzay A, Epplin F M. Land requirements, feedstock haul distance, and expected profit response to land use restrictions for switchgrass production [J]. *Energy Economics*, 2016, 58 (1): 59-66.
- [18] Enrique Amigó, Damiano Spina, Jorge Carrillo-de-Albornoz. An axiomatic analysis of diversity evaluation metrics: Introducing the rank-biased utility metric [C] //The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. ACM, 2018: 120-128.
- [19] Stephen SC, Xie H. Measures of similarity in memory-based collaborative filtering recommender system: A comparison [C] //Multidisciplinary International Social Networks Conference, 2017: 1-8.
- [20] ZHANG Yihao, LI Liang, ZHAO Qinghua, et al. Personalized recommendation algorithm of social network based on global similarity [J]. *Computer Engineering*, 2018, 44 (10): 210-214 (in Chinese). [张艺豪, 李梁, 赵清华, 等. 基于全局相似度的社交网络个性化推荐算法 [J]. *计算机工程*, 2018, 44 (10): 210-214.]
- [21] Yan Menghan, Shang Wenqian, Li Zhenzhong. Application of SVD technology in video recommendation system [C] // IEEE/ACIS 15th International Conference on Computer and Information Science. IEEE, 2016: 256-259.
- [22] Ren Y, Li G, Zhang J, et al. The maximum imputation framework for neighborhood-based collaborative filtering [J]. *Social Network Analysis & Mining*, 2014, 4 (1): 207-208.
- [23] Jannach D, Ludewig M. When recurrent neural networks meet the neighborhood for session-based recommendation [C] //Eleventh Acm Conference on Recommender Systems, 2017: 306-310.
- [24] Pal B, Jenamani M. Trust inference using implicit influence for item recommendation [C] //International Conference on Signal-image Technology & Internet-based Systems, 2018: 21-28.