

doi:10.11835/j.issn.1000-582X.2020.11.006

# 基于标签辅助边优化的推荐算法

蔡彪, 陈润

(成都理工大学 信息科学与技术学院, 成都 610059)

**摘要:**传统的二部图推荐模型只考虑了用户和物品的历史交互行为。为了提供更加准确、多样和可解释的推荐,需要在用户-物品交互式建模的基础上充分考虑标签辅助信息及权值的计算方式。文中提出了基于自然语言处理的标签相似性辅助边优化的推荐算法(LWV)。该方法结合用户历史行为和标签辅助信息,通过 word2vec 在节点间生成新用于节点交互的边并构建边的权重,来更新基础推荐算法的推荐列表。最后,在公开数据集上对文中算法与基准算法在 6 个公共评测标准进行对比,实验结果表明,LWV 更新过的推荐算法相比原算法在准确性、多样性和新颖性方面获得更好平衡。

**关键词:**推荐;自然语言处理;word2vec;标签;辅助信息

**中图分类号:**TP181

**文献标志码:**A

**文章编号:**1000-582X(2020)11-052-11

## Recommendation algorithm based on label auxiliary edge optimization

CAI Biao, CHEN Run

(College of Information Science and Technology, Chengdu University of Technology, Chengdu 610059, P. R. China)

**Abstract:** The traditional bipartite recommendation model only considers the historical interaction behavior of users and items. In order to provide more accurate, diverse and interpretable recommendations, it is necessary to fully consider the label auxiliary information and the calculation method of weights on the basis of user-item interactive modeling. This paper proposed a recommendation algorithm (LWV) based on natural language processing for tag similarity auxiliary edge optimization. This method combined user historical behavior and tag assistance information to generate new edges for node interaction between nodes through word2vec and constructed the weight of the edges to update the recommendation list of the basic recommendation algorithm. A comparison between this algorithm and the benchmark algorithm in six public evaluation standards on the public data set shows that the updated recommendation algorithm of LWV achieves a better balance in terms of accuracy, diversity and novelty than the original algorithm.

**Keywords:** recommendation; natural language processing; word2vec; tags; auxiliary information

荐系统的成功使其在 Web 应用程序中日益普及<sup>[1]</sup>。近年来,个性化推荐已在电影<sup>[2]</sup>、视频<sup>[3]</sup>、论文引用<sup>[4]</sup>、地点<sup>[5]</sup>等方面获得了广泛的应用<sup>[6]</sup>。到目前为止,已经提出了多种个性化推荐算法<sup>[7-9]</sup>,为了从用户的

**收稿日期:**2020-07-21

**基金项目:**国家自然科学基金资助项目(61802034,61701049);四川省软科学研究项目(2019JDR0117)。

Supported by National Natural Science Foundation of China(61802034,61701049) and Soft Science fund of Sichuan Province(2019JDR0117).

**作者简介:**蔡彪(1973—),男,博士,教授,主要从事人工智能、社交媒体挖掘、推荐系统等方向研究,(E-mail)caibiao@cdut.edu.cn.

历史行为来预测用户的偏好,人们投入了许多研究,如协同过滤(CF)。尽管 CF 方法具有有效性和普遍性,但它无法对物品属性、用户配置文件和各种辅助信息进行建模,因此,在用户和物品交互较少的情況下表现不佳。为了解决这些问题,提出了各种各样的推荐方法,如,知识图谱<sup>[10-12]</sup>、热传导<sup>[13]</sup>和质量扩散<sup>[14]</sup>。Google 的 PageRank 算法<sup>[15]</sup>根据网页间的超链接随机跳跃被用来进行网页排名,受到 PageRank 算法的启发,基于推荐系统二部图的随机游走算法 PersonalRank<sup>[15]</sup>被开发出来,传统的 PersonalRank 算法认为二部图中相关性高的节点具有以下特征:1)两个节点之间有很多路径相连;2)连接两个节点之间的路径长度都比较短;3)连接两个节点之间的路径不会经过出度比较大的节点<sup>[16]</sup>。

虽然,PersonalRank 可以通过随机游走算法对推荐结果进行比较好的理论解释,但该算法在时间复杂度上有明显的缺点。因为,一是在为每个用户进行推荐时都需要在整个二部图上进行迭代,直到图中每个节点的访问概率值收敛;二是该算法对图中节点间边的权重简单依靠有无交互设为 1 或者 0,导致算法推荐准确性不高<sup>[17]</sup>。

文中在 PersonalRank 算法的基础上提出了一种新的 LWV 模型,该模型通过引入自然语言处理,计算商品标签辅助信息在词向量空间中的相似性来生成新的用于交互的边<sup>[18]</sup>,根据其向量空间中的相似性和用户物品历史行为赋予权重,最终在目标用户的三部图中随机游走获取新的推荐物品以更新其他算法的推荐列表。

## 1 方法

### 1.1 标签辅助推荐模型

基于邻域的算法是推荐系统中最基本的算法,该算法不仅在学术界得到了深入研究,在业界也得到了广泛应用。基于邻域的算法分为两大类,一类是基于用户的协同过滤算法(UCF),另一类是基于物品的协同过滤算法(ICF)。LWV 算法实现过程的第一步是通过任意基础推荐算法获取目标用户的前  $nL$  ( $L$  在本算法中设为 50,  $n > 1$ ) 个物品推荐值,并将原推荐值作为边的权重,如图 1 所示。

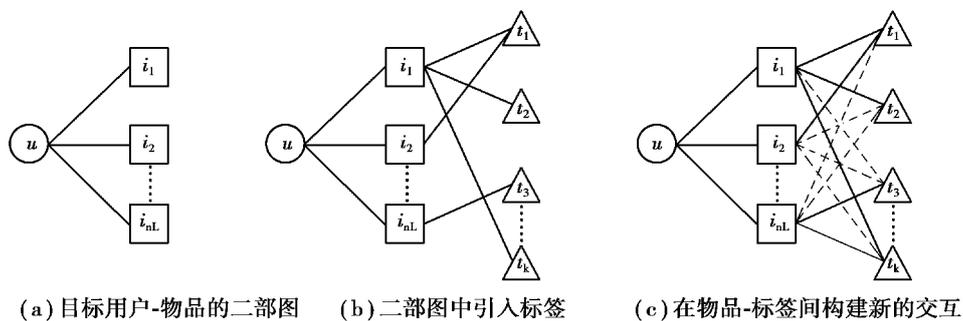


图 1 LWV 三部图模型

Fig. 1 LWV three-part graph model

图 1 是 LWV 方法构建三部图的流程图。图 1(a)构建目标用户-物品的二部图,并赋予  $e(u, i)$  权重;图 1(b)在二部图中引入标签;图 1(c)在物品-标签两两间构建新的交互并由 word2vec 计算标签相似性结合用户历史行为赋予  $e(i, t)$  权重。

通过目标用户与基础算法推荐的  $nL$  个物品来构建了用户-物品二部图,接着通过在用户物品二部图中引入标签节点建立三部图模型。令  $G(u, I, T)$  表示用户-物品-标签三部图。其中,节点  $V = u \cup V_i \cup V_t$  由单个目标用户  $u$ , ICF 推荐的  $nL$  个物品集合  $V_i$ , 和这  $nL$  个物品在标签数据集中对应的标签集合  $V_t$  组成。对于目标用户  $u$  和物品节点  $V_i$ , 图中都有一条对应的边  $e(u, i)$ , 边的权重  $W_{u,i} = A(u, i)$ ,  $A(u, i)$  是任意推荐算法给出的对于目标用户  $u$ , 物品  $i$  的推荐值。对于图中任意二元组  $(i, t)$ , 图中都有一条对应的边  $e(i, t)$ , 值得注意的是用户和标签之间没有交互边。

### 1.2 边的权重计算

赋予三部图中的物品-标签之间的边  $e(i, t)$  权重。word2vec 模型计算的是词与词之间的余弦相似性, 范围为  $-1 \sim 1$ 。在建立的模型中, 为了最后在整张三部图上进行随机游走推荐, 边的权重范围必须是一个正

数,且  $e(u, i)$  权重范围是  $0 \sim 1$ , 与之对应, 运用式(1)将相似性范围缩放到  $0 \sim 1$  之间:

$$y = \frac{\omega_{tt'} - \min}{\max - \min}, \quad (1)$$

其中,  $\min = -1, \max = 1, \omega_{tt'}$  为 word2vec 计算出的标签  $t$  和标签  $t'$  的相似性;  $y$  是缩放后标签  $t$  和  $t'$  的相似性。

传统的图随机游走算法只考虑图中节点间有无交互而构建节点之间边的权重, 文中将标签辅助信息引入传统的用户-物品二部图, 并基于用户历史行为和辅助信息生成新的节点间交互关系构建更加有说服力的权重是成功推荐的重要因素。标签的辅助信息采用在词向量空间中的相似性来为模型在图中构建新的交互关系提供理论依据。如图 1(b) 中, 实线是数据集中已有的行为交互关系, 图 1(c) 中的虚线是 LWV 模型生成的新的交互关系。比如, 用户购买了一个物品  $a$ , 物品  $a$  被其他用户打上了“鼠标”这个标签, 而物品  $b$  有一个“键盘”标签, 而“键盘”, “鼠标”两个词在文本上的相似性是比较高的, 此时可以生成两条边  $e(a, \text{“键盘”})$  和  $e(b, \text{“鼠标”})$ , 并根据“键盘”, “鼠标”的相似性赋予这两条边一个权重。虽然,  $e(a, \text{“键盘”})$  和  $e(b, \text{“鼠标”})$  的权重可能远远小于  $e(a, \text{“鼠标”})$  和  $e(b, \text{“键盘”})$ , 但是许多条边的生成最终会对最后的推荐列表产生影响。这种处理方式可以更好地挖掘数据间深层次的关系, 并使推荐结果具有可解释性。更详细的如图 1(c) 中对于物品  $i_1$  在数据集中被用户打上了  $t_1, t_2, t_k$  3 种标签, 那么对于  $i_1$  这个物品, 它的标签集合为  $N_{i_1} = \{t_1, t_2, t_k\}$ , 此时, 标签  $t_3$  不在集合中, 模型要生成  $i_1$  到  $t_3$  的新边, 该边的权重要同时受  $N_{i_1}$  中所有的元素影响, 具体如式(2)所示:

$$S_{i,t} = \begin{cases} n_{i,t} t \in N_i, \\ \frac{\sum_{t' \in N_i} (\omega_{tt'} + 1) * n_{i,t'}}{2m_i} t \notin N_i, \end{cases} \quad (2)$$

其中:  $\omega_{tt'}$  是标签  $t$  和  $t'$  之间通过 word2vec 计算出来的相似性;  $m_i$  是  $N_i$  的长度;  $n_{i,t}$  是物品  $i$  被打上标签  $t$  的次数, 一个物品被打上相同标签次数越多, 证明该标签更加符合这个物品的本质。在数据集中有些标签并不准确, 因此, 通过引入  $n_{i,t}$ , 惩罚了这些错误标签对推荐结果的影响。

$$S'_{i,t} = \frac{S_{i,t}}{\max(S_{i,t})}, \quad (3)$$

同样的, 需要将  $S_{i,t}$  缩放到  $0 \sim 1$  之间。

### 1.3 排名更新

在传统二部图的基础上, 文中构建了用户-物品-标签三部图推荐模型, 通过引入标签辅助信息, 在物品-标签间生成新的用于随机游走的带权重的边。

在构造好三部图后, 通过随机游走算法来确定最终推荐列表。由于每个用户都拥有一个唯一的三部图, 随机游走算法计算所有物品节点相对于用户节点的重要度, 按照重要度的高低生成推荐列表, 因此, 本质是一个图上节点的排名问题。模型要计算用户  $u$  的三部图中其余节点相对于用户  $u$  的重要度, 则从用户  $u$  对应的节点开始执行图随机游走算法: 每到一个节点都以  $1-d$  的概率停止游走并从  $u$  重新开始, 或者以  $d$  的概率继续游走, 确定游走概率后从当前节点指向的节点中按照均匀分布随机选择一个节点赋予部分重要度。这样经过很多轮游走之后, 每个节点被访问到的概率都会收敛并趋于一个稳定值<sup>[19]</sup>。

$$PR(v) = \begin{cases} d \sum_{v' \in \text{out}(v)} \frac{PR(v')s}{|\text{out}(v')|} & (v \neq v_u), \\ (1-d) + d \sum_{v' \in \text{out}(v)} \frac{PR(v')s}{|\text{out}(v')|} & (v = v_u), \end{cases} \quad (4)$$

式中:  $PR(v)$  是节点  $v$  的访问概率(重要度);  $v_u$  是用户节点;  $d$  是用户继续访问网页的概率,  $d$  值在 [20] 中 被认为值为 0.85 时效果最好;  $\text{out}(v)$  表示与  $v$  有边相连的节点集合。在每次游走的初始, 用户的  $PR$  值设置为 1, 其余节点的  $PR$  值均设置为 0。

文中设置最大迭代次数为 100, 在图中游走完所有边算一个迭代, 在图中迭代 100 次后每个节点的概率都趋于稳定, 最后取重要度最高的前  $L$  个物品节点作为最终推荐列表。

模型通过基础算法获取单个目标用户推荐值前  $nL$  ( $n > 1$ ) 的物品,再由 LWV 算法更新该  $nL$  物品列表中每个物品的推荐系数,对推荐列表进行重新排列来获取最终的长度为  $L$  的推荐列表。

## 2 评价指标

为了测试算法性能,将每个实验数据集按照 9:1 随机分为两部分:训练集和测试集。对于每个用户,根据训练集提供推荐列表,并在测试集上测试性能。为了量化推荐的效果,在此应用 6 个广泛使用的指标<sup>[21]</sup>,包括 3 个准确性指标(AUC, Precision, Recall), 2 个多样性指标(Hamming, Intra-similarity)和 1 个新颖性指标(Novelty)。

准确性是评估推荐算法质量的最重要指标之一。首先,介绍 AUC<sup>[22]</sup>(ROC 曲线下的面积),给定测试集中的物品推荐指数,AUC 的值可以解释为随机选择的推荐物品的排名高于随机选择的未推荐物品的概率。为了计算 AUC,每次选择一对已推荐和未推荐的物品以比较它们的推荐值。经过  $N$  次独立比较后,如果推荐物品中有  $N_1$  次具有更高推荐值,而  $N_2$  次其推荐值相同,则将所有用户的 AUC 平均值定义为<sup>[23]</sup>

$$AUC = \frac{1}{m} \sum_{i=1}^m \frac{(N_1 + 0.5N_2)}{N}. \quad (5)$$

较大的 AUC 值意味着较高的算法准确性。然后,介绍 2 个与  $L$  相关的精度指标,即 Precision 和 Recall<sup>[24]</sup>。准确率被定义为测试集中出现的推荐物品数与推荐物品总数之比。在数学上,对于所有用户, Precision 的平均值定义为

$$P(L) = \frac{1}{m} \sum_{i=1}^m \frac{d_i(L)}{L}, \quad (6)$$

其中: $d_i(L)$ 是推荐给用户的物品与测试集中与用户交互物品的交集; $L$ 是推荐给用户的物品的总数。召回率定义为用户推荐列表中出现的推荐物品数量与测试集中所有用户有过交互的物品数量的比值。在数学上,对于所有用户,Recall 的平均值定义为

$$R(L) = \frac{1}{m} \sum_{i=1}^m \frac{d_i(L)}{D(i)}, \quad (7)$$

更高的召回率意味着更高的精度。多样性是评估由个性化推荐算法推荐的物品种类的重要指标。为了量化物品相似性,使用汉明距离。所有用户的汉明距离<sup>[25]</sup>平均值定义为

$$H(L) = \frac{1}{m(m-1)} \sum_{i=1}^m \sum_{j=1}^m \left(1 - \frac{C(i,j)}{L}\right), \quad (8)$$

其中, $C(i,j) = |OL_i \cap OL_j|$ 是用户  $i$  和  $j$  的推荐列表中相同物品的数量。汉明距离的值越大,则多样性越高。

另一个多样性指标是内相似性<sup>[26]</sup>,它是指目标用户的推荐列表中出现对象之间的相似性。数学上,对于所有用户,内部相似性的平均值定义为

$$I(L) = \frac{1}{mL(L-1)} \sum_{i=1}^m \sum_{O_\alpha, O_\beta \in OL_i, \alpha \neq \beta} S\text{Cos}_{\alpha\beta}, \quad (9)$$

其中, $S\text{Cos}_{\alpha\beta}$ 是用户  $i$  的长度为  $L$  的推荐列表  $OL_i$  中对象  $\alpha$  和  $\beta$  之间的余弦相似性<sup>[27]</sup>。内相似性值越小意味着多样性越高。

新颖性是量化算法推荐新颖(即不热门)物品的能力<sup>[28]</sup>。在这里,使用推荐对象的平均人气来量化新颖性,其定义为

$$N(L) = \frac{1}{mL} \sum_{i=1}^m \sum_{O_\alpha \in OL_i} K_{O_\alpha}, \quad (10)$$

其中, $K_{O_\alpha}$ 是用户  $i$  的推荐列表  $OL_i$  中物品  $\alpha$  的度。较低的  $N$  值表示较高的新颖性和潜在的更好的用户体验。

## 3 实 验

将实验分为 2 部分,第 1 部分是构建 word2vec 模型并测试其性能。研究发现,在大量数据上用 word2vec 训

练高维词向量时,所得的向量可用于回答单词之间非常微妙的语义关系<sup>[29]</sup>,例如,城市和它所属的国家/地区,例如法国-巴黎,德国-柏林。具有这种语义关系的词向量可以让模型解释 LWV 的推荐结果且可以使其他尚未发明算法具有可能性。第 2 部分在 3 个公共数据集上对比 UCF,ICF,CosRA 这 3 种基础算法通过 LWV 更新前后的性能表现。在 UCF 中,将向目标用户推荐具有相似爱好的用户所购买的物品。类似地,在 ICF 中,将向目标用户推荐与他过去喜欢的物品相似的物品。CosRA 算法过程和 ICF 类似,不过前者对物品相似性做了更多的考虑。文中采用余弦相似性分别量化 UCF 和 ICF 中的用户和对象相似性。

### 3.1 Word2vec 构建

2003 年,Bengio 等<sup>[30]</sup>发表了一篇开创性的论文,总结出了一套用神经网络建立统计语言模型的框架,首次提出了 wordembedding<sup>[31]</sup>的概念,从而奠定了包括 word2vec 在内后续研究的基础。word2vec 分为 2 个计算模型,分别是 CBOW 模型和 Skip-gram 模型,Skip-gram 模型是根据中间词,预测前后词,CBOW 模型刚好相反,根据前后的词,预测中间词,如图 2 所示。

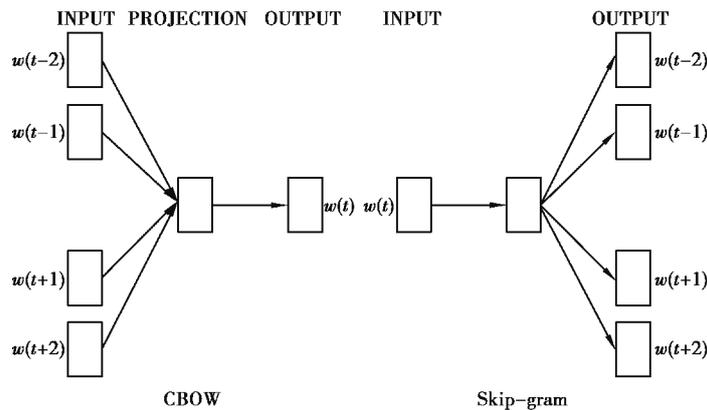


图 2 Word2vec 模型

Fig. 2 Word2vec model

2 个模型都包含 3 层:输入层、投影层和输出层。前者是在已知当前词  $W_t$  的上下文  $W_{t-2}, W_{t-1}, W_{t+1}, W_{t+2}$  的前提下预测当前词  $W_t$ 。而后者恰恰相反,是在已知当前词  $W_t$  的前提下,预测上下文  $W_{t-2}, W_{t-1}, W_{t+1}, W_{t+2}$ 。文中采用 Skip-gram 构建 word2vec 训练模型。

Word2vec 本质是一个小型神经网络,神经网络的输出是概率分布,概率代表着输入单词文本上相似性的大小<sup>[32]</sup>。

对于完整的神经网络模型,输出层需要 Softmax<sup>[33-35]</sup>分类:

$$\text{softmax}(x) = \frac{e^{x_i}}{\sum_i^N e^{x_i}}, \quad (11)$$

其中:神经网络输出层的输出是一个长度为  $N$  的向量; $x_i$  表示向量第  $i$  维的值。计算  $x$  的 softmax 值就是计算  $x$  的指数与输出向量中所有元素指数和的比值。Softmax 可以将输入映射为 0-1 之间的实数,而且和为 1。文中使用交叉熵作为神经网络的损失函数:

$$\text{Loss} = - \sum_i t_i \ln y_i, \quad (12)$$

其中: $t_i$  表示真实值; $y_i$  表示求出的 softmax 值,交叉熵损失函数可以衡量  $t_i$  与  $y_i$  的相似性。

为了将 word2vec 模型引用到 LWV 的推荐算法中需要衡量词向量嵌入的质量,文中定义了 1 个综合测试集,其中包含 5 种类型的语义问题和 9 种类型的句法问题。表 1 中显示了每个类别的 2 个示例。总体而言,共有 3 648 个语义问题和 9 989 个句法问题。每个类别中的问题分 2 个步骤创建:首先,手动创建相似单词对的列表。然后,通过连接 2 个单词对形成大量问题。例如,文中列出了 68 个美国大城市及其所属州的列表,并通过随机选择 2 个单词对形成了约 2K 个问题。

试验评估所有问题类型的总体准确性。仅当与使用 word2vec 计算出的向量最接近的词与问题中的正

确词完全相同时,才能假定问题得到正确回答;因此,同义词被视为错误。这也意味着达到 100%的准确性可能是不可能的。但是,文中认为单词向量在某些应用中的有用性应与此精度的度量值正相关。通过结合有关单词结构的信息,尤其是针对句法问题的信息,可以取得进一步的提升。

表 1 测试集中的 5 种语义和 9 种句法问题的示例

Table 1 Examples of five semantic and nine syntactic problems in the test set

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwana	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossible	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

### 3.2 测试数据集

为了评估 LWV 的有效性,使用了 3 个基准数据集:MovieLens-100K,MovieLens-1M 和 Last-FM。他们可公开访问,并且在域、大小和稀疏性方面各不相同。

MovieLens-100K&1M:一个广泛用于电影推荐的数据集,将其中部分不能用于 word2vec 计算相似性的标签及其所对应的用户交互数据删除。为了确保试验质量,将数据集进行十倍交叉划分,对每一次划分的数据集独立进行试验,最后取实验结果的平均值。

Last-FM:这是从 Last.fm 在线音乐系统收集的音乐收听数据集,其中,音轨被视为物品,同样删除了其中不能用于 word2vec 的部分数据,并将数据十倍交叉划分进行试验。

表 2 数据集描述

Table 2 Data set description

Item	MovieLens-100K	MovieLens-1M	Last-FM
# users	943	6 039	1 890
# items	836	3 158	10 226
# interactions	41 430	766 108	84 437
# tags	9 912	19 395	102 440

### 3.3 参数设置

文中使用维基百科上随机爬取的六千万篇英文文章作为训练词向量的语料库,并将词汇量限制为一百万个最常用的单词。在综合数据集上就不同词向量的维度做了准确性比较,实验结果如表 3 所示。可以看出,随着词向量维度的增加,模型评估语义,句法关系的准确率也随之提升。考虑到维度的增加会提升训练模型的时间,并且维度在 300~600 的情况下准确率提升不高,文中最终选择 300 维度下的 word2vec 模型来计算标签的相似性。

表 3 不同词向量维度下 Word2vec 准确性比较

Table 3 Comparison of Word2vec accuracy under different word vector dimensions

Accuracy / Dimensionality	50	100	300	600
Semantic Accuracy[%]	15.2	26.1	50.0	50.6
Syntactic Accuracy[%]	23.2	35.8	55.9	57.0

为了确定 LWV 对每个目标用户构造的三部图中物品节点的数量,定义了 2 个参数  $K$  和  $M$ ,其中, $K$  代表原推荐算法给训练集中每个用户推荐的正确物品之和; $M$  代表测试集中每个用户产生过交互的物品数量之和。

$$C = \frac{K}{M}, \quad (13)$$

式中: $C$  能够衡量推荐算法的性能,相同长度的推荐列表下  $C$  的值越高说明算法发掘用户真正喜欢物品的能力越好。文中在 Movielens-100K, Movielens-1M 和 Last-FM 上就不同的推荐列表长度(1L-20L,  $L=50$ )进行了比较,如图 3 所示。

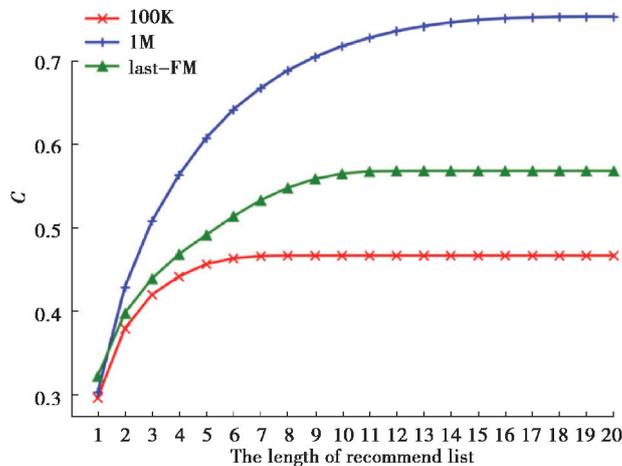


图 3 三种数据集上  $C$  值对比

Fig. 3 Comparison of  $C$  values on three data sets

图中在 3 个不同数据集上对比了 UCF, ICF 和 CosRA 算法在不同  $L$  下  $C$  的平均值,其中,红色、蓝色、绿色分别对应于 Movielens-100K, Movielens-1M, Last-FM 下的结果。

可以看出,在 3 个数据集中, $C$  最终在  $8L$ ,  $13L$ ,  $19L$  的情况下收敛,考虑到推荐列表长度对算法时间复杂度的影响,在  $C$  值收敛的情况下,在尽可能短的列表中获得最多的正确的推荐物品。文中选择  $n=8L$  作为基础算法生成的推荐列表的长度(用做 LWV 三部图中目标用户的物品节点)。此时,在 100K, 1M, Last-FM 上发掘正确推荐物品的能力分别达到原算法最大值的 100%、91% 和 96%。

### 3.4 实验结果

文中将基于 LWV 的方法应用于 3 个公共数据集,并在 3 种数据集中分别对基于用户的协同过滤(UCF),基于物品的协同过滤(ICF)和 CosRA 这 3 种基础算法来更新推荐列表( $L=50$ ),如表 4 所示。

表 4 3 种数据集上算法性能对比(粗体表示更新算法后结果较优的数据)

Table 4 Comparison of algorithm performance on the three data sets

数据集	算法	AUC	P	R	H	I	N
Movielens-100K	ICF	0.6645	0.0571	0.3328	0.6706	0.3141	180
	LWV-I	0.596 4	0.064 3	0.330 4	0.634 4	0.305 8	187
	UCF	0.803 8	0.051 6	0.296 9	0.484 8	0.361 1	216
	LWV-U	0.814 0	0.057 2	0.304 9	0.432 5	0.382 0	177
	CosRA	0.670 8	0.058 8	0.330 8	0.664 2	0.239 9	166
	LWV-C	0.680 4	0.062 9	0.339 9	0.688 0	0.214 8	139
Movielens-1M	ICF	0.725 3	0.088 1	0.363 8	0.722 6	0.363 8	1 320
	LWV-I	0.845 9	0.085 3	0.373 9	0.542 6	0.355 6	1 895
	UCF	0.864 5	0.064 6	0.260 2	0.416 3	0.442 7	1 632
	LWV-U	0.767 7	0.065 8	0.301 9	0.370 0	0.436 3	1 486
	CosRA	0.726 0	0.085 3	0.374 2	0.683 9	0.343 9	1 385
	LWV-C	0.579 9	0.089 2	0.374 5	0.693 7	0.440 7	1 446
Last-FM	ICF	0.693 7	0.031 6	0.343 1	0.904 4	0.109 6	96
	LWV-I	0.712 8	0.046 8	0.342 1	0.846 5	0.254 6	132
	UCF	0.897 7	0.029 3	0.315 0	0.616 7	0.171 1	194
	LWV-U	0.904 6	0.031 5	0.326 4	0.543 2	0.201 9	186
	CosRA	0.675 3	0.029 2	0.316 5	0.914 6	0.107 3	87
	LWV-C	0.495 5	0.032 5	0.316 8	0.912 5	0.075 1	78

6 个公共评价指标显示,被 LWV 更新过推荐列表的 3 种推荐算法在 3 种数据集上相对于原方法在准确性上具有显著优势。另一方面,经由 LWV 更新过的算法其在多样性上表现不足,说明物品之间的多样性确实可以通过文本上的相似性表示,LWV 通过将更多在文本上相似而不仅仅是用户行为上相似的物品推荐给用户,从而提高了推荐算法的准确率。可以得出结论,基于 LWV 更新过的算法能够发掘传统推荐算法所无法推荐的物品,从而推荐给用户。

### 3.5 结果分析

文中对 3 个不同的数据集分别用 LWV 更新了 3 种基础算法,每种算法随机抽取了 100 个用户并由原推荐算法得到长度为 50 推荐列表,LWV 更新过的推荐算法分别给每个用户推荐了 50 个物品的推荐列表。图 4 中红色部分代表原算法通过 LWV 更新了的那部分物品,该物品确实测试集中与用户有交互;蓝色部分代表那些在测试集中与用户有过交互但被 LWV 错误的更新掉的那部分物品。

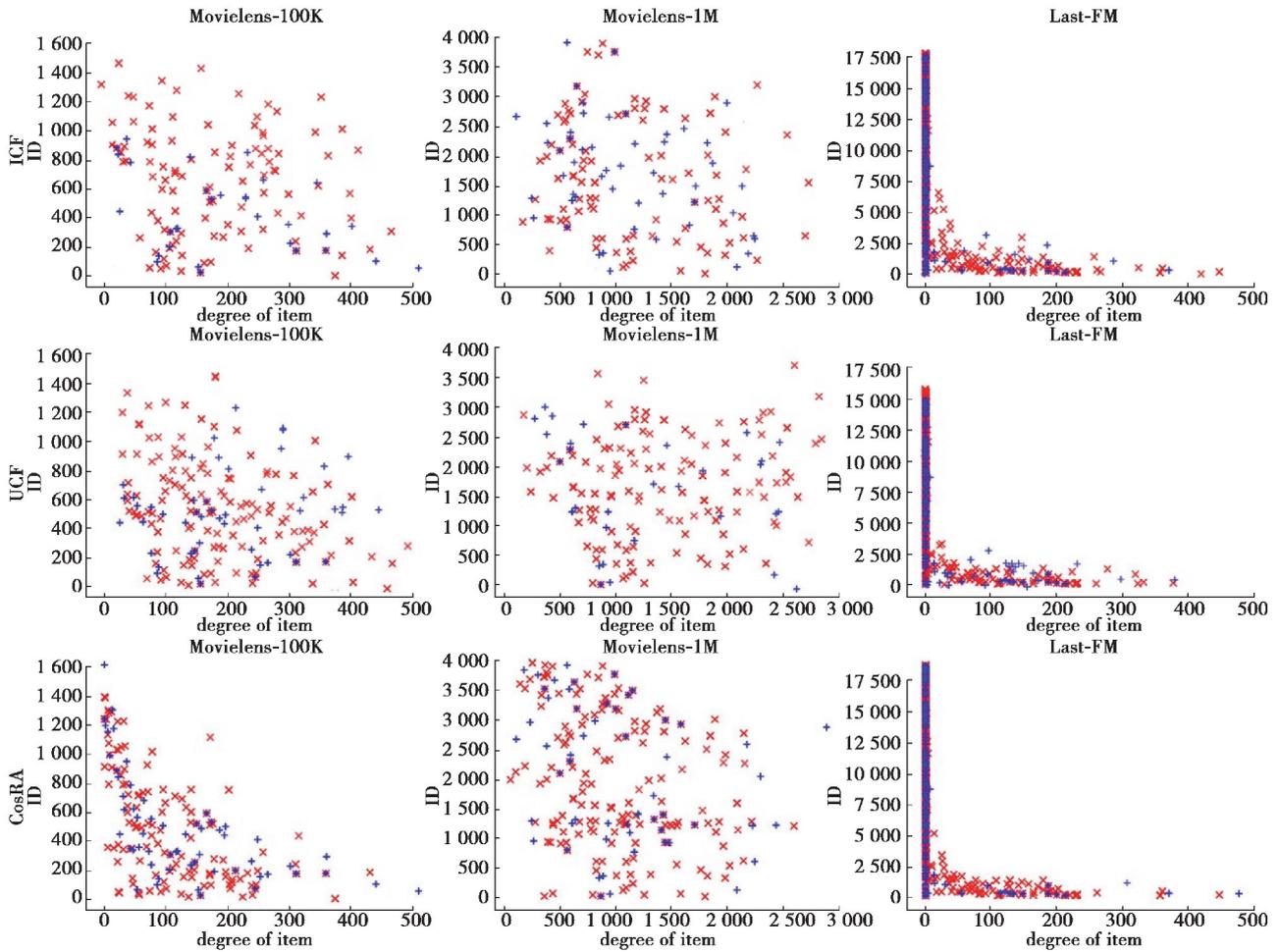


图 4 LWV 算法与原算法的散点图显示对比

Fig. 4 Comparison of scatter graph display between LWV algorithm and original algorithm

可以看出,红色部分明显多于蓝色部分。说明 LWV 确实能通过标签在向量空间中相似性来提高推荐算法的准确性,可以观察到 LWV 推荐的物品对度并没有强烈的偏好,这在实际应用中是一个显著的优势。

## 4 结 论

提出了一个新的通过增加标签辅助信息在向量空间中的相似性来增强对实际推荐影响的新模型 LWV<sup>[36]</sup>。该模型基于标签辅助信息在向量空间中的相似性生成新的用于随机游走的新边,并结合传统的协同过滤和自然语言处理模型-word2vec 构建三部图中各边的权重。在真实数据集上的大量实验表明,通过 LWV 更新过的算法相比较于原算法具备更好的准确性、多样性和新颖性。为了进一步了解算法工作情况,文中对原算法和经 LWV 更新过的算法提供了推荐列表更新前后物品在度上的分布<sup>[37]</sup>。结果表明,该方法对物品的度数没有强烈的偏见。LWV 被扩展为了一般形式,也就是说 LWV 没有任何其他的参数,这在实际应用中是一个显著的优势。

文中探索了自然语言处理在推荐中的潜力,代表利用自然语言处理的方法在三部图中生成新边结合图随机游走算法优化推荐列表的初步尝试。凸显了将机器学习应用在个性化推荐中<sup>[38]</sup>的重要作用,通过该方法合理的更新推荐列表能够使用户获得更加准确的推荐物品。但是如何在推荐系统中平衡准确性、多样性和新颖性仍然是一个悬而未决的问题。该模型在时间复杂度上仍然有着很大的劣势<sup>[38]</sup>,文中虽然将传统的包含所有用户、物品节点和交互关系的三部图拆分成多个小的三部图,在游走上大大减少了时间的消耗,但由于需要对每一个用户都生成一张唯一的三部图,因此,在面对用户特别多的实际情况中,时间复杂度问题仍不能有效解决<sup>[39]</sup>。后续研究会考虑矩阵转移的方法来代替在图中一次次迭代以产生推荐列表。

## 参考文献:

- [ 1 ] Zheng Z B, Ma H, Lyu M R, et al. QoS-aware web service recommendation by collaborative filtering[J]. IEEE Transactions on Services Computing, 2011, 4(2): 140-152.
- [ 2 ] Jinhu L, Tao Z, Zike Z, et al. Promoting cold-start items in recommender systems[J]. PLoS One, 2014, 9(12): e113457.
- [ 3 ] Davidson J, Livingston B, Sampath D, et al. The youtube video recommendation system[C]. Proceedings of the fourth ACM conference on Recommender systems-RecSys'10. New York, USA: ACM Press, 2010: 293-296.
- [ 4 ] Bogers T, van den Bosch A. Recommending scientific articles using citeulike[C]. Proceedings of the 2008 ACM conference on Recommender systems-RecSys'08. New York, USA: ACM Press, 2008: 287-290.
- [ 5 ] Lian D, Xie X, Zhang F, et al. Mining location-based social networks: A predictive perspective[J]. IEEE Data Engineering bulletin, 2015, 38(2), 35-46.
- [ 6 ] Schafer J B, Konstan J A, Riedl J. E-commerce recommendation applications[M]. Boston, MA: Springer US, 2001: 115-153.
- [ 7 ] Yao X J, Tan B H, Hu C, et al. Recommend algorithm combined user-user neighborhood approach with latent factor model[M]. Cham: Springer International Publishing, 2017: 275-280.
- [ 8 ] Cheng P, Wang S, Ma J, et al. Learning to recommend accurate and diverse items[J]. Proceedings of the 26th international conference on World Wide Web, 2017:183-192.
- [ 9 ] Ahmedi L, Rrmoku K, Sylejmani K, et al. A bimodal social network analysis to recommend points of interest to tourists[J]. Social Network Analysis and Mining, 2017, 7: 14.
- [10] Paulheim H. Knowledge graph refinement: a survey of approaches and evaluation methods[J]. Semantic Web, 2016, 8 (3): 489-508.
- [11] Wang Q, Mao Z D, Wang B, et al. Knowledge graph embedding: a survey of approaches and applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(12): 2724-2743.
- [12] Dettmers T, Minervini P, Stenetorp P, et al. Convolutional 2D knowledge graph embeddings[J]. 32nd AAAI Conference on Artificial Intelligence, 2018: 1811-1818.
- [13] Zhang Y, Blattner M, Yu Y. Heat conduction process on community networks as a recommendation model[J]. Physical review letters, 2008, 99(15): 154301.
- [14] Zhang Y C, Medo M, Ren J, et al. Recommendation model based on opinion diffusion[J]. Europhysics Letters (EPL), 2007, 80(6): 68003.
- [15] Langville A, Meyer C. Deeper inside PageRank[J]. Internet Mathematics, 2004, 1(3): 335-380.
- [16] Hu J, Liu L, Zhang C, et al. Hybrid recommendation algorithm based on latent factor model and personal rank[J]. Journal of Internet Technology, 2018,19(3): 919-926.
- [17] Xing W, Ghorbani A. Weighted PageRank algorithm[C]. Proceedings Second Annual Conference on Communication Networks and Services Research. Piscataway, NJ: IEEE, 2004: 305-314.
- [18] Ji S H, Satish N, Li S, et al. Parallelizing word2Vec in shared and distributed memory[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(9): 2090-2100.
- [19] Lofgren P, Banerjee S, Goel A. Personalized pagerank estimation and search[C]. Proceedings of the Ninth ACM International Conference on Web Search and Data Mining-WSDM 16. New York, USA: ACM Press, 2016: 163-172.
- [20] Kloumann I M, Ugander J, Kleinberg J. Block models and personalized pagerank[J]. Proceedings of the National Academy of Sciences of the United States of America, 2017, 114(1): 33-38.
- [21] Zhou T, Ren J, Medo M, et al. Bipartite network projection and personal recommendation[J]. Physical Review E, 2007, 76(4): 046115.
- [22] Liu R H, Hall L O, Bowyer K W, et al. Synthetic minority image over-sampling technique: How to improve AUC for glioblastoma patient survival prediction[C]. 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Piscataway, NJ: IEEE, 2017: 1357-1362.
- [23] Zhou T, Lü L, Zhang Y C. Predicting missing links via local information[J]. The European Physical Journal B, 2009, 71 (4): 623-630.
- [24] Herlocker J L, Konstan J A, Terveen L G, et al. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information Systems, 2004, 22(1): 5-53.

- [25] Cao Y, Long M S, Liu B, et al. Deep cauchy hashing for hamming space retrieval[C]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Piscataway, NJ: IEEE, 2018: 1229-1237.
- [26] Zhou T, Su R, Liu R, et al. Accurate and diverse recommendations via eliminating redundant correlations[J]. New Journal of Physics, 2009, 11(12): 123008.
- [27] Zhou Y, Lü L, Liu W, et al. The power of ground user in recommender systems[J]. PloS one, 2013, 8(8): e70094.
- [28] Resnick P, Varian H R. Recommender systems[J]. Communications of the ACM, 1997, 40(3):56-58.
- [29] Kottur S, Vedantam R, Moura J M F, et al. VisualWord2Vec (vis-W2V): learning visually grounded word embeddings using abstract scenes[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ: IEEE, 2016: 4985-4994.
- [30] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J/OL]. Computer ence, 2013[2020-09-29]. <https://arxiv.org/abs/1301.3781>.
- [31] Yao, Li X, Liu X P, et al. Sensing spatial distribution of urban land use by integrating points-of-interest and Google Word2Vec model[J]. International Journal of Geographical Information Science, 2017, 31(4): 825-848.
- [32] Church K W. Word2Vec[J]. Natural language engineering, 2016, 23(1):155-162.
- [33] Wang F, Cheng J, Liu W Y, et al. Additive margin softmax for face verification[J]. IEEE Signal Processing Letters, 2018, 25(7): 926-930.
- [34] Jang E, Gu S, Poole B. Categorical reparameterization with gumbel-softmax[J/OL]. arXiv e-prints, 2016[2020-09-29]. <https://arxiv.org/abs/1611.01144>
- [35] Liu W, Wen Y, Yu Z, et al. Large-margin softmax loss for convolutional neural networks[J]. ICML, 2016: 507-516.
- [36] Yang Q. A robust recommended system based on attack detection[J]. Concurrency and computation, 2019, 31(12): e4660.
- [37] Babeetha S, Muruganatham B, Ganesh Kumar S, et al. An enhanced kernel weighted collaborative recommended system to alleviate sparsity[J]. International Journal of Electrical and Computer Engineering (Malacca, Malacca), 2020, 10: 447-454.
- [38] Mi H Y, Muruganujan A, Huang X S, et al. Protocol update for large-scale genome and gene function analysis with the PANTHER classification system (v.14.0)[J]. Nature Protocols, 2019, 14(3): 703-721.
- [39] Putriany V, Jauhari J, Izwan Heroza R. Item clustering as an input for skin care product recommended system using content based filtering[J]. Journal of Physics: Conference Series, 2019, 1196: 012004.

(编辑 陈移峰)