

基于主题分组与随机游走的 App 推荐算法*

赵海燕¹, 张健¹, 曹健²

(1. 上海理工大学 光电信息与计算机工程学院, 上海 200093; 2. 上海交通大学 计算机科学与技术系, 上海 200030)

摘要: 近年来, App 的数量呈爆炸式地增长, 在庞大数量的手机 App 中找到合适的 App 给用户带来了困难。传统的推荐系统方法运用到 App 推荐上时有很多的局限性, 如难以解决冷启动和用户对不同类别的应用具有不同的选择偏好等问题。提出了一种基于主题分组和随机游走的个性化推荐算法 TGRW。TGRW 针对用户对每类 App 需要的数量、偏好的程度各不一样的特点, 根据用户的 App 使用记录, 构造了 user-App 组-App 的三元图模型, 通过对不同的用户在不同的推荐组上设置不同的权重, 再利用随机游走算法计算出用户对每个 App 的偏好概率, 从而形成推荐列表。在真实用户数据集上的实验表明, 该方法比其他方法在推荐效果上得到了明显提升。

关键词: 手机应用; 主题模型; 随机游走

中图分类号: TP301.6

文献标志码: A

文章编号: 1001-3695(2018)08-2277-04

doi:10.3969/j.issn.1001-3695.2018.08.009

Personalized App recommendation algorithm based on topic grouping and random walk

Zhao Haiyan¹, Zhang Jian¹, Cao Jian²

(1. School of Optoelectronic Information & Computer Engineering, University of Shanghai for Science & Technology, Shanghai 200093, China; 2. Dept. of Computer Science & Technology, Shanghai Jiao Tong University, Shanghai 200030, China)

Abstract: In recent years, the explosive growth of the number of App brings difficulties for user to find a suitable App. There are many limitations of the traditional recommendation approaches when they are applied to App, such as the cold start problem and different choice bias on different types of applications. This paper proposed a personalized recommendation algorithm TGRW, which was based on topic grouping and random walk. Because users have different biases and underlying choice factors on the App of different categories, TGRW firstly divided App into categories based on their description information. Then it constructed a triple tuple graphic model consisting of user, App group and App. By applying random walk to calculate the preferential probability of user to each App, it obtained the recommendation list. Extensive experiments on the real data set shows TGRW gains significant improvements on recommendation performances than other methods.

Key words: mobile phone application; topic model; random walk

0 引言

随着智能手机的普遍使用,大量的手机应用(App)被开发出来并得到了推广。据统计,每年都会有上百万的 App 被开发出来。这给用户搜索、选择合适的应用带来很大的困难,为了便于用户选择 App,许多 App 市场中将 App 按照类别进行了组织,在同一类别中又按照 App 的流行度进行排列^[1]。这种方法虽然对用户选择 App 提供了一定的便捷性,但是,它需要用户已经明确自己需要某种类别的 App,并在这一类别中也能进行自主选择。显然,流行的 App 也许能满足大众化的需求,但不一定能满足用户个性化的要求。因而需要给用户 App 的个性化推荐服务,来提高用户的体验。

传统的推荐算法如基于用户的协同过滤、基于物品的协同过滤、矩阵分解等可以运用到 App 推荐上。但是,这些方法在应用时将遇到以下问题:

a) App 数据非常稀疏,相较于巨大的 App 数量,每个用户安装的 App 非常有限,因此 App 的使用数据是非常稀疏的,在推荐时经常遇到冷启动问题。

b) 从功能出发,App 分为了不同的类别,显然在不同类别上,用户需要的 App 数量不同,同时选择的依据也不一样。传统的推荐方法难以考虑不同类别的差异。

因而,按照类别来刻画用户的偏好,一方面可以克服数据稀疏的问题,另一方面,也将使推荐具有针对性。对 App 分类可以采用不同的方法。LDA 主题模型,是隐语义分析的一种重要方法。该模型主要用于挖掘文本中隐含的主题分布,在基于文本内容的推荐中得到广泛应用,由于 App 通常有一些介绍性的文字描述,所以利用该模型可以实现对 App 分类。

为了进一步反映在同一类别内部不同 App 的偏好,本文构造了用户-App 组-App 三部图,并通过随机游走来获得推荐列表,该方法(topic grouping and random walk based recommendation algorithm, TGRW)结合了主题分组和随机游走,使得 App 推荐更加具有针对性。

本文使用了真实的用户用 App 记录去验证模型的有效性。TGRW 与按流行度推荐的算法、基于用户的协同过滤算法、基于主题相似性的协同过滤方法、基于二分图的随机游走算法、概率矩阵分解算法进行了比较,实验结果表明本文提出的方法在推荐的准确度上明显高于对比算法。

1 相关工作

随着 App 爆炸似地增长,人们如何有效选择需要的 App 成了突出问题。在目前的 App 市场中就提出了一些解决办法,除了通过广告对某些 App 进行商业推广外,经常按照下载

收稿日期: 2017-03-10; 修回日期: 2017-04-22 基金项目: 国家自然科学基金资助项目(61272438, 61472253); 上海市科委基于大数据的个性化服务资助项目(14511107702)

作者简介: 赵海燕(1975-),女,上海人,副教授,博士,主要研究方向为数据挖掘、服务计算(1718151079@qq.com); 张健(1991-),男,硕士,主要研究方向为数据挖掘; 曹健(1972-),男,教授,博士(后),主要研究方向为服务计算、网格计算、大数据分析等。

量、推荐星级和评论数对 App 进行排列。然而, App 的排列对所有人都是同样的, 无法满足个性化的要求。

为了提供个性化的 App 推荐方法, 学术界也进行了相关研究。例如, Applause 系统是一个基于位置上下文的 App 推荐系统^[2], 其位置信息是通过移动设备自动感知或者人为输入得到, 但该系统不考虑用户的个人喜好, 只是根据当前位置附近 App 的使用频繁程度向用户进行推荐。为了改进这一点, 另一个基于位置的 App 推荐算法被提了出来^[3], 它收集用户平时在不同地方的活动信息, 再结合当前的位置推荐能够更好地匹配用户需求的 App。在文献[4, 5]中, 进一步结合了额外的信息, 包括用户对 App 的反馈、评论信息, App 的生命周期及其他的信息来提高推荐效果。文献[6]中考虑了用户对不同类别 App 的下载历史信息, 通过对这些信息的分析, 以推荐概率最高的 App。

以上算法主要基于 App 的使用情况来获取用户的偏好, 不能很好地解决物品冷启动问题。TGRW 既考虑用户的历史使用信息挖掘用户兴趣偏好, 又利用 App 描述信息解决物品冷启动问题, 更好地提高了推荐效果。

LDA 模型^[7]是一个典型的文档主题模型, 一个文档包含了若干主题, 每个单词在一个主题中出现的概率也符合一定的分布, 通过采样可以确定某一文档中的主题分布情况。许多基于 LDA 的模型已经被提出, 并且被广泛应用于包括推荐系统中^[8-10]。同样, App 的描述信息及评论信息可以用 LDA 方法进行分析^[11]。利用 LDA 模型进行 App 主题特征向量的提取后, 可以计算 App 之间的相似度, 再根据用户的 App 使用记录进行推荐。

随机游走算法已经广泛用于搜索领域^[12], 由于用户—物品关系能够构成二元图, 随机游走算法被引入到推荐系统中。由于 App 潜在地属于不同的分类组, 本文中构造了 user、App 组、App 三部图, 再利用随机游走算法挖掘潜在的用户对 App 的兴趣偏好。

2 基于主题分组与随机游走的个性化推荐算法

2.1 随机游走算法

基于二分图的随机游走推荐算法 (PersonalRank) 的思想与 PageRank 类似, 即假设要给用户进行个性化推荐时, 可以从用户对应的节点开始在用户—物品二分图上进行随机游走。游走到任何一个节点时, 首先按照一定概率决定是继续游走、还是停止这次游走并从初始节点开始重新游走。如果决定继续游走, 那么就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作为游走下次经过的节点。这样经过很多次随机游走之后, 每个物品节点被访问到的概率会收敛到一个数, 最终的推荐列表中物品的权重就是物品节点的访问概率。随机游走算法过程可表示为

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

其中: α 是继续游走的概率; $\text{in}(v)$ 是指向节点 v 的节点集合; $\text{out}(v')$ 是节点 v' 指向的节点集合。

图 1 是图模型的简单例子, users 集 $U = \{A, B, C\}$, items 集 $I = \{a, b, c, d\}$ 。

2.2 概率主题模型

LDA 主题模型的思想是: 利用物品本身的描述信息挖掘文档—词语之间所蕴涵的潜在语义关系 (隐含主题), 然后利用主题分布刻画物品, 进而计算物品之间的关系。主题模型在文本挖掘领域发挥着重要作用, 并逐步从传统信息检索扩展到关系网络数据挖掘、社交媒体计算、文本情感分析等领域。该

算法模型利用贝叶斯理论, 先假设物品文档词的先验为狄利克雷分布, 样本数据是多项分布, 后验概率也为狄利克雷分布。本文通过统计得到词的分布, 然后利用吉布斯采样的方法, 去计算文档中主题分布和主题中词的分布。

α, β 是两个超参数, 是文档主题分布 θ_d 的先验分布的参数, β 是 topic-word 概率分布的先验分布参数, 主题模型的具体生成过程如图 2 所示。

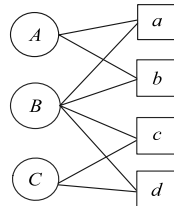


图1 PersonalRank图模型

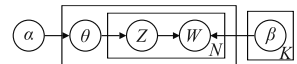


图2 LDA图模型

a) 按照 Dirichlet 分布, 生成文档 d 的主题分布 $\theta_d \sim \text{Dir}_N(\alpha)$ 。

b) 从主题分布 θ_d 中, 选取一个主题 $Z_{d,n} \sim \text{Mult}(\theta_d)$, $Z_{d,n} \in \{1, \dots, K\}$ 。

c) 根据选取主题选取一个单词 $W_{d,n} \sim \text{Mult}(\beta_{Z_{d,n}})$, $W_{d,n} \in \{1, \dots, N\}$ 。

其中: α 是 N 维向量; N 为语料库中单词数目; $\text{Dir}_N(\alpha)$ 表示以向量 α 为参数的 N 维 Dirichlet 分布; d 为文档数目; $Z_{d,n}$ 为第 d 个文档中第 n 个词的主题; $W_{d,n}$ 为第 d 个文档中第 n 个词; β_k 为第 K 个主题词的分布。

LDA 通过抽象的主题表示文档的隐含语义, 通过后验概率方法, 在给定文档和主题个数的前提下, 可以计算文档下的主题概率分布。由概率模型图可得 LDA 的联合概率为

$$p(\theta, Z, W | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^N p(z_n | \theta) p(w_n | z_n, \beta) \quad (2)$$

有了联合分布 p , 就可以考虑使用 Gibbs Sampling 算法对分布进行采样, 最终会推导得到对 LDA 模型的 Gibbs sampling 公式:

$$p(z_i = k | Z_{-i}, W) \propto \frac{n_{m, \cdot, i}^{(k)} + \alpha_k}{\sum_{k=1}^K (n_{m, \cdot, i}^{(k)} + \alpha_k)} \cdot \frac{n_{k, \cdot, i}^{(t)} + \beta_t}{\sum_{t=1}^V (n_{k, \cdot, i}^{(t)} + \beta_t)} \quad (3)$$

LDA 模型的具体详细算法描述及 Gibbs Sampling 的详细推导过程可见文献[13, 14]。

2.3 基于主题分组和随机游走个性化推荐算法—TGRW

在现实生活中使用的 App 总是分为不同的类, 例如音乐、聊天、视频……还有其他更具体的分类。在现实中, 每个人在不同类别中会选择一些 App 进行下载使用, 而且由于每类的特点不同, 其下载的 App 数量也会差别明显。例如, 游戏类 App 可能下载数量会比较多, 而聊天类由于重叠性比较大, 一般不会下载许多 App。另一方面, 不同类别中 App 的选择理由是不一样的, 选择游戏和选择聊天类 App 的依据就完全不同。

这就启发笔者推荐 App 时应该依据 App 的类别进行。然而, 目前的分类是由 App 的发布者或者 App Store 的管理者人为进行的。每个管理者分类的标准可能不一样, 这样对 App 的分类就不能反映其真实功能和用途。主题模型是挖掘文本的隐含主题的经典模型, 被广泛用于微博、Web 等文本的分析, 都收到良好的效果。可以借鉴其思想根据隐含主题对 App 进行分类, 这样就可以根据用户在不同 App 组的使用情况, 再推荐各组中的 App。

PageRank 在进行搜索网页排序上体现出很好的效果, 根据此原理的 PersonalRank 也在基于图进行推荐中得到广泛的应用, 并且效果良好。因而提出了 TGRW-基于主题分组的随机游走推荐算法。该算法的中心思想是: 根据用户使用 App 的历史数据集, 结合使用的 LDA 主题模型对物品分组构造三类节点, 即用户节点 U 、App 分组节点 G 、App 节点 A , 根据用户使用 App 组中 App 个数的情况, 设置用户到 App 分组的权

值;接着根据 App 属于某个分组的程度设置在各个 App 分组中的权重设置 App 组节点到节点 App 边的权值;最后利用 PersonalRank 算法在改造好的图上随机游走,得到用户到各个 App 的偏好权值,根据权值的排名形成对用户的推荐列表。模型图如图 3 所示。首先根据数据集记录分析用户对各个类别上的 App 的点击次数,以此来判定用户对各个类的需求情况。这样,就可以量化需求、兴趣情况来设置随机游走边的权重。

下面通过 K-similar 对 App 分组,K-similar 主要思想来自于 K-means,K-means 主要依据是物品特征向量的欧氏距离:

$$d_{12} = \sqrt{(a-b)(a-b)^T} \quad (4)$$

其中: a 、 b 表示两个特征向量; d_{12} 的大小作为两点聚类的依据,同理,K-similar 也可以作为物品特征向量的聚类依据,计算公式如下:

$$\cos(\theta) = \frac{a \cdot b}{\|a\| \times \|b\|} \quad (5)$$

\cos 值大小作为两点聚类的依据。这里的 a 、 b 是指通过 LDA 主题模型来获取得到 App 的特征向量。根据 LDA 模型中主题分布特征向量进行相似计算把 App 聚类即挖掘出潜在分类。由于 App 属于不同的类,所以可根据用户具体点击 App 的次数概率以及属于各类的情况得到对此类 App 的需求程度。

具体计算方式如下:

$$PR(v) = \begin{cases} \alpha \sum_{v' \in \text{in}(v)} \eta_{v,v'} \frac{PR(v')}{|\text{out}(v')|} & v \neq v_u \\ (1-\alpha) + \alpha \sum_{v' \in \text{in}(v)} \eta_{v,v'} \frac{PR(v')}{|\text{out}(v')|} & v = v_u \end{cases} \quad (6)$$

$$\text{s. t. } \begin{cases} \eta_{v,v'} = \sqrt{\frac{N}{t_1+1}} & U \leftrightarrow G \\ \eta_{v,v'} = \cos_sim(o, v) \times \text{popular_weight}(v) & G \leftrightarrow A \\ \eta_{v,v'} = \sqrt{\frac{m}{N}} & U \leftrightarrow A \end{cases}$$

其中: $\eta_{v,v'}$ 表示 TGRW 图模型中三类节点之间的权重系数; α 是决定是否继续游走的概率; N 是用户点击 App 总次数,用户点击 App 次数在各个类中的分布 (t_1, t_2, \dots, t_k); $\cos_sim(o, v)$ 的值是 App 与 App 类中心值的相似关系; $\text{popular_value}(v)$ 是 App v 流行度的度量值,主要依据是 App 的用户数; m 是用户对某个 App 的点击次数。

$\eta_{v,v'}$ 作为节点间的权重参数,用数值量化了用户对 App 分组、App 的兴趣关系及 App 分组与 App 之间潜在联系,这样就可以通过实验设置不同的迭代次数,在访问各个节点稳定时,根据其被访问到的概率大小来形成对各个用户的推荐列表。

算法的伪代码如下:

```
Algorithm TGRW: topic group personalRank-recommendation algorithm
Input: all node construct graph;  $G$ ,
Random choice walk node probability;  $\alpha$ 
Iteration number; iterNum
Target user: TUser
The number of recommendation;  $N$ 
Output: Recommendation List; rank
# Initialization all node the weight of visit
rank ← {  $x:0$  for  $x$  in  $G$ .keys() }
# the target user's initialization
rank[TUser] ← 1
for  $k$  in range(iterNum):
temp ← {  $x:0$  for  $x$  in  $G$ .keys() }
for  $i$ , values in  $G$ .items():
#  $w_{ij}$  is the Relation coefficient between node  $i$  and node  $j$ 
for  $j$ ,  $w_{ij}$  in values.items():
temp[j] +=  $\alpha \times \text{rank}[i] / (1.0 \times \text{len}(\text{values})) \times w_{ij}$ 
if  $j = \text{TUser}$ :
temp[j] +=  $(1-\alpha)$ 
rank ← temp
rank ← { the item(App) node for node in rank.keys() not the group
node and not other user node }
return dict(sorted(rank.items(), key = lambda x: x[1], reverse =
true)[0:N])
```

其中: α 是决定是否继续游走的概率参数; iterNum 是迭代次数,它们的最优取值需要通过实验获得; rank 是给用户的推荐列表,上述模型的复杂度是 $O(n^3)$ 。迭代过程是从给定用户节点开始通过不断游走,迭代修改到达其他各个节点的概率值,最后收敛于稳定值,即为推荐列表中各节点的权值。

3 实验及分析

3.1 数据集

实验所采用的数据集来自真实的用户 App 的使用记录,时间为 2014/10/1 ~ 2014/11/29, 每条记录表示用户的一次使用行为,每条记录包含的信息如表 1 所示。

表 1 数据集信息详情表

数据字段	示例	解释
UserID	725051837742690	用户标志
APPNAME	随手记	App 名称
RECTIME	2014/10/1 0:05:23	程序启动时间
TRAFFIC	3786	使用消耗的流量,单位:Byte
ONLINETIME	260	在线时长,单位:s
CITY_NAME	上海	所在城市名称
NETMODENAME	联通 GSM	所用网络的制式
GCELL	7-3289-57003	位置信息

数据集中包含 127 512 条用户使用记录。经过统计,可以得到用户使用 APP 的数量。将使用了 10 个以上 App 的用户筛选出来,这里定义为比较活跃的用户。

经过处理,筛选出用户数及 App 数如表 2 所示。

表 2 实验数据集(活跃用户)

	用户数	APP 数
数量	1 000	1 730

现在分析一下在用户中使用频率最高的 10 个 App,如图 4 所示。

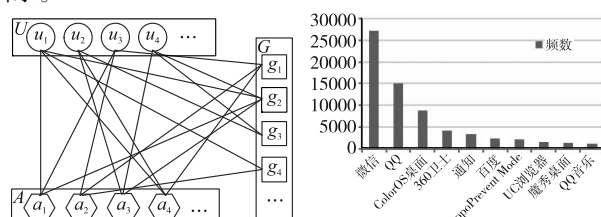


图 3 TGRW 模型图

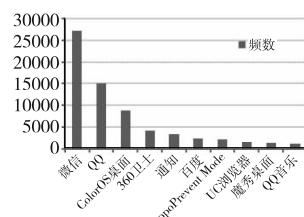


图 4 最常用的 App 统计

如图 4 所示点击率最高的都是常用的一些 App,几乎每个用户都会安装,因而再对这些 App 推荐是没有必要的,所以可通过统计每个 App 被使用的用户数,去除用户数较高的 App,这样会使得推荐更有意义。

3.2 评价指标与对比算法

实验效果的评价指标选择准确率 precision 和召回率 recall。计算公式如下所示:

$$\text{precision} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (7)$$

$$\text{recall} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (8)$$

其中: U 表示测试集中的用户集合; $R(u)$ 是根据用户在训练集上的行为为用户计算出的推荐列表; $T(u)$ 是用户在测试集上的行为列表。

实验中与 TGRW 作对比的算法有:基于流行度推荐的算法、基于用户的协同过滤算法、基于主题相似性的协同过滤方法、基于二分图的随机游走算法。

3.3 实验结果与分析

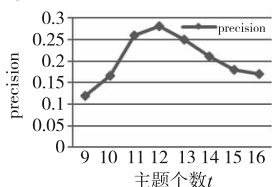
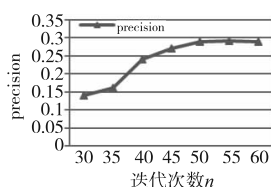
3.3.1 分组数 t 对 precision 的影响

App 分组的个数是根据 LDA 主题模型的主题数而定的,

可以通过实验来确定主题数。图 5 中 t 表示主题个数,也就是 App 分组数对 precision 的影响,由图 5 可以看到随着分组数 t 的增加,precision 的值逐渐增加之后又减小,本文的算法在分组数 $t = 12$ 的情况下,precision 的值最大。因此,本文确定 t 的值为 12。

3.3.2 迭代次数 n 对 precision 的影响

图 6 表示当迭代次数 n 达到一定值时,每个 App 节点被访问到的概率趋向稳定于某一个值,在这种情况下,可认为训练模型收敛。本文根据稳定的概率作为推荐的依据,由图 6 可以看出当 n 的值在 50 附近的时候,precision 基本达到稳定的状态,当迭代次数较多时会耗费系统的性能,因此确定 n 的值为 50。

图5 分组数 t 对precision的影响图6 迭代次数 n 对precision的影响

3.3.3 对比实验分析

在确定了算法 TGRW 的各个参数的取值后,现在将该算法与五个算法进行比较分析。其中 Base-popular 为依据流行度进行的推荐方法;LDA-CF 为依据主题相似性进行的推荐方法;Base-userCF 为基本的基于用户的协同过滤方法;RWR 为 PersonalRank 算法;PMF 为概率矩阵分解算法。

将原始数据集随机分开:80% 为训练集,20% 为测试集,这里设置先验参数 $\alpha = 0.05, \beta = 0.05$,主题个数设置为 12,迭代次数 n 设置为 50。如图 7.8 所示分别比较在不同推荐数量下在指标 precision、recall 上的效果,可以看出在 App 数据集上 TGRW 在两指标的比较上要优于其他五种算法。

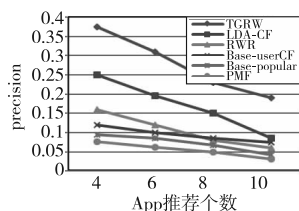


图7 六种算法在precision上的对比图

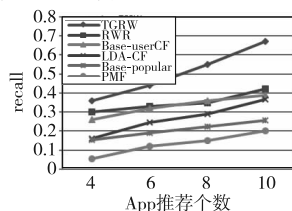


图8 六种算法在recall上的对比图

下面将数据集按不同的比例分开时,分析对实验结果的影响,如图 9、10 所示。

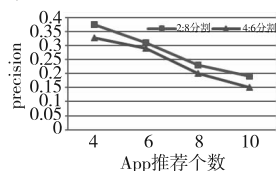


图9 不同分割比例对precision的影响

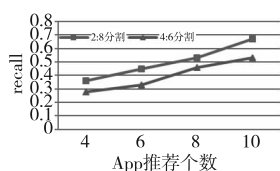


图10 不同分割比例对recall的影响

由图 9、10 可得:测试集、训练集 2:8 比例分割比 6:4 分割效果较好,原因是由于训练集数据量减少计算得到的主题模型不能反映准确的用户主题偏好,也可以说是算法的欠拟合。同样,由于测试集数量增多,在计算召回率时 $T(u)$ 变大,推荐准确性没有提高,导致召回率也降低。同样也不能使训练集过大,一是计算复杂度会增加,二是可能会使算法过拟合,所以数据集的分割比例是根据实验效果调整的。

4 结束语

该算法的创新在于融合了其他算法的优势,很好地使用了用户的隐式反馈信息和物品本身的信息。因为实际应用中有时候很难获取用户对物品的直接行为信息,推测其偏好会误差

很大。本文提出的 TGRW 算法中,没有用户对物品的直接偏好信息,算法通过物品客观描述信息计算分析得到的用户偏好主题类,类比于标签,得到用户偏好的总体方向,然后通过主题类的连接间接得到用户与物品的关系,用户与各个物品的潜在的稳定相关性就可以通过随机游走的算法获得。最后根据节点间收敛的访问概率得到推荐列表。

算法可应用于出现用户信息不足、物品冷启动等问题的场景,TGRW 算法能充分利用物品描述信息和隐式反馈信息挖掘用户兴趣偏好,提高了推荐准确性。接下来的工作准备尝试通过统计 APP 的使用时间的长短、位置信息等来更实时、准确地衡量用户的偏好,以提高个性化推荐的效果。

参考文献:

- [1] Davidsson C. Mobile application recommender system[M]. [S. l.]: Uppsala University Publications, 2010.
- [2] Davidsson C, Moritz S. Utilizing implicit feedback and context to recommend mobile applications from first use[C]//Proc of Workshop on Context-awareness in Retrieval and Recommendation. New York: ACM Press, 2011: 19-22.
- [3] Liu Yanliang, Hu Yupeng, Yu Ruiyun, et al. Application recommendation at places for mobile users[C]//Proc of the 17th International Symposium on World of Wireless, Mobile and Multimedia Networks. Washington DC: IEEE Computer Society, 2016: 1-9.
- [4] Girardello A, Michahelles F. AppAware: which mobile applications are hot? [C]//Proc of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services. New York: ACM Press, 2010: 431-434.
- [5] Karatzoglou A, Baltrunas L, Church K, et al. Climbing the App wall: enabling mobile App discovery through context-aware recommendations[C]//Proc of the 21st ACM International Conference on Information and Knowledge Management. [S. l.]: Uppsala University Publications. New York: ACM Press, 2012: 2527-2530.
- [6] Cui Yidong, Liang Kang. A probabilistic top- n algorithm for mobile applications recommendation[C]//Proc of the 5th IEEE International Conference on Broadband Network & Multimedia Technology. Piscataway, NJ: IEEE Press, 2013: 129-133.
- [7] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3(1): 993-1022.
- [8] Jin Xin, Zhou Yanzan, Mobasher B. A maximum entropy Web recommendation system: combining collaborative and content features [C]//Proc of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. New York: ACM Press, 2005: 612-617.
- [9] Porteous I, Bart E, Welling M. Multi-HDP: a non parametric Bayesian model for tensor factorization[C] //Proc of the 23rd AAAI Conference on Artificial Intelligence. 2008: 1487-1490.
- [10] Singh A P, Gordon G J. Relational learning via collective matrix factorization[C]//Proc of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2008: 650-658.
- [11] Pan Tianhao, Zhang Weifeng, Wang Ziyuan, et al. Recommendations based on LDA topic model in Android applications[C]//Proc of IEEE International Conference on Software Quality, Reliability and Security Companion. Piscataway, NJ: IEEE Press, 2016: 151-158.
- [12] Bagci H, Karagoz P. Random walk based context-aware activity recommendation for location based social networks[C]//Proc of IEEE International Conference on Data Science and Advanced Analytics. Piscataway, NJ: IEEE Press, 2015: 1-9.
- [13] Yan Xiaohui, Guo Jiafeng, Lan Yanyan, et al. A biterm topic model for short texts [C]//Proc of the 22nd International Conference on World Wide Web. New York: ACM Press, 2013: 1445-1456.
- [14] Bär A, Berger A, Egger S, et al. A lightweight mobile TV recommender[M]//Changing Television Environments. Berlin: Springer, 2008: 143-147.