一种基于随机游走的多维数据推荐算法

李 芳[1] 李永进[2]
(湖北理工学院计算机学院 黄石 435000)[1] (国防科技大学计算机学院 长沙 410073)[2]

摘 要 在推荐系统中,推荐算法不但要具备很高的准确性,还需要满足灵活性。为了使推荐算法满足准确性,同时尽量提高算法的灵活性,提出了一种基于随机游走的多维推荐算法。首先,应用用户的上下文信息建立一个多维的推荐系统模型;其次,将用户的查询分解为多个子查询,并建立相应的二部图;最后,应用随机游走模型将候选项排序,并将top-k个选项作为结果返回。实验结果表明,提出的推荐算法能灵活满足用户多样化的推荐查询,并具有很好的准确性,明显优于相关的推荐算法。

关键词 推荐系统,多维数据,随机游走,二部图
中图法分类号 TP319 **文献标识码** A

Multidimensional Data Recommender Algorithm Based on Random Walk

LI Fang[1] LI Yong-jin[2]
(School of Computer, Hubei Institute of Technology, Huangshi 435000, China)[1]
(School of Computer Science, National University of Defense Technology, Changsha 410073, China)[2]

Abstract In recommender system, both of accuracy and flexibility are important for recommender algorithms. In order to provide a high flexibility while keeping high accuracy, this paper proposed a random walk based multidimensional recommender algorithm. First, this paper built a multidimensional recommender system model using users' context, second, divided the user query into several sub-queries, and built a bipartite graph, finally, ranked candidate items according to the random walk model, and returned top-k results. Experiments show that the proposed algorithm can satisfy flexible recommender requests while keeping high prediction accuracy, and is more effective than related algorithms.

Keywords Recommender system, Multidimensional data, Random walk, Bipartite graph

推荐系统是根据用户的兴趣特点和购买行为,向用户推荐其感兴趣的信息和商品。随着电子商务规模的不断扩大,商品个数和种类快速增长,顾客需要花费大量的时间才能找到自己想买的商品。这种浏览大量无关的信息和产品过程无疑会不断流失淹没在信息过载问题中的消费者。为了解决这些问题,推荐系统应运而生。推荐系统是建立在海量数据挖掘基础上的一种高级商务智能平台,以帮助电子商务网站为其顾客购物提供完全个性化的决策支持和信息服务。

一个典型的推荐系统包括用户集合U、商品集合I以及用户对商品的评价值矩阵R。由于系统中含有大量的商品,用户只对部分商品进行了评价,因此这个矩阵是稀疏的。推荐算法的目标是对R中的未知项进行预测。给定U、I以及R,推荐算法的目标是根据已知的评价值矩阵R建立一个评估函数$f:U\times I\rightarrow R'$,并利用评价函数$f$对未知的$U\times I$进行评价[1]。在实际应用中,推荐算法常常将那些评价值高的k个商品推荐给某用户。在推荐系统的性能评价中,常用的评价标准有准确性(accuracy)、多样性(diversity)、偶然性(serendipity)和可解释性(explainability)[2]。

然而,在现实的推荐系统应用中,除了用户和商品这两个维度以外,还存在着大量的上下文信息(如购买商品的时间和地点)、元数据信息(如年龄、住址和性别等)。这些上下文信息可以应用于推荐系统,使得推荐算法的结果更准确。此外,应用这些上下文信息,除了向用户推荐商品之外,还可以对用户进行用户推荐,对某一小组的用户推荐商品等。对于推荐系统来说,这种处理用户-商品-上下文信息的多维信息处理能力以及应用多维信息进行不同种类的推荐服务是灵活性[3]。本文应用用户的上下文信息,研究了多维数据下的灵活的推荐算法。

1 相关工作

现有的推荐方法主要可以分为两类:基于内容的推荐方法和协同过滤推荐方法。基于内容的推荐方法对用户以前访问过的商品进行分析,并将与其相似的未知商品推荐给用户,这种方法主要是对商品的资料(如大小、类别、生产商等)进行分析,然后将未知的商品与之比较以发现相似的商品[4]。协同过滤推荐方法的基本思想来源于口碑营销,其应用已有的用户对商品的偏好来预测用户对某商品的喜好程度[5]。协同过滤推荐方法是一种有效的推荐方法,且已经被众多的电子

到稿日期:2013-02-22 返修日期:2013-06-02 本文受国家自然科学基金项目(61170045)资助。
李 芳(1971—),女,硕士,副教授,主要研究方向为计算机网络等,E-mail:lifangxz@163.com;李永进(1964—),男,硕士,副研究员,主要研究方向为高性能计算机体系结构、高性能多核处理器体系结构、高性能存储系统。

商务网站如亚马逊所采用。相对于基于内容的推荐方法，协同过滤推荐方法的优势是准确性高，不需要过高的商品专业化知识，并且还可以充分利用用户之间的关系网络[6]。然而当系统中没有足够的相似项时(冷启动问题)，协同过滤方法的准确性低，此时需要基于内容的推荐方法作为补充。

协同过滤推荐方法主要可以分为 Memory-based 和 Model-based 两种推荐方法。Memory-based 推荐方法包括基于用户的(user-based)协同过滤推荐方法和基于项的(item-based)协同过滤推荐方法。基于用户的协同过滤方法[7]的核心是寻找目标用户的相似用户，通过分析用户-项评分矩阵来发现相似的用户，并将这些相似用户对项的评分值进行聚合，然后得到用户对商品的评价值。基于项的协同过滤[8]与基于用户的协同过滤方法相似，区别在于其目标是寻找目标项的相似项。Memory-based 协同过滤方法通过用户对项的历史评价发现相似的用户或项，并用它们对未知的评价值进行预测。Model-based 协同过滤方法[9]则根据用户对项的历史评价建立一个模型，并根据该模型对用户-项评价矩阵中的未知评价值进行预测。常用的 Model-based 协同过滤方法有矩阵分解[10,11]、聚类方法[12]、回归方法[13]和潜在语义模型[14]等。

Adomavicius 等人在文献[3]中首次将灵活性(flexibility)作为性能的评价标准引入到推荐系统中，他们提出了一种类似 SQL 的推荐查询语言——REQUEST，REQUEST 能很好地表现出用户不同的推荐需求。Koutrika 等人[15]提出了一种推荐查询框架——FlexRecs。FlexRecs 将操作符进行混合，以表达用户不同的推荐需求。REQUEST 和 FlexRecs 试图通过定义声明式语言来表达用户的推荐需求，然而并没有说明如何处理高维数据以高效地完成推荐查询。为了实现高效的推荐查询，Adomavicius 等人[16]提出了一种基于规约的预先过滤方法。该方法对相关数据集进行预先过滤，算法简单且容易实现，但是由于真实数据集往往是稀疏的，该算法的效果往往不尽如人意。为了解决数据稀疏性问题，Lee 等人[17]提出了一种基于析取的方法。然而，基于析取的方法在解决了稀疏性的同时却假设数据的各个维度是相互独立的。此外，Kahng 等人[18]应用 Learning-to-Rank 方法提出了一种集成多维数据上下文的方法。本文通过对多维数据进行建模，建立一个支持不同类型推荐请求的模型来实现推荐的灵活性。

2 多维数据推荐方法

2.1 问题定义

传统的推荐系统往往只考虑包含 $User \times Item$ 的二位数据空间，其它的上下文信息往往应用在预先过滤阶段[16]。为了在多维数据上应用推荐算法，以便提供更大的推荐灵活性，本文对多维推荐系统的定义如下：系统包括 n 维不相交的集合 $D_1, D_2, \cdots, D_n$，其中 $D_i = \{e_{ij} \mid 1 \leqslant j \leqslant k_i\}$ 是一个包含 k_i 个实体的领域集合(如用户或项)，并且 $U = D_1 \cup D_2 \cup \cdots \cup D_n$；给定一个用户查询集合 $Q \subset U$，用函数 $f(Q, e_{ij})$ 评价实体 e_{ij} 与用户查询 Q 之间的相关性；我们的目标是返回 top-k 个与查询 Q 相关的实体。

系统中只包含 *User* 和 *Item* 两个集合时，该系统等价于传统的推荐系统。当用户的查询输入为某用户 $Q=\{u\}$ 时，系统返回 $f(u, e_{ij})$ 的 top-k 个实体。

2.2 数据模型

在推荐系统中，除了 *User* 和 *Item* 两个领域集合外，还有许多的用户上下文信息。例如在电影推荐网站上，除了用户对电影的显示评分外，还包含用户对电影的收看记录、购买记录以及租用记录等。本文把每种类型的记录都看作推荐系统的一个领域，并把记录值作为用户在该领域的属性值。我们将用户的上下文信息看作一个事件集合 $L=\{v_1, v_2, \cdots, v_r\}$，其中每个事件是一个包含多个领域值的元组 $(e_{D_1}, e_{D_2}, \cdots, e_{D_n})$，并且 $e_{D_i} \in D_i$。表 1 是一个简单的电影租用历史记录。在表 1 中，我们可以定义 4 个领域集合：*USER* = {'Matt', 'Jack'}，*MOVIE* = {'Superman', '300', 'Rent'}，*DATE* = {'09-16-2008', '09-17-2008'} 和 *LOCATION* = {'Seoul'}，事件集合 L 包含 ('Matt', 'Superman', '09-16-2008', 'Seoul')，('Jack', '300', '09-17-2008', 'Seoul') 和 ('Matt', 'Rent', '09-17-2008', 'Seoul') 3 个元组。

表 1 电影租用记录表

USER	MOVIE	DATE	LOCATION
Matt	Superman	09-16-2008	Seoul
Jack	300	09-17-2008	Seoul
Matt	Rent	09-17-2008	Seoul

为了便于讨论，我们假设每个领域的元素取值范围是有限的。当用户的查询请求为 $Q = D_1 \cup D_2 \cup \cdots \cup D_k$ 时，将 $f(Q, e_{ij})$ 划分为 k 个子查询，$f(D_1, e_{1j})$ 和 $f(D_i, e_{ij})$ $(2 \leqslant i \leqslant k)$，然后将这 k 个子查询进行加权和，$w_1 + w_2 + \cdots + w_k = 1$。

在对子查询进行计算的过程中，我们建立一个有向的二部图 $G=(V, E)$。节点集合 $V = V_1 \cup V_2 \cup \cdots \cup V_k \cup V_T$，其中 $V_i = D_1 \times D_i$ $(1 \leqslant i \leqslant k)$ 来源于 $f(D_i, e_{ij})$，V_T 来源于目标领域。边的集合 E 是一个从 $V_1 \cup V_2 \cup \cdots \cup V_k$ 到 V_T 的二部图。如果令 $Q = \{USER, DATE, LOCATION\}$，那么图 1 为表 1 的数据在查询 Q 下所构成的二部图。

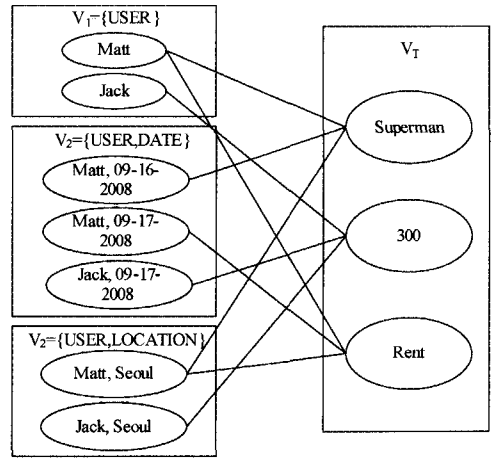

图 1 电影租用记录二部图

对于给定的两个顶点 $v_i \in V_1 \cup V_2 \cup \cdots \cup V_k$ 和 $v_j \in V_T$，令边 (i, j) 的权值 $w_{i,j}$ 为该边在事件集合中出现的次数，那么有 $w_{i,j} = w_{j,i}$。此外，由于对不同的子查询赋予了不同的权重，我们将图 G 的矩阵 M 定义如下：

$$m_{i,j}=\begin{cases} w_{i,j}, & v_i \notin V_T \wedge v_j \in V_T \\ w_{i,j} \times w_k, & v_i \in V_T \wedge v_j \notin V_T \wedge v_j \in V_k \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

式中，w_k 为子查询 $f(D_k, e_{kj})$ 所占的权重。于是，图 G 的转移概率矩阵为：

$$p_{i,j}=\frac{m_{i,j}}{\sum_{j=1}^{|V|} m_{i,j}} \quad (2)$$

式中，$|V|$ 为图 G 中节点的个数。

2.3 节点个性化排名方法

本文采用 Personalized PageRank[19] 作为二部图中节点排名的方法。Personalized PageRank 的基本思想是：图中节点的取值与其父节点(in-links)的值成正比，与其父节点的子节点(out-links)个数成反比。给定一个图 $G=(V,E)$，其节点个取值向量 π_u 为：

$$\pi_u=(1-\varepsilon)P^T\pi_u+\varepsilon\delta_u \quad (3)$$

式中，ε 为阻尼系数，通常取值为 0.15；δ_u 是一个 $|V|$ 维列向量，并且满足：

$$\delta_u(v)=\begin{cases} 1, & u=v \\ 0, & u\neq v \end{cases} \quad (4)$$

Personalized PageRank 的随机游走模型是：给定图 G 和个性化节点 u，以 u 为出发点开始随机游走；在每一步游走中，以概率 ε 返回到初始节点 u，以概率 $(1-\varepsilon)$ 沿着当前节点的出边随机选取一个目标点。式(3)得到的节点取值向量就是随机游走在稳态时停留在该节点的概率。

在本文所述的推荐系统中，用户的推荐查询 Q 是一个包含多个领域的查询，其结果是一个多维的实体集合。给定查询 $Q=\{e_1,e_2,\cdots,e_k\}$，定义向量 $\vec{q}$ 如下：

$$\vec{q}_i=\begin{cases} 1, & E_{v_i}\subseteq Q \\ 0, & \text{其他} \end{cases} \quad (5)$$

式中，E_{v_i} 是与 v_i 相关的实体集合。我们对 $\vec{q}$ 进行规范化，得到：

$$\vec{q}_i'=\frac{\vec{q}_i}{\sum_{j=1}^{k}\vec{q}_j} \quad (6)$$

再令 $\delta_u=\vec{q}'$，并将其代入到式(3)，得到：

$$\pi_u=(1-\varepsilon)P^T\pi_u+\varepsilon\vec{q}' \quad (7)$$

根据用户的推荐查询 Q，我们应用式(7)计算节点的 Personalized Page-Rank 取值，并根据节点的 Personalized Page-Rank取值进行个性化排名。在结果返回时，我们将 V_T 中排名靠前的 top-k 项作为查询 Q 的结果返回给用户。

3 实验

3.1 评价标准

当用户输入查询时，推荐系统返回 top-k 个结果。这里我们不考虑用户对商品的具体估计值，而关心的是结果的 top-k 项。因此，为了评估推荐算法的准确性，本实验应用 HR@k[17] 作为算法的准确性度量标准。HR@k 的定义如下：

$$HR@k=\frac{\#hit}{k} \quad (8)$$

式中，k 为返回的 top-k 个结果，#hit 为该 k 个结果中在真实数据集中排在前 k 个的个数。

3.2 数据集

本文采用两个真实的数据集即 last. fm 和 LG Oz Store 对算法的性能进行评估。last. fm 数据集包含 1699 个用户、3000 首歌曲，以及 673293 个用户日志记录。LG Oz Store 是一个移动应用市场数据集，包含 4000 个用户、4000 个应用，以及 862291 个用户购买记录。在上述两个数据集中，日志记录包含了用户、项及时间属性。为了获得多维的数据信息，我们将时间属性进一步划分为 Day、DOW(day of week)和 Month。在数据集中，数据的日志记录按照时间进行升序排列，我们应用数据集的前 90% 作为训练集合，并用剩下的 10%作为测试数据。

3.3 结果分析

我们用 MultiD 来表示本文提出的算法，并与 POP[20]、UKNN[21]、PSVD[11] 和 IRANK[22] 算法进行了对比。

首先，我们应用本文提出的算法进行 top-k 结果的推荐。为了测试算法对指定用户进行推荐，我们应用不同的领域集进行推荐。表 2 为本实验采用的领域集分布及相应的权重分布表。

表 2 多维权重分配表

Query Set	Weight
F1={f1={USER}}	w1=1.0
F2={f1={USER,Day}}	w1=1.0
F3={f1={USER,DOW}}	w1=1.0
F4={f1={USER,Month}}	w1=1.0
F5={f1={USER}, f2={USER,Day}}	w1=0.5, w2=0.5
F6={f1={USER}, f2={USER,DOW}}	w1=0.5, w2=0.5
F7={f1={USER}, f2={USER,Month}}	w1=0.5, w2=0.5
F8={f1={USER}, f2={USER,Day}, f3={USER,DOW}, f4={USER,Month}}	w1=0.25, w2=0.25, w3=0.25, w4=0.25

我们应用上述的查询集合及相应的权重分布对本文提出的算法进行了评估，实验结果如表 3 所列。从表 3 中我们可以看出，在两个数据集下，F8 对几个推荐结果有较好的准确性。实验结果表明，当用户查询包括多个属性时，综合多个属性的推荐值可以进一步提高推荐结果的准确性。

表 3 查询属性对推荐结果影响

HR@k	OZSTORE							
	F1	F2	F3	F4	F5	F6	F7	F8
HR@10	0.048	0.046	0.028	0.008	0.008	0.058	0.057	**0.062**
HR@20	0.089	0.070	0.060	0.022	0.018	0.100	0.095	**0.103**
HR@30	0.118	**0.147**	0.091	0.039	0.031	0.125	0.135	0.130
HR@40	0.139	0.170	0.106	0.064	0.047	0.159	**0.183**	0.169
HR@50	0.157	0.207	0.138	0.091	0.070	0.195	0.223	**0.233**
HR@k	LASTFM							
	F1	F2	F3	F4	F5	F6	F7	F8
HR@10	0.096	0.011	0.015	0.098	0.024	0.097	0.095	**0.099**
HR@20	0.158	0.022	0.024	0.160	0.037	**0.164**	0.161	**0.164**
HR@30	0.193	0.038	0.039	0.208	0.044	**0.210**	0.209	**0.210**
HR@40	0.232	0.047	0.047	0.240	0.054	0.248	0.242	**0.249**
HR@50	0.274	0.057	0.053	**0.294**	0.062	0.291	0.270	0.292

其次，我们将本文提出的算法 MultiD 与 POP、UKNN、PSVD 和 IRANK 进行了对比。为了对比算法在不同 top-k 下的准确性，我们分别对比了算法在 top-10，top-20，top-30，top-40 和 top-50 下的准确性，结果如表 4 所列。从表 4 中可以看出，本文提出的 MultiD 算法在多数情况下都有好的准确性。

表 4　推荐结果性能比较表

HR@k	OZSTORE				
	POP	UKNN	PSVD	IRANK	MultiD
HR@10	0.027	0.041	0.050	0.045	**0.051**
HR@20	0.043	0.084	**0.090**	0.081	0.089
HR@30	0.075	**0.124**	0.123	0.108	0.118
HR@40	0.100	0.137	0.152	0.135	**0.153**
HR@50	0.126	0.157	0.149	0.153	**0.157**
HR@k	LASTFM				
	POP	UKNN	PSVD	IRANK	MultiD
HR@10	0.011	0.014	0.05	0.088	**0.096**
HR@20	0.024	0.036	0.090	0.151	**0.158**
HR@30	0.036	0.062	**0.121**	0.181	0.193
HR@40	0.044	0.078	0.152	0.215	**0.232**
HR@50	0.053	0.097	0.181	0.241	**0.274**

结束语　灵活性是评价推荐系统性能的重要指标之一。本文应用用户的上下文信息，建立了一个多维数据模型的推荐系统。通过将用户的查询分解为多个子查询，并建立相应的二部图，然后应用随机游走模型将候选项排序，该系统能灵活反映出用户的多种推荐请求，并且具有很好的准确性。

参 考 文 献

[1] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6): 734-749

[2] Steck H. Training and testing of recommender systems on data missing not at random[C]// Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2010: 713-722

[3] Adomavicius G, Tuzhilin A, Zheng R. REQUEST: A query language for customizing recommendations[J]. Information Systems Research, 2011, 22(1): 99-117

[4] 赵琴琴，鲁凯，王斌. SPCF：一种基于内存的传播式协同过滤推荐算法[J]. 计算机学报，2013，36(3)：671-676

[5] 刘艳，郝忠孝. 一种基于主存 Δ-tree 的高维数据 KNN 连接算法[J]. 计算机研究与发展，2010(7)：1234-1243

[6] Wei K, Huang J, Fu S. A survey of e-commerce recommender systems[C]// 2007 International Conference on Service Systems and Service Management. IEEE, 2007: 1-5

[7] 黄创光，印鉴，汪静，等. 不确定近邻的协同过滤推荐算法[J]. 计算机学报，2010，33(8)：1369-1377

[8] Wei S, Ye N, Zhang S, et al. Item-Based Collaborative Filtering Recommendation Algorithm Combining Item Category with Interestingness Measure[C]// 2012 International Conference on Computer Science & Service System (CSSS). IEEE, 2012: 2038-2041

[9] Ling G, Yang H, Lyu M R, et al. Response aware model-based collaborative filtering[R]. UAI-P-2012-PG-501-510. 2012

[10] Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks[C]// Proceedings of the fourth ACM conference on recommender systems. ACM, 2010: 135-142

[11] Cremonesi P, Koren Y, Turrin R. Performance of recommender algorithms on top-n recommendation tasks[C]// Proceedings of the fourth ACM conference on recommender systems. ACM, 2010: 39-46

[12] Lee C H. Mining spatio-temporal information on microblogging streams using a density-based online clustering method[J]. Expert Systems with Applications, 2012, 39(10): 9623-9641

[13] Ghazanfar M A, Prugel-Bennett A. A scalable, accurate hybrid recommender system[C]// Third International Conference on Knowledge Discovery and Data Mining, 2010, WKDD' 10. IEEE, 2010: 94-98

[14] Harvey M, Carman M J, Ruthven I, et al. Bayesian latent variable models for collaborative item rating prediction[C]// Proceedings of the 20th ACM international conference on information and knowledge management. ACM, 2011: 699-708

[15] Koutrika G, Bercovitz B, Garcia-Molina H. FlexRecs: expressing and combining flexible recommendations[C]// Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. ACM, 2009: 745-758

[16] Adomavicius G, Tuzhilin A. Context-aware recommender systems[M]. Recommender Systems Handbook, US: Springer, 2011: 217-253

[17] Lee D, Park S E, Kahng M, et al. Exploiting contextual information from event logs for personalized recommendation[M]. Computer and Information Science 2010. Berlin Heidelberg: Springer, 2010: 121-139

[18] Kahng M, Lee S, Lee S. Ranking in context-aware recommender systems[C]// Proceedings of the 20th international conference companion on World Wide Web. ACM, 2011: 65-66

[19] Jeh G, Widom J. Scaling personalized Web search[C]// Proceedings of the 12th international conference on World Wide Web. ACM, 2003: 271-279

[20] Zhuang F, Karypis G, Ning X, et al. Multi-view learning via probabilistic latent semantic analysis[J]. Information Sciences, 2012, 100: 20-30

[21] Burke R, O'Mahony M P, Hurley N J. Robust collaborative recommendation[M]// Recommender Systems Handbook. US: Springer, 2011: 805-835

[22] Gori M, Pucci A, Roma V, et al. Itemrank: A random-walk based scoring algorithm for recommender engines[C]// Proceedings of the 20th international joint conference on Artifical intelligence. Morgan Kaufmann Publishers Inc., 2007: 2766-2771