基于最短路径信任关系的推荐项目计算方法

刘贵松，解修蕊，黄海波，屈 鸿

(电子科技大学计算机科学与工程学院 成都 611731)

【摘要】针对社交网络中协同过滤推荐算法的推荐速度计算问题，提出了一种基于最近邻方法的改进计算方法，并对算法有效性进行了分析。该算法对用户的相似性度量采用基于最短路径的信任关系，用分层图和动态规划的方法进行计算，并在社交网络的应用中对关系链的深度进行限制。对该算法基于KDD Cup 2012 Track 1的数据进行了仿真，并与其他方法做了性能比较。实验表明，改进算法可以很好地平衡推荐效率和准确率指标。

关 键 词 协同过滤； 推荐系统； 相似性度量； 最短路径； 信任关系
中图分类号 TP393 文献标志码 A doi:10.3969/j.issn.1001-0548.2014.02.001

Fast Computing Method for Items Recommendation Based on Shortest-Path Trust Relationship

LIU Gui-song, XIE Xiu-rui, HUANG Hai-bo, and QU Hong

(School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 611731)

Abstract In order to increase the speed of collaborative filtering recommendation in social networks, an improved nearest-neighbor algorithm is proposed in this paper. The proof of its correctness is also given in detail. The similarity measurement between users is based on trust relationship by using shortest path method. Layered graph and dynamic programming are applied to calculate the similarity. Furthermore, the recommendation speed can also be improved by limiting the depth of relationship chain in practical applications of social networks. The comparative simulations are carried out based on the KDD Cup 2012 Track 1 datasets. The results show that the better balance between the accuracy and the recommendation efficiency can be achieved by the proposed algorithm.

Key words collaborative filtering; recommender system; similarity measurement; shortest path; trust relationship

推荐系统属于信息系统研究的一个分支，在 20 世纪 90 年代中期逐渐成为一个独立的研究领域[1]。目前推荐系统的研究热度不减，在理论研究和实际应用方面取得了很多研究成果[2]，文献[3]给出了推荐系统研究的最新的全面综述。有效的推荐系统需要尽可能搜集有效的用户信息，而社交网络恰恰提供了大量真实世界中人们之间的社交信息，这对于进行快速精确推荐意义重大。然而，随着社交网络使用人数的急剧增加，用户分享数据规模的不断加大，推荐系统研究中出现了如大数据处理和算法效率、数据稀疏性和冷启动、多样性和精确性平衡等挑战[4]。

目前，社交网络的推荐算法主要基于协同过滤，旨在实现向用户推荐过去和此用户喜好相似的其他用户所喜欢的项目[5]。基于相似性的方法(similarity-based methods)通过用户/项目自身特征计算用户相似性或项目相似性，从而给出合理的推荐[3,6]。因此，快速计算用户与用户或项目与项目之间的相似性在协同过滤算法中非常重要。

协同过滤作为个性化推荐的重要方法，目前仍然面临诸多挑战。协同过滤方法可以分为最近邻方法和基于模型的方法两大类[7]。在最近邻方法中，对于每个用户，系统找到和该用户在某些测度下最接近的一组用户，并推荐这些用户打分较高的项目[8]。

本文提出了一种基于最近邻方法的改进优化算法，传统方法对用户相似性度量主要是采用对最短路径用户枚举进行计算，本文算法主要采用分层和动态规划的方法计算用户相似性参数。另外，在社

收稿日期：2012－09－21；修回日期：2013－09－20
基金项目：国家自然科学基金(61273308)；中央高校基本科研业务费(ZYGX2013J076)
作者简介：刘贵松(1973－)，男，博士，副教授，主要从事计算智能、模式识别方面的研究.

交网络的应用中，还可以考虑实际情况对关系链深度做出限制，以提高推荐速度。

1　最近邻协同过滤推荐算法

1.1　用户特征选择

对于一个用户，定义其特征向量为 $\boldsymbol{u}_i=(p_{i1},p_{i2},\cdots,p_{in})$，其中 p_{ij} 是用户 i 与项目 j 的关联度，系统中共有 n 个待推荐项目。关联度 p_j 的选择应该考虑项目热度、用户行为数的归一化因素：对于热门项目，即大量用户都选择过的项目，相应的权值应当降低；对于行为较多的用户，某一个单一项目的一致并不能说明偏好的相似，也有可能是偶然产生。而对于行为总数较少的用户，每一个相同的行为都更能表明偏好的一致性。

考虑到以上两个因素，选用TF/IDF权重(term frequency-inverse document frequency)作为特征。这一方法在信息检索和文本挖掘领域被广泛使用[8]。

$$p_{ij}=\mathrm{tf}(a_{ij})\times\mathrm{idf}(j) \tag{1}$$

$$\mathrm{tf}(a_{ij})=a_{ij}\Big/\sum_j a_{ij} \tag{2}$$

$$\mathrm{idf}(j)=\log\left(|\{u\}_i|\Big/|\{u_i\mid a_{i,j}\neq 0\}|\right) \tag{3}$$

式中，a_{ij} 为用户 i 对项目 j 的行为数，如点击次数、查看次数、微博转发次数等；tf(a_{ij})为对用户本身行为数的归一化，用于衡量这一项目对用户的重要程度；idf(j)为对项目 j 本身流行程度的归一化，用于衡量项目本身的重要程度。

1.2　用户相似性度量

假设所有用户关系的集合为 E。即：

$$E=\{<i,j>\mid j\text{是}i\text{的朋友}\} \tag{4}$$

定义直接相连的用户 i 与 j 之间的权重 $\boldsymbol{W}$：

$$\boldsymbol{W}(<i,j>)=\cos(\boldsymbol{u}_i,\boldsymbol{u}_j)=\frac{\boldsymbol{u}_i\boldsymbol{u}_j}{|\boldsymbol{u}_i||\boldsymbol{u}_j|},\quad <i,j>\in E \tag{5}$$

显然，$0\le\boldsymbol{W}\le 1$。

对于一个关系链，定义其权重为其中各边权重之积[10]。即：

$$\boldsymbol{W}(<v_0,v_1,\cdots,v_n>)=\prod\boldsymbol{W}(<v_i,v_{i+1}>)\quad <v_i,v_{i+1}>\in E \tag{6}$$

对于两个不直接相连的用户 i 与 j，其权重为两者之间所有最短路径的权重之和。

$$\boldsymbol{W}(<i,j>)=\sum_{\text{最短路}p}\boldsymbol{W}(p)\qquad <i,j>\notin E \tag{7}$$

而最终对 i 推荐的项目通过对所有其他用户的加权和得到：

$$\boldsymbol{R}(i)=\sum_{j\neq i}\boldsymbol{W}(<i,j>)*\boldsymbol{u}_j \tag{8}$$

式(8)中得到的向 $\boldsymbol{R}$ 量中，每个元素对应一个待推荐项目的分值。一般认为分值越高的项目系统更加值得推荐。在实际使用中，还必须排除用户已经选择过的项目。

2　算法改进与分析

2.1　算法改进

对于上述算法，计算中需枚举每一个目标用户，并通过每一条最短路依次计算每条边权重的乘积，效率较低。本文提出一种基于分层图及动态规划的快速计算方法。

首先定义集合 D_i 为到当前用户 u 距离为 i 的用户的集合：

$$D_0=\{u\} \tag{9}$$

$$D_i=\{u'\mid <u',v>\in E,v\in D_{i-1},u'\notin D_{i-1}\} \tag{10}$$

显然，处于最短路上的有向边的集合为：

$$\mathrm{ShortestPath}=\{<u,v>\mid u\in D_i,v\in D_{i-1}\} \tag{11}$$

对每个用户，定义 $\boldsymbol{P}$ 为：

$$\boldsymbol{P}(u)=\boldsymbol{u}+\sum_{<v,u>\in\mathrm{ShortestPath}}\boldsymbol{P}(v)*\boldsymbol{W}(<v,\boldsymbol{u}>) \tag{12}$$

则对于待推荐用户 u_0，$\boldsymbol{R}(u_0)=\boldsymbol{P}(u_0)-u_0$。

在实际使用中，可以通过限制关系链的深度来减少计算量，因为实际中 3 层以上的朋友关系对结果的影响非常小。

算法中首先从待推荐用户开始，广度优先搜索，查找一定范围内的所有用户，将用户列表依照距离次序进行保存。按照从远到近的顺序计算每个用户的 $\boldsymbol{R}$ 的值，对于 u_0，其 $\boldsymbol{R}$ 即是各个待推荐项目的权重。系统可以选择权重最大的一个或多个项目，向用户进行推荐。具体算法描述如下。

改进的推荐项目计算算法：

建立数组 q

head ← 0

tail ← 1

dist(u) ← 正无穷

dist(u_0) ← 0

q(0) ← u_0

$\boldsymbol{R}(\boldsymbol{u})\leftarrow\boldsymbol{u}$ 的特征向量

while head < tail do

　for all <q(head), u>∈E and dist(u) = 正无穷 do

　　q(tail) ← u

dist(u) ← dist(q(head)) + 1
end for
head ← head + 1
end while
while tail >= 0 do
tail ← tail – 1
for all <q(tail), u>∈E and dist(u) =
dist(q(tail)) + 1 do
$\boldsymbol{R}(q(\text{tail})) \leftarrow \boldsymbol{R}(q(\text{tail})) + \boldsymbol{R}(u) * \boldsymbol{W}(<u, q(\text{tail})>)$
end for
end while

2.2 算法分析

对算法正确性可作简单分析：假设所有用户集合为 User，当前待推荐用户为 u_0，对于某个用户 $u \in D_n$，设用户 u 的所有最短路径的集合为：

$$\mathrm{SP}(u)=\{<u,v_{n-1},\cdots,u_0>|\, v_i \in D_i \text{且} <v_i,v_{i-1}>\in E\} \tag{13}$$

易知：

$$\boldsymbol{W}(<v_0,v_1,\cdots,v_n>)= \boldsymbol{W}(<v_0,v_1,\cdots,v_{n-1}>)*\boldsymbol{W}(<v_{n-1},v_n>) \tag{14}$$

为推导方便，设：$\boldsymbol{W}(<u>)=1$，归纳假设：

$$\boldsymbol{P}(u)= \sum_{u'\in \mathrm{User}} \sum_{<u',\cdots,v_{n+1},u,v_{n-1},\cdots,u_0>\in \mathrm{SP}(u)} \boldsymbol{u}'*\boldsymbol{W}(<u',\cdots,v_{n+1},u>) \tag{15}$$

则对于另一用户 $t \in D_{n-1}$，将式(14)和式(15)代入式(12)得：

$$\boldsymbol{P}(t)=\boldsymbol{t}+\sum_{<x,t>\in \mathrm{ShortestPath}} \boldsymbol{P}(x)*\boldsymbol{W}(<x,t>)=$$

$$\boldsymbol{t}\times\boldsymbol{W}(t)+\sum_{<x,t>\in \mathrm{ShortestPath}} \boldsymbol{W}(<x,t>)*$$

$$\sum_{u'\in \mathrm{User}} \sum_{<u',\cdots,v_{n+1},x,v_{n-1},\cdots,u_0>\in \mathrm{SP}(x)} \boldsymbol{u}'*\boldsymbol{W}(<u',\cdots,v_{n+1},x>)=$$

$$\boldsymbol{t}\times\boldsymbol{W}(t)+\sum_{<x,t>\in \mathrm{ShortestPath}} \sum_{u'\in \mathrm{User}} \sum_{<u',\cdots,v_{n+1},x,t,v_{n-2},\cdots,u_0>\in \mathrm{SP}(x)} \boldsymbol{u}'*\boldsymbol{W}(<u',\cdots,v_{n+1},x,t>)=$$

$$\sum_{u'\in \mathrm{User}} \sum_{<u',\cdots,v_n,t,v_{n-2},\cdots,u_0>\in \mathrm{SP}(u)} \boldsymbol{u}'*\boldsymbol{W}(<u',\cdots,v_n,t>)$$

即原假设成立。对于待推荐用户 u_0：

$\boldsymbol{R}(u_0)=\boldsymbol{P}(u_0)=\sum_{u'\in \mathrm{User}} \sum_{p\in \mathrm{SP}(u)} \boldsymbol{u}'*\boldsymbol{W}(p)$ 为最终得到的推荐结果。

3 实验与分析

本文采用KDD Cup 2012 Track 1(腾讯微博)数据[11]，数据内容包括用户行为数据、用户信息、每个推荐目标信息等，分为训练集(包含用户是否接受推荐标签)和测试集。腾讯微博拥有超过两亿注册用户，真实用户数据规模庞大。数据中主要包含用户的各项信息，目的是预测用户是否应当关注某人或某组织，进而给出关注推荐。在2012竞赛中，针对Track1的数据，文献[12]提出FFM(feature-based factorization models)模型与AFM(additive forest models)模型相结合的方法，充分考虑数据中用户的年龄、性别，微博关键词以及用户标签等因素，得到推荐的平均精度MAP(mean average precision)为42.65%，排名第一；文献[13]采用FM(factorization machines)方法，对影响推荐结果的因素进行了分类研究，将所有的影响因子视为类别变量，进而作为FM的参数。FM方法同时在Tack1和Track2数据集上取得了很好的效果，其中在微博推荐数据集上得到MAP值为41.62%，排名第二。如何客观有效地评价推荐系统仍然非常困难，文献[14]对评价指标进行了系统论述。为便于比较，本文采用与KDD Cup 2012竞赛相同的评估方法：给出每个用户的推荐用户数目为3，根据官方提供的评估数据(用户是否点击命中)计算平均精度MAP值。

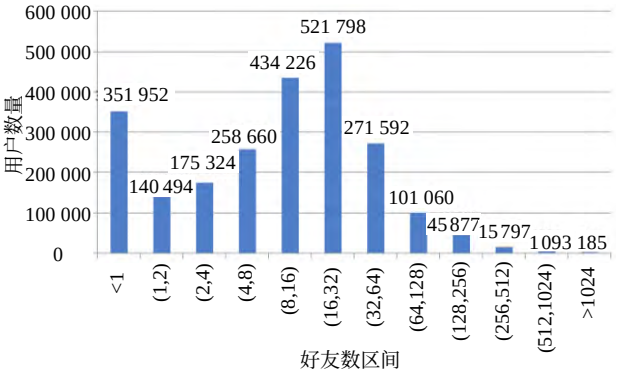

图 1 用户好友数分布

权重信息在复杂网络链路预测中起到非常重要的作用，但目前的研究尚未有明确的答案[12]。本文实验中，对于用户的行为数据，由于转发、评论与引用的作用类似，故本文实验中将这些行为的总数作为用户之间的权重度量。建图过程中，除用户关注情况以外，权重的大小作为建立用户之间有向图关系的阈值，使用本文改进算法计算对每个用户应当推荐的项目。首先以权重 5 为例(即转发、评论、引用总数 5 次以下忽略)，对数据集中用户关系网数据进行统计，结果如图 1 所示。

初步分析表明，大量用户只有 0 或 1 个好友，主要由于系统中存在大量不活跃的用户。对于多数

活跃用户，好友数通常在 20 多的数量。有少量用户则存在非常多的好友，有的甚至超过 1 000 人；这些往往是一些机器人或者有特殊目的的用户；而好友最多的用户有 5 313 个好友。由此也可以看出，整个网络非常稀疏。即使是好友最多的用户，其好友数与总用户数相比，比例也非常低。

好友权重设置的大小直接影响到用户关系图的复杂度，同时对计算以及推荐精度也有一定的影响。实验中，将权重设置为 3～10，计算向所有用户推荐的可关注用户列表(推荐值为 3)。在一台普通的笔记本电脑上单线程计算，得到推荐平均精度变化如图 2 所示。可以看出在权值设为 5 或 6 时可以获得较高的推荐精度，分别为 0.387 2 和 0.386 5。这个精度结果在 KDD Cup 2012 微博推荐竞赛中可排在第 12 名(第 11 名 MAP 值为 0.388 07)。

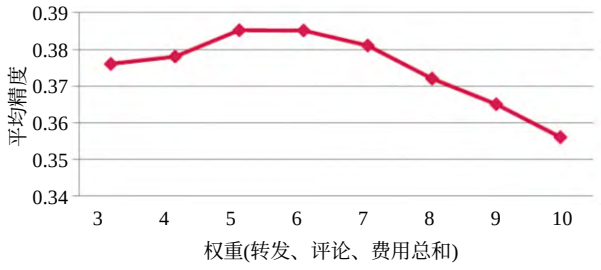

图 2　权重对平均精度的影响

当权重为 5 或 6 时，程序运行耗时约 17 h 时，平均每个用户的计算耗时约 0.026 s，基本上可以满足在线实时应用的需求。当权重越大时，用户关系图相对越稀疏，计算效率相对提高(权重为 10 时，每个用户计算耗时约 0.021 s)，但实验表明推荐平均精度下降。

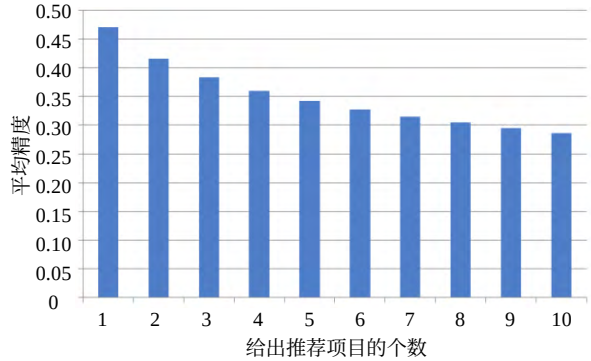

图 3　推荐项目个数与平均精度

对于每个用户，推荐特定个数的待关注用户，一般总是取预测分值最高的几个用户进行推荐。对于一个较为合理的系统，推荐数量越多，精度越低，因为需要的项目较少时可以得到更加有把握的结果[15]，本文实验结果也给出了相应的验证。根据算法计算结果(权重为 5)，在这种应用方式中平均精度与推荐个数的柱状图如图 3 所示。图中显示首个推荐的正确率约为 47%，当推荐项目个数为 10 时，其正确率降低约为 28%。

另一种给出推荐的方式为只给出对其分数的预测大于某个阀值的推荐，其优点是并非每次总给定同样数量的推荐，只需要置信度高于一定值的推荐结果。多数情况下，这种方式能够减少对用户的干扰，提高用户体验。图 4 是在这种方式下推荐的精度曲线(同样长度的阀值区间内包含同样多的待推荐条目数)。本文实验中，当阀值(归一化处理)取到最大值(图中最右端)时，平均每个用户将被推荐 7 000 个以上待关注目标，此时的精度趋向于被关注人数与所有用户数的比值。

最后，将本文方法和经典的最小值方法、随机游走方法[16]做了比较，结果如图 5 所示。

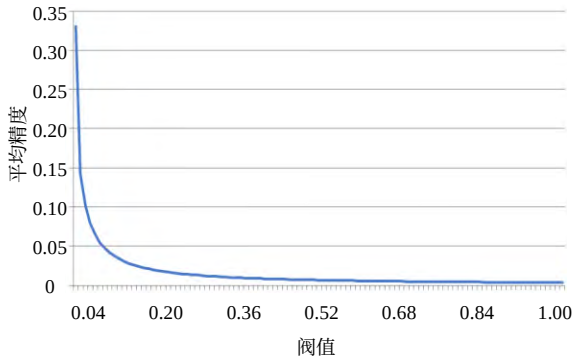

图 4　推荐阀值与准确性关系

最小值方法是使用当前用户与目标用户路径上权重最小的边的权重，作为二者之间的最终权重。随机游走方法从待推荐用户开始，每次根据当前用户所有出边的权重，加权随机选择一条路径前进；选择前进的边时，忽略最开始的待推荐用户(此处实验选择 10 步后停止，每次迭代结束后记录下停止处的用户)；这个过程重复一定次数，得到的一批用户，即认为他们是和当前用户相似的用户。当然随机游走方法随着迭代增加，其准确度也会提升(10 步和 500 步的比较)，但同时其效率也会随之降低。

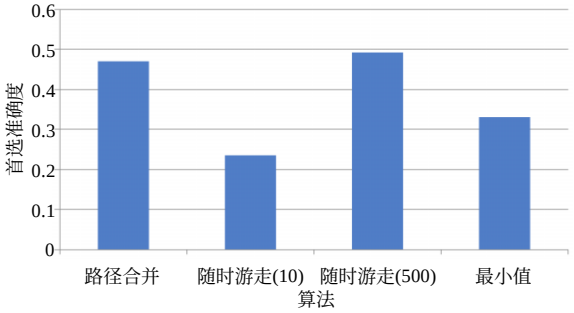

图5　方法比较

实验表明，本文提出的改进算法较好地保证了准确率，而且提高了系统的推荐效率，在推荐效率和准确率之间可以很好地平衡。

4 结论与展望

本文关注协同过滤算法中基于最短路径的信任关系传播与聚合的方法，并提出一种快速计算这种度量方法下两个用户之间信任程度的算法。基于腾讯微博数据KDD Cup 2012 Track 1得到该方法的仿真实验结果。后续工作将考虑充分利用社交网络中所有用户信息，提高推荐精度；同时要研究社交网络中采用快速K最短路径算法计算用户相似度的方法及其实际意义。

参 考 文 献

[1] RESNICK P, VARIAN H R. Recommender systems[J]. Communications of the ACM, 1997, 40(3): 56-58.

[2] RICCI F, ROKACH L, SHAPIRA B, et al. Recommender systems handbook[M]. New York: Springer, 2011.

[3] LU L, MEDO M, YEUNG C H, et al. Recommender systems[J]. Physics Reports, 2012, 519(1): 1-49.

[4] 周涛. 个性化推荐的十大挑战[J]. 中国计算机学会通讯, 2012, 8(7): 48-61.
ZHOU Tao. Ten major challenges in personalized recommendation[J]. Communications of CCF, 2012, 8(7): 48-61.

[5] SCHAFER J B, KONSTAN J A, RIEDL J. E-commerce recommendation applications[J]. Data Mining and Knowledge Discovery, 2001, 5(1/2): 115-153.

[6] LINDEN G, SMITH B, YORK J. Amazon.com recommendations: item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1): 76-80.

[7] SCHAFER J B, FRANKOWSKI D, HERLOCKER J, et al. The adaptive web: Collaborative filtering recommender systems[M]//Lecture Notes in Computer Science: 4321. Berlin, Heidelberg: Springer, 2007: 291-324.

[8] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions[J]. IEEE Trans Knowledge and Data Engineering, 2005, 17(6): 734-749.

[9] BURKE R. The adaptive web: Hybrid web recommender systems[M]//Lecture Notes in Computer Science: 4321. Berlin, Heidelberg: Springer, 2007: 377-408.

[10] RICHARDSON M, AGRAWAL R, DOMINGOS P. Trust management for the semantic web[C]//Proc Of the Second International Semantic Web Conference. [S.l.]: Springer, 2003, 2870: 351-368.

[11] DATASET. KDD Cup 2012 Track 1[EB/OL], [2012-10-20]. http://www.kddcup2012.org/.

[12] CHEN T, TANG L, LIU Q, et al. Combining factorization model and additive forest for collaborative followee recommendation[EB/OL]. [2012-10-20]. http://kddcup 2012.org/workshop.

[13] RENDLE S. Social network and click-through prediction with factorization machines[EB/OL]. [2012-10-20]. http://kddcup 2012.org/workshop.

[14] 吕琳媛. 复杂网络链路预测[J]. 电子科技大学学报, 2010, 39(5): 651-661.
LÜ Lin-yuan. Link prediction on complex networks[J]. Journal of University of Electronic Science and Technology of China, 2010, 39(5): 651-661.

[15] 朱郁筱, 吕琳媛. 推荐系统评价指标综述[J]. 电子科技大学学报, 2012, 41(2): 163-175.
ZHU Yu-xiao, LÜ Lin-yuan. Evaluation metrics for recommender systems[J]. Journal of University of Electronic Science and Technology of China, 2012, 41(2): 163-175.

[16] JAMALI M, ESTER M. TrustWalker: a random walk model for combining trust-based and item-based recommendation[C]//Proc of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [S.l.]: ACM, 2009: 397-406.

编 辑 蒋 晓