基于改进LSH的协同过滤推荐算法

李红梅 郝文宁 陈 刚
(解放军理工大学指挥信息系统学院 南京 210007)

摘 要 协同过滤是个性化推荐系统中应用较为成功与广泛的技术之一,影响协同过滤推荐质量的关键在于获取目标用户的k近邻用户,然后基于k近邻对其未评价的项目进行评分预测与推荐。针对用户评分数据的规模大、维度高、高度稀疏以及直接进行相似性度量的实时性差等对推荐性能的影响,提出一种基于LSH的协同过滤推荐算法,并对其进行改进。该算法基于p稳态分布的局部敏感哈希对用户评分数据进行降维与索引,并采用多探寻的机制对其进行改进,缓解多个哈希表对内存的压力,快速获取目标用户的近邻用户集合,然后采用加权方法来预测用户评分并产生推荐。标准数据集上的实验结果表明,该方法能有效克服评分数据的高维稀疏,并在保证一定推荐精度的前提下,大幅度提高推荐效率和降低内存消耗。

关键词 推荐系统,近似近邻,协同过滤,相似性度量,局部敏感哈希

中图法分类号 TP391.3 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.10.052

Collaborative Filtering Recommendation Algorithm Based on Improved Locality-sensitive Hashing

LI Hong-mei HAO Wen-ning CHEN Gang
(Institute of Command and Information System, PLA University of Science and Technology, Nanjing 210007, China)

Abstract Collaborative filtering is one of the key technologies widely applied in personalized recommendation system with great success. The critical step of collaborative filtering is to get k nearest neighbors (kNNs), which is utilized to predict user ratings and recommend. In order to improve the recommendation quality which is affected by the matter that rating data is characterized by its large scalability, high dimensionality, extreme sparsity, and the lower real-time ability by direct similarity measuring method in finding the nearest neighbors, we proposed a collaborative filtering recommendation algorithm based on locality-sensitive hashing, and improved it. The algorithm applies locality-sensitive hashing technology based on p-state distribution to get lower dimensionality and index for large rating data. Then a multi-probe mechanism is utilized to improve the algorithm with great efficiency in obtaining the approximate nearest users of target user. Then, a weighted method is used to predict the user ratings, and finally perform collaborative filtering recommendation. Experiment results on typical dataset show that the proposed algorithm can overcome the limitation of high dimensionality and sparsity in some degree, and has good recommendation performance, high efficiency and less memory consumption.

Keywords Recommendation system, Approximate nearest neighbor, Collaborative filtering, Similarity measuring, Locality-sensitive hashing

1 引言

个性化推荐系统通过捕捉用户的行为偏好与信息需求,主动为用户提供最符合其兴趣度的信息服务,满足用户对信息与服务的个性化需求,帮助用户缓解"信息海洋"查找等问题[1]。目前,个性化推荐系统已成功应用于社会网络、电子商务等主流Web服务,如Netflix、Amazon、淘宝等,并日益受到各领域的关注与研究。

协同过滤是目前推荐系统中研究与应用非常广泛的技术之一。不同于基于内容的推荐,协同过滤推荐建立在用户对项目的评分上,不需要考虑项目内容本身,因此不受内容限制,且推荐效果具有多样性和新颖性。基于协同过滤的推荐建立在群体用户的行为分析或兴趣相似性度量的基础上,通过收集用户对信息的评价或其它行为,搜索与其兴趣相似的用户,然后根据相似用户对其它信息的评价向当前用户产生推荐结果。

近邻关系模型是当前应用较为成功的协同过滤推荐技术之一[2],包括基于用户的协同过滤和基于项目的协同过滤技术。其中,基于用户的协同过滤推荐技术是通过目标用户的最近邻对其未评价的项目进行评分预测,以产生推荐群。该种协同过滤的关键是基于用户的相似性度量来寻找近邻用户,其推荐效能严重依赖于相似性度量的准确性以及相似性

到稿日期:2014-05-18 返修日期:2014-07-29

李红梅(1990—),女,硕士生,主要研究方向为中文信息检索、数据挖掘;郝文宁(1971—),男,博士,教授,主要研究方向为海量高维数据归约、作战效能评估;陈 刚(1974—),男,硕士,副教授,主要研究方向为作战指挥训练模拟。

空间搜索的效能。

随着用户和资源的激增，推荐系统的规模逐渐扩大，协同过滤算法面临着大规模、高维与稀疏的用户评分数据，推荐算法的适应性及推荐的实时性受到严重影响，进而影响推荐性能。针对此类问题，多种方法被提出，包括用户评分填值、项目聚类分析、降维技术、关联规则分析等。

Zhang 等[3]基于非负矩阵分解技术来预测用户评分并填值，以提高协同过滤推荐质量，但该方法仍直接采用传统相似性度量来计算用户间的相似性，没有考虑用户或项目的类别信息给推荐带来的影响，推荐精度不够高。基于项目聚类[4]和基于用户聚类[5]的方法用于减少最近邻居查找的时间并提高相似性度量准确性，但聚类分析面临着聚类类别多维性、度量标准难以控制等问题，从而影响推荐精度。Goldberg 等[7]利用主成分分析的降维技术进行协同过滤，成功将其用于笑话推荐系统 Jester。文献[8]使用奇异值分解得到低阶近似矩阵来进行协同过滤推荐，文献[9]对其进行改进，以提高评分预测准确性，该类方法可有效降低项目空间维数，同时提取项目的隐含特征，但面对大规模高维数据矩阵时其分解效率低，推荐效率不太理想；同时在面对大规模数据的最近邻查找时，传统方法会出现离线计算量较大、扩展性不强的缺陷，严重影响推荐效率。

因此，本文在分析传统最近邻协同过滤技术的基础上，提出了一种基于改进 LSH 的协同过滤推荐方法。该方法引入当前应用最为流行的局部敏感哈希技术，并对其进行改进，然后基于近邻用户集合的用户评分预测实现协同过滤推荐。该方法能有效解决用户评分数据的高维、稀疏以及传统上直接利用相似性度量进行近邻空间搜索的低效问题。

本文第 2 节介绍相关工作，即常用的相似性度量方法和相似性空间搜索策略；第 3 节引入一种近似近邻搜索技术——局部敏感哈希技术，并基于局部敏感哈希算法对海量高维数据进行降维和索引处理，快速获取目标用户的近似近邻用户集合；第 4 节在原有基于 p 稳态分布的局部敏感哈希函数基础上，引入多探寻机制进行改进，缓解多个哈希表对内存的压力，然后采用加权方法预测评分，进而产生推荐；第 5 节利用真实的电影评分数据集进行实验测试与分析；最后，总结全文工作，分析不足及进一步的研究方向。

2 相关工作

2.1 常用的相似性度量方法

基于用户的协同过滤中，可以通过相似性度量来完成发现用户关系群，通过比较用户间的相关性或相似性，来确定用户关系的紧密程度。相似性度量主要通过对象的特征向量或属性集合间的相似系数、相关系数以及相异距离等来计算，包括余弦相似系数、修正的余弦相似系数、皮尔逊相关系数、Jaccard 相似系数、欧氏距离等。

(1)余弦相似系数：用户间的相似性用评分向量的夹角余弦来度量。

$$sim(\boldsymbol{x},\boldsymbol{y})=\cos(\boldsymbol{x},\boldsymbol{y})=\frac{\boldsymbol{x}\cdot\boldsymbol{y}}{\|\boldsymbol{x}\|\,\|\boldsymbol{y}\|}$$

其中，每个用户评分为 d 维项目空间中的向量，用户 $\boldsymbol{x}$ 和用户 $\boldsymbol{y}$ 的项目评分向量分别为 $\boldsymbol{x}=(x_1,x_2,\cdots,x_d)^{\mathrm{T}}$、$\boldsymbol{y}=(y_1,y_2,\cdots,y_d)^{\mathrm{T}}$。

(2)修正的余弦相似系数：为减少因用户的评分标准不同而对相似性度量的影响。修正的余弦相似系数减去了用户对项目评分的平均值，以修正余弦系数，修正后的相似性为

$$sim(\boldsymbol{x},\boldsymbol{y})=\frac{\sum\limits_{c\in I_{xy}}(x_c-\bar{x})(y_c-\bar{y})}{\sqrt{\sum\limits_{c\in I_x}(x_c-\bar{x})^2}\sqrt{\sum\limits_{c\in I_y}(y_c-\bar{y})^2}}$$

(3)Person 相关系数：基于 Pearson 相关系数的用户间相似性为

$$sim(\boldsymbol{x},\boldsymbol{y})=\frac{\sum\limits_{c\in I_{xy}}(x_c-\bar{x})(y_c-\bar{y})}{\sqrt{\sum\limits_{c\in I_{xy}}(x_c-\bar{x})^2}\sqrt{\sum\limits_{c\in I_{xy}}(y_c-\bar{y})^2}}$$

其中，I_x 和 I_y 分别表示用户 $\boldsymbol{x}$ 和用户 $\boldsymbol{y}$ 评分的项目集合，I_{xy} 表示用户 $\boldsymbol{x}$ 和用户 $\boldsymbol{y}$ 共同评分项目集合，x_c 和 y_c 分别表示用户 $\boldsymbol{x}$ 和用户 $\boldsymbol{y}$ 对项目 c 的评分，$\bar{x}$ 和 $\bar{y}$ 分别表示两用户的项目平均评分。

(4)Jaccard 相似系数：Jaccard 相似系数利用两个向量集合的交集与并集之比来度量用户间的相似性，主要用来比较二元向量的相似性。用户间的相似性为

$$sim(\boldsymbol{x},\boldsymbol{y})=\frac{\boldsymbol{x}\cdot\boldsymbol{y}}{|\boldsymbol{x}|^2+|\boldsymbol{y}|^2+\boldsymbol{x}\cdot\boldsymbol{y}}$$

(5)基于欧氏距离的相似度：欧氏距离是欧氏空间的一种距离度量方式，用户之间的欧氏距离是一种 L_d 范式，当 $d=2$ 时，欧氏距离为

$$dis(\boldsymbol{x},\boldsymbol{y})=L_2=\|\boldsymbol{x}-\boldsymbol{y}\|^2$$

可见，欧氏距离越大，用户间的相似度越小。

面对不同维度与稀疏度的用户评分数据，部分相似性度量方法存在一定的缺陷。例如，对于过于稀疏的用户评分数据，当共同评分项目非常稀少时，利用皮尔逊相关系数度量相似性效果不佳，相似度计算不准确。

2.2 相似性空间搜索

当数据规模较小、维度较低时，直接利用相似性度量进行两两相似度计算，时间复杂度并不高。但面对大规模、高维的评分数据时，时间复杂度急剧上升，之前常用的很多相似性检索算法(例如基于空间划分的索引方法等)常常不可避免地陷入“维灾”困境，尤其对高度稀疏数据检索效果不理想。例如，常用的 Tree 算法用于低维数据的精确性检索时性能较好，但随着维数的增加(大于 20 维)，检索性能急剧下降。因此，获取 k 近邻的关键是如何实现快速相似性检索[10]。

精确最近邻搜索的主要问题在于几何计算，其精确解难度较大，复杂性高。但在很多实际情况下，实现 k 近邻检索时，一组近似却高效的结果比低效的精确检索结果往往更具有吸引力。因此，人们设计了一种效率更高的近似算法，即近似最近邻算法，并且基于近似近邻技术的快速索引方法已被成功应用于多领域相似性检索。下面给出 c-近似近邻搜索的定义。

定义 1(c-近似近邻搜索，Approximate Nearest Neighbor Search) 给定 d 维数据空间 R^d 的一个数据集合，以及一个近似因子 $c>1$。对于空间 R^d 中的查询点 p，若能找到一点 $q\in D$，使得对任一元 $v\in D$ 满足 $d(q,p)\leqslant c\cdot d(v,p)$，则 q 属

于 p 的 c -近似最近邻。即近似近邻点 q 到查询点 p 的距离最多不超过 p 到其最近邻点的距离的 c 倍。

局部敏感哈希(Locality-Sensitive Hashing, LSH)[11-13]是当前实现近似近邻搜索的最快与最好的解决方法，特别是针对高维稀疏数据的 c -近似近邻搜索问题[13]，它能较好地解决传统索引方法存在的"维灾"问题，是当前备受关注且应用广泛的索引技术。

3 近似近邻用户检索

3.1 相关定义

LSH 基于随机映射机制将高维空间数据映射为低维数据，并保证数据间的相近性，即原向量空间中距离较近的两个点经映射后距离仍然很近。LSH 虽是近似技术，不保证相似性检索的精确性，但在实际计算中，其通常在很小的时间复杂度下返回精确或近似精确的相似结果，能在很大程度上满足用户的需求[12]。

LSH 的基本思想是将相似的对象以较高的碰撞概率哈希到同一个哈希"桶"中，通过过滤掉大量的不相似的对象来避免不必要的相似性计算，降低相似性计算的代价，以快速获取近邻对象。实验证明，LSH 在数据规模与维度增大时仍具有良好的相似性检索性能。

LSH 函数的定义如下。

定义 2(局部敏感哈希函数) LSH 依赖于某度量空间 D 下的哈希函数族 $H=\{h:S\to U\}$，该函数族是点域 S 到某整数域 V 的一组映射函数。对于 R^d 内的查询点 p，若满足以下的条件：

(1)若 $q\in B(p,r)$，则 $\Pr[h_i(p)=h_i(q)]>p_1$

(2)若 $q\notin B(p,r)$，则 $\Pr[h_i(p)=h_i(q)]<p_2$

其中，$B(p,r)=\{q|D(p,q)\leqslant r\}$ 表示以 p 为中心、r 为半径的球，$\Pr[\cdot]$ 是概率函数，表示 p,q 哈希值相等的概率，$r>0$，p_1、p_2 为常数且 $0<p_1<p_2<1$。该位置敏感函数是基于数据点之间距离增大而的函数，数据点 p 和 q 的碰撞概率随着它们之间的距离增大而递减。

首次提出的 LSH 方法主要针对海明空间中的二进制点[14]，其存储与计算速度相对于基于树结构的空间划分方法快很多；但同时有着局限性，即该算法的输入必须位于海明空间，非二进制数据需嵌入到海明空间才能进行计算，这无疑增加了算法时空复杂性，降低了查询的准确率。文献[15]提出直接在欧氏空间计算的快速方法，该方法特别适用于处理高维稀疏数据，且运行时限比较稳定，p-稳态分布则是保证这一稳定性的基础。p-稳态分布是广义上的高斯分布，具有稳定性，即具有相同指数的 p-稳态分布的随机变量经线性组合后仍保持稳态分布。

下面给出 p-稳态分布的定义。

定义 3(p-稳态分布，p-Stable Distributions) 存在 $p\geqslant 0$ 对于实数集 R 中的任意 n 个实数 $v_1,v_2,\cdots,v_n$ 以及 R 上的一个分布 D 的独立同分布变量 $X_1,X_2,\cdots,X_n$，若随机变量 $\sum_i v_iX_i$ 和 $(\sum_i|v_i|^p)^{\frac{1}{p}}\boldsymbol{X}$ 服从同一分布(其中，$\boldsymbol{X}$ 是服从 D 分布的随机变量)，则称 D 为 p-稳态分布。对任何 $p\in(0,2]$ 都存在稳态分布[13]。

基于 p-稳态分布的哈希函数族为：

$$h_{a,b}(v)=\left\lfloor\frac{\boldsymbol{a}\cdot v+b}{w}\right\rfloor \tag{1}$$

其中，$\boldsymbol{a}$ 为 d 维空间 R^d 内服从 p-稳态分布的独立随机向量，b 为 $[0,w]$ 内的任一整数，$\lfloor\cdot\rfloor$ 为向下取整操作，该哈希函数 $h_{a,b}(v)$ 将一个 d 维空间向量 $v=(v_1,v_2)$ 映射为一个整数。

这样的哈希函数保持位置敏感的性质。设 $f_p(t)$ 代表 p-稳态分布的绝对值的概率密度函数，$e=\|v_1-v_2\|_p$，投影距离 $(\boldsymbol{a}\cdot v_1-\boldsymbol{a}\cdot v_2)$ 与 $e\boldsymbol{X}$ 同分布，很容易计算距离为 e 的两点之间的碰撞概率 $p(e)$：

$$p(e)=p[h_{a,b}(v_1)=h_{a,b}(v_2)]=\int_0^w\frac{1}{e}f_p\left(\frac{t}{e}\right)\left(1-\frac{t}{w}\right)\mathrm{d}t$$

对于固定的参数 w，冲突概率 $p(e)$ 随 $e=\|v_1-v_2\|_p$ 单调递减，因此，哈希函数 $h_{a,b}(v)$ 为 (r_1,r_2,p_1,p_2)-位置敏感哈希函数族。其中，$r_2=(1+\varepsilon)r_1$，$p_1=p(1)$，$p_2=p(1+\varepsilon)$。大量实验证明，当哈希函数中的 w 取 4 时检索效果较好。

3.2 LSH 索引构建

仅仅依靠单个 LSH 函数进行映射容易产生大量碰撞，使得大量不相似的数据哈希在一起，错误率较高。为拉大相似数据冲突的概率与不相似数据冲突的概率之间的差距，需要对哈希函数进行"与构造"。即随机、均匀地选取 k 个哈希函数连接起来形成 k 维的哈希函数组 $G=\{g:S\to U^k\}$，其中 $g(v)=(h_1(v),h_2(v),\cdots,h_k(v))$。这样，$d$ 维空间 R^d 的数据点经过函数 $g(v)$ 的映射处理降至 $k(k\ll d)$ 维，达到降维的目的，同时保证数据间的相似性。设降维后得到空间数据点 $\boldsymbol{u}=(u_1,u_2,\cdots,u_k)$。

哈希函数的选取是随机、独立的，单个哈希表并不能满足查询的需求。因此，为减小这种随机性对数据映射的影响，提高查询率，需要进行哈希函数的"或构造"，即建立多个哈希表，每个哈希表中存在若干个哈希桶。从 G 中随机均匀地选取 L 个函数 $g_1(v),g_2(v),\cdots,g_L(v)$，构造 L 个哈希表，分别对数据点进行映射处理，将数据集存储在 L 个哈希表中的不同桶中，每个哈希表的大小即为数据集的大小。

算法 1 基于 LSH 的索引构建

input：用户-项目评分记录、参数 k(每个哈希表中选取的哈希函数个数)、参数 L(哈希表数目)

output：索引结构(L 个哈希表)

steps：

step1 数据预处理

将用户-项目评分记录转化为用户-项目评分矩阵，具有 m 个用户、n 个项目的评分矩阵 $D_{m\times n}$ 可表示为

$$D_{m\times n}=\begin{bmatrix}r_{1,1}&\cdots&r_{1,n}\\ \vdots&\ddots&\vdots\\ r_{m,1}&\cdots&r_{m,n}\end{bmatrix}$$

Step2 LSH"与构造"和"或构造"

选取 L 个函数 $g_1(\cdot),g_2(\cdot),\cdots,g_L(\cdot)$，其中 $g_i(\cdot)=(h_1^i(\cdot),h_2^i(\cdot),\cdots,h_k^i(\cdot))$，$i=1,\cdots,L$；$h_1(\cdot),h_2(\cdot),\cdots,h_k(\cdot)$ 分别是 LSH 函数族中随机、独立、均匀选取的哈希函数(见式(1))。

Step3 基于 LSH 的索引构建

对于评分矩阵 $D_{m\times n}$ 的每个评分向量 v，利用 LSH 函数 $g_i(v)=(h_1^i(v),h_2^i(v),\cdots,h_k^i(v))$，$i=1,\cdots,L$，进行映射降维，得到 L 个低维向量 $\mathbf{u}=(\mathbf{u}_1,\mathbf{u}_2,\cdots,\mathbf{u}_k)$，从而得到 L 个低维评分矩阵 $D_{m\times k}$，即为每个哈希表中的每个桶的索引向量(其中相同向量表示不同的评分向量被哈希到同一个桶中，这些向量相似的可能性比较大)。

3.3 基于 LSH 的近似近邻用户检索

进行相似性检索时，只需搜索目标用户评分向量 q 所在的 L 个哈希桶，并将所有桶中用户的并集作为目标用户的近似近邻集合。然后，采用基于欧氏距离的相似性度量公式分别度量目标用户与近似近邻集合中各个用户的相似性，并选择满足条件的前 m 个相似性较高的用户，从而产生最近邻用户集合。

算法 2 基于 LSH 的相似性检索

input：目标用户的评分向量

output：目标用户的近似近邻用户评分矩阵

steps：

Step1 目标用户的 LSH 处理

利用算法 1 分别计算目标用户 p 所在 L 个哈希表中的索引向量，即为 L 个哈希桶指示向量。

Step2 近似近邻检索

分别查找目标用户 p 所在的 L 个哈希桶，将 L 个哈希桶中对象的并集作为 p 的近似近邻集合 S，形成目标用户的近似近邻用户评分矩阵。

Step3 相似性度量

对于目标用户的近似近邻集合 S 中的每个向量 q，基于欧氏距离分别度量 q 与 p 之间的相似性，返回前 m 个相似性较高的向量，形成目标用户的最近邻用户评分矩阵。

4 基于改进 LSH 的协同过滤推荐算法

4.1 改进的 LSH

对于 LSH 方法，参数 k、l 值的选取与原数据集的大小有关。但 LSH 存在两个主要问题：一是需要大量的哈希表，每个哈希表的大小正比于原数据集的大小。特别地，当哈希表占用空间超过主存大小时，进行检索就会有大量的磁盘 I/O，引起严重的延迟，降低时间效率。二是在每个哈希表上查询利用率并不高，每次仅限在哈希表中的一个桶内进行检索，即在与检索点哈希值相等的桶内检索，而忽略待检索点的“近邻”桶，没有考虑检索点的邻近桶。

针对上述问题，本文引入一种多探寻的检索方法，对基本的 LSH 算法进行改进。一方面，减少哈希表的使用数量来减少内存的开销；另一方面，增加每个哈希表上的检索范围以减少时间的开销，同时又能保证与原算法近似的检索质量。

多探寻 LSH 的核心思想是在每个哈希表中最近邻点出现概率高的地方进行多次检索，探寻所有可能包含检索点的邻近点的哈希桶。根据局部敏感哈希函数的性质，若一个相似对象原本靠近检索对象，却“不幸”被哈希到其它的桶中，则它以很高的概率被哈希到邻近待检索对象的某个桶中。因此，多探寻 LSH 的目标就是探寻这些邻近的桶来找到尽可能多的邻近点。多探寻 LSH 算法描述如下：

对于每个 d 维数据点 v，哈希函数 $h_{a,b}(v)=\left\lfloor\frac{a\cdot v+b}{w}\right\rfloor$ 中参数 w 在合理情况下较大时，与数据点 v 哈希值相等的概率就会很高。实际上在计算过程中发现，几乎所有的近似近邻点的哈希值要么相等，要么相差 1 或 -1。这样，相似点经函数 $g_i(v)=(h_1^i(v),h_2^i(v),\cdots,h_k^i(v))(i=1,\cdots,L)$ 降维后，得到的 k 维向量 $g_i(v)$ 之间可能存在一个向量差 δ，该向量差的每个分量可能为 0、1 或 -1。然后，通过探寻各个邻近的向量 $g_i(v)+\delta$ 来保证查找到检索点的更多邻近点，同时减小 L 的大小。

4.2 用户评分预测

基于改进 LSH 的近似近邻技术可快速获取目标用户的近邻用户，然后采用一种加权评分方法来预测用户的未评分项目，进而产生推荐。

加权平均的思想是：用户对项目的预测评分可通过用户的各个最近邻用户对项目的加权评分获得预测，加权权重即为用户间的相似性。

设用户 u 的最近邻用户集合为 U，则用户 u 对各个项目 i 的预测评分 P_{ui} 为

$$P_{ui}=\bar{R}_u+\frac{\sum_{v\in U}sim(u,v)\times(R_{vi}-\bar{R}_v)}{\sum_{v\in U}sim(u,v)} \tag{2}$$

其中，$sim(u,v)$ 表示用户 u 与其近邻用户 v 的相似度，R_{vi} 为用户 v 对项目 i 的评分，$\bar{R}_u$、$\bar{R}_v$ 分别表示用户 u 和用户 v 对已评分项目的平均评分。

在采用欧氏距离表达相似度 $sim(u,v)$ 时，本文对其进行改进，转化为如下相似度表示公式：

$$sim(u,v)=\frac{1}{1+dis(u,v)} \tag{3}$$

其中，$dis(u,v)$ 表示用户 u 与用户 v 评分向量的欧氏距离。式(3)将 $[0,\infty)$ 之间的欧氏距离转化为 $(0,1]$ 之间的相似系数，更直接地表达用户 u 和 v 之间的相似性，当欧氏距离 $dis(u,v)=0$ 时，基于欧氏距离的相似系数为 1。

接下来，依据用户未评分项目的评分高低，产生项目推荐列表。

5 实验结果与分析

5.1 实验数据与度量标准

实验数据集采用美国明尼苏达大学 GroupLens 研究组提供的电影评价数据集 MovieLens (http://www.grouplens.org/node/73)。该数据集包括 6040 个用户对 3706 部电影的投票，共 1000000 个评分记录。评分大小分布在 [0,5] 区间内，评分越高，用户对该部电影的兴趣越大。

推荐系统质量的度量标准采用最为常用的统计精度度量方法即平均绝对误差(Mean Absolute Error，MAE)，通过计算预测的用户评分与实际的用户评分之间的偏差来度量预测的准确性。MAE 值越小，推荐质量越高。设预测的用户评分向量为 $\boldsymbol{p}=(p_1,p_2,\cdots,p_t)$，实际的用户评分向量为 $\boldsymbol{r}=(r_1,r_2,\cdots,r_t)$，则平均绝对误差为

$$MAE=\frac{\sum_{i=1}^{t}|p_i-r_i|}{t}$$

5.2 实验设计与结果分析

在数据集上进行实验，首先根据参数 k 对推荐质量 MAE 的影响，训练最佳参数，然后比较不同相似性度量下的推荐质量及运行效率。

(1)参数 k 与 MAE

推荐质量 MAE 的好坏主要依赖于 LSH 算法的关键的参数 k 与 L。考虑到数据集大小及实际运行内存大小，本文选取参数 $L=10$。参数 k 的选择则根据数据集进行训练，在保证最佳推荐质量的情况下选择参数 k。设定用户最近邻数目为 50，并进行 5 次实验取 MAE 均值。实验结果如图 1 所示。

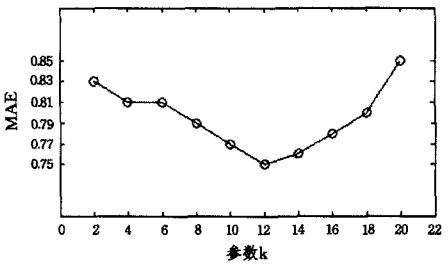

图 1　不同参数对 *MAE* 的影响

实验结果显示，当参数 $k=12$ 时，*MAE* 相对来说最低，为实验最佳参数。即该参数下相似用户查找的查全率和查准率相对较高且平衡，从而保证获得较高的推荐质量。若参数 k 过小，则近邻用户查找中存在大量“假阳”现象；若参数 k 过大，则会造成“假阴”现象。以上两种现象都会严重影响近邻用户选取的精确性，进而影响推荐质量的精确性。

(2)与传统基于用户协同过滤(UBCF)算法的 *MAE* 对比

为测试本文方法的有效性，选取不同近邻用户 n 的数目，比较 n 不同时，本文提出的 LSH-UBCF 与传统 UBCF(此处选取基于用户聚类的协同过滤算法)的 *MAE* 值。实验设定用户最近邻数目为 10～100，参数 $k=12$，$L=10$。

如图 2 所示，当近邻用户较少时，传统 UBCF 方法的 *MAE* 值高于 LSH-UBCF 方法。但随着近邻用户数目增多，LSH-UBCF 方法的性能逐渐高于传统 UBCF 方法，并趋向于稳定，这说明在近邻用户数目大于某种程度时本文提出的方法的有效性。同时不同近邻用户数目的取值会影响推荐性能，因此在实际推荐系统中，应谨慎选取近邻用户数目。

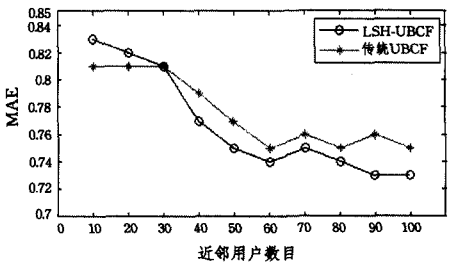

图 2　传统 UBCF 与 LSH-UBCF 的 *MAE* 随近邻用户数目变化的对比

可进一步解释，LSH 这种随机投影方法能在某种程度上偏向于选择具有共同评分的近邻用户，同直接进行相似度计算的传统方法相比，准确性更高。

(3)同传统 UBCF 运行效率的对比

为验证本文方法实际运行的高效性，选取不同近邻用户数目，同传统 UBCF 方法的运行时间进行比较。

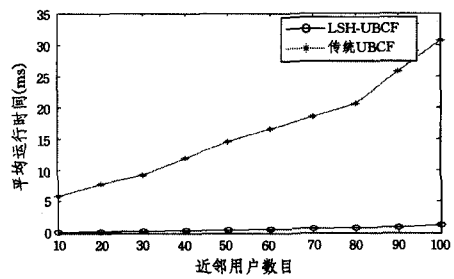

图 3　传统 UBCF 与 LSH-UBCF 的平均运行时间随近邻用户数目变化的对比

如图 3 所示，随着近邻用户数目的增多，传统 UBCF 方法的运行时间急剧增加，而 LSH-UBCF 的运行时间整体相对稳定，且远远少于传统 UBCF 方法的运行时间，其效率高于传统方法 40 多倍。这证明了局部敏感哈希算法具有高效性与稳定性，能够适应于不同规模的数据，并在线性时间内完成相似性检索。

结束语　针对协同过滤中用户评分数据的海量高维、稀疏性对推荐质量的影响，提出了一种基于 LSH 的协同过滤推荐算法。实验表明，本文提出的基于 LSH 的协同过滤算法能很好地应对上述问题，缓解海量高维与稀疏数据对推荐质量以及扩展性能的影响，一定程度上提高了推荐质量与效率。由于该算法性能主要依赖于参数选取，且对参数比较敏感，因此下一步工作是研究如何提高算法对不同规模数据的自适应性；同时，该方法适应于基于项目的协同过滤推荐的性能有待进一步验证。

参 考 文 献

[1] 孟祥武，胡勋，王立才，等. 移动推荐系统及其应用[J]. 软件学报，2013，24(1)：91-108
Meng Xiang-wu, Hu Xun, Wang Li-cai, et al. Mobile Recommender Systems and Their Applications[J]. Journal of Software, 2013, 24(1): 91-108

[2] 罗辛，欧阳元新，熊璋，等. 通过相似度支持度优化基于 K 近邻的协同过滤算法[J]. 计算机学报，2010，33(8)：1437-1445
Luo Xin, Ouyang Yuan-xin, Xiong Zhang, et al. The Effect of Similarity Support in K-nearest-neighborhood Based Collaborative Filtering[J]. Chinese Journal of Computers, 2010, 33(8): 1437-1445

[3] Deng A L, Zhu Y Y, Shi B L. A collaborative filtering recommendation algorithm based on item rating prediction[J]. Journal of Software, 2003, 14(9): 1621-1628

[4] Zheng S, Wang W H, Ford J, et al. Learning from incomplete ratings using non-negative matrix factorization[C]// Ghosh J, ed. Proc. of the 6th SIAM Conf. on Data Mining. Bethesda: SIAM, 2006: 549-553

[5] 韦素云，业宁，朱健，等. 基于项目聚类的全局最近邻的协同过滤算法 [J]. 计算机科学，2012，39(12)：149-152
Wei Su-yun, Ye Ning, Zhu Jian, et al. Collaborative filtering recommendation algorithm based on item clustering and global similarity [J]. Computer Science, 2012, 39(12): 149-152

[6] Anand D, Bharadwaj K K. Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities [J]. Expert Systems with Applications, 2011, 38(5): 5101-5109

[7] Goldberg K, Roeder T, Gupta D, et al. Eigentaste: a constant time collaborative filtering algorithm [J]. Information Retrieval, 2001, 4(2): 133-151

[8] 曾小波，魏祖宽，金在弘. 协同过滤系统的矩阵稀疏性问题的研究[J]. 计算机应用，2010，30(4)，1079-1082
Zeng X B, Wei Z K, Jin Z H. Research of matrix sparsity for collaborative filtering[J]. Journal of Computer Applications, 2010, 30(4): 1079-1082

[9] 方耀宁，郭云飞，丁雪涛，等. 一种基于局部结构的改进奇异值分解推荐算法[J]. 电子与信息学报，2013，35(6)：1284-1289
Fang Y N, Guo Y F, Ding X T, et al. An improved singular value

decomposition recommender algorithm based on local structures [J]. Journal of Electronic & Information Technology, 2013, 35 (6): 1284-1289

[10] Cai Rui, Zhang Chao, Zhang Lei, et al. Scalable Music Recommendation by Search [C] // International Multimedia Conference. 2007: 1065-1074

[11] Andoni A, Indyk P. Nearest-optimal hashing algorithms for approximate nearest neighbor in high dimensions[J]. Communications of the ACM, 2008, 51(1): 117-122

[12] 高毫林，彭天强，李弼程，等. 近似最近邻搜索算法——位置敏感哈希[J]. 信息工程大学学报，2013，14(3)：332-340
Gao H L, Peng T Q, Li B C, et al. Approximate nearest neighbor searching algorithm—locality sensitive hashing[J]. Journal of Information Engineering University, 2013, 14(3): 332-340

[13] Slaney M, Casey M. Locality-sensitive Hashing for Finding Nearest Neighbors [J]. IEEE Signal Processing Magazine, 2008, 8 (3): 128-131

[14] Indyk P, Motwani R. Approximate nearest neighbors: towards removing the curse of dimensionality[C] // The Symposium on Theory of Computing. 1998: 604-613

[15] Datar M, Indyk P, Iimmorlica N, et al. Locality-Sensitive Hashing scheme Based on p-stable Distributions [C]// Proceedings of the Twentieth Annual Symposium on Computational Geometry. 2004: 253-262

（上接第 234 页）

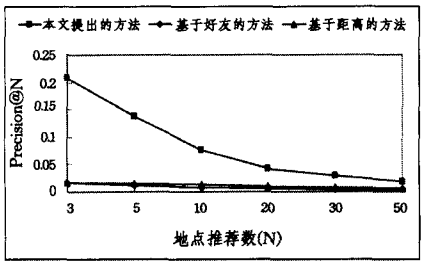

图 4　3 种地点推荐方法的查准率

结束语　本文根据用户在基于位置的社交网络上的签到行为及社交圈的组成，通过综合考虑用户的个人偏好、地点对用户的影响及其好友的推荐 3 个因素，来为用户进行个性化的地点推荐。实验结果说明，考虑用户对出行区域的熟悉性以及人的遗忘特点对其好友推荐作用的影响有助于提高地点推荐的效果。但本文提出的方法没有对用户的出行动机进行预测，即识别用户所在区域是否为其社交、购物或旅游观光的场所，以致不能事先对候选地点进行筛选，从而在一定程度上影响了方法的地点推荐效果。

参 考 文 献

[1] Jia-Ching Y, Huan-Sheng C, Kawuu W L, et al. Semantic trajectory-based high utility item recommendation system[J]. Expert Systems with Applications, 2014, 41(10): 4762-4776

[2] Panagiotis S, Antonis K, Yannis M. GeoSocialRec: explaining recommendations in location-based social networks[C] // Proceedings of the 17th East-European Conference on Advances in Databases and Information Systems. Germany: Springer Verlag, 2013: 84-97

[3] Mao Y, Pei-feng Y, Wang-Chien L. Location recommendation for location-based social networks[C] // Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York: Association for Computing Machinery, 2010: 458-461

[4] Mao Y, Pei-feng Y, Wang-Chien L, et al. Exploiting geographical influence for collaborative point-of-interest recommendation[C] // Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: Association for Computing Machinery, 2011: 325-334

[5] Quan Y, Gao C, Zong-yang M, et al. Time-aware point-of-interest recommendation[C]// Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: Association for Computing Machinery, 2013: 363-372

[6] Jia-Ching Y, Eric Hsueh-Chan L, Wen-ning K, et al. Urban point-of-interest recommendation by mining user check-in behaviors[C] // Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2012: 63-70

[7] Nai-Hung C, Chia-Hui C. Evaluation of social, geography, location effects for point-of-interest recommendation[C] // Proceedings of IEEE 13th International Conference on Data Mining Workshops. Los Alamitos: IEEE Computer Society, 2013: 766-772

[8] Jia-Ching Y, Wen-ning K, Vincent S T, et al. Mining user check-in behavior with a random walk for urban point of interest recommendations[J]. ACM Transactions on Intelligent Systems and Technology, 2014, 5(3): 1-26

[9] 任克江. 基于地理信息的检索和用户数据挖掘[D]. 大连：大连理工大学，2013
Ren Ke-jiang. Information retrieval and user data mining based on geographic information [D]. Dalian: Dalian University of Technology, 2013

[10] 潘果，徐雨明. LBSN 中位置信息与网络拓扑相融合的好友预测[J]. 计算机科学，2014，41(9)：115-118
PAN Guo, Xu Yu-ming. Friends predication based on fusion of topology and location in LBSN[J]. Computer Science, 2014, 41 (9): 115-118

[11] Eunjoon C, Seth A M, Jure L. Friendship and mobility: user movement in location-based social networks[C]// Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2011: 1082-1090

[12] 李秀艳. 多生物特征身份识别方法研究[D]. 天津：天津大学，2010
Li Xiu-yan. Research on personal identity recognition method based on multi-biometric[D]. Tianjin: Tianjin University, 2010