基于滚动时间窗的动态协同过滤推荐模型及算法

沈 键 杨煜普
(上海交通大学自动化系系统控制与信息处理教育部重点实验室 上海 200240)

摘 要 为了提高传统的协同过滤推荐系统的性能,首次提出了考虑时序性的基于滚动时间窗的用户-项目-时间三维动态模型,并在此基础上研究了针对该模型的协同过滤推荐算法。该模型算法对不同时间的兴趣评分按时间序列处理,用户兴趣相似度由不同时间段的分量组合而成,提高了算法的时效性;进而推导出了该模型的增量算法,利用增量算法减少了计算相似度的时间复杂度,从而提高了算法的扩展性;最后设计了合理的实验,实验结果表明提出的三维动态模型及算法在命中率性能上优于传统的二维协同过滤推荐模型及算法。

关键词 滚动时间窗,协同过滤,用户-项目-时间三维模型,推荐算法,时间序列,增量算法
中图法分类号 TP311 **文献标识码** A

Dynamic Collaborative Filtering Recommender Model Based on Rolling Time Windows and its Algorithm

SHEN Jian YANG Yu-pu
(Key Laboratory of System Control and Information Processing, Ministry of Education of China, Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract For improving the performance of the traditional collaborative filtering recommender system, a dynamic user-item-time three-dimensional model based on rolling time windows was proposed, which considers the time sequence problem. Then a special collaborative filtering (CF) algorithm was explored to work with the model. The interest scores at different times are regarded differently according to the time sequence and the similarities between users are composed of components at different times, which increases the timeliness of the algorithm. In addition, the similarities can also be calculated quickly by an incremental formula deduced in this paper so as to improve the scalability of the algorithm. At last, some reasonable experiments show that the model and algorithm presented in this paper outperform the traditional 2D collaborative filtering model and algorithm in terms of the hit rate.

Keywords Rolling time windows, Collaborative filtering, User-item-time 3D model, Recommender algorithm, Time sequences, Incremental algorithm

1 引言

随着互联网的迅速发展,电子商务、网上服务、交易等网络业务越来越普及,大量的网上资源被聚集起来形成海量资源库。海量资源库的形成导致了用户想在其中准确找到自己所需的资源如同大海捞针一样困难,因此出现了搜索引擎。然而即使用户通过搜索引擎寻找所需资源,也必然需要花费不少的精力和时间。针对这个问题,人们在不断探索中又发展出了推荐引擎。推荐引擎的出现在一定程度上可以解决上述问题。

推荐引擎根据用户历史的行为计算出用户的兴趣取向,然后主动向用户推荐出可能感兴趣的项目。将推荐引擎技术融入到Web应用中可以帮助用户快速发现感兴趣和高质量的信息,而且这种技术可以为每一个用户提供一套个性化的推荐服务,从而提升用户体验。另一方面,资源提供商可以利用推荐引擎主动地推荐服务功能以提高销售额,促进盈利[2]。目前,在各类推荐引擎中,协同过滤推荐引擎因其创造性的主动推送能力已被广泛地应用在网上购物等电子商务领域。随着网络业务的多样化发展,协同过滤推荐引擎技术将在越来越多的领域发挥其重要的作用,因此协同过滤推荐引擎技术的研究日趋重要。

近年来,国内外在研究协同过滤推荐技术上大多没考虑到用户兴趣的时序性,而时序性又是用户兴趣变化规律的特点之一,因此本文对协同过滤推荐引擎中的时序性问题做了深入研究,提出了一种优化模型及推荐算法,并实验证明了这种模型算法的有效性。

2 传统协同过滤推荐引擎及时序性问题

协同过滤推荐引擎通过分析用户兴趣,在用户群中找到与目标用户兴趣相似的用户,并综合这些相似用户对某一项

到稿日期:2012-04-26 返修日期:2012-08-12 本文受国家高技术研究发展计划(863计划)项目(2011AA040605)资助。
沈 键(1988—),男,硕士,主要研究方向为推荐系统、数据挖掘与机器学习,E-mail:dicisout@126.com;杨煜普(1957—),男,教授,博士生导师,主要研究方向为智能控制、机器学习。

目的评价，形成系统对该目标用户可能感兴趣项目的预测。它将推荐给用户一些与该用户兴趣相似的其他用户感兴趣的项目，这种推荐引擎的最大优点是具有创造性，即推荐结果不仅仅是用户历史的兴趣关注点，还可能是用户并未关注过但可能感兴趣的项目。

协同过滤推荐引擎技术目前主要有 3 种类型：1）基于内存的推荐（Memory-based Recommendation）；2）基于模型的推荐（Model-based Recommendation）；3）混合推荐[8]。其中基于内存的推荐又可以分为基于用户的推荐（User-based Recommendation）和基于项目的推荐（Item-based Recommendation）两类。本文主要针对基于内存的推荐算法进行了相关研究。

以基于用户的推荐为例，传统的协同过滤推荐算法基本原理如图 1 所示，它根据所有用户对项目的兴趣偏好，利用 K 近邻算法（KNN）[1] 挖掘出与目标用户兴趣相似的 K 个“邻居”用户，然后基于这 K 个“邻居”的历史兴趣偏好信息，为目标用户推荐项目。

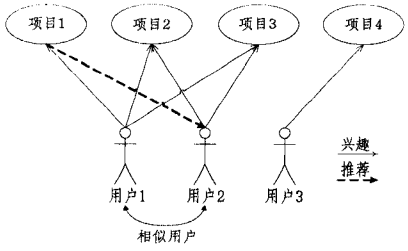

图 1　基于用户的协同过滤推荐原理图

传统的基于内存的协同过滤推荐算法研究大多是基于二维矩阵模型，即将存储在网站中的每个用户信息和网站提供的每个项目建立一个二维矩阵，然后对用户向量组或者项目向量组进行分类，使得类中的成员具有最大的相似度，最后利用同类成员信息向目标用户推荐项目。这种基于二维矩阵模型的协同过滤推荐算法将一个用户不同时间产生的兴趣评分不加以区分地记录在一个向量中，然后以此向量代表该用户进行计算，这种处理方法虽简单但却丢失了很多有用的信息。因为用户兴趣是时变的，用户过去感兴趣的事物可能是现在并不再感兴趣的，而且用户在某个特定的时期所感兴趣的对象一般都不同，因此用户的兴趣具有时序性[7]。一个推荐模型若能很好地挖掘出这种时序性特征，其性能将得到改善。

3　基于滚动时间窗的用户-项目-时间三维动态模型

为了使推荐引擎能够挖掘出用户兴趣的时序特征，从而优化传统的协同过滤推荐引擎性能，本文提出了一种基于滚动时间窗的用户-项目-时间（以下称 UIT）三维动态模型。这种模型是在传统的用户-项目二维模型中加入了时间维，并给时间轴赋予遗忘权值，与当前时刻较近时间段内的用户行为对于最终的推荐结果影响较大，给予较大的权值，而给予与当前时刻较远时间段内的用户行为较小的权值，与当前时刻很远的时间内的评分信息对最终的推荐结果影响很小，可以忽略。另外还需考虑用户的兴趣随时间消退，用户的历史评分呈现出向均值衰减的规律，直到被用户最近时间的评分刷新。

基于上述思想构造出的 UIT 三维模型如图 2 所示，X 轴表示时间，Y 轴表示用户群，Z 轴表示项目群。在时间轴上从原点起依次截取时间窗 T_1-T_s，这些时间窗代表了时间序列，T_1 是离当前时刻最远的时间窗，T_s 是离当前时刻最近的时间窗，每个时间窗权值用 T_i 表示，其中 $T_1<T_2<\cdots<T_s$。过用户轴上某一点并垂直用户轴截取一个封闭平面如图 2 中虚线矩形框所示，它记录了某个用户在 T_1-T_s 时间内对所有项目的评分信息，每个注册用户都有这样一个封闭平面。随着时间向前推移，时间窗向前滚动，记录了用户评分的这些封闭平面同时向前平移，用户的评分信息也随时间窗的推移逐渐向均值衰减，所以基于 UIT 三维模型的协同过滤推荐引擎是动态时序性的，它是以滚动时间窗的方式迭代更新。

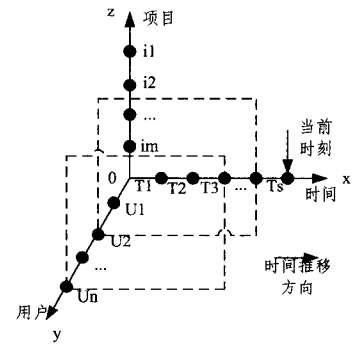

图 2　基于用户-项目-时间三维模型图

基于 UIT 三维模型的协同过滤推荐引擎将用户在不同时间的行为分离，能够更好地察觉到用户兴趣随时间的变化转移，而且随着时间窗的滚动，封闭平面只作平移，据此可以推导出增量算法公式以快速计算相似度。与传统的基于用户-项目二维模型的静态协同过滤推荐引擎相比保留了用户历史兴趣的时序信息，更符合客观现实。

4　基于动态模型的协同过滤推荐算法

4.1　算法核心思想

定义时间窗数为 s，用 $T_1,T_2,\cdots,T_s$ 表示；用户数为 n，用 $u_1,u_2,\cdots,u_n$ 表示；项目数为 m，用 $i_1,i_2,\cdots,i_m$ 表示。

在动态模型中找到 n 个封闭平面，映射成 n 个项目-时间窗二维兴趣度矩阵，通过计算这 n 个二维矩阵两两间的距离确定用户两两间的相似度，然后根据相似度最终可以找到每个用户的最近 K 个邻居，即 K 个与自己相似度最高的用户，最后根据这 K 个用户的共同兴趣向目标用户推荐相应的项目。当时间窗滚动一次后，只需将项目-时间窗二维兴趣度矩阵作列平移并更新最近一列，然后利用增量算法计算滚动后的相似度并获得推荐结果即可。

4.2　算法具体步骤

根据上述推荐算法的核心思想，算法分成以下几步执行：

1. 用户评分数据格式化

用户的兴趣评价信息获取方式是多渠道的，可以从用户对项目的历史评分和用户 Cookies 等要素进行综合判断，得到用户对项目的兴趣评分 r。评分 r 需格式化到一个闭区间，如 $r\in\{1,2,3,4,5\}$，1 表示最不感兴趣，5 表示最感兴趣。

2. 模型数据预处理

由于用户不可能在每个时间窗对每个项目类都留下兴趣评分信息，因此对其中空缺的评分需要进行猜测。一方面，考虑到一般用户对某一项目的兴趣是随时间淡化的，因此把用户上一次相对均值的评分差乘以淡化因子作为空缺评分相对均值的差。另一方面，即使用户对某些项目的兴趣随时间强化，但用户会多次接触这些项目，从而对应的兴趣评分自然会不断被新评分刷新，因此这种情况无需额外处理。最终得到

各个用户的项目-时间窗二维兴趣评分矩阵。

定义 1(淡化因子) 由于用户的兴趣随时间逐渐消退，评分向均值衰减。淡化因子是用户对同一项目在某个时间窗内相对均值的评分差与之前一个时间窗内相对均值的评分差的比值。

定义 2(兴趣评分矩阵) 兴趣评分矩阵记录了用户对项目的评分，如用户 u_i、u_j 的兴趣评分矩阵：

$$A_i=\begin{array}{c} \begin{matrix} T_1 & T_2 & \cdots & T_s \end{matrix} \\ \begin{pmatrix} r_{11}^{(i)} & r_{12}^{(i)} & \cdots & r_{1s}^{(i)} \\ r_{21}^{(i)} & r_{22}^{(i)} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ r_{m1}^{(i)} & \cdots & \cdots & r_{ms}^{(i)} \end{pmatrix}_{m\times s} \end{array} \begin{matrix} i_1 \\ i_2 \\ \cdots \\ i_m \end{matrix}$$

$$A_j=\begin{array}{c} \begin{matrix} T_1 & T_2 & \cdots & T_s \end{matrix} \\ \begin{pmatrix} r_{11}^{(j)} & r_{12}^{(j)} & \cdots & r_{1s}^{(j)} \\ r_{21}^{(j)} & r_{22}^{(j)} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ r_{m1}^{(j)} & \cdots & \cdots & r_{ms}^{(j)} \end{pmatrix}_{m\times s} \end{array} \begin{matrix} i_1 \\ i_2 \\ \cdots \\ i_m \end{matrix}$$

式中，$i_1,i_2,\cdots,i_m$ 表示项目，$T_1,T_2,\cdots,T_s$ 表示时间窗，r_{pk} 表示用户在第 k 个时间窗内对项目 p 的评分。当时间窗滚动一次后，矩阵的 T_1 列删除，其余各列向左平移一列，T_s 列是用户在新时间窗内的兴趣评分，从而得到新的评分矩阵。

3. 计算两个用户之间的兴趣相似度

用户 u_i 和用户 u_j 的兴趣相似度体现在两个兴趣评分矩阵的距离上，可以通过分别计算两个矩阵对应列向量的距离间接求得矩阵之间的距离。因为构成两个用户的兴趣相似度分量与各自的时间窗有着密切联系，所以在计算相似度时需考虑时间窗权值。因此得到计算当前时刻用户 u_i 和用户 u_j 之间的兴趣相似度 $\mathrm{SIM}_{i,j}(t)$公式如下：

$$\mathrm{SIM}_{i,j}(t)=\frac{\sum_{k=1}^{s}T_k\cdot \mathrm{sim}_{i,j}^{(T_k)}}{\sum_{k=1}^{s}T_k} \tag{1}$$

式中，$\mathrm{sim}_{i,j}^{(T_k)}=\dfrac{\sum_{\{p|r_{pk}^{(i)},r_{pk}^{(j)}\neq 0\}} r_{pk}^{(i)}r_{pk}^{(j)}}{\sqrt{\sum_{\{p|r_{pk}^{(i)},r_{pk}^{(j)}\neq 0\}}(r_{pk}^{(i)})^2}\cdot\sqrt{\sum_{\{p|r_{pk}^{(i)},r_{pk}^{(j)}\neq 0\}}(r_{pk}^{(j)})^2}}$，表示用户 u_i 和用户 u_j 在第 k 个时间窗内对具有共同评分的所有项目的兴趣相似度分量；$T_1,T_2,\cdots,T_s$ 是时间窗权值，其具体取值可参考 Ebbinghaus 遗忘曲线函数[3](见图 4)，即 $T_i=\mathrm{e}^{-\lambda(1-\frac{i}{s})}$，其中遗忘系数 λ 值根据实际情况而定，评分的淡化因子取值为遗忘曲线上的 T_{s-1}。经实验验证，这种矩阵相似度计算方法优于矩阵 Frobenius 范数法[9]。

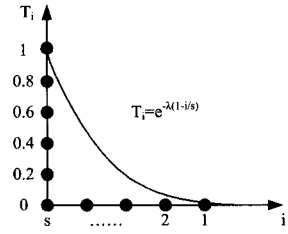

图 3 Ebbinghaus 遗忘曲线函数

4. 兴趣相似度的增量算法

考虑到算法的扩展性，我们对用户相似度的计算进行简化。利用上述求兴趣相似度的时序规律推导出以下增量公式：

$$\begin{aligned}\mathrm{SIM}_{i,j}(t+T)&=\frac{\sum_{k=1}^{s-1}T_k\cdot \mathrm{sim}_{i,j}^{(T_{k+1})}+T_s\cdot \mathrm{sim}_{i,j}^{*}}{\sum_{k=1}^{s}T_k}\\&=\mathrm{e}^{-\frac{\lambda}{s}}\mathrm{SIM}_{i,j}(t)+\frac{1}{\sum_{k=1}^{s}T_k}(\mathrm{sim}_{i,j}^{*}-\mathrm{e}^{-\lambda}\mathrm{sim}_{i,j}^{(T_1)})\\&\approx\mathrm{e}^{-\frac{\lambda}{s}}\mathrm{SIM}_{i,j}(t)+\frac{1}{\sum_{k=1}^{s}T_k}\mathrm{sim}_{i,j}^{*}\end{aligned} \tag{2}$$

式中，$sim_{i,j}^{*}$表示下个时间窗内用户之间的兴趣相似度分量。根据式(2)，随着时间窗的滚动，用户之间的兴趣相似度只需通过增量公式方便快速求得而无需再用式(1)计算 $\mathrm{SIM}_{i,j}(t+T)$。

5. 推荐决策

按照第 4 步的算法求出用户两两之间的兴趣相似度，最终可以得到用户群的相似度矩阵。

定义 3(相似度矩阵) 相似度矩阵是一个记录了所有用户之间兴趣相似度的对称方阵，表示为：

$$S=\begin{array}{c} \begin{matrix} u_1 & u_2 & \cdots & u_n \end{matrix} \\ \begin{pmatrix} 0 & \mathrm{SIM}_{12} & \cdots & \mathrm{SIM}_{1n} \\ \mathrm{SIM}_{12} & 0 & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \mathrm{SIM}_{1n} & \cdots & \cdots & 0 \end{pmatrix}_{n\times n} \end{array} \begin{matrix} u_1 \\ u_2 \\ \cdots \\ u_n \end{matrix}$$

根据得到的兴趣相似度矩阵采取推荐策略如下：

a)当目标用户 $u_x\in\{u_1,u_2,\cdots,u_n\}$时，直接在矩阵 S 的 u_x 行中找到最大的 K 个兴趣相似度，其对应的列元素即为目标用户 u_x 的 K 个最近邻居用户集。

b)当目标用户 $u_x\notin\{u_1,u_2,\cdots,u_n\}$时，推荐系统遇到用户冷启动问题[6]。一种解决方法是按照上述算法分别求出兴趣相似度 $\mathrm{SIM}_{x1},\mathrm{SIM}_{x2},\cdots,\mathrm{SIM}_{m}$，然后从中找到最大的 K 个值，所对应的用户作为目标用户 u_x 的最近邻居用户集。也可以采用其他方法(如基于内容或基于项目的协同过滤推荐方法[4,5])解决用户冷启动问题。

找到了目标用户的 K 个最近邻居用户集后，利用式(3)求得 K 个邻居用户对所有项目的共同兴趣度向量，然后再从中找出共同兴趣度最高且目标用户还未发掘的 N 个项目作为最终的项目推荐给目标用户。

$$P_x=\frac{\sum_{i=1}^{K}\mathrm{SIM}_{xi}A_i^{(s)}}{\sum_{i=1}^{K}\mathrm{SIM}_{xi}}=\frac{\sum_{i=1}^{K}\mathrm{SIM}_{xi}\begin{pmatrix} w_{1s}^{(i)} \\ w_{2s}^{(i)} \\ \cdots \\ w_{ms}^{(i)} \end{pmatrix}}{\sum_{i=1}^{K}\mathrm{SIM}_{xi}} \tag{3}$$

4.3 算法流程图

根据上述核心思想和步骤，算法的流程图如图 4 所示。

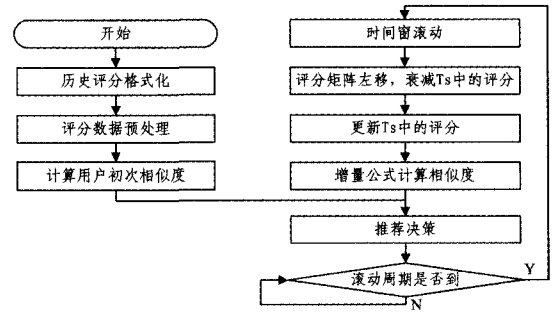

图 4 三维动态协同过滤推荐算法流程图

5 实验与分析

5.1 算法评价标准

为了验证模型及算法的有效性，本文用一个时序性特征明显的应用实例——电影行业进行测试分析，利用 Movielens 电影评分网上采集的真实数据集[11]测试了本文提出的模型及算法。实验数据集共用了 56770 条记录，包括了 500 个用户对 1682 部电影的评分信息，时间跨度是 8 个月。实验采用命中率作为评价标准，即如果用户在下一个时间窗中观看的电影在当前推荐电影列表中存在则记为当前时刻对该用户命中，如果用户在下一个时间窗中无任何行为则不考虑其命中情况。综合各个用户的命中情况，求出该时间窗下的用户命中率。

5.2 算法性能实验

实验中，时间窗数设为 10，时间窗滚动周期为 2 周（推荐列表 2 周更新一次），用户的邻居数 $K=50$，用户的推荐电影列表数 $N=10$，以每次实验的下一个时间窗用户的评分作为测试数据，每隔一个时间窗进行 1 次实验，共测试了 9 次，得到最终的测试结果，如图 5 所示。图中横坐标表示每次实验的时间，纵坐标表示用户命中率。当取合适的遗忘系数时 3 维模型推荐算法在命中率性能上优于传统的二维模型，对 9 次实验结果进行加权平均得到二维模型平均命中率只有 33.4%，三维模型为 40.4%，即平均每次推荐命中率提高了 7%。

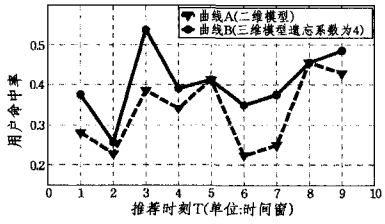

图 5 三种推荐情形的用户命中率图

5.3 算法参数实验

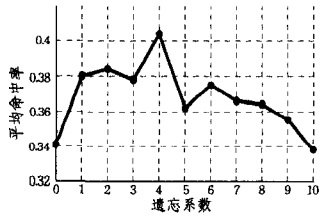

图 6 不同遗忘系数下的算法平均命中率图

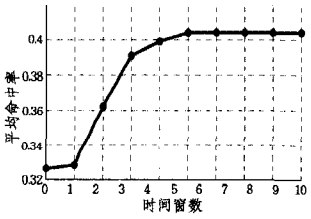

图 7 不同时间窗数下的算法平均命中率图（遗忘系数 $\lambda=4$）

在本文提出的模型及算法中，遗忘系数和时间窗数两个参数大小的选取至关重要。因此，通过实验研究了这两个关键参数对推荐结果的影响。遗忘系数反映了用户群的整体兴趣变化速度，在图 6 中，平均命中率曲线呈两端低中间高的形状，即当本实验数据集上遗忘系数在[1,4]之间平均命中率较高，遗忘系数过小或者过大都会导致算法性能降低，因此需根据实际用户群的兴趣变化速度选取合理的遗忘系数。由于用户的行为具有一定的随机性，图中曲线上某些点存在跳变现象，但不影响此规律。图 7 说明了在淡化因子不变的前提下时间窗数越大，平均命中率越高，但是随着时间窗数增加，命中率会达到饱和，因此考虑到空间复杂度，时间窗数无需太大，选择适宜即可。综上分析，本文提出的推荐模型及算法挖掘了用户兴趣的时序信息，与传统的二维模型算法相比具有更高的命中率。针对不同的应用场景，通过选取合理的遗忘系数 λ 值和时间窗数可以得到较准确、稳定的推荐结果。

结束语 基于滚动时间窗的动态协同推荐模型及其算法充分挖掘出了用户兴趣的时序信息，实现了对用户兴趣变化的动态追踪。该推荐算法以时间序列为主线，将用户的兴趣评分按时间序列进行处理，用户的相似度由不同时间的相似度分量构成，能够有效区分用户不同时间的兴趣并察觉用户兴趣的转移。计算用户相似度利用了增量算法，降低了计算时间复杂度，提高了动态协同过滤推荐算法的可扩展性。实验结果表明，协同过滤推荐算法中考虑时间序列因素可以有效提高推荐性能，与传统的二维推荐模型算法相比，基于滚动时间窗的 UIT 三维动态协同过滤推荐模型及算法在推荐命中率性能上得到了提高，不失为一种优化方法。

虽然 UIT 三维推荐模型算法改善了推荐性能，但是这种算法在数据稀疏性问题[10]上还有改进的空间，所以下一步的研究重点是如何在该三维动态模型中解决数据稀疏性问题，从而提高用户相似度的可靠性，最终进一步改善推荐性能。

参 考 文 献

[1] Liu Bing. Web Data Mining[M]. Springer, 2006

[2] 康雨洁. 基于协同过滤的个性化社区推荐方法研究[D]. 合肥：中国科学技术大学，2011

[3] 李可潮. 基于访问时间和评分时间的协同过滤算法研究[D]. 南宁：广西大学，2010

[4] 邢春晓. 高凤荣. 战思南，等. 适应用户兴趣变化的协同过滤推荐算法[J]. 计算机研究与发展，2007，44(2)：296-301

[5] Strunjas S. Algorithms and models for collaborative filtering from large information corpora[D]. Cincinnati: University of Cincinnati, 2008

[6] Leung W K C. Enriching user and item profiles for collaborative filtering: from concept hierarchies to user-generated reviews [D]. Hong Kong: The Hong Kong Polytechnic university, 2008

[7] 项亮. 动态推荐系统关键技术研究[D]. 北京：中国科学院研究生院，2011

[8] Achary R R. An author recommender system using both content-based and collaborative filtering methods[D]. Northridge: California State University, 2011

[9] France S. Distance metrics for high dimensional nearest neighborhood recovery[J]. Compression and normalization, Elsevier Science, 2011, 184(1): 92-100

[10] Pan Wei-ke, Xiang E W, Liu N N, et al. Transfer Learning in Collaborative Filtering for Sparsity Reduction [C]//Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence. 2010: 230-235

[11] Grouplens. Movielens data set [OL]. http://www.grouplens.org/node/73#attachments, 2011