

传统 Item-Based 协同过滤推荐算法改进

李 雪¹ 左万利^{1,2} 赫枫龄¹ 王 英¹

¹(吉林大学计算机科学与技术学院 长春 130012)

²(吉林大学符号计算与知识工程教育部重点实验室 长春 130012)
(jlujislixue@gmail.com)

An Improved Item-Based Collaborative Filtering Recommendation Algorithm

Li Xue¹, Zuo Wanli^{1,2}, He Fengling¹, and Wang Ying¹

¹(College of Computer Science and Technology, Jilin University, Changchun 130012)

²(Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012)

Abstract Conventional item-based collaborative filtering algorithm selects neighbors by the similarity between items. However, existing similarity measures suffer from their drawbacks respectively, thus cannot bring about accurate and reliable neighbors. A new method is proposed to select k -nearest neighbors by combining item similarity and neighbor grade based on the number of users that co-rate both items. In addition, for new items lacking rating information, two measures, aggregate neighborhood and common ratings, are presented. Experiments on the MovieLens data set show that the new algorithmic elements can help to increase the accuracy of recommendations.

Key words item-based; collaborative filtering; recommender systems; k -nearest neighbors; neighbor grade

摘 要 传统 item-based 协同过滤算法根据项目之间的相似性来选取最近邻居。然而,现存的几种相似性度量方法都存在相应的弊端,因此只根据相似性无法找到准确可靠的最近邻。根据对两项目共同评分的用户个数,建立项目近邻等级,提出了结合项目近邻等级与相似性求取最近邻的新方法。另外,对于系统中新加入的项目,因为其上评分信息的匮乏,求得的最近邻往往是不准确的。为此,提出了聚合最近邻和“集体评分”两种改进方法。在 MovieLens 数据集上的实验结果表明,将上述改进应用于传统 item-based 协同过滤算法,推荐质量有明显的提高。

关键词 item-based; 协同过滤; 推荐系统; k 最近邻; 项目近邻等级

中图法分类号 TP391

随着因特网信息的激增,如何在海量信息中快速准确地筛选出我们感兴趣的知识已经成为人们关注的焦点。推荐系统便应运而生,并且正在电子商务领域发挥越来越大的作用。

目前,已有许多适用于推荐系统的技术,如数据挖掘^[1]、内容过滤、协同过滤^[2-3]、综合使用内容过滤和协同过滤^[4]等等。其中最成功的是基于协同过滤

的推荐思想。

传统的 user-based 协同过滤推荐算法根据当前在线用户显式或隐式评分的历史记录,利用某个相似性度量方法,找出与该用户最相似的 k 个用户,称为“ k 最近邻”。针对某个特定的、当前用户尚未评分或浏览的项目,利用邻居们对该项目的感兴趣程度来预测当前用户对它的评分,或是直接将邻居们最

收稿日期:2009-06-19

基金项目:国家自然科学基金项目(60373099);吉林省科技发展计划基金项目(20070533);国家教育部高等学校博士学科点专项科研基金项目(200801830021);吉林大学基本科研业务费交叉学科与创新基金项目(200810025)

感兴趣的 N 个项目推荐给当前用户。

然而,协同过滤技术也存在几个潜在的局限性:

1) 可扩展性^[5]; 2) 准确率^[5]; 3) 新项目问题^[4]; 4) 同义词问题^[6]. 数据的稀疏性是制约准确率的主要因素。

1 相关工作

GroupLens 研究系统首先为 Usenet 新闻和电影提供了一个匿名的协同过滤解决方案^[7]. Herlocker 给出了一个 user-based 协同过滤算法框架,并提出了 k 最近邻的思想^[3]. Sarwar 提出了 item-based 协同过滤算法^[5], Amazon. com 采用的便是这种方法^[8]. Lemire 提出的 Slope One 算法也可以作为一种新的协同过滤参考模型^[9].

目前,研究人员已经给出了多种策略来缓解数据稀疏性,如填充法^[2,10-11];将基于内存的算法与基于模型的算法相结合^[12-14];利用多种可用信息^[15];基于信任因子的方法^[16]等等。

在大量用户(潜在邻居)中寻找当前用户的 k 最近邻是传统 user-based 协同过滤算法的“瓶颈”^[3]. 文献[17]通过矩阵奇异值分解来减少项目空间的维数,以提高效率,但降维的效果难以保证^[18]. item-based 协同过滤算法^[5]利用项目之间的相似性,对于给定的一个(用户,项目)对,根据用户对当前项目“最近邻”的喜好,来判断用户对该项目的喜好. 因为项目之间的相似性相对稳定^[5],所以求项目间的相似性以及最近邻的过程都可以离线执行,大大提高了系统的可扩展性. 当系统内用户的数量远大于项目的数量时, item-based 算法就更能体现出这一优越性。

本文基于传统的 item-based 协同过滤算法,针对求取最近邻和评分预测两个步骤,分别给出了有效的改进方法,并指出了每种改进适用的情况。

2 传统的 item-based 协同过滤推荐算法

推荐系统中,数据的核心是一个用户-项目评分矩阵 $A(m, n)$, 它包含 m 个用户的集合 $U = \{u_1, u_2, \dots, u_m\}$ 和 n 个项目的集合 $I = \{i_1, i_2, \dots, i_n\}$. 元素 R_{ui} 表示用户 u 对项目 i 的评分. 若用户 u 未对项目 i 评分,则 $R_{ui} = 0$.

2.1 项目的相似性度量方法

定义 1. 对于 $\forall i \in I$, 定义项目-评分矩阵 $A(m, n)$ 中对应于 i 的列为项目 i 的评分向量,记为 I_i .

定义 2. 对于 $\forall u \in U$, 定义项目-评分矩阵 $A(m, n)$ 中对应于 u 的行为用户 u 的评分向量,记为 I_u .

基于项目 i 和项目 j 的评分向量,通常用以下 3 种方法来度量 i 和 j 的相似性^[5]:

1) 标准的余弦相似性.

$$\text{sim}(i, j) = \cos(i, j) = \frac{i \cdot j}{\|i\|_2 * \|j\|_2} = \frac{\sum_{u=1}^m R_{ui} * R_{uj}}{\sqrt{\sum_{u=1}^m (R_{ui})^2} * \sqrt{\sum_{u=1}^m (R_{uj})^2}}. \quad (1)$$

2) 修正的余弦相似性. 不同用户存在不同评分尺度的偏见,下面将采取减去对应项目上所有用户的平均评分的方法,来刻画用户对某一项目的评分与“公众意见”的偏差.

$$\text{sim}(i, j) = \frac{\sum_{u=1}^m (R_{ui} - \bar{R}_i)(R_{uj} - \bar{R}_j)}{\sqrt{\sum_{u=1}^m (R_{ui} - \bar{R}_i)^2} \sqrt{\sum_{u=1}^m (R_{uj} - \bar{R}_j)^2}}, \quad (2)$$

其中, \bar{R}_i 和 \bar{R}_j 分别表示项目 i 和 j 上所有用户打分的平均值。

3) 相关相似性. 根据 Pearson 提出的相关系数来度量项目之间的相似性. 定义对项目 i 和 j 都有评分的用户集合为 U_{ij} .

$$\text{sim}(i, j) = \frac{\sum_{u \in U_{ij}} (R_{ui} - \bar{R}_i)(R_{uj} - \bar{R}_j)}{\sqrt{\sum_{u \in U_{ij}} (R_{ui} - \bar{R}_i)^2} \sqrt{\sum_{u \in U_{ij}} (R_{uj} - \bar{R}_j)^2}}. \quad (3)$$

选择合适的相似性度量方法,求出项目集 I 中任意两项目之间的相似性,存入项目相似性矩阵 Sim 中. 显然, Sim 是对称矩阵。

2.2 最近邻的选取

1) 中心最近邻. 它是当前最常用也是最成功的最近邻选取方法^[6]. 对于 $\forall i \in I$, 根据相似性矩阵 Sim 中其他项目与 i 的相似性,按从大到小顺序排列,将相应的项目编号存入最近邻矩阵 $TkNN$ 的相应行中,构成项目 i 的最近邻集合;即第一最近邻与 i 的相似性最高,第二最近邻次之,依此类推。

2) 聚合最近邻. 对于某些新加入的项目,其评分向量与其他项目的评分向量的交集很小,不利于准确地计算相似性. 比如说,已知当前项目 i 的第一最近邻为 j , 项目 k 和 i 被 1 个用户共同评分,项目 t 和 i 没有被共同评过. 这样,在 i 的最近邻列表

中, k 排在 t 前. 但此时, 若 t 是 j 的第一最近邻, 那么 t 也很有可能是 i 的最近邻. 聚合最近邻的思想就是基于上述考虑.

设当前项目为 i , 欲求 i 的 N 个最近邻, 当前最近邻个数为 $J (J < N)$, 当前最近邻集合记为 NN . 则使得

$$\text{sim}(i, k) + \sum_{j \in NN} \text{sim}(j, k), \quad (4)$$

最大的项目 k 为第 $J+1$ 个邻居. 显然, 随着 J 的不断增大, 求和运算的计算量逐渐加大. 所以可以设置一个“信任上限”值 δ , 只信任前 δ 个最近邻的推荐, 取与当前项目 i 和 i 的前 δ 个邻居最相似的项目 k 为第 $J+1$ 个邻居. 另设一个参数 σ 来划分数据“极端”稀疏的界限. 用 N_i 表示对 i 评过分的用户个数. 当 $N_i < \sigma$ 时, 采用聚合最近邻; 否则, 采用中心最近邻.

2.3 产生推荐

基于项目 i 的 k 个最近邻居, 有以下两种方法可以预测用户 a 对其未评分或浏览项目的评分 $P_{a,i}$:

1) 加权平均值

$$P_{a,i} = \frac{\sum_{j \in kNN_i} \text{sim}(i, j) R_{aj}}{\sum_{j \in kNN_i} |\text{sim}(i, j)|}, \quad (5)$$

其中, kNN_i 表示项目 i 的 k 最近邻集合.

2) 基于项目均值的加权平均值

$$P_{a,i} = \bar{R}_i + \frac{\sum_{j \in kNN_i} \text{sim}(i, j) (R_{aj} - \bar{R}_j)}{\sum_{j \in kNN_i} |\text{sim}(i, j)|}, \quad (6)$$

其中 \bar{R}_i 和 \bar{R}_j 分别表示项目 i 和 j 上所有用户打分的平均值.

文献[3]指出了式(6)相对于式(5)的优越性. 所以本文将采用式(6)进行评分预测.

3 算法改进

3.1 求取最近邻过程的改进——结合使用项目相似性与项目近邻等级

求取最近邻是整个协同过滤算法最关键的一步. 最近邻越准确, 其推荐结果就越可靠. 因为最近邻根据项目相似性矩阵 Sim 来选取, 所以我们有必要首先分析一下 3 种相似性度量方法存在的弊端.

设两个项目 i 和 j 评分向量的交集为 $U_{ij} = U_i \cap U_j$, 并集为 $U'_{ij} = U_i \cup U_j$. 用 $||$ 表示取集合元素的

个数. 令 $N_{ij} = |U_{ij}|$, $N'_{ij} = |U'_{ij}|$.

在标准余弦相似性中, 根据式(1), $N'_{ij} - N_{ij}$ 的值越小, $\text{sim}(i, j)$ 就有越大的趋势 (当然, 由向量中具体的元素值决定). 假设有如下 3 个项目, i, j, k , 其中 $|U_i| = 50$, $|U_j| = 100$, $|U_k| = 200$, $N_{ij} = 10$, $N_{ik} = 20$. 显然, 在 i 的最近邻列表中, j 排在 k 前, 因为 k 过长的评分向量使得 $\text{sim}(i, k)$ 的分母值大于 $\text{sim}(i, j)$ 的分母值, 并且掩盖了 $N_{ik} > N_{ij}$ 的优势. 然而事实上, 对项目 i 来说, k 极有可能是比 j 更可靠的推荐者. 修正的余弦相似性也存在同样的问题.

在相关相似性中, 因为我们只考虑两个项目评分向量的交集, 所以经常会出现与当前项目基于很少的共同评分而排在前面的邻居, 即“小交集”最近邻问题. 这样的邻居往往导致不可靠的预测.

针对上述两点, 基于相似性矩阵 Sim 和两项目评分向量的交集元素个数, 提出了如下求取最近邻的方法: 设当前项目为 i . 对于 $\forall j \in I$ 且 $j \neq i$, $N_{ij} = |U_{ij}|$, 表示对项目 i 和项目 j 共同评分的用户个数; $N_i = |U_i|$, 表示对项目 i 评分的用户个数.

首先, 要根据 N_{ij} 将 I 中的项目相对于 i 分成 n 个等级, 即项目 i 的近邻等级. 若用 G_{ij} 表示项目 j 相对于当前项目 i 的等级, 则有

$$G_{ij} = \left\lceil \frac{N_{ij}}{N_i} \times n \right\rceil, \quad (7)$$

其中 $\lceil \cdot \rceil$ 表示四舍五入. 对于固定的当前项目 i , N_i 是常数, n 是需在实验中调节的参数. 则对于固定的项目 i 和参数 n , G_{ij} 的值与 N_{ij} 成正比. 也就是说, 与当前项目 i 拥有的共同评分越多, 项目 j 的等级越高, 也就越具有推荐能力. 而 n 则反映了一个等级细化的程度. 比如, 设 $N_i = 100$, $N_{ij} = 9$, $N_{ik} = 6$. 当 $n = 10$ 时, j 和 k 都属于等级 1; 当 $n = 20$ 时, j 和 k 则分别属于等级 2 和等级 1. 因为推荐系统数据的稀疏性, 通常 N_{ij} 不会达到 N_i 的一半, 所以当 $n = 1$ 时, 便相当于所有项目都属于同一级别, 项目等级不会对求取最近邻产生影响. 当 $n = N_i$ 时, $G_{ij} = N_{ij}$, N_{ij} 便是项目 j 的级别.

然后, 将 I 中除 i 以外的项目, 按照其等级主排序, 与 i 的相似性次排序, 并将项目编号存入最近邻矩阵 $TkNN$ 的相应行中. 即对于 i 的最近邻列表中任意两相邻项目 j 和 k (j 在 k 前), 有:

1) $G_{ij} > G_{ik}$; 或者

2) $G_{ij} = G_{ik}$, 且 $\text{sim}(i, j) \geq \text{sim}(i, k)$.

最后, 将上述过程应用于 I 中的每一个项目, 最终生成完整的最近邻矩阵 $TkNN$. 显然, $TkNN$ 不是

对称矩阵。整个计算过程可离线执行。

另外,文献[3]对相关相似性的“小交集”最近邻问题给出了一种改进方法:重要性加权(significance weighting)。设置一个调节参数 sw ,文献[3]中 $sw=50$ 。

$$sim(i,j)=\begin{cases} corr(i,j), N_{ij} \geq sw, \\ corr(i,j) \times \frac{N_{ij}}{sw}, N_{ij} < sw. \end{cases} \quad (8)$$

然后直接按照相似性排序产生最近邻集合。

在后面的实验中,分别将 $sw=50$ 及使推荐结果最优的 sw 应用于相关相似性,并与项目近邻等级方法进行比较。

3.2 产生推荐过程的改进——集体评分

第 2.3 节给出的两种方法都是基于 k 最近邻的推荐过程。但是,对于一个新加入的项目,其上的评分非常少,难以找到准确的最近邻,从而会降低推荐的准确率。第 2.2 节给出的聚合最近邻是缓解这个问题的一个思路。现在,给出另一种简单易行的方法。

设当前用户为 a ,欲评分项目为 i ,则有如下公式:

$$P_{a,i} = R_a + \frac{\sum_{u \in U_i} (R_u - R_u)}{N_i}, \quad (9)$$

其中, $N_i = |U_i|$, 表示对项目 i 评过分的用户个数; \bar{R}_a 和 \bar{R}_u 分别表示用户 a 和用户 u 对所有项目的平均评分。因为综合利用了当前用户和所有与当前项目有关的用户的评分信息,所以这种方法可以称为“集体评分”(common ratings)。设置一个调节参数 σ , 作为数据“极端”稀疏的界限。则当 $N_i < \sigma$ 时,应用式(9);否则,应用式(6)。

4 实验结果及分析

4.1 数据集

本实验采用 MovieLens 数据集(<http://www.grouplens.org/node/73>)。该版本包含 943 个用户在 1682 部电影上的 100000 条评分记录,其中,每个用户至少对 20 部电影进行了评分。评分值采用 5 分制。实验采用 5 折交叉验证。该数据集的稀疏等级为 0.9369^[6]。

4.2 度量标准

本文采用平均绝对偏差(MAE)作为评价推荐系统推荐质量的度量标准,MAE 定义^[6]为

$$MAE = \frac{\sum_{i=1}^N |p_i - q_i|}{N}, \quad (10)$$

N 为测试集大小, p_i 为预测评分, q_i 为实际评分。MAE 越小,推荐质量越高。实验中,对 5 折交叉验证中 5 次实验得到的 MAE 取均值。

4.3 实验及分析

以下实验的图例中,用 c_cosine 代表修正余弦。

4.3.1 三种相似性度量方法的比较

本实验使用的是传统的基于 k 最近邻的 item-based 协同过滤推荐算法, k 的值从 10 递增到 200,目的是找出传统算法的最好结果,以便和改进结果进行比较。

从图 1 可以看出,标准余弦是 3 种相似性方法中表现最好的。 $sw=150$ 是 significance weighting 的最好结果,它与标准余弦的结果在 $k=40$ 后几乎重合。 sw 的调节过程不在此赘述。

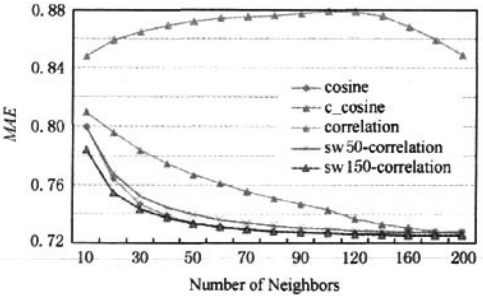


图 1 相似性度量方法比较

4.3.2 中心最近邻和聚合最近邻的比较

信任上限 δ 和极端稀疏界限 σ 的调节过程不在此赘述,只选择一个最好的结果与只采用中心最近邻的方法进行比较, $\delta=5, \sigma=20$ 。此处,采用标准余弦来度量项目之间的相似性。

从图 2 可以看出,聚合最近邻方法并没有起到积极的作用。这很可能是因为当前项目的前 δ 个最近邻中,某些邻居上的评分记录也是“极端”稀疏的,进而根据这样的邻居去求取当前项目的最近邻是更不可靠的。

4.3.3 传统的只利用相似性求取最近邻的方法与

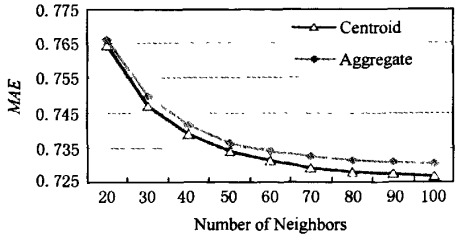


图 2 中心最近邻和聚合最近邻的比较

结合使用项目近邻等级信息的改进方法的比较

实验中,分别令 $n=2,3,5,N_i$, 其中 N_i 表示对项目 i 评过分的用户个数. 图例中,将 n 值置于相似性方法名字的后面. 传统算法直接按相似性排序产生最近邻,所有用户属于同一级别,近似于 $n=1$.

从图 3~5 可以看出, $n=2$ 取得了最小的 MAE;随着 n 的增大,MAE 的值也随着增大. 从图 3 可以看出,改进方法对标准余弦几乎没有起到作用,只有 $n=2$,最近邻个数 $k \leq 60$ 时,推荐质量略有提高. 由图 5 可以看出,使用 $n=2$ 的项目近邻等级与使用 significance weighting($sw=50$)取得了相近甚至略好的结果. 图 4 中 $n=2$ 与修正余弦的结合,获得了图 3~5 中最低的 MAE. 图 6 将这个结果与已有的两个最好的结果进行了比较,可以看出,结合 $n=2$ 的项目近邻等级和修正余弦,取得了明显的改进.

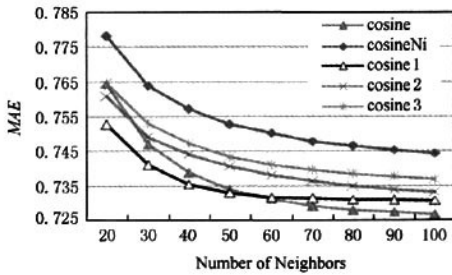


图 3 将不同项目近邻等级与 cosine 联合使用

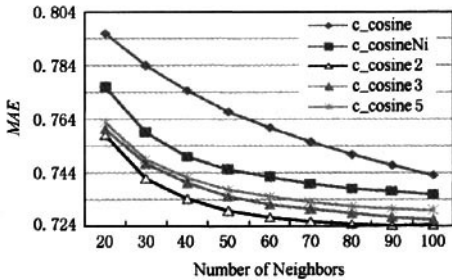


图 4 将不同项目近邻等级与修正余弦联合使用

关于标准余弦没有得到期望的改进的原因,仍需进一步的研究. 通过与另两种相似性方法的对比,改进不成功的原因可能是标准余弦中没有考虑不同用户评分尺度的差异.

4.3.4 传统算法与结合使用项目近邻等级和“集体评分”的改进方法的比较

最后,将式(9)应用于产生推荐的过程. 极端稀疏界限 σ 的调节过程不在此赘述. $\sigma=20$ 时,效果最好.

从图 7 的 cosine 和 20σ -cosine 可以看出,“集体

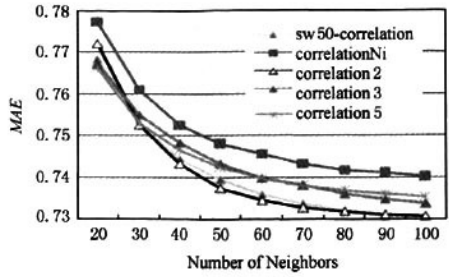


图 5 将不同项目近邻等级与相关相似性联合使用

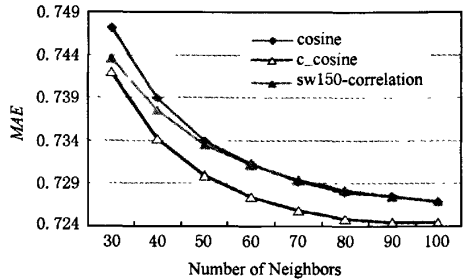


图 6 效果最好的 3 种方法的比较

评分”对评分预测过程起到了改进作用. 当系统中新项目较多时,这种改进将更加明显. 结合了 $n=2$ 的项目近邻等级,修正余弦以及“集体评分”的方法 (20σ -c_cosine2) 与传统算法的最好结果 (cosine) 相比,有明显的改进.

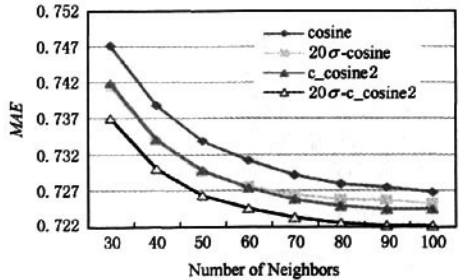


图 7 最终结果比较

5 结 论

针对传统 item-based 协同过滤算法只利用项目间的相似性求取最近邻的弊端,提出了结合项目近邻等级与相似性的新方法. 对于系统中新加入的项目,提出了“集体评分”的改进方法. 实验证明,上述方法对推荐准确率的提高都起到了积极的作用. 聚合最近邻是一种新的思路,仍需进一步的研究. 需要注意的是,对于不同的数据集,3 种相似性方法的效果不一,故而使用前要通过实验来选择.

参 考 文 献

- [1] 王实, 高文, 李锦涛. 基于分类方法的 Web 站点实时个性化推荐. 计算机学报, 2002, 25(8): 845-852
- [2] Breese J, Hecherman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering //Proc of the 14th Conf on Uncertainty in Artificial Intelligence (UAI'98). San Francisco: Morgan Kaufmann, 1998: 43-52
- [3] Herlocker J L, Konstan J A, Borchers A, et al. An algorithmic framework for performing collaborative filtering //Proc of the 22nd Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 1999: 230-237
- [4] Melville P, Mooney R J, Nagarajan R. Content-boosted collaborative filtering for improved recommendations //Proc of the 18th National Conf on Artificial Intelligence. 2002: 187-192
- [5] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms //Proc of the 10th Int World Wide Web Conf. New York: ACM, 2001: 285-295
- [6] Sarwar B, Karypis G, Konstan J, et al. Analysis of recommendation algorithms for E-commerce //Proc of the 2nd ACM Conf on Electronic Commerce. New York: ACM, 2001: 158-167
- [7] Resnick P, Iacovou N, Suchak M, et al. Grouplens: An open architecture for collaborative filtering of netnews //Proc of the ACM CSCW'94 Conf on Computer-Supported Cooperative Work. New York: ACM, 1994: 175-186
- [8] Linden G, Smith B, York J. Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 2003, 7(1): 76-80
- [9] Lemire D, MacLachlan A. Slope one predictors for online rating-based collaborative filtering //Proc of SIAM Data Mining (SDM'05). 2005
- [10] 邓爱林, 朱扬勇, 施伯乐. 基于项目评分预测的协同过滤推荐算法. 软件学报, 2003, 14(9): 1621-1628
- [11] 张锋, 常会友. 使用 BP 神经网络缓解协同过滤推荐算法的稀疏性问题. 计算机研究与发展, 2006, 43(4): 667-672
- [12] 高滢, 齐红, 刘杰, 等. 和用户等级的协同过滤推荐算法. 计算机研究与发展, 2008, 45(9): 1463-1469
- [13] 陈健, 印鉴. 基于影响集的协作过滤推荐算法. 软件学报, 2007, 18(7): 1685-1694
- [14] 张光卫, 李德毅, 李鹏, 等. 基于云模型的协同过滤推荐算法. 软件学报, 2007, 18(10): 2403-2411
- [15] 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. 计算机研究与发展, 2007, 44(2): 296-301
- [16] O'Donovan J, Smyth B. Trust in recommender systems //Proc of UII'05. New York: ACM, 2005: 167-174
- [17] 赵亮, 胡乃静, 张守志. 个性化推荐算法设计. 计算机研究与发展, 2002, 39(8): 986-991
- [18] Aggarwal C C. On the effects of dimensionality reduction on high dimensional similarity search //Proc of the 20th ACM SIGMOD-SIGACT-SIGART Symp on Principles of Database Systems. New York: ACM, 2001: 256-266

李 雪 女, 1985 年生, 硕士研究生, 主要研究方向为数据挖掘与 Web 挖掘。

左万利 男, 1957 年生, 教授, 博士生导师, 发表论文 40 余篇, 被三大检索机构收录 8 次。主要研究方向为数据库理论、机器学习、数据挖掘与 Web 挖掘、网络搜索引擎。

赫枫龄 男, 1962 年生, 教授, 主要研究方向为 Web 挖掘与网络搜索引擎。

王 英 女, 1981 年生, 博士研究生, 主要研究方向为 Web 挖掘、网络搜索引擎、本体。

作者: 李雪, 左万利, 赫枫龄, 王英, Li Xue, Zuo Wanli, He Fengling, Wang Ying
作者单位: 李雪, 赫枫龄, 王英, Li Xue, He Fengling, Wang Ying(吉林大学计算机科学与技术学院, 长春, 130012), 左万利, Zuo Wanli(吉林大学计算机科学与技术学院, 长春, 130012; 吉林大学符号计算与知识工程教育部重点实验室, 长春, 130012)
刊名: 计算机研究与发展 **ISTIC EI PKU**
英文刊名: JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT
年, 卷(期): 2009, 46(z2)

参考文献(18条)

1. 陈健; 印鉴 基于影响集的协作过滤推荐算法[期刊论文]-软件学报 2007(07)
2. 高滢; 齐红; 刘杰 和用户等级的协同过滤推荐算法[期刊论文]-计算机研究与发展 2008(09)
3. 张锋; 常会友 使用BP神经网络缓解协同过滤推荐算法的稀疏性问题[期刊论文]-计算机研究与发展 2006(04)
4. Lemire D; Maclachlan A Slope one predictors for online rating-based collaborative filtering 2005
5. Linden G; Smith B; York J Amazon.com recommendations: Item-to-item collaborative filtering[外文期刊] 2003(01)
6. Resnick P; Iacovou N; Suchak M Grouplens: An open architecture for collaborative filtering of netnews 1994
7. Sarwar B; Karypis G; Konstan J Analysis of recommendation algorithms for E-commerce 2001
8. Sarwar B; Karypis G; Konstan J Item-based collaborative filtering recommendation algorithms 2001
9. Melville P; Mooney R J; Nagarajan R Content-boosted collaborative filtering for improved recommendations 2002
10. Herlocker J L; Konstan J A; Borchers A An algorithmic framework for performing collaborative filtering 1999
11. Breese J; Hecherman D; Kadie C Empirical analysis of predictive algorithms for collaborative filtering 1998
12. Aggarwal C C On the effects of dimensionality reduction on high dimensional similarity search 2001
13. 赵亮; 胡乃静; 张守志 个性化推荐算法设计[期刊论文]-计算机研究与发展 2002(08)
14. O' Donovan J; Smyth B Trust in recommender systems 2005
15. 邢春晓; 高凤荣; 战思南 适应用户兴趣变化的协同过滤推荐算法[期刊论文]-计算机研究与发展 2007(02)
16. 张光卫; 李德毅; 李鹏 基于云模型的协同过滤推荐算法[期刊论文]-软件学报 2007(10)
17. 邓爱林; 朱扬勇; 施伯乐 基于项目评分预测的协同过滤推荐算法[期刊论文]-软件学报 2003(09)
18. 王实; 高文; 李锦涛 基于分类方法的Web站点实时个性化推荐[期刊论文]-计算机学报 2002(08)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_jsjyjfz2009z2123.aspx