第 33 卷 第 4 期 计 算 机 工 程 2007 年 2 月
Vol.33 No.4 Computer Engineering February 2007
·软件技术与数据库· 文章编号：1000—3428(2007)04—0042—03 文献标识码：A 中图分类号：TP311

基于用户潜在偏好的协同过滤

陈晓红[1,2]，沈 洁[1]，顾天竺[1]，吴 颜[1]，张 舒[1]，李 慧[1]
(1. 扬州大学信息工程学院，扬州 225009；2. 南通大学计算机科学与技术学院，南通 226001)

摘 要：提出了一种新的协同过滤模型，解决了不同用户在项目上，有相似的偏好、不同的评分习惯的问题。该模型可有效地改进传统协同过滤模型相似性度量方法，提高了用户相似性度量准确性。实验结果表明，该模型在个性化推荐系统应用中取得了较好的效果。
关键词：协同过滤；相似性；评价模型；偏好模型

Collaborative Filtering Based on Users' Underlying Preference Model

CHEN Xiaohong[1,2], SHEN Jie[1], GU Tianzhu[1], WU Yan[1], ZHANG Shu[1], LI Hui[1]
(1. College of Information Engineering, Yangzhou University, Yangzhou 225009;
2. College of Computer Science and Technology, Nantong University, Nantong 226001)

【Abstract】 This paper describes a new model for collaborative filtering, it can be used to slove the problem that two users with similar preferences on items may have different rating schemes. This model may effectively improve the traditional collaborative filtering method used to compute the similarity between users, and enhances the accuracy of user similarity measurement. Experiment results show that the new model performs well in personalized recommendation system.
【Key words】 Collaborative filtering; Similiarity; Rating model; Preference model

1 概述

近年来随着 Internet 技术的迅猛发展，WWW 的资源不断增加，要找到所需要的信息变得越来越困难。由此产生了信息过滤技术，它的任务是帮助用户去除不相关的信息，找到有用的信息。本文主要讨论协同过滤，它是推荐系统中最为成功的技术之一。它的基本思想是：要为一个用户找到他真正感兴趣的内容，首先要找到与此用户有相似偏好的用户群，然后以他们的兴趣，为目标用户产生推荐。

多年来，研究人员已经提出了多种协同过滤推荐算法[1~4]。根据文献[1]所描述，协同过滤算法可以分为两类：基于用户的算法和基于项目的算法。基于用户的算法首先要找到与当前测试集用户评分模型最相似的训练集，然后联合这些相似用户的评分来预测测试集用户的评分，产生对应的推荐列表。这类算法主要是计算用户相似度，包括相关相似性、余弦相似性、修正的余弦相似性等。基于项目的算法首先根据用户评分模型把训练集数据库中的不同用户分为几类，然后把测试集用户归到一个已有的用户类中，最后用该类对某个特定项目的评分作为对测试集用户的预测评分。这类算法包括贝叶斯算法等。

随着协同过滤技术的发展，现在有 2 个问题需要解决：(1)大部分用户仅评分了整个数据库中的一小部分项目，这样 2 个用户都评价过的项目更少，会导致相似度准确性的下降。已经有一些方法来解决这个问题，比如，在相关相似度方法中，未评分的项目引入缺省的评分。(2)有些用户对项目有相似的偏好类型，但他们的评分却不同。实际上，在之前的研究过程中，可以发现相关相似性的推荐质量比余弦相似性的好，这两种方法的主要区别是，在相关相似性方法中计算相似度时，要在原始评分中减去每个用户的评分平均值。而在余弦相似性方法中，直接使用原始评分。由此得出，不同用户有不同的评分习惯，评分信息不能直接代表偏好信息。应该做一个转换，将用户的评分信息转换成他们的潜在偏好信息。本文提出了一个新的相似度计算方法，计算用户偏好相似度，直接应用用户潜在偏好信息，提高了推荐的质量。

2 新的协同过滤模型

传统的基于用户的协同过滤模型存在的一个主要问题是不同用户的评分习惯不一致，用户的评分不能直接代表他对项目的真正偏好。协同过滤的推荐系统是根据最近邻居对项目的评分来预测目标用户对项目的评分，最近邻居的选择是否准确，直接关系到整个推荐系统的推荐质量。而选择目标用户的最近邻居，是要计算用户相似性，传统的模型都是基于用户表面的评分。因为用户会有不同的评分习惯，两个有相同偏好的用户可能有不相似的表面评分，这就要求推荐系统获取用户的真正偏好信息。本文提出一种基于用户的概率协同过滤模型，使用用户的潜在偏好信息。把用户分为训练集和测试集两部分进行实验，用贝叶斯分类方法结合训练集预测出测试集用户对一个新项目的偏好。

在这个模型中，为了预测测试集用户的偏好，有 3 个步骤：(1)所有用户都使用“转换模型”计算，包括训练集和测试集中的用户。转换模型就是通过表面观察到的评分得到用户偏好某项目的可能性。(2)使用用户的偏好信息，计算训练

基金项目：江苏省自然科学基金资助项目(Bk2005046)
作者简介：陈晓红(1981－)，女，硕士生，主研方向：Web 信息检索，数据挖掘；沈 洁，教授；顾天竺、吴 颜、张 舒、李 慧，硕士生
收稿日期：2006-02-20 **E-mail**：chen8102@ntu.edu.cn

集和测试集用户的偏好相似度。(3)通过联合训练集用户的偏好及偏好相似度，计算得到测试集用户对未评分项目的偏好。

2.1 计算用户潜在偏好

这是本模型的一个关键步骤，把用户的表面评分转换成他们的潜在偏好。下面首先给出理论描述，然后给出一个简单的例子直观地比较用户的原始评分和本文提出的方法计算出的用户的偏好度。

2.1.1 理论描述

令 x 和 y 代表一个项目和一个用户，$R_y(x)$ 代表用户 y 对项目 x 的评分，$P_y(x)$ 代表用户 y 对项目 x 的偏好可能性。

$R_y(x)$ 满足正态分布，即

$N(\mu(R_y(x),y),\sigma(R_y(x),y))$

注意到均值 $\mu(R_y(x),y)$ 和方差 $\sigma(R_y(x),y)$ 都不仅与评分相关也与用户相关，用户偏好某项目的概率 $P_y(x)$，可以通过 $R_y(x)$ 的概率密度分布得出。

设用户评分了 I 个项目，计算用户对某个特定项目 x_i ($i \in I$)的偏好概率，具体计算步骤如下：

Step1 计算用户 y 的评分均值：

$$\mu(R_y(x),y)=\frac{\sum_{i\in I}R_y(x_i)}{n} \tag{1}$$

Step2 计算方差：

$$\sigma(R_y(x),y)=\sqrt{\frac{\sum_{i\in I}(R(x_i)-\mu(R_y(x),y))^2}{n}} \tag{2}$$

Step3 对特定的项目 x_i，令

$$R_y^{'}(x_i)=\frac{R_y(x_i)-\mu(R_y(x),y)}{\sigma(R_y(x),y)} \tag{3}$$

这样可使 $R_y^{'}(x_i)$ 满足标准正态分布，即

$R_y^{'}(x_i)\sim N(0,1)$

Step4 通过 $R_y^{'}(x_i)$ 的概率密度分布，得到用户 y 对特定的项目 x_i 的偏好概率(即偏好度)。

$$P_y(x_i)=\Phi(R_y^{'}(x_i))=\int_{-\infty}^{R_y^{'}(x_i)}\frac{1}{\sqrt{2\pi}}e^{-t^2/2}dt \tag{4}$$

2.1.2 直观的例子

有 2 个用户(“ A ”和“ B ”)对 10 个项目有着类似的偏好。假设用户 A 一向比用户 B 评分宽松，如表 1 所示。

表 1 用户 A 和用户 B 的评分信息

Item	1	2	3	4	5	6	7	8	9	10
Rating(A)	1	2	3	3	3	4	4	4	5	5
Rating(B)	1	1	2	2	2	3	3	3	4	5

如果用传统的协同过滤方法，只依赖于用户的表面评分来计算这 2 个用户之间相似度，会发现他们相似度不是很高。这就会使之后的推荐质量下降。

用 2.1.1 节中描述的计算方法，可得到用户 A，用户 B 对这 10 个项目的偏好度，如表 2 所示。

表 2 用户 A 和用户 B 的偏好信息

item	1	2	3	4	5	6	7	8	9	10
Pref(A)	0.023	0.122	0.369	0.369	0.369	0.691	0.691	0.691	0.909	0.909
Pref(B)	0.091	0.091	0.309	0.309	0.309	0.631	0.631	0.631	0.878	0.977

很明显，表 2 中两个用户对项目的偏好比表 1 相似得多，更能反映实际情况。

2.2 偏好相似性度量方法

有两种概念的用户的相似度，评分相似度和偏好相似度(Preference Similar, PS)。传统的相似性度量方法都是计算评分相似度。为了准确地预测用户的偏好，可以用 2.1 节求得的用户偏好度，计算用户的偏好相似度。

用 w^{p}_{y,y_1} 代表用户y和用户y_1的偏好相似度。设X(y)为用户y评分过的项目集，而用户的偏好模式定义为

$$\mathrm{P(y)}=\{\,P_y(x)\mid\forall x\in X(y)\,\}=\{\,\Phi(R_y^{'}(x))\mid\forall x\in X(y)\,\} \tag{5}$$

其中，$\Phi(R_y^{'}(x))$ 为用户对项目的偏好度。则用户y和用户y_1的偏好相似度，即

$$W^{p}_{y,y_1}=p(y_1\mid P(y))=\frac{p(P(y)\mid y_1)p(y_1)}{p(P(y))} \tag{6}$$

其中，用户对每个项目的评分独立于其他项目，则 $p(P(y)\mid y_1)$ 可被描述为

$$p(P(y)\mid y_1)=\prod_{x\in X(y)}P_{y_1}(x)^{P_y(x)}=\prod_{x\in X(y)}\Phi(R_{y_1}^{'}(x))^{\Phi(R_y^{'}(x))} \tag{7}$$

在上述讨论中，如果有些项目用户y评价过但用户y_1没有评价过，这样，就没法得到式(7)要求的用户y_1对x的评分信息 $R_{y_1}^{'}(x)$。本文做了启发式处理，定义默认的用户偏好度为用户对所有项目偏好的平均概率。

2.3 产生推荐

通过上面的用户偏好相似度计算得到目标用户的最近邻居，下一步需要产生推荐。对一个测试集用户y_0，可用先计算出他的偏好模型，然后计算出y_0与其他用户y的偏好相似度 w^{p}_{y,y_0}，最后用户y_0对没评分的项目 x 的偏好可能性 $P_{y_0}(x)$ 可表示为

$$P_{y_0}(x)=\overline{P_{y_0}}+\frac{\sum_{y\in Y}w^{p}_{y_0,y}(P_y(x)-\overline{P_y})}{\sum_{y\in Y}w^{p}_{y_0,y}} \tag{8}$$

3 实验及其结果分析

以传统的协同过滤模型中相似度计算方法做参照，并在不同情况下验证本文提出偏好相似性的有效性。

3.1 传统的相似性度量方法

(1)相关相似性 (Pearson Correlation Coefficient, PCC)：用户之间的相似性通过 Pearson 相关系数度量，即

$$W_{y_1,y}=\frac{\sum_{x\in X(y_1)\cap x\in X(y)}(R_y(x)-\overline{R_y})(R_{y_1}(x)-\overline{R_{y_1}})}{\sqrt{\sum_{x\in X(y_1)\cap x\in X(y)}(R_y(x)-\overline{R_y})^2}\sqrt{\sum_{x\in X(y_1)\cap x\in X(y)}(R_{y_1}(x)-\overline{R_{y_1}})^2}}$$

(2)余弦相似性 (Cosine Similarity, CS)：用户之间的相似性通过向量间的余弦夹角度量，即

$$W_{y_1,y}=\frac{\sum_{x\in X(y_1)\cap x\in X(y)}R_y(x)R_{y_1}(x)}{\sqrt{\sum_{x\in X(y_1)\cap x\in X(y)}R_y(x)^2}\sqrt{\sum_{x\in X(y_1)\cap x\in X(y)}R_{y_1}(x)^2}}$$

3.2 数据库描述

采用 MovieLens 站点提供的数据集(http://movielens.umn.edu)，该站点是一个基于 Web 的研究性推荐系统。目前该站点用户已超过 45 000 人，评价过的电影已超过 6 600 部。

从用户评分数据库中选择 6 000 条评分数据作为实验数据集，实验数据集中共包含 152 个用户和 832 部电影，其中每个用户至少对 20 部电影进行了评分。

整个实验数据集划分为训练集和测试集，通过变化训练集用户个数，变化测试集用户评价过的项目来检验本文提出

模型在不同情况下的有效性。分别选取 50 个用户和 100 个用户作为训练集用户，其余用户则作为测试集用户。已知测试集用户评价过的项目分别选取为 5 个、10 个、20 个，来预测测试集用户对其他项目的评分，并与实际评分相比较得到评价准确度。

3.3 度量标准

评价推荐系统推荐质量的度量标准主要包括统计精度度量方法和决策支持精度度量方法两类。统计精度度量方法中的平均绝对偏差(Mean Absolute Error, MAE)可以直观地对推荐质量进行度量，是最常用的一种度量方法，本文采用 MAE 作为度量标准。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度量预测的准确性，MAE 越小，推荐质量越高。设预测的用户评分集合表示为 $\{R_1^{'}, R_2^{'}, ..., R_n^{'}\}$，对应的实际用户评分集合 $\{R_1^{'}, R_2^{'}, ..., R_n^{'}\}$，则平均绝对偏差 MAE 定义为

$$MAE = \frac{\sum_{i=1}^{n} | R_i^{'} - R_i |}{n}$$

3.4 实验结果

实验采用 MAE 作为度量标准，比较本文提出的模型和 2 种传统的方法，结果如表 3 所示。

表 3 不同方法的 MAE 性能

	Method	5 Items given	10 Items given	20 Items given
50 个训练集用户	相关相似性 (PCC)	0.851	0.843	0.821
	余弦相似性(VS)	0.861	0.845	0.831
	偏好相似性(PS)	0.846	0.835	0.826
100 个训练集用户	相关相似性 (PCC)	0.870	0.838	0.814
	余弦相似性(VS)	0.872	0.913	0.843
	偏好相似性(PS)	0.840	0.832	0.817

从表 3 中，可以发现在已知测试集用户评价项目较多时，相关相似性(PCC)表现较好。而用户评价项目较少时，本文的新模型(PS)优势比较明显，可以取得较好的推荐质量。

在传统的协同过滤推荐算法中，2 个用户之间的相似性由用户评价过的项目的表面评分决定，使得最近邻居的选择不够准确，推荐质量下降。

本文提出的方法深入挖掘了用户的潜在信息，最近邻居的选择是基于用户的偏好相似度的，计算结果更具有实际意义和准确性。

4 小结

本文提出了一种新的协同过滤模型。这种模型有效地解决了协同过滤中的一个关键问题，用户评价习惯不一致，偏好相似表面评分却不相似。传统的方法，用户相似度都是通过表面评分直接计算，本文的方法首先将用户对某项目的表面评分转换成用户潜在偏好某项目的可能性，然后基于偏好模型计算用户的相似度。本文的预测都是基于用户潜在偏好作出的，实验结果表明，本文提出的方法有较好的推荐质量。

参考文献

1 Breese J S, Heckerman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering[C]//Proceedins of the 14th Conference on Uncertainty in Artificial Intelligence. 1998.

2 Soboroff I M, Nichoal C. Collaborative Filtering and the Generalized Vector Space Model[C]//Proceedings of the 23rd Annual International Conference on Researech and Development in Information Retrieval. 2000.

3 Si L, Jin R. Flexible Mixture Model for Collaborative Filtering[C]//Proc. of the 20th International Conference on Machine Learning. 2003.

4 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[J]. 软件学报, 2003, 14(9): 1621-1628.

(上接第 41 页)

误，应在 delete 某个指向动态内存的指针后立即将其置为空。

4 利用 Windows 结构化异常处理机制处理发布版本软件的内存崩溃

在程序的发布阶段，应尽量减少程序错误尤其是内存崩溃。如果崩溃了，应该“优雅”地退出，尽量收集程序崩溃时的运行信息以帮助程序供应商后续的调试。要捕捉内存非法访问并获知非法访问的指令地址、寄存器内容等信息，需要用到Windows的结构化异常处理（Structured Exception Handling，SEH）机制[6]。MiniDumpWriteDump是dbghelp.dll提供的一个 API函数(参考MSDN)，用于转储用户模式程序的一些信息(比如堆栈情况等)并存为一个文件(比如.dmp文件)，此文件可以被微软的调试器(VC++或者WinDBG)利用进行事后调试。使用此函数需要dbghelp.h、dbghelp.lib和dbghelp.dll(这些文件可以在Windows Platform SDK中找到)。

要事后根据.dmp 文件调试代码，需要为发布版本软件产生 debug symbols (pdb)文件（打开编译器/DEBUG 选项）。在拿到.dmp 文件以后，用 VC++打开.dmp 文件，然后调试执行（按 F5 键）。这样，崩溃现场就会重现。文献[5]基于上述的方法实现了崩溃报告系统。

5 结论

实践证明，在上述方法和工具支持下的减少软件内存缺陷的方法和工具，可以有效防止和查找代码中的内存错误和内存泄漏，并且能和开发人员日常编码无缝结合，执行起来非常高效。上述方法配合单元测试、代码评审、每日构建、Bug 追踪等措施，形成了一个高效的质量保证流程，在我们的大型平台软件开发过程中起到了重要作用。

参考文献

1 Sutter H, Alexandrescu A. C++ Coding Standards: 101 Rules, Guidelines, and Best Practices[M]. Addison-Wesley Professional, 2004-10.

2 Stroustrup B. The Design and Evolution of C++[M]. Addison-Wesley Professional, 1994-03.

3 Karlsson B. Beyond the C++ Standard Library: An Introduction to Boost[M]. Addison-Wesley Professional, 2005-08.

4 Zyzyck J. A Report Generator for PC-Lint[J]. Dr. Dobb's Journal, 2003, 28(2): 52.

5 Dietrich H. XCrash Report: Exception Handling and Crash Reporting[Z]. 2003-10. http://www.codeproject.com/debug/ XCrash ReportPt4.asp.

6 Richter J M. Programming Applications for Microsoft Windows[M]. Microsoft Press, 1999-09.