华南理工大学

South China University of Technology

《数据仓库与数据挖掘》

--推荐系统协同过滤实现

院（系）： 软件学院

专　业： 10 软件四班

学生姓名： 庄灿杰 201030633279

蔡锐涛 201030630032（组长）

戴颖毅 201030630261

陶升奇 201030631855

指导教师 ： 蔡毅

提交日期： 2012/11/21

一．问题分析

数据来源：MovieLens

数据特性：

1.Ratings are made on a 5-star scale (whole-star ratings only).
2.Each user has at least 20 ratings.
3.100,000 ratings (1-5) from 943 users on 1682 movies.
4.Training set/test set = 80,000/20,000
5.Trainnig ratio (0.1-0.9).
6.Each record contain user Id、 item Id、a rating that user comment on the item、 and a time stamp.

问题描述：

根据所提供的 MovieLens 的数据集合，使用数据仓库与数据挖掘课程上所学习的知识，设计一个推荐系统。

质量指标：

1. MAE (Mean Average Error)

$$MAE = \frac{\sum_{m} \left| r_{i,j}^{calc} - r_{i,j}^{user} \right|}{m}$$

2. RMSE (Root Mean Square Error)

$$RMSE = \sqrt{\frac{\sum_{m} (r_{i,j}^{calc} - r_{i,j}^{user})^2}{m}}$$

3. TIME (Time costs)

【Where m is the number of user-movie pairs in the test set
$r_{i,j}^{calc}$ is the rating obtained by calculation of using our method
$r_{i,j}^{user}$ is the rating given by user in the test set】

二，分析设计

推荐系统根据系统中已有的用户信息（Training Set），利用信息过滤技术来预测特定的用户对特定的商品的喜好，或者向特定的用户推荐最感兴趣的商品。常用的推荐方法有基于内容的推荐(CB)、协同过滤推荐(CF)以及组合推荐

(Hybrid)等，其中协同过滤推荐是广为应用的个性化推荐技术之一，协同过滤推荐算法基本思想是通过计算目标用户与各个基本用户对项目评分之间的相似性(User-based) ,搜索目标用户的最近邻居(KNN),然后由最近邻居的评分数据向目标用户产生推荐 ,即目标用户对未评分项目的评分可以通过最近邻居对该项目评分的加权平均值进行逼近 ,从而产生推荐。

本次实验中采用的是协同过滤推荐，主要原因：源数据的特殊性，仅有(uID、iID、rating、timeStamp)，数据记录相对简单，数据中没提供资源标签，所以使用基于内容(CB)的推荐,并不合适。

实验中我们通过修改了协同过滤推荐算法中基于用户里的 KNN 算法的相关设置，增设阈值，减少了一定的计算量，并做了相应的测试，和详细的记录。

三，算法流程

1. 算法流程如下图所示

2. 在 KNN 算法中，目标用户 u 和邻居用户 v 之间的相似性是通过 Peason 相关系数来度量的。其计算公式为

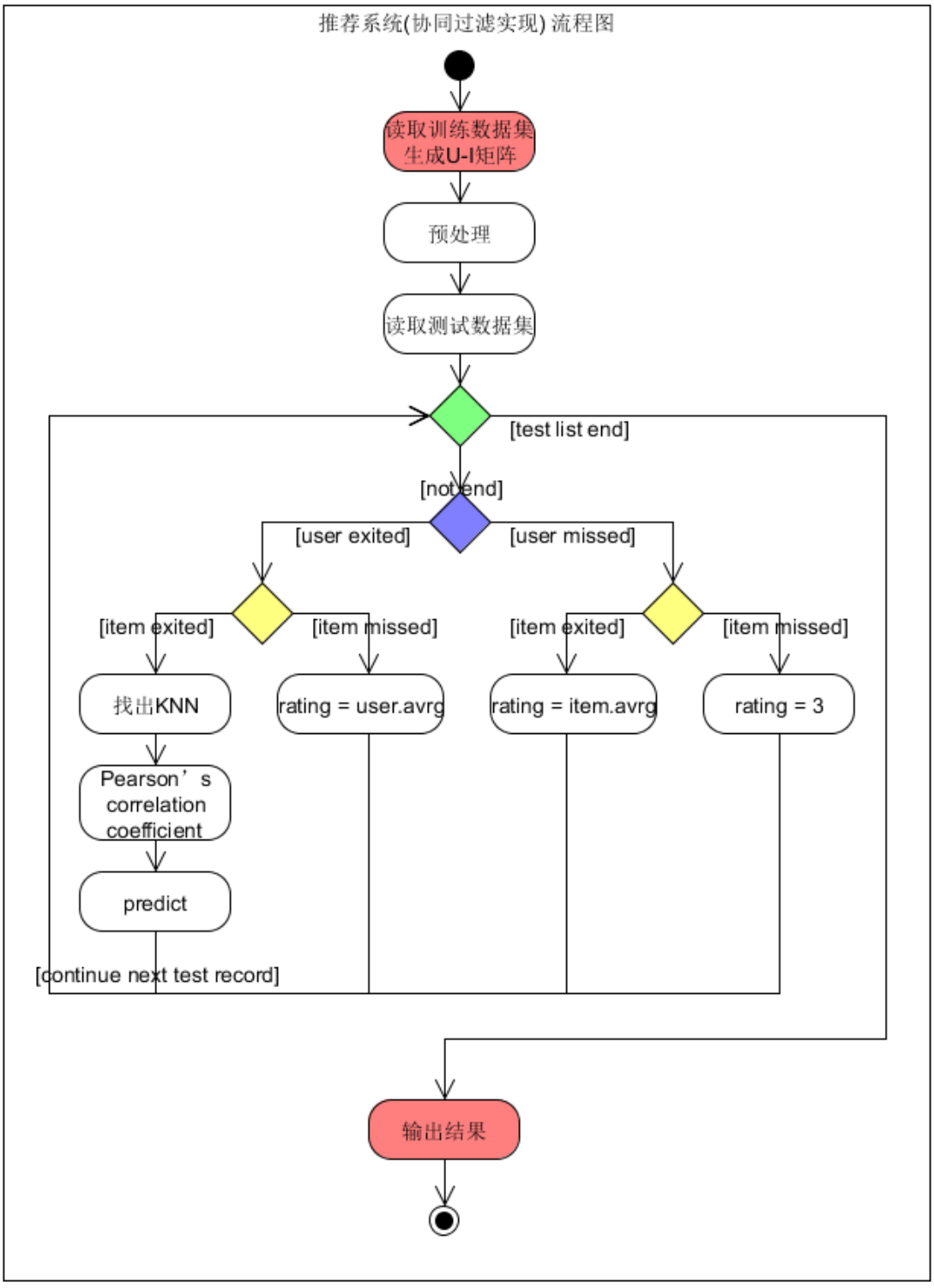
$$sim(\mathbf{u},\mathbf{v})=\frac{\sum_{i\in C}(r_{\mathbf{u},i}-\bar{r}_{\mathbf{u}})(r_{\mathbf{v},i}-\bar{r}_{\mathbf{v}})}{\sqrt{\sum_{i\in C}(r_{\mathbf{u},i}-\bar{r}_{\mathbf{u}})^2}\sqrt{\sum_{i\in C}(r_{\mathbf{v},i}-\bar{r}_{\mathbf{v}})^2}},$$

$r_{u,i}$ 表示用户 u 对物品 i 的评分，r_u 和 r_v 分别表示用户 u 和用户 v 对物品 i 的平均评分

3. 根据用户集预测评分的算法为

$$p(\mathbf{u},i)=\bar{r}_{\mathbf{u}}+\frac{\sum_{\mathbf{v}\in V}sim(\mathbf{u},\mathbf{v})\times(r_{\mathbf{v},i}-\bar{r}_{\mathbf{v}})}{\sum_{\mathbf{v}\in V}\left|sim(\mathbf{u},\mathbf{v})\right|}$$

V 是存在 k 个相似用户的用户集。


推荐系统(协同过滤实现) 流程图
读取训练数据集
生成U-I矩阵
预处理
读取测试数据集
[test list end]
[not end]
[user exited]
[user missed]
[item exited]
[item missed]
[item exited]
[item missed]
找出KNN
rating = user.avrg
rating = item.avrg
rating = 3
Pearson’s correlation coefficient
predict
[continue next test record]
输出结果

四，参数设置分析

1. //KNN –K 的设置
   //（太小不稳定，太大会将相似度太小的邻居也加进来，计算量增加）
   USER_KNEIGHBOURS: 20
   ITEM_KNEIGHBOURS: 10

2. //相似性阈值
   //将相似性太小的邻居丢弃
   USER_SIMILARITY_CUT: 0.2
   ITEM_SIMILARITY_CUT: 0.2

3. //重要性加权
   //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．
   USER_SIGNIFICANCE_WEIGHTING: 150
   ITEM_SIGNIFICANCE_WEIGHTING: 100

4. //User-Based 与 Item-Based 比重
   //在对每条测试进行预测时，分别进行 User-based 和
   //Item-based 预测,然后按权重取值。
   //这样做稳定了最终的结果。
   RATIO(USER): 0.5
   RATIO(ITEM): 0.5

五，实验结果分析

MAE: 0.67675
RMSE: 0.822648
RUN TIME: 494.484 sec == 8.5 分钟左右
（硬件：CPU ：Inter Core i5-460M 和 4G 内存，操作系统：Apple Mac 此系

统无限制 CPU 使用，可以 100%）