

# One Class Collaborative Filtering Algorithm Based on Transfer Learning

Shengmei Luo<sup>1</sup>, Yunzhen Lin<sup>2</sup>, Xiaowei Ye<sup>1</sup>, Hailong Wen<sup>2</sup>

<sup>1</sup>ZTE Corporation, Nanjing

<sup>2</sup>School of Software, Tsinghua University, Beijing

Email: [luo.shengmei@zte.com.cn](mailto:luo.shengmei@zte.com.cn), [ynlnlyz@126.com](mailto:ynlnlyz@126.com), [ye.xiaowei@zte.com.cn](mailto:ye.xiaowei@zte.com.cn), [youxiaabsyw@gmail.com](mailto:youxiaabsyw@gmail.com)

Received: Nov. 22<sup>nd</sup>, 2012; revised: Dec. 20<sup>th</sup>, 2012; accepted: Dec. 28<sup>th</sup>, 2012

**Abstract:** Collaborative filtering is a useful algorithm for problems of personalized recommendation. For these problems, there are many mature collaborative filtering algorithms. One class collaborative filtering is a new field of personalized recommendation. Because of its data characteristics, common collaborative filtering algorithms have a lot of defects in the field of one class collaborative filtering. We studied the algorithm based on weighted matrix decomposition, and optimized this algorithm by transfer learning. We prove the improvement of this optimization by experiments.

**Keywords:** Recommendation; Collaborative Filtering; One Class; Transfer Learning

## 基于迁移学习的单类协同过滤算法

罗圣美<sup>1</sup>, 林运祯<sup>2</sup>, 叶小伟<sup>1</sup>, 文海龙<sup>2</sup>

<sup>1</sup>中兴通讯, 南京

<sup>2</sup>清华大学软件学院, 北京

Email: [luo.shengmei@zte.com.cn](mailto:luo.shengmei@zte.com.cn), [ynlnlyz@126.com](mailto:ynlnlyz@126.com), [ye.xiaowei@zte.com.cn](mailto:ye.xiaowei@zte.com.cn), [youxiaabsyw@gmail.com](mailto:youxiaabsyw@gmail.com)

收稿日期: 2012年11月22日; 修回日期: 2012年12月20日; 录用日期: 2012年12月28日

**摘要:** 协同过滤算法是现在个性化推荐领域流行的算法。对常见的推荐问题, 协同过滤算法已有成熟的实现。单类协同过滤问题是推荐领域的一个新问题, 其数据特征导致其不适用于常见的协同过滤算法。本文研究了基于加权矩阵分解的单类协同过滤算法, 并对其进行基于迁移学习的改进。通过在真实数据集上的验证, 证明其效果优于传统的单类协同过滤算法。

**关键词:** 推荐系统; 协同过滤; 单类; 迁移学习

### 1. 引言

随着互联网的发展, 很多有一定规模的公司都积累了大量的用户数据, 如何利用这些数据找出用户的使用倾向成为商家的迫切需求; 同时, 作为普通用户, 如何从互联网的海量数据当中找出自己最需要的内容, 也是用户的急切需要。这些事实成为推动推荐技术发展的主要助力。

而无论对于企业还是用户而言, 对每个用户产生针对其使用特点的个性化推荐都是远远优于给所有

用户同样的推荐内容的。因此, 个性化推荐成为推荐技术的主要发展方向。对于个性化推荐, 各种成熟的协同过滤算法是其主要实现方式。

在个性化推荐问题当中, 有一类特殊协同过滤问题。其数据集的用户和项之间缺乏或者完全没有打分数据, 而只有二值化的点击数据(如对于新闻网站, 用户只有点击与否的二值数据; 对视频网站, 用户只有观看与否的二值数据; 都缺乏数值化的打分数据)。这种情况, 被称作“单类协同过滤”。单类协同过滤问

题近年来得到了广泛的研究。

本文针对单类协同过滤问题,提出了一种基于迁移学习的单类协同过滤算法。并在实际数据集上对其进行了验证。实验结果表明,该算法的效果优于传统的协同过滤算法,以及基于权值矩阵分解的单类协同过滤算法。

## 2. 背景算法

### 2.1. 单类协同过滤的基础算法

单类协同过滤问题,有几种直观的解决方式,一种是在数据集中标注负评分值的样本,或是引入其他数据来源的评分,从而将其转化为传统的协同过滤问题。但这种方法通常是需要人工实现的。并且由于难以找到合理的新数据来源,实现的难度很大。

另一种方法是把所有的丢失数据(即用户没有正面操作(如点击、观看)的样本)当作负样本(AMAN, All Missing Are Negative),然后通过协同过滤算法进行计算。这种方法较容易被实现。

在 AMAN 的前提下,大多数传统的协同过滤算法能够直接应用:在 AMAN 基础上可实现基于矩阵分解的协同过滤算法,如奇异值分解(SVD)等,还可实现基于用户相似度和项相似度的协同过滤方法。例如,AMAN 的基于 SVD 分解的执行过程是:

1) 从用户的浏览数据提取评分矩阵  $R$ 。在矩阵  $R$  中,用户有浏览记录的项的评分设为 1,其余设为 0。

2) 对矩阵  $R$  进行 SVD 分解,得到分解结果  $R = USV$ 。

3) 对目标用户  $u$ ,将矩阵  $U$  的第  $u$  行与  $V$  中不在  $u$  的浏览列表中的项目的第  $i$  行进行相乘,即得到用户  $u$  对项  $i$  的评分结果。对评分结果进行排序,取前  $K$  个推荐结果,得到 TopK 推荐。

还有一种方法是,把丢失数据当成未知(AMAU, All Missing Are Unknown),亦即忽略所有的丢失数据并把正样本通过分类或预测算法进行预测,将单类协同过滤问题转化为分类或预测问题。以贝叶斯分类为例,算法过程为:

1) 记录用户对项的浏览记录,储存成用户-项的浏览向量。

2) 预测用户  $u$  对项  $i$  是否应推荐时,构建分类数为两类(推荐或不推荐)的贝叶斯分类器,训练数据为

除用户  $u$  外的用户的浏览向量,以及这些用户是否浏览了  $i$ 。测试数据为用户  $u$  的浏览向量。

3) 通过分类器得出用户  $u$  对  $i$  是否应推荐。

### 2.2. 基础算法的局限性

对 AMAN 而言,其思路符合基本的协同过滤思想。但因为有些丢失数据实际上是正样本,其推荐结果可能会有偏向性。而对于 AMAU,其实现过程无视所有的丢失数据并把正样本通过分类算法进行预测,这种方法会产生一些无意义的结果,亦即所有对丢失数据的预测都是正面的,没有区分性。所有的丢失数据都是负样本(AMAN)和所有丢失数据都是未知的(AMAU)是这个问题的两种极端的解决方法。

## 3. 相关算法的改进

### 3.1. 单类协同过滤的数据集特征

单类协同过滤的数据集的数据特征是其可供学习的数据极少。常见推荐算法的主要制约就是数据的稀疏性。而单类协同过滤问题的数据取值范围较一般的协同过滤问题大大缩小,因而数据包含的信息也较少。这是单类协同过滤问题面临的瓶颈。

### 3.2. 单类协同过滤的优化方向

针对单类协同过滤的数据特点,可以得出的主要优化方向有两个:

1) 从待推荐数据集中挖掘隐含信息。

无论是 AMAU 的分类思想,还是 AMAN 比较用户或项间的相似度的思想,都是直接利用数据集内个体与个体的相似度,使用相似度进行计算,从而赋予待评分项评分。从稀疏数据中挖掘信息的一个通用方法是矩阵分解。但直接将 SVD 分解应用在单类协同过滤上,没有将丢失项与该项对应的用户或项的潜在信息结合起来。因而可以考虑针对用户或项的已有信息(如评分的数目)给予丢失项以一定的权值,从而改善矩阵分解的效果。

2) 将来自别的数据集的信息引入待推荐数据集。

针对单类协同过滤数据集的稀疏性,可以考虑从别的数据集引入信息到现有数据集,以增加矩阵分解时可以利用的信息。可行的方法有结合基于内容的推荐、结合用户信息、以及通过迁移学习引入等。

### 3.3. 基于加权矩阵分解的算法

单类协同过滤问题中，常用“1”用来表示有记录的样本，“0”用来表示没有记录的缺失数据。然而，因为缺失项中可能有隐含的正样本，这样的处理方式可能会有遗漏。因此，可以赋予通过给丢失项以低的权重来处理这个问题。这个思想是从数据集中挖掘隐含信息来优化单类协同过滤问题的。由这个思想，可以导出基于加权矩阵分解的单类协同过滤算法(wALS)<sup>[1]</sup>。

加权矩阵分解的基本思想是，将原评分矩阵  $\mathbf{R}$  分解成两个秩较低的矩阵  $\mathbf{U}$  和  $\mathbf{V}$ 。使  $\mathbf{U}$  和  $\mathbf{V}$  的乘积尽量逼近  $\mathbf{R}$ ，再用  $\mathbf{U}$  和  $\mathbf{V}$  所蕴含的特征向量来计算推荐结果。而加权指的是在进行  $\mathbf{U}$  和  $\mathbf{V}$  的乘积对  $\mathbf{R}$  的逼近时，根据原评分矩阵的特性，给予逼近时用来比较的项一定的权重，从而在逼近时改善丢失项的处理<sup>[1]</sup>。

给定与  $m$  个用户及  $n$  个项相关的评分矩阵  $\mathbf{R} = (\mathbf{R}_{ij})_{m \times n} \in \{0,1\}^{m \times n}$  以及相应的权值矩阵  $\mathbf{W} = (\mathbf{W}_{ij})_{m \times n} \in \mathbf{R}_+^{m \times n}$ ，加权低秩逼近的目标是用一个低秩的矩阵  $\mathbf{X} = (\mathbf{X}_{ij})_{m \times n}$  来使目标函数  $L(\mathbf{X})$  最小化，其中：

$$L(\mathbf{X}) = \sum_{ij} \mathbf{W}_{ij} (\mathbf{R}_{ij} - \mathbf{X}_{ij})^2$$

并使用  $\mathbf{X}$  矩阵作为推荐结果。

在这个目标函数  $L(\mathbf{X})$  中， $(\mathbf{R}_{ij} - \mathbf{X}_{ij})^2$  是在低秩逼近中常见的误差项，而  $\mathbf{W}_{ij}$  反映了最小化该项对目标函数的贡献。在单类协同过滤问题中，为有评分的正样本设置  $\mathbf{R}_{ij} = 1$ 。对没有评分的丢失项，设置  $\mathbf{R}_{ij} = 0$ 。因为在  $\mathbf{R}_{ij} = 0$  处的样本有较高的可信度，设置其相应的  $\mathbf{W}_{ij}$  为 1。作为对比，减低“负”样本的权值。通常在  $\mathbf{R}_{ij} = 0$  处设置  $\mathbf{W}_{ij} \in [0,1]$ 。

首先处理如何使目标函数最小化的问题。

对  $\mathbf{X}$  进行分解，令  $\mathbf{X} = \mathbf{UV}^T$ ，其中有  $\mathbf{U} \in \mathbf{R}^{m \times d}$  和  $\mathbf{V} \in \mathbf{R}^{n \times d}$ 。通常  $d$  远小于矩阵  $\mathbf{R}$  的秩。为了避免过拟合，加入正则化项，有：

$$L(\mathbf{U}, \mathbf{V}) = \sum_{ij} \mathbf{W}_{ij} (\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 + \lambda (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2)$$

或者：

$$L(\mathbf{U}, \mathbf{V}) = \sum_{ij} \mathbf{W}_{ij} \left( (\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 + \lambda (\|\mathbf{U}_i\|_F^2 + \|\mathbf{V}_j\|_F^2) \right)$$

在上述两个公式中， $\lambda$  是正则化项的参数，在不同的情况下， $\lambda$  由交叉校验决定。

对  $L$  分别求对  $\mathbf{U}$  或  $\mathbf{V}$  的偏导数，并令偏导数为 0，可得  $\mathbf{U}$  和  $\mathbf{V}$  的迭代更新公式：

$$\mathbf{U}_i = \mathbf{R}_i \mathbf{W}_i \mathbf{V} \left( \mathbf{V}^T \mathbf{W}_i + \lambda \left( \sum_j \mathbf{W}_{ij} \right) \mathbf{I} \right)^{-1}, \forall 1 \leq i \leq m$$

$\widetilde{\mathbf{W}}_i \in \mathbf{R}^{n \times n}$  是将向量  $\mathbf{W}_i$  包含的值放置在一个零矩阵的对角线上而生成的对角矩阵。 $\mathbf{I}$  是  $d$  维的单位矩阵。

$$\mathbf{V}_j = \mathbf{R}_j^T \widetilde{\mathbf{W}}_j \mathbf{U} \left( \mathbf{U}^T \widetilde{\mathbf{W}}_j \mathbf{U} + \lambda \left( \sum_i \mathbf{W}_{ij} \right) \mathbf{I} \right)^{-1}, \forall 1 \leq j \leq n$$

$\widetilde{\mathbf{W}}_j \in \mathbf{R}^{m \times m}$  是将向量  $\widetilde{\mathbf{W}}_j$  包含的值放置在一个零矩阵的对角线上而生成的对角矩阵。

基于以上公式，就得到基于迭代的 wALS 算法<sup>[2-4]</sup>。

算法过程：

输入：评分  $\mathbf{R}$ 、权重、特征数  $d$

输出：秩为  $d$  的矩阵  $\mathbf{U}$  和  $\mathbf{V}$

随机初始化  $\mathbf{V}$

重复以下过程：

对所有  $\mathbf{U}_i$ ，通过迭代公式更新之

对所有  $\mathbf{V}_j$ ，通过迭代公式更新之  
直至收敛

返回  $\mathbf{U}$  和  $\mathbf{V}$

分解后，得到的子矩阵中，有包含用户侧的潜在语义信息的矩阵  $\mathbf{U}$ ，以及包含项侧的潜在语义信息的矩阵  $\mathbf{V}$ 。 $\mathbf{U}$  中的每一行表示用户潜在倾向。对目标用户  $u$ ，将  $\mathbf{U}$  的第  $u$  行与  $\mathbf{V}$  中未被推荐的项目的第  $i$  行进行相乘，即得到用户  $u$  对项  $i$  的评分结果。对评分结果进行排序，取前  $K$  个推荐结果，得到 TopK 推荐。

### 3.4. wALS 算法的分析与改进

wALS 算法通过挖掘评分矩阵的潜在数据，将丢失项赋予不同的权值，从而在逼近时提高了合理性。但在数据量较大时，这个算法的执行效率会成为其瓶颈。为了提高这个算法的执行效率，R. Pan 等人提出了基于将权值矩阵进行分解的优化算法<sup>[5]</sup>，提高了算法的执行速度。Yifan Hu 等人提出了一种基于置信度的加权矩阵分解协同过滤算法<sup>[6]</sup>，并对其进行了执行

效率上的优化。本文主要侧重于从准确率上对 wALS 算法进行进一步的优化。本文的基于迁移学习的优化方法,亦可以应用在 R. Pan 和 Yifan Hu 等的在效率上改进后的 wALS 算法上。

### 3.5. 权值策略的选取

wALS 算法过程中,  $\mathbf{W}$  矩阵的生成方法是一个关键因素。正样本有相对高的可信度,因此可让  $R_{ij} = 1$  处  $W_{ij} = 1$ 。丢失项大多数都应是负样本,所以应该给予“负”样本更低的权值<sup>[1]</sup>。对包含加权矩阵的  $F$  范式最优化问题,可通过上文求导数的方法解决, [7]中有详细描述。

有三种朴素的权重策略,第一个权值策略认为一个丢失项是负样本的可能性在任何条件下都是相同的。第二种权值策略认为如果一个用户有较多的正样本,他有可能不喜欢这些样本以外的项,那么他的丢失项就更有可能是负样本。第三种权值策略认为如果一个项有较少的正样本,那么与这个项相关的丢失项就更有可能是负样本。如表 1。

### 3.6. 基于迁移学习的优化

基于 wALS 的单类协同过滤算法虽然优于其他朴素的单类协同过滤算法,但往往受制于数据的稀疏而难以达到很好的效果。单类数据(例如新闻的浏览推荐)因其是点击类型的数据,往往有较多的辅助数据可以使用。例如,在用户侧可以使用用户其他操作的数据(如对视频的观看)作为辅助,项一侧也可以使用诸如新闻的互相引用的数据<sup>[8]</sup>。可以通过迁移学习的方式,将这些辅助数据引入到 wALS 算法中。这种优化方式的思想是将来自别的数据集的信息引入数据集的优化思想。

除迁移学习之外, Li 等人提出了通过提取用户信息对单类协同过滤算法进行优化的方法<sup>[9]</sup>。但这种方法的限制条件较严格,需要推荐目标领域能挖掘到丰

富的用户特征。

将迁移学习应用到 wALS 算法中,需要一个目标矩阵。目标推荐领域用以进行最终的推荐工作,而辅助推荐领域(评分矩阵)和一到两个辅助推荐领域(评分辅助推荐领域则提供相关的迁移知识。

一般而言,辅助领域分为用户侧的辅助领域和项侧的辅助领域。用户侧的辅助领域与目标推荐领域拥有相同的用户集合,而项集合不同。项侧的辅助领域与目标推荐领域拥有相同的项的集合,而用户集合不同。例如,目标矩阵可以是某个网站 a 的用户群对该网站图书类商品的评分,而用户侧的辅助领域可以是 a 网站的用户群对该网站音乐类商品的评分,项侧的辅助领域可以是网站 b 的用户群对与 a 网站图书类商品的相同商品的集合的评分。

要将辅助领域的知识迁移到目标领域,就必须借助矩阵分解的方法。从用户辅助领域得到用户侧的知识,而从项侧的领域得到项侧的知识。

基于加权矩阵分解的协同过滤是要求使以下目标函数最小化:

$$L(\mathbf{U}, \mathbf{V}) = \sum_{ij} W_{ij} (\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2$$

其中  $\mathbf{W}$  为权值矩阵;  $\mathbf{U}$  矩阵和  $\mathbf{V}$  矩阵分别表达了用户层面和项层面的潜在特征。

对目标推荐领域,有用户-项的评分矩阵  $\mathbf{R}$ ,而对辅助推荐领域,也有用户-项的评分矩阵  $\mathbf{R}^{(1)}$  和  $\mathbf{R}^{(2)}$ ,对他们分别进行 SVD 分解,使  $\mathbf{R}^{(1)} = \mathbf{U}^{(1)} \mathbf{B}^{(1)} \mathbf{V}^{(1)}$ ,  $\mathbf{R}^{(2)} = \mathbf{U}^{(2)} \mathbf{B}^{(2)} \mathbf{V}^{(2)}$ 。设领域 1 为用户方面的辅助领域,领域 2 为项方面的辅助领域,则可取  $\mathbf{U}^{(1)}$  和  $\mathbf{V}^{(2)}$  参与目标推荐领域的推荐工作,将其作为目标函数的补偿项,代替原来的正则化项。将目标函数修正为:

$$L(\mathbf{U}, \mathbf{V}) = \sum_{ij} W_{ij} (\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 + \frac{\rho_u}{2} \|\mathbf{U}^{(1)} - \mathbf{U}\|_F^2 + \frac{\rho_v}{2} \|\mathbf{V}^{(2)} - \mathbf{V}\|_F^2$$

或者

$$L(\mathbf{U}, \mathbf{V}) = \sum_{ij} W_{ij} \left( (\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 + \frac{\rho_u}{2} \|\mathbf{U}^{(1)}_i - \mathbf{U}_i\|_F^2 + \frac{\rho_v}{2} \|\mathbf{V}^{(2)}_j - \mathbf{V}_j\|_F^2 \right)$$

Table 1. Weighting strategies  
表 1. 权重取值策略表

	正样本	“负”样本
标准	$W_{ij} = 1$	$W_{ij} = \delta$
偏向用户	$W_{ij} = 1$	$W_{ij} \propto \sum_j R_{ij}$
偏向项	$W_{ij} = 1$	$W_{ij} \propto m - \sum_i R_{ij}$

在更新  $U$ 、 $V$  时，可求得目标函数对  $U$  的偏导数为：

$$\begin{aligned} \frac{1}{2} \frac{\partial L(U, V)}{\partial U_{ik}} &= \sum_j W_{ij} (U_i V_j^T - R_{ij}) V_{jk} \\ &+ \frac{\rho_u}{2} \left( \sum_j W_{ij} \right) (U_{ik} - U_{ik}^{(1)}), \\ &\forall 1 \leq i \leq m, 1 \leq k \leq d \end{aligned}$$

其中，对  $U$  的第  $i$  行  $U_i$  的导数认为是：

$$\begin{aligned} \frac{1}{2} \frac{\partial L(U, V)}{\partial U_i} &= \frac{1}{2} \left( \frac{\partial L(U, V)}{\partial U_{i1}}, \dots, \frac{\partial L(U, V)}{\partial U_{id}} \right) \\ &= U_i \left( V^T \widetilde{W}_i V + \frac{\rho_u}{2} \left( \sum_j W_{ij} \right) I \right) \\ &- R_i \widetilde{W}_i V - U_i^{(1)} \frac{\rho_u}{2} \left( \sum_j W_{ij} \right) I \end{aligned}$$

令偏导为 0，即可求得在该迭代中  $U$ 、 $V$  的值。

$$\begin{aligned} U_i &= \left( R_i \widetilde{W}_i V + U_i^{(1)} \frac{\rho_u}{2} \left( \sum_j W_{ij} \right) I \right) \\ &\left( V^T \widetilde{W}_i V + \frac{\rho_u}{2} \left( \sum_j W_{ij} \right) I \right)^{-1} \end{aligned}$$

而  $V$  的处理过程与  $U$  是对称的。

## 4. 实验及实验结果

### 4.1. 数据集的选取

针对单类协同过滤问题，我们选取了 Movielens 数据集进行测试。

Movielens 数据集<sup>[10]</sup>，是一个在协同过滤领域被广泛应用的数据集。该数据集来自 GroupLens 收集的用户对电影的评分。我们使用的数据集是 Movielens 的一部分，包含 943 个项和 1682 个用户，用户对项有 1~5 的整数评分。评分共 100,000 个。为了使该数据集单类化，我们用了如下策略：将评分 4 和评分 5 的评分视为 1，其余视为丢失项。

为了取得迁移学习的辅助数据，将原始数据集分为四份。用来进行实验的部分包括原始数据集的一半用户和一半项；用原始数据集中与实验部分具有相同用户，但项不同的部分作为用户侧辅助数据；用原始数据集中与实验部分具有相同项，但用户不同的部分作为项侧辅助数据。将辅助数据的评分矩阵进行 SVD

分解，并将相应的辅助  $U$  和  $V$  矩阵传入实验数据的推荐过程。

### 4.2. 实验设置与评价标准

实验采用交叉验证法对结果进行验证。数据集以 80/20 的比例被分为训练集和测试集。对数据集进行多次采样，并计算结果的平均值。

测试集的评价使用了基于 MAP 和 HLU 的方法。

基于 HLU 的误差度量对位于较前的推荐项的对错赋予较大的权值。在进行 Top-N 推荐时，用户比较关心推荐结果的头几项是否正确，而对推荐结果较靠后的项的正误并不关心。基于 HLU 的误差度量使用了这样的权值策略：排名第一的资源权值为 1/2，排名第二的资源权值为 1/4，其次为 1/8，等等。这种评价方式突出了推荐项真实的重要性。

HLU 定义如下公式：

$$R = \frac{\sum_u R_u}{\sum_u R_u^{\max}}$$

$R_u$  是对于用户  $u$  的 HLU 效应值， $R_u^{\max}$  是对该用户当，所有的正确推荐结果都在推荐结果列表顶部时达到的最大效应值。 $R_u$  的定义如下：

$$R_u = \sum_j \frac{\delta(j)}{2^{(\beta-1)(j-1)}}$$

当排在第  $j$  个的项是正确结果时  $\delta(j)$  为 1，否则为 0。 是用于调整结果数量级的参数。

MAP(平均准确率均值)是平均准确率(AP)的对于所有用户的均值，其思想与 HLU 类似，排名较前的项具有较大的权重。

对特定用户的 AP 通过以下公式计算：

$$AP_u = 100 \frac{\sum_{i=1}^N \text{prec}(i) * \text{pref}(i)}{\#}$$

$i$  是在推荐结果列表中的位置，而  $N$  是推荐项的总数。 $\text{prec}(i)$  是(被用户所喜欢的)从 1 到  $i$  的列表的一部分的准确度， $\text{pref}(i)$  是一个二元标志符，当用户喜欢第  $i$  个项时其为 1，否则为 0。

### 4.3. 参与实验的算法

参与实验的算法有以下几种：

1) AMAU 策略的基于流行度的分类推荐算法。对特定用户的推荐是除该用户已有评分项外最流行的项。

2) AMAN 策略。有基于用户相似度的协同过滤算法、基于项相似度的协同过滤算法以及基于 SVD 矩阵分解的协同过滤算法。这些算法的介绍详见 2.1 节。

3) wALS 算法，分别是权值偏向用户的 wALS 算法、权值偏向项的 wALS 算法、标准权值策略的 wALS 算法。

4) 基于迁移学习的单类协同过滤算法。

#### 4.4. 实验结果及分析

各个方法的结果对比(见表 2)。

从实验结果看, AMAN 的几种方法对单类协同过滤问题基本是无意义的。这也说明了单类协同过滤问题的特殊性, 亦即其数据集可用数据十分稀疏。这种特性使传统的协同过滤算法在单类协同过滤问题上很难起作用。也证明了单类协同过滤问题亟待更多有效的算法。

wALS 算法在不同的权值策略下有截然不同的表现。这说明 wALS 算法的效果对权值策略较为敏感。因选择何种权值策略与使用的数据集密切相关, 因此在实际生产中使用 wALS 算法时需要多种权值策略进行测试。

Table 2. Experimental result  
表 2. 实验结果表

	HLU	MAP
AMAN - 用户	0.0	0.0
AMAN - 项	0.0	0.0
AMAU	3.13	3.23
AMAN-SVD	0.07	0.35
wALS - 标准	7.56	9.40
wALS - 项	13.31	10.86
wALS - 用户	2.98	3.29
迁移学习 - 用户	25.16	11.92
迁移学习 - 项	27.09	11.33

实验结果表明, 迁移学习对推荐结果有了一定程度的提高。

#### 5. 结论

本文提出的基于迁移学习的单类协同过滤算法, 通过实验得到了预期的结果, 从而验证了算法的优越性。该算法针对单类协同过滤数据集的特征, 从数据集中挖掘隐藏信息及来自别的数据集的信息引入数据集两方面进行了优化, 是一种较高效的协同过滤算法。但算法仍需结合用户信息、结合基于内容的推荐等方面进行进一步的优化。

#### 参考文献 (References)

- [1] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. M. Lukose, M. Scholz and Q. Yang. One-class collaborative filtering. IEEE International Conference on Data Mining, 15-19 December 2008: 502-511.
- [2] N. D. Buono, T. Politi. A continuous technique for the weighted low-rank approximation problem. Lecture Notes in Computer Science, 2007, 3044: 988-997.
- [3] S. Oh. Matrix completion: Fundamentak limits and efficient algorithms. Stanford University, 2010.
- [4] P. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. New Brunswick: Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics, July 2002: 417-424.
- [5] R. Pan, M. Scholz. Mind the gaps: Weighting the unknown in large-scale one-class collaborative filtering. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009: 667-676.
- [6] Y. Hu, Y. Koren and C. Volinsky. Collaborative filtering for implicit feedback datasets. Proceedings of the 2008 8th IEEE International Conference on Data Mining, 2008: 263-272.
- [7] N. Srebro, T. Jaakkola. Weighted low-rank approximations. Proceedings of the 20th International Conference on Machine Learning, 2003.
- [8] W. Pan, E. W. Xiang, N. N. Liu and Q. Yang. Transfer learning in collaborative filtering for sparsity reduction. Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10), 2010
- [9] Y. Li, J. Hu, C. X. Zhai and Y. Chen. Improving one-class collaborative filtering by incorporating rich user information. Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010: 959-968.
- [10] <http://www.grouplens.org/node/73>