基于矩阵分解的单类协同过滤推荐算法*

李 改[1,2,3], 李 磊[2,3]

(1. 顺德职业技术学院 电子与信息工程系, 广东 顺德 528333; 2. 中山大学 信息科学与技术学院, 广州 510006; 3. 中山大学 软件研究所, 广州 510275)

摘 要: 新闻网页和书签的推荐被认为是单类协调过滤问题。通常这类数据是相当稀疏的,仅仅一小部分数据是正例,在非正例数据中负例和没有标记的正例是混合在一起的,难以区分开来,因此,就如何解释非正例数据出现了歧义。为了解决该问题,提出了一种加权的带正则化的基于迭代最小二乘法的单类协同过滤算法。即通过对正例赋予权值 1,负例赋予一个较小的正实数权值来反映数据的正负置信度。在两个真实的实验数据集上验证了该算法在性能上均优于几个经典的单类协同过滤推荐算法。

关键词: 推荐系统; 单类协同过滤; 矩阵分解; wALS

中图分类号: TP311 **文献标志码**: A **文章编号**: 1001-3695(2012)05-1662-04

doi:10.3969/j.issn.1001-3695.2012.05.016

One-class collaborative filtering based on matrix factorization

LI Gai[1,2,3], LI Lei[2,3]

(1. *Dept. of Electronics & Information Engineering, Shunde Polytechnic, Shunde Guangdong* 528333, *China*; 2. *School of Information Science & Technology, Sun Yat-Sen University, Guangzhou* 510006, *China*; 3. *Software Institute, Sun Yat-Sen University, Guangzhou* 510275, *China*)

Abstract: News item recommendation and bookmarks recommendation are most naturally thought of as OOCF problems. Usually this kind of data are extremely sparse, just a small fraction are positive examples. Negative examples and unlabeled positive examples are mixed together and are typically unable to distinguish them, therefore ambiguity arises in the interpretation of the non-positive example. This paper proposed a CF algorithm-weighted alternating least squares(wALS). That was, by using weighting scheme assigning "1" to observed examples and low positive real number weights to unobserved examples to reflect the confidence of positive examples and negative examples. The experimental evaluation using two real-world datasets shows that wALS achieves better results in comparison with several classical one-class collaborative filtering recommendation algorithms.

Key words: recommendation systems; one-class collaborative filtering(OOCF); matrix decomposition; wALS

0 引言

随着互联网的快速发展,互联网上的数据量急剧增长,从而使得如何快速而高效地从如此浩瀚的数据海洋中获取人们所需要的信息变得越来越紧迫。在此背景下,推荐系统应运而生。推荐系统通过收集和分析用户的各种信息来学习用户的兴趣和行为模式,根据分析得到的用户兴趣和行为模式,为用户推荐他所需要的服务。这些系统的例子包括卓越亚马逊(www.amazon.cn)、当当网(www.dangdang.com)为用户推荐各种其可能喜欢的商品,如书籍、音像、电器、服装等;Netflix 电影出租系统(www.netflix.com)为用户推荐各种其可能喜欢的电影;Google、Baidu、Yahoo 等为用户推荐个性化的新闻和搜索服务。推荐系统中运用最广泛的是基于协同过滤的推荐算法[1~3]。

协同过滤的算法核心是分析用户兴趣,在用户群中找到与指定用户的相似(兴趣)用户,综合这些相似用户对某一信息的评价,形成系统对该指定用户对此信息的喜好程度预测[4,5]。近年来协同过滤的算法在国内外得到了广泛研究,按处理数据的不同主要分为两类:一类是处理明确的偏好数据,如评分;另一类则是处理隐式数据,如对网页点击与否。这种隐式数据广泛存在于真实世界的应用环境中,如用户是否购买过某个商品,用户是否点击过某个网页等。由于这里的信息不需要用户提供明确的评分,因此相比评分数据更容易获取。该类数据中仅有正例可以明确区分开来,负例不确定,故把该类问题称为单类协同过滤(OCCF)问题。单类协同过滤的任务就是通过分析这些隐式信息来针对特定用户的偏好对推荐对象集按该用户的喜欢程度排序。尽管这类数据获取很容易,但解释起来却存在着很大的困难。以用户点击网页的数据为例,这些数据中用户点击过的网页构成的数据可以解释为正例,其余数据是负例和漏掉的正例的混合。如何解释这类混合数据,以及如何来对解释后的数据进行有效处理,是当前单类协同过滤问题研究的难点所在。目前对该问题的研究还很少。Zhou 等人[6]把低秩逼近(LRA)技术运用到单类协同过滤问题,把

收稿日期: 2011-09-01; **修回日期**: 2011-10-19 **基金项目**: 国家自然科学基金资助项目(61003140,61033010);中山大学高性能与网格计算平台资助项目

作者简介: 李改(1981-),男,讲师,博士研究生,主要研究方向为数据挖掘、推荐系统(ligai999@126.com);李磊(1951-),男,教授,博导,博士,主要研究方向为数据库、数据挖掘、人工智能.

观察到的点击数据作为正例数据,其余的混合数据均作为负例数据;Paterek[7]提出运用奇异值分解(SVD)技术来解决该类问题;Rendle 等人[8]提出运用基于 KNN 的协同过滤算法来解决该类问题。

本文的主要贡献是:在前人的研究基础上提出一种加权的迭代最小二乘法(wALS)来解决单类协同过滤问题,即对明确的正例数据赋予权重 1,对于无法解释的混合数据赋予一个小于 1 大于 0 的正实数权重,进而在真实的数据集上实现本文所提出的算法,并将其与几个传统的 OCCF 算法的性能做比较。实验结果表明,wALS 算法在各个数据集下均优于几个传统的 OCCF 算法。

1　相关工作

协同过滤算法的研究数据按不同的系统设计可大致分为三种数据类型[2]:

a)评分(rating)数据。在理想情形下,系统会提示用户输入对不同对象的喜好/厌恶程度。它用分数表示,一般是 1 ~ 5 星(分),1 星表示非常不喜欢,5 星代表非常喜欢。

b)正负面(thumbs up ↑/thumbs down ↓)的评价数据。用户不给出喜恶程度,只表达是否喜欢。

c)基于正面反馈(positive feedback)的数据。在一些情况下,不能收集类似前两类的显式评分数据,但是可以收集一些用户的行为(用户阅读了哪些新闻或购买了哪些产品),一般假设这些行为数据反映了用户的兴趣,因而把这类数据称为基于“正”反馈的数据。

当前对于协同过滤算法的研究大多数均集中在对评分数据的研究[2],该类数据既有正例数据(高评分点),又有负例数据(低评分点)。对该类问题的研究,Srebro 等人提出了 MMMF(maximum-margin matrix factorization),Salakhutdinov 等人提出了 PMF(probabilistic matrix factorization)和 RBM(restricted Boltzmann machines),Lee 等人提出了 NNMF(nonnegative matrix factorization)。对于正负面的评价数据,以协同过滤的角度来研究的很少,大多当做两类分类问题来看待[9,10]。

在现实世界的实际应用中,基于正面反馈(positive feedback)的数据也广泛存在,当前对该类问题的研究主要也是把其看做两类的分类问题,如文献[9]把该问题当做只有正类的两类分类问题来看待,提出运用 SVM 分类器来解决单类分类问题;文献[11,12]通过 EM 算法来迭代预测负例,从而学习出单类分类器来解决此类单类分类问题。近年来也有些研究者从协同过滤的角度来解决该类问题。

2　基于 wALS 的单类协同过滤算法

2.1　基于 wALS 的单类协同过滤算法简介

给定一个矩阵 $\boldsymbol{R}=(R_{ij})_{m\times n}\in\{0,1\}^{m\times n}$($R_{ij}$表示它的一个元素,$R_{i\cdot}$ 表示矩阵 $\boldsymbol{R}$ 的第 i 行,$R_{\cdot j}$表示矩阵 $\boldsymbol{R}$ 的第 j 列,m 表示用户数,n 表示推荐对象数)和一个权重矩阵 $\boldsymbol{W}=(W_{ij})_{m\times n}\in\{\mathscr{R}_+\}^{m\times n}$($\mathscr{R}_+$表示大于 0 小于 1 的正实数)。在这里,希望找到一个低秩矩阵 $\boldsymbol{X}=(X_{ij})_{m\times n}$来逼近矩阵 $\boldsymbol{R}$。其中:$\boldsymbol{X}=\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}$,$\boldsymbol{U}\in C^{m\times d}$,$\boldsymbol{V}\in C^{n\times d}$,$\boldsymbol{U}$、$\boldsymbol{V}$ 表示用户和推荐对象的特征矩阵,d 表示特征个数,一般 $d<<r$,r 表示矩阵 $\boldsymbol{R}$ 的秩,$r\leqslant\min(m,n)$。

为了找到一个低秩矩阵 $\boldsymbol{X}$ 来最大程度地逼近矩阵 $\boldsymbol{R}$,最小化加权的 Frobenius 损失函数。

$$L(\boldsymbol{X})=\sum_{ij}W_{ij}(R_{ij}-X_{ij})^2 \tag{1}$$

其中:$(R_{ij}-X_{ij})^2$ 是低秩逼近中常见的平方误差项,W_{ij}表示各个数据点对最小化总的加权的 Frobenius 损失函数的贡献。在 OCCF 中,对于正例,设置 $R_{ij}=1$;对于缺失值,假定所有的缺失值均为负例,其值均设置为 0,即 $R_{ij}=0$。因为对正例有很高的置信度,故对于这类数据的权值设置为 1,即 $W_{ij}=1$。与 LRA 不同的是,在本算法中不是简单地把所有的混合数据均当成负例,而是在把它们当成负例的同时给这些数据一个小的权值 $W_{ij}=\alpha\in[0,1]$。

考虑如何有效且快速地求解最优化问题 $\mathrm{argmin}_X L(\boldsymbol{X})$。

式(1)可以改写为

$$L(\boldsymbol{U},\boldsymbol{V})=\Sigma_{ij}W_{ij}(R_{ij}-U_{i\cdot}V_{j\cdot}^{\mathrm{T}})^2 \tag{2}$$

为了防止过拟合,给式(2)加上正则化项,则式(2)可改写为

$$L(\boldsymbol{U},\boldsymbol{V})=\Sigma_{ij}W_{ij}(R_{ij}-U_{i\cdot}V_{j\cdot}^{T})^2+\lambda(\|U_{i\cdot}\|_F^2+\|V_{j\cdot}\|_F^2) \tag{3}$$

固定 $\boldsymbol{V}$,对 U_i 求导$\dfrac{\partial L(\boldsymbol{U},\boldsymbol{V})}{\partial U_i}=0$,得到求解 $U_{i\cdot}$ 的公式:

$$U_{i\cdot}=\boldsymbol{R}_{i\cdot}\widetilde{\boldsymbol{W}}_{i\cdot}\boldsymbol{V}(\boldsymbol{V}^{\mathrm{T}}\widetilde{\boldsymbol{W}}_{i\cdot}\boldsymbol{V}+\lambda(\Sigma_{ij}W_{ij})I)^{-1}\quad i\in[1\sim m] \tag{4}$$

其中:$\boldsymbol{R}_{i\cdot}$ 表示用户 i 评过的推荐对象的评分组成的向量,$\widetilde{\boldsymbol{W}}_{i\cdot}\in\mathrm{R}^{n\times n}$表示一个由 $W_{i\cdot}$ 中的元素构成的对角矩阵。

同理,固定 $\boldsymbol{U}$,可以得到求解 $V_{j\cdot}$ 的公式。

$$V_{j\cdot}=\boldsymbol{R}_{\cdot j}\widetilde{\boldsymbol{W}}_{\cdot j}\boldsymbol{U}(\boldsymbol{U}^{\mathrm{T}}\widetilde{\boldsymbol{W}}_{\cdot j}\boldsymbol{U}+\lambda(\Sigma W_{ij})I)^{-1}\quad j\in[1\sim n] \tag{5}$$

其中:$\boldsymbol{R}_{\cdot j}$表示评过推荐对象 j 的用户的评分组成的向量,$\widetilde{\boldsymbol{W}}_{\cdot j}\in\mathrm{R}^{m\times m}$表示一个由 $W_{\cdot j}$中的元素构成的对角矩阵。

在式(4)(5)中 I 表示一个 $d\times d$ 的单位矩阵。

基于式(4)(5),本文提出下面的加权的基于代正则化的交叉最小二乘法(wALS)的单类协同过滤推荐算法。首先用均值 V 为 0、偏差为 0.01 的高斯随机数初始化矩阵 $\boldsymbol{V}$,然后用式(4)更新 $\boldsymbol{U}$,接着用式(5)更新 $\boldsymbol{V}$,直到本算法计算出的 AUC 值收敛或迭代次数足够多而结束迭代为止。具体算法描述如下:

算法 1　基于 wALS 的单类协同过滤推荐算法

输入:用户的评分矩阵 $\boldsymbol{R}$,特征个数 d。

输出:矩阵 $\boldsymbol{R}$ 的逼近矩阵 $\boldsymbol{X}$。

a)用一个小于 1 的随机数初始化 $\boldsymbol{V}$;

b)反复迭代运用式(4)(5)更新 $\boldsymbol{U}$、$\boldsymbol{V}$,直到本算法计算出的 AUC 值收敛或迭代次数足够多而结束迭代;

c)$\boldsymbol{X}=\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}$,返回矩阵 $\boldsymbol{X}$。

d)运用计算得到的 $\boldsymbol{X}$ 产生 Top$N$ 推荐。

2.2　基于 wALS 的单类协同过滤算法的时间复杂度分析

为了分析本算法的时间复杂度,假定 d 表示特征个数,n_t 表示算法的迭代次数,m 表示用户的个数,n 表示推荐对象的个数。以标准的矩阵操作来计算该算法的时间复杂度,分析式(4)可知每次更新 $\boldsymbol{U}$ 的一行所需要的时间复杂度的上界为 $O(d^2n)$,分析式(5)可知每次更新 $\boldsymbol{V}$ 的一行所需要的时间复杂

度的上界为 $O(d^2m)$,因此更新 U、V 所需要的时间复杂度的上界为 $O(d^2mn)$。假定本算法总共迭代 n_t 次后停止,则本算法总的运行时间复杂度的上界为 $O(d^2mnn_t)$。

综上,可得到有关本算法运行时间复杂度的定理 1。

定理 1 对于 wALS,每次更新 U、V 所需要的时间复杂度的上界为 $O(d^2mn)$,如果算法总共迭代 n_t 次后停止,则其总的运行时间复杂度的上界为 $O(d^2mnn_t)$。

分析算法 1 可知,该算法的关键是第二步,即反复迭代运用式(4)(5)更新 U、V。分析式(4)(5)可知,每次调用式(4)(5)只是计算更新矩阵 U、V 的一行值。因此可对矩阵 U、V 进行分割,分成多个等列长的子矩阵来进行并行运算。故本文所提出的 wALS 协同过滤算法完全可以并行化运算,从而可以解决其他单类协同过滤算法难以并行化、可扩展性差的问题,进而简化本算法的实现复杂度,提高其运算效率。

3 实验结果及分析

本章首先介绍本文实验所采用的数据集及评价标准;接着以 AUC 为评价指标,比较了本文所提出的 wALS 算法和传统的 LRA、SVD、基于用户的 KNN 和基于项目的 KNN 算法的性能[6~8],并对结果进行分析。

3.1 实验数据集

在本实验中使用了两个数据集,一个是 Netflix 数据集,一个是 MovieLens 数据集[2,4,5]。

Netflix 数据集是 Netflix 对外发布的一个电影评分数据集[1,2,13]。这个数据集包括了 480、189 个用户在对 17、770 部电影的 103、297、638 个评分。所有的评分值都是 1 ~ 5 中的整数值,其中分数越高表示客户对相应电影的评价越高(越喜欢)。这个数据集非常稀疏,有将近 99% 的评分值未知。从这个数据集中随机抽取一个包括 10 000 个用户、5 000 部电影的子集,总共包含 569 019 个评分点。该评分子集要求每个用户至少评过 10 部电影,每部电影至少被 10 个用户评过。这个选取的子数据集也非常稀疏,稀疏度为 1.138%,即仅 1.138% 的项有评分。

MovieLens 数据集是由美国 Minnesota 大学的 GroupLens 研究小组创建并维护的[4,5],其中包括 943 个用户对 1 682 部电影的 100 000 条评分记录。所有的评分值也都是 1 ~ 5 中的整数值,其中分数越高表示客户对相应电影的评价越高(越喜欢)。这个数据集也非常稀疏,稀疏度为 6.305%,即仅 6.305% 的项有评分。

实验中把选取的 Netflix 子数据集和 MovieLens 数据集中评分数值为 1 ~ 5 分的数据点均赋值为 1,用于表示正例数据,其余数据点仍为缺失数据,从而把基于评分的数据转变为基于正面反馈的数据。

3.2 实验的评价方法

本文实验采用留一策略作为评价机制,也即从每个用户的评分历史中随机地选取并移除一个评分点作为测试集 S_{test},余下的数据构成训练集 S_{train},这两个集合不相交。在 S_{train} 上训练出相应的模型,接着在测试集上评估模型预测出的推荐对象的个性化排序。这里采用的评估标准是平均 AUC[8]:

$$\text{AUC} = \frac{1}{|U|}\sum_u \frac{1}{|E(u)|}\sum_{(i,j)\in E(u)} \delta(\hat{x}_{ui} > \hat{x}_{uj}) \tag{6}$$

在这里 δ 是一个指标函数。

$$\delta(b) := \begin{cases} 1 & \text{if } b \text{ is true} \\ 0 & \text{else} \end{cases}$$

每个用户的评估对象对为

$$E(u) := \{(i,j) \mid (u,i) \in S_{\text{test}} \wedge (u,j) \notin (S_{\text{test}} \vee S_{\text{train}})\}$$

AUC 值越高表示该算法的性能越好。由随机模型产生的 AUC 值为 0.5,最好模型的 AUC 值为 1。对每个实验均反复运行 10 次,每次均对每个用户随机选取一个评分点,构成新的训练集和测试集,最终结果取 10 次运算结果的平均值。

3.3 实验结果

wALS 算法的参数主要是负例权值参数 α 和正则化参数 λ。本实验中,wALS 算法的这两个参数在 MovieLens 和 Netflix 子数据集上的最优值将通过交叉确认的方式来确定。

本实验将分别在 Netflix 子数据集和完整的 MovieLens 数据集上把本文所提出的 wALS 算法与传统的 LRA、SVD、基于用户的 KNN 和基于项目的 KNN 算法的性能进行比较分析。

图 1 表示在 MovieLens 数据集上,本文所提出的 wALS 算法与传统的 LRA、SVD、基于用户的 KNN 和基于项目的 KNN 算法的性能对比。图 1 中横轴表示 wALS、传统的 LRA 和 SVD 算法中用户/推荐对象的特征矩阵中特征的个数,特征个数从 1 变化到 40,纵轴表示各个算法的 AUC 值。通过实验验证,wALS 算法在 MovieLens 数据集上取得最优值的负例权值参数 $\alpha = 0.15$,正则化参数 $\lambda = 0.015\,625$。对于基于用户的 KNN 和基于项目的 KNN 算法,该实验中的结果取最优值。由于这两种算法的性能与特征矩阵中特征的个数无关,故在各个特征数下取值均一致。从图 1 中可以看出,在各个特征数下,本文所提出的 wALS 算法几乎均优于传统的 LRA 和 SVD 算法,也优于最优的基于用户的 KNN 和基于项目的 KNN 算法,并且这种优势随着特征数的增加越发明显。其中 SVD 算法的 AUC 值随着特征数的增大先增大,而后急速下降。这是由于 SVD 算法在解决单类协同过滤问题时也出现了过拟合现象。

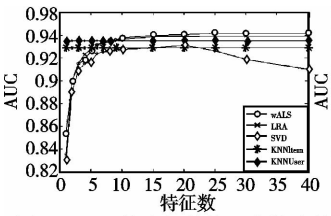

图1 wALS算法与其他经典算法的比较(MovieLens数据集)

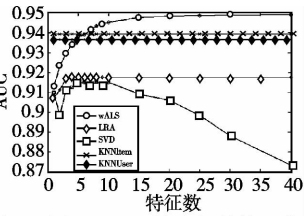

图2 wALS算法与其他经典算法的比较(Netflix子数据集)

图 2 表示在 Netflix 子数据集上,本文所提出的 wALS 算法与传统的 LRA、SVD、基于用户的 KNN 和基于项目的 KNN 算法的性能对比。坐标轴的定义与图 1 一致。通过实验验证,wALS 算法在 Netflix 子数据集上取得最优值的负例权值参数 $\alpha = 0.018$,正则化参数 $\lambda = 0.013$。从图 2 中不难看出,当特征个数大于 7 后,本文所提出的 wALS 算法明显优于传统的 LRA、SVD、基于用户的 KNN 和基于项目的 KNN 算法,且随着特征个数的增大,这种优势愈加明显。在这里基于用户的 KNN 和基于项目的 KNN 算法也均取最优值。图 2 显示在 Net-

flix 子数据集上,这两种 KNN 算法的性能均优于 LRA 和 SVD 算法,这可以理解为 KNN 算法的抗稀疏性更强,而该实验所采用的 Netflix 子数据集相比于本实验所采用的 MovieLens 数据集更加稀疏,故在更稀疏的 Netflix 子数据集上 KNN 算法的性能远优于 LRA 和 SVD 算法。同时在更稀疏的 Netflix 子数据集上,SVD 算法也出现了过拟合现象,且过拟合的程度更大。

综合两个数据集上的实验结果,可以得出本文所提出的 wALS 算法的性能明显优于传统的 LRA、SVD、基于用户的 KNN 和基于项目的 KNN 算法,wALS 算法的抗稀疏性强,在更加稀疏的 Netflix 子数据集上这种优势更加明显。而且当特征数较小时,wALS 算法的性能随着特征数的增加而显著提高;当特征数增加到一定程度后即迅速趋于收敛,得到较理想的实验结果,如在本实验所采用的两个数据集上当特征个数达到 30 个特征后即趋于收敛。由于在较小的特征个数下即可以达到较理想的实验结果,而从定理 1 可知本算法的运算时间与特征个数的平方成正比,故本算法的运算效率很高。

4　结束语

本文在前人研究的基础上提出一种加权的迭代最小二乘法(wALS)来解决单类协同过滤问题,对明确的正例数据赋予权重 1,对于无法解释的混合数据赋予一个小于 1 大于 0 的正实数权重,进而分别在真实的 Netflix 子数据集和 MovieLens 数据集上实现本文所提出的算法;并以 AUC 值为性能评价标准,将其与传统的 LRA、SVD、基于用户的 KNN 和基于项目的 KNN 算法的性能进行了比较。实验结果表明本文所提出的 wALS 算法远优于其他几个经典的单类协同过滤推荐算法。在以后的工作中笔者还将考虑 wALS 算法的冷启动、并行化的问题,以及与其他算法结合提出性能更高的混合模型。

参考文献:

[1] 吴金龙. NetfLix Prize 中的协同过滤算法[D]. 北京:北京大学,2010.

[2] RICC F,ROKACH L,SHAPIRA B,*et al*. Recommender system handbook[M]. [S. l.]:Springer,2011.

[3] ADOMAVICIUS G,TUZHILIN A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions[J]. IEEE Trans on Knowledge and Data Engineering,2005,17(6):734-749.

[4] 罗辛,欧阳元新,熊璋,等. 通过相似度支持度优化基于 K 近邻的协同过滤算法[J]. 计算机学报,2010,33(8):1437-1445.

[5] 陈健,印鉴. 基于影响集的协作过滤推荐算法[J]. 软件学报,2007,18(7):1685-1694.

[6] ZHOU Yun-hong,WILKINSON D,SCHREIBER R,*et al*. Large-scale parallel collaborative filtering for the Netflix prize[C]//Proc of the 4th International Conference on Algorthmic Aspects in Information and Management. Berlin: Springer,2008:337-348.

[7] PATEREK A. Improving regularized singular value decomposition for collaborative filtering[C]//Proc of KDD Cup and Workshop. 2007: 39-42.

[8] RENDLE S,FREUDENTHALER C,GANTNER Z,*et al*. BPR: Bayesian personalized ranking from implicit feedback[C]//Proc of the 25th Conference on Uncertainty in Artificial Intelligence. Arlington: AUAI Press,2009:452-461.

[9] PRINZIE A, Van Den POEL D. Random forests for multiclass classification:random multinomial logit[J]. Expert Systems with Applications,2008,34:1721-1732.

[10] LI Xiao-li,YU P S,LIU Bing,*et al*. Positive unlabeled learning for data stream classification [C]//Proc of SIAM International Conference on Data Mining. [S. l.]:SIAM,2009:257-268.

[11] WARD G,HASTIE T,BARRY S,*et al*. Presence-only data and the EM algorithm[J]. Bio-metrics,2008,65(2): 554-563.

[12] ZHANG Cang-li, ZENG D, LI Jie-xun,*et al*. Sentiment analysis of Chinese documents:from sentence to document level[J]. Journal of the American Society for Information Science and Technology,2008,60(12):2474-2487.

[13] Netflix. Netflix prize[EB/OL]. http://www.netflixprize.com.

[14] SU Xiao-yuan,KHOSHGOFTAAR T M. A survey of collaborative filtering techniques[C]//Advances in Artificial Intelligence. New York: Hindawl Publishing Corp,2009:421-425.