

SCMF:一种融合多源数据的软约束矩阵分解推荐算法*

满彤, 沈华伟, 黄俊铭, 程学旗

中国科学院计算技术研究所, 北京, 100190

E-mail: mantong@software.ict.ac.cn

摘要: 数据稀疏是推荐系统面临的主要挑战之一。近年来, 多源数据融合为解决数据稀疏问题提供了新思路。然而, 现有方法大多假设对象在不同数据源中具有相同的表示, 这种硬约束方式无法刻画对象在不同数据源中的差异性。本文提出一种基于软约束矩阵分解的推荐算法, 通过约束不同数据源中对象表示之间的距离, 能够同时刻画同一对象表示的共性及其在不同数据源中的差异性。在 Netflix 和 MovieLens 数据集上的实验表明, 本文提出的软约束矩阵分解算法在准确率方面优于现有的单数据源推荐算法和多源数据硬约束融合推荐算法, 可以有效解决推荐系统面临的数据稀疏问题。

关键词: 协同过滤; 推荐系统; 异质网络

SCMF: A Matrix Factorization Model With Soft Constraint for Multi-Source Heterogeneous Network

Tong Man, HuaWei Shen, JunMing Huang, Xueqi Cheng

Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190

E-mail: mantong@software.ict.ac.cn

Abstract: Data sparsity is a big challenge for recommender systems. In recent years, the integration of data from different contexts provides a promising way to combat the problem of data sparsity. However, existing methods for data integration mostly assume that the representation of a single user/item is the same across different contexts. This assumption is a kind of hard constraint and is incapable of depict the distinct characteristics of different contexts. In this paper, we propose a matrix factorization model with soft constraint that the difference between the representations of a single user/item is minimized together with the error function of matrix factorization model. Experiments on two widely-used datasets, i.e., Netflix and MovieLens, demonstrate that the proposed model outperforms the state-of-the-art models, including the matrix factorization model separately for each context and the collective matrix model with a hard constraint. The proposed model performs particularly well on the case where the data is sparse in one context but is not sparse in others. This indicates that the proposed model is effective at alleviating the problem of data sparsity.

Keywords: collaborative filtering; recommender system; heterogeneous entropy model

1 引言

互联网上规模快速增长的数据带来了严峻的信息过载问题, 推荐系统在该背景下应运而生^[1]。通过分析用户的历史行为数据, 推荐系统能够根据用户兴趣向用户推荐其感兴趣的对象, 例如电影、书籍、商品等。推荐系统的有效性很大程度上取决于历史行为数据, 然而用户往往不能形成足够多的数据。例如, Amazon¹上购买家具的用户可能在未来几年

*本课题得到国家自然科学基金(No. 61202215, No. 61232010, No. 60933005)和信息网络安全公安部重点实验室开放课题项目的资助

¹www.amazon.com

内不会再产生家具购买行为；一个刚刚进入豆瓣¹的新用户可能只会给很少的对象打分。另外，大部分用户的兴趣都集中在少数热门对象上，形成长尾效应^[2]。图1是MovieLens 10M²数据集中用户和电影的打分次数分布图，从图中可以看出，大部分用户的打分次数以及大多数电影获得的打分次数都很少，这类数据稀疏问题是推荐算法面临的主要挑战之一。

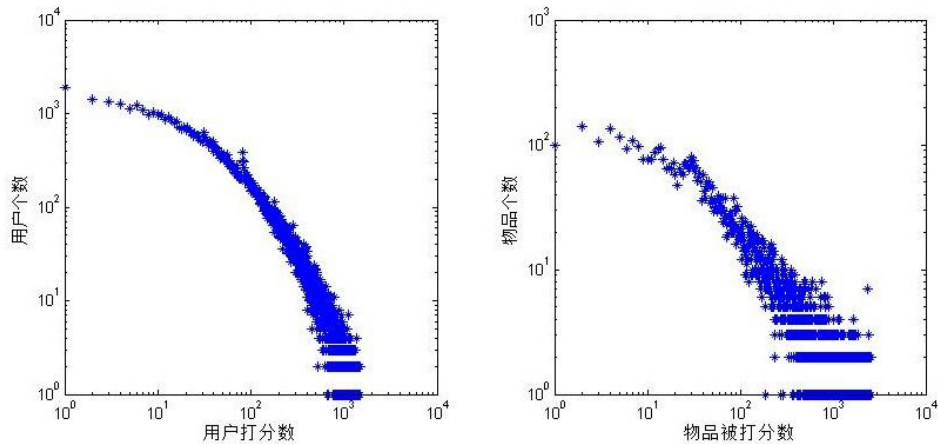


图1 MovieLens 数据集中用户打分次数分布与电影被打分次数的分布

Fig.1 Users' and items' rating distribution in MovieLens dataset

为了解决推荐算法面临的数据稀疏问题，研究人员开始尝试融合多个不同的数据源。对于推荐算法而言，不同的数据源通常呈现为由用户和对象构成的网络。例如，豆瓣网中的同一个用户，可能同时会对电影、音乐、书籍三类对象打分，在用户和对象之间形成三个不同类型的打分网络，用户同时出现在这三个网络中；类似地，一部电影在豆瓣网上得到豆瓣用户的一系列评分，同时会得到时光网³用户的一系列评分，豆瓣网和时光网就扮演着不同的数据源，形成用户和电影之间的两个打分网络。

多源数据融合的推荐算法，其基本思想是将从一个数据源的打分网络中学习到的有关用户和对象的知识，应用到了另外一个数据源中，综合利用多个数据源的知识解决数据稀疏问题，从而来提高推荐算法的准确率。特别是，一些用户和物品在某些网络中的评分数目较少，而在另外一些网络中的评分数目较多，通过整合这些不同数据源的打分网络，建立一个多数据源融合的模式，使各个数据源的信息彼此补充，解决数据稀疏问题。

本文基于推荐系统中广泛使用的矩阵分解算法，通过对不同数据源网络中的用户/对象的表示向量进行约束，提出了一种基于约束矩阵分解的推荐算法。Netflix⁴和MovieLens数据集上的实验表明，本文提出的软约束矩阵分解算法在准确率方面优于现有的单数据源推荐算法和多源数据硬约束融合推荐算法，可以有效解决推荐系统面临的数据稀疏问题。

本文的组织结构如下：第二章介绍相关工作；第三章介绍基本的矩阵分解模型以及CMF模型。第四章介绍我们提出的SCMF模型以及学习算法。实验结果与分析在第五章给出，最后一章总结我们的工作以及对未来的工作进行展望。

¹www.douban.com

²www.grouplens.org/node/73

³www.mtime.com

⁴www.netflix.com

2 相关工作

协同过滤(Collaborative Filtering)算法是推荐系统中使用广泛且有效的算法^[1]。协同过滤算法主要可以分为两种：一种是基于邻居(Neighborhood-Based)的算法，一种是基于模型(Model-Based)的算法。基于邻居的算法^[2]主要通过寻找相似的用户或者对象来完成推荐。基于模型的算法通过利用用户在对象上的历史信息学习出一个模型，进而利用模型来预测用户未来可能会喜欢的对象。矩阵分解模型^[3,4](Matrix Factorization)是近年来非常流行一种基于模型的推荐算法。矩阵分解模型通过分解用户和对象间的打分矩阵，利用得到的用户隐因子矩阵和物品隐因子矩阵来预测缺失的分数。矩阵分解模型假设一个用户对一个对象的打分是由该用户和对象的隐因子向量相互作用得到的，其中最常用的相互作用假设就是向量点积。Ruslan^[5]对矩阵分解做了很好的概率化的解释，进一步提高了矩阵分解推荐算法的准确率。

推荐系统面临的一个挑战是数据稀疏问题^[6]。用户的打分集中集中在少数物品上，同时少数用户给出了大部分的打分，形成长尾效应^[7]。为了解决数据稀疏问题，Ma^[8]等人引入用户之间的社交关系来指导模型的学习过程，Noam^[9]等人考虑引入对象间的层次关系来对模型中的参数进行控制。这些方式都是通过引入一些打分网络之外的信息，来缓解数据稀疏问题。

然而，上述工作均在用户和对象间的单个网络上进行的，真实情况下用户和对之间可以形成多个网络，特别是当数据来自多个数据源时。近年来，在推荐算法方面，涌现出了大量的基于多源异质网络的研究工作。Berk^[10]等人提出了基于邻居的多源的协同过滤算法，通过传递不同网络中用户和对象的相似性来缓解单个网络中用户或对象的打分稀疏问题。Pan^[11]等人提出了一个 Coordinate System Transfer (CST) 模型，基于迁移学习的想法，通过在一个较为稠密的用户物品评分矩阵投射到一个子空间坐标系，然后将该坐标系作为信息迁移到一个稀疏矩阵中，从而缓解稀疏性问题。Singh^[12]等人提出了协同矩阵分解模型(CMF)，CMF同时分解多个网络的打分矩阵，当同一个节点(用户或物品)出现在多个网络中时，该节点在所有网络中的隐因子向量都是一致的。CMF模型考虑到了相同节点在不同的网络中的同质性，而忽略了其异质性。例如，受到用户群体，网站广告策略的影响，一部电影在两个电影评分网站中的行为会存在着差异性；同样，同一个用户在不同的网络中的行为也会有一些差异性，在一个网站中很活跃的用户在另一个网站中可能只是一个很少发布信息的观看者。

3 基于约束矩阵的多源异质网络推荐算法

推荐系统旨在预测用户对未知物品的偏好程度，可以形式化为矩阵填充问题。以一个 $M \times N$ 的偏好矩阵 R 表示 M 个用户在 N 个物品上的偏好程度，通常以用户对物品的打分表示。通过对偏好矩阵 R 的部分观测推断 R 的全部单元的值。在真实场景中，每个用户往往只会对一部分的物品进行评分，因此 R 通常是稀疏的。

定义矩阵 $U \in \mathbb{R}^{K \times M}$ 表示用户的隐因子矩阵，第 i 列 U_i^T 表示用户 i 的在 K 维隐空间中的偏好向量。 $V \in \mathbb{R}^{K \times N}$ 表示物品的隐因子矩阵。则用户 i 对物品 j 的打分应该服从以 $U_i^T V_j$ 为均值的高斯分布

$$R_{ij} \sim N(U_i^T V_j, \sigma^2) \quad (1)$$

同时隐因子矩阵 U 和 V 应满足均值为 $\mathbf{0}$ 的高斯分布

$$U \sim N(\mathbf{0}, \sigma_u^2) \quad (2)$$

$$V \sim N(\mathbf{0}, \sigma_v^2) \quad (3)$$

对 R 的已观测部分作最大似然估计可以推断隐因子矩阵 U 和 V 的值

$$[U, V] = \underset{U, V}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N I_{ij} (R_{ij} - U_i^T V_j)^2 + \frac{\lambda_u}{2} \|U\|_F^2 + \frac{\lambda_v}{2} \|V\|_F^2 \quad (4)$$

其中 I_{ij} 是一个指示函数，当用户 i 对物品 j 有打分信息时 I_{ij} 的值为1，反之为0。约束系数 $\lambda_u, \lambda_v > 0$ ， $\|\cdot\|_F$ 表示 Frobenius 范数。根据推断的 U 和 V 的值可以求得未知的打分的期望。矩阵分解的概率图模型如图2中所示：

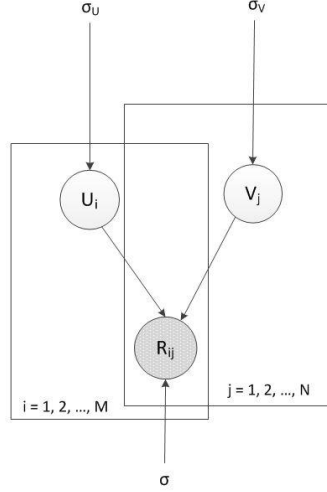


图2 矩阵分解的概率图模型

Fig.2 Probability graph model of matrix factorization

当观测到多个数据时，可以简单地对每个数据源设立一组隐因子矩阵 U 和 V 分别求解。例如对于两个数据源中的打分矩阵 R_1 和 R_2 ，分别寻求隐因子矩阵 U_1, U_2, V_1, V_2 ，问题可以形式化为：

$$[U_1, U_2, V_1, V_2] = \underset{U_1, U_2, V_1, V_2}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{M_1} \sum_{j=1}^{N_1} I_{ij} (R_{1,ij} - U_{1,i}^T V_{1,j})^2 + \frac{\lambda_u}{2} \|U_1\|_F^2 + \frac{\lambda_v}{2} \|V_1\|_F^2 + \frac{1}{2} \sum_{i=1}^{M_2} \sum_{j=1}^{N_2} I_{ij} (R_{2,ij} - U_{2,i}^T V_{2,j})^2 + \frac{\lambda_u}{2} \|U_2\|_F^2 + \frac{\lambda_v}{2} \|V_2\|_F^2 \quad (5)$$

可以看到，其求解过程相当于分别求解两次矩阵分解。在这一模型中，两个数据源中的用户和物品完全独立，称为 **Separate Matrix Factorization**，简称 **SMF**。

如果某些用户或物品同时出现在多个数据源中，这些重叠用户或物品在各个数据源中的隐因子向量应该具有一定的相关性。通过对这些相关性的分析和建模，可以改进估计的准确性（我们以物品重叠为例展开本文接下来的讨论。根据对称性，这一讨论可以应用于用户重叠的场景）。例如，一个物品在某一数据源中的可用数据非常稀疏，难以准确地估计它的隐因子向量，可以用它在另一数据源中的向量辅助估计。协同矩阵分解模型(**CMF**)将这一相关性表示为硬约束下的一致性，要求一个物品在所有数据源中的隐因子向量都是一致的。考虑具有共同物品的两个数据源，问题可以形式化为：

$$[U_1, U_2, V] = \underset{U_1, U_2, V}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{M_1} \sum_{j=1}^N I_{ij} (R_{1,ij} - U_{1,i}^T V_j)^2 + \frac{\lambda_u}{2} \|U_1\|_F^2 + \frac{1}{2} \sum_{i=1}^{M_2} \sum_{j=1}^N I_{ij} (R_{2,ij} - U_{2,i}^T V_j)^2 + \frac{\lambda_u}{2} \|U_2\|_F^2 + \frac{\lambda_v}{2} \|V\|_F^2 \quad (6)$$

这种情况下 **CMF** 的概率图模型如图3所示。然而，这一约束没有考虑到节点的异质性。事实上，同一物品在不同数据源中可能有不同的隐因子向量。例如一个用户在社交网站上和音乐网站上可能表现出不同的兴趣分布。

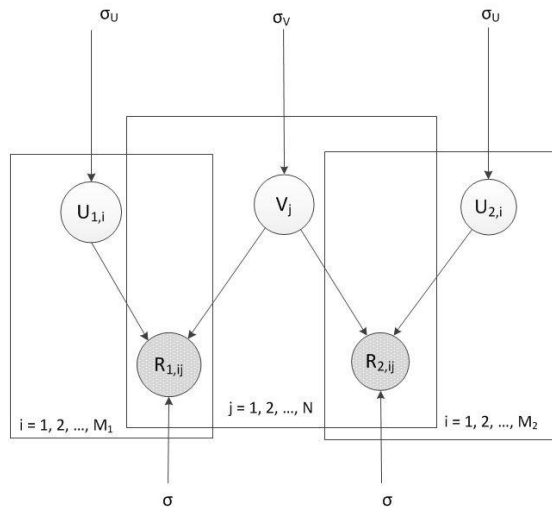


图 3 CMF 的概率图模型
Fig.3 Probability graph model of CMF

基于以上讨论, 我们提出了一个基于软约束的协作矩阵分解模型(Soft-Constraint Matrix Factorization Model, SCMF)。我们同时建模重叠物品在多个数据源中的隐因子向量的同质性和异质性。我们为一个物品在不同数据源中定义不同的隐因子向量以保证其在不同数据源中的异质性, 同时添加一个距离函数作为约束, 限制同一物品的多个隐因子向量保留一定程度的相似性以保证了节点的同质性。同样考虑具有共同物品的两个数据源, SCMF 的概率图模型如图 4 所示。

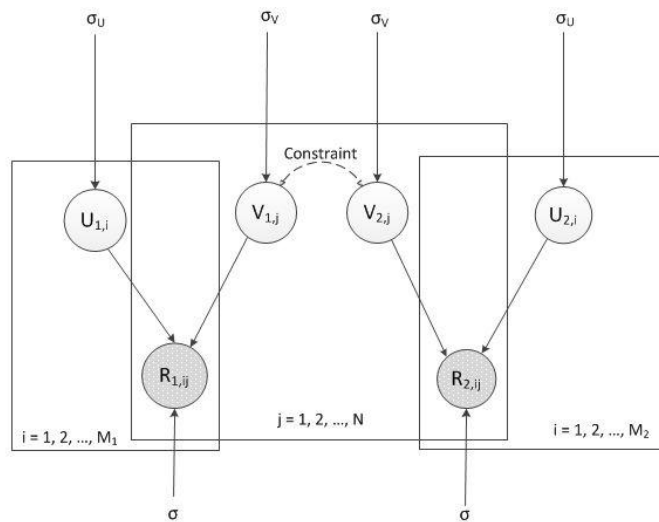


图 4 SCMF 的概率图模型
Fig.4 Probability graph model of SCMF

我们的方法可以抽象为如下的目标函数:

$$[U_1, U_2, V_1, V_2] = \arg \min_{U_1, U_2, V_1, V_2} \frac{1}{2} \sum_{i=1}^{M_1} \sum_{j=1}^N I_{ij} (R_{1,ij} - U_{1,i}^T V_{1,j})^2 + \frac{\lambda_u}{2} \|U_1\|_F^2 + \frac{\lambda_v}{2} \|V_1\|_F^2 + \frac{1}{2} \sum_{i=1}^{M_2} \sum_{j=1}^N I_{ij} (R_{2,ij} - U_{2,i}^T V_{2,j})^2 + \frac{\lambda_u}{2} \|U_2\|_F^2 + \frac{\lambda_v}{2} \|V_2\|_F^2 + \alpha \sum_{l=1}^N \text{dist}(V_{1,l}, V_{2,l}) \quad (7)$$

目标函数中的前两项和传统的矩阵分解算法一致，希望针对每个源数据网络的打分矩阵分解出来的因子矩阵能够较好的拟合真实的情况；同时我们引入第三项，控制出现在多个网络中的用户或者物品其隐因子向量在不同的网络中依然有一定的相似性，其中参数 α 控制对相似性的约束程度，其值越大对相似的约束越大。当 α 取0时，我们的方法与SMF算法一致；当 α 取 $+\infty$ 时，相当于约束所有用户和物品在所有的源网络中的隐因子向量都一致，我们的模型与CMF模型一致。 $\text{dist}(*,*)$ 函数度量两个向量之间的距离，在我们的模型中选取点积的负值作为距离函数。

$$\text{dist}(U_{1,i}, U_{2,i}) = -U_{1,i} \cdot U_{2,i} \quad (8)$$

考虑到用户和物品都可能出现在不同的源网络中，我们的模型可以推广为以下形式，其中源网络的个数为 L ，用户个数为 M ，物品的个数为 N 。

$$[U_1, \dots, U_L, V_1, \dots, V_L] = \arg \min_{U_1, \dots, U_L, V_1, \dots, V_L} \frac{1}{2} \sum_{s=1}^L \sum_{i=1}^M \sum_{j=1}^N I_{ij} (R_{s,ij} - U_{s,i}^T V_{s,j})^2 + \frac{1}{2} \sum_{s=1}^L \frac{\lambda_u}{2} \|U_s\|_F^2 + \frac{\lambda_v}{2} \|V_s\|_F^2 + \alpha \sum_{i=1}^M \sum_{s_1=1}^L \sum_{s_2 \neq s_1}^L \text{dist}(U_{s_1,i}, U_{s_2,i}) + \alpha \sum_{i=1}^N \sum_{s_1=1}^L \sum_{s_2 \neq s_1}^L \text{dist}(V_{s_1,i}, V_{s_2,i}) \quad (9)$$

4 模型推断

我们通过随机梯度下降算法推断模型中的隐因子矩阵 U 和 V 。在更新过程中，我们依次遍历每个打分矩阵中的每一个打分对，并且更新与改打分对相关的用户和物品的隐因子向量。对于式(9)中的目标函数中的隐因子向量，我们计算目标函数关于它的偏导并沿着偏导方向更新

$$U_{s,i} \leftarrow U_{s,i} - \gamma \frac{\partial L}{\partial U_{s,i}} = U_{s,i} + \gamma (R_{s,ui} - U_{s,i}^T \cdot V_{s,j} - \lambda_u \cdot U_{s,i} + \alpha \sum_{l \neq s} U_{s,i}) \quad (10)$$

$$V_{s,j} \leftarrow V_{s,j} - \gamma \frac{\partial L}{\partial V_{s,j}} = V_{s,j} + \gamma (R_{s,ui} - U_{s,i}^T \cdot V_{s,j} - \lambda_v \cdot V_{s,j} + \alpha \sum_{l \neq s} V_{s,j}) \quad (11)$$

其中 γ 为学习的步长， (i, j) 表示源网络 s 中的一个打分对。

表 1 SCMF 学习算法

Tab.1 Learning algorithm of SCMF

SCMF 学习算法:

$$[U_1, U_2, \dots, U_L, V_1, V_2, \dots, V_L] = \text{SCMF}(R_1, R_2, \dots, R_L, \lambda_u, \lambda_v, \alpha, \gamma, K)$$

输入: S 个打分矩阵 R^1, R^2, \dots, R^S

正则约束系数 λ_u, λ_v

相似度控制参数 α

迭代次数 K 步长 γ

输出: 用户和物品在每个源网络中的隐因子矩阵

$$U_1, U_2, \dots, U_L, V_1, V_2, \dots, V_L$$

```

for l=1 to K do
  for s= 1 to L do                                (遍历所有的源打分矩阵)
    for each pair (i,j) in Rs do
      Update Us,i,Vs,j
    end for
  end for
end for

```

5 实验

5.1 数据集

我们使用 MovieLens 和 Netflix 两个电影评分数据作为我们的实验数据集。根据电影的标题和年份信息我们能够确定两个数据集中的同一部电影，整合两个数据集中的相同电影，我们能够构建一个电影评分的多源异质网络。数据集的信息描述在表 2 中。

表 2 电影评分数据信息

Tab.2 Information of our movie rating data

数据集	用户数	电影数	评分数	稀疏性
MovieLens	69,878	10,677	10,000,054	1.34%
Netflix	480,189	17,769	100,479,586	1.18%
MovieHetero	548,971	5,871	87,820,281	2.72%

5.2 实验设计描述

我们使用 C 语言实验我们的算法。我们的实验在一台多核机器上的一个单核上运行，机器的 CPU 为 Intel(R) Xeon(R) E5620, 2.40GHz，内存为 16G。我们实验在运行过程中会用到大约 2G 的内存。

我们实验中采用五折交叉验证(5-fold cross validation)。每一次验证过程中，数据会被切分，其中 80%的数据作为训练数据(training data)，20%的数据作为测试数据(test data)。

我们采用 RMSE(root mean square error)作为我们的评价指标，其定义如下

$$RMSE = \sqrt{\frac{\sum_{(u,v) \in R_{test}} (r_{u,v} - \hat{r}_{u,v})^2}{|R_{test}|}} \quad (12)$$

我们采用 SMF、CMF 作为基准算法，与 SCMF 分别求解并估计未知打分，以比较 SCMF 的估计准确性。

5.3 实验结果及分析

我们的实验数据是由两个电影源数据组成的一个多源异质网络，在训练阶段，我们在整个网络上学习参数，在评价阶段我们分别评价在各自的源数据上的预测效果。我们选择隐因子向量的维度为 30，用户和物品的正则约束系数均为 0.007，学习步长为 0.005，迭代次数为 30 次。

首先看我们相似性约束系数 α 与准确性的关系。从图 5 中可以看到，以 RMSE 度量的预测准确度随着 α 呈现先减后增的趋势， α 取值为 0.1 的时候两个数据集都能够得到最低的预测误差，过大或过小的 α 均不合适。如前所述， α 度量了我们对于同一物品在不同数据集上相似性的约束程度。这表明一个物品在多个数据集中的隐因子向量既有同质性又有异质性，单纯考虑异质性($\alpha = 0$)或单纯考虑同质性($\alpha = +\infty$)均会导致性能下降。

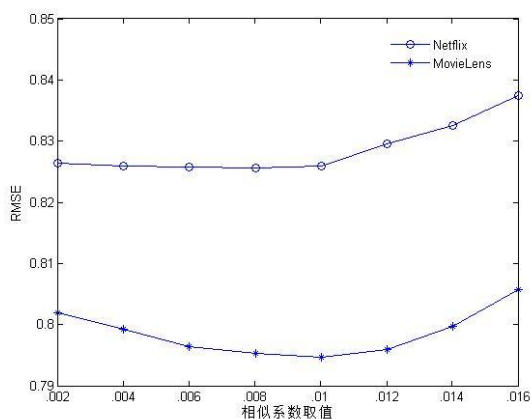


图 5 RMSE 结果与相似系数的关系

Fig.5 RMSE result versus similarity parameter α

接下来我们来看我们的方法相对于基准算法的性能。图 6 是我们的算法与 SMF 算法以及 CMF 算法的比较,可以看到,在 MovieLens 数据集上我们的算法相对于 SMF 有约 1.34%的提升(0.8054 \rightarrow 0.7946),相对于 CMF 有约 0.54%的提升(0.7984 \rightarrow 0.7946)。在 Netflix 上我们的算法改进较小,这是由于 Netflix 数据集集较为稠密,大部分物品受到稀疏性影响较小,传统矩阵分解算法已经可以给出准确的估计。我们的算法作为缓解稀疏性困扰的方法,在 Netflix 这样的稠密数据集上对准确度的提升有限。

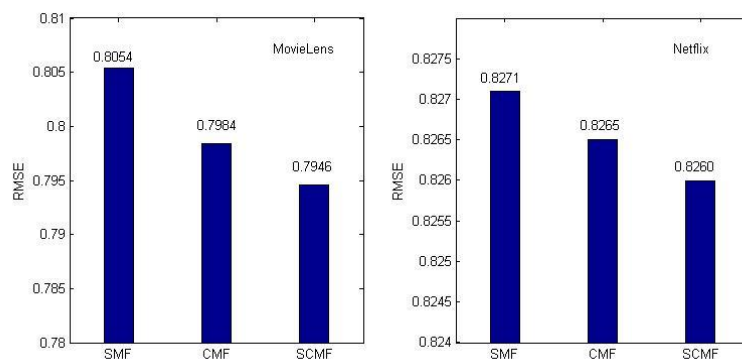


图 6 不同算法的性能比较(两个数据集上的整体表现)

Fig.6 Comparing the performance of three methods on the whole dataset

图 7 中分析了不同算法在不同的稀疏物品上的表现。我们将网络中的物品按照稀疏程度分为几类,分别比较几种算法在不同稀疏程度的物品上的预测性能。可以看到,我们的

算法在稀疏的物品上的性能要远远的好于其它的 baseline 算法，例如在 MovieLens 数据集上，对于那些打分次数在 10 次以下的物品，我们的算法相对于 SMF 算法有近 9% 的提升 (0.9596 \rightarrow 0.8750)，相对于 CMF 算法也有近 3% 的提升 (0.9010 \rightarrow 0.8750)。对于那些打分次数的比较高的物品，我们可以看到三个算法的性能类似。因此相对于其它的两个 baseline 算法，我们的算法较好的缓解了数据的稀疏性问题。

从实验结果中可以看到，我们的模型刻画出了多源异质网络中处于多个网络中的相同节点的同质性（相对于 CMF 模型），又较好的抓住了相同的节点处于多个异质源网络里的异质性（相对于 SMF 模型）。

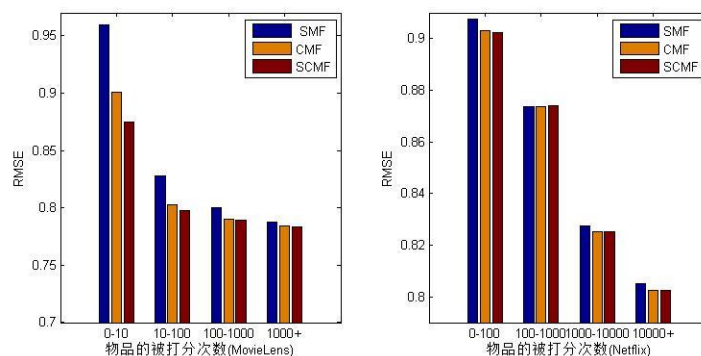


图 7 不同算法的性能比较(不同稀疏性的物品上的表现)

Fig.7 Comparing the performance of three methods by the sparsity of the items

6 总结和未来工作

本文中提出了一个基于多源异质网络的推荐算法，在矩阵分解模型的基础上，目标函数中加入了约束条件使得出现在多个源网络中的用户和物品的隐因子向量具有一定的相似性。通过在数据集上的实验发现我们的算法在推荐精度上要优于传统的算法，尤其是在稀疏性物品上的推荐更是有非常大的改进。

然而，我们的工作中还存在一些需要改进的地方。一个是控制相似程度的参数的选取，通过调参选择结果最优的方式选择较为费时，可以考虑将模型概率化，从贝叶斯模型的角度使得能够不用调节参数而在学习的过程中自适应的找到最好的参数。另外，我们的模型中通过目标函数的约束项来保证用户和物品在不同网络中的同质性，而直接分别建模用户和物品的同质性因子和异质性因子似乎更加的直接。在未来的工作中，我们会尝试解决这些问题。

参 考 文 献

- [1] Adomavicius Gediminas and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." *Knowledge and Data Engineering, IEEE Transactions on* 17.6 (2005): 734-749.

- [2] Sarwar Badrul, et al. "Item-based collaborative filtering recommendation algorithms." *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001.
- [3] Koren Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008
- [4] Koren Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 42.8 (2009): 30-37.
- [5] Salakhutdinov Ruslan, and Andriy Mnih. "Probabilistic matrix factorization." *Advances in neural information processing systems* 20 (2008): 1257-1264.
- [6] Herlocker Jonathan L., et al. "Evaluating collaborative filtering recommender systems." *ACM Transactions on Information Systems (TOIS)* 22.1 (2004): 5-53.
- [7] Park Yoon-Joo, and Alexander Tuzhilin. "The long tail of recommender systems and how to leverage it." *Proceedings of the 2008 ACM conference on Recommender systems*. ACM, 2008.
- [8] Ma Hao, Irwin King, and Michael R. Lyu. "Learning to recommend with social trust ensemble." *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009.
- [9] Koenigstein Noam, Gideon Dror, and Yehuda Koren. "Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy." *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 2011.
- [10] Berkovsky Shlomo, Tsvi Kuflik, and Francesco Ricci. "Cross-domain mediation in collaborative filtering." *User Modeling 2007*. Springer Berlin Heidelberg, 2007. 355-359.
- [11] Pan Weike, et al. "Transfer learning in collaborative filtering for sparsity reduction." *Proceedings of the 24rd AAAI Conference on Artificial Intelligence*. 2010.
- [12] Singh, Ajit P. and Geoffrey J. Gordon. "Relational learning via collective matrix factorization." *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008.