一种基于局部结构的改进奇异值分解推荐算法

方耀宁*① 郭云飞① 丁雪涛② 兰巨龙①
①(国家数字交换系统工程技术研究中心 郑州 450002)
②(清华大学软件学院 北京 100084)

摘 要：基于奇异值分解(Singular Value Decomposition, SVD)的推荐算法，在预测准确性、稳定性上具有明显优势，但在用随机梯度下降法求解过程中误差下降速度逐渐变慢、迭代次数较多，这极大限制了其在实际项目中的应用。针对这个问题，该文利用评分矩阵的差分矩阵来表征局部结构信息，并作为新的目标函数来优化 SVD 推荐算法。在 MovieLens 和 Netflix 数据集合上的实验结果表明：与经典 SVD 算法相比，该优化算法能够用更少的迭代次数得到更准确的预测结果；与当前的其他算法相比，该优化算法在预测准确性上仅次于 SVD++，在训练时间上具有显著优势。

关键词：信息处理；推荐系统；协同过滤；奇异值分解(SVD)；局部结构
中图分类号：TP393 **文献标识码**： A **文章编号**：1009-5896(2013)06-1284-06
DOI: 10.3724/SP.J.1146.2012.01299

An Improved Singular Value Decomposition Recommender Algorithm Based on Local Structures

Fang Yao-ning① Guo Yun-fei① Ding Xue-tao② Lan Ju-long①
①(*National Digital Switching System Engineering and Technological R&D Center, Zhengzhou* 450002, *China*)
②(*School of Software, Tsinghua University, Beijing* 100084, *China*)

Abstract: Recommender algorithms based on Singular Value Decomposition (SVD) performs better in prediction accuracy and stability, while the speed of error descent slows down gradually and the iteration number is huge when applying Stochastic Gradient Descent (SGD) to SVD , which greatly limits its usage in actual projects. To address this problem, the difference matrix of the rating matrix is used to represent the local structures and to optimize SVD as a new target function. Experiments on MovieLens and Netflix dataset show that the improved SVD algorithm can obtain better performances with fewer iterations compared to classic SVD. When compared with other state-of-the-art recommender algorithms, the proposed algorithm is only second to SVD++, however within far less training time than SVD++.

Key words: Information processing; Recommender system; Collaborative filtering; Singular Value Decomposition (SVD); Local structure

1 引言

互联网的飞速发展将人类带入了信息膨胀的时代，从海量信息中获取所需内容却成为了一件复杂的事情——信息过载(information overload)。与传统的搜索引擎不同，推荐系统利用数据挖掘、人工智能等技术能够主动帮助人们过滤信息，提供个性化的服务。在过去的十多年里，商业利益驱动的电子商务和用户体验驱动的社交网络极大促进了推荐系统的研究与发展。推荐系统不仅可以提供娱乐、生活方面的服务推荐，如电影推荐、好友推荐；而且可以提供更加专业化的推荐，如科技论文推荐、专利技术推荐[1−3]。

推荐系统主要分为 3 大类：基于内容的(content-based)推荐系统、协同过滤(collaborative filtering)推荐系统和混合(hybrid approaches)推荐系统[2,3]，其他还有基于谱分析、扩散的推荐方法等[1]。基于内容的协同过滤推荐系统侧重用户(user)和项目(item)的内容信息分析，比如年龄、性别、教育背景等个人信息会暗示用户的兴趣，色彩、种类、价格等信息会暗示商品面向的消费群体；协同过滤算法从用户以往的行为信息中挖掘用户的兴趣和偏好，侧重的是用户与项目之间的二元关系。协

2012-10-12 收到，2013-01-16 改回
国家 973 计划项目(2012CB315901)和国家 863 计划项目(2011AA01A103)资助课题
*通信作者：方耀宁 fyn07@163.com

同过滤推荐系统结构简单，不需要具备特定领域内的知识，能处理不易进行内容分析的艺术品、音乐等，是目前最为成功的推荐系统[1−3]。

协同过滤算法主要分为基于内存(memory-based)和基于模型(model-based)两类：基于内存的算法建立在相似用户对同一项目的评分相似、相似项目被同一用户评分也相似的假设之上，基本方法是寻找相似的用户/项目，根据相似的用户/项目对目标进行预测, 如 K 近邻算法(K-Nearest Neighbor, KNN) ;基于模型 算法首先利用已知信息训练参数、建立某种模型来表示评分规律，然后利用该模型来进行推荐，如基于矩阵分解、基于网络、概率模型算法等[2−4]。

一般来说，基于内存的算法在原理上更简单，选择合适的相似性计算方法是关键，在评分矩阵较密集时能够达到合理的预测准确性，但稀疏性、冷启动问题非常严重，在评分矩阵密度较低时性能非常差。基于模型的算法相对比较复杂，多数建立在提取用户/项目潜在特征的基础上, 能够较好解决稀疏性问题，其中最有代表性的是基于矩阵分解的推荐算法[3,4]。

奇异值分解(Singular Value Decomposition, SVD)是一种常用的降低数据维度的矩阵分解算法，隐含语义分析(Latent Semantic Analysis, LSA)也是基于 SVD[3]。SVD 算法最先由文献[5]用于协同过滤推荐，在 Netflix Prize 中表现出了优秀的预测准确性和稳定性, 迅速成为最流行的推荐算法之一[4,6]。文献[7]提出了 SVD2 算法分别给用户和项目添加偏置(bias), 为了减小参数个数把用户特征看作是项目特征的函数, 进一步提出了 NSVD 和 NSVD2 算法。文献[4,6]提出了利用含蓄反馈信息的 SVD++算法，在只有明确评分信息的情况下把用户评分过的项目作为一种含蓄信息, timeSVD++进一步考虑用户和项目特性会随时间变化。

非负矩阵分解(Non-negative Matrix Factorization, NMF)与概率矩阵分解(Probabilistic Matrix Factorization, PMF)也是推荐系统中常用的矩阵分解方法。NMF 是将非负矩阵分解成两个低秩非负矩阵的乘积，由于分解出来的矩阵元素都是非负值, 能够较好地解释各个特征的物理意义[8]。PMF 假设了用户和项目的特征向量都符合高斯分布，把用户评分看作是一个概率问题来解决[9]。文献[10]证明了 PMF 与概率主成分分析(Probabilistic PCA)是等价的，进而提出了用高斯过程来对 PMF 模型进行非线性的扩展(Non-linear PMF)。文献[11]使用马尔可夫链蒙特卡罗算法训练 BPMF(Bayesian Probabilistic Matrix Factorization)，取得了比较理想的推荐效果。

基于 SVD 的推荐算法多使用梯度下降法进行求解，在求解过程中误差下降速度越来越慢，而且需要很多的迭代次数和训练时间[3,4,6]。为了解决 SVD 算法存在的这些问题，本文提出利用评分矩阵的差分矩阵来表征局部信息，并作为一个新的目标函数，这样就把 SVD 变成了一个多目标优化问题。然而，评分矩阵往往非常稀疏，差分矩阵不能直接获得，使用相邻项目的平均差信息构造近似差分矩阵。在多目标优化求解过程中采用分级求解的思想，先根据经典的目标函数求出近似最优解，然后根据新的目标函数在近似最优解附近(可行域)寻找最优解。在 MovieLens 1 M 数据集合和部分 Netflix 数据集合上的实验表明, 改进方法能够用比 SVD 更少的训练时间，获得与 SVD++算法接近的性能；在特征维度为 10 时，能够降低约 0.01 的预测误差 MAE，这种优势随着特征维度 K 的增加仍能保持；与其他当前主流推荐算法相比，在预测准确性上仅次于 SVD++，但训练时间大为缩减。

2　SVD 推荐算法模型

考虑一个具有 m 个用户和 n 个项目的推荐系统，评分矩阵为 R , R_{ij} 表示用户 i 对项目 j 的评分。$R_{ij}=0$ 表示用户 i 没有对项目 j 进行评分。推荐算法的目的是对评分矩阵中 $R_{ij}=0$ 的部分进行预测。SVD 推荐算法是根据已知的评分信息将矩阵 R 分解成为

$$R = X\Sigma Y^{\mathrm{T}} = X\Sigma^{1/2}(Y\Sigma^{1/2})^{\mathrm{T}} \tag{1}$$

其中 X , Y 为正交矩阵， Σ 是对角矩阵。这样就可以得到 R 的低阶逼近 $R_k = X_k\Sigma_k Y_k^{\mathrm{T}}$, 这是一种降低数据维度的方法。SVD 算法通常加入正则化约束项来防止过拟合，实际上是求解用户特征矩阵 $M = X_k\Sigma_k^{1/2}$ 和项目特征矩阵 $N = Y_k\Sigma_k^{1/2}$ ，使得 $E = \left\|MN^{\mathrm{T}} - R\right\|^2 + \lambda\left(\|M\|^2 + \|N\|^2\right)$ 最小, 其中 λ 为正则化参数[4,7]。用户特征矩阵 M 和项目特征矩阵 N 都是 K 维的矩阵， K 为特征空间维度。求解方法一般有两种：随机梯度下降法(Stochastic Gradient Descent, SGD)和交替最小二乘法(Alternating Least Squares, ALS)。一般来说，前者相对于后者而言更容易实现，而且运算速度快，应用最为广泛[4,6]。

使用 SGD 的 SVD 推荐算法步骤[7]：

步骤 1　对于训练集合中所有的用户和项目组合 (i, j) ：

(1)计算用户 i 对项目 j 的评分： $\hat{R}_{ij} = M_i N_j^{\mathrm{T}}$ ；

(2) 计算与真实评分之间的误差：$e_{ij} = R_{ij} - \widehat{R}_{ij}$；

(3)对于用户 i 和项目 j 所有特征 $f(1 \le f \le K)$，用梯度下降法进行更新：

$$\left.\begin{aligned} M_{if} &\leftarrow M_{if} + \mathrm{lr} \cdot (e_{ij} \cdot M_{if} - \lambda N_{jf}) \\ N_{jf} &\leftarrow N_{jf} + \mathrm{lr} \cdot (e_{ij} \cdot N_{jf} - \lambda M_{if}) \end{aligned}\right\}, \ \mathrm{lr}\text{为学习速率} \quad (2)$$

步骤 2 判断是否满足终止条件，否则迭代步骤 1。

步骤 3 对测试集合中所有的用户和项目组合 (i, j)，计算用户 i 对项目 j 的评分：$\widehat{R}_{ij} = M_i N_j^{\mathrm{T}}$。

其中，步骤 1，步骤 2 为训练过程，是一个 3 层循环的嵌套；步骤 3 为预测评估过程。一般情况下，将用户特征矩阵 M 和项目特征矩阵 N 初始化为随机值，设定固定的学习速率 lr 和正则化参数 λ。迭代终止条件一般有两种：平均误差或者两次平均误差的差值小于某个阈值时则停止训练；从已知样本中随机抽取一个小的测试集合，当测试集合上出现过拟合时则停止训练[7]。为了防止无限迭代，也可以根据经验设定最大迭代次数。

3 结构化 SVD 推荐算法

使用梯度下降法进行奇异值分解的过程中并没有考虑到评分矩阵的局部结构特性，是一种全局优化的学习过程。评分矩阵的局部结构信息有很多，一种简单直观的局部结构信息是差分矩阵 $D = RS^{\mathrm{T}}$。

$$D = \begin{bmatrix} R_{11} - R_{12} & R_{12} - R_{13} & \cdots & R_{1n-1} - R_{1n} \\ R_{21} - R_{22} & R_{22} - R_{23} & \cdots & R_{2n-1} - R_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ R_{m1} - R_{m2} & R_{m2} - R_{m3} & \cdots & R_{mn-1} - R_{mn} \end{bmatrix},$$

$$S = \begin{bmatrix} 1 & -1 & 0 & \cdots \\ 0 & 1 & -1 & \cdots \\ 0 & 0 & 1 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \quad (3)$$

D 的物理意义是用户对相邻项目的评分差值矩阵，$D_{ij} = R_{ij} - R_{ij+1}$ 是用户 i 对项目 $j+1$ 与项目 j 的评分差值，表示项目 $j+1$ 与项目 j 的相对优劣。用户特征矩阵 M 和项目特征矩阵 N 也应该使得 $\left\|MN^{\mathrm{T}}S^{\mathrm{T}} - D\right\|^2$ 尽量小。但要使用 D 来训练 SVD 模型，还要解决两个问题：

(1)如何从稀疏矩阵 R 中得到差分矩阵 D；

(2)如何求解多目标优化问题：$\min \left\|MN^{\mathrm{T}} - R\right\|^2 + \lambda(\|M\|^2 + \|N\|^2), \min \left\|MN^{\mathrm{T}}S^{\mathrm{T}} - D\right\|^2$。

对于第(1)个问题，根据 D 的物理意义，虽然不同用户的评分尺度不同，但是项目之间的评分差值体现项目的相对优劣，主要由项目的类型和质量决定而不受用户评分尺度的影响(绝大多数用户具备区分项目好坏的能力)，因此可以近似认为不同用户的评分差值向量是相同的，即对于任意用户 i 有 $\widehat{D}_{ij} = \widehat{D}_{1j}, \widehat{D}_{1j} = |T_j|^{-1} \sum_{i \in T_j} (R_{ij} - R_{ij+1})$，其中 T_j 表示同时选择了项目 j 和 $j+1$ 的用户集合，这样就近似得到了局部结构信息 $\widehat{D}$。

因为差分矩阵 $\widehat{D}$ 中的元素不是精确值，$\left\|MN^{\mathrm{T}}S^{\mathrm{T}} - D^2\right\|$ 的值并不能准确表达，所以把 $\min \left\|MN^{\mathrm{T}} - R\right\|^2 + \lambda(\|M\|^2 + \|N\|^2)$ 作为第 1 目标，把 $\min \left\|MN^{\mathrm{T}}S^{\mathrm{T}} - \widehat{D}\right\|^2$ 作为第 2 目标，用分级求解的思想来解决第(2)个问题。首先以 $\min \left\|MN^{\mathrm{T}} - R\right\|^2 + \lambda(\|M\|^2 + \|N\|^2)$ 为目标，利用梯度下降法进行迭代，当误差下降速度较慢时，比如前后两次均方误差(Mean Square Error, MSE)之差小于阈值 β，迭代次数大于阈值 L 等，则以 $\min \left\|MN^{\mathrm{T}}S^{\mathrm{T}} - \widehat{D}\right\|^2$ 为目标以较小的学习速率 lr2 进行迭代，我们称这种方法为结构化 SVD。

以两次 MSE 之差 β 做阈值为例，结构化 SVD 算法步骤如下：

步骤 1 对于训练集合 T 中所有的用户和项目组合 (i, j)：

(1)计算用户 i 对项目 j 的评分：$\widehat{R}_{ij} = M_i N_j^{\mathrm{T}}$；

(2) 计算与真实评分之间的误差：$e_{ij} = R_{ij} - \widehat{R}_{ij}$, $\mathrm{Sum} += e_{ij} \cdot e_{ij}$；

(3)对于用户 i 和项目 j 所有特征 $f(1 \le f \le K)$，用梯度下降法进行更新：

$$\left.\begin{aligned} M_{if} &\leftarrow M_{if} + \mathrm{lr} \cdot (e_{ij} \cdot M_{if} - \lambda N_{jf}) \\ N_{jf} &\leftarrow N_{jf} + \mathrm{lr} \cdot (e_{ij} \cdot N_{jf} - \lambda M_{if}) \end{aligned}\right\}, \ \mathrm{lr}\text{为学习速率} \quad (4)$$

步骤 2 如果 $|\mathrm{PreMSE} - \mathrm{Sum}/|T|| < \beta$，转到步骤 3；

否则，$\mathrm{PreMSE} = \mathrm{Sum}/|T|, \mathrm{Sum} = 0$，迭代步骤 1；

步骤 3 对于训练集合 T 中所有的用户和项目组合 $(i, j+1)$：

(1) 计算误差：$e_{ij} = \widehat{D}_{ij} - (M_i N_j^{\mathrm{T}} - R_{ij+1})$, $\mathrm{Sum} += e_{ij} \cdot e_{ij}$；

(2)对于用户 i 和项目 j 所有特征 $f(1 \le f \le K)$，用梯度下降法进行更新：

$$\left.\begin{aligned} M_{if} &\leftarrow M_{if} + \mathrm{lr2} \cdot e_{ij} \cdot M_{if} \\ N_{jf} &\leftarrow N_{jf} + \mathrm{lr2} \cdot e_{ij} \cdot N_{jf} \end{aligned}\right\}, \ \mathrm{lr2} \ll \mathrm{lr} \quad (5)$$

步骤 4 如果满足终止条件则结束，否则迭代步骤 3。

经典 SVD 算法的训练复杂度为 $O(kmn)$，预测复杂度为 $O(1)$，其中 k 为迭代次数，m 为用户数目，

n 为项目数目[3]。结构化 SVD 相对于经典 SVD 而言，增加的仅仅是计算近似差分矩阵 $\hat{D}$ 的复杂度 $O(mn)$，总的训练复杂度为 $O(mn+k_1mn+k_2mn)$，其中 k_1 为第 1 个循环的迭代次数，k_2 为第 2 个循环的迭代次数。若用 $k=k_1+k_2$ 表示迭代次数的和，则结构化 SVD 的训练复杂度为 $O((k+1)mn)$。可见，结构化 SVD 与经典 SVD 的训练复杂度相同，迭代次数 k 直接反映了训练时间多少，所以后续实验中用迭代次数来对比训练时间。

4　实验设计及结果分析

为了测试结构化 SVD 算法的有效性，主要采用 MovieLens 1 M 数据集合作为测试数据[12]。该数据集合由 GroupLens 提供，包含了 6039 名用户对 3883 部电影的一百多万条评分信息(评分密度为 4.27%)，是测试推荐系统算法性能最常用的数据集合之一。为了验证结构化 SVD 在不同数据集合上都是有效的，从 Netflix Prize[13]数据集合中随机抽取了包含 8662 名用户对 3000 部视频的约 3 万条评分信息(评分密度为 1.13%)，作为对比测试数据集合。

文献[14]全面总结了推荐领域各种不同的评价标准，本文采用了检验推荐算法最常用的预测误差 MAE 和 RMSE 作为评价依据，预测误差越小则算法性能越好。

$$\begin{aligned}\mathrm{MAE}&=|T|^{-1}\sum_{(i,j)\in T}|M_iN_j^{\mathrm{T}}-R_{ij}|\\ \mathrm{RMSE}&=\sqrt{|T|^{-1}\sum_{(i,j)\in T}\left\|M_iN_j^{\mathrm{T}}-R_{ij}\right\|^2}\end{aligned}\tag{6}$$

其中 T 是测试集合，$|T|$ 是 T 中的元素个数。

设计了 3 组实验从不同方面对结构化 SVD 算法进行了测试，实验中经典 SVD 和结构化 SVD 采用相同的默认参数：特征维度 $K=10$，学习率 $\mathrm{lr}=0.005$，正则化因子 $\lambda=0.1$[4,7]；当训练集合上前后两次 MSE 之差小于阈值 $\beta=0.0002$ 时，利用局部结构信息以学习率 $\mathrm{lr2}=0.00003$ 进行更新。

SVD++算法中学习率 $\mathrm{lr}=0.005$，正则化因子 $\lambda_1=0.015,\lambda_2=0.005$[3,4]。基于矩阵分解的算法(经典 SVD，SVD++，结构化 SVD，NMF，BPMF)选取的默认特征空间维度 K 为 10，都只记录了训练过程中的最优值，并未记录过拟合现象。

KNN 算法中以皮尔森相似度公式计算相似性，选择与目标用户最相似的 20 个用户作为邻居(UCF)进行预测。一般来说，KNN 算法中邻居数目小于 20 时，预测准确性会明显下降[3]，基于项目的 KNN 算法要略优于基于用户的 KNN 算法[1,3]。

图 1，图 2 分别显示了在默认参数下，随着特征空间维度 K 增加，SVD 与结构化 SVD 预测误差 MAE，RMSE 的比较结果。从图 1，图 2 中可以看出结构化 SVD 相对于 SVD 算法能够降低约 0.01MAE 和 0.005RMSE，而且这种优势随着特征空间维度的增加保持相对的稳定。从图 1，图 2 中还可以看出，特征空间维度 K 小于 10 时，SVD 与结构化 SVD 预测误差随着 K 的增大而迅速减小；K 大于 10 时，SVD 与结构化 SVD 预测误差随着 K 的增大继续减小，但速度逐渐变缓。这与文献[10]中结论一致，也是选择默认特征空间维度 K 为 10 的原因。

在默认参数下，选取不同迭代次数阈值 L，对比 SVD 与结构化 SVD 的收敛速度，图 3，图 4 分别是 MovieLens 和 Netflix 数据集合上的对比结果。从图中可以看出，在两组数据集合上 SVD 算法的误差下降速度随着迭代次数的增加明显变缓，结构化 SVD 能够明显提高经典 SVD 算法的收敛速度。图 3 中 $L=80$ 时，结构化 SVD 迭代 90 次的效果与 SVD 迭代 200 次的效果相近，节省约 55%的迭代次数；$L=120$ 时，结构化 SVD 迭代 130 次的效果远远好于 SVD 迭代 220 次的效果。Netflix 和 MovieLens 数据集合上的实验结果基本一致，如图 3，图 4 所示。图 4 中 $L=40$ 时，结构化 SVD 很快就发生了过拟合，效果比 SVD 要差。可能的原因是结构化 SVD 在第 1 个学习过程中用的精确信息作为指导，但在第 2 个学习过程利用的近似信息，如果过早进入第 2 个学习过程预测效果反而不好。从两幅图中可以看出，阈值 L 有一个比较大的取值范围，都能使得结构化 SVD 比 SVD 用更少的迭代次数获得更好的预测准确性。

随机选取不同比例样本作为训练集合，其余作为测试集合，对比了结构化 SVD 与 SVD，SVD++算法的预测准确性，如图 5 所示。从图中可以看出 SVD++算法的预测准确性是最好的，结构化 SVD 性能与 SVD++更接近，都明显好于 SVD。但是，SVD++的训练时间是最多的，甚至可以达到 SVD 训练时间的 10 倍之多[3]，而结构化 SVD 算法的训练时间只有 SVD 的一半左右。

图 6 显示了随机选取不同比例样本作为训练集合，结构化 SVD，UCF[1,3]，Slope One[3]，NMF[8]，BPMF[11]算法的预测误差变化曲线。从图 6 中可以看出结构化 SVD 的性能总是最好的，而且对训练集比例不敏感。BPMF 算法在训练集比例较高时性能与结构化 SVD 接近，但对于训练集比例最敏感，在评分矩阵比较稀疏时性能很差，而且算法本身较复杂。相对而言，NMF 算法的预测准确性是最差的。文献[3]详细对比了不同算法复杂度，本文不再赘述。

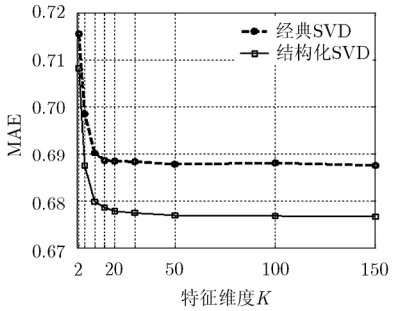

图 1 MAE 随特征空间维度 K 变化的曲线对比

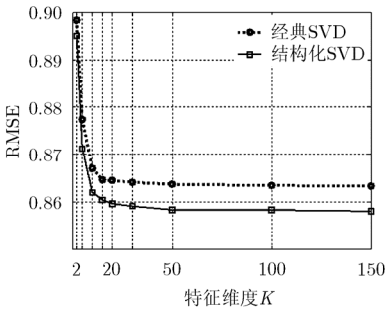

图 2 RMSE 随特征空间维度 K 变化的曲线对比

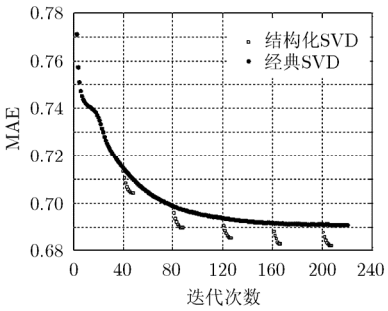

图 3 MovieLens 数据集上收敛速度的对比

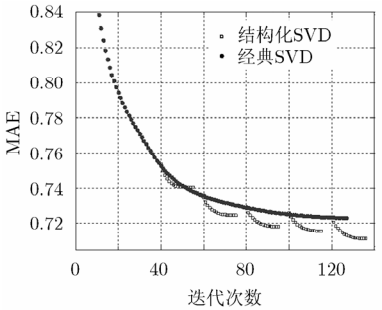

图 4 Netflix 数据集上收敛速度的对比

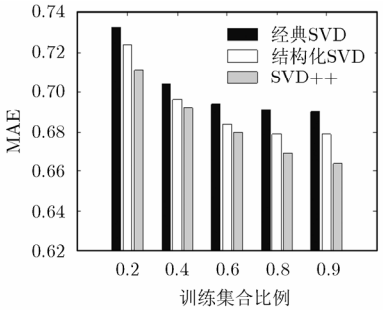

图 5 结构化 SVD 与 SVD, SVD++ 预测准确性 MAE 的对比

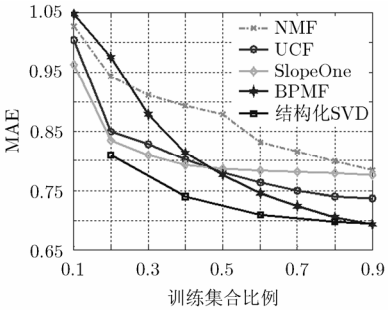

图 6 结构化 SVD 与其他推荐算法的预测准确性 MAE 对比

本文的实验结果与文献[3,15,16]中的结论基本一致。综上所述，SVD++算法的预测准确性最好，但是训练时间太长；结构化 SVD 能够兼顾训练时间和预测准确性，准确性仅次于 SVD++。局部结构信息也可以用来优化 SVD++算法，但是并不能明显降低 SVD++算法的训练时间。值得说明的是，本文给出的结构化 SVD 参数并非是最优的，算法性能还有很大的提升空间。为了解决这个问题，一方面可以寻找更合适的阈值 β，另一方面也可以根据经验设定迭代次数阈值 L 来进行优化。

5 结束语

本文针对经典 SVD 推荐算法收敛速度慢、训练时间长的问题，提出了一种利用评分矩阵局部结构信息来优化 ESVD 的结构化 SVD 算法。在两组实际数据上的实验结果表明，结构化 SVD 在不增加算法复杂度的情况下，能够明显提高预测准确性，同时减少迭代次数。结构化 SVD 的预测准确性略次于 SVD++，比其他主流的算法 KNN, Slope One, NMFBPMF, SVD 都要好，而且在训练时间上具有显著优势。本文的创新点在于提出并证实了评分矩阵的局部结构信息可以提高 SVD 推荐算法的性能。如何更好的利用评分矩阵的结构信息，如何自适应地确定最优参数还有待进一步的探索和研究。

参 考 文 献

[1] Lü Lin-yuan, Medo M, Yeung C H, *et al*.. Recommender systems[J]. *Physics Reports*, 2012, 1(3): 159-172.

[2] Konstan J A and Riedl J T. Recommender systems: from algorithms to user experience[J]. *User Modeling and User-Adapted Interaction*, 2012, 22(1): 101-123.

[3] Cacheda F, Carneiro V, Fernandez D, *et al*.. Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems[J]. *ACM Transactions on the Web*, 2011, 5(2): 1-33.

[4] Ricci F, Rokach L, Shapira B, *et al*.. Recommender Systems Handbook: A Complete Guide for Scientists and Practioners [M]. New York, 2011.

[5] Funk S. Netflix update: try this at home[OL]. http://sifter.org/~simon/journal/20061211.html. 2012.3.

[6] Koren Y, Bell R, and Volinsky C. Matrix factorization techniques for recommender systems[J]. *IEEE Computer*, 2009, 42(1): 30-37.

[7] Paterek A. Improving regularized singular value decomposition for collaborative filtering[C]. Preceedings of KDD Cup and Workshop, California, 2007: 39-42.

[8] Lee D and Seung H. Algorithm for non-negative matrix factorization[C]. Proceedings of Advances in Neural Information Processing Systems, Denver, 2000: 556-562.

[9] Salakhutdinov R and Mnih A. Probabilistic matrix factorization[J]. *Advances in Neural Information Processing Systems*, 2008, 20(1): 1257-1264.

[10] Lawrence N D and Urtasun R. Non-linear matrix factorization with Gaussian processes[C]. Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, 2009: 601-608.

[11] Salakhutdinov R and Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo[C]. Proceedings of the 25th International Conference on Machine Learning, New York, 2008: 880-887.

[12] GroupLens. MovieLens dataset[OL]. http://www. grouplens. org/. 2012.3.

[13] Netflix. Netflix dataset[OL]. http://www.netflixprize.com/. 2012.3.

[14] 朱郁筱, 吕林媛. 推荐系统评价指标综述[J]. 电子科技大学学报, 2012, 41(2): 163-175.
Zhu Yu-xiao and Lü Lin-yuan. Evaluation metrics for recommender systems[J]. *Journal of Electronic Science and Technology of China*, 2012, 41(2): 163-175.

[15] Liu Qi, Chen En-hong, Xiong Hui, *et al*.. Enhancing collaborative filtering by user interest expansion[J]. *IEEE Transactions on Systems, Man, and Cybernetics-part B*: *Cybernetics*, 2012, 42(1): 218-233.

[16] Huang Z, Zeng D, and Chen H. A comparison of collaborative filtering recommendation algorithms for e-commerce[J]. *IEEE Intelligent Systems*, 2007, 22(1): 68-78.

方耀宁： 男，1987 年生，硕士生，研究方向为社会化网络、推荐系统.

郭云飞： 男，1963 年生，教授，博士生导师，研究方向为高性能交换技术、网络安全.

丁雪涛： 男，1988 年生，硕士生，研究方向为文本挖掘、推荐算法、社交网络.

兰巨龙： 男，1962 年生，教授，博士生导师，研究方向为宽带信息网络、下一代互联网.