计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2016.20160582
53(12):2793-2800,2016

基于广义高斯分布的贝叶斯概率矩阵分解方法

燕彩蓉　张青龙　赵　雪　黄永锋
(东华大学计算机科学与技术学院　上海　201620)
(cryan@dhu.edu.cn)

A Method of Bayesian Probabilistic Matrix Factorization Based on Generalized Gaussian Distribution

Yan Cairong, Zhang Qinglong, Zhao Xue, and Huang Yongfeng
(*School of Computer Science and Technology, Donghua University, Shanghai* 201620)

Abstract　The method of Bayesian probability matrix factorization (Bayesian PMF) is widely used in the personalized recommendation systems due to its high prediction accuracy and excellent scalability. However, the accuracy is affected greatly by the sparsity of the initial scoring matrix. A new Bayesian PMF method based on generalized Gaussian distribution called GBPMF is proposed in this paper. In the method, the generalized Gaussian distribution (GGD) is adopted as the prior distribution model in which some related parameters are adjusted automatically through machine learning to achieve desired effect. Meanwhile, we apply the Gibbs sampling algorithm to optimize the loss function. Considering the influence of the time difference of scoring in the prediction process, a temporal factor is integrated into the sampling algorithm to optimize the method and improve its prediction accuracy. The experimental results show that our methods GBPMF and GBPMF-T can obtain higher accuracy when dealing with both sparse matrix and non-sparse matrix, and the latter can even get better effect. When the matrix is very sparse, the accuracy of Bayesian PMF decreases sharply while our methods show stable performance.

Key words　personalized recommender systems; Bayesian PMF; machine learning; generalized Gaussian distribution (GGD); sparse matrix

摘　要　贝叶斯概率矩阵分解方法因较高的预测准确度和良好的可扩展性,常用于个性化推荐系统,但其推荐精度会受初始评分矩阵稀疏特性的影响.提出一种基于广义高斯分布的贝叶斯概率矩阵分解方法GBPMF(generalized Gaussian distribution Bayesian PMF),采用广义高斯分布作为先验分布,通过机器学习自动选择最优的模型参数,并基于Gibbs采样进行高效训练,从而有效缓解矩阵的稀疏性,减小预测误差.同时考虑到评分时差因素对预测过程的影响,在采样算法中添加时间因子,进一步对方法进行优化,提高预测精度.实验结果表明:GBPMF方法及其优化方法GBPMF-T对非稀疏矩阵和稀疏矩阵均具有较高的精度,后者精度更高.当矩阵非常稀疏时,传统贝叶斯概率矩阵分解方法的精度急剧降低,而该方法则具有较好的稳定性.

关键词　个性化推荐系统;贝叶斯概率矩阵分解;机器学习;广义高斯分布;稀疏矩阵

中图法分类号　TP391

收稿日期:2016-08-15;**修回日期**:2016-10-25

基金项目:国家自然科学基金项目(61402100);中央高校基本科研业务费专项资金(16D111210)
This work was supported by the National Natural Science Foundation of China (61402100) and the Fundamental Research Funds for the Central Universities of China (16D111210).

推荐系统作为一种有效的信息过滤手段,是当前解决信息过载问题及实现个性化信息服务的有效方法之一[1].近几年举办的比赛,如 Netflix 百万美金大奖赛、KDD CUP 2011 音乐推荐比赛、百度电影推荐竞赛以及阿里巴巴大数据竞赛更是把推荐系统的研究推向了高潮.

作为协同过滤推荐系统中的一种新型推荐生成方法,基于矩阵分解(matrix factorization, MF)的潜在因子模型(latent factor model)因准确度高、可扩展性好等因素受到了广泛的关注[2].常用的矩阵分解方法主要包括规范化的 SVD(regularized SVD)[3]、非负矩阵分解(nonnegative matrix factorization, NMF)[4]、概率矩阵分解(probabilistic matrix factorization, PMF)[5]和贝叶斯概率矩阵分解(Bayesian PMF)[6]等.其中 Bayesian PMF 从概率的角度探讨矩阵分解的最优化问题,算法的预测准确性比较高,且不需要设定正则化系数,因此得到了广泛应用[7].

现实生活中,数据矩阵往往极其稀疏,以最近热映的电影《魔兽》为例,中国票房 14.7 亿元,观影人次近 4 000 万,豆瓣评分人次为 129 888,评分密度仅为 0.325%.为了缓解数据稀疏问题,学者们提出了社会化推荐(social recommendation)方法[8-9].为了提取更优的潜在特征向量,文献[10]将用户的各种社会网络关系融合到矩阵的优化分解过程中,提出社会化矩阵分解;文献[11]提出因子分解机(factorization machines)模型.然而,用户对项目的评分信息与用户之间的社会关系网络信息往往来源于不同的数据源,因此社会化推荐在推广应用中有一定的局限性[12].

针对矩阵稀疏性影响预测精度的问题,本文在贝叶斯概率矩阵分解的基础上,提出一种基于广义高斯分布的贝叶斯概率分解方法,同时考虑用户评价行为对用户评分的影响会随着时间弱化的情况[13],在方法中添加时间因子,进一步提高预测精度.

1 相关定义

推荐系统中最基本的数据就是关于用户对项目的评分,通过对它们进行分析,了解用户与项目之间的关联,从而实现项目推荐.

定义 1. 评分矩阵.假设有用户向量 $\boldsymbol{u}$(大小为 n)、项目向量 $\boldsymbol{v}$(大小为 m),每个用户对每个项目都可能产生一个评分,其值构成了用户-项目评分矩阵 $\boldsymbol{R}_{n\times m}$.

定义 2. 矩阵分解.把用户-项目评分矩阵 $\boldsymbol{R}$ 分解成用户特征向量矩阵 $\boldsymbol{U}_{n\times k}$ 和项目特征向量矩阵,即 $\boldsymbol{R}\approx\boldsymbol{U}^{\mathrm{T}}\times\boldsymbol{V}$.其中,$k<\min(n,m)$,矩阵 $\boldsymbol{U}_{n\times k}$ 的第 i 列向量 $\boldsymbol{U}_i$ 和 $\boldsymbol{V}_{k\times m}$ 的第 j 列向量 $\boldsymbol{V}_j$ 的乘积 $\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{V}_j$ 表示用户 i 对项目 j 的预测评分.为了获得更加准确的预测值,需要建立目标损失函数:

$$E=\sum_{i=1}^{n}\sum_{j=1}^{m}(r_{ij}-\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{V}_j)^2+\lambda(\|\boldsymbol{U}_i\|^2+\|\boldsymbol{V}_j\|^2). \tag{1}$$

通常采用随机梯度下降(stochastic gradient descent, SGD)方法优化目标损失函数,当其取最小值时对应的 $\boldsymbol{U}_{n\times k}$ 和 $\boldsymbol{V}_{k\times m}$ 即为最优解.

定义 3. 稀疏矩阵.指矩阵中非零元素占全部元素的百分比很小的矩阵(通常设为 5%以下).实际应用中,由于多数用户不会对其所浏览的所有项目做出显式反馈,因此评分矩阵通常是稀疏的.本文主要研究稀疏矩阵的分解.

定义 4. 贝叶斯概率矩阵分解.假设用户特征向量矩阵 $\boldsymbol{U}_{n\times k}$ 和项目特征向量矩阵 $\boldsymbol{V}_{k\times m}$ 服从均值为 μ_U,μ_V,方差为 Λ_U,Λ_V 的高斯分布,用户、项目特征向量矩阵的条件概率分布如下:

$$P(\boldsymbol{U}|\mu_U,\Lambda_U)=\prod_{i=1}^{N}N(\boldsymbol{U}_i|\mu_U,\Lambda_U^{-1}), \tag{2}$$

$$P(\boldsymbol{V}|\mu_V,\Lambda_V)=\prod_{j=1}^{M}N(\boldsymbol{V}_i|\mu_V,\Lambda_V^{-1}). \tag{3}$$

用户对项目的评分变成一个概率问题:

$$P(\boldsymbol{R}|\boldsymbol{U},\boldsymbol{V},\Lambda)=\prod_{i=1}^{N}\prod_{j=1}^{M}[N(r_{ij}|\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{V}_j,\Lambda^{-1})]^{I_{ij}}, \tag{4}$$

其中,$N(x|\mu,\Lambda^{-1})$ 是期望为 μ、方差为 Λ^{-1} 的高斯分布,I_{ij} 是示性函数;若 $r_{ij}\neq0$,则 $I_{ij}=1$,否则 $I_{ij}=0$.Bayesian PMF 进一步设定 $\Theta_U=\{\mu_U,\Lambda_U\}$,$\Theta_V=\{\mu_V,\Lambda_V\}$ 的先验分布为高斯-威沙特分布(Gaussian-Wishart distribution),将参数 $\{\Lambda_U,\Lambda_V\}$ 整合到算法内部,概率调整为

$$P(\Theta_U|\Theta_0)=P(\mu_U|\Lambda_U)P(\Lambda_U)=N(\mu_U|\mu_0,(\beta_0\Lambda_U)^{-1})W(\Lambda_U|\omega_0,\nu_0), \tag{5}$$

$$P(\Theta_V|\Theta_0)=P(\mu_V|\Lambda_V)P(\Lambda_V)=N(\mu_V|\mu_0,(\beta_0\Lambda_V)^{-1})W(\Lambda_V|\omega_0,\nu_0), \tag{6}$$

其中:

1) $\Theta_0=\{\mu_0,\nu_0,\omega_0,\Lambda,\beta_0\}$;

2) $W(\Lambda_\nu|\omega_0,\nu_0)$ 是自由度为 υ_0、尺度参数为 ω_0 的威沙特分布.

贝叶斯推断是将先验的思想和样本数据相结合得到后验分布,然后根据后验分布进行统计推断,其精度受样本数量及其先验分布准确性的影响.

Bayesian PMF 采用高斯分布作为先验分布,对数据比较敏感,当评分矩阵中非零元素较少时,这种方法具有较高的精度,但当评分矩阵非常稀疏时,很难断定样本分布服从高斯分布,其推荐效果不理想[6].本文针对评分矩阵稀疏性不确定问题,提出采用适用范围更加宽泛的广义高斯分布作为先验分布来缓解数据的稀疏问题,提高推荐精度.

Bayesian PMF 采用 Markov 链蒙特卡罗算法(Markov chain Monte Carlo, MCMC)进行训练,该算法具有较低的算法复杂度和较高的检测性能,目前在 MCMC 方法中最常用的是 Gibbs 采样(Gibbs sampling)算法.因此,本文在训练基于广义高斯分布的贝叶斯概率矩阵分解方法 GBPMF(generalized Gaussian distribution Bayesian PMF)的过程中,使用 Gibbs 采样算法进行贝叶斯推断.

2 基于广义高斯分布的贝叶斯概率矩阵分解

2.1 广义高斯分布对稀疏数据的影响

广义高斯分布(generalized Gaussian distribution, GGD)的密度函数(probability density function)是广义伽玛分布的密度函数的推广形式,其密度函数定义为[14]

$$GGD(x|\alpha,\beta,\mu)=\frac{\alpha}{2\beta\Gamma(1/\alpha)}e^{-\left|\frac{x-\mu}{\beta}\right|^{\alpha}}, \tag{7}$$

其中:

1) $\beta=\sqrt{\frac{\sigma^2\Gamma(1/\alpha)}{\Gamma(3/\alpha)}}=\sigma\sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}}$,$\sigma>0$;

2) $\Gamma(*)$是 Gamma 函数,$\Gamma(x)=\int_0^{\infty}e^{-t}t^{x-1}dt$;

3) 参数 $\mu,\sigma^2,\alpha,\beta$ 分别称为 GGD 的均值、方差、形状参数和尺度参数.

图 1 所示为 $\mu=0,\sigma^2=10,\alpha$ 分别为 2.0,1.0,0.8 的 GGD 概率密度图.其中纵坐标表示样本的概率分布密度,0 点处纵坐标值越大表示样本取 0 值时的概率密度越大,即样本的稀疏性越大.通过图 1 所示,我们可以看出:样本的稀疏率与 α 值呈负相关,即 α 值越小时,GGD 在 0 附近有越高的峰值;当 $\alpha=2.0$ 时 GGD 为高斯分布,我们可以通过调节 α 值来有效缓解数据的稀疏性.同时,相对于高斯分布,GGD 在数据两侧出现的概率较大,这有助于提高推荐系统对项目长尾特性的发掘能力.

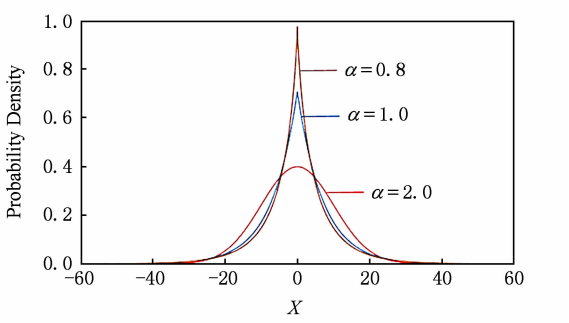

Fig. 1 Probability density comparison of GGD.

图 1 GGD 概率密度变化示例

针对稀疏矩阵导致推荐结果误差较大的问题,本文提出一种改进的贝叶斯概率矩阵分解方法 GBPMF,采用 GGD 作为用户-项目特征向量矩阵的先验分布来有效缓解矩阵稀疏性.

2.2 GBPMF 方法

GBPMF 方法中,用户真实评分矩阵仍服从均值为 $\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{V}_j$、方差为 Λ 的高斯分布,用户特征向量 $\boldsymbol{U}_i$ 和项目特征向量 $\boldsymbol{V}_j$ 分别服从形状参数为 α_U,α_V 且尺度参数为 β_U,β_V 的广义高斯分布.为了得到完整的贝叶斯过程,我们为 Λ^{-1} 引入形状参数为 α、尺度参数为 β 的逆伽马分布 $\Gamma^{-1}(\alpha,\beta)$:

$$P(\Lambda|\alpha,\beta)=\frac{b^{a}}{\Gamma(\alpha)}(\Lambda)^{-(\alpha+1)}e^{-\beta/\Lambda}, \tag{8}$$

$\{\mu_U,\mu_V\}$对评分预测的影响不大,一般可以设为 0.文献[15]已经证明 GGD 参数比函数为 $R(\alpha)=\frac{\Gamma^2(2/\alpha)}{\Gamma(1/\alpha)\Gamma(3/\alpha)}$.由图 1 可知,GGD 概率密度分布函数为非光滑曲线,很难直接进行 Gibbs 采样,为了便于 MCMC 训练过程中后验概率的计算,可以假设 GGD 为混合尺度的高斯分布,即:

$$f(x|\alpha,\beta,\mu)=\int_0^{\infty}T\times N(x|\mu,\sigma^2)R(\sigma|\alpha)d\sigma^2. \tag{9}$$

GBPMF 方法的评分矩阵预测过程如下:

1) 根据 GGD 得到用户特征向量 $\boldsymbol{U}_i$ 和项目特征向量 $\boldsymbol{V}_j$;

2) 根据逆伽马分布计算高斯分布的方差 Λ;

3) 根据高斯分布 $N(r_{ij}|\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{V}_j,\Lambda)$,可以得到最终的预测评分 r_{ij}.

Gibbs 采样是一种典型的 MCMC 算法,适用于联合概率未知,条件概率容易获取的情况.GBPMF 方法在训练过程中,采用 Gibbs 采样进行贝叶斯推断,即利用条件概率构造平稳分布为所求联合概率的 Markov 链,进行 K 次抽样,此时的样本$\{\boldsymbol{U},\boldsymbol{V}\}$可近似认为是来自联合概率 $P(\boldsymbol{U},\boldsymbol{V}|\boldsymbol{R},\alpha_U,\beta_U,\alpha_V,\beta_V)$的抽样,最后利用式(10)进行评分预测:

$$P(r_{ij}^{*}|\boldsymbol{R})=\frac{1}{K}\sum_{k=1}^{K}P(r_{ij}^{*}|U_i^{k},V_j^{k}). \tag{10}$$

Gibbs 采样的具体过程可以描述为:

1) 对参数 $\boldsymbol{U}_i$ 进行采样,提取出与之相关的所有变量,利用贝叶斯公式,可得:

$$P(\boldsymbol{U}_i|\boldsymbol{R},\boldsymbol{V},\Lambda,(\alpha_U)_i,(\beta_U)_i)\propto \prod_{i=1}^{N}[N(r_{ij}|\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{V}_j,\Lambda)]^{I_{ij}}GGD(\boldsymbol{U}_i|(\alpha_U)_i,(\beta_U)_i). \tag{11}$$

令 $P((\lambda_U)_{ki}|(\alpha_U)_{ki},(\beta_U)_{ki})=R((\lambda_U)_{ki}|(\alpha_U)_{ki})$,则有

$$P(\boldsymbol{U}_i|\boldsymbol{R},\boldsymbol{V},\Lambda,(\alpha_U)_i,(\beta_U)_i)=N(\boldsymbol{U}_i|\mu_i^*,(\Lambda_i^*)^{-1}), \tag{12}$$

其中:

① $\Lambda_i^*=(\lambda_U)_i^{-1}+\sum_{j=1}^{M}\frac{(\boldsymbol{V}_j^{\mathrm{T}}\boldsymbol{V}_j)^{I_{ij}}}{\Lambda}$;

② $\mu_i^*=[\Lambda_i^*]^{-1}\left(\sum_{j=1}^{M}\frac{(r_{ij}\boldsymbol{V}_j^{\mathrm{T}}\boldsymbol{V}_j)^{I_{ij}}}{\Lambda}\right)$.

2) 对超参数 $(\lambda_U)_i$,$(\alpha_U)_i$ 采样,根据贝叶斯公式,可得:

$$P((\lambda_U)_i|\boldsymbol{U}_i,(\alpha_U)_i)\propto P(\boldsymbol{U}_i|(\lambda_U)_i,(\alpha_U)_i)P((\lambda_U)_i). \tag{13}$$

根据指数函数的性质,可以得到 $(\lambda_U)_i$ 服从逆高斯分布(inverse Gauss distribution)

$$P((\lambda_U)_i^{-1}|\boldsymbol{U}_i,(\alpha_U)_i)=G^{-1}(\sqrt{(\lambda_U)_i}|\boldsymbol{U}_i,(\alpha_U)_i). \tag{14}$$

对广义高斯分布的形状参数 $(\alpha_U)_i$,它的条件概率满足广义逆高斯分布

$$P((\alpha_U)_i|\boldsymbol{U}_i,(\lambda_U))=GIG(\gamma+1,(\lambda_U)_i+a,b), \tag{15}$$

其中,a,b 为常数,本文为了计算方便,取 $\gamma=0.5$ 将广义逆高斯分布简化为逆高斯分布.

$\boldsymbol{U}$ 和 $\boldsymbol{V}$ 具有对称性,采样具有相同的形式.

3) 对参数 Λ 进行采样. 参数 Λ 的条件概率形式为

$$P(\Lambda|\boldsymbol{R},\boldsymbol{U},\boldsymbol{V})=\Gamma^{-1}(a_\Lambda,b_\Lambda), \tag{16}$$

其中:

① $a_\Lambda=\frac{N\times M}{2}+1+a$;

② $b_\Lambda=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}(r_{ij}-\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{V}_j)^2+b$.

GBPMF 的 Gibbs 采样算法见算法 1 所示.

算法 1. GBPMF 的 Gibbs 采样算法.

输入:原始评分矩阵 $\boldsymbol{R}$、采样数目 K、迭代次数 D;

输出:K 个样本点.

① 初始化算法参数 $(\boldsymbol{U}^1,\boldsymbol{V}^1)$;

② for 每一个采样点 do

③ 根据式 $\boldsymbol{U}_i^{k+1}\sim P(\boldsymbol{U}_i|\boldsymbol{V}^k,\Lambda,(\lambda_U)_i)$ 对每一个用户特征向量 $\boldsymbol{U}_i$ 进行采样;

④ 根据式 $\boldsymbol{V}_j^{k+1}\sim P(\boldsymbol{V}_j|\boldsymbol{U}^k,\Lambda,(\lambda_V)_j)$ 对每一个项目特征向量 $\boldsymbol{V}_j$ 进行采样;

⑤ 根据式 $\Lambda\sim P(\Lambda|\boldsymbol{R},\boldsymbol{U},\boldsymbol{V})$ 对参数 Λ 进行采样;

⑥ end for

⑦ 返回 K 个采样点 $(\boldsymbol{U}^1,\boldsymbol{V}^1),(\boldsymbol{U}^2,\boldsymbol{V}^2),\cdots,(\boldsymbol{U}^k,\boldsymbol{V}^k)$.

每次 Gibbs 采样的时间为 $K\times\max(m,n)$,假设迭代 D 次,则采样时间为 $D\times K\times\max(m,n)$;又矩阵分解的运行时间是 $F\times S\times p$,其中 F 为用户对物品的评分记录数,p 为分解维度,S 为迭代次数[16]. 故整个模型的运行时间为 $D\times K\times\max(m,n)+F\times S\times p$. 通常情况下,$D\times K\times\max(m,n)<F\times S\times p$,故整个模型的时间复杂度为 $O(F\times S\times p)$,这与 Bayesian PMF 方法的时间复杂度一样.

3 基于评分时差的 GBPMF 方法优化

GBPMF 方法在处理时不考虑评分产生的时间因素. 实际应用中,评分产生的时间因素能够反映用户的行为变化,因而对预测有较大影响. 这点在已有的矩阵分解模型中没有考虑到. 本文通过在采样算法中添加评分时差因素来进行调优,优化后的方法简称为 GBPMF-T,处理过程如下:

1) 为 GBPMF 模型添加偏置项:

$$b_{ij}=\mu+b_i+b_j, \tag{17}$$

其中,$\mu=\sum_{i,j\in T}r_{ij}\Big/|T|$ 表示全局平均评分,$b_i=\sum_{j\in N(i)}r_{ij}\Big/|R(i)|$ 表示用户评分偏置,$b_j=\sum_{i\in N(j)}r_{ij}\Big/|R(j)|$ 表示项目评分偏置. 此时预测评分为 $\hat{r}_{ij}=b_{ij}+\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{V}_j$,$b_{ij}$ 为评分偏置项. 文献[2]已经证明,添加偏置因素可以有效提高评分预测准确率.

2) 参考文献[17]建立邻域模型,此时预测评分为

$$\hat{r}_{ij}=b_{ij}+\sum_{l\in R(i)}(r_{il}-b_{il})\omega'_{jl}+\sum_{l\in N(i)}c'_{jl}, \tag{18}$$

其中,$R(i)$,$N(i)$ 分别为用户 i 评分的物品集合、用户 i 潜在偏好的物品集合,ω'_{jl},c'_{jl} 为相应权重. 本文摈弃了传统的相似度概念,ω'_{jl},c'_{jl} 是相对于基准预测的偏移,可以在模型优化的过程中通过训练获得. 以 ω'_{jl} 为例,对于 2 个相关的物品 j,l,当用户 i 对物

品 l 的评分高于基准预测值 $r_{il}-b_{il}$,本文通过添加 $(r_{il}-b_{il})\omega'_{jl}$ 来提高用户 i 对物品 j 的基准预测.这样模型在训练的过程中,可以实时更新用户对物品的偏好程度,进而获得更高的精度.

3) 为了提高推荐准确率,对权重矩阵进行 0-1 标准化,即:

$$\begin{aligned}\boldsymbol{\omega}&=\frac{\boldsymbol{\omega}'-\min(\boldsymbol{\omega}')}{\max(\boldsymbol{\omega}')-\min(\boldsymbol{\omega}')},\\ \boldsymbol{c}&=\frac{\boldsymbol{c}'-\min(\boldsymbol{c}')}{\max(\boldsymbol{c}')-\min(\boldsymbol{c}')}.\end{aligned}\tag{19}$$

文献[18]已经证明,对权重矩阵标准化处理可以有效提高评分预测准确率.

4) 将评分时差融入到基于领域的算法中,修正相关参数,建立算法为

$$\hat{r}_{ijt}=\frac{\sum\limits_{l\in R(i)}f(\omega_{jl},\Delta t)(r_{il}-b_{il})}{|R(i)|},\tag{20}$$

其中,$\Delta t=t_{ij}-t_{il}$ 表示用户 i 对项目 j 和项目 l 的评分时差;$f(\omega_{jl},\Delta t)$ 是一个考虑了时间衰减后的相似度函数,它的主要目的是建立用户行为与评分时差的函数,提高用户最近行为在推荐系统中的权重.

5) 定义 f 为

$$f(\omega_{jl},\Delta t)=\omega_{jl}\sigma\left(\frac{-|\Delta t|}{\beta}+\gamma\right),\tag{21}$$

其中,$\sigma(x)=\frac{1}{1+\mathrm{e}^{-x}}$ 表示 sigmoid 函数,目的是将时间对用户行为的影响控制在(0,1)范围内.从定义中可以看出,随着 Δt 的增加,$f(\omega_{jl},\Delta t)$ 会越来越小,即用户行为的预测随时间衰减.

6) 加入时间信息后的目标损失函数为

$$\begin{aligned}E=&\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\Bigg(r_{ij}-\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{V}_j-b_{ij}-\\&\frac{\sum\limits_{l\in R(i)}f(\omega_{jl},\Delta t)(r_{il}-b_{il})}{|R(i)|}-\frac{\sum\limits_{l\in N(i)}c_{jl}}{|N(i)|}\Bigg)^2+\\&\frac{\lambda}{2}(\|\boldsymbol{U}_i\|^2+\|\boldsymbol{V}_j\|^2+\|\boldsymbol{b}_i\|^2+\\&\|\boldsymbol{b}_j\|^2+\|\boldsymbol{\omega}\|^2+\|\boldsymbol{c}\|^2),\end{aligned}\tag{22}$$

其中,$\lambda/2$ 是为防止过拟合添加的正则化参数.

7) 为了节省存储空间,我们对 $\boldsymbol{\omega},\boldsymbol{c}$ 进行分解,即 $\boldsymbol{\omega}=\boldsymbol{V}_j^{\mathrm{T}}\boldsymbol{X}_l$,$\boldsymbol{c}=\boldsymbol{V}_j^{\mathrm{T}}\boldsymbol{Y}_l$.加入时间信息后的目标损失函数为

$$\begin{aligned}E=&\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(r_{ij}-b_{ij}-\boldsymbol{V}_j^{\mathrm{T}}\left(\boldsymbol{U}_i+\frac{\sum\limits_{l\in R(i)}\sigma\left(\frac{-|\Delta t|}{\beta}+\gamma\right)(r_{il}-b_{il})\boldsymbol{X}_l}{|R(i)|}+\frac{\sum\limits_{l\in N(i)}\boldsymbol{Y}_l}{|N(i)|}\right)\right)^2+\\&\frac{\lambda}{2}(\|\boldsymbol{U}_i\|^2+\|\boldsymbol{V}_j\|^2+\|\boldsymbol{b}_i\|^2+\|\boldsymbol{b}_j\|^2+\|\boldsymbol{X}_l\|^2+\|\boldsymbol{Y}_l\|^2).\end{aligned}\tag{23}$$

同样我们可以用 Gibbs 采样对 GBPMF-T 方法进行贝叶斯推断.此时,K 次抽样时的样本 $\{\boldsymbol{U},\boldsymbol{V},\boldsymbol{X},\boldsymbol{Y}\}$ 可认为是来自联合概率 $P(\boldsymbol{U},\boldsymbol{V},\boldsymbol{X},\boldsymbol{Y}|\boldsymbol{R},\alpha_U,\beta_U,\alpha_V,\beta_V,\alpha_X,\beta_X,\alpha_Y,\beta_Y)$ 的抽样,其近似计算为

$$P(r_{ij}^*|\boldsymbol{R})=\frac{1}{K}\sum_{k=1}^{K}P(r_{ij}^*|U_i^k,V_j^k,X_l,Y_l^k).\tag{24}$$

由于考虑了评分时差和偏置因素,在 GBPMF-T 方法中,假设真实评分矩阵服从均值为 $b_{ij}+\boldsymbol{V}_j^{\mathrm{T}}\cdot\left(\boldsymbol{U}_i+\frac{\sum\limits_{l\in R(i)}\sigma\left(\frac{-|\Delta t|}{\beta}+\gamma\right)(r_{il}-b_{il})\boldsymbol{X}_l}{|R(i)|}+\frac{\sum\limits_{l\in N(i)}\boldsymbol{Y}_l}{|N(i)|}\right)$,方差为 Λ 的高斯分布,$\boldsymbol{X},\boldsymbol{Y}$ 分别服从形状参数为 α_X,α_Y 且尺度参数为 β_x,β_y 的广义高斯分布.

由于 $\boldsymbol{X},\boldsymbol{Y}$ 和 $\boldsymbol{U},\boldsymbol{V}$ 类似,可以按式(12)进行采样.此时参数 Λ 的条件概率形式为

$$P(\Lambda|\boldsymbol{R},\boldsymbol{U},\boldsymbol{V},\boldsymbol{X},\boldsymbol{Y})=\Gamma^{-1}(a_\Lambda,b_\Lambda),\tag{25}$$

其中:

1) $a_\Lambda=\frac{N\times M}{2}+1+a$;

2) $b_\Lambda=\frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m}\left(r_{ij}-b_{ij}-\boldsymbol{V}_j^{\mathrm{T}}\left(\boldsymbol{U}_i+\boldsymbol{\varphi}+\frac{\sum\limits_{l\in N(i)}\boldsymbol{Y}_l}{|N(i)|}\right)\right)^2+b$;

3) $\boldsymbol{\varphi}=\frac{\sum\limits_{l\in R(i)}\sigma\left(\frac{-|\Delta t|}{\beta}+\gamma\right)(r_{il}-b_{il})\boldsymbol{X}_l}{|R(i)|}$.

GBPMF-T 的 Gibbs 采样算法见算法 2 所示.

算法 2. GBPMF-T 的 Gibbs 采样算法.

输入:原始评分矩阵 $\boldsymbol{R}$、时间矩阵 $\boldsymbol{T}$、数目 K、迭代次数 D;

输出:K 个样本点.

① 初始化算法参数$(\boldsymbol{U}^1,\boldsymbol{V}^1,\boldsymbol{X}^1,\boldsymbol{Y}^1)$;

② for 每一个采样点 do

③ 根据式 $\boldsymbol{U}_i^{k+1}\sim P(\boldsymbol{U}_i|\boldsymbol{V}^k,\Lambda,(\lambda_U)_i)$ 对每一个用户特征向量 $\boldsymbol{U}_i$ 进行采样;

④ 根据式 $\boldsymbol{V}_j^{k+1}\sim P(\boldsymbol{V}_j|\boldsymbol{U}^k,\boldsymbol{X}^k,\boldsymbol{Y}^k,\Lambda,(\lambda_V)_j)$ 对每一个项目特征向量 $\boldsymbol{V}_j$ 进行采样;

⑤ 根据式 $\boldsymbol{X}_l^{k+1}\sim P(\boldsymbol{X}_l|\boldsymbol{V}^k,\Lambda,(\lambda_X)_l)$ 对每一

个 $\boldsymbol{X}_l$ 进行采样;

⑥ 根据式 $\boldsymbol{Y}_l^{k+1} \sim P(\boldsymbol{Y}_l \mid \boldsymbol{V}^k, \Lambda, (\lambda_Y)_l)$ 对每一个 $\boldsymbol{Y}_l$ 进行采样;

⑦ 根据式 $\Lambda \sim P(\Lambda \mid \boldsymbol{R}, \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{X}, \boldsymbol{Y})$ 对参数 Λ 进行采样;

⑧ end for

⑨ 返回 K 个采样点 $(\boldsymbol{U}^1, \boldsymbol{V}^1, \boldsymbol{X}^1, \boldsymbol{Y}^1), (\boldsymbol{U}^2, \boldsymbol{V}^2, \boldsymbol{X}^2, \boldsymbol{Y}^2), \cdots, (\boldsymbol{U}^k, \boldsymbol{V}^k, \boldsymbol{X}^k, \boldsymbol{Y}^k)$.

整个模型的时间复杂度为 $O(F \times S \times p)$,这和 Bayesian PMF 方法的时间复杂度一样.

4 实验和结果

4.1 数据集和评价标准

实验采用 ml-1m 数据集、ml-10m 数据集和 Netflix 数据集作为测试数据集来检验 GBPMF 和 GBPMF-T 方法的预测精度. MovieLens 数据集为用户对自己看过的电影进行评分的数据集,评分分值为 1～5. MovieLens 数据集包括 2 个不同大小的库,小规模的库 ml-1m 数据集是 6 040 个用户对 3 900 部电影的大约 100 万次评分;大规模的库 ml-10m 数据集是 71 567 个用户对 10 681 部电影的大约 1 000 万次评分的数据. Netflix 数据集来自于电影租赁网 Netflix 的数据库,包含了 480 189 个匿名用户对大约 17 770 部电影作出的大约 1 亿次评分. 3 个数据集的统计信息如表 1 所示:

Table 1 Information of the Datasets

表 1 数据集的统计信息

Dataset Name	Number of Users	Number of Movies	Ratings	Density/%
ml-1m	6 040	3 952	1 000 209	4.19
ml-10m	71 567	10 681	10 000 054	1.31
Netflix	480 189	17 770	100 480 507	1.1

实验使用的主要评价标准是在推荐预测系统中常用的均方根误差(root mean square error, *RMSE*)和平均绝对误差(mean absolute error, *MAE*). *RMSE*, *MAE* 值越小,表示算法性能越好. *RMSE* 定义为

$$RMSE = \frac{\sqrt{\sum_{i,j \in T} (r_{ij} - \hat{r}_{ij})^2}}{|T|}. \tag{26}$$

MAE 采用绝对值计算预测误差,定义为

$$MAE = \frac{\sum_{i,j \in T} |r_{ij} - \hat{r}_{ij}|}{|T|}, \tag{27}$$

其中, T 是测试集, $|T|$ 是测试集的记录数, r_{ij} 表示测试集的第 ij 条记录, r_{ij} 是真实值, $\hat{r}_{ij}$ 是预测值.

本文设计了 3 组实验对经典 Bayesian PMF, GBPMF, GBPMF-T 性能进行对比.

实验运行硬件平台为 Inter® Core™ i5-4460 CPU @ 3.20 GHz、15.6 GB 内存、976 GB 硬盘、64 位 Ubuntu 15.1 操作系统,编译软件为 IntelliJ IDEA,算法编程语言为 Python.

4.2 实验结果与分析

实验分别将 ml-1m 数据集、ml-10m 数据集和 Netflix 数据集,按照 9∶1的比率随机划分为训练集和测试集,实验运行 10 次,结果取平均值. 考虑到时间和精度问题,经过多次实验,我们设置学习速率为 0.05,初始化正则参数为 0.01.

由于整个实验所采用的 3 个影响因子相互独立,在实验设计时,我们主要考虑单个因子对实验结果的影响.

A 组实验考虑 Gibbs 采样迭代次数对实验结果的影响,选取 ml-1m 数据集测试,依次增大 Gibbs 采样迭代次数,直到 Bayesian PMF 和 GBPMF 测试的 *MAE* 值趋于收敛. 不失一般性,实验选取矩阵分解特征维数为 10,实验结果如图 2 所示:

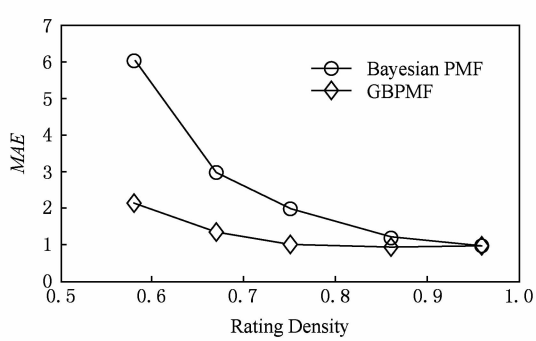

Fig. 2 *MAE* comparison on Gibbs iterations.

图 2 Gibbs 采样次数对性能的影响

从图 2 中可以看出,相对于 Bayesian PMF, GBPMF 进行 Gibbs 采样, *MAE* 值趋于稳定,需要的迭代次数更少. 通过第 2 节对 Gibbs 采样算法时间复杂度的分析可知, Gibbs 采样耗时与迭代次数成正相关. 可见,相对于经典 Bayesian PMF, GBPMF 在运行时间上具有一定的优势.

我们设计 *B* 组实验测试矩阵分解特征维数对模型性能的影响. 实验选取 3 种不同的数据集,以 *RMSE* 值作为最终测评标准,其中 Gibbs 采样迭代次数取 50,实验结果如图 3,4 所示:

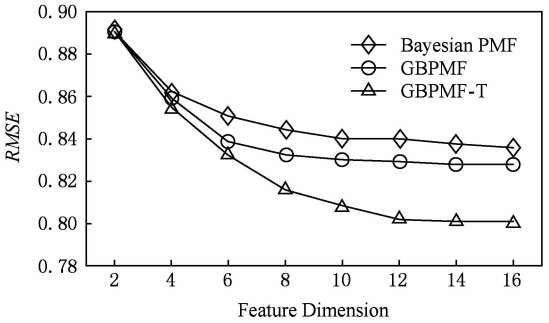

Fig. 3 *RMSE* comparison on ml-1m.

图 3 ml-1m 的 *RMSE* 值比较

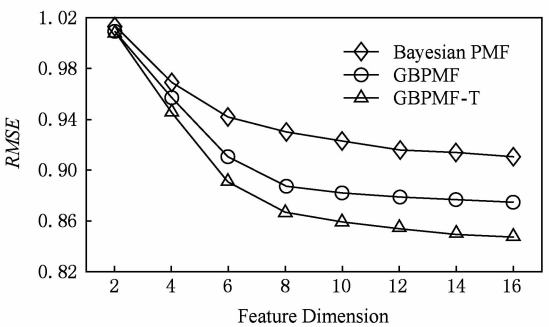

Fig. 4 *RMSE* comparison on Netflix.

图 4 Netflix 的 *RMSE* 值比较

从图 3,4 中可以看出,在 ml-1m 数据集上,与经典 Bayesian PMF 相比,GBPMF 的预测精度提高了 1.02%,GBPMF-T 的预测精度提高了 4.25%;在 Netflix 数据集上,与经典 Bayesian PMF 相比,GBPMF 的预测精度提高了多少 3.83%,GBPMF-T 的预测精度提高了 6.78%,可见评分时差和偏置因素对推荐系统预测精度的影响很大. 另外,由于 Netflix 数据集的评分密度只有 1.1%,小于 ml-1m 数据集的评分密度(4.19%),我们大胆推测,在数据稀疏的条件下,GBPMF 和 GBPMF-T 算法能够获得更高的精度.

为了验证上述推测,我们只考虑矩阵稀疏性对推荐精度的影响,进行 C 组实验,利用 ml-10m 数据集生成评分密度小于 1%的测试环境(设定用户对电影评分数量的阈值为 *num_m*. 若用户评分记录数量小于 *num_m*,评分记录取实际评分数量;若用户评分数量大于 *num_m*,随机抽取 *num_m* 条评分记录). 实际应用中的评分密度一般都小于 1%,稀疏矩阵可以更好地反映算法提取潜在特征的能力. 实验结果如图 5 所示.

从图 5 中可以看出,在稀疏数据集上(数据稀疏率为 0.96%),与经典 Bayesian PMF 相比,GBPMF

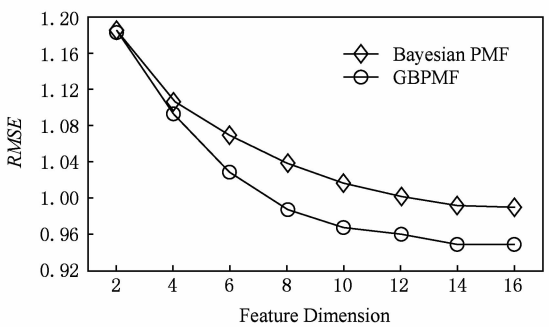

Fig. 5 *RMSE* comparison on sparse matrix.

图 5 稀疏矩阵的 *RMSE* 值比较

的预测精度提高了 4.03%,且 2 种算法在矩阵分解特征维数为 16 的时候都已经收敛.

在实际的推荐系统中,数据的稀疏性往往小于 1%,为了更直观地显示数据的稀疏性,我们设计了 D 组实验,选择矩阵分解特征维数为 16,通过调节 *num_m* 的值改变数据集的稀疏性,实验结果如图 6 所示:

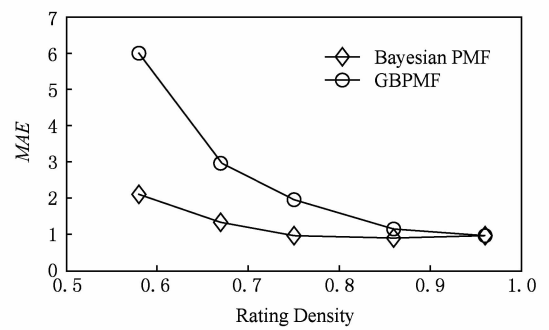

Fig. 6 Influence of matrix sparsity on two methods.

图 6 矩阵稀疏性对 Bayesian PMF 和 GBPMF 的影响

通过图 6 可以看出,随着评分密度的减小,Bayesian PMF 和 GBPMF 的预测精度都在迅速减小,但 GBPMF 预测精度减小的幅度要小于 Bayesian PMF. 值得注意的是,当矩阵极其稀疏(<0.5%)时,GBPMF 预测精度要远好于 Bayesian PMF,从而说明 GBPMF 能够有效缓解矩阵稀疏性问题.

5 结束语

本文针对稀疏评分矩阵会降低推荐精度的问题,提出基于广义高斯分布的贝叶斯概率矩阵分解方法 GBPMF. 方法采用广义高斯分布作为先验分布,通过调节参数值来有效缓解数据的稀疏性,也有助于提高推荐系统对项目长尾的发掘能力;在方法训练过程中,使用 Gibbs 采样进行贝叶斯推断,适用

于联合概率未知,条件概率容易获取的情况;最后通过添加评分时差因子对方法进行优化,进一步提高方法的精度.实验表明:在数据稀疏的情况下,方法仍能保持较高的精度,从而有效提高推荐系统的预测准确率.

在未来的研究中,我们将有效挖掘用户和项目的属性以及属性之间的关系,从而确定有用的隐含特征,进一步提高矩阵分解方法的精度.

参 考 文 献

[1] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry [J]. Communications of the ACM, 1992, 35(12): 61-70

[2] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems [J]. Computer, 2009, 42(8): 30-37

[3] Billsus D, Pazzani M J. Learning collaborative information filters [C] //Proc of the 4th Int Conf on Machine Learning. New York: ACM, 1998: 46-54

[4] Lee D, Seung H S. Learning the parts of objects by non-negative matrix factorization [J]. Nature, 1999, 401(6755): 788-91

[5] Mnih A, Salakhutdinov R. Probabilistic matrix factorization [C] //Proc of the 29th Int Conf on Machine Learning. New York: ACM, 2012: 880-887

[6] Salakhutdinov R. Bayesian probabilistic matrix factorization using MCMC [C] //Proc of the 25th Int Conf on Machine Learning. New York: ACM, 2008: 880-887

[7] Fang Yaoning, Guo Yunfei, Lan Julong. A Bayesian probabilistic matrix factorization algorithm based on logistic function [J]. Journal of Electronics & Information Technology, 2014(3): 715-720 (in Chinese)
(方耀宁, 郭云飞, 兰巨龙. 基于 Logistic 函数的贝叶斯概率矩阵分解算法[J]. 电子与信息学报, 2014(3): 715-720)

[8] Wang Zhi, Sun Lifeng, Zhu Wenwu, et al. Joint social and content recommendation for user-generated videos in online social network [J]. IEEE Trans on Multimedia, 2013, 15(3): 698-709

[9] Quijano-Sanchez L, Recio-Garcia J A, Diaz-Agudo B, et al. Social factors in group recommender systems [J]. ACM Trans on Intelligent Systems & Technology, 2013, 4(1): 1199-1221

[10] Jamali M, Ester M. A transitivity aware matrix factorization model for recommendation in social networks [C] //Proc of the 22nd Int Joint Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2011: 2644-2649

[11] Rendle S. Factorization machines [C] //Proc of the 10th IEEE Int Conf on Data Mining. New York: ACM, 2010: 995-1000

[12] Meng Xiangwu, Liu Shudong, Zhang Yujie, et al. Research on social recommender systems [J]. Journal of Software, 2015, 26(6): 1356-1372 (in Chinese)
(孟祥武, 刘树栋, 张玉洁, 等. 社会化推荐系统研究[J]. 软件学报, 2015, 26(6): 1356-1372)

[13] Koren Y. Collaborative filtering with temporal dynamics [C] //Proc of the 15th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2009: 89-97

[14] Miller J, Thomas J. Detectors for discrete-time signals in non-Gaussian noise [J]. IEEE Trans on Information Theory, 1972, 18(2): 241-250

[15] Wang Taiyue, Li Zhiming. A fast parameter estimation of generalized Gaussian distribution [J]. Chinese Journal of Engineering Geophysics, 2006, 3(3): 172-176 (in Chinese)
(汪太月, 李志明. 一种广义高斯分布的参数快速估计法[J]. 工程地球物理学报, 2006, 3(3): 172-176)

[16] Xiang Liang. Recommended System Practice [M]. Beijing: Posts & Telecom Press, 1998 (in Chinese)
(项亮. 推荐系统实践[M]. 北京: 人民邮电出版社, 2012: 72-73)

[17] Koren Y. Factor in the neighbors: Scalable and accurate collaborative filtering [J]. ACM Trans on Knowledge Discovery from Data, 2010, 4(1): 1-24

[18] Karypis G. Evaluation of item-based Top-N, recommendation algorithms [C] //Proc of the 10th Int Conf on Information and Knowledge Management. New York: ACM, 2001: 247-254

Yan Cairong, born in 1978. PhD of Xi'an Jiaotong University. Associate professor and MS supervisor. Member of China Computer Federation. Her main research interests include parallel computing, distributed system and big data analyzing.

Zhang Qianglong, born in 1990. MSc. His main research interests include concentrate on personalized recommender, data mining and deep learning.

Zhao Xue, born in 1992. MSc. Her main research interests include concentrate on social network analyzing and data mining.

Huang Yongfeng, born in 1971. PhD of Shanghai Jiaotong University. Associate professor and MS supervisor. His main research interests include pattern recognition, Internet of things and big data analyzing.