计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.190900016

基于牛顿法的自适应高阶评分距离推荐模型研究

邹海涛　郑　尚　王　琦　于化龙　高　尚
江苏科技大学计算机学院　江苏 镇江 212003
(nkroben@outlook.com)

摘　要　现有的一些算法引入了隐语义模型克服数据稀缺带来的问题,为用户提供更有效的推荐。一般情况下,这些方法通过线性组合若干多项式,引入相应参数平衡各个部分比重,以构造优化函数,最终达到最小评分误差或实现最大的偏好等目的。经典模型通常只考虑用户对某一产品的预测评分和实际评分差异(即,一阶评分距离),忽略了其在不同产品上的预测评分与实际评分之间的差值(即,二阶评分距离)。因此,高阶评分距离模型同时将两种距离集成到算法之中,并使用随机梯度下降法求解目标函数。可是,上述优化函数中的相关参数往往是手动设置,而且随机梯度下降法求解目标函数的收敛速度较慢,使得该模型缺乏灵活性,也增加了时间消耗。为了提高模型的适应性和效率,文中提出了一种融合归一化函数的自适应高阶评价距离模型,并利用牛顿法求解改进后的高阶评分距离凸优化函数。此方法不仅移除了若干静态参数,而且加快了优化函数的收敛速度。提出的模型具有坚实的理论支持,经过3个实际数据集的实验结果表明,此模型具有较好的预测精度和运行效率。

关键词:推荐系统;隐语义模型;牛顿法;凸优化;高价评分距离
中图法分类号　TP181

Adaptive High-order Rating Distance Recommendation Model Based on Newton Optimization

ZOU Hai-tao,ZHENG Shang,WANG Qi,YU Hua-long and GAO Shang
School of Computer Science,Jiangsu University of Science and Technology,Zhenjiang,Jiangsu 212003,China

Abstract　Some existing recommendation algorithms introduce latent factor model to overcome the problems caused by data scarcity,so as to provide more effective recommendations for users. In general,those methods construct an optimization function to achieve the minimum rating error or maximum preference,etc,by integrating several polynomials with the corresponding parameters to balance each part,and use stochastic gradient descent to solve this function. Nevertheless,the above mentioned models only consider the difference between the estimated and real ratings of the same user-item pair (i.e.,the first-order rating distance),and ignore the difference between the estimated and real ratings of the same user across different items (i.e.,the second-order rating distance). Hence,high-order rating distance model,HoORaYs,with good accuracy in terms of item ranking and predictive ratings which takes these two kinds of distances into account is proposed. Unfortunately,this model still has some flaws in adaptability and efficiency due to its manually setting parameters,its non-convergence. Aiming at improving the recommendation adaptability and efficiency,an adaptive high-order rating distance model which integrates a data scale sensitive function is proposed. It utilizes Newton method to solve the convex optimization problem about rating distance. This method not only eliminates manually setting parameters,but also accelerates the optimization function convergence speed. The proposed model has a solid theoretical support. Experiments on three real datasets show that,it has good prediction accuracy and operation efficiency.

Keywords　Recommender systems,Latent factor model,Newton method,Convex optimization,High-order rating distance

1　引言

推荐系统已经在电子商务领域中应用得十分普遍,其中所涉及的算法大部分是基于用户的个人信息为其生成推荐商品列表。目前在这些算法中,针对隐语义模型的探讨比较多。这类模型使用随机梯度下降技术处理损失函数,自动地填充用户评分缺失。文献[1]尝试将用户产品矩阵分解成两个低维矩阵的乘积形式,以减小用户对产品的真实评分与预测评分的误差。但该模型需要为所有产品计算用户预测评分,并且预测评分还需按照数值排序,严重影响了算法的效率。文献[2]试图将用户潜在向量与产品潜在向量转化成二进制编码,基于这种编码计算用户相似度,并利用相似用户推荐产品。然而,用户相似度与用户对产品的偏好并不等价,这使得该算法的准确度受到很大影响。为了解决上述问题,文献[3]设计了一种能把产品偏好转化成二进制数值的映射函数。与文献[2]相比,该算法的精确度有所提升,但推荐结果的准确度仍然不够理想。文献[4]对稀疏用户生成的数据构造目标函数,该函数能同时对异构数据源构造的耦合张量和矩阵进

基金项目:江苏省高等学校自然科学研究基金(18JBK520011);镇江市重点研发计划-社会发展(SH2019021)
This work was suppored by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province, China (18JBK520011) and Primary Research & Developement Plan (Social development) of Zhenjiang City (SH2019021).
通信作者:高尚(gao_shang@sohu.com)

行因式分解。文献[5]在评论文本上应用了主题模型,从不同方面对用户偏好和产品特征进行建模,估计用户对产品的兴趣。

总体来说,上述模型只考虑了用户对某一产品的预测评分和实际评分差异(即,一阶评分距离),忽略了其在不同产品上的预测评分与实际评分之间的差值(即,二阶评分距离)。文献[6]同时将两种距离集成到算法之中,提出了高阶评分距离模型的概念,并且使用静态参数 λ_d 平衡一阶评分距离与二阶评分距离的权重。可是,λ_d 是通过实验分析选择的静态数值,限制了模型的灵活性。此外,该模型使用随机梯度下降法对用户潜在向量和产品潜在向量进行迭代更新,其中引入的学习率参数需要手动设定,导致其收敛速度不够稳定。

基于文献[6],本文提出了一种自适应高阶评分距离模型。本模型设计了基于产品评分变化的归一化函数,替代了原本的静态参量。同时本模型没有引入学习率,而是使用牛顿法解决推荐系统中的凸优化问题。该方法可以在不调整任何参数的情况下,适应任意大小的数据集,并且收敛速度极快。

2 相关工作

假设用 m 和 n 分别表示用户数目和产品数目,其对应的用户和产品潜在向量分别表示为 $\boldsymbol{U}=\{\boldsymbol{u}_i\}_{i=1}^{m}$ 和 $\boldsymbol{V}=\{\boldsymbol{v}_j\}_{j=1}^{n}$,它们的维度设为 d。$m\times n$ 矩阵 $\boldsymbol{R}=\{r_{ij}\mid r_{ij}\in[1,r_{\max}]\}$ 表示用户产品矩阵,其中 $r_{i,j}$ 是用户 i 对产品 j 的喜好评分,$r_{\max}$ 表示用户所能评分的最大值。如果 i 没有对 j 评分,则 $r_{i,j}$ 为空。推荐技术的目的就是预测并填充 $\boldsymbol{R}$ 中的空白。

2.1 隐语义模型:矩阵分解模型

基于协同过滤技术[7-8]的矩阵分解模型在推荐技术中应用的十分普遍,该模型通过最小化一阶评分距离来预测产品评分,如式(1)所示:

$$\underset{\boldsymbol{U},\boldsymbol{V}}{arg\,min}\sum_{(i,j,r_{ij})\in D_T}(r_{ij}-\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v}_j)^2 \tag{1}$$

其中,D_T 表示训练集,(i,j,r_{ij}) 是数据集中的数据条目。

为了避免过拟合,通常在式(1)中加入特征向量范数的平方作为正则化项,使其变化为式(2)。其中 λ 为实验得出的静态参数,此公式可通过随机梯度下降法[9]迭代更新 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 实现。

$$\arg\min_{\boldsymbol{U},\boldsymbol{V}}\sum_{(i,j,r_{ij})\in D_T}(r_{ij}-\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v}_j)^2+\lambda(\|\boldsymbol{u}_i\|^2+\|\boldsymbol{v}_j\|^2) \tag{2}$$

2.2 高阶评分距离优化模型:HoORaYs

与矩阵分解模型不同,HoORaYs 模型[6]引入了高阶评分距离准则,在最小化一阶评分距离的同时,还要最小化二阶评分距离。假设用户 i 购买的产品集合为 S_i,购买产品 j 的用户集合表示为 S_j,则用户 i 的二阶评分距离可以通过式(3)计算得到,产品 j 的二阶评分距离可以通过式(4)计算得到,其中 $\sigma(x,y)=1/(1+\mathrm{e}^{-(x-y)})$ 是归一化函数,用于调整计算结果。

$$\sum_{k\in S_i,k\neq j}[\sigma(\boldsymbol{u}_i\boldsymbol{v}_j{}^{\mathrm{T}},r_{ik})-\sigma(r_{ij},r_{ik})]^2 \tag{3}$$

$$\sum_{k\in S_j,k\neq i}[\sigma(\boldsymbol{u}_i\boldsymbol{v}_j{}^{\mathrm{T}},r_{kj})-\sigma(r_{ij},r_{kj})]^2 \tag{4}$$

上述公式反映了如下特点:如果用户或产品具有相同的评分距离,那么它们对计算其他用户或产品评分距离的贡献是相等的。因此,对可以用户和产品的二阶评分距离计算进行整合,如式(5)所示。其中,$|\Omega_{r_{ij}},r|$ 是评分数值为 r 的用户数目(即所有为产品 j 评分为 r 的用户数目)及产品数目(即所有获得用户 i 对其评分为 r 的产品数目)之和。

$$\sum_{(i,j,r_{ij})\in D_T}\sum_{r=1}^{r_{\max}}|\Omega_{r_{ij}},r|[\sigma(\boldsymbol{u}_i\boldsymbol{v}_j^{\mathrm{T}},r)-\sigma(r_{ij},r)]^2 \tag{5}$$

最后,HoORaYs 的优化函数表示为式(6)。其中的第一行为一阶评分距离优化函数,第二行为二阶评分距离优化函数,λ_d 用于平衡二阶评分距离模块的影响。与矩阵分解模型相比,HoORaYs 模型提高了产品评分的预测精度。

$$\begin{aligned}\underset{\boldsymbol{U},\boldsymbol{V}}{\operatorname{argmin}}&\sum_{(i,j,r_{ij})\in T}(r_{ij}-\boldsymbol{u}_i\boldsymbol{v}_j^{\mathrm{T}})^2+\lambda_u\sum_{i=1}^{m}\|\boldsymbol{u}_i\|^2+\lambda_v\sum_{j=1}^{n}\|\boldsymbol{v}_j\|^2+\\&\lambda_d\sum_{(i,j,r_{ij})\in T}\sum_{r=1}^{r_{\max}}|\Omega_{r_{ij}},r|\,[\sigma(\boldsymbol{u}_i\boldsymbol{v}_j^{\mathrm{T}},r)-\sigma(r_{ij},r)]^2\end{aligned} \tag{6}$$

3 问题陈述

3.1 HoORaYs 模型的不足之处

本文在前期实验中使用了 3 个不同量级的 MovieLens 稳定数据集对 HoORaYs 模型仿真,其预测评分的平均绝对误差 MAE 随迭代次数的变化如图 1 所示(实验中已预先设置合适的 λ_d 值)。

MovieLens1m,MovieLens10m,MovieLens20m 数据集分别含有 100 万条,1000 万条,及 2000 万条用户评分数据。总体趋势表明,在 λ_d 固定的情况下,高阶评分距离模型的误差随数据规模的增大而变大。其规范性表述如图 1 所示。

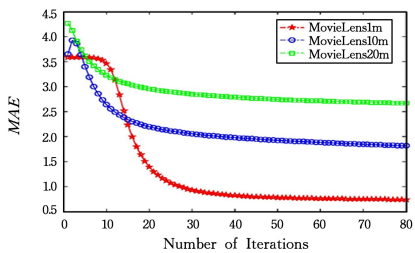


图 1 高阶评分距离模型的 *MAE* 数值比较

Fig. 1 *MAE* comparison of HoORaYs

命题 1 假设 λ_d 数值恒定,那么 HoORaYs 模型不收敛。

证明:根据式(6),随着数据规模增大,$\sum_{r_{ij}}\sum_{r=1}^{r_{\max}}|\Omega_{r_{ij}},r|[\sigma(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v}_j,r)-\sigma(r_{ij},r)]^2$ 也逐渐增大。若将 $[\sigma(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v}_j,r)-\sigma(r_{ij},r)]^2$ 视为在 $(1,r_{\max})$ 这一区间的常量(用 k 表示),$|\Omega_{r_{ij}},r|$ 根据定义会随着数据规模增大而呈一定增长趋势。假设 $x=|\Omega_{r_{ij}},r|$,那么函数 $f(x)=|\Omega_{r_{ij}},r|[\sigma(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v}_j,r)-\sigma(r_{ij},r)]^2=kx$,当 $x\to+\infty$ 时,$\lim_{x\to+\infty}f(x)=+\infty$。因此,二阶评分距离函数不收敛,进而高阶评分距离模型不收敛。

为了进一步研究高阶评分距离模型中各模块的变化趋势,前期实验在上述 3 个数据集上分别计算并比较模型中一阶评分距离与二阶评分距离的数值,其结果如图 2 所示(为使结果对比清晰,实验中对两种距离的数值取对数,由于篇幅限制,这里仅引用了 MovieLens1m 的实验结果,MovieLens10m 和 MovieLens20m 数据集上结果均有相似的表现)。

实验数据表明,随着数据规模不断增大,二阶评分距离的数值量级远远超过一阶评分距离且在模型中占主导地位。为保证预测评分的准确性,需降低二阶评分距离模块的比重,即进一步缩小 λ_d 的量级,如从 0.01 减为 0.001。此外,HoORaYs 模型使用随机梯度下降法更新潜在向量 $\boldsymbol{U}$ 和 $\boldsymbol{V}$,不可避免的引入了学习率参数 α。而 α 的不适当设置可能导致目标

函数不能收敛，或者拖慢其收敛速度。

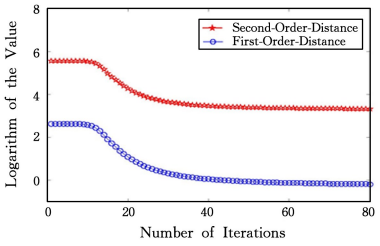


图 2 一阶评分距离与二阶评分距离的数值对比

Fig. 2 Numerical comparison between first-order distance and second-order distance

由于凸优化问题总是可以求解，在推导 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 的过程中，若将一者固定更新另一者，式(6)则可变为凸优化问题，这使得该目标函数能被有效解决。这一结论的详细证明可以参见文献[6]。

3.2 问题定义

本文要解决的问题来自两个方面：1)如何设计一个数据敏感的函数 Λ，使优化函数中的二阶评级距离模块的数值不随数据规模的增大而无限增加，仍然可以起到有效的辅助作用；2)如何在求解推荐域的凸优化问题时，找到一种方法代替随机梯度下降法，以避免学习率 α 的影响，且保持较好的推荐预测精度。

4 自适应高阶评分距离模型设计

为了实现对数据敏感函数的设计，可以直观地构造一个包含 $|\Omega_{r_{ij}},r|$ 和一个规范化过程的函数。同时，由于牛顿法具有收敛速度快、无需静态参数等优点，非常适合求解凸优化问题，所以本文将从这两个方面解决目标问题。

4.1 二阶评分距离模块的归一化设计

根据前述讨论的结果，$|\Omega_{r_{ij}},r|$ 是导致高阶评分距离模型不收敛的原因，以至于 λ_d 必须不断减小数值来保证模型的准确性。那么构建函数 $\Lambda=\lambda_d\cdot|\Omega_{r_{ij}},r|$，如果 Λ 恒定，高阶评分距离模型就可以收敛。由于 $|\Omega_{r_{ij}},r|$ 是用户信息相关的参数，其数值域不受限制，因此，Λ 需进行归一化操作。本文中按式(7)进行计算，其中 $\max(|\Omega_{r_{ij}},r|)$ 为 $|\Omega_{r_{ij}},r|$ 在统计数据中出现的最大值。

$$\Lambda=\frac{|\Omega_{r_{ij}},r|}{\max(|\Omega_{r_{ij}},r|)} \tag{7}$$

原本的高阶评分距离对应的目标函数则变为式(8)，其中 $\boldsymbol{u}_i\in\boldsymbol{U},\boldsymbol{v}_j\in\boldsymbol{V}$ 对应的是用户 i 和产品 j 的潜在向量。由于 Λ 恒定，改进后的模型不仅保持原模型的辅助效果，而且具有收敛性。

$$\begin{aligned}\mathcal{L}(\boldsymbol{U},\boldsymbol{V})=&\sum_{(i,j,r_{ij})\in T}(r_{ij}-\boldsymbol{u}_i\boldsymbol{v}_j^{\mathrm{T}})^2+\lambda_u\sum_{i=1}^{m}\|\boldsymbol{u}_i\|^2+\lambda_v\sum_{j=1}^{n}\\&\|\boldsymbol{v}_j\|^2+\sum_{(i,j,r_{ij})\in T}\sum_{r=1}^{r_{\max}}\frac{|\Omega_{r_{ij}},r|)}{\max(|\Omega_{r_{ij}},r|)}[\sigma\\&(\boldsymbol{u}_i\boldsymbol{v}_j^{\mathrm{T}},r)-\sigma(r_{ij},r)]^2\end{aligned} \tag{8}$$

4.2 凸优化问题的解法设计

随机梯度下降是解决推荐领域优化问题的经典方法[10-12]，如果使用该方法解决式(8)，对于每条训练数据 (i,j,r_{ij})，其用户特征向量 $\boldsymbol{u}_i$ 和产品特征向量 $\boldsymbol{v}_j$ 可以通过以下方法进行计算(见式(9))。其中 α 为学习率，$\nabla\boldsymbol{u}_i$ 和 $\nabla\boldsymbol{v}_j$ 是 $\boldsymbol{u}_i$ 和 $\boldsymbol{v}_j$ 对应的一阶偏导。

$$\begin{aligned}&\boldsymbol{u}_i\leftarrow\boldsymbol{u}_i-\alpha\nabla\boldsymbol{u}_i\\&\boldsymbol{v}_j\leftarrow\boldsymbol{v}_j-\alpha\nabla\boldsymbol{v}_j\end{aligned} \tag{9}$$

然而，α 往往是手工设置的，是根据实验结果或研究人员的主观意见来选择的。另外，对偶问题中目标函数值的减小有时并不意味着原始目标函数值的减小，这往往导致收敛速度较慢[13]。

为提高算法灵活性以及收敛速度，本文计划使用牛顿法对式(8)求解。与随机梯度下降法相比，此方法可以利用曲率信息获得更好的下降路径，收敛速度更快。

当应用牛顿法时，需要使用海森矩阵的逆矩阵替换式(9)中的 α，具体计算如式(10)所示，其中海森矩阵 $\boldsymbol{H}_{u_i}$ 和 $\boldsymbol{H}_{v_j}$ 是用户潜在向量和产品潜在向量基于式(8)的二阶偏导。

$$\begin{aligned}&\boldsymbol{u}_i\leftarrow\boldsymbol{u}_i-\boldsymbol{H}_{u_i}^{-1}\nabla\boldsymbol{u}_i\\&\boldsymbol{v}_j\leftarrow\boldsymbol{v}_j-\boldsymbol{H}_{v_j}^{-1}\nabla\boldsymbol{v}_j\end{aligned} \tag{10}$$

具体操作如算法 1 所示。Θ_U 和 Θ_V 分别代表用户特征向量梯度和项目特征向量梯度，其中，θ_{u_i} 表示用户 i 的特征向量梯度，θ_{v_j})表示产品 j 的特征向量梯度。s 表示使用牛顿法更新 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 的步长，其初始值设为 1。如果式(8)的值小于其更新后的值，那么就完成更新过程。否则，$s\leftarrow s/2$，并且重新进行更新过程(对应算法 1 的 12－15 行，以及 25－28 行)。

算法 1 使用牛顿法更新 $\boldsymbol{U}$ 和 $\boldsymbol{V}$

Input: Training data D_T

Output: $\mathbf{U},\mathbf{V}$

1. Randomly initialize $\mathbf{U},\mathbf{V}$
2. Initialize $\Theta_{\mathbf{U}}=\{\theta_{\mathbf{u}_i}\}_{i=1}^{m}$
3. Initialize $\Theta_{\mathbf{V}}=\{\theta_{\mathbf{v}_i}\}_{j=1}^{n}$
4. while not converged do
5. procedure STAGE1 //Fix $\mathbf{V}$ and update $\mathbf{U}$
6. Initialize s=1
7. for i=1 to m do
8. Compute $\nabla\mathbf{u}_i$
9. Compute $\mathbf{H}_{\mathbf{u}_i}$
10. $\boldsymbol{\theta}_{\mathbf{u}_i}=\mathbf{H}_{\mathbf{u}_i}^{-1}\nabla\mathbf{u}_i$
11. end for
12. if $\mathcal{L}(\mathbf{U}-\mathbf{s}\cdot\Theta_{\mathbf{U}},\mathbf{V})>\mathrm{L}(\mathbf{U},\mathbf{V})$ then
13. $s=\frac{1}{2}s$
14. goto STAGE1
15. end if
16. endprocedure
17. procedureSTAGE 2 //Fix $\mathbf{U}$ and update $\mathbf{V}$
18. Initialize s=1
19. for j=1 to n do
20. Compute $\nabla\mathbf{v}_j$
21. Compute $\mathbf{H}_{\mathbf{v}_j}$
22. $\boldsymbol{\theta}_{\mathbf{v}_j}=\mathbf{H}_{\mathbf{v}_j}^{-1}\nabla\mathbf{v}_j$
23. end for
24. if $\mathcal{L}(\mathbf{U},\mathbf{V}-\mathbf{S}\cdot\Theta_{\mathbf{V}})>\mathrm{L}(\mathbf{U},\mathbf{V})$ then
25. $s=\frac{1}{2}s$
26. goto STAGE 2
27. end if
28. endprocedure
29. end while
30. return $\mathbf{U},\mathbf{V}$

由于海森矩阵的不可逆性，$\boldsymbol{\theta}_{u_i}$ 无法通过 $\boldsymbol{\theta}_{u_i}=\boldsymbol{H}_{u_i}^{-1}\nabla\boldsymbol{u}_i$ 直

接算得。因此,本文使用共轭梯度法[14]对其求解,具体求法如算法 2 所示。其中 p_k 表示第 $k+1$ 步的共轭方向,α_k 表示第 $k+1$ 步的最佳步长,$maxitier$ 表示最大迭代次数(通常不超过向量维数,即 $maxitier<d$,类似的操作同样适用于求解 $\boldsymbol{\theta}_{v_j}$。

算法 2 使用线性共轭梯度法求解 $\boldsymbol{\theta}_{u_j}$

Input:$\mathbf{H}_{\mathbf{u}_i}$,$\nabla\boldsymbol{\theta}_{\mathbf{u}_i}$

Output:θ_{K+1}

1. Initialize $\theta_0=0$
2. $r_0=\mathbf{H}_{\mathbf{u}_i}\theta_0-\nabla\mathbf{u}_i$,$p_0=-r_0$
3. for k=0 to maxitier do
4. $q=\mathbf{H}\cdot\mathbf{p}_k$
5. $\alpha_k=-r_k^{\mathrm{T}}p_k/p_k^{\mathrm{T}}q$
6. $\theta_{k+1}=\theta_k+\alpha_k p_k$
7. $r_{k+1}=r_k+\alpha_k q$
8. if $\|\mathbf{r}_{k+1}\|_2<\|\mathbf{r}_0\|_2\cdot10^{-2}$ then
9. break
10. end if
11. $\beta_{k+1}=-r_{k+1}^{\mathrm{T}}q/p_k^{\mathrm{T}}q$
12. $p_{k+1}=-r_{k+1}+\beta_{k+1}p_k$
13. end for
14. return θ_{K+1}

5 模型的合理性分析

5.1 引入归一化函数的合理性分析

在高阶评分距离模型中,一阶评分距离模块起主要作用,二阶评分距离模块是改进项。假设 $h=(\sigma(\boldsymbol{u}_i\boldsymbol{v}_j^{\mathrm{T}},r)-\sigma(r_{ij},r))^2$,这里设置 $r=3$,并且假设预测评分与实际评分之差为 0.5,即 $\boldsymbol{u}_i\boldsymbol{v}_j^{\mathrm{T}}-r_{ij}=0.5$,那么可获得 h 关于 r_{ij} 的变化趋势(见图 3)。

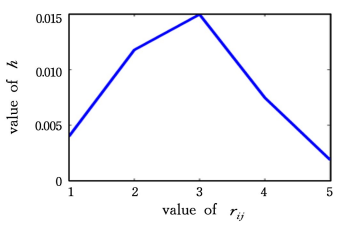


图 3 二阶评分距离数值关于 r_{ij} 的变化趋势

Fig. 3 Variation of second-order distance about with r_{ij}

这一趋势表明,h 在 $r_{ij}=3$ 时达到峰值;$|r_{ij}-r|$ 越大,h 值就越小。因此,使用 $\frac{|\Omega_{r_{ij}},r|}{\max(|\Omega_{r_{ij}},r|)}$ 替换 $\lambda_d\cdot|\Omega_{r_{ij}},r|$(从式(7)变为式(8)),避免了二阶评分距离模块增加到无穷大的情况,并保持了用户和产品对评分数值的偏好比例。

5.2 使用牛顿法的合理性分析

本节主要证明利用牛顿法求解式(8)的优越性,其规范性表述如下。

命题 2 在求解凸优化问题时,牛顿法比随机梯度下降法收敛速度快。

证明:在求解凸优化问题时,随机梯度下降法用平面拟合当前局部曲面,牛顿法采用二次曲面拟合当前局部曲面。因此,与随机梯度下降法相比,牛顿法可以找到更好的最优体面路径。具体地说,当考虑一元函数时,使用牛顿法之后,目标函数 $f(x)$ 可由二阶泰勒公式展开,如式(11)所示:

$$f(x)=f(x_n+\Delta x)\approx f(x_n)+f'(x_n)\Delta x+\frac{1}{2}f''(x_n)\Delta x^2 \tag{11}$$

其中,x_n 是当前点,$x_n+\Delta x$ 是需要找到的最优点。让式(11)对 $\Delta x=0$ 求导,即可得到式(12),于是 $\Delta x=\frac{-f'(x_n)}{f''(x_n)}$,$x_{n+1}=x_n+\Delta x=x_n-\frac{f'(x_n)}{f''(x_n)}$;当使用随机梯度下降法求解一元函数时,$\Delta x=-\alpha f'(x_n)$,其中 α 表示学习率。通过对比,牛顿法将学习率转化为自适应参数,利用函数的二阶导数计算得到,提高了迭代过程的稳定性,加快了收敛速度[15]。

$$\begin{aligned}0&=\frac{d}{d\Delta x}(f(x_n)+f'(x_n)\Delta x+\frac{1}{2}f''(x_n)\Delta x^2)\\&=f'(x_n)+f''(x_n)\Delta x\end{aligned} \tag{12}$$

当考虑多元函数时,式(11)中的变量可以视为向量 $\boldsymbol{X}=(x_1,\cdots,x_n)^{\mathrm{T}}$。使用牛顿法之后,$\boldsymbol{X}_{n+1}=\boldsymbol{X}_n-[\boldsymbol{H}(f(\boldsymbol{X}_n))]^{-1}\nabla f(\boldsymbol{X}_n)$,其中 $\nabla f(\boldsymbol{X}_n)=\left(\frac{\partial f}{\partial x_1},\cdots,\frac{\partial f}{\partial x_n}\right)^{\mathrm{T}}$,$\boldsymbol{H}(f(\boldsymbol{X}_n))$ 是由 $f(\boldsymbol{X}_n)$ 二阶偏导构造的海森矩阵(见式(13))。在使用随机梯度下降法求解多元函数时,$\Delta x=-\alpha\nabla f(\boldsymbol{X}_n)$。同样,与随机梯度下降法相比,牛顿法仍然可以找到更好的收敛路径。

$$\boldsymbol{H}(f(\boldsymbol{X}_n))=\begin{bmatrix}\frac{\partial^2 f}{\partial x_1^2}&\cdots&\frac{\partial^2 f}{\partial x_1\partial x_n}\\\vdots&\ddots&\vdots\\\frac{\partial^2 f}{\partial x_n\partial x_1}&\cdots&\frac{\partial^2 f}{\partial x_n^2}\end{bmatrix} \tag{13}$$

式(8)中的优化问题是固定 $\boldsymbol{U}$ 更新 $\boldsymbol{V}$,或固定 $\boldsymbol{V}$ 更新 $\boldsymbol{U}$,该问题是一个凸优化问题[6]。因此,用牛顿法求解函数(8)收敛速度要快于随机梯度下降法。

5.3 模型的时间复杂度分析

本文提出的自适应高阶评分距离模型的优化过程由两个步骤组成:1)计算一阶偏导向量和海森矩阵;2)更新向量 $\boldsymbol{U}$ 和 $\boldsymbol{V}$。

由于 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 的维数通常较小,那么用共轭梯度法计算的海森矩阵是常数级的。由此可以计算:步骤 1)的时间复杂度为 $O(|R|+c|R|)$,其中 c 为常数,$c\ll|R|$;步骤 2)中更新 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 的时间复杂度与用户及产品数量成正比,小于 $O(|R|)$。所以,本文提出的模型的总时间复杂度为 $O(|R|)$。

鉴于 HoORaYs 模型的时间复杂度与数据集大小成正比,即 $O(|R|)$,因而本文提出的自适应高阶评分距离模型具有与 HoORaYs 相同的时间复杂度,但收敛速度更快,适应性和灵活性更好。

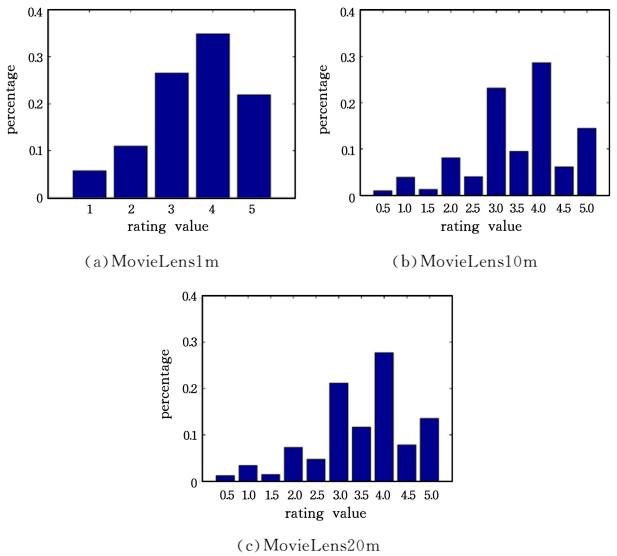
6 实验结果及讨论

6.1 数据集

本文涉及的所有实验均在 Microsoft Windows Server 2012 的四核 CPU (Intel i5,2.4 GHz)、32.0 GB 内存和 3T 硬盘的 PC 上实现,主要从算法的速度、精确度等方面对比本文提出的自适应高阶平分距离模型与 HoORaYs 模型[6]的运行效果。实验中选定了 3 个稳定的 MovieLens 数据集[1)],其中 MovieLens 1M(实验中记为 MovieLens1m)包含 6 040 名用户,3 952 部电影,100 万条电影评分,评分数值为 1−5 的整

[1)] https://grouplens.org/datasets/movielens/

数；MovieLens 10M(实验中记为 MovieLens10m)包含 71 567 名用户，65 133 部电影，以及 1 000 万条电影评分；MovieLens 20M(实验中记为 MovieLens20m)包含 138 493 名用户，131 262 部电影，以及 2 000 万条电影评分。MovieLens10m 和 MovieLens20m 中的评分值范围为 0.5～5，最小评分单位为 0.5。各个数据集的评分数值分布如图 4 所示。可以看出，评分数值百分比随着评分数值在增长，到评分为 4 时，百分比增长到最高，之后开始回落。由于其总体规律类似于正态分布的变化趋势，因此将其看作数值为 4 的正态分布。



(a) MovieLens1m (b) MovieLens10m

(c) MovieLens20m

图 4 MovieLens 数据集的评分分布

Fig. 4 Rating distribution of MovieLens datasets

6.2 测试指标

推荐算法的衡量指标主要来自两个方面：1)预测评分的准确性；2)预测用户购买行为的准确性，即预测用户会购买的产品。在预测产品评分时，本文选用平均绝对误差(Mean Absolute Error，MAE)及标准差(Root Mean Square Error，RMSE)进行指标检测，其计算方法如式(14)和式(15)所示。其中，$error_i=|R^*_{i,k}-R_{i,k}|$，是真实评分与预测评分的误差绝对值。一般来说，良好的推荐算法其 *MAE* 和 *RMSE* 值都会很小。

$$MAE=\frac{\sum_{r_{ij}\in D}|\hat{r}_{ij}-r_{ij}|}{|T|} \tag{14}$$

$$RMSE=\sqrt{\frac{\sum_{r_{ij}\in D}(\hat{r}_{ij}-r_{ij})^2}{|T|}} \tag{15}$$

在预测用户购买行为时，本文选用归一化累积折扣信息增益(normalized Discounted Cumulative Gain，*n*DCG)检测推荐模型的排序质量。假设推荐的产品集合为 $j_1,\cdots,j_l$，该列表中排名第 k 的产品对目标用户 u 的折扣信息增益(DGC)以及 *n*DCG 可以分别通过式(16)和式(17)计算得到。*nDCG* 越大，表示推荐算法越好。

$$nDCG@K=\frac{DCG@K}{iDCG@K} \tag{16}$$

$$DCG@K=\sum_{i=1}^{k}\frac{2^{r(i)}-1}{\log_2(i+1)} \tag{17}$$

在本文的实验中，主要计算 *nDCG*@10，因此用于测试的用户须至少对 10 个产品进行评分。为了使实验结果更加清晰，实验中剔除了评分项数少于 50 的用户，并保留用户中的 80%作为训练集，其余 20%作为测试集。

6.3 参数选取

根据式(6)，HoORaYs 模型需要手动设置 4 个参数：λ_u，λ_v，λ_d 和学习率 α。其中 λ_u 和 λ_v 是数据不敏感参数，将它们设置为常量不会影响算法性能。实验中采用与文献[6]相同的设置，即设置 $\lambda_u=0.1$，$\lambda_v=0.1$。由于 λ_d 对数据敏感，前期通过初步实验已为每个数据集选择最优值，MovieLens1m 数据集中设置 $\lambda_d=0.01$；MovieLens10m 数据集中设置 $\lambda_d=0.001$；MovieLens20m 数据集中设置 $\lambda_d=0.0005$。实验中为 HoORaYs 模型设置学习率 $\alpha=0.005$，并将迭代次数设置为 80。

根据式(8)，本文提出的自适应高阶评分距离模型需要设置 λ_u 和 λ_v，由于它们对数据不敏感，所以实验中选择了与 HoORaYs 模型相同的设置，即设置 $\lambda_u=0.1$，$\lambda_v=0.1$。

此外，前期实验表明，随着用户和产品潜在向量的特征维数 d 的增加，预测精度也会提高。但是当 $d>45$ 时，这种差异就消失了。所以在实验中所有潜在向量的维数均设置为 45。

6.4 实验结果讨论

为了突显本文提出的高阶评分距离模型的改进细节，实验中将使用随机梯度下降法解决式(8)的方法记为 NLH，将使用牛顿法解决式(8)的方法记为 NTH，实验中与 HoORaYs 模型(实验中记为 HoO)进行比较，观察本文提出的模型的特点与优势。

首先，对比所有参与的方法达到最大迭代次数时的性能，其结果如表 1－表 3 所列。实验结果显示，HoORaYs，NLH，NTH 都有最好的表现情况。总体来说，3 个模型的表现数值都是相似的。值得注意的是，HoORaYs 模型中的 λ_d 是通过前期实验找到的最优值。当该参数设置不当时，HoORaYs 模型的表现将逊色于 NLH 和 NTH。而且，在更换数据集时，需要重新进行实验进行设定。而本文提出的 NLH 和 NTH 模型是自适应的，它们不需要设置这样的参数，且在不同数据集均表现优秀。

表 1 MovieLens1m 上的实验结果

Table 1 Results on MovieLens1m

Metrics	HoO	NLH	NTH
MAE	**0.743**	0.745	0.753
RMSE	0.941	**0.940**	0.965
nDCG@10	0.795	**0.796**	0.791

表 2 MovieLens10m 上的实验结果

Table 2 Results on MovieLens10m

Metrics	HoO	NLH	NTH
MAE	0.711	0.714	**0.702**
RMSE	0.916	0.918	**0.900**
nDCG@10	**0.798**	0.796	0.785

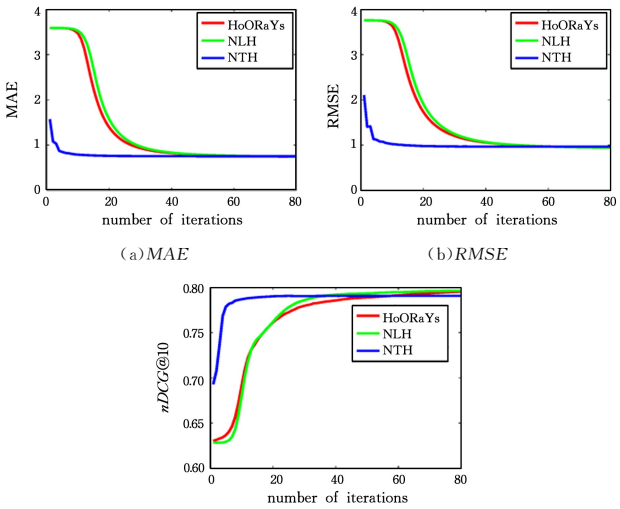
表 3 MovieLens20m 上的实验结果

Table 3 Results on MovieLens20m

Metrics	HoO	NLH	NTH
MAE	0.734	**0.718**	0.721
RMSE	0.944	0.932	**0.926**
nDCG@10	0.787	0.793	0.790

接着，本文对 HoORaYs，NLH 和 NTH 模型的收敛性进行比较，观察各个模型收敛时对应的迭代次数(见图 5，由于篇幅限制，这里仅引用了 MovieLens1m 的实验结果)。结果显示，HoORaYs 模型和 NLH 模型的曲线几乎重合，它们对

MAE,*RMSE* 和 *nDCG*@10 的变化趋势非常相似。这说明本文中提出的归一化处理不仅对不同规模的数据集有良好的适应性,而且在预测评分精度和预测行为精度上保持了良好的性能。



(a)*MAE* (b)*RMSE*

(c)*nDCG*@10

图 5 HoORaYs,NLH 和 NTH 模型的收敛性比较

Fig. 5 Convergence comparison among HoORaYs, NLH and NTH models

具体来说,1)在观察 HoORaYs 和 NTH 在 MAE 和 RMSE 上的表现时:当 HoORaYs 的迭代次数低于 10 时,预测评分精度较差;当该次数大于 10 时,相关评测指标数值下降;当迭代次数达到 50 时,HoORaYs 模型收敛。而 NTH 在预测评分精度上有一个更好的起点,当迭代次数小于 10 时即可收敛。2)在观察 HoORaYs 和 NTH 在 *nDCG*@10 上的表现时:HoORaYs 在 50 次迭代之后收敛,而 NTH 在不到 10 次迭代之后收敛。虽然最后 HoORaYs 比 NTH 稍好一点,但是更快的收敛速度可以保持更高的效率,从这一方面考虑,本文提出的模型更具有优势。

综上所述,本文提出的自适应高阶评分距离模型具有与 HoORaYs 模型相似的精度。使用的归一化操作,以及利用牛顿法求解凸优化问题均能使模型在灵活性和效率方面有明显改进。

结束语 本文通过引入数据归一化函数,提出了一种自适应高阶评分距离模型。利用牛顿法实现了一阶评分距离和二阶评分距离的最小化,加快了收敛速度。该模型具有坚实的理论基础,保持了较好的预测精度。通过对 3 个稳定的 MovieLens 数据集的实验比较,证明了该方法在灵活性和效率方面的优越性。

参 考 文 献

[1] KABBUR S, NING X, KARYPIS G. FISM: factored item similarity models for top-N recommender systems[C]// International Conference on Knowledge Discovery & Data Mining, 2013: 659-667.

[2] ZHOU K, ZHA H. Learning binary codes for collaborative filtering[C]// International Conference on Knowledge Discovery and Data Mining. 2012: 498-506.

[3] ZHANG Z, WANG Q, RUAN L, et al. Preference preserving hashing for efficient recommendation[C]// International Conference on Research and Development in Information Retrieval. 2014: 183-192.

[4] BHARGAVA P, PHAN T, ZHOU J, et al. Who, What, When, and Where: Multi-Dimensional Collaborative Recommendations Using Tensor Factorization on Sparse User-Generated Data [C]// International Conference on World Wide Web. 2015: 130-140.

[5] CHENG Z, YING D, LEI Z, et al. Aspect-aware latent factor model: rating prediction with ratings and reviews[C]// International Conference on World Wide Web. 2018: 639-648.

[6] XU J, YAO Y, TONG H, et al. HoORaYs: High-order optimization of rating distance for recommender systems[C]// Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17). 2017: 525-534.

[7] ZOU H, GONG Z, HU W. Identifying diverse reviews about products[J]. World Wide Web, 2017, 20(2): 351-369.

[8] LIANG D, ALTOSAAR J, CHARLIN L, et al. Factorization meets the item embedding: regularizing matrix factorization with item co-occurrence[C]// International Conference on Recommender Systems. 2016: 59-66.

[9] GEMULLA R, NIJKAMP E, HAAS P J, et al. Large-scale matrix factorization with distributed stochastic gradient descent [C]// International Conference on Knowledge Discovery and Data Mining, 2011: 69-77.

[10] XU C, ZHENG Q, ZHANG Y, et al. Learning to rank features for recommendation over multiple categories[C]// International Conference on Research and Development in Information Retrieval. 2016: 305-314.

[11] GUO G, ZHANG J, YORKE-SMITH N. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings[C]// Twenty-ninth AAAI Conference on Artificial Intelligence. 2015: 123-129.

[12] OH J, HAN W S, YU H, et al. Fast and robust parallel SGD matrix factorization[C]// International Conference on Knowledge Discovery and Data Mining. 2015: 865-874.

[13] WU L, HSIEH C J, SHARPNACK J. Large-scale collaborative ranking in near-linear time[C]// International Conference on Knowledge Discovery and Data Mining. 2017: 515-524.

[14] HESTENES M R, STIEFEL E. Methods of conjugate gradients for solving linear systems[J]. J. Res. Nat. Bur. Stand, 1952, 49(6): 409-436.

[15] RODIN E Y. Nonlinear programming analysis and methods[J]. Computers & Mathematics with Applications, 1977, 3(2): 151-151.

ZOU Hai-tao, born in 1984, Ph.D, lecturer. His main research interests include data mining and information retrieval.

GAO Shang, born in 1972, Ph.D, professor, is a member of China Computer Federation. His main research interests include intelligent computing and pattern recognition.