DOI: 10.11992/tis.201706030
网络出版地址: http://kns.cnki.net/kcms/detail/23.1538.TP.20171109.1250.008.html

稀疏化的因子分解机

郭少成, 陈松灿
（南京航空航天大学 计算机科学与技术学院, 江苏 南京 210016）

摘　要: 因子分解机 (简称为 FM) 是最近被提出的一种特殊的二阶线性模型, 不同于一般的二阶模型, FM 对二阶项系数进行了分解, 这种特殊的结构使得 FM 特别适用于高维且稀疏的数据。虽然 FM 在推荐系统领域已获得了应用, 但 FM 本身并未显式考虑变量的稀疏性, 特别当变量中包含结构稀疏信息时。因此, FM 的二阶特征结构使其特征选择时应当满足这样一种性质, 即涉及同一个特征的线性项和二阶项要么同时被选要么同时不被选, 当该特征是噪音时, 应当同时不被选, 而当该特征是重要变量时, 应当同时被选。考虑到这种结构特性, 本文提出了一种基于稀疏组 Lasso 的因子分解机 (SGL-FM), 通过添加稀疏组 Lasso 的正则项, 不仅实现了组间稀疏, 还实现了组内稀疏。从另一个角度看, 组内稀疏也相当于对因子分解的维度 k 进行了控制, 使其能根据数据的不同而自适应地调整维度 k。实验结果表明, 本文提出的方法在保证了相当精度甚至更优精度的情况下, 获得了比 FM 更稀疏的模型。

关键词: 因子分解机; 稀疏; 稀疏组 Lasso; 特征选择; 推荐系统
中图分类号: TP391　　**文献标志码:** A　　**文章编号:** 1673−4785(2017)06−0816−07

中文引用格式: 郭少成, 陈松灿. 稀疏化的因子分解机[J]. 智能系统学报, 2017, 12(6): 816–822.
英文引用格式: GUO Shaocheng, CHEN Songcan. Sparsified factorization machine[J]. CAAI transactions on intelligent systems, 2017, 12(6): 816–822.

Sparsified factorization machine

GUO Shaocheng, CHEN Songcan
(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: Factorization machine (FM) is a recently proposed second-order linear model. One of its main advantages is that the interactions within it are factorized, making it suitable for data with high dimensionality and high sparsity. Though FM has been applied in recommender systems, it fails to consider the sparsity of variables explicitly, especially when the variable contains information on structural sparsity. Therefore, the process of feature selection should meet the following requirements: the linear terms and second-order terms that share the same feature should be included or excluded at the same time; when the feature is noise, both should be excluded, otherwise, both should be included. Based on the sparse structure described above, this paper proposes a sparse group lasso-based factorization machine (SGL-FM). By adding sparse group lasso to the loss function, SGL-FM not only achieves sparsity between groups but also within groups. From another point of view, sparsity within groups can be seen as a method of controlling the dimensionality of the factorization; therefore, SGL-FM chooses the best k automatically when faced with datasets with different properties. The experimental results show that by applying the proposed method, under conditions of excellent precision, a model with more sparsity than FM was obtained.

Keywords: factorization machine; sparsity; sparse group lasso; feature selection; recommender systems

线性回归模型因为其较好的泛化性能及相对简单快速的求解方法而受到广泛关注, 并已成为统计机器学习中一类最基本的方法[1]。虽然上述线性模型在现实中有广泛应用。但只有当问题的输入呈线性关系时, 它才会有较好的效果。另一方面, 线性模型本身缺乏对输入变量间关系的探索机制。由此

收稿日期: 2017−06−09.　**网络出版日期:** 2017−11−09.
基金项目: 国家自然科学基金项目(61472186).
通信作者: 陈松灿. E-mail: s.chen@nuaa.edu.cn.

自然地转向考虑含有二阶交叉项的线性模型 (此处线性相对参数而言)。含有二阶交叉项的线性模型考虑了输入特征间的相互关系。因此, 当输入数据的特征间呈非线性关系, 特别是二次关系时, 其性能优于一般的线性模型。

推荐系统近年来广受关注[2], 广义上的推荐系统就是给用户推荐物品的系统, 它还可具体到一些特定领域, 如音乐、书籍等。推荐系统的主要任务是根据用户的历史行为记录去预测用户未来可能会购买的物品。从本质上说就是探索用户与用户间以及用户与物品间的关系, 也就是变量与变量间的关系。针对推荐系统, 最近 S. Rendle 等[3]提出了一个新的二阶交叉项模型: 因子分解机 (FM)。在 FM 中, 每个变量 $\boldsymbol{x}_i$ 都对应着一个在隐空间的 k 维的向量 $\boldsymbol{v}_i$, $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 的交叉项系数等于 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 的内积, 当输入数据非常稀疏时, 一般的二阶交叉模型无法学习到有效的交叉项系数。而 FM 分解了交叉项系数, 这个特性使得 FM 能学习到数据中隐藏的变量间相互关系[3-4]。因此, FM 特别适用于稀疏数据。

虽然对交叉项系数进行分解能显著提升模型性能, 但当 k(因子分解维度) 较大时, FM 模型包含大量参数, 为了避免过拟合, 常常需要对目标函数添加正则化项。一种常用的正则化方法是添加 ℓ_2 范数。但是, 对于高维数据, 我们希望选出那些最具判别性的特征。通常的做法是向目标损失函数添加能够诱导稀疏解的正则化项, 通过优化正则化的目标函数来获得稀疏解。比如在线性回归中, 向目标函数添加 ℓ_1 范数的正则项[5], 不仅能防止过拟合, 还能起到特征选择的作用。虽然, ℓ_1 范数能获得稀疏解, 但是, 这种稀疏并不包含结构信息。在 FM 中, 其特征应当满足这样一种性质, 即涉及同一个特征的线性项和二阶项要么同时被选要么同时不被选, 当该特征是噪音时, 应当同时不被选, 而当该特征是重要变量时, 应当同时被选。而 ℓ_1 范数不能利用此先验结构信息。

Group Lasso(GL) 是 M.Yuan 等[6]基于 Lasso 提出的用于对组变量进行特征选择的方法, 与 Lasso 不同的是, GL 能同时选择或者不选组内的所有变量。首先根据先验知识将变量按照相关性划分为不同组, 从聚类角度看就是将同类变量划分为同组, 不同类变量划分为不同组。在 FM 中, 将关于特征 $\boldsymbol{x}_i$ 的线性项系数 $\boldsymbol{\omega}_i$ 和其在隐空间的特征表示向量 $\boldsymbol{v}_i$ 分在同组中, 这样, GL 就能保证当 $\boldsymbol{x}_i$ 是噪音时, $\boldsymbol{\omega}_i$ 和 $\boldsymbol{v}_i$ 同时不选, 反之, 则同时选择。虽然 GL 能实现这种结构稀疏, 但是, 对于选中的组, 并不是所有特征都是有用的。因此, GL 的使用有非常大限制, 有必要继续选择组内重要的特征。

Simon 等[7]在 GL 的基础上提出了基于 Sparse Group Lasso(SGL)的线性回归模型。与 GL 相同的是它们都对变量进行分组, 与 GL 不同的是, SGL 在 GL 的基础上, 继续选择组内重要特征。因此, SGL 能同时实现组间稀疏和组内稀疏, 而 GL 只实现了组间稀疏。SGL 结合了 Lasso 和 GL 的优点, 当待求变量存在结构稀疏信息时, 仅使用 Lasso 会丢失结构信息, 而仅使用 GL 又会导致求得冗余的解。基于上述事实, SGL 既保留了 GL 的结构信息, 又具有 Lasso 的高效特征选择的能力。

从另一角度看, 当输入的数据非常稀疏而 k 选择较大值时, FM 容易过拟合。此时, SGL 的组内稀疏能通过特征选择控制 k 的大小。而且, 不同的特征由于重要程度的不同, 其对应的分解向量 $\boldsymbol{v}$ 的维度也应当不同, 所以, 组内稀疏在一定程度上通过特征选择对不同维度特征自适应了最优的 k 值。

当前虽然已有一些关于 FM 的研究, 如 Mathieu 等[8]在 FM 的基础上进一步提出了高阶因子分解机 (阶数≥3), M. Li 等[9]提出了分布式的 FM 以及 W-S CHIN 等[10]提出了针对二类分类问题的 FM 的优化算法并将其并行化。但是, 它们并没有探索 FM 的稀疏化机制。本文首次针对 FM 的二阶特征结构提出了 SGL-FM, 而且, 本文的方法也可以直接推广到高阶的 FM 中以探索高阶 FM 的稀疏化机制。

1　因子分解机

1.1　目标函数

FM 的基本模型如下:

$$\hat{y}(\boldsymbol{x}) = \omega_0 + \sum_{i=1}^{p} \boldsymbol{\omega}_i \boldsymbol{x}_i + \sum_{i=1}^{p} \sum_{j=i+1}^{p} < \boldsymbol{v}_i, \boldsymbol{v}_j > \boldsymbol{x}_i \boldsymbol{x}_j \tag{1}$$

式 (1) 也可变形为

$$\hat{y}(\boldsymbol{x}) = \omega_0 + \sum_{i=1}^{p} \boldsymbol{\omega}_i \boldsymbol{x}_i + \frac{1}{2} \sum_{f=1}^{k} ((\sum_{i=1}^{p} \boldsymbol{v}_{i,f} \boldsymbol{x}_i)^2 - \sum_{i=1}^{p} \boldsymbol{v}_{i,f}{}^2 \boldsymbol{x}_i{}^2) \tag{2}$$

FM 的目标函数如下:

$$\underset{\omega, V}{\operatorname{argmin}} \sum_{i=1}^{n} \ell(\hat{y}_i, y_i) + \lambda_1 \|\boldsymbol{\omega}\|_2{}^2 + \lambda_2 \|\boldsymbol{V}\|_F{}^2 \tag{3}$$

式中: $\ell(\hat{y}_i, y_i)$ 表示第 i 个样本的损失, $\hat{y}_i$ 是预测的标号值, y_i 是样本的真实标记, λ_1 和 λ_2 均为控制模型复杂度的超参数。FM 用于回归问题时通常采用最小平方损失函数, 因此有

$$\ell(\hat{y}_i, y_i) = (\hat{y}_i - y_i)^2 \tag{4}$$

1.2　优化方法

目前已经有多种基于迭代的优化算法被提出用于优化 FM, 如 MCMC, ALS[12]等。其中最常用的是随机梯度下降法(SGD), 即每次随机选取一个样本

计算损失函数关于变量的梯度, 之后用梯度更新待求变量, 如此迭代便可优化目标函数。

假设Θ为所有待求参数的集合, 而 θ 表示Θ的分量, θ 可以是 ω_0、$\boldsymbol{\omega}_i$ 或者 $\boldsymbol{V}_{ij}$, 则在第 t+1 时刻的更新公式为

$$\theta^{t+1}=\theta^{t}-\eta\left(\frac{\partial}{\partial\theta^{t}}\ell(\hat{y}(\boldsymbol{x}_i|\Theta^{t}),y_i)+2\lambda\theta^{t}\right) \tag{5}$$

式中: η 是学习率, 表示每次梯度更新的步长。对于最小平方损失函数有:

$$\frac{\partial\ell(\hat{y}(\boldsymbol{x}_i|\Theta),y_i)}{\partial\theta}=2(\hat{y}(\boldsymbol{x}_i|\Theta)-y_i)\frac{\partial\hat{y}(\boldsymbol{x}_i|\Theta)}{\partial\theta} \tag{6}$$

将式 (2) 对各个参数求导可得[12]:

$$\frac{\partial}{\partial\theta}\hat{y}(\boldsymbol{x})=\begin{cases}1, & \theta=\omega_0\\ \boldsymbol{x}_i, & \theta=\boldsymbol{\omega}_i\\ \boldsymbol{x}_i(\sum_{j=1}\boldsymbol{x}_j\boldsymbol{v}_{j,f}-\boldsymbol{x}_i\boldsymbol{v}_{i,f}), & \theta=\boldsymbol{v}_{i,f}\end{cases} \tag{7}$$

基于式 (5)~(7), 即可根据给定的样本计算损失函数关于变量的梯度。

2 基于 SGL 的因子分解机

2.1 目标函数

本文通过对损失函数添加 SGL 的正则项以期望得到含有结构稀疏性质的解向量, SGL-FM 的目标函数如下:

$$\underset{\omega,V}{\operatorname{argmin}}\sum_{i=1}^{n}\ell(\hat{y}_i,y_i)+\lambda_1\sum_{i=1}^{p}\left\|\begin{matrix}\boldsymbol{\omega}_i\\ \boldsymbol{v}_i\end{matrix}\right\|_2+\lambda_2\sum_{i=1}^{p}\left\|\begin{matrix}\boldsymbol{\omega}_i\\ \boldsymbol{v}_i\end{matrix}\right\|_1 \tag{8}$$

式中: 将 $\boldsymbol{\omega}_i$ 和$\boldsymbol{v}_i(1\leqslant i\leqslant p)$分为一组, 共分为 p 组, 其中$\left\|[\boldsymbol{\omega}_i\ \boldsymbol{v}_i]^{\mathrm{T}}\right\|_2$表示同时选择或同时不选 $\boldsymbol{\omega}_i$ 和 $\boldsymbol{v}_i$, 实现了组间稀疏。$\left\|[\boldsymbol{\omega}_i\ \boldsymbol{v}_i]^{\mathrm{T}}\right\|_1$表示对选中的 $\boldsymbol{\omega}_i$ 和 $\boldsymbol{v}_i$ 进一步进行特征选择, 实现了组内稀疏, 而组内稀疏也相当于对各个维度自适应选择最优 k 值。值得注意的是, ℓ_2、ℓ_1范数均非光滑, 且损失函数非凸。因此, 目标函数非凸非光滑, 而 FM 的目标函数非凸但光滑, 因此, 优化式 (8) 具有更大的挑战, 不能照搬 FM 的优化方法。在下一节中, 我们给出优化该目标函数的有效算法。实验结果也表明, 该算法能有效收敛。

式 (8) 还包括了另外两个稀疏化模型, 当 $\lambda_1=0$时, 目标函数只有ℓ_1项, 简记该模型为 L1-FM。当$\lambda_2=0$时, 目标函数仅有 Group Lasso 项, 简记该模型为 GL-FM。

2.2 优化方法

因为 L1-FM 和 GL-FM 均为 SGL-FM 的特例, 给出 SGL-FM 的优化方法后, L1-FM 和 GL-FM 的优化方法可以直接得到, 因此本文仅关注 SGL-FM 的优化。FM 可以使用 SGD 算法优化, 但是在 SGL-FM 中, 由于ℓ_2范数和ℓ_1范数在零点不可微, 虽然也可利用次梯度的方法迭代。但是, 直接利用次梯度迭代很少能使变量达到不可微点[11], 也即很少会得到含有零元素的解向量, 而在很多情况下零点才是目标函数的局部最小点。从另一个角度看, 稀疏解能够体现目标变量的结构信息。使用次梯度优化方法得到的结果相悖于我们期望的稀疏结果。所以, 本文引入了前向后向切分算法 (forward backward splitting, FOBOS)[11]来优化该问题。

2.2.1 FOBOS 算法

FOBOS 是一种基于迭代优化的算法框架, 主要用来求解含有正则项的目标函数的优化问题, 特别是一些不可微的正则项如: ℓ_1、ℓ_2、1(Group Lasso)、ℓ_∞等[11], 相比于直接用次梯度计算, 使用 FOBOS 算法得到的模型具有更好的预测性能和更符合问题先验的稀疏结构[11]。

假设待求目标函数由两部分组成: $f(\boldsymbol{\omega})+r(\boldsymbol{\omega})$, 其中$f(\boldsymbol{\omega})$一般为损失函数, 本文中损失函数为最小平方损失, 第 2 项$r(\boldsymbol{\omega})$是关于目标变量 $\boldsymbol{\omega}$ 的正则项。其每次迭代过程分 2 步:

$$\boldsymbol{\omega}^{t+\frac{1}{2}}=\boldsymbol{\omega}^{t}-\eta_t g_f^{t} \tag{9}$$

$$\boldsymbol{\omega}^{t}=\operatorname{argmin}_{\omega}\left\{\frac{1}{2}\left\|\boldsymbol{\omega}-\boldsymbol{\omega}^{t+\frac{1}{2}}\right\|^2+\eta_{t+\frac{1}{2}}r(\boldsymbol{\omega})\right\} \tag{10}$$

式中: g_f^{t} 为在 t 时刻损失函数$f(\boldsymbol{\omega})$关于权重的梯度, η_t为 t 时刻的学习率, $\eta_{t+\frac{1}{2}}$是在式 (10) 迭代中正则项前的系数, 在具体算法实现中, 通常设置$\eta_{t+\frac{1}{2}}=\eta_t$。步骤 1)等价于标准的无正则项梯度下降过程, 式 (10) 中结果是在第一步结果$\boldsymbol{\omega}^{t+\frac{1}{2}}$基础上进行了微调, 一方面希望新的结果尽可能靠近第一步的结果, 另一方面还需要尽可能最小化$r(\boldsymbol{\omega})$。

2.2.2 利用 FOBOS 算法求解 SGL-FM

根据 FOBOS 算法, 首先将 SGL-FM 的目标函数分为两部分:

$$f(\omega_0,\boldsymbol{\omega},\boldsymbol{V})+r(\boldsymbol{\omega}+\boldsymbol{V}) \tag{11}$$

式中: $f(\omega_0,\boldsymbol{\omega},\boldsymbol{V})=\sum_{i=1}^{n}\ell(\hat{y}_i,y_i)$且$r(\boldsymbol{\omega}+\boldsymbol{V})=\lambda_1\sum_{i=1}^{p}\left\|\begin{matrix}\boldsymbol{\omega}_i\\ \boldsymbol{v}_i\end{matrix}\right\|_2+\lambda_2\sum_{i=1}^{p}\left\|\begin{matrix}\boldsymbol{\omega}_i\\ \boldsymbol{v}_i\end{matrix}\right\|_1$。假设输入一个新的样本$(x,y)$, 根据式 (5)~(7)、式 (9) 可知 FOBOS 算法的第 1 步更新公式为

$$\begin{cases}\omega_0^{t+\frac{1}{2}}=\omega_0^{t}-\eta_t\cdot 2(\hat{y}(\boldsymbol{x}|\Theta)-y)\\ \boldsymbol{\omega}_i^{t+\frac{1}{2}}=\boldsymbol{\omega}_i^{t}-\eta_t\cdot 2(\hat{y}(\boldsymbol{x}|\Theta)-y)\cdot\boldsymbol{x}_i\\ \boldsymbol{v}_{i,f}^{t+\frac{1}{2}}=\boldsymbol{v}_{i,j}^{t}-\eta_t\cdot 2(\hat{y}(\boldsymbol{x}|\Theta)-y)\cdot\boldsymbol{x}_i(\sum_{j=1}\boldsymbol{x}_j\boldsymbol{v}_{j,f}-\boldsymbol{x}_i\boldsymbol{v}_{i,f})\end{cases} \tag{12}$$

式中$1\leqslant i\leqslant p$且$1\leqslant f\leqslant k$。

设 $\lambda_1'=\eta_t\lambda_1$, $\lambda_2'=\eta_t\lambda_2$, $\boldsymbol{\theta}_i^{t}=[\boldsymbol{\omega}_i^{t}\ \boldsymbol{v}_i^{t}]^{\mathrm{T}}\in R^{k+1}$, 则 FOBOS 算法的第 2 步等价于优化以下问题:

$$\boldsymbol{\theta}_i^t = \arg\min_{\theta_i}\{\frac{1}{2}\left\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_i^{t+\frac{1}{2}}\right\|^2 + \lambda_1'\|\boldsymbol{\theta}_i\|_2 + \lambda_2'\|\boldsymbol{\theta}_i\|_1\} \quad (13)$$

式中: $1 \leqslant i \leqslant p$。文献[11]中给出了当正则项分别为 ℓ_1、$\ell_{2,1}$、ℓ_∞时相应的求解算法, 但是当正则项为 SGL 时, 文献[11]并没有给出其求解算法, 而且直接将ℓ_1, $\ell_{2,1}$的解法推广到 SGL 是非平凡的。

Liu 等在文献[13]中提出了一种有效算法用于求解含有树结构信息的正则化问题。本文中的 SGL 是其中一种特例。如图 1 所示, SGL 的结构可以表示成 p 棵树, 每棵树的根节点包含了第 i 维特征的一阶系数 $\boldsymbol{\omega}_i$ 和其在隐空间的特征表示向量 $\boldsymbol{v}_i$, 子节点分别是其各个分量, SGL 相当于对树的每个节点都添加了ℓ_2范数的约束。

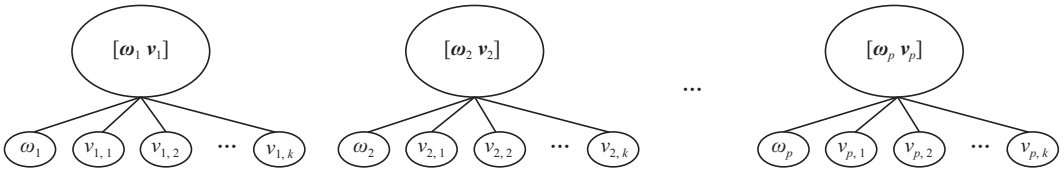


图 1　树结构的 SGL

Fig. 1　SGL can be represented as tree structures

由文献[13], 我们在算法 1 中直接给出了优化式 (13) 的具体流程。并在算法 2 中给出了利用 FOBOS 算法优化 SGL-FM 的完整流程。

算法 1　树结构正则项的优化算法

输入　Step 1 的输出$\boldsymbol{\theta}_i^{t+\frac{1}{2}}$,$(1 \leqslant i \leqslant p)$, λ_1',λ_2';

输出　更新后的参数$\boldsymbol{\theta}_i^t = [\boldsymbol{\omega}_i^t, \boldsymbol{v}_i^t]^{\mathrm{T}} (1 \leqslant i \leqslant p)$。

1) for i = 1: p do
2) $\boldsymbol{\theta}_i = \boldsymbol{\theta}_i^{t+\frac{1}{2}}$
3) for j = 1: k+1
4) if ($|\boldsymbol{\theta}_i[j]| \leqslant \lambda_2'$) then $\boldsymbol{\theta}_i[j] = 0$
5) else $\boldsymbol{\theta}_i[j] = \left(\frac{|\boldsymbol{\theta}_i[j]| - \lambda_2'}{|\boldsymbol{\theta}_i[j]|}\right) \cdot \boldsymbol{\theta}_i[j]$
6) end if
7) end for
8) if $\|\boldsymbol{\theta}_i\|_2 \leqslant \lambda_1'$ then $\boldsymbol{\theta}_i = 0$
9) else $\boldsymbol{\theta}_i = \left(\frac{\|\boldsymbol{\theta}_i\|_2 - \lambda_1'}{\|\boldsymbol{\theta}_i\|_2}\right) \cdot \boldsymbol{\theta}_i$
10) end if
11) end for

算法 2　用于求解 SGL-FM 的 FOBOS 算法

输入　训练数据, 正则项参数λ_1, λ_2;

输出　ω_0, $\boldsymbol{\omega} \in R^p$及$\boldsymbol{V} \in R^{p \times k}$。

1) for k=1:num_epoch % 迭代次数
2) 随机排列所有训练样本
3) for i = 1:num_samples % 遍历所有样本
4) 取出样本 $\boldsymbol{x}_i$
5) 根据式 (12) 执行随机梯度下降
6) 根据算法 1 优化式 (13)
7) end for
8) end for

3　实验与分析

3.1　实验设置与实验数据

为了验证算法的性能, 在 3 个推荐系统数据集上进行了实验, 数据的基本信息如表 1 所示, 其中 Movielens 的两个数据均为电影评分数据, Last.fm 为音乐推荐数据, 所有数据均采用 One-Hot-Encoding 编码方式。本文将所有数据均划分 70% 作为训练集, 30% 作为测试集。

表 1　实验数据

Table 1　Experimental datasets

数据库	样本数	数据维度(user + item)
Movielens 100k	100 000	2 625=943+1 682
Movielens 1m	1 000 209	9 940=6 040+3 900
Last.fm	109 750	22 272=12 523+9 749

实验不仅对比了 SGL-FM、FM、L1-FM 和 GL-FM 等方法。还加入了线性模型 Lasso 和一般的二阶回归模型(SEC-REG)作为基准对比方法。

所有方法的超参数均采用 3 折交叉验证选取。FM、Lasso 以及 SEC-REG 的所有正则化参数均从{0.000 01, 0.000 1, 0.001, 0.01, 0.1, 1}中选取, 而 SGL-FM、L1-FM 和 GL-FM 的超参数均从{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}}中选取。

实验以均方根误差(RMSE)作为评价准则, 其计算公式为

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

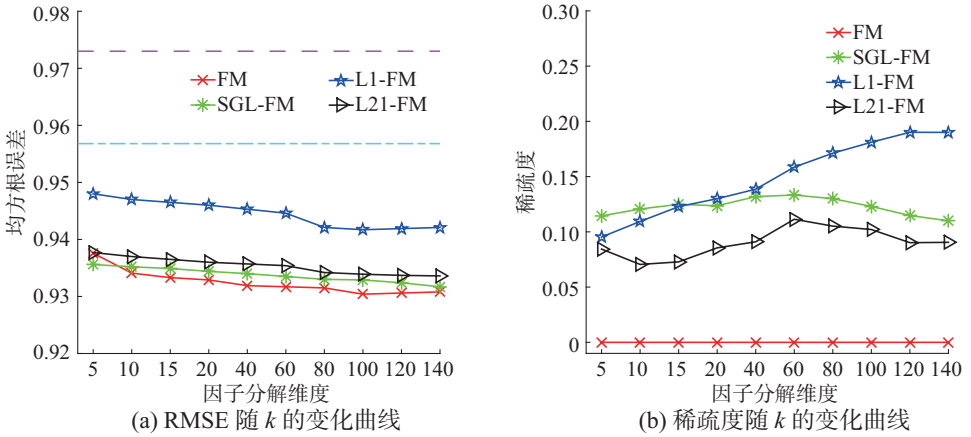
式中: n 为测试样本数, $\hat{y}_i$,y_i分别为第 i 个样本的预测标号和真实标号。实验也比较了各个模型所得系数的稀疏度, 稀疏度的计算方式为

$$\mathrm{sparsity} = \frac{n_z}{n_a}$$

式中: n_a 表示线性项系数$\boldsymbol{\omega} \in R^p$和二阶项系数矩阵$\boldsymbol{V} \in R^{p \times k}$中所有分量的个数, 即$n_a = p(k+1)$; n_z 表示这些分量中零元素个数。

3.2 性能与稀疏度分析

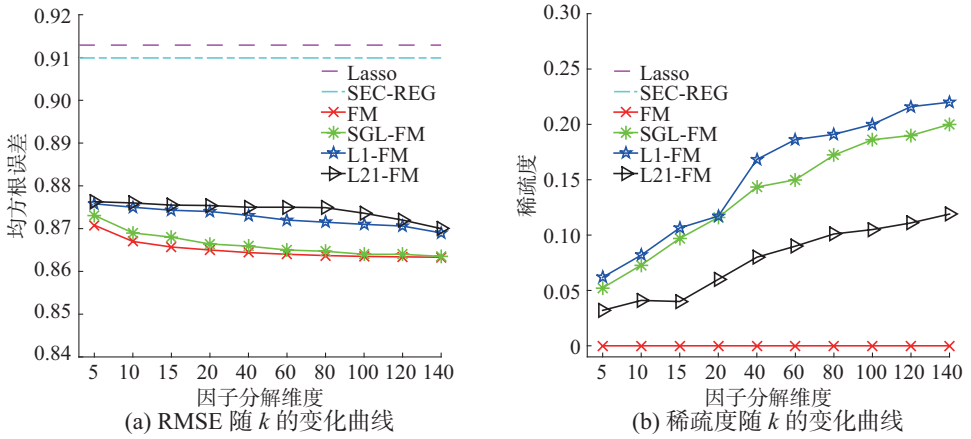
图 2~4 分别比较了各个算法在 3 个数据集上的 RMSE 和稀疏度随着 k 的变化趋势，可以看出，线性模型 Lasso 由于未探索变量间的关系，因此效果较差。而由于数据过于稀疏，二阶模型 SEC-REG 的精度也差于因子分解机类算法。



(a) RMSE 随 k 的变化曲线　(b) 稀疏度随 k 的变化曲线

图 2　Movielens 100 k 实验结果

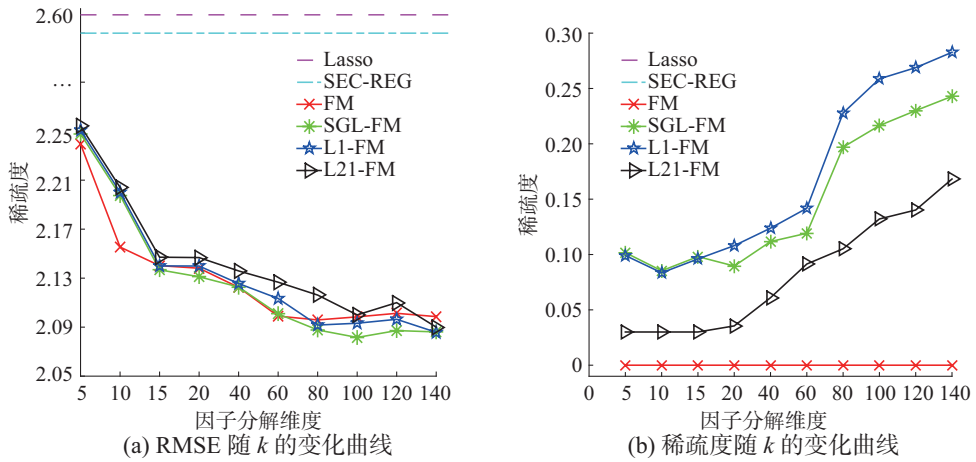
Fig. 2　Results on movielens 100 k



(a) RMSE 随 k 的变化曲线　(b) 稀疏度随 k 的变化曲线

图 3　Movielens 1 m 实验结果

Fig. 3　Fig 2. results on movielens 1 m



(a) RMSE 随 k 的变化曲线　(b) 稀疏度随 k 的变化曲线

图 4　Last.fm 实验结果

Fig. 4　Results on last.fm

比较 SGL-FM 与 FM，可以看出在 Movie-lens 数据集上 SGL-FM 的稀疏度最高达到了 20%，虽然 FM 有更多的参数，但是 SGL-FM 的性能与 FM 的性能非常接近，说明 SGL-FM 能进行有效的特征选

择。在 Last.fm 数据上, 当 k>100 时, SGL-FM 的稀疏度达到了 25%~30%, 虽然 SGL-FM 的参数更少, 但是其性能要优于稀疏度等于 0 的 FM, 说明由于数据各个特征的分布不同, 不同特征有各自最优的 k 值, SGL-FM 通过特征选择为各个维度自适应了最佳的 k 值, 去除了冗余的组内特征。图 5 给出了在 Last.fm 数据集上, 当 k=100 时, SGL-FM 求出的特征表示向量$\boldsymbol{v}_i(1 \leqslant i \leqslant p)$所自适应的 k 值分布, 从图中可以看出, 对于 Last.fm 数据, 大部分特征的 k 值分布在 50~70 之间, 从图 5 中也可以看出, 当 k=60 时, FM 的效果最好, 大于 60 时, FM 的效果开始变差, 这是因为参数过多, FM 发生了过拟合。比较 SGL-FM 与 GL-FM, SGL-FM 由于既实现了组内稀疏又实现了组间稀疏, 因此其性能优于只实现了组间稀疏的 GL-FM。从稀疏度变化中也可以看出, 相比于 GL 包含了太多的冗余组内特征, SGL 能进一步地对组内特征做特征选择, 从而不仅提高稀疏度, 更能提高模型的性能。比较 SGL-FM 与 L1-FM, 由于 L1-FM 不包含结构信息且对所有特征无差别选择, 因此, 虽然 L1-FM 有更高的稀疏度, 但是其性能比 SGL-FM 差。

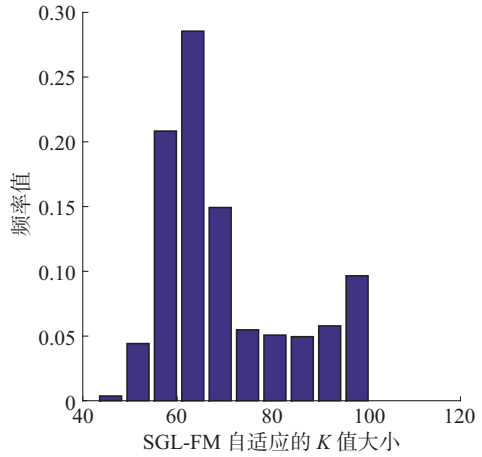


图 5　SGL-FM 在 Last.fm 上自适应的 k 值分布

Fig. 5　The distribution of k selected by SGL-FM

3.3　收敛性分析

当采用随机梯度优化这种算法时, 算法的收敛性是常常需要考虑的问题, 由于 FM 的特殊性, 其目标函数关于待求参数非凸[3], 原始文献[3]中并没有给出收敛性证明, 但是实验结果表明, FM 是收敛的, 图 6 给出了本文提出的算法和 FM 在两个数据集上的迭代过程, 可以看出, 所有算法均稳定收敛。而且本文提出的 SGL-FM, L1-FM 以及 GL-FM 具有更快的收敛速率, 这是由于 SGL-FM 等去除了噪音的干扰, 因而收敛更快。

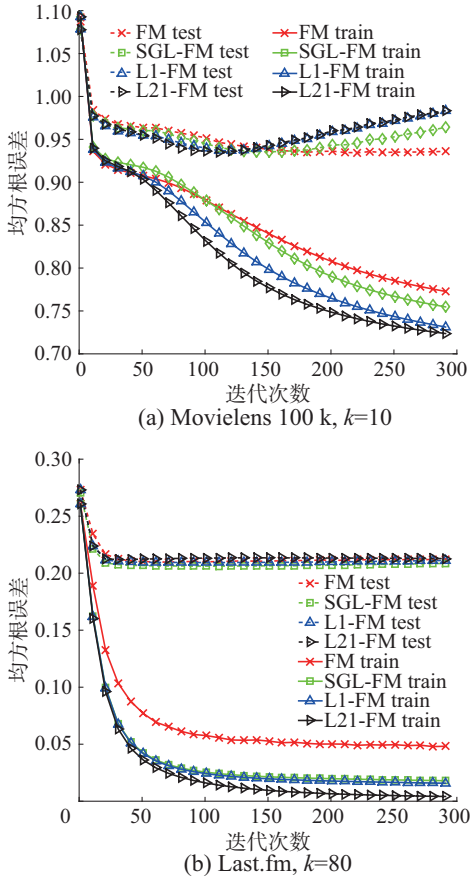


图 6　各算法训练和测试 RMSE 随着迭代次数的变化

Fig. 6　The training RMSE and testing RMSE w.r.t the iteration times

4　结束语

考虑到因子分解机特殊的二阶特征结构, 本文结合了 GL 和 Lasso 的优点, 提出了基于 Sparse Group Lasso 的因子分解机。同时, 作为 SGL-FM 的特例, 我们还导出了 L1-FM 和 GL-FM。不同于一般的二阶模型和一般的 FM, SGL-FM 的目标函数非凸且非光滑, 本文引入了 FOBOS 算法来优化该问题。SGL-FM 不仅获得了比 FM 更稀疏的解, 节省了内存空间, 更能通过去除噪音特征, 从而提升性能, 实验结果也证明了这一点。

参考文献:

[1] RAO C R, TOUTENBURG H. Linear models[M]. New York: Springer, 1995: 3–18.

[2] ADOMAVICIUS G, TUZHILIN A. Context-aware recommender systems[M]. US: Springer, 2015: 191–226.

[3] RENDLE S. Factorization machines[C]//IEEE 10th International Conference on Data Mining. Sydney, Australia, 2010: 995–1000.

[4] RENDLE S. Learning recommender systems with adaptive regularization[C]//Proceedings of the fifth ACM internation-

al conference on Web search and data mining. Seattle, USA, 2012: 133–142.

[5] TIBSHIRANI R. Regression shrinkage and selection via the lasso[J]. Journal of the royal statistical society, Series B (Methodological), 1996, 73(3): 267–288.

[6] YUAN M, LIN Y. Model selection and estimation in regression with grouped variables[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2006, 68(1): 49–67.

[7] SIMON N, FRIEDMAN J, HASTIE T, et al. A sparse-group lasso[J]. Journal of computational and graphical statistics, 2013, 22(2): 231–245.

[8] BLONDEL M, FUJINO A, UEDA N, et al. Higher-order factorization machines[C]//Advances in Neural Information Processing Systems. Barcelona, Spain 2016: 3351–3359.

[9] LI M, LIU Z, SMOLA A J, et al. DiFacto: distributed factorization machines[C]//Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. San Francisco, USA, 2016: 377–386.

[10] CHIN W S, YUAN B, YANG M Y, et al. An efficient alternating newton method for learning factorization machines [R].NTU:NTU,2016.

[11] DUCHI J, SINGER Y. Efficient online and batch learning using forward backward splitting[J]. Journal of Machine Learning Research, 2009, 10(12): 2899–2934.

[12] RENDLE S. Factorization machines with libfm[J]. ACM transactions on intelligent systems and technolog, 2012, 3(3): 57.

[13] LIU J, YE J. Moreau-Yosida regularization for grouped tree structure learning[C]//Advances in Neural Information Processing Systems. Vancouver, Canada, 2010: 1459–1467.

作者简介：

郭少成，男，1993 年生，硕士研究生，主要研究方向为机器学习、模式识别。

陈松灿，男，1962 年生，教授，博士生导师，博士，中国人工智能学会机器学习专委会主任，CCF 高级会员，主要研究方向为模式识别、机器学习、神经计算。在国际主流期刊和顶级会议上发表多篇学术论文并多次获奖。