软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2019,30(3):799−821 [doi: 10.13328/j.cnki.jos.005698]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

因子分解机模型研究综述*

赵衍衍[1,2], 张良富[1,2], 张 静[1,2], 李翠平[1,2], 陈 红[1,2]

[1](中国人民大学 信息学院,北京 100872)

[2](数据工程与知识工程教育部重点实验室(中国人民大学),北京 100872)

通信作者: 李翠平, E-mail: licuiping@ruc.edu.cn

摘 要: 传统矩阵分解方法因其算法的高可扩展性和较好的性能等特点,在预测、推荐等领域有着广泛的应用.然而大数据环境下,更多上下文因素的获取变得可能,传统矩阵分解方法缺乏对上下文信息的有效利用.在此背景下,因子分解机模型提出并流行.为了更好地把握因子分解机模型的发展脉络,促进因子分解机模型与应用相结合,针对因子分解机模型及其算法进行了综述.首先,对因子分解机模型的提出进行了溯源,介绍了从传统矩阵分解到因子分解机模型的演化过程;其次,从模型准确率和效率两方面对因子分解机模型存在的基本问题和近年来的研究进展进行了总结,然后综述了适用于因子分解机模型求解的 4 种代表性优化算法;最后分析了因子分解机模型目前仍存在的问题,提出了可能的解决思路,并对未来的研究方向进行了展望.

关键词: 因子分解机;高阶交互;特征选择;概率模型;凸优化;分布式框架;优化方法

中图法分类号: TP311

中文引用格式: 赵衍衍,张良富,张静,李翠平,陈红.因子分解机模型研究综述.软件学报,2019,30(3):799−821. http://www.jos.org.cn/1000-9825/5698.htm

英文引用格式: Zhao KK, Zhang LF, Zhang J, Li CP, Chen H. Survey on factorization machines model. Ruan Jian Xue Bao/Journal of Software, 2019,30(3):799−821 (in Chinese). http://www.jos.org.cn/1000-9825/5698.htm

Survey on Factorization Machines Model

ZHAO Kan-Kan[1,2], ZHANG Liang-Fu[1,2], ZHANG Jing[1,2], LI Cui-Ping[1,2], CHEN Hong[1,2]

[1](School of Information, Renmin University of China, Beijing 100872, China)

[2](Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education (Renmin University of China), Beijing 100872, China)

Abstract: The traditional matrix factorization method has a wide range of applications in prediction and recommendation tasks because of its high scalability and good performance. In the big data era, more and more contextual features can be obtained easily, while the traditional matrix factorization approach lacks effective use of context information. In this context, Factorization Machines (FM) is proposed and popular. To better grasp the development process of FM model and adapt FM approach to the real application, this paper reviews existing FM models and their optimization algorithms. First, it introduces the evolution process from traditional Matrix Factorization (MF) to FM model. Second, the paper summarizes the existing researches on FM method from the perspective of model accuracy and efficiency; Third, the paper presents the studies of four representative optimization algorithms, which are suitable for various FM models. Finally, the paper analyzes the challenges in the current FM model, proposes possible solutions for these problems, and discusses the future work.

Key words: factorization machine; high-order interaction; feature selection; probability model; convex optimization; distributed framework; optimization algorithm

* 基金项目: 国家自然科学基金(61772537, 61772536, 61702522 , 61532021)
Foundation item: National Natural Science Foundation of China (61772537, 61772536, 61702522, 61532021)
本文由智能数据管理与分析技术专刊特约编辑樊文飞教授、王国仁教授、王朝坤副教授推荐.
收稿时间: 2018-07-20; 修改时间: 2018-09-20; 采用时间: 2018-11-01

随着云计算、大数据、物联网等技术的迅猛发展,互联网的各类服务应用层出不穷,数据规模呈现指数级增长.根据国际数据集团 IDC 的 2012 年报告显示:到 2020 年,全球数据总量预计将达到 2011 年的 22 倍,即 35.2ZB[1].虽然大数据中蕴含着丰富的价值和巨大的潜力,但同时也使得信息过载问题越发严重.在此背景下,推荐系统诞生.作为解决信息过载问题的有效方法,推荐系统已经成为学术界和工业界的关注热点并得到广泛应用,形成了众多相关研究成果.一般地,推荐系统从用户历史数据中自动学习用户的需求和兴趣,通过多样化的推荐算法从海量数据中挖掘出用户感兴趣的项目(如信息、服务和物品等).目前,推荐系统已经在很多领域得到成功应用,包括电子商务(亚马逊、阿里巴巴、京东等)、信息检索(如 Google、百度等)、社交网络服务(如 Facebook、QQ、微博等)、位置服务(如 Foursquare、Yelp、大众点评等)、新闻推荐(如今日头条、Google news 等)、电影推荐(如 Netflix、豆瓣等)等[2].

从信息过滤的角度来看,传统的推荐系统主要分为基于内容推荐系统、协同过滤推荐系统和混合推荐系统.其中,最经典的算法非协同过滤算法莫属,而矩阵分解在重要问题上出色的预测能力,让其成为协同过滤算法中应用最为广泛的模型.在矩阵分解模型中,数据被组织为用户-物品矩阵,矩阵中每个值被编码成 0/1 或真实评分,以表示用户与物品间的交互行为.矩阵分解模型的目标在于从用户-物品矩阵中分别学习每个用户和物品的隐因子向量,最终学习得到的这种隐因子向量可用于近似重建可观察的交互值,并预测缺失或未知交互值.

在更多上下文特征可获取的今天,随着多种上下文特征引入建模,传统矩阵分解方法的扩展研究快速展开,使其能够将更多上下文因素引入模型.然而大多数基于上下文特征的分解方法都有其特定的应用场景,相比传统的矩阵分解模型而言,算法泛化能力大大降低.在此背景下,文献[3]提出了著名的因子分解机模型.

因子分解机(factorization machines,简称 FM)是一个通用的模型,其在 Movielens 数据集上的数据组织形式如图 1 所示,每个数据实例中的所有特征(用户、电影、当前用户对其他电影评分、观看时间、当前用户观看的上一部电影)被组织成一个向量 $\boldsymbol{X}_i$,并对应一个目标值 $\boldsymbol{y}_i$,特征之间互相平等.凭借该模型,Rendle 在 KDD Cup 2012 中分别取得 Track1 第 2 名和 Track2 第 3 名的成绩.与原有的分解方法相比,该模型将特征工程的一般性与分解模型的优越性相融合.它能够通过特征工程模拟绝大多数的特定分解模型,如 SVD++[4],FPMC[5],timeSVD++[6],BPTF[7],PITF[8]等.

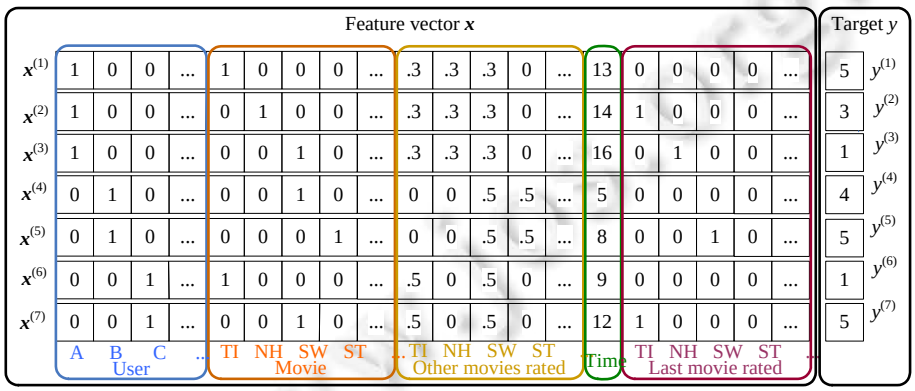


Fig.1 Data organization example of FM on Movielens dataset

图 1 基于 Movielens 的 FM 数据组织示例

本文首先对因子分解机模型进行溯源,探讨因子分解机模型和其他流行的基于上下文的分解方法的关系,指出因子分解机模型能够通过其强大的泛化能力自然地模拟这些特定场景下的分解算法;然后综述近年来涌现的针对不同问题的各种因子分解机模型的变种,从模型的准确性及性能角度出发,将其分为基于准确性的优化和基于性能的扩展;接着,从独立于问题模型的角度综述适用于因子分解机模型求解的代表性优化方法,指出每种优化方法的优势和不足;最后,本文分析了现有因子分解机模型仍然存在的问题,针对这些问题,提出了可能的解决思路,并对未来研究方向进行了展望.

本节第 1 节对因子分解机模型的提出进行溯源,指出分解方法如何从传统矩阵分解一步步发展为因子分解机模型.第 2 节综述现有因子分解机模型及其变种.第 3 节综述现阶段适用于因子分解机模型求解的代表性优化算法.第 4 节对因子分解机模型研究目前仍存在的问题进行分析,指出可能的解决思路,并对未来的研究方向进行展望.最后,在第 5 节总结本文.

1 因子分解机模型溯源

矩阵分解模型作为推荐算法的研究热点,已经在推荐系统相关领域得到大规模的应用.该算法的核心思想是:把推荐问题转化为矩阵完全分解问题,把稀疏用户评分矩阵映射到给定的用户集合和项目集合,通过矩阵运算预测缺省评分,反映用户对项目的潜在偏好,按照用户对未评分项目的预测评分值对用户进行推荐[9].矩阵分解算法能够有效降低高维数据稀疏性,并且对噪声和冗余不敏感,拥有良好的可扩展性,但是可解释性较差,计算复杂度高[10].

传统的矩阵分解算法有奇异值分解(SVD)[11]、非负矩阵分解(NMF)[12]、概率矩阵分解(PMF)[13]等.这些算法的共同特点是将高维矩阵分解成为 2 个或多个低维矩阵的乘积形式,便于在一个低维空间研究高维数据的性质.PureSVD[14]直接对用户评分矩阵做 SVD 分解,未知值采用 0 值填充,可以快速获取用户对项目预测评分.但是 SVD 允许分解出现负值,这在物理上缺乏可解释性.与 SVD 相比,NMF 可以保证分解所得矩阵的每个元素均是正值,这使得 NMF 具有直观的物理意义.不同于 SVD 和 NMF,PMF 从概率的角度预测用户评分,假设用户和商品的特征向量矩阵都符合高斯分布,基于这个假设,把用户偏好问题转换为概率组合问题,从更深层次讨论了矩阵分解的概率解释.

在大数据时代移动设备智能化的今天,上下文因素的获取变得极为容易,例如用户的人口统计信息、物品的本身属性、时间、位置、社交网络等.早在 2005 年,文献[15]研究指出,把上下文信息融入推荐系统将有利于提高推荐精确度.因此,研究者考虑基于丰富的上下文信息提升传统用户-物品矩阵分解模型的准确率.文献[16]把位置上下文引入推荐系统,引入高维 SVD(HOSVD),很好地处理了用户-位置-活动三维张量.文献[6]把时间上下文引入建模,提出一种 timeSVD++算法,提高预测用户电影评分的精准度.文献[17]将联合 PMF(UPMF)引入上下文广告推荐,把用户评分矩阵分解为用户、广告和网页特征矩阵的乘积.

上述推荐算法分别将不同的上下文因素融入分解模型中,提高了特定场景下的推荐精准度,然而这种基于特定上下文特征的分解方法其泛化能力较低,没有一个方法能将大多数上下文特征以特征工程的方式融入分解模型中去.在此背景下,文献[3]提出了著名的因子分解机模型.接下来,第 1.1 节将对因子分解机模型的数据组织形式和模型表达式进行说明.在后续章节中,给出如何使用因子分解机模型模拟其他特定模型.

1.1 标准因子分解机模型

假设一个预测问题的训练数据 $D=(\boldsymbol{X},\boldsymbol{y})$,其中,$\boldsymbol{X}\in\mathbb{R}^{n\times p}$ 表示当前数据集 D 有 n 个实例,每个实例由一个维度为 p 的稀疏向量组成,$\boldsymbol{y}\in\mathbb{R}^n$ 则表示 n 个实例对应的真实标签,$(\boldsymbol{X}_i,\boldsymbol{y}_i)$表示第 i 个实例 $\boldsymbol{X}_i$ 对应标签为 $\boldsymbol{y}_i$.

FM 能够对输入训练集 $D=(\boldsymbol{X},\boldsymbol{y})$不同特征间的交互进行分解建模,其 d 阶交互模型表示见公式(1).

$$\hat{y}(\boldsymbol{x})=w_0+\sum_{j=1}^{p}\boldsymbol{w}_j\boldsymbol{x}_j+\sum_{l=2}^{d}\sum_{j_1=1}^{p}\cdots\sum_{j_d=j_{d-1}+1}^{p}\left(\prod_{i=1}^{l}\boldsymbol{x}_{j_i}\right)\sum_{f=1}^{k_l}\prod_{i=1}^{l}\boldsymbol{V}_{j,f}^{i} \tag{1}$$

其中,模型参数 w_0,$\boldsymbol{w}_j$ 和 $\boldsymbol{V}_j^i$ 分别表示全局偏置、特征 i 对应权重以及特征 j 在与其他特征进行 i 阶交互时对应的隐因子向量;k 表示分解所得隐因子向量维度,由用户手动指定:

$$w_0\in\mathbb{R},\boldsymbol{w}\in\mathbb{R}^p,\boldsymbol{V}^i\in\mathbb{R}^{p\times k_i}.$$

然而,随着特征交互阶数的增加,模型参数规模成指数级增长,导致计算复杂度不可接受.通常在实际应用时,二阶 FM 模型已有较好的表现,故称二阶 FM 为标准 FM 模型.将公式(1)简化为标准 FM,其表达式见公式(2).

$$\hat{y}(\boldsymbol{x})=w_0+\sum_{j=1}^{p}\boldsymbol{w}_j\boldsymbol{x}_j+\sum_{j=1}^{p}\sum_{j'=j+1}^{p}\boldsymbol{x}_j\boldsymbol{x}_{j'}\sum_{f=1}^{k}\boldsymbol{V}_{j,f}\boldsymbol{V}_{j',f} \tag{2}$$

与多项式回归的不同之处在于:特征的两两交互并不由一个独立的参数 $\boldsymbol{w}_{j,j'}$来建模,而是使用两个分解的隐因子向量来估计交互参数 $\boldsymbol{w}_{j,j'}$的值,即有 $\boldsymbol{W}_{j,j'}=\langle \boldsymbol{V}_j,\boldsymbol{V}_{j'}\rangle=\sum_{f=1}^{k}\boldsymbol{V}_{j,f}\boldsymbol{V}_{j',f}$,这使得 FM 能够在高稀疏数据上有较好的表现.

1.2 矩阵分解与因子分解机模型

假设当前训练数据包含 m 个用户和 n 个物品的交互.对于传统矩阵分解,构造用户-物品交互矩阵 $R_{m\times n}$,通过矩阵运算预测缺省评分,完成对用户隐因子矩阵 $\boldsymbol{U}$ 和物品隐因子矩阵 $\boldsymbol{V}$ 的学习:

$$\boldsymbol{R}\approx \boldsymbol{U}^T\boldsymbol{V}=\hat{\boldsymbol{R}},\boldsymbol{U}\in\mathbb{R}^{m\times k},\boldsymbol{V}\in\mathbb{R}^{n\times k} \tag{3}$$

然而,这种最基本的矩阵分解思想在实际情况下并不能很好地度量用户和物品的交互.用户之间是具有差异性的,例如有的用户偏向给出物品较高的评分.同样的,这种情况也会出现在物品中.通过对公式(3)加入用户/物品和全局偏置项,得到一个更加合理的分解模型:

$$\boldsymbol{R}\approx b_0+\boldsymbol{b}_u+\boldsymbol{b}_v+\boldsymbol{U}^T\boldsymbol{V}=\hat{\boldsymbol{R}},\boldsymbol{U}\in\mathbb{R}^{m\times k},\boldsymbol{V}\in\mathbb{R}^{n\times k},b_0\in\mathbb{R},\boldsymbol{b}_u\in\mathbb{R}^m,\boldsymbol{b}_v\in\mathbb{R}^n \tag{4}$$

同样地,使用上述例子构建基于标准 FM 模型的训练集,形成一个指示矩阵 $\boldsymbol{X}\in R^{|\boldsymbol{U}|\times|\boldsymbol{V}|}$,每个数据实例 $\boldsymbol{x}$ 由 $|\boldsymbol{U}|+|\boldsymbol{V}|$维稀疏向量构成.假设当前数据实例表示用户 u 与物品 i 的交互,则该向量 $\boldsymbol{x}$ 仅第 u 和$|\boldsymbol{U}|+i$ 个位置为 1,其他均为 0:

$$(u,i)\to \boldsymbol{x}=\underbrace{(0,...,0,1,0,...,0}_{|\boldsymbol{U}|},\underbrace{0,...,0,1,0,...,0)}_{|\boldsymbol{V}|}.$$

在这种情况下,FM 与矩阵分解模型表达式相同:

$$\hat{y}(\boldsymbol{x})=w_0+\sum_{j=0}^{|\boldsymbol{U}|+|\boldsymbol{V}|}\boldsymbol{w}_j\boldsymbol{x}_j+\sum_{j=0}^{|\boldsymbol{U}|+|\boldsymbol{V}|}\sum_{j'=j+1}^{|\boldsymbol{U}|+|\boldsymbol{V}|}\boldsymbol{x}_j\boldsymbol{x}_{j'}\sum_{f=1}^{k}\boldsymbol{V}_{j,f}\boldsymbol{V}_{j',f}=w_0+\boldsymbol{w}_u+\boldsymbol{w}_v+\sum_{f=1}^{k}\boldsymbol{V}_{u,f}\boldsymbol{V}_{v,f} \tag{5}$$

1.3 SVD++与因子分解机模型

假设训练集由 m 个用户、n 个物品以及上下文特征 L 组成,其中,特征 L 不为 0 的列数为 c.标准 FM 数据组织格式可表示如下:

$$(u,i,\{l_1,...,l_c\})\to \boldsymbol{x}=\underbrace{(0,...,0,1,0,...,0}_{|\boldsymbol{U}|},\underbrace{0,...,0,1,0,...,0}_{|\boldsymbol{V}|},\underbrace{0,...,1/c,0,...,1/c,0,...)}_{|\boldsymbol{L}|}.$$

基于上述场景,标准 FM 模型可表示如公式(6):

$$\hat{y}(\boldsymbol{x})=w_0+\boldsymbol{w}_u+\boldsymbol{w}_i+\langle \boldsymbol{V}_u,\boldsymbol{V}_i\rangle+\frac{1}{c}\sum_{j=1}^{c}\langle \boldsymbol{V}_i,\boldsymbol{V}_{l_j}\rangle+\frac{1}{c}\sum_{j=1}^{c}w_{l_j}+\frac{1}{c}\sum_{j=1}^{c}\langle \boldsymbol{V}_u,\boldsymbol{V}_{l_j}\rangle+\frac{1}{c^2}\sum_{j=1}^{c}\sum_{j'=j+1}^{c}\langle V_{l_j},V_{l_{j'}}\rangle \tag{6}$$

如果$\{l_1,l_2,\ldots,l_m\}$表示用户对物品的隐式反馈特征,那么公式(7)前 5 项与完整 SVD++模型完全相同.

1.4 两两交互张量分解与因子分解机模型

假设训练数据包含 3 类特征,分别为用户、物品和物品对应的标签,与第 1.3 节中应用场景类似,其 FM 数据组织形式可表示如下:

$$(u,i,t)\to \boldsymbol{x}=\underbrace{(0,...,0,1,0,...,0}_{|\boldsymbol{U}|},\underbrace{0,...,0,1,0,...,0}_{|\boldsymbol{V}|},\underbrace{0,...,0,1,0,...,0)}_{|\boldsymbol{T}|}.$$

两两交互张量分解模型(pairwise interaction tensor factorization,简称 PITF)可表示如公式(7):

$$\hat{y}(\boldsymbol{x})=w_0+\boldsymbol{w}_u+\boldsymbol{w}_i+\boldsymbol{w}_t+\langle \boldsymbol{V}_u,\boldsymbol{V}_i\rangle+\langle \boldsymbol{V}_{u'},\boldsymbol{V}_t\rangle+\langle \boldsymbol{V}_{i'},\boldsymbol{V}_{t'}\rangle \tag{7}$$

从上式可以看出:与标准 FM 模型相同特征共享隐因子向量不同,PITF 中,每个特征在与不同类别特征交互时,其对应的隐因子参数是不同的.

1.5 支持向量机与因子分解机模型

假设训练数据中,每个数据实例包含 p 维特征,基于线性核的支持向量机可表示如公式(8):

$$\hat{y}(\boldsymbol{x}) = w_0 + \sum_{i=1}^{p} \boldsymbol{w}_i \boldsymbol{x}_i, w_0 \in \mathbb{R}, \boldsymbol{w} \in \mathbb{R}^p \tag{8}$$

显然可见,线性核支持向量机与一阶 FM 模型(不存在特征间两两交互)具有相同表达式.继而,使用非奇次多项式核替代线性核,支持向量机在特征二阶交互时可以表示为公式(9):

$$\hat{y}(\boldsymbol{x}) = w_0 + \sqrt{2}\sum_{i=1}^{p} \boldsymbol{w}_i \boldsymbol{x}_i + \sum_{i=1}^{p} \boldsymbol{W}_{i,i} \boldsymbol{x}_i^2 + \sqrt{2}\sum_{i=1}^{p}\sum_{j=i+1}^{p} \boldsymbol{W}_{i,j} \boldsymbol{x}_i \boldsymbol{x}_j, w_0 \in \mathbb{R}, \boldsymbol{w} \in \mathbb{R}^n, \boldsymbol{W} \in \mathbb{R}^{n \times n} \tag{9}$$

由公式(9)可知,多项式核使得支持向量机能够对特征间更高阶的交互进行建模.与标准 FM 相比,两个模型都能够对特征间高阶交互进行建模,不同之处在于:多项式核 SVM 所有的交互参数都是完全相互独立的,这点与多项式回归相同;而标准 FM 的交互参数通过分解学习得到,并且由于每个特征对应的隐因子参数在与任何其他特征交互时是共享的,因此进一步降低了模型复杂度.此外,通常对非线性核 SVM,需要转化为其对偶问题进行求解;而标准 FM 可以直接进行优化.

除去上述分解模型,标准 FM 还能高度模拟其他特定上下文分解模型,例如 FPMC(factorizing personalized markov chains)[5]、BPTF(Bayesian probabilistic tensor factorization)[7]、TimeSVD++[6]、最近邻模型[18]以及基于用户(物品)属性上下文模型[19,20].

2 因子分解机模型研究进展

虽然 FM 已经被广泛应用于预测、推荐等领域,且通常都能够获得较好的结果,然而在具体实践中仍存在大量挑战.从模型的准确率和效率性能两个方向出发,FM 存在的问题可总结如下.

- 在准确率方面

(1) 一般具体的 FM 模型的应用中取二阶交互,即特征的交互仅限两两相互交互建模.虽然理论上可以证明随着交互度的增加,FM 模型的时间复杂度呈线性增长趋势,但是由于 FM 数据表示的高维稀疏性,其高阶交互的线性时间复杂度在现实应用中亦难以接受;

(2) 神经网络在特征的高阶非线性交互建模及高阶特征表示方面具有较好的效果,而标准 FM 在特征的低阶交互建模方面有一定的优势,两者的融合必定会取得更好的模型性能;

(3) FM 的成功很大程度源于它对特征间的交互行为进行了建模,这种交互的加入更加符合事物的规律,有效地提升了算法的性能.然而在现实生活中,并不是所有的特征交互都能得到正向的增益,噪音特征的加入甚至会损害模型的准确性,因此,如何对特征交互加以甄别,进一步提升算法的性能和效率,仍是一个巨大挑战;

(4) 由于 FM 表达式的整体非凸性,标准 FM 基于梯度的优化,其步长及正则化超参的选择都会影响到模型训练的收敛.如何规避此类超参初值选择对模型收敛的影响、如何重构模型使得其目标函数整体呈现凸优化,有一定的困难;

(5) 在标准 FM 中,所有特征之间相互平等,类别和层次信息无法得到体现.如何将这些信息融入模型,进一步提升模型性能,是一个问题.

- 在模型训练效率和性能方面

由于 FM 模型和数据组织形式的高维稀疏性,在大数据环境下:一方面,更多的上下文特征不断涌入;另一方面,数据规模不断增加.两者共同导致因子分解机的模型复杂度增加和训练数据增长,传统的单机环境已无法满足其需求(内存无法全部存储且计算效率低下).因此,如何对模型的学习进行扩展以提高其效率变得尤为迫切.

针对上述问题,研究者从不同角度给出了解决方案.

2.1 模型准确性提升

2.1.1 高阶交互

文献[3]提出:标准 FM 模型可以适用于任意阶特征交互场景,其模型表达式如公式(1)所示.然而,一般情况下只使用 d=2,即两两特征交互.原因有二.

- 其一,同阶交互时,在建模当前特征在与其他特征的交互过程中,当前特征对应的分解隐因子向量是共享的.然而当存在多阶特征交互时,不同阶特征对应的隐因子向量是不共享的,模型的参数增长量会随着特征维度的大小线性增长.由于因子分解机模型将用户、物品本身也作为特征平等对待,因此在实际工业环境下,其特征维度会非常之大.当特征的高阶交互存在时,模型参数的增长会非常迅速,导致存储及计算资源指数级增加,限制了模型的推广.例如,假设特征维数维 m,不同阶分解所得隐向量的维度均为 k,忽略全局偏置和单个特征对应权重.当 d=3 时,与两两特征交互相比,新增模型参数;
- 其二,文献[3,21–23]仅给出了二阶特征交互下模型的 4 种优化策略——随机梯度下降、坐标下降、马尔可夫蒙特卡洛法和基于自适应正则的随机梯度下降算法.

一般地,高阶交互的加入能够使模型的性能进一步提高.针对标准 FM 模型在高阶交互中存在的问题,研究者给出了自己的方案.

文献[24]直接扩展二阶因子分解模型至三阶,并沿用二阶的交互项表达式变换技巧对三阶交互项进行了优化.相比原始的三阶交互,时间复杂度降低.最后使用随机梯度下降学习得到最终模型.然而,该三阶模型依然未解决能够向更高阶通用扩展的问题,且模型复杂度依然随阶按倍数增长.相似的,文献[25]提出一种三阶因子分解模型,与标准因子分解机模型不同,作者使用 ANOVA 核建模特征间的三阶交互,使用坐标下降法对三阶 ANOVA 核因子分解机模型优化.

与上述只达到三阶交互的模型不同,文献[26]提出一种任意高阶共享因子分解模型.作者同样使用了 ANOVA 核对其进行重构.ANOVA 核通常被用来建模特征间的交互,与因子分解机模型的交互项相似.ANOVA 核的特性使得高阶的 ANOVA 核可以递归转化为对应的低一阶交互核.因此,作者使用非齐次 ANOVA 核的特性,递归地从高到低建模不同阶特征交互过程,在该核中,同一特征的隐因子向量在不同阶的特征交互中是可共享的.此外,不同阶的交互对模型的影响可使用权重参数建模.就如何高效地优化高阶共享因子分解机模型,作者提出了一种动态规划算法,用于计算不同阶交互对模型的贡献值和计算梯度.由于核的递归特性,该动态规划算法能够避免许多重复性计算.此外,作者使用基于随机梯度和坐标下降这两种优化算法对模型进行了优化.

文献[27]提出一种基于多任务多视图(multi-task multi-view)的高阶因子分解机模型.多任务多视图学习假设一个大的任务能够分解为多个子任务,每个子任务则由多个视图构成.其最终目标则是根据所有子任务中的训练集信息,利用多个视图的交互,学习得到一个非线性的分类或回归模型.在多任务多视图中,整个任务可以看做一个张量,每个视图和子任务为张量的一个维度.然而,由于训练数据的稀疏性,无需物理构建一个张量训练集合.作者首先摒弃了直接使用单个参数来建模某个特定交互,转而采用使用 CP 分解(canonical polyadic decomposition),大大降低了参数规模.进而分析若直接使用多视图交互建模,最终只能得到一个基于满(最高)阶交互的非线性模型的限制,通过给每个视图中添加一个常量为 1 的列,使得不同视图间的交互可以由原来的满阶交互变为从一阶到满阶的不同阶交互并存.与标准高阶 FM 的不同阶交互由不同参数负责相比,这种基于多任务多视图的高阶 FM 的不同阶交互的参数是共享的.这种机制大大减少了参数的规模,与上述基于 ANOVA 核的高阶 FM 有异曲同工之妙.最后,由于多任务维度的存在,经过基于坐标下降的 CP 分解,该维度生成与其他视图的交互隐因子矩阵对应的特定任务权重矩阵.即,用户针对不同的任务(话题)具有不同的喜好程度(权重).

文献[28]在文献[27]的基础上提出一种新的基于多视图的结构因子分解机模型.在文献[27]中,一个视图中仅包含一项实体,由一个向量表示,且视图间实体不存在重叠.而在文献[28]中,作者改变思路,文中每个视图可包含若干项实体,因此,视图以一个向量的形式存在,且不同实体间允许存在实体重叠.由于不同视图间存在实体重叠,因此该重叠的实体在进行 CP 分解时,所生成的隐因子矩阵可共享.此外,作者采用了与文献[27]中相同的技巧,使得同一视图内可进行不同阶交互隐因子共享.

文献[29]提出多视图分解机(multi-view machine,简称 MVM),与文献[27]类似,作者使用基于 CP 分解的多视图来完成高阶 FM 的建模,并且采用与之相同的技巧完成不同阶的交互隐因子的共享.此外,作者认为:并非所有的高阶交互都是有其物理意义的,且高阶交互通常难以解释,因此,作者提出通过对原始的全部视图进行分割,形成多个视图集合,使用 MVM 对多个视图集合分别学习的方法来降低交互阶数.在 CP 分解过程中,使用随机梯

度下降法来优化目标函数.最后,作者基于 Spark 完成了 MVM 的分布式实现.

2.1.2 神经网络与因子分解机

随着神经网络和深度学习的兴起,研究者考虑如何将因子分解机与神经网络相结合,提高最终模型的性能.文献[30,31]提出使用神经网络完成用户/物品特征隐因子表示和学习,然后将其串联成一个单独特征向量,最后基于该特征向量使用标准 FM 完成模型的训练.与文献[30,31]先神经网络学习特征表示后因子分解机训练模型的方式不同,文献[32-36]均利用因子分解机模型作为神经网络的输入进行特征的高阶非线性交互建模.其中,文献[32]提出一种基于因子分解机和神经网络的高阶非线性特征交互模型,简称 FNN.具体地,在使用神经网络进行点击率预测时,FNN 并未直接使用稀疏的 0-1 向量特征,而是考虑先利用标准 FM 基于随机梯度下降法预训练得到模型所包含的全局偏置、每个特征对应的一阶权重和二阶交互因子向量,然后将上述全部模型参数作为隐含层的输入,使用神经网络方法生成最终模型.该方法能够对标准 FM 模型所生成的参数进行高阶非线性交互,提升神经网络方法的预测能力.与文献[32]类似,文献[33]提出一种神经网络与非线性因子分解机模型的混合模型 NLFM.NLFM 基于标准 FM 模型的全部参数(除去全局偏置参数),使用神经网络完成后续的训练.然而,与 FNN 将模型参数作为隐含层的输入不同,NLFM 首先将模型参数输入一个嵌入层,生成并输出标准 FM 模型中每个特征对应的一阶交互和与其他特征之间的二阶交互,然后将嵌入层的输出作为隐含层的输入,进而利用神经网络完成训练.为了提高最终模型预测能力,文献[33]进一步提出使用堆栈去噪自动编码器对原始用户/物品及其上下文特征进行降维,降低数据稀疏度.在生成的高级特征基础上,使用 NLFM 进行模型训练.与文献[32,33]不同,文献[34–36]仅对特征的交互因子使用神经网络方法进行高阶非线性交互建模.其中,文献[34]基于 Wide&Deep 网络,结合因子分解机和深度学习,提出一种基于深度网络的因子分解机模型 DeepFM 进行广告点击率预测.具体地,DeepFM 对共享二阶交互因子矩阵分别采用标准 FM 和深度神经网络建模二阶特征交互和高阶特征交互,然后基于因子分解机模型输出和深度神经网络输出结果的和完成最终的预测.文献[35]研究了稀疏输入数据情况下的推荐问题,基于因子分解机提出了一种神经因子分解机模型 NFM.NFM 保持因子分解机模型的全局偏置和特征的一阶权重形式不变,将特征交互因子作为嵌入层,在嵌入层上增加一个双线性交互池化操作,即因子分解机模型特征二阶交互部分,然后将池化的输出作为隐含层的输入实现特征间的高阶非线性交互.值得注意的是:双线性交互池化操作的加入,使得 NFM 能在更少隐含层的情况下获得更好的预测能力,而且参数更少,训练更加容易.文献[36]通过扩展 NFM 模型提出一种注意力因子分解机模型 AFM,通过将注意力机制引入双线性交互池化操作中,进一步提升 NFM 的表示能力和可解释性.

2.1.3 交互特征选择

在标准 FM 模型中,特征间的交互涵盖整个特征集.然而在真实场景下,有些特征是不相关的,它们的交互是没有任何物理意义且不可解释的,甚至由于噪音特征的引入损害了预测结果.换言之,并非所有的特征交互都会对最终的预测值起到正向增益的效果.基于此,研究者提出对特征交互进行选择以提升模型性能和效率.

文献[37]提出了一种基于梯度 Boosting 的贪婪的特征交互选择方法,并与标准 FM 模型整合为一个统一的框架——GBFM.具体地,在每次迭代中,使用一个贪婪的梯度 Boosting 方法选择一组交互特征,基于上一步预测与真实值的残差来优化所选交互特征的隐因子向量.在特征选择过程中,作者采用一种多层贪婪启发式算法,在每层总是选择使得目标下降最快的一类特征.实验证明,GBFM 很好地提升了模型的性能.然而,这种基于梯度 Boosting 的特征交互选择是有问题的:迭代的交互特征选择机制意味着不同次迭代可能选择重复的特征,但重复特征在不同次迭代所对应的隐因子向量是不同的,从而一定程度上导致模型参数规模的增大,使得训练效率降低.

与迭代的贪婪特征选择机制不同,文献[38]仅仅考虑选择有用的用户和物品之间的交互.具体地,作者首先将标准 FM 模型中的交互隐因子矩阵按照特征划分为用户隐因子矩阵和物品隐因子矩阵,通过矩阵变换重构标准 FM 模型;然后在损失函数中分别对用户和物品隐因子矩阵加组稀疏正则项,达到移除无关用户和物品特征的效果;最后使用块坐标下降学习得到用户和物品隐因子矩阵.这种特征选择方法最大的问题在于:它限制了 FM 模型基于特征工程的强大泛化性,导致上下文特征无法参与建模.

文献[39]认为基于梯度 Boosting 和稀疏正则的方法在大规模高阶特征交互选择时是不可行的,提出一种可进行任意阶交互特征选择的贝叶斯回归因子分解机模型.具体地,使用超图来表示特征间的任意阶交互关系,其中:特征为顶点,特征的任意阶交互子集为超边,交互特征选择由随机超图上的先验分布指导完成.与上述两种方法相比,该方法具有更强的泛化性.

2.1.4 概率模型

在标准 FM 模型中,作者首先给出了包括随机梯度下降[3]和交替最小二乘[21](坐标下降)在内的两种优化策略.这两种策略在优化时,其超参的初始值都需要用户指定.然而,由于FM模型的非凸性,这种随机指定超参的方式很可能会使得模型陷入一个局部最优解;而 FM 模型在真实环境下因为模型参数高维且数据规模巨大,训练一次是非常耗时的.因此,得到一个局部最优解再重新训练,这无疑是不可接受的.在不考虑对模型本身不作出修正时,研究者考虑从概率模型角度出发,得到一个无需手动指定超参的概率 FM 模型.

文献[22]率先提出一个概率 FM 模型,该模型在标准 FM 概率图模型基础上添加了一层对超参的超先验:对除了全局偏置外所有模型参数的正太分布所对应的平均值添加高斯超先验,对所有模型参数的正太分布所对应的准确率添加伽马超先验,最后利用吉布斯抽样和共轭先验分布求得模型最优参数.文献[40]提出一个基于非线性上下文协同过滤方法:高斯过程 FM(GPFM),GPFM 能够无缝利用用户对物品的显式和隐式行为.具体地,作者使用高斯过程,对每一个用户生成高斯先验,针对用户的显式和隐式行为生成高斯似然,在此基础上计算得到边缘似然,最终使用随机梯度下降对负的对数边缘似然最小化函数进行优化,得到最优模型参数.文献[41]认为文献[22]中所提数据服从正态分布,对于某些使用整数评分的场景是不合适的,且在选择隐因子向量维度时需要进行交叉验证,这无疑是非常耗时的.基于这两个问题,作者对整数评分场景使用伽马分布,并使用一个伽马过程自动搜寻一个理想的隐因子向量维度,该过程并不需要交叉验证.文献[42]首先提出在点击率预测这种高度稀疏数据上利用拉普拉斯分布代替传统的高斯分布.与高斯分布相比,拉普拉斯分布更加适合于高维稀疏数据并辨别度量相关特征的关系.由于拉普拉斯分布是非平滑的,基于贝叶斯推论的 FM 模型难以优化,而拉普拉斯分布能够使用混合高斯分布来模拟,最终使用马尔可夫蒙特卡洛方法对基于拉普拉斯分布的稀疏FM模型进行优化.

2.1.5 凸优化及在线学习

为了从根本上解决标准FM模型的非凸问题并保持保证两阶特征交互矩阵的低秩特性,研究者给出了多种方案[43−46].文献[43]提出一种针对二阶特征交互因子矩阵改进的凸因子分解机模型 CFM.具体地,CFM 保持标准 FM 模型的一阶特征权重参数不变,将原有的二阶特征交互因子矩阵 $V\in\mathbb{R}^{p\times k}$ 重构为 $Z\in\mathbb{R}^{p\times p}$.虽然表面上 Z 的参数规模大于 V,但是 CFM 并不物理存储 Z,而是通过矩阵特征分解的方式将对称矩阵 Z 分解为多个秩为 1 的矩阵的和.相比标准 FM 模型需要指定二阶特征交互隐因子矩阵 V 的维度 k,CFM 在最终的优化目标函数中使用核范数保证分解的低秩特性,无需用户手动设置.文献[44]将标准 FM 模型中的全局偏置参数糅合至特征的一阶权重,形成一个增广向量,并对二阶特征交互隐因子矩阵采用与文献[43]类似的表示.不同的是:文献[43]对所有的特征二阶交互进行建模并采用块坐标下降的方法对目标函数进行优化;而文献[44]仅考虑不同特征间的二阶交互(不考虑顺序),采用 Hazan 算法对模型参数进行更新.总结文献[43,44]可知:两者仅对二阶特征交互隐因子矩阵进行变换,最终得到一个一阶特征权重参数和二阶特征交互隐因子矩阵分别优化的凸因子分解机模型.基于上述文献,文献[45]进一步提出一种可在线学习的凸因子分解机模型 OCCFM.OCCFM 将因子分解机中的全局偏置、一阶特征权重和二阶特征交互隐因子矩阵等所有模型参数全部糅合,形成一个增广参数矩阵,文献[46]采用与之相同的凸因子分解机模型表示.在模型优化时,对所有模型参数统一进行更新.由于因子分解机模型独特的数据结构表示,随着越来越多上下文特征的加入,其参数规模不断增加,导致模型面临所有分解方法都存在的巨大挑战——新数据到来后模型参数的更新.巨大的参数规模和训练数据集合,使得模型的全量更新所消耗的计算资源和时间是不可接受的,而且线下的更新方式势必导致用户体验变差.因此,如何对因子分解模型进行在线更新成为研究热点.文献[45,46]在凸因子分解机模型基础上分别使用在线条件梯度和 FTRL(follow the regularized leader)两个经典的在线凸优化方法完成模型的学习.

2.1.6 层次信息引入

在标准 FM 模型中,并不存在类别信息,所有特征之间是平等的.同一特征与任意其他特征的同阶交互,其对应的隐因子向量是共享的,这种共享机制极大地减少了参数规模.然而这种设置并不符合实际情况,一些研究者认为:当特征在与不同类别特征交互时,应该使用不同的隐因子向量.因为隐因子向量刻画了两类特征交互的内涵,而不同的类别特征交互其所具备的物理含义很可能是全不同的.基于这种假设,文献[8]提出一种基于因子分解的个性化标签推荐方法,作者使用用户、标签和物品这 3 类特征,分别对(用户-物品)、(用户-标签)和(标签-物品)这 3 类交互进行因子分解,得到了较好的推荐效果.在文献[8]的基础上,文献[47]提出了一种通用的基于类别上下文的因子分解机模型 FFM,针对同一特征与不同类别间特征的同阶交互使用不同的特征隐因子向量,并给出一种并行思路,应用至点击率预测中.相较于标准 FM 模型,虽然 FFM 有效地提高了模型准确率,但是模型的参数规模也成倍增加.假设训练数据中存在 f 类不同特征,总的特征维度为 p,每个特征的二阶交互隐因子向量维度为 k,那么标准 FM 模型参数规模为 $p(k+1)+1$,FFM 的参数规模为 $p(fk+1)+1$,其中,$1<<p$ 并且 $f<<p$.因此 $(p(fk+1)+1)/(p(k+1)+1)\approx f$,即 FFM 的参数规模是标准 FM 的 f 倍.而由于因子分解机模型所对应的特殊特征结构导致参数规模本身已经极大,f 倍于标准 FM 参数规模的 FFM 的优化将需要更多的时间和计算资源,这在真实工业环境中是不可接受的.因此,为达成在尽可能小的参数规模下对特征类别影响进行建模的目标,文献[48]提出一种基于类别权重的因子分解机模型 FwFM.相比 FFM 为每个特征学习 f 个不同的隐因子向量,FwFM 采取与标准 FM 模型相同的策略——每个特征对应一个隐因子向量,同时采用给不同类特征间的交互乘上一个类别交互因子权重的方式来建模不同类特征交互的强度.由于 $f<<p$,因此 FwFM 的参数规模与标准 FM 基本相同.

与上述泛化的基于不同类别特征交互的隐因子向量来刻画类别层次信息不同,文献[49]聚焦移动广告点击率预测场景,利用场景中不同类别隐含的层次信息和重要度信息构建树状训练集划分,实现数据实例与不同类别的交互.

2.1.7 其他应用场景

针对标准 FM 本身存在的问题和挑战,上述研究分别从一个或多个角度进行了优化研究.如何将因子分解机模型应用至具体不同场景下不同问题下,也有研究者给出了多样的解决方案.

文献[50]提出同时使用两个因子分解机模型来同时分别建模用户兴趣(是否转发某条推特)和发现推特中的话题,两个模型之间可以通过不同的共享机制联系,作者共提出 3 种不同的共享机制,包括特征共享、隐式空间共享和隐式空间正则化(距离).文献[51]将因子分解机模型应用至跨领域协同过滤中,这种方法通过连接其他领域和目标领域的特征,使得模型能够充分利用其他领域的信息来补充当前领域上下文,以提高目标领域的推荐性能.受文献[52]针对矩阵基于层次上下文相似度分割的启发,文献[53]认为,基于似上下文环境的数据实例建模能够提高模型精度,提出利用数据集中隐含的层次信息,多次随机构造多颗决策树.在每个层次节点的训练集划分时,使用当前节点上所有的训练集基于随机梯度下降或坐标下降法进行训练,然后利用 K-means 聚类算法对当前节点对应类别生成的隐因子向量进行聚类划分,得到多个聚类中心,按照聚类中心,将当前训练数据分割分发至下一层.重复上述过程.在测试阶段,使用多个随机决策树生成值的平均作为最终的预测结果.文献[54–56]将用户之间的信任度、相似度和相互关注等信息融入因子分解机模型特征向量中,提高推荐准确率.一般地,标准 FM 模型用于推荐、预测等领域,然而针对排序场景,模型并未给出相应的优化,因此,文献[57]提出结合 Learning-to-Rank 的因子分解机模型,针对 AUC 度量方法进行直接优化.受文献[58]的 BPR 排序理论影响,文献[59]提出基于 AUC 度量方法的排序因子分解机模型,在进行负例选择时,作者利用隐式反馈和内容信息提出一种自适应的采用算法,其主要思路是:相同类别中,用户对有过历史行为的物品比未有过历史行为的物品有更高的评分.然而,AUC 度量并不适合于 Top-N 推荐任务.基于 AUC 的排序使得在度量错误的排序时,无论其处在 Top-N 推荐列表的顶部或底部具有一样的影响力.这种设定是不符合常理的.文献[60]认为,在 Top-N 推荐列表的顶部具有比在底部更高的正确率是非常重要的,而能达到这种排序效果的度量方法有 NDCG 和 MRR.因此,提出了 LambdaFM,直接基于 NDCG 和 MRR 等排序度量方法进行优化,对处在不同排序位置的物品对赋予不同的权重,并提出 3 种采样策略.文献[61]提出一种基于 Boosting 的排序因子分解机模型,其主要思想是:在每次迭代

时,生成一个新的排序因子分解机模型用于优化上一步的残差,其中,每步的排序模型基于 Learning-to-Rank 优化.此外,作者维护了一个权值向量用于对每一对物品对进行动态权值分配,主要是对当前表现较差的物品对赋予更高的权值,以达到纠正的效果.

2.2 模型性能提升

FM 现在已经被广泛地应用于预测、推荐等领域,特征工程和分解模型的结合,使得它通常能获得较好的性能[3,21,50,51,54].然而模型独有的特征组织形式导致它在大规模数据集下,传统的单机环境已无法满足其需求.因此,模型的扩展应运而生.目前,针对因子分解机模型的扩展研究主要集中在两个方面:数据/参数形式重组和分布式扩展.

在数据重组方面,文献[62]基于标准模型提出一种不改变单机环境,而是针对模型底层数据的组织特征重新建立新的模型,不但有效地降低了底层数据的存储,而且提高了模型的计算效率.文献[28]也使用了同样的技巧,以提升其提出的基于多视图的结构因子分解机模型.与改变底层数据组织、提升模型训练效率不同,文献[63]认为:在真实工业环境中,其所能获得的特征维度之高直接造成了昂贵的存储和计算代价,这大大阻碍了快速推荐在计算资源有限设备(如移动设备)上的应用.因此,作者提出离散化因子分解机(DFM),将原模型的实数型特征交互因子矩阵改为布尔型.相较之下,布尔型因子矩阵具有低存储、可快速计算等优点,然而在原优化方案下,取值范围的缩小必然导致模型性能的下降.为此,作者提出一种特殊的离散参数优化方案来学习布尔因子矩阵.实验结果表明:DFM 在保证良好模型准确率的同时,能 16 倍加速于 FM.

在分布式扩展方面,文献[64]基于 Map-Reduce 对标准 FM 模型进行了实现.文献[65]首次提出使用基于异步随机梯度下降的参数服务器框架来实现模型的分布式扩展.为了进一步提高模型计算效率和性能,作者对模型本身进行了两方面的优化:首先,作者认为,针对所有特征的隐因子向量使用相同维度是不必要的,浪费了存储和计算资源,针对数据集中出现频率低的特征,可以适当地降低对应的隐因子向量维度,就此,作者提出一种基于频繁度的启发式方法度量隐因子向量维度;第二,在正则化项部分,作者加入了稀疏正则项和基于特征出现频率自适应的正则项进一步提高了模型效率和性能.与文献[65]类似,文献[66]在基于 Map-Reduce 的参数服务器框架下对标准 FM 模型进行了优化.所不同的是,优化方法使用了一种分布式坐标下降算法.为了减少服务器和 worker 端的通信代价并减少参数更新冲突,作者提出一种启发式数据划分策略.与主从分布式框架不同,文献[67,68]提出一种环形的分布式框架.该框架摒弃了服务器端,使得集群中仅存在两类节点:调度节点和 worker 节点.其中:调度节点负责响应模型调度请求,存储全局变量;worker 节点则负责模型的更新和存储.基本思想是:在数据分发时,调用 Spark 接口将整个训练集按照既定数据分发策略平均分发至每一个 worker 节点,然后进行模型训练.训练时,令每个 worker 节点各自独立保存一部分模型参数,使用各个节点的子数据集对保存的模型参数进行更新,最终得到与客户端节点相同数目的相互独立的部分模型.然而由于 FM 数据组织形式的高维稀疏性,将数据分发至每个 worker,数据稀疏性愈发严重.如果仅使用每个 worker 节点上的子数据集对自身的模型进行更新,最终学习得到的模型准确率无法保证.因此,作者设计了一套环形的模型调度机制.这种调度机制允许每个 worker 节点不但可以更新它本身存储的模型,还可以更新不属于它的其他 worker 节点的模型.即:节点在更新完模型后,将模型推送至它的原始保存节点,然后通过调度函数,节点选择另一个新的从未更新过的模型,从所属节点拉取该模型,继续使用子数据集对新模型更新.在下一个模型选择时,优先考虑同一机器上其他节点(每个核可表示一个节点).另外,为避免频繁的模型调度,每个模型在一个节点上的学习基于多次迭代完成.模型训练结束后,在模型预测阶段,利用已经学习得到的多个模型对每个测试实例进行独立预测,对多个模型预测的结果采用多数投票的方法进行合并.相比主从分布式框架,该环形分布式系统能够有效地减少节点间的通信频度和规模.与上述对标准 FM 模型的优化不同,文献[69]使用参数服务器框架实现了基于领域的因子分解机模型的分布式扩展,分别与 All-Reduce、类 Map-Reduce 等分布式实现进行了对比.文献[29]对提出的多视图分解机模型进行了基于 Spark 平台的分布式实现.

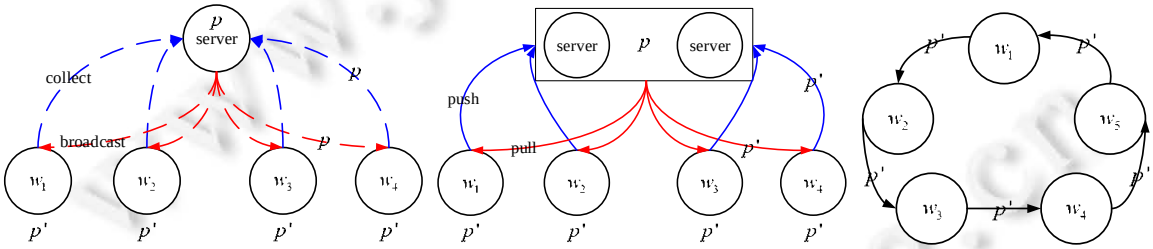
总结 FM 模型的 3 种不同分布式底层框架,给出不同框架的详细比较如表 1 和图 2 所示.按照架构分类,基于 Map-Reduce/Spark 和参数服务器的分布式属于主从式架构,环形分布式则属于端到端的架构.主从式架构中

必须包含一个或多个服务器节点,最新的模型参数全部存储于服务器节点,参数的更新计算则由 worker 节点完成.然而,端到端的结构中并不存在服务器节点,因此模型参数的存储和更新均由各个 worker 节点完成.相比 Map-Reduce/Spark,参数服务器的服务器组机制能够极大地分担通信压力.在通信方面,Map-Reduce/Spark 和参数服务器框架在每次迭代时,服务器与 worker 节点间都必须进行参数传递,不同之处在于:Map-Reduce/Spark 结构需要广播整个参数空间至每个 worker 节点,参数服务器架构则只需传送与 worker 节点上数据对应的参数即可.而环形分布式框架由于模型参数和数据在同一节点,只有 worker 间进行模型交换时才会进行必要的参数传递.综上所述,Map-Reduce/Spark 通信代价最大,参数服务器框架次之,环形分布式最小.在参数更新同/异步方面,若由于数据划分不均衡,同步更新中有些节点计算快,有些节点计算慢,由此产生的等待时延会大大降低训练效率.因此一般情况下,异步更新效率要高于同步更新.在模型学习完成时,主从式框架最终只产生一个模型.然而在环形分布式架构下,每个节点都会产生一个独立的部分模型,在模型应用阶段,使用多个部分模型独立预测每个实例,根据场景不同,利用投票表决或求平均的方式得到最终预测结果.

Table 1 Comparison of different distributed frameworks

表 1 不同分布式框架对比

框架名称	架构分类	服务器节点	参数传递	同步异步	生成模型个数
Map-Reduce/Spark	主从式	1 个	全部参数,广播	同步	1 个
参数服务器框架	主从式	多个,用户指定	仅拉取/推送部分参数	可异步	1 个
环形框架	端到端式	无	仅拉取/推送部分参数	同步	多个



(a) Spark/Map-Reduce (b) 参数服务器框架 (c) 环形分布式框架

Fig.2 Architecture of different distributed frameworks used in FM model

图 2 FM 模型不同分布式系统框架图

3 因子分解机模型优化算法

本节将综述适用于求解因子分解机模型的优化算法.为便于描述,我们首先将约束优化因子分解机模型的损失函数统一形式.假设训练集$(\boldsymbol{x},y)\in D$,我们使用 $\hat{y}(\boldsymbol{x}\mid\Theta)$ 表示因子分解机模型表达式,则模型参数的优化问题通常通过定义损失函数来使得训练集 D 的损失和最小来完成.即有:

$$OPT(D):=\arg\min_{\Theta}\sum_{(\boldsymbol{x},y)\in D}l(\hat{y}(\boldsymbol{x}\mid\Theta),y) \tag{10}$$

为避免模型训练时产生过拟合,向上述优化目标中加入正则项函数 $R(\Theta)$:

$$OPT(D):=\arg\min_{\Theta}\left(\sum_{(\boldsymbol{x},y)\in D}l(\hat{y}(\boldsymbol{x}\mid\Theta),y)+R(\Theta)\right) \tag{11}$$

针对具体的任务,可选择不同的优化算法.因子分解机模型在分类和回归场景中都有较好的表现,这里以回归和二分类问题为例.

- 在回归问题中,其损失函数通常定义如下:

$$l^{LS}(\hat{y},y):=(\hat{y}-y)^2 \tag{12}$$

- 在二分类问题中,其损失函数可如下定义:

$$l^C(\hat{y},y):=-\ln\sigma(\hat{y}y) \tag{13}$$

其中,$\sigma(x)$为 logistic 函数.

目前,针对 FM 模型的优化有 4 种优化学习算法,分别是随机梯度下降、交替最小二乘法、基于自适应正则项值的随机梯度下降和 MCMC.下面我们将对这 4 种算法分别进行剖析.

3.1 随机梯度下降

随机梯度下降算法是众多机器学习算法中最常用的优化算法,在不同损失函数下均能有较好的表现以及较低的计算复杂度、存储复杂度等特性.它的具体思路是每次从训练集中随机选择一个样本来更新模型参数.

再次回顾标准 FM 模型如公式(2)可知,模型共包含 3 类参数:w_0,$\boldsymbol{w}$,$\boldsymbol{V}$.虽然 FM 模型在整体上表现出非凸性,但是在优化每一个参数$\Theta=\{w_0,\boldsymbol{w}_1,\boldsymbol{w}_i,\ldots,\boldsymbol{V}_{1,1},\ldots,\boldsymbol{V}_{1,f},\ldots,\boldsymbol{V}_{i,f}\}$时,目标函数可看做凸函数.

在随机梯度下降算法中,每一步每个参数的更新都可表示为

$$\theta\leftarrow\theta-\eta\left(\frac{\partial l(\hat{y}(\boldsymbol{x}\mid\Theta),y)}{\partial\theta}+\frac{\partial R(\Theta)}{\partial\theta}\right) \tag{14}$$

其中,$\eta\in\mathbb{R}^+$表示每次迭代的学习速率.

针对不同的任务场景下不同损失函数梯度的计算是不同的.在回归问题中有:

$$\frac{\partial l^{LS}(\hat{y}(\boldsymbol{x}\mid\Theta),y)}{\partial\theta}=\frac{\partial}{\partial\theta}(\hat{y}(\boldsymbol{x}\mid\Theta)-y)^2=2(\hat{y}(\boldsymbol{x}\mid\Theta)-y)\frac{\partial}{\partial\theta}\hat{y}(\boldsymbol{x}\mid\Theta) \tag{15}$$

对于二分类问题:

$$\frac{\partial}{\partial\theta}l^C(\hat{y}(\boldsymbol{x}\mid\Theta),y)=\frac{\partial}{\partial\theta}-\ln\sigma(\hat{y}(\boldsymbol{x}\mid\Theta)y)=(\sigma(\hat{y}(\boldsymbol{x}\mid\Theta)y)-1)y\frac{\partial}{\partial\theta}\hat{y}(\boldsymbol{x}\mid\Theta) \tag{16}$$

具体到每一个参数,其相对模型表达式的偏导可表示为

$$\frac{\partial}{\partial\theta}\hat{y}(\boldsymbol{x}\mid\Theta)=\begin{cases}1, & \text{if } \theta \text{ is } w_0\\ x_l, & \text{if } \theta \text{ is } \boldsymbol{w}_l\\ x_l\sum_{j\neq l}\boldsymbol{V}_{j,f}\boldsymbol{x}_j, & \text{if } \theta \text{ is } \boldsymbol{V}_{l,f}\end{cases} \tag{17}$$

算法 1 给出了 FM 模型在随机梯度下降算法下的详细优化过程.

算法 1. 随机梯度下降法.

输入:训练集 D,迭代次数 T,隐因子维度 k,正则化项参数λ,学习速率η,初始化参数σ;

输出:FM 模型参数$\Theta=w_0,\boldsymbol{w},\boldsymbol{V}$.

初始化模型参数 $w_0\leftarrow0,\boldsymbol{w}\leftarrow(0,\ldots,0),\boldsymbol{V}\sim N(0,\sigma)$

for iter in Range $\{1,\ldots,T\}$ **do**

  随机选择一个数据实例$(\boldsymbol{x},y)$

  按照公式(14)更新全局偏置 w_0

  for $i\in\{1,\ldots,p\}\wedge\boldsymbol{x}_i\neq0$ **do**

    按照公式(14)更新单变量权重 w_i

    for $f\in\{1,\ldots,k\}$ **do**

      按照公式(14)更新隐因子参数 $v_{i,f}$

    end for

  end for

end for

通过分析随机梯度下降可知,学习速率η的选取对目标函数的收敛影响非常大.其中:如果η过大,则会导致算法不收敛;若η选择过小,则会导致算法收敛过慢,严重影响模型训练效率.因此,如何设置适当大小的η,是随机梯度下降优化算法成功的关键所在.

3.2 交替最小二乘法

基于训练数据实例的迭代,随机梯度下降算法沿着损失函数梯度下降的方向小步地行进来完成损失函数的最小化.与随机梯度下降优化不同,交替最小二乘法采用最小化每一个模型参数的方法来完成优化.具体地,在固定其他参数不变的情况下,每次更新一个参数,使当前参数沿着梯度方向寻找其最优解,每个参数的最优解可通过使其偏导为 0 得到.

为了方便取得最优解,这里我们确定公式(11)中的正则化项采用 L2 范数.然而在标准 FM 模型中,假设训练集 D 共 n 个数据实例,每个数据实例包含 p 维特征,那么在隐因子向量维度为 k 时,共有 $1+p+kp$ 个模型参数.若为每一个模型参数都给定一个独立的正则化项系数,这无疑大大增加存储和计算复杂度.通过对同一类模型参数采用相同正则项系数的方法,可大大简化这一过程.因此,公式(11)可进一步表达如公式(18):

$$(w_0^*,\boldsymbol{w}^*,\boldsymbol{V}^*)=\arg\min_{w_0,\boldsymbol{w},\boldsymbol{V}}\left(\sum_{(\boldsymbol{x},y)\in D} l(\hat{y}(\boldsymbol{x}\mid w_0,\boldsymbol{w},\boldsymbol{V}),y)+\lambda_{w_0}w_0^2+\lambda_{\boldsymbol{w}}\sum_{i=1}^{p}\boldsymbol{w}_i^2+\lambda_{\boldsymbol{V}}\sum_{i=1}^{p}\sum_{f=1}^{k}\boldsymbol{V}_{i,f}^2\right) \tag{18}$$

假设当前任务为回归问题,通过对损失函数在模型参数 θ 下求偏导并令偏导为 0,得到 θ 最优解.在标准 FM 模型中,其具有多线性特点,即:对任意模型参数 $\theta\in\Theta$,标准 FM 模型可以写成两个函数 g_θ 和 h_θ 线性组合的方式,独立于参数 θ:

$$\hat{y}(\boldsymbol{x})=g_\theta(\boldsymbol{x})+\theta h_\theta(\boldsymbol{x}) \tag{19}$$

其中,h_θ 即当前参数相对模型表达式的偏导数:

$$h_\theta(\boldsymbol{x})=\frac{\partial}{\partial\theta}\hat{y}(\boldsymbol{x}) \tag{20}$$

基于上述表述,模型参数 θ^* 的最优解可表示如公式(21):

$$\theta^*=-\frac{\sum_{(\boldsymbol{x},y)\in D}(g_\theta(\boldsymbol{x})-y)h_\theta(\boldsymbol{x})}{\sum_{(\boldsymbol{x},y)\in D}h_\theta^2(\boldsymbol{x})+\lambda_\theta} \tag{21}$$

在固定其他参数的前提下,使用公式(21)对每个模型参数 θ 进行更新,多次迭代直至收敛,即可完成对所有模型参数的优化.通过分析参数更新公式(21)可知,其中最耗时计算部分主要集中在以下两个式子:

$$\sum_{(\boldsymbol{x},y)\in D}h_\theta^2(\boldsymbol{x}),\ \sum_{(\boldsymbol{x},y)\in D}(y-\hat{y}(\boldsymbol{x}))h_\theta(\boldsymbol{x}) \tag{22}$$

从公式(22)可以看出,在每次更新一个参数时,都需对 $y-\hat{y}(\boldsymbol{x})$ 重新计算.而在更新参数 $\boldsymbol{V}$ 时,$h_\theta(\boldsymbol{x})$ 也需要进行复杂的重新计算完成.为了提高训练效率,可通过预计算的方式避免上述部分的重复计算.

令 e 和 q 分别表示如下:

$$e(\boldsymbol{x},y)=\hat{y}(\boldsymbol{x})-y,q(\boldsymbol{x},f)=\sum_{l=1}^{p}\boldsymbol{V}_{l,f}\boldsymbol{X}_{i,l} \tag{23}$$

则针对参数 $\boldsymbol{V}_{l,f}$ 的计算可简化为

$$h_{\boldsymbol{V}_{l,f}}(\boldsymbol{X}_i)=\boldsymbol{X}_{i,l}(q_{i,f}-\boldsymbol{V}_{l,f}\boldsymbol{X}_{i,l}) \tag{24}$$

针对每次参数更新后 $\hat{y}(\boldsymbol{x})-y$ 和 $q_{i,f}$ 的计算可简化为

$$e(\boldsymbol{x},y\mid\theta^*)\leftarrow e(\boldsymbol{x},y\mid\theta)+(\theta^*-\theta)h_\theta(\boldsymbol{x}),q(\boldsymbol{x},f\mid\theta^*)\leftarrow q(\boldsymbol{x},f\mid\theta)+(\boldsymbol{V}_{l,f}^*-\boldsymbol{V}_{l,f})\boldsymbol{x}_l \tag{25}$$

算法 2 给出了 FM 模型在交替最小二乘算法下的详细优化过程.

算法 2. 交替最小二乘法.

输入:训练集 D,迭代次数 T,隐因子维度 k,正则化项参数 λ,学习速率 η,初始化参数 σ;

输出:FM 模型参数 $\Theta=w_0,\boldsymbol{w},\boldsymbol{V}$.

初始化模型参数 $w_0\leftarrow 0,\boldsymbol{w}\leftarrow(0,\ldots,0),\boldsymbol{V}\sim N(0,\sigma)$

for each $(\boldsymbol{x},y)\in D$ **do**

按照公式(23)预计算 $e(\boldsymbol{x},y|\Theta)$

for $f\in\{1,\ldots,k\}$ **do**
按照公式(23)预计算 $q(\boldsymbol{x},f|\Theta)$
end for
end for
for iter in Range $\{1,\ldots,T\}$ **do**
按照公式(21)计算全局偏置 w_0^*
按照公式(25)更新预计算 $e(\boldsymbol{x},y|\Theta^*)\leftarrow e(x,y|\Theta)+(w_0^*-w_0)$
更新 $w_0\leftarrow w_o^*$
for $i\in\{1,\ldots,p\}$ **do**
按照公式(21)计算单变量权重 w_i^*
按照公式(25)更新预计算 $e(\boldsymbol{x},y|\Theta^*)\leftarrow e(\boldsymbol{x},y|\Theta)+(w_i^*-w_i)\boldsymbol{x}_i$
更新 $w_i\leftarrow w_i^*$
end for
for $f\in\{1,\ldots,k\}$ **do**
for $i\in\{1,\ldots,p\}$ **do**
按照公式(21)计算隐因子参数 $v_{i,f}^*$
按照公式(25)更新预计算 $e(\boldsymbol{x},y|\Theta^*)\leftarrow e(\boldsymbol{x},y|\Theta)+(v_{i,f}^*-v_{i,f})(\boldsymbol{x}_i q(\boldsymbol{x},f|\Theta)-\boldsymbol{x}_i^2 v_{i,f})$
按照公式(25)更新预计算 $q(\boldsymbol{x},f|\Theta^*)\leftarrow q(\boldsymbol{x},f|\Theta)+(v_{i,f}^*-v_{i,f})\boldsymbol{x}_i$
更新 $v_{i,f}\leftarrow v_{i,f}^*$
end for
end for
end for

通过分析交替最小二乘可知:相比随机梯度下降,一个显著的优势在于交替最小二乘中不存在学习速率η的选取,但是正则项超参λ仍然会影响模型的优化,好的正则项系数的搜索是十分耗时的.

3.3 基于自适应正则的随机梯度下降

在随机梯度下降和交替最小二乘中,一般正则项超参λ的设置由用户手动完成,若选择不好,则导致模型的欠拟合或过拟合.严格来说,一个好的正则项超参λ需要经过昂贵的搜索得到.为克服这一缺点,基于自适应正则项的随机梯度下降算法被提出[23].该方法能够避免最佳正则化项系数的网格搜索过程,具体地,通过对模型参数Θ与正则化项系数λ使用交替最小二乘固定其中一个、优化另一个的方法来达到正则化项系数λ的自适应.

通常,理想正则化项系数λ^*的选取使用验证集的方法.即,将训练集 D 分割成不相交的两部分:$D=D_T\cup D_V$.首先,使用给定的正则化常数λ,在D_T上完成模型参数Θ的优化;然后,在验证集D_V评估学习到的模型参数Θ的质量.由于训练集D_T和验证集D_V不相交,因此学习得到的模型参数Θ在验证集上的评估质量能够充分说明其在更大数据集上的有效性.

根据上述描述,为得到优化的正则化项系数λ^*,可构建损失函数如公式(26):

$$\lambda^*:=\arg\min_{\lambda}\sum_{(\boldsymbol{x},y)\in D_V} l(\hat{y}(\boldsymbol{x}|OPT(D_T,\lambda)),y) \tag{26}$$

在这个嵌套损失函数中,最外面的优化目标是通过在验证集上最小化损失来决定最佳正则化项系数λ^*.但这个损失并不直接取决于正则化项系数,而是依赖训练集上模型参数Θ的学习情况,即有公式(27):

$$\Theta^*|\lambda=OPT(D_T,\lambda) \tag{27}$$

对于这种嵌套的损失函数,一个直接优化上述目标的方法就是使用交替最小二乘法,其中,参数分为两类:

一类是模型参数Θ,一类是正则化项系数λ.然而这种方法是有问题的——在固定模型参数时,正则化项系数λ并不会显式出现在公式(26)中.

为了解决上述问题,一个比较巧妙的方法是,Θ^{t+1} 的值依赖于λ,因此可以利用模型参数更新公式对上述式子进行改进:

$$\left.\begin{aligned}\hat{y}(\boldsymbol{x}\mid\Theta^{t+1}) &= w_0^{t+1}+\sum_{j=1}^{p}\sum_{j'=j+1}^{p}\langle \boldsymbol{V}_j^{t+1},\boldsymbol{V}_{j'}^{t+1}\rangle \boldsymbol{x}_j\boldsymbol{x}_{j'}\\ &= w_0^t-\eta\left(\frac{\partial l(\hat{y}(\boldsymbol{x}\mid\Theta^t),y)}{\partial w_0^t}+2\lambda_{w_0}w_0^t\right)+\sum_{j=1}^{p}\boldsymbol{x}_j\left(\boldsymbol{w}_j^t-\eta\left(\frac{\partial l(\hat{y}(\boldsymbol{x}\mid\Theta^t),y)}{\partial \boldsymbol{w}_j^t}+2\lambda_{\boldsymbol{w}}\boldsymbol{w}_j^t\right)\right)+\\ &\quad\sum_{j=1}^{p}\sum_{j'=j+1}^{p}\sum_{f=1}^{k}\left[\boldsymbol{x}_j\left(\boldsymbol{V}_{j,f}^t-\eta\left(\frac{\partial l(\hat{y}(\boldsymbol{x}\mid\Theta^t))}{\partial \boldsymbol{V}_{j,f}^t}+2\lambda_{\boldsymbol{V}}\boldsymbol{V}_{j,f}^t\right)\right)\boldsymbol{x}_{j'}\left(\boldsymbol{V}_{j',f}^t-\eta\left(\frac{\partial l(\hat{y}(\boldsymbol{x}\mid\Theta^t))}{\partial \boldsymbol{V}_{j',f}^t}+2\lambda_{\boldsymbol{V}}\boldsymbol{V}_{j',f}^t\right)\right)\right]\end{aligned}\right\} \tag{29}$$

基于上述改进,优化目标函数(26)转化为公式(29):

$$\lambda^*\mid\Theta^t:=\arg\min_{\lambda}\sum_{(\boldsymbol{x},y)\in D_V} l(\hat{y}(\boldsymbol{x}\mid\Theta^{t+1}),y) \tag{29}$$

正则化项系数λ在验证集 D_V 上的更新见公式(30):

$$\lambda_\theta^{t+1}=\lambda_\theta^{t+1}-\eta\frac{\partial}{\partial\lambda_\theta^t}l(\hat{y}(\boldsymbol{x}\mid\Theta^{t+1}),y) \tag{30}$$

在清楚了模型参数更新及正则化项系数更新梯度后,算法3给出了FM模型在基.于自适应正则化随机梯度下降的详细优化过程.

算法 3. 基于自适应正则化的随机梯度下降算法.

输入:训练集 D,迭代次数 T,隐因子维度 k,学习速率η,初始化参数σ;

输出:FM 模型参数Θ=w_0,$\boldsymbol{w}$,$\boldsymbol{V}$.

初始化模型参数 $w_0\leftarrow 0,\boldsymbol{w}\leftarrow(0,\ldots,0),\boldsymbol{V}\sim N(0,\sigma)$

初始化正则化项系数 $\lambda_{w_0}=0,\lambda_{\boldsymbol{w}}\leftarrow(0,...,0),\lambda_f\leftarrow(0,...,0)$

for iter in Range {1,…,T} **do**

  for ($\boldsymbol{x}$,y)$\in D_T$ **do**

    按照公式(14)更新 w_0

    for $i\in\{1,\ldots,p\}\wedge\boldsymbol{x}_i\neq 0$ **do**

      按照公式(14)更新 w_i

      for $f\in\{1,\ldots,k\}$ **do**

        按照公式(14)更新 $v_{i,f}$

      end for

    end for

  end for

  抽样选择一个数据实例($\boldsymbol{x}'$,y')$\in D_V$

  $\lambda_{w_0}\leftarrow\max\left(0,\lambda_{w_0}-\eta\partial/\partial\lambda_{w_0}\, l(\hat{y}(\boldsymbol{x}'\mid\Theta^*),y')\right)$

  $\lambda_{\boldsymbol{w}}\leftarrow\max(0,\lambda_{\boldsymbol{w}}-\eta\partial/\partial\lambda_{\boldsymbol{w}}\, l(\hat{y}(\boldsymbol{x}'\mid\Theta^*),y'))$

  for $f\in\{1,\ldots,k\}$ **do**

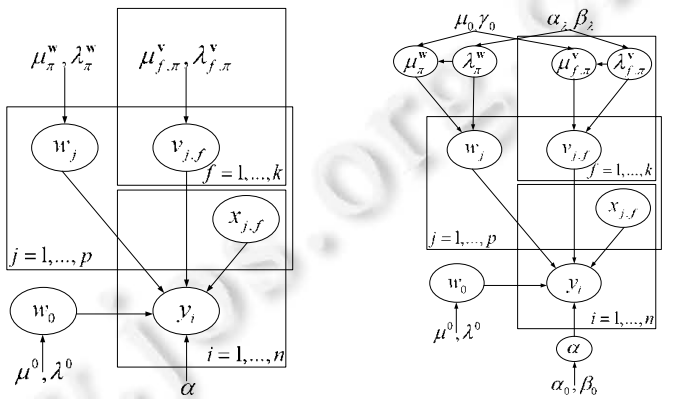
    $\lambda_f\leftarrow\max\left(0,\lambda_f-\eta\partial/\partial\lambda_f\, l(\hat{y}(\boldsymbol{x}'\mid\Theta^*),y')\right)$

  end for

end for

3.4 马尔可夫蒙特卡洛法

随机梯度下降和交替最小二乘算法使用点估计 $\hat{y}$ 的方法学习出模型参数 Θ;而马尔可夫蒙特卡洛法(MCMC)则是基于贝叶斯推论,通过抽样的方法来生成 $\hat{y}$ 的分布.与随机梯度下降和交替最小二乘法相比,MCMC 允许将超参放入模型中一起进行优化,避免了费时的最优超参搜索过程.而这种将超参加入模型优化的方法需要我们将标准 FM 的概率图模型(图 3(a)所示)扩展为基于超参超先验的 FM 概率图模型,如图 3(b)所示.



(a) 标准 FM 概率图模型 (b) 基于超先验的 FM 概率图模型

Fig.3 Comparison of the probabilistic interpretation of standard factorization machines (left) to Bayesian factorization machines (right) extended by hyperpriors

图 3 标准 FM 概率模型与基于超先验的贝叶斯 FM 概率模型对比

具体地,基于吉布斯采样的 MCMC 方法下,FM 模型中每个参数的条件后验分布可表示如下:

$$\theta \mid X, y, \Theta / \{\theta\}, \Theta_H \sim N(\tilde{\mu}_\theta, \tilde{\sigma}_\theta^2) \tag{31}$$

其中, $\tilde{\mu}_\theta$ 和 $\tilde{\sigma}_\theta^2$ 如公式(32)所示,Θ_H 表示图 3 所示的超参:

$$\tilde{\sigma}_\theta^2 := \left(\alpha \sum_{i=1}^{n} h_\theta^2(\boldsymbol{x}_i) + \lambda_\theta\right)^{-1}, \tilde{\mu}_\theta := \tilde{\sigma}_\theta^2 \left(\alpha\theta \sum_{i=1}^{n} h_\theta^2(\boldsymbol{x}_i) + \alpha \sum_{i=1}^{n} h_\theta(\boldsymbol{x}_i) e_i + \mu_\theta \lambda_\theta\right) \tag{32}$$

观察 MCMC 下每个参数的条件后验分布,可以发现,其与交替最小二乘法优化公式十分的相似.即:当 α=1 且 μ=0 时,有 $\theta^* = \tilde{\mu}_\theta$.两种方法的不同之处在于:MCMC 从参数的后验分布进行采样,而交替最小二乘法则使用其期望值.

在 FM 模型的 MCMC 推论中,假设先验参数 μ_θ 服从标准正太分布,而 λ_θ 和 α 服从 Gamma 分布,可得:

$$\mu_\pi^w \sim N(\mu_0, \gamma_0 \lambda_\pi^w), \lambda_\pi^w \sim \Gamma(\alpha_\lambda, \beta_\lambda), \mu_{f,\pi}^V \sim N(\mu_0, \gamma_0 \lambda_{f,\pi}^V), \lambda_{f,\pi}^V \sim \Gamma(\alpha_\lambda, \beta_\lambda), \alpha \sim \Gamma(\alpha_0, \beta_0) \tag{33}$$

其中,Θ_0:={$\mu_0, \gamma_0, \alpha_\lambda, \beta_\lambda, \alpha_0, \beta_0$}用来描述超先验分布.

基于上述表示,超参的值可以通过从其相应的条件后验分布中采样自动获得:

$$\left.\begin{aligned}
&\alpha \mid \boldsymbol{y}, \boldsymbol{X}, \Theta_0, \Theta \sim \Gamma\left(\frac{\alpha_0 + n}{2}, \frac{1}{2}\left[\sum_{i=1}^{n} (\boldsymbol{y}_i - \hat{y}(\boldsymbol{x}_i \mid \Theta))^2 + \beta_0\right]\right) \\
&\lambda_\pi^{\cdot} \mid \Theta_0, \Theta_H \setminus \{\lambda_\pi^{\cdot}\}, \Theta \sim \Gamma\left(\frac{a_\lambda + p^\pi + 1}{2}, \frac{1}{2}\left[\sum_{j=1}^{p} \delta(\pi(j) = \pi)(\theta_j - \mu_\theta)^2 + \gamma_0(\mu_\pi - \mu_0)^2 + \beta_\lambda\right]\right) \\
&\mu_\pi^{\cdot} \mid \Theta_0, \Theta_H \setminus \{\mu_\pi^{\cdot}\}, \Theta \sim N\left((p^\pi + \gamma_0)^{-1}\left[\sum_{j=1}^{p} \delta(\pi(j) = \pi)\theta_j + \gamma_0 \mu_0\right], \frac{1}{(p^\pi + \gamma_0)\lambda_\pi^{\cdot}}\right)
\end{aligned}\right\} \tag{34}$$

其中,

$$p^{\pi} := \sum_{j=1}^{p} \delta(\pi(j) = \pi) \tag{35}$$

算法 4 给出了 FM 模型在 MCMC 下解决回归 i 的详细优化过程.若想将其扩展到二分类任务的解决,则需要将正态分布的 $\hat{y}$ 映射到伯努利分布.这样,MCMC 算法就能预测一个实例属于正类或者负类的概率.

算法 4. 马尔可夫蒙特卡洛算法.

输入:训练集 D_{tr},测试集 D_{te},迭代次数 T,隐因子维度 k,初始化参数 σ;

输出:测试集 D_{te} 上预测结果 $\hat{y}$.

初始化模型参数 $w_0 \leftarrow 0, \boldsymbol{w} \leftarrow (0,\ldots,0), \boldsymbol{V} \sim N(0,\sigma)$

for iter in Range $\{1,\ldots,T\}$ **do**

  在训练集 D_{tr} 上预测计算 $\hat{y}$

  计算残差 $e(\boldsymbol{x}, y \mid \Theta) \leftarrow y - \hat{y}$

  按照公式(34)抽样得到 α

  for $(\mu_{\pi}^{\cdot}, \lambda_{\pi}^{\cdot}) \in \Theta_H$ **do**

    按照公式(34)抽样得到 $\lambda_{\pi}^{\cdot}$ 和 $\mu_{\pi}^{\cdot}$

  end for

  按照公式(31)抽样得到 w_0

  更新残差 $e(\boldsymbol{x},y|\Theta)$

  for $i \in \{1,\ldots,p\}$ **do**

    按照公式(31)抽样得到 w_i

    更新残差 $e(\boldsymbol{x},y|\Theta)$

  end for

  for $f \in \{1,\ldots,k\}$ **do**

    for $i \in \{1,\ldots,p\}$ **do**

      按照公式(31)抽样得到 $v_{i,f}$

      更新残差 $e(\boldsymbol{x},y|\Theta)$

    end for

  end for

  测试集上计算预测值 $\hat{y}_{test}^{*}$

  $\hat{y}_{test} \leftarrow \hat{y}_{test} + \hat{y}_{test}^{*}$

end for

求测试集预测平均值 $\hat{y}_{test} \leftarrow \hat{y}_{test} / T$

通过对 MCMC 方法分析可知 MCMC 的正则项超参 Θ_H 是抽样选取的,为此引进了新的超先验参数 Θ_0.但是,超先验参数的数目要小于正则化项超参的数量.另外,更重要的一点在于:相比随机梯度下降和交替最小二乘算法,MCMC 对于超先验参数 Θ_0 的选择不敏感.即使该参数选择不好,FM 模型也能得到较好的结果.

3.5 4种优化算法比较

本节综述了 FM 模型常用的 4 种优化学习方法,分别是随机梯度下降法、交替最小二乘法、基于自适应正则的随机梯度下降法和马尔可夫蒙特卡洛法.下面我们从时间复杂度、空间复杂度、适用任务场景、超参数类型这 4 个方面对上述方法总结说明,见表 2.其中,$N_z(\boldsymbol{X})$表示训练集中所有不为 0 特征的总个数.由于每次迭代时参数的更新计算通过遍历一次训练集即可完成,因此 4 种优化方法在时间复杂度上是一致的.然而在存储复杂度上,各个优化算法所需的存储空间是不同的.两类随机梯度下降算法下的 FM 优化只需要存储模型中每一个参数的迭代更新值,故其存储复杂度为常数级.与上述梯度更新相比,交替最小二乘法和马尔可夫蒙特卡洛方法除去 $1+p(k+1)$大小内存用于参数的存储外,还需要 $O(nk)$的内存去存储预计算结果.在适用任务场景中,使用不

同损失函数或分布的 4 种优化方法都能胜任常见的回归或分类问题.在超参类型上,随机梯度下降需用户自定义的超参最多,包括学习速率、分布初始化参数和正则化项系数.基于自适应正则化的随机梯度下降和马尔可夫蒙特卡洛法将正则化项系数作为参数糅合到模型中进行共同学习,避免了最优正则化项系数的搜索过程,模型学习过程更加健壮.而除了随机梯度下降,其他 3 种优化方法都不存在学习速率的指定,也使得模型的优化更加稳定.

Table 2 Comparison of different optimization methods

表 2 不同优化方法对比

算法名称	时间复杂度	空间复杂度	适用回归	适用分类	超参类型
随机梯度下降	$O(kN_z(\boldsymbol{X}))$	$O(1)$	是	是	学习速率/正则项系数/分布参数
基于自适应正则的随机梯度下降	$O(kN_z(\boldsymbol{X}))$	$O(1)$	是	是	学习速率/分布参数
交替最小二乘	$O(kN_z(\boldsymbol{X}))$	$O(nk)$	是	是	正则项系数/分布参数
马尔可夫蒙特卡洛法	$O(kN_z(\boldsymbol{X}))$	$O(nk)$	是	是	超先验参数/分布参数

4 存在问题及未来研究方向

4.1 存在问题

在第 2 节给出了因子分解机模型在准确性提升和性能加速两个方面取得的研究进展,然而仍存在以下几点不足.

(1) 模型准确性方面

现有对 FM 模型准确性提升的工作基本从低阶到高阶交互、神经网络与 FM 的结合、好的交互特征选择、概率模型推导、凸优化模型、在线学习、层次信息引入以及具体场景分析等 8 个方面展开.在高阶交互方面,基于更高阶特征交互的 FM 模型固然能够进一步挖掘特征之间的相互关联,从而提升模型准确率,然而 FM 的高阶交互会导致模型可解释性进一步降低,并可能进一步放大噪音特征对模型准确性的影响.在神经网络与 FM 的结合方向,现有结合方案可分为两类:① 将标准 FM 模型作为神经网络的输入,以便于后续利用神经网络完成特征的更高阶非线性交互;② 同时使用标准 FM 模型和神经网络分别完成特征的低阶和高阶交互建模,未来可考虑与其他更多类型神经网络的结合以适应不同的应用场景.在交互特征选择领域,已有的工作多是基于标准 FM 模型展开的,存在可扩展性较低以及模型参数规模并未因为特征选择而减小等问题.针对 FM 模型的在线学习,现有工作基于凸 FM 模型展开,分别利用流行的凸优化学习算法在线条件梯度和 FTRL 完成模型的更新.然而这种单机在线学习算法已无法适用于大规模参数和数据集的场景.最后,在层次信息引入上,研究者从树状层次特征引入和不同类别特征交互使用不同隐因子向量/权重两个角度展开来提高模型准确性,但是依然存在拘泥于特定场景、模型规模指数级增加及无法适用于高阶特征交互建模等问题.

(2) 模型效率方面

现有针对 FM 模型的分布式扩展工作已有多种方案,然而挑战依然存在.按照分布式框架的不同,可分为 Map-Redce/Spark、参数服务器框架和环形分布式框架.基于 Map-Reduce/Spark 平台的扩展既有高阶 FM 模型,也有标准 FM 算法,最大的短板在于全局模型需存放于一个服务器节点且模型传输时通信开销巨大,导致其在模型规模庞大的真实生产环境中无法应用.基于参数服务器框架的扩展目前仅限于标准 FM 模型,相比 Map-Reduce/Spark 架构,通信开销更少,模型存储也不再限制于一个服务器节点.然而由于模型的高维性,多次迭代下通信开销依然巨大.在此背景下,基于环形分布式的二阶 FM 模型被提出,优势在于通信开销和频次进一步减小,但稍显劣势之处在于单个节点上存储的模型规模有所增加.从上述描述可以看出:标准 FM 模型的分布式扩展已趋于成熟,然而基于 FM 的高阶交互、特征选择、凸模型以及在线学习等多种模型变种的分布式扩展依然存在瓶颈,有待进一步完善.

4.2 未来研究方向

针对上述因子分解机模型研究中依然存在的问题,对其未来研究方向进行探讨.

(1) 模型准确性研究

在高阶模型可解释性方面:一方面,可借助特征选择算法降低高阶建模时所需的特征数目,从而降低交互阶数,并剔除噪音、冗余和不相关特征;另一方面,对所有特征的相关性进行分析,使用多个高阶 FM 模对每组特征完成交互建模,最终根据任务场景,利用加权平均或投票表决的方式进行预测.

在神经网络与 FM 模型的结合方面,现有研究神经网络部分都是基于传统神经网络或向传统神经网络中加入注意力机制,后续研究可从神经网络部分入手,考虑 FM 模型与其他高级神经网络类型如长短期记忆网络 LSTM 的结合,可用于在流式数据中挖掘用户的长期和短期兴趣,提升推荐结果质量.

在交互特征选择方面,研究目标大致可分为两个方向:① 考虑扩展基于标准 FM 的交互特征选择至高阶 FM 模型,与现有高阶交互研究相结合,不仅可以达到降低模型参数规模的目标,而且提高了模型准确性和可解释性;② 考虑采用现有流行方法高效的选择好的交互特征,如可利用神经网络基于原始数据完成特征的表示学习,利用强化学习思想选择.

在 FM 模型的在线学习方面,有两个基本问题需要考虑:其一,假设新进数据特征维度不变,即不存在新特征的加入,这种情况下,如何利用新进数据增量更新已有模型;其二,假设新进数据中存在新用户、新物品和新的上下文特征,那么如何基于已有模型较小调整甚至不改变的情况下学习得到新的模型.现有工作都是以特征维度不变为前提展开的,后续的研究方向可从 3 个方向入手:① 利用其他已有的在线学习方法完成 FM 模型及其变种的增量更新;② 扩展单机环境下 FM 模型的在线学习方法至分布式环境下,以适应真实生产环境下的大规模参数和数据集;③ 考虑新的特征加入条件下,利用矩阵运算达到基于已有模型的较小调整甚至不改变情况下更大规模新的模型的学习.

在层次信息引入方面,现有工作主要存在特定场景特殊分析、模型规模指数级增长以及不适用于高阶特征交互建模等短板.因此,未来可考虑提出一种通用的层次特征建模方法,使得 FM 模型能够利用这种层次的特征有效提高模型的表达能力,如结合神经网络和注意力机制,使得特征间的高阶交互建模得以实现,并利用注意力机制学习得到不同类别特征交互的强度.

(2) 模型效率研究

现有针对 FM 模型效率提升主要分为基于数据/参数重组和基于分布式两种.其中,由于分布式优化在大数据环境下的高效性使其成为主流研究方向,得到较多关注.然而多数工作都是基于标准 FM 模型展开的,在变种 FM 模型,如高阶交互、特征选择等方面仅仅做了简单的分布式实现.在 FM 模型的增量更新方面,目前仍无相关研究.因此,如何针对上述两个方向进行分布式扩展是非常值得关注的.

5 总 结

作为一种通用的分解模型,因子分解机模型能够取得较好的预测和推荐结果,近年来在机器学习领域得到了广泛的应用和关注,取得了诸多研究成果.本文首先从分解模型的演化角度说明了传统的矩阵分解模型如何一步步进化到基于特定上下文的分解方法,进而得到本文综述重点——通用因子分解机模型,并通过因子分解机模型与其他流行的特定分解模型的相互关联性说明了因子分解机模型强大的泛化性;然后,从因子分解机模型的准确性和性能两方面出发,说明标准因子分解机模型存在的不足,并给出近年来研究者针对这两个方面所存在问题的优化方案;接着,从独立于问题模型的角度综述了 FM 模型常用的 4 种优化方法,并指出各个优化算法的优势和不足;最后指出了现有因子分解模型研究中存在的不足和未来可能的研究方向.

References:

[1] Gantz J, Reinsel D. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. IDC iView: IDC Analyze the Future, 2012,2007(2012):1−16.

[2] Huang LW, Jiang BT, Lv SY, Liu YB, Li DY. Survey on deep learning based recommender systems. Chinese Journal of Computers, 2018,41(7):1619−1647 (in Chinese with English abstract).

[3] Rendle S. Factorization machines. In: Proc. of the 2010 IEEE 10th Int'l Conf. on Data Mining (ICDM). IEEE, 2010. 995−1000.

[4] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2008. 426−434.

[5] Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next-basket recommendation. In: Proc. of the Int'l Conf. on World Wide Web. ACM Press, 2010. 811−820.

[6] Koren Y. Collaborative filtering with temporal dynamics. Communications of the ACM, 2010,53(4):89−97.

[7] Xiong L, Chen X, Huang TK, *et al*. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In: Proc. of the 2010 SIAM Int'l Conf. on Data Mining. 2010. 211−222.

[8] Rendle S, Schmidt-Thieme L. Pairwise interaction tensor factorization for personalized tag recommendation. In: Proc. of the ACM Int'l Conf. on Web Search and Data Mining. ACM Press, 2010. 81−90.

[9] Meng XW, Ji WY, Zhang YJ. A survey of recommendation systems in big data. Journal of Beijing University of Posts and Telecommunications, 2015,38(2):1−15 (in Chinese with English abstract).

[10] Dror G, Koenigstein N, Koren Y, *et al*. The Yahoo! music dataset and KDD-Cup'11. In: Proc. of the 17th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining. San Diego: ACM Press, 2012. 8−18.

[11] Golub G, Kahan K. Calculating the singular values and pseudo-inverse of a matrix. Journal of the Society for Industrial and Applied Mathematics, 1965,2(2):205−224.

[12] Lee DD, Seung H. Algorithms for non-negative matrix factorization. In: Proc. of the 13th Advances in Neural Information Processing Systems (NIPS 2000). Denver: MIT Press, 2000. 556−562.

[13] Salakhutdinov R, Mnih A. Probabilistic matrix factorization. In: Proc. of the 20th Advances in Neural Information Processing Systems. 2007,20(3):432−451.

[14] Ding WF, Zheng XL, Chen DR. Active sampling based on PureSVD model for collaborative filtering. Journal of Beijing University of Posts and Telecommunications, 2013,36(4):23−26 (in Chinese with English abstract).

[15] Adomavicius G, Sankaranarayanan R, Sen S, *et al*. Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. on Information Systems (TOIS), 2005,23(1):103−145.

[16] Symeonidis P, Papadimitriou A, Manolopoulos Y, *et al*. Geo-social recommendations based on incremental tensor reduction and local path traversal. In: Proc. of the 3rd ACM SIGSPATIAL Int'l Workshop on Location-Based Social Networks. 2011. 89−96.

[17] Tu DD, Shu CC, Yu HY. Using unified probabilistic matrix factorization for contextual advertisement recommendation. Ruan Jian Xue Bao/Journal of Software, 2013,24(3):454−464 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4238.htm [doi: 10.3724/SP.J.1001.2013.04238]

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

[19] Gantner Z, Drumond L, Freudenthaler C, *et al*. Learning attribute-to-feature mappings for cold-start recommendations. In: Proc. of the 2010 IEEE 10th Int'l Conf. on Data Mining (ICDM). IEEE, 2010. 176−185.

[20] Agarwal D, Chen BC. Regression-based latent factor models. In: Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2009. 19−28.

[21] Rendle S, Gantner Z, Freudenthaler C, *et al*. Fast context-aware recommendations with factorization machines. In: Proc. of the 34th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM Press, 2011. 635−644.

[22] Freudenthaler C, Schmidt-Thieme L, Rendle S. Bayesian factorization machines. 2011.

[23] Rendle S. Learning recommender systems with adaptive regularization. In: Proc. of the 5th ACM Int'l Conf. on Web Search and Data Mining. ACM Press, 2012. 133−142.

[24] Knoll J. Recommending with higher-order factorization machines. In: Proc. of the Research and Development in Intelligent Systems XXXIII. Springer Int'l Publishing, 2016. 103−116.

[25] Blondel M, Ishihata M, Fujino A, *et al*. Polynomial networks and factorization machines: New insights and efficient training algorithms. IEEE Trans. on Wireless Communications, 2016,15(1):131−145.

[26] Blondel M, Fujino A, Ueda N, *et al*. Higher-order factorization machines. Advances in Neural Information Processing Systems. 2016. 3351−3359.

[27] Lu CT, He L, Shao W, *et al*. Multilinear factorization machines for multi-task multi-view learning. In: Proc. of the 10th ACM Int'l Conf. on Web Search and Data Mining. ACM Press, 2017. 701−709.

[28] Lu CT, He L, Ding H, *et al*. Learning from multi-view multi-way data via structural factorization machines. In: Proc. of the WWW. 2018.

[29] Cao B, Zhou H, Li G, *et al*. Multi-view machines. In: Proc. of the Ninth ACM Int'l Conf. on Web Search and Data Mining. ACM, 2016. 427−436.

[30] Zheng L, Noroozi V, Yu PS. Joint deep modeling of users and items using reviews for recommendation. In: Proc. of the Int'l Conf. on Web Search and Data Mining. ACM Press, 2017. 425−434.

[31] Cai Y, Dong S, Hu J. Jointly modeling user and item reviews by CNN for multi-domain recommendation. In: Proc. of the China Conf. on Information Retrieval. Cham: Springer-Verlag, 2018. 237−248.

[32] Zhang W, Du T, Wang J. Deep learning over multi-field categorical data. In: Proc. of the European Conf. on Information Retrieval. Cham: Springer-Verlag, 2016. 45−57.

[33] Liu Y, Guo W, Zang D, *et al*. A hybrid neural network model with non-linear factorization machines for collaborative recommendation. In: Proc. of the China Conf. on Information Retrieval. Cham: Springer-Verlag, 2018. 213−224.

[34] Guo H, Tang R, Ye Y, *et al*. DeepFM: A factorization-machine based neural network for CTR prediction. In: Proc. of the 26th Int'l Joint Conf. on Artificial Intelligence. AAAI Press, 2017. 1725−1731.

[35] He X, Chua TS. Neural factorization machines for sparse predictive analytics. In: Proc. of the 40th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM Press, 2017. 355−364.

[36] Xiao J, Ye H, He X, *et al*. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In: Proc. of the 26th Int'l Joint Conf. on Artificial Intelligence. AAAI Press, 2017. 3119−3125.

[37] Cheng C, Xia F, Zhang T, *et al*. Gradient boosting factorization machines. In: Proc. of the 8th ACM Conf. on Recommender Systems. ACM Press, 2014. 265−272.

[38] Xu J, Lin K, Tan PN, *et al*. Synergies that matter: Efficient interaction selection via sparse factorization machine. In: Proc. of the 2016 SIAM Int'l Conf. on Data Mining. Society for Industrial and Applied Mathematics, 2016. 108−116.

[39] Yurochkin M, Nguyen XL. Multi-way interacting regression via factorization machines. In: Proc. of the Advances in Neural Information Processing Systems. 2017. 2598−2606.

[40] Nguyen TV, Karatzoglou A, Baltrunas L. Gaussian process factorization machines for context-aware recommendations. In: Proc. of the 37th Int'l ACM SIGIR Conf. on Research & Development in Information Retrieval. ACM Press, 2014. 63−72.

[41] Saha A, Acharya A, Ravindran B, *et al*. Nonparametric poisson factorization machine. In: Proc. of the IEEE Int'l Conf. on Data Mining (ICDM 2015). IEEE, 2015. 967−972.

[42] Pan Z, Chen E, Liu Q, *et al*. Sparse factorization machines for click-through rate prediction. In: Proc. of the 2016 IEEE 16th Int'l Conf. on Data Mining (ICDM). IEEE, 2016. 400−409.

[43] Blondel M, Fujino A, Ueda N. Convex factorization machines. In: Proc. of the Joint European Conf. on Machine Learning and Knowledge Discovery in Databases. Cham: Springer-Verlag, 2015. 19−35.

[44] Chang Y, *et al*. Convex factorization machine for toxicogenomics prediction. In: Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining. ACM Press, 2017. 1215−1224.

[45] Lin X, Zhang W, Zhang M, *et al*. Online compact convexified factorization machine. In: Proc. of the Int'l Conf. on World Wide Web. ACM Press, 2018. 1633−1642.

[46] Luo L, Zhang W, Zhang Z, *et al*. Sketched follow-the-regularized-leader for online factorization machine. In: Proc. of the 24th Int'l Conf. on Knowledge Discovery & Data Mining. ACM Press, 2018. 1900−1909.

[47] Juan Y, Zhuang Y, Chin WS, *et al*. Field-aware factorization machines for CTR prediction. In: Proc. of the 10th ACM Conf. on Recommender Systems. ACM Press, 2016. 43−50.

[48] Pan J, Xu J, Ruiz AL, *et al*. Field-weighted factorization machines for click-through rate prediction in display advertising. In: Proc. of the 2018 World Wide Web Conf. on World Wide Web. 2018. 1349−1357.

[49] Oentaryo R, Lim E, Low J, Lo D, Finegold M. Predicting response in mobile advertising with hierarchical importance-aware factorization machine. In: Proc. of the 7th ACM Int'l Conf. on Web Search and Data Mining. New York: ACM Press, 2014. 123−132.

[50] Hong L, Doumith A, Davison B. Co-factorization machines: Modeling user interests and predicting individual decisions in Twitter. In: Proc. of the 6th ACM Int'l Conf. on Web Search and Data Mining. ACM Press, 2013. 557−566.

[51] Loni B, Shi Y, Larson M, *et al*. Cross-domain collaborative filtering with factorization machines. In: Proc. of the European Conf. on Information Retrieval. Cham: Springer-Verlag, 2014. 656−661.

[52] Zhong E, Fan W, Yang Q. Contextual collaborative filtering via hierarchical matrix factorization. In: Proc. of the 2012 SIAM Int'l Conf. on Data Mining. Society for Industrial and Applied Mathematics, 2012. 744−755.

[53] Wang S, Du C, Zhao K, *et al*. Random partition factorization machines for context-aware recommendations. In: Proc. of the Int'l Conf. on Web-Age Information Management. Cham: Springer-Verlag, 2016. 219−230.

[54] Rendle S. Social network and click-through prediction with factorization machines. In: Proc. of the KDD-Cup Workshop. 2012. 113.

[55] Ding Y, Wang D, Xin X, *et al*. SCFM: Social and crowdsourcing factorization machines for recommendation. Applied Soft Computing, 2018,66:548−556.

[56] Zhou J, Wang D, Ding Y, *et al*. SocialFM: A social recommender system with factorization machines. In: Proc. of the Int'l Conf. on Web-Age Information Management. Cham: Springer-Verlag, 2016. 286−297.

[57] Qiang RW, Liang F, Yang JW. Exploiting ranking factorization machines for microblog retrieval. In: Proc. of the 22nd ACM Int'l Conf. on Information and Knowledge Management (CIKMn 2013). 2013. 1783−1788.

[58] Rendle S, Freudenthaler C, Gantner Z, *et al*. BPR: Bayesian personalized ranking from implicit feedback. In: Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence. AUAI Press, 2009. 452−461.

[59] Guo W, Wu S, Wang L, *et al*. Personalized ranking with pairwise factorization machines. Neurocomputing, 2016,214:191−200.

[60] Yuan FJ, Guo GB, Jose JM, Chen L, Yu HT, Zhang WN. Lambdafm: Learning optimal ranking with factorization machines using lambda surrogates. In: Proc. of the 25th ACM Int'l on Conf. on Information and Knowledge Management (CIKM 2016). 2016. 227−236.

[61] Yuan FJ, Guo GB, Jose JM, *et al*. Boostfm: Boosted factorization machines for top-*n* feature-based recommendation. In: Proc. of the 22nd Int'l Conf. on Intelligent User Interfaces. ACM Press, 2017. 45−54.

[62] Rendle S. Scaling factorization machines to relational data. Proc. of the VLDB Endowment, 2013,6(5):337−348.

[63] Liu H, He X, Feng F, *et al*. Discrete factorization machines for fast feature-based recommendation. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence. 2018. 3449−3455.

[64] Sun H, Wang W, Shi Z. Parallel factorization machine recommended algorithm based on MapReduce. In: Proc. of the Int'l Conf. on Semantics, Knowledge and Grids. IEEE, 2014. 120−123.

[65] Li M, Liu Z, Smola AJ, *et al*. Difacto: Distributed factorization machines. In: Proc. of the 9th ACM Int'l Conf. on Web Search and Data Mining. ACM Press, 2016. 377−386.

[66] Zhong E, Shi Y, Liu N, *et al*. Scaling factorization machines with parameter server. In: Proc. of the 25th ACM Int'l Conf. on Information and Knowledge Management. ACM Press, 2016. 1583−1592.

[67] Zhao K, Zhang J, Zhang L, *et al*. CDSFM: A circular distributed SGLD-based factorization machines. In: Proc. of the Int'l Conf. on Database Systems for Advanced Applications. Cham: Springer-Verlag, 2018. 701−709.

[68] Zhao KK, Zhang J, Zhang LF, *et al*. Signed network prediction method based on the client-to-client distributed framework. Ruan Jian Xue Bao/Journal of Software, 2018,29(3):614−626 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5447.htm [doi: 10.13328/j.cnki.jos.005447]

[69] Ma C, Liao Y, Wang Y, *et al*. F2M: Scalable field-aware factorization machines. In: Proc. of the MLSys on NIPS. 2016.

附中文参考文献:

[2] 黄立威,江碧涛,吕守业,等.基于深度学习的推荐系统研究综述.计算机学报,2018,41(7):1619−1647.

[9] 孟祥武,纪威宇,张玉洁.大数据环境下的推荐系统.北京邮电大学学报,2015,38(2):1−15.

[14] 丁伟峰,郑小林,陈德人.基于 PureSVD 模型的协同过滤主动采样.北京邮电大学学报,2013,36(4):23−26.

[17] 涂丹丹,舒承椿,余海燕.基于联合概率矩阵分解的上下文广告推荐算法.软件学报,2013,24(3):454−464. http://www.jos.org.cn/1000-9825/4238.htm [doi: 10.3724/SP.J.1001.2013.04238]

[68] 赵衍衍,张静,张良富,等.基于端到端分布式框架的符号网络预测方法.软件学报,2018,29(3):614−626. http://www.jos.org.cn/1000-9825/5447.htm [doi: 10.13328/j.cnki.jos.005447]

赵衍衍(1991－),男,陕西渭南人,博士生,主要研究领域为推荐系统,大数据分析.

张良富(1991－),男,博士生,主要研究领域为机器学习,数据挖掘.

张静(1984－),女,博士,讲师,CCF 专业会员,主要研究领域为数据挖掘,社会网络挖掘.

李翠平(1971－),女,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为社交网络分析,社会推荐,大数据分析及挖掘.

陈红(1965－),女,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为数据库技术,新硬件平台下的高性能计算.