

融合复杂先验与注意力机制的变分自动编码器

沈学利,马玉营,梁振兴

(辽宁工程技术大学 软件学院,辽宁 葫芦岛 125105)

摘要:传统变分自动编码器模型通常使用标准正态分布作为隐向量先验,当应用于推荐系统等复杂任务时容易导致模型过度正则化和隐向量解耦表现不佳。融合复杂隐向量先验与注意力机制,建立变分自动编码器模型。使用多层神经网络生成的隐向量先验分布替代标准正态分布作为假设先验分布,使得模型能根据数据学习先验分布并获得更多的潜在表征。在单层隐向量的基础上添加辅助隐向量,联合辅助隐向量与数据特征向量再生成隐向量,增强了隐向量的低维表现能力和解耦性。借助注意力机制的特征信息选择特点,对隐向量中重要节点赋予更大的权重值,使其能传递更重要的信息。在数据集 Movielens-1M、Movielens-Latest-Small、Movielens-20M 和 Netflix 上的实验结果表明,该模型的 Recall@20、Recall@50、NDCG@100 相较于基线模型平均提升了 12.95%、10.80%、10.48%,具有更高的推荐精确度。

关键词:推荐系统;协同过滤;深度学习;变分自动编码器;辅助隐向量;复杂先验;注意力机制

开放科学(资源服务)标志码(OSID):



中文引用格式:沈学利,马玉营,梁振兴.融合复杂先验与注意力机制的变分自动编码器[J].计算机工程,2022,48(11):55-61.

英文引用格式:SHEN X L, MA Y Y, LIANG Z X. Variational auto-encoder combining complex priori and attention mechanism[J]. Computer Engineering, 2022, 48(11): 55-61.

Variational Auto-Encoder Combining Complex Priori and Attention Mechanism

SHEN Xueli, MA Yuying, LIANG Zhenxing

(School of Software, Liaoning Technical University, Huludao, Liaoning 125105, China)

[Abstract] The traditional Variational Auto-Encoder(VAE) model typically uses the standard normal distribution as the implicit vector priori. When solving complex tasks such as the recommendation system, it easily leads to over-regularization of the model and poor performance of implicit vector decoupling. A VAE model combining a complex implicit vector priori and attention mechanism is built to solve these problems. First, the implicit vector prior distribution generated by a multilayer neural network is used to replace the standard normal distribution as the hypothetical prior distribution so that the model can learn the most appropriate prior distribution based on the data and obtain more potential representations. Next, an auxiliary implicit vector is added to regenerate the auxiliary implicit and data feature vectors into an implicit vector based on the single-layer implicit vector. Compared with the original structure, it can significantly improve the low-dimensional representation ability and decoupling of the implicit vector. Finally, based on the feature information selection of the attention mechanism, the attention mechanism is added to two implicit vectors to increase the weights of the critical nodes so that the implicit vector can transmit more important information. Experiments were performed on the public datasets: Movielens-1M, Movielens-Latest-Small, Movielens-20M, and Netflix. The results show that the proposed model has better evaluation indexes than the experimental comparison model on the Recall@20, Recall@50, and NDCG@100. The average increase values are 12.95%, 10.80%, and 10.48%, which show increased recommendation accuracy.

[Key words] recommendation system; collaborative filtering; deep learning; Variational Auto-Encoder(VAE); auxiliary implicit vector; complex priori; attention mechanism

DOI: 10.19678/j.issn.1000-3428.0063401

基金项目:辽宁省教育厅科学技术项目(LJ2020FWL001)。

作者简介:沈学利(1969—),男,教授,主研方向为智能信息处理、网络与信息安全;马玉营(通信作者)、梁振兴,硕士研究生。

收稿日期:2021-11-30 **修回日期:**2021-12-31 **E-mail:** myy950904@163.com

0 概述

随着互联网的快速发展,网络数据量呈指数级增长,用户面临信息过载问题^[1]。由于面对海量信息用户很难快速获取感兴趣信息,信息生产者也很难将信息推荐给需要的用户,因此推荐系统应运而生。推荐系统是基于对用户和物品的分析,推荐给用户最有可能感兴趣的物品^[2]。在推荐系统中协同过滤算法是使用较广泛的算法,其利用用户和项目的相似模式来提供个性化推荐^[3]。但是协同过滤算法过多依赖用户对项目的显式反馈或隐式反馈,存在数据稀疏的问题。然而深度神经网络的应用在很大程度上缓解了数据稀疏问题,推荐系统的准确度也得到较大提升^[4]。近些年,变分自动编码器(Variational Auto-Encoder, VAE)^[5]受到研究人员的广泛关注,也产生了很多的变体模型。文献[6]将VAE应用在协同过滤算法中,构建多项变分自动编码器(Multinomial Variational Auto-Encoder, Multi-VAE)模型,使用多项似然函数替代高斯似然函数,同时在损失函数中加入正则化项以增强模型的解耦能力和提高模型的推荐效果。考虑模型限制条件,文献[7]设计基于条件的变分自动编码器(Conditioned Variational Auto-Encoder, C-VAE)模型,该模型将标签条件加入到VAE中,同时对损失函数做出了改进,增加了模型的可解释性。文献[8]提出宏观和微观双层结构变分自动编码器(Macro-Micro Disentangled Variational Auto-Encoder, Macrid-VAE)模型,该模型使用多层神经网络获取高层次概念,再与用户行为进行结合,然后通过VAE获取微观表现,提高了模型精确度,同样也增加了模型可解释性。

上述研究成果提高了模型的可解释性和精确度,但却忽略了VAE模型本身的局限性。传统的VAE模型除了会导致模型过度正则化外,还会造成只有少量隐向量被激活,该情况会随着网络深度的加深变得更加严重^[7]。为此,本文提出融合复杂隐向量先验和注意力机制的变分自动编码器模型,简称为CCPAM-VAE。使用复杂隐向量先验分布替代标准正态分布,利用神经网络训练生成的伪输入作为假数据替代原数据,提高模型执行效率。通过添加辅助隐向量,增加隐向量的低维表现能力和解耦性。基于注意力机制强化具有较高权重值的重要信息,弱化具有较低权重值的不重要信息。

1 VAE模型

变分自动编码器由KINGMA等^[9]于2013年提出,是一个深度生成模型,结合了变分贝叶斯方法和神经网络,将求和问题转化为优化问题,大大增加了模型适用范围,广泛应用于自然语言处理、计算机视觉、推荐系统等领域^[10-12]。变分自动编码器的整体框架由编码器和解码器组成,因为编码器和解码器具有很大的灵活性,所以可以选择不同的神经网络

作为组成部分。用户-物品的评分矩阵(见图1)作为输入数据输入模型,解码器最终输出得到重构后的数据,通过最小化与输入数据的误差来优化模型参数。同时,重构后的数据包含了之前评分矩阵的未评分数据^[13-15]。VAE模型分为变分推理过程和生成过程,对于每个用户 u 的生成部分过程表示如下:

$$\mathbf{z}_u \sim p_\theta(\mathbf{z}_u) \quad (1)$$

$$\mathbf{x}_u | \mathbf{z}_u \sim p(x_u | \mathbf{z}_u) \quad (2)$$

$$p_\theta(x_u) = \int p_\theta(x_u | \mathbf{z}_u) p_\theta(\mathbf{z}_u) d\mathbf{z}_u \quad (3)$$

其中: \mathbf{z}_u 表示用户 u 的隐向量; $p_\theta(\mathbf{z}_u) \sim N(0, I)$ 表示隐向量 \mathbf{z}_u 的先验分布服从标准正态分布。

	物品					
	I_1	I_2	I_3	...	I_m	
用户	U_1	r_{11}	r_{12}	?	...	I_{1m}
	U_2	r_{21}	?	r_{23}	...	I_{2m}
	U_3	r_{31}	r_{32}	?	...	I_{3m}
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
	U_n	I_{n1}	I_{n2}	I_{n3}	...	I_{nm}

图1 用户-物品评分矩阵

Fig.1 User-item rating matrix

2 CCPAM-VAE模型

2.1 模型结构

本文设计的CCPAM-VAE模型整体采用编码器-解码器结构,如图2所示。在编码器阶段,输入数据经过密集块(Dense Block, DB)生成辅助隐向量 \mathbf{z}_1 ,将 \mathbf{z}_1 与上一层DB块的输出联合,作为下一个DB块的输入,生成隐向量 \mathbf{z}_2 。在解码器阶段,将辅助隐向量 \mathbf{z}_1 和隐向量 \mathbf{z}_2 经过注意力机制模块的输出联合,输入到DB块,得到的结果输入到Softmax函数中获得重构数据。在模型的编码器和解码器阶段,使用跳跃连接,加快模型优化速度,缓解深层网络的退化现象。

DB模块结构如图3所示。首先对输入数据 \mathbf{x} 进行Dropout^[16]操作,可以暂时屏蔽部分神经元,由于每次训练所屏蔽的神经元不一定相同,因此在一定程度上增加了模型的泛化能力;然后经过一个带有Tanh激活函数的全连接层进行特征提取;最后使用批量标准化(Batch Normalization, BN)进行归一化处理,加快模型收敛速度。DB模块的表达式如式(4)所示:

$$\mathbf{x}_1 = \text{BN}(\text{Tanh}(w_0 \times (\text{Dropout}(\mathbf{x}_0) + b_0))) \quad (4)$$

其中: w_0 是全连接层的权重值; b_0 是偏置; \mathbf{x}_0 是模块输入; \mathbf{x}_1 是模块输出。

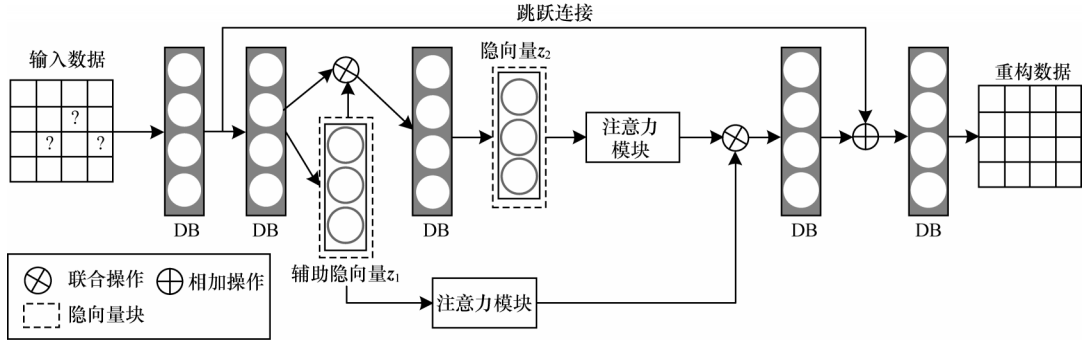


图2 CCPAM-VAE 模型结构

Fig.2 CCPAM-VAE model structure

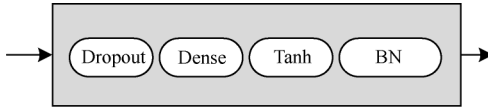


图3 DB模块结构

Fig.3 DB module structure

2.2 伪输入

2.2.1 伪输入计算

通过最大化下界 $\lg p_\theta(\mathbf{x}_u)$ 来优化参数 θ, φ , $\lg p_\theta(\mathbf{x}_u)$ 的下界如下:

$$\lg p_\theta(\mathbf{x}_u) \geq E_{q_\varphi(\mathbf{z}_u|\mathbf{x}_u)}[\lg p_\theta(\mathbf{x}_u|\mathbf{z}_u)] - \text{KL}(q_\varphi(\mathbf{z}_u|\mathbf{x}_u)||p_\lambda(\mathbf{z}_u)) = L(\mathbf{x}_u; \theta, \varphi, \lambda) \quad (5)$$

式(5)也被称为证据下界(Evidence Lower Bound, ELBO), 其中, $q_\varphi(\mathbf{z}_u|\mathbf{x}_u)$ 表示后验分布, $\text{KL}(\cdot)$ 表示 Kullback-Leibler 散度, 用来衡量两个概率分布之间的差距。

对 KL 散度部分进行分解可得到:

$$L(\mathbf{x}_u; \theta, \varphi, \lambda) = E_{q_\varphi(\mathbf{z}_u|\mathbf{x}_u)}[\lg p_\theta(\mathbf{x}_u|\mathbf{z}_u)] - E_{q_\varphi(\mathbf{z}_u|\mathbf{x}_u)}[\lg q_\varphi(\mathbf{z}_u|\mathbf{x}_u)] + E_{q_\varphi(\mathbf{z}_u|\mathbf{x}_u)}[-\lg p_\lambda(\mathbf{z}_u)] \quad (6)$$

其中, θ, φ, λ 分别代表不同的分布参数。

式(6)右边的第一部分是重构误差, 第二部分是变分后验分布的熵, 第三部分是变分后验和先验的交叉熵。使用拉格朗日乘子对第三部分求最大值, 如式(7)所示:

$$\max_{p_\lambda(\mathbf{z}_u)} -[E_{q_\varphi(\mathbf{z}_u|\mathbf{x}_u)}[-\lg p_\lambda(\mathbf{z}_u)]] + \beta \left(\int p_\lambda(\mathbf{z}_u) d\mathbf{z} - 1 \right) \quad (7)$$

后验分布如式(8)所示:

$$p_\lambda(\mathbf{z}_u) = \frac{1}{N} \sum_{i=1}^N q_\varphi(\mathbf{z}_u|\mathbf{x}_i) \quad (8)$$

其中: N 是数据集的大小。

由于求解先验分布需要使用整个数据集, 这样做不仅浪费计算资源, 还会导致模型过拟合, 因此使用可学习数据替代原数据, 可以很好地解决该问题, 如式(9)所示:

$$p_\lambda(\mathbf{z}_u) = \frac{1}{M} \sum_{i=1}^M q_\varphi(\mathbf{z}_u|\tilde{\mathbf{x}}_i) \quad (9)$$

其中: $\tilde{\mathbf{x}}_i$ 是一个 D 维的向量, 称为伪输入, 它是一组可

通过梯度下降法学习的变量; M 是伪输入数据的数量, 先验 $p_\lambda(\mathbf{z}_u)$ 的参数为 $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_M, \varphi\}$, 在实验中 $M \ll N$, 使用伪输入会大大减少计算量, 同时避免了使用真实数据计算先验导致模型出现过拟合的情况。

2.2.2 伪输入模块

如图4所示, 在伪输入模块中, 首先将输入数据 $\tilde{\mathbf{x}}$ 输入 a 个 DB_1 块中, 所得的结果再输入到 DB_1 块, 输出 μ_1 和 σ_1 (如式(10)所示), 然后使用隐向量块中的重参数化技术, 如图5所示, 其中 σ^2 表示方差, μ 表示均值, o 表示重采样, 将数据采样问题转化为标准正态分布采样问题, 最后由式(10)得到 \mathbf{z}_1 。

$$\mu_1 = \sigma_1 = \text{DB}_1(a \times \text{DB}_1(\tilde{\mathbf{x}})) \quad (10)$$

$$\mathbf{z}_1 = \mu_1 + \sigma_1 \times \varepsilon, \varepsilon \sim N(0, I) \quad (11)$$

接着将 \mathbf{z}_1 与式(10)上一层的输出联合, 输入到 DB_2 块中得到 μ_2 和 σ_2 。使用重参数化技术得到 \mathbf{z}_2 。该过程不仅能够使得采样过程独立, 而且可以通过反向传播更新参数。

$$\mu_2 = \sigma_2 = \text{DB}_2((a \times \text{DB}_1(\tilde{\mathbf{x}})) \otimes \mathbf{z}_1) \quad (12)$$

$$\mathbf{z}_2 = \mu_2 + \sigma_2 \times \varepsilon \quad (13)$$

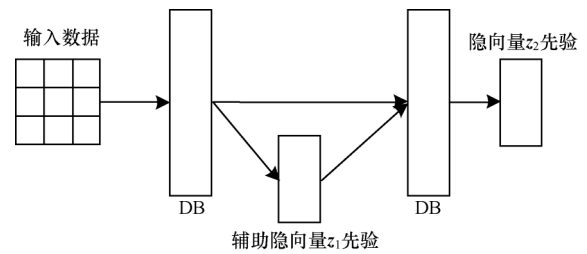


图4 伪输入模块结构

Fig.4 Pseudo input module structure

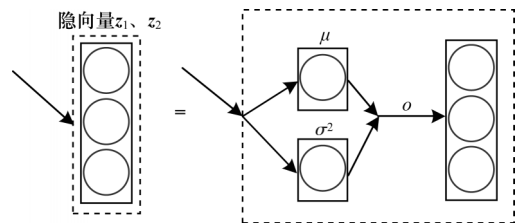


图5 隐向量模块

Fig.5 Hidden vector module

2.3 辅助隐向量

在训练过程中,传统的变分自动编码器模型经常出现隐向量非激活情况,这种情况会随着网络深度的加深变得更加严重。为解决该问题,在模型中添加辅助隐向量,双隐向量结构如图6所示,包含 x 作为输入数据的变分过程和 x 作为生成数据的生成过程。

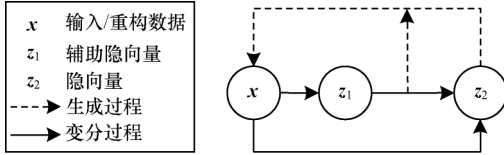


图6 双隐向量结构

Fig.6 Double implicit vector structure

变分过程如式(14)所示:

$$q_{\phi}(z_u^2|x_u, z_u^1)q_{\delta}(z_u^1|x_u) \quad (14)$$

生成过程如式(15)所示:

$$p_{\theta}(x_u|z_u^1, z_u^2) \quad (15)$$

其中: $p(z_u^2)$ 是输入获取的先验分布。

最终得到带有辅助隐向量的损失函数,如式(16)所示:

$$\begin{aligned} L(x_u; \theta, \phi, \lambda) = & E_{q_{\phi}(z_u^2|x_u, z_u^1)q_{\delta}(z_u^1|x_u)}[\lg p_{\theta}(x_u|z_u^1, z_u^2)] - \\ & E_{q_{\phi}(z_u^2|x_u, z_u^1)q_{\delta}(z_u^1|x_u)}[\lg(q_{\phi}(z_u^2|x_u, z_u^1)q_{\delta}(z_u^1|x_u)) + \lg q_{\delta}(z_u^1|x_u)] - \\ & E_{q_{\phi}(z_u^2|x_u, z_u^1)q_{\delta}(z_u^1|x_u)}[-\lg p_{\lambda}(z_u^1) - \lg p(z_u^2)] \end{aligned} \quad (16)$$

2.4 隐向量注意力机制

随着近些年深度学习的快速发展,注意力机制已成为计算机视觉、自然语言处理、语音识别等领域的重要技术^[17-19]。Transformer^[20]等神经网络模型与注意力机制相结合后,性能得到大幅提升。

隐向量作为用户-物品评分数据的高维表示,对于重构数据起着至关重要的作用。因为隐向量非激活问题也同样影响着重构数据的质量,所以本文通过融合注意力机制,对隐向量中含有重要信息的节点赋予更大的权重值,增强重要信息在重构数据中的作用;对于不重要的信息赋予更小的权重值,弱化其在重构数据中的影响,同时也减少了噪声。

在注意力模块中,首先将隐向量输入函数 Z 生成注意力得分,然后经过Softmax函数,对注意力得分进行0-1范围映射,将其转化为和为1的概率值,最后将概率值与隐向量做乘积,如式(17)~式(19)所示,注意力模块结构如图7所示。函数 Z 的选择很多,本文仅使用带有激活函数的全连接层作为函数 Z 。

$$Z = \text{ReLU}(w \cdot x) \quad (17)$$

$$a = \text{Softmax}(Z) \quad (18)$$

$$o_{\text{out}} = a \times x \quad (19)$$

其中: w 是全连接层权重; Z 是注意力评分; a 是经过0-1范围映射的评分值; o_{out} 是该模块的输出。

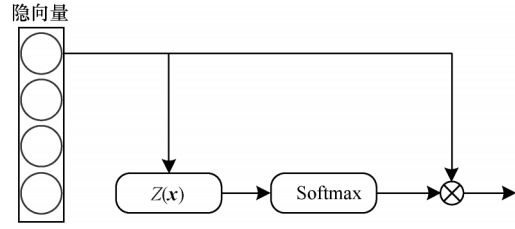


图7 注意力模块

Fig.7 Attention module

3 实验与评估

3.1 实验数据

采用4个公开的数据集 MovieLens-1M (ML-1M)、MovieLens-Latest-Small (ML-LS)、MovieLens-20M (ML-20M)、Netflix 测试模型性能,数据集包含用户对物品的评分,评分范围为[1,5]。数据集介绍具体如下:

1) ML-1M 数据集是 MovieLens 数据集之一,包括 6×10^3 个用户、 4×10^3 部电影和 1×10^8 个电影评分。

2) ML-LS 数据集来源于 MovieLens 电影网站,能够进行自动采集更新,包含 6×10^2 个用户、 9×10^3 部电影和 1×10^5 个电影评分。

3) ML-20M 数据集来源于 MovieLens 电影网站 2015 年 4 月提供的数据,包括 1.38×10^5 个用户、 2.7×10^4 部电影和 2×10^7 个电影评分。

4) Netflix 数据集来源于电影租赁网站 Netflix 2005 年公布的数据,包含 4.8×10^5 个用户、 1.7×10^4 部电影和 1×10^8 个电影评分。

数据集详细信息如表1所示。

表1 数据集详细信息

Table 1 Dataset details

数据集	用户数	物品数	评分数	稀疏性/%
ML-LS	600	6 200	48 000	1.279
ML-1M	6 000	3 500	575 000	2.699
ML-20M	136 000	20 000	9 990	0.353
Netflix	437 000	14 000	56 600	0.919

3.2 数据处理

为测试模型在强泛化下的性能,对数据集进行如下处理^[20]:

1) 将数据集划分为训练集(80%)、验证集(10%)、测试集(10%)3个部分,使用训练集的用户历史点击记录训练模型。首先从验证集和测试集的用户中获取一部分历史点击记录了解模型性能表现,然后通过查看模型对其他未显示的历史点击记录的排名来计算性能指标。

2) 对数据集进行预处理。首先将用户-项目评分表转化为值为0或1的二值化矩阵,将其作为隐式反馈表,具体方式是将评分大于3.5的值设为1,评分小于或者等于3.5的值设为0;然后移除评分次数少于5的用户。

3.3 实验环境设置

CCPAM-VAE模型算法由Python语言实现。实验硬件设备为Intel® Core™ i7-8700 @3.2 GHz处理器, GeForce RTX 2080Ti 11 GB显卡, 32 GB运行内存。为了使模型达到较好的效果, 使用网格搜索法进行超参数搜索。超参数搜索范围如表2所示。使用Adam优化器^[21]训练模型, 使用ReLU作为层与层之间的激活函数。采用Xavier随机初始化^[22]对模型参数进行初始化, 保证了每层输出的方差不受该层输入个数影响, 且每层梯度的方差也不受该层输出个数影响。

表2 超参数取值范围

Table 2 Super-parameter value ranges	
超参数	取值范围
学习率	{0.01, 0.005, 0.002, 0.001}
隐向量维度	{50, 100, 200, 300}
隐藏层维度	{100, 200, 300, 400, 500}
伪输入维度	{100, 200, 300, 400, 500}
批大小	{128, 256, 512, 1 024}

3.4 基线模型

Multi-DAE^[5]将DAE应用于协同过滤算法, 并使用多项式逻辑似然函数替代经典高斯似然函数, 实验结果证明其取得了不错的效果。

Multi-VAE^[5]将VAE应用于协同过滤算法, 并使用 β -VAE^[4]损失函数, 其中超参数 β 作为重建损失和KL损失之间的权衡参数添加到损失函数中。

Macrid-VAE^[7]在Multi-VAE的基础上, 将神经网络拟合生成的类别向量矩阵与输入相乘, 对输入数据高维度和低维度解耦, 增强了模型的可解释性。

CDAE^[7]将用户潜在因素作为额外输入, 添加到标准去噪自动编码器模型中。与标准去噪自动编码器模型一样, 使用Dropout方式对输入数据添加噪声。

NCF^[23]构建用于协同过滤的神经网络特征学习

框架, 并将用户和物品潜在特征输入到该框架中。

3.5 评估指标

在测试集上, CCPAM-VAE模型使用NDCG@K和Recall@K两个评估指标为每个用户预测项目排名, 并将其与真实项目排名进行比较。

DCG@K为折损累计增益, 是一种对排序结果的评价指标, 用户 u 的DCG@K如式(20)所示, 其中, $I[\cdot]$ 是指示函数, $r(i)$ 是排名为 i 的物品, T_u 是用户 u 点击过的物品集合。NDCG@K是DCG@K经过归一化后的值, 取值范围为 $[0, 1]$ 。

$$DCG@K_u = \sum_{i=1}^k \frac{2^{I[r(i) \in T_u]} - 1}{\lg(i + 1)}$$

(20)

Recall@K为召回率, 是一种衡量推荐效果的指标, 用户 u 的Recall@K如式(21)所示, 其中分母是 K 和用户点击的物品数的最小值。

$$Recall@K_u = \frac{\sum_{i=1}^K I[r(i) \in T_u]}{\min(K, |T_u|)}$$

(21)

3.6 实验结果分析

3.6.1 模型性能对比

表3从Recall@20、Recall@50、NDCG@100 3个指标对CCPAM-VAE模型进行评估, 在数据集ML-1M上, 指标平均提升了10.79%、5.12%、8.74%; 在数据集ML-LS上, 指标平均提升了26.20%、22.26%、17.42%; 在数据集ML-20M上, 指标平均提升了7.47%、6.97%、6.97%; 在数据集Netflix上, 指标平均提升了7.34%、8.86%、8.82%。上述结果表明, 相较于基线模型, 本文模型不仅在大数据集ML-1M、ML-20M、Netflix上有较好的表现, 在小数据集ML-LS上也有不错的效果。这表明复杂先验能够帮助模型学习更加复杂的数据关系, 减少模型出现过拟合的情况。同时, 基线模型Macrid-VAE在这4个数据集上的表现要优于其他基线模型, 这表明高维度解耦能够显著提升性能。

表3 CCPAM-VAE模型与基线模型的性能对比

Table 3 Performance comparison among CCPAM-VAE model and baseline models							
数据集	性能指标	Multi-DAE	Multi-VAE	Macrid-VAE	CDAE	NCF	CCPAM-VAE
ML-1M	Recall@20	0.349 14	0.330 56	0.353 58	0.346 62	0.335 31	0.366 24
	Recall@50	0.480 91	0.475 29	0.488 60	0.478 30	0.476 82	0.499 62
	NDCG@100	0.417 26	0.398 12	0.421 95	0.408 21	0.404 28	0.432 93
ML-LS	Recall@20	0.290 68	0.267 48	0.328 31	0.278 37	0.272 88	0.344 37
	Recall@50	0.399 05	0.365 72	0.409 35	0.388 19	0.380 25	0.447 13
	NDCG@100	0.305 66	0.278 88	0.304 27	0.290 33	0.286 61	0.327 46
ML-20M	Recall@20	0.388 20	0.382 63	0.392 29	0.383 25	0.385 29	0.411 20
	Recall@50	0.528 26	0.517 23	0.526 83	0.519 38	0.520 69	0.553 29
	NDCG@100	0.417 75	0.410 24	0.420 35	0.413 67	0.412 54	0.442 53
Netflix	Recall@20	0.340 22	0.338 08	0.341 65	0.339 50	0.339 25	0.362 90
	Recall@50	0.425 59	0.421 05	0.427 33	0.424 18	0.423 38	0.458 36
	NDCG@100	0.378 34	0.366 82	0.380 82	0.372 55	0.371 82	0.399 18

3.6.2 不同隐藏层维度和K值对模型性能的影响

图8给出了当隐藏层维度 d 取100、200、300、400、500而其他参数不变的情况下,CCPAM-VAE模型在4个数据集上的NDCG@100值。从图8可以看出,隐藏层维度的增加不一定能带来模型性能的提升,过高维度或者过低维度会导致模型过拟合或者欠拟合的现象,使模型表现不佳。对于数据集ML-LS,在隐藏层维度 $d=200$ 时,模型性能达到最佳。对于数据集ML-1M、ML-20M、Netflix,隐藏层维度 $d=300$ 时,模型性能达到最佳。这说明对于不同规模的数据集,模型达到最佳性能所需的隐藏层维度不一定是相同的。

图9给出了当 K 取5、10、15、20时,CCPAM-VAE模型和基线模型在4个数据集上的NDCG@ K 值。在数据集ML-20M、Netflix、ML-1M上,当 K 取5时,NDCG@ K 的值几乎最大,这表明在数据量比较大的情况下,CCPAM-VAE模型和基线模型不仅适合对物品进行Top- K 推荐,而且适合应用于推荐数量有限的场

景。在数据量较小的数据集ML-LS中,随着 K 值的增大,NDCG@ K 的值也不断增大,这表明在数据量比较小的情况下,CCPAM-VAE模型和基线模型不适合进行Top- K 推荐。由此可见,CCPAM-VAE模型在4个数据集上的表现均优于基线模型。

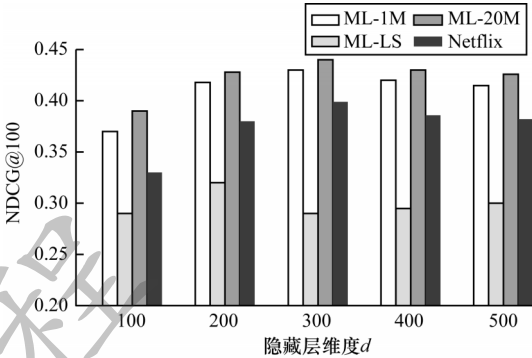


图8 隐藏层维度对NDCG@100的影响

Fig.8 Impact of hidden layer dimensions on NDCG@100

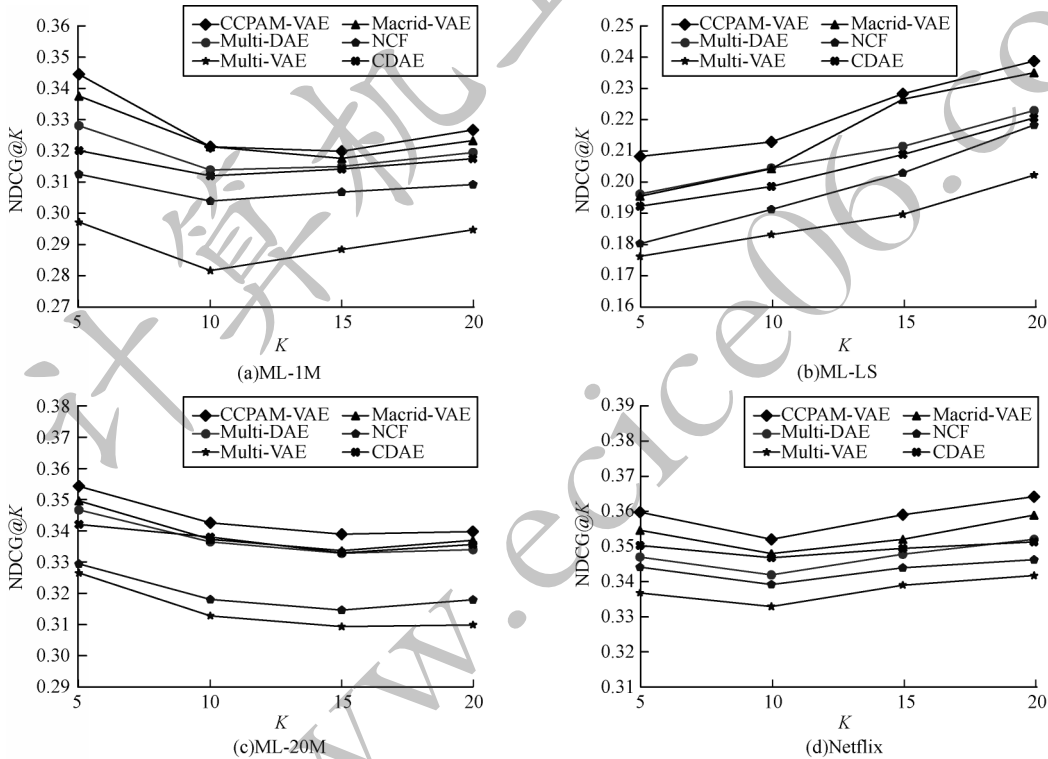


图9 CCPAM-VAE模型与基线模型在4个数据集上的NDCG@K对比

Fig.9 NDCG@K comparison among CCPAM-VAE model and baseline models on four datasets

3.6.3 消融实验分析

在数据集ML-1M、ML-LS、ML-20M和Netflix上对CCPAM-VAE模型的复杂先验和注意力模块进行有效性验证,结果如表4所示。其中,#-表示不使用复杂先验和注意力模块,#a表示使用复杂先验,#b表示添加注意力模块,#ab表示使用复杂先验和注意力模块。从表4可以看出,独立使用复杂先验和添加注意力模块均能提升模型精确度,将两者同时加入CCPAM-VAE模型,可使模型精确度达到最佳。

表4 CCPAM-VAE模型消融实验的Recall@50对比

Table 4 Recall@50 comparison of ablation experiment of CCPAM-VAE model

模型	ML-1M	ML-LS	ML-20M	Netflix
CCPAM-VAE#-	0.482 19	0.433 20	0.535 25	0.442 62
CCPAM-VAE#a	0.491 08	0.444 15	0.548 26	0.454 29
CCPAM-VAE#b	0.489 23	0.440 64	0.539 48	0.448 27
CCPAM-VAE#ab	0.499 62	0.447 13	0.553 29	0.458 36

4 结束语

本文提出融合复杂隐向量先验与注意力机制的变分自动编码器模型, 使用复杂隐向量先验分布使模型获得更多的潜在表征, 通过添加辅助隐向量增加隐向量的低维表现能力和解耦性。在公开的数据集 Movielens-1M、Movielens-Latest-Small、Movielens-20M 和 Netflix 上, 基于 Recall@20、Recall@50 和 NDCG@100 指标进行性能评估, 验证了该模型相较于基线模型能够有效提高推荐精确度。但由于本文仅将评分矩阵作为模型输入, 在后续研究中还将在变分自动编码器中融入知识图谱、图神经网络^[24-25]等更复杂的数据形式和数据关系, 进一步研究其对推荐效果的影响。

参考文献

- [1] 刘华玲, 马俊, 张国祥. 基于深度学习的内容推荐算法研究综述[J]. 计算机工程, 2021, 47(7): 1-12.
LIU H L, MA J, ZHANG G X. Review of studies on deep learning-based content recommendation algorithms[J]. Computer Engineering, 2021, 47(7): 1-12. (in Chinese)
- [2] FELFERNIG A, JERAN M, NINAUS G, et al. Toward the next generation of recommender systems: applications and research challenges[M]//TSIRINTZIS G A, VIRVOU M, JAIN L C. Multimedia services in intelligent environments. Berlin, Germany: Springer, 2013: 81-98.
- [3] HU Y F, KOREN Y, VOLINSKY C. Collaborative filtering for implicit feedback datasets[C]//Proceedings of the 8th IEEE International Conference on Data Mining. Washington D. C., USA: IEEE Press, 2009: 263-272.
- [4] HE X N, LIAO L Z, ZHANG H W, et al. Neural collaborative filtering[C]//Proceedings of International World Wide Web Conferences Steering Committee. Washington D. C., USA: IEEE Press, 2017: 173-182.
- [5] BURGESS C P, HIGGINS I, PAL A, et al. Understanding disentangling in β -VAE[EB/OL]. [2021-10-07]. <https://arxiv.org/abs/1804.03599>.
- [6] LIANG D W, KRISHNAN R G, HOFFMAN M D, et al. Variational autoencoders for collaborative filtering[C]//Proceedings of 2018 World Wide Web Conference. New York, USA: ACM Press, 2018: 689-698.
- [7] POLATO M, CARRARO T, AIOLLI F. Conditioned variational autoencoder for top-N item recommendation[EB/OL]. [2021-10-07]. <https://arxiv.org/abs/2004.11141>.
- [8] SIDDHARTH N, PAIGE B, DESMAISON A, et al. Learning disentangled representations for recommendation[EB/OL]. [2021-10-07]. http://stat.columbia.edu/~cunningham/teaching/GR8201/STAT_GR8201_2019_SPRG_slides.pdf.
- [9] KINGMA D P, WELING M. Auto-encoding variational Bayes[EB/OL]. [2021-10-07]. <https://arxiv.org/abs/1312.6114v4>.
- [10] MIAO Y S, YU L, BLUNSOM P. Neural variational inference for text processing[EB/OL]. [2021-10-07]. <https://arxiv.org/abs/1511.06038>.
- [11] LU Y, HUANG B. Structured output learning with conditional generative flows[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(4): 5005-5012.
- [12] TAHIR R, SARGANO A B, HABIB Z. Voxel-based 3D object reconstruction from single 2D image using variational autoencoders[J]. Mathematics, 2021, 9(18): 2288.
- [13] SOHN K, YAN X C, LEE H, et al. Learning structured output representation using deep conditional generative models[C]//Proceedings of International Conference on Neural Information Processing Systems. Cambridge, USA: MIT Press, 2015: 3483-3491.
- [14] BOUCHACOURT D, TOMIOKA R, NOWOZIN S. Multi-level variational autoencoder: learning disentangled representations from grouped observations[C]//Proceedings of AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 2018: 1-10.
- [15] QUANG Y X, LIU W Q, RONG W G, et al. Autoencoder-based collaborative filtering[M]. Berlin, Germany: Springer, 2014.
- [16] HINTON G E, SRIVASTAVA N, KRIZHEVSKY A, et al. Improving neural networks by preventing co-adaptation of feature detectors[EB/OL]. [2021-10-07]. <https://arxiv.org/abs/1207.0580>.
- [17] LUONG M T, PHAM H, MANNING C D. Effective approaches to attention-based neural machine translation[EB/OL]. [2021-10-07]. <https://arxiv.org/abs/1508.04025>.
- [18] SPERBER M, NIEHUES J, NEUBIG G, et al. Self-attentional acoustic models[EB/OL]. [2021-10-07]. <https://arxiv.org/abs/1803.09519v1>.
- [19] BRAUN S, NEIL D, ANUMULA J, et al. Multi-channel attention for end-to-end speech recognition[EB/OL]. [2021-10-07]. https://www.researchgate.net/publication/327388671_Multi-channel_Attention_for_End-to-End_Speech_Recognition.
- [20] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[EB/OL]. [2021-10-07]. <https://arxiv.org/abs/1706.03762>.
- [21] KINGMA D P, BA J. Adam: a method for stochastic optimization[EB/OL]. [2021-10-07]. <https://arxiv.org/abs/1412.6980>.
- [22] GLOROT X, BENGIO Y. Understanding the difficulty of training deep feedforward neural networks[C]//Proceedings of International Conference on Artificial Intelligence and Statistics. Washington D. C., USA: IEEE Press, 2010: 249-256.
- [23] HE X N, LIAO L Z, ZHANG H W, et al. Neural collaborative filtering[C]//Proceedings of the 26th International Conference on World Wide Web. Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2017: 1-8.
- [24] JI S, PAN S, CAMBRIA E, et al. A survey on knowledge graphs: representation, acquisition, and applications[EB/OL]. [2021-10-07]. <https://blog.csdn.net/byn12345/article/details/106128275>.
- [25] WANG X, HE X N, WANG M, et al. Neural graph collaborative filtering[C]//Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM Press, 2019: 165-174.