近年来, 以神经网络为核心的深度学习技术突飞猛进. 由于其高效的特征提取能力和非线性的学习方式, 越来越多的研究将深度学习应用于协同过滤方法中.

为解决线性模型性能差, 难以处理打分矩阵稀疏性的问题. 2007 年, Salakhutdinov 等人提出一种基于受限玻尔兹曼机的协同过滤推荐模型, 第一次将深度学习应用到推荐系统中[1]. Strub 等人采用两个栈式降噪自编码器 (SDAE), 分别学习用户和项目的隐因子, 然后通过隐因子模型对缺失评分进行预测[2]. Cheng 等人使用了一种深广学习模型处理多源数据, 该方法同时具有高的记忆能力和泛化能力[3]. Liang 等人首次将变分自编码器 (VAE) 应用到协同过滤模型中, 通过用户的隐式反馈数据预测缺失值数据[4]. He 等人将多层感知机和矩阵分解结合起来, 提供了协同过滤模型的一种通用架构[5]. 在混合推荐模型方面, 霍欢等人将栈式降噪自编码器应用于基于内容的推荐中, 并和协同过滤算法相结合[6]. 李晓菊等人先用循环神经网络和变分自编码器处理商品的文本信息, 再与概率矩阵分解相结合预测商品的缺失评分[7].

在以上研究文献的基础上, 本文提出了基于聚类变分自编码器的协同过滤算法. 该方法利用神经网络拟合的概率图模型学习用户的隐式反馈数据, 与传统的聚类方式不同, 它允许我们同时无监督地完成聚类和生成, 并且, 生成器以多项式分布的方法来训练重构数据.

本文的主要贡献如下:

(1) 与以往对用户和内容的特征进行聚类的方法不同, 本文直接将隐变量特征设定为带有聚类效果的二元变量, 将聚类统一到算法的整体框架中;

(2) 在大规模数据上对四种模型进行了实验, 对其性能进行了评价和对比, 并且对正则项的超参数进行了研究, 避免了过度正则化.

2 变分自编码器

变分自编码器[8]是一种无监督的生成模型, 其结构如图 1 所示. 它将神经网络技术与概率图模型结合在一起, 能够拟合出原始数据所服从的分布, 同时能够生成出类似的数据. 对于每一个用户u, 它都对应着一组数据x_u, 同时对应着一个服从标准正态分布的K维隐变量z_u. 对z_u进行采样, 生成重构数据x'_u, 其服从条件概率$p_\theta(x_u|z_u)$. 由于该条件概率无法直接求出, 可以用一个非线性函数$f_\theta(z_u)$进行替代. 该函数是一个带有参数θ的多层神经网络, 其输出为使用 softmax 函数进行了归一化的概率矩阵$\pi(f_\theta(z_u))$. 本文将$p_\theta(x_u|z_u)$设定为多项式分布, 希望通过优化参数θ使该函数能够以尽可能大的概率生成类似x_u的数据, 损失函数公式:

$$\log p_\theta(x_u|z_u) = \sum_i x_{ui} \log \pi_i(f_\theta(z_u)) \tag{1}$$

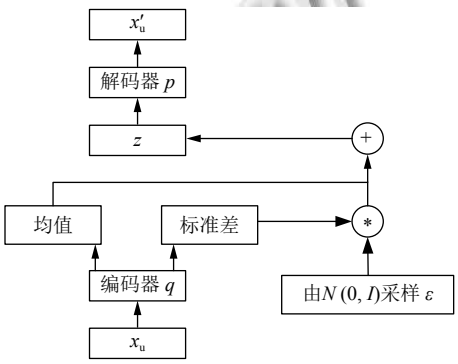

图 1 变分自编码器结构图

生成模型的目标就是通过最大化条件概率$p_\theta(x_u|z_u)$进而最大化重构数据的产生概率$p(x_u)$, 使重构数据尽量接近原始数据. 但仅靠随机采样一组隐变量是无法与其生成数据一一对应的, 还需要构建其与原始数据x_u的概率关系来获得隐变量的分布参数. 所以, 我们用贝叶斯变分推断的方法构造一个高斯分布$q_\phi(z_u|x_u)$来对隐变量进行采样. 采样的参数实质上是神经网络生成的均值(μ)和标准差(σ)两个 K 维向量. 编码器产生的分布是否接近标准分布是使用 KL 散度来计算的. 所以用编码器构建的神经网络所计算出的条件概率$q_\phi(z_u|x_u)$来近似真实后验概率$p_\theta(z_u|x_u)$, 两者之间的相似度:

$$\begin{aligned} &KL(q_\phi(z_u|x_u)\|p_\theta(z_u|x_u)) = \\ &E_{q_\phi(z_u|x_u)}[\log q_\phi(z_u|x_u) - \log p_\theta(z_u|x_u)] \end{aligned} \tag{2}$$

由于 KL 散度非负, 可以将式 (2) 变化, 得到:

$$\log p_\theta(x_u) \geq L(x_u;\theta,\phi) \tag{3}$$

其中,

$$\begin{aligned} L(x_u;\theta,\phi) =& E_{q_\phi(z_u|x_u)}[\log p_\theta(x_u|z_u)] \\ &- KL(q_\phi(z_u|x_u)\|p_\theta(z_u)) \end{aligned} \tag{4}$$

式 (4) 为变分自编码器的变分下界, 在最大化变分下界时, $\log p_\theta(x_u)$也在增加. 因此模型的优化目标可以

转化为最大化式 (4).

但是, 均值与方差都是用神经网络算出来的, 然后再对其进行随机采样, 由于随机采样不是一个连续过程, 无法求导, 但采样的结果可以求导, 以此可以实现反向传播以优化网络参数. 因此, 我们用一个随机变量ε对隐变量进行重参数化, 可得:

$$z=\mu+\varepsilon*\sigma,\varepsilon\sim N(0,I) \tag{5}$$

3 推荐算法

3.1 构建点击矩阵

本文用元素$i\in\{1,\cdots,I\}$索引每个内容, 将每个用户u的数据设为向量$x_u=[x_{u1},\cdots,x_{uI}]^{\mathrm{T}}$, 其中, x_{ui}代表用户u对内容i的打分值. 因为实验所用数据为 MovieLens 数据集, 所以打分值大小为 1 到 5. 但为了提高推荐的预测准确率, 本文将x_u转换为隐式反馈数据, 先筛选出观看数超过五部电影的用户再保留评分大于等于 4 的电影, 将这些电影的打分值转化为 1, 表示用户所点击过的喜爱的项目, 最后用 0 填充缺失值.

3.2 构建 CVAE 算法

本文将隐变量设置为二元变量(z,y), 其中z为连续变量, 代表着对交互特征进行编码的编码向量, 而y为离散变量, 代表着聚类类别, 可以在隐变量计算阶段完成对特征的聚类[9]. 因离散变量y是在连续变量z的基础上计算而得, 我们假设:

$$q_\phi(z_u,y_u|x_u)=q_\phi(y_u|z_u)q_\phi(z_u|x_u) \tag{6}$$

$$p_\theta(x_u|z_u,y_u)=p_\theta(x_u|z_u) \tag{7}$$

于是, 有:

$$KL(q_\phi(z_u,y_u|x_u)\|p_\theta(z_u,y_u|x_u))=\sum_y\iint q_\phi(y_u|z_u)q_\phi(z_u|x_u)\ln\frac{q_\phi(y_u|z_u)q_\phi(z_u|x_u)}{p_\theta(x_u|z_u)p_\theta(z_u|y_u)}dzdx \tag{8}$$

由第 1 节可知, z_u是服从标准正态分布的, 所以$p_\theta(z_u|y_u)$是服从均值为μ_y, 方差为 1 的正态分布, μ_y为解码网络参数之一; $p_\theta(y)$为均匀分布即各类别的电影数量大致相同; $q_\phi(y_u|z_u)$是对隐变量z_u的分类器, 可以通过 softmax 网络进行拟合. 因此, 可以得到:

$$L(x_u;\theta,\phi)=E_{q_\phi(z_u|x_u)}[\log p_\theta(x_u|z_u)]-\sum_y q_\phi(y_u|z_u)\log\frac{q_\phi(z_u|x_u)}{p_\theta(z_u|y_u)}-KL(q_\phi(y_u|z_u)\|p_\theta(y_u)) \tag{9}$$

模型优化目标为最大化式 (9). 神经网络的激活函数均为 tanh, 而最后一层的 Softmax 分类网络的输出$\pi(f_\theta(z_u))$为模型的归一化概率, 其参与到服从多项式分布的重构误差中进行网络优化, 以使更多的概率分配给更有可能被观看的电影项目.

3.3 引入正则化系数

式 (9) 中的第二和第三项可看作是重构误差项的正则化表达式, 以避免其过拟合. 同时, 为了权衡拟合效果, 本文引入了参数β来控制正则化的强度[10], 再将优化目标转换为最小化损失函数:

$$L(x_u;\theta,\phi)=-E_{q_\phi(z_u|x_u)}[\log p_\theta(x_u|z_u)]+\beta\sum_y q_\phi(y_u|z_u)\log\frac{q_\phi(z_u|x_u)}{p_\theta(z_u|y_u)}+\beta KL(q_\phi(y_u|z_u)\|p_\theta(y_u)) \tag{10}$$

如果$\beta<1$, 那么会削弱正则项的影响, 也就是避免了过度正则化. 从模型角度来看, 该方法对于第二项避免了过度的聚类效果, 同时, 对于第三项避免了聚类类别的分布过度均衡, 这符合推荐内容多类别多标签、无法完全归纳到单一类的实际情况, 在实验中也展现了正则项参数的良好效果.

3.4 SDG 训练与预测

CAVE 的随机梯度下降算法 (SDG) 以一个训练样本x_u和其重构数据x_u'计算梯度$\nabla_\theta L$和$\nabla_\phi L$, 再对批量数据的梯度求均值, 利用该值更新网络的参数:

$$\theta=\theta-\alpha\frac{\sum\nabla_\theta L}{n} \tag{11}$$

$$\phi=\phi-\alpha\frac{\sum\nabla_\phi L}{n} \tag{12}$$

对于一个用户的历史数据x_u, 通过训练好的模型, 可以利用预测出的未归一化的多项式分布概率$f_\theta(z_u)$对所有的推荐项目进行排序.

4 实验研究

4.1 数据集及实验环境

本文实验所使用的数据为 MovieLens 100k、MovieLens 1M 和 MovieLens 20M 三个规模不同的公开数据集. 我们只保留至少观看过五部电影的用户, 最终输入模型的特征数据为用户的隐式反馈数据. 数据集的详细内容如表 1 所示.

表 1 数据集

	ML-100k	ML-1M	ML-20M
用户数	938	6034	136 677
电影数	1447	3533	20 720
打分量	55 361	575 272	9 990 682
稀疏度 (%)	95.92	97.30	99.65

本文实验所用语言为 python3.5, 深度学习框架为 tensorflow 1.9+keras 2.2, 操作系统为 Windows10, 处理器为 Intel(R) Core(TM) i7-7700 CPU @3.6 GHz, 内存为 8 GB.

4.2 评价方法

本文使用两个 top-K 排序的指标作为实验结果的评价方法, 分别是召回率Re$call@K$和归一化折扣累积增益$NDCG@K$. 同时, 定义$w(k)$为排名k的项目, $h[\cdot]$为等级关联性函数, 如果真正打过分的项目在预测集中则该函数值为 1, 否则为 0, I_u为测试集用户u评过分的项目集合. 两者的定义分别如下:

$$\text{Re}call@K = \frac{\sum_{k=1}^{K} h[w(k) \in I_u]}{\min(K, |I_u|)} \tag{13}$$

$$NDCG@K = Z_K \sum_{k=1}^{K} \frac{2^{h[w(k)\in I_u]} - 1}{\log(k+1)} \tag{14}$$

其中, Z_K是归一化系数, 表示$h[w(k) \in I_u] = 1$都成立的理想情况下, Z_K其后的累加项值的倒数. 因为都使用了归一化方法, 所以两指标的数值都在 0–1 之内.

4.3 实验参数设置与基线

4.3.1 基线

DAE[4]: 降噪自编码的训练过程中, 输入的数据有一部分是“损坏”的, 能够对“损坏”的原始数据编码、解码, 然后尽可能接近原始数据地预测打分矩阵的缺失值.

SDAE: 栈式降噪自编码器就是在数据部分“损坏”的基础上多个自编码器相接, 以完成逐层特征提取的任务, 最后得到的特征作为分类器的输入, 完成推荐项目的概率预测.

WMF[11]: 加权矩阵分解, 这是一种线性的、低秩的矩阵分解模型.

SLIM[12]: 稀疏线性模型, 该方法是基于物品相似度的推广形式.

CDAE[13]: 协同降噪自编码器通过向输入添加每个用户的潜在因子来表示用户偏好, 同时在隐变量层加入了偏置表示.

DAE 和 SDAE 在 ML-100k 和 ML-1M 上的评价结果由本文实验得出; WMF、SLIM 和 CDAE 在 ML-20M 上的实验数据源于文献[4].

4.3.2 参数

为了训练不同模型的性能, 我们把所有样本分为训练/验证/测试 3 个集合, 验证集和测试集的样本数一样. 同时, 对模型的输入层使用 dropout 方法, 对最后的输出使用 Softmax 层进行归一化. CVAE 模型的隐变量z到y的分类器结构为 200→n, 即聚类类别为 n 类, 激活函数为 Softmax. 其他参数如表 2 所示, I 为项目个数.

4.4 实验结果与分析

为了观察 β 值对算法评价结果的影响, 本文将其从 0 到 1 分成十份并使用 MovieLens1M 测试集数据进行计算, 发现 CVAE 协同过滤模型在 3 个指标上都随着 β 值的增加而先增后减, 门槛值均在 0.4 附近. 所以后续实验均将 β 值设置为 0.4, 以此为最优的 CVAE 协同过滤模型. 如图 2 所示.

由于聚类的类别数会影响到算法的性能[14], 本文从 0 到 50 依次选值进行实验. 发现当类别数为 20 时, 该算法在 3 个指标上的表现均最佳, 所以以此值为最优的超参数. 并且该大小也符合电影分类的类别数. 如图 3 所示.

表 2 实验超参数

	SDAE	DAE	CVAE
网络结构	I→1000→200→1000→I	I→200→I	I→1200→200→1200→I
激活函数	tanh	tanh	tanh
学习率	0.001	0.001	0.001
dropout	0.5	0.5	0.5
L2 正则系数	0.01	0.01	0
epoch	100	100	100
训练集批尺寸	100	100	100
验证集批尺寸	50	50	50
β	0	0	0.4

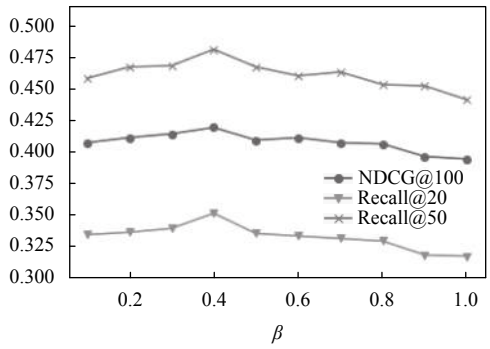

图 2 β 值对结果的影响

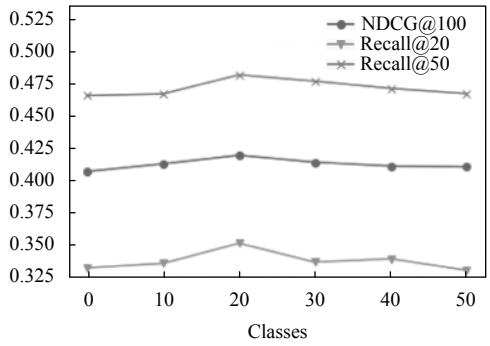

图 3 类别个数对结果的影响

图 4 显示了在 MovieLens1M 验证集上的 CVAE 协同过滤模型的 NDCG@100 值的迭代过程. 随着模型迭代次数的增加, 评价指标依次逐渐上升, 直至稳定. 实验最优的迭代次数大概在 60 代–80 代. 在之后 MovieLens 100K 和 MovieLens 20M 的实验中, 其走势与 MovieLens1M 的类似.

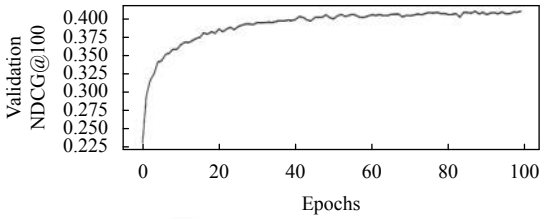

图 4 NDCG@100 指标的迭代

由表 3、表 4 和表 5 所知, CVAE 协同过滤模型在三个数据集上的八个评价结果上均优于基线. 但在 MovieLens 100K 数据集上的 Recall@50 指标表现最好的是 SDAE 模型, 并发现随着 K 值的增加, CVAE 方法的 Recall@K 值表现不如其他两个模型, 对于该方面的问题是由于 CVAE 方法在小规模数据集和高 K 值召回率上性能欠佳造成的, 还是由于其他原因造成的, 需要设置多个 K 值进一步实验, 对比研究. 除此之外, 可以看出对于更加稀疏、规模更大的打分矩阵, CVAE 的处理能力是更强的, 比基线方法表现出了更为优越的推荐性能.

表 3 MovieLens 1M

	Recall@20	Recall@50	NDCG@100
CVAE	**0.351**	**0.481**	**0.419**
DAE	0.345	0.474	0.416
SDAE	0.344	0.467	0.412

表 4 MovieLens 100K

	Recall@20	Recall@50	NDCG@100
CVAE	**0.435**	0.553	**0.491**
DAE	0.426	0.598	0.475
SDAE	0.403	**0.608**	0.455

表 5 MovieLens 20M

	Recall@20	Recall@50	NDCG@100
CVAE	**0.402**	**0.535**	**0.437**
WMF	0.36	0.498	0.386
SLIM	0.37	0.495	0.401
CDAE	0.391	0.523	0.418

5 结束语

本文提出了一种具有聚类效果的变分自编码器, 并将其运用到协同过滤推荐算法中. 该方法既能学习到用户和项目间的隐因子, 又可以在编码阶段完成对项目特征的聚类. 该模型还引入了正则化系数, 通过对其在 0–1 之间的研究, 发现了拟合效果更好的参数值. 最后, 以多项式分布对缺失值进行了预测. 该方法在 3 个规模不同的数据集上进行测试, 展现了其良好的推荐性能.

未来还可以使用自然语言处理技术处理文本信息, 将电影标签信息和用户评语融入到该算法中, 用混合模型提高推荐系统的性能.

参考文献

1 Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering. Proceedings of the 24th International Conference on Machine Learning. Corvalis, OR, USA. 2007. 791-798.

2 Strub F, Mary J. Collaborative filtering with stacked denoising autoencoders and sparse inputs. Proceedings of 2015 NIPS Workshop on Machine Learning for eCommerce. Montreal, Canada. 2015.

3 Cheng HT, Koc L, Harmsen J, *et al*. Wide & deep learning for recommender systems. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. Boston, MA, USA. 2016. 7-10.

4 Liang DW, Krishnan RG, Hoffman MD, *et al*. Variational autoencoders for collaborative filtering. arXiv: 1802. 05814, 2018.

5 He XN, Liao LZ, Zhang HW, *et al*. Neural collaborative filtering. Proceedings of the 26th International Conference on World Wide Web. Perth, Australia. 2017. 173-182.

6 霍欢, 郑德原, 高丽萍, 等. 栈式降噪自编码器的标签协同过滤推荐算法. 小型微型计算机系统, 2018, 39(1): 7–11. [doi: 10.3969/j.issn.1000-1220.2018.01.003]

7 李晓菊, 顾君忠, 程洁. 基于变分循环自动编码器的协同推荐方法. 计算机应用与软件, 2018, 35(9): 258–263, 280. [doi: 10.3969/j.issn.1000-386x.2018.09.046]

8 Kingma DP, Welling M. Auto-encoding variational bayes. arXiv: 1312. 6114, 2013.

9 苏剑林. 变分自编码器 (四): 一步到位的聚类方案. https://spaces.ac.cn/archives/5887, [2018-09-17].

10 Higgins I, Matthey L, Pal A, *et al*. *β*-VAE: Learning basic visual concepts with a constrained variational framework. Proceedings of 2017 International Conference on Learning Representations. Toulon, France. 2017. 1–13.

11 Hu YF, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. Proceedings of the 20088th IEEE International Conference on Data Mining. Pisa, Italy. 2009. 263-272.

12 Ning X, Karypis G. SLIM: Sparse linear methods for top-N recommender systems. Proceedings of the 2011 IEEE 11th International Conference on Data Mining. Vancouver, Canada. 2011. 497-506.

13 Wu Y, Dubois C, Zheng AX, *et al*. Collaborative denoising auto-encoders for top-N recommender systems. Proceedings of the 9th ACM International Conference on Web Search and Data Mining. San Francisco, CA, USA. 2016. 153-162.

14 Sedhain S, Menon AK, Sanner S, *et al*. AutoRec: Autoencoders meet collaborative filtering. Proceedings of the 24th International Conference on World Wide Web. Florence, Italy. 2015. 111-112.