

# 协同深度学习推荐算法研究<sup>①</sup>

冯楚滢, 司徒国强, 倪玮隆

(兰州交通大学 电子与信息工程学院, 兰州 730070)

通讯作者: 冯楚滢, E-mail: f\_chuy@163.com



**摘要:** 针对当用户评分较少时, 推荐系统由于数据稀疏推荐性能显著降低这一问题, 介绍了协同深度学习算法 (Collaborative In Deep Learning, CIDL). 本算法首先对大量数据进行深度学习, 然后对数据文本进行挖掘提取词汇表, 最后对评级 (反馈) 矩阵进行协同过滤, 从而得出对用户的推荐项目. 本文使用真实的电影数据进行实验, 与另外四种优秀算法进行对比, 证明该算法可以真实有效得解决由于数据稀疏使得性能降低的问题, 并提高推荐的准确度.

**关键词:** 深度学习; 推荐系统; 协同过滤; 文本挖掘; 堆叠降噪自编码器

引用格式: 冯楚滢, 司徒国强, 倪玮隆. 协同深度学习推荐算法研究. 计算机系统应用, 2019, 28(1): 169-175. <http://www.c-s-a.org.cn/1003-3254/6742.html>

## Research on Collaborative Deep Learning Recommendation Algorithm

FENG Chu-Ying, SITU Guo-Qiang, NI Wei-Long

(School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

**Abstract:** For the problem that when the user score is not enough, the recommender system significantly reduces the data sparse recommendation performance, a Collaborative In Deep Learning algorithm (CIDL) is proposed. The algorithm firstly conducts the deep learning on a large amount of data, and then performs collaborative filtering on the rating (feedback) matrix to arrive at a recommendation item for the user. This study uses real movie data to test and to compare it with the other four excellent algorithms. It is proved that CIDL can effectively solve the problem of reduced performance due to data sparseness and improve the accuracy of the recommendation.

**Key words:** recommender systems; deep learning; collaborative filtering; text mining; stacked denoising autoencoders

协同过滤 (Collaborative Filtering) 是许多推荐系统常用的方法<sup>[1]</sup>. 传统的基于协同过滤的方法使用用户给出的项目评分作为学习推荐的唯一信息来源. 但是, 许多应用程序的评分通常较少, 导致基于协同过滤的方法的推荐性能显著降低. 为了解决这个稀疏性问题, 本文利用诸如项目内容信息的辅助信息一同作为信息来源提出了称为协同深度学习 (Collaborative In Deep Learning, CIDL) 的算法, 该算法分别执行用于内容信息的深度表示学习以及用于评价 (反馈) 矩阵的协同过滤.

本文将协同深度学习算法应用于电影推荐系统, 对大量的电影评分进行建模, 分析观影用户对电影的偏好随着时间的推移而变化的情况, 从而对客户进行个性化电影推荐<sup>[2]</sup>. 本文的创新点主要在于: (1) 与以往使用分类<sup>[3]</sup>和重构<sup>[4]</sup>等简单目标的深度学习模型不同, 本文在概率框架中使用协同过滤作为更复杂的目标. (2) 除了获得最大后验概率 (Maximum A Posteriori, MAP) 估计的算法外, 本文还推导了一种基于采样的 CIDL 贝叶斯处理算法, 该算法是一种贝叶斯推广的反向传播算法. (3) 协同深度学习是一个用于弥合现有深度学

① 收稿时间: 2018-07-13; 修改时间: 2018-08-09; 采用时间: 2018-08-27; csa 在线出版时间: 2018-12-26

习模型和推荐系统之间差距的等级贝叶斯模型。此外, 由于其贝叶斯性质, CIDL 可以轻松扩展并纳入其他辅助信息以进一步提升性能。

## 1 相关工作

由于许多在线服务中用户有大量的选择, 推荐系统得到了大量运用<sup>[5]</sup>。对于个人而言, 使用推荐系统可以让我们更有效地使用信息。此外, 许多公司(例如淘宝和各大音乐播放器)已经广泛地使用推荐系统来定位他们的客户。推荐系统的现有方法大致可以分三类<sup>[6]</sup>: 基于内容的方法, 基于协同过滤的方法和混合方法。基于内容的方法利用用户程序或产品描述来进行推荐

基于协同过滤的方法使用过去的活动或偏好, 例如用户对项目的评分, 而不使用用户或产品信息。混合方法通过结合基于内容的方法和基于协同过滤的方法来获得两全其美的结果。

根据判断评分信息与辅助信息之间是否存在双向交互, 可以进一步将混合方法分为两类: 松耦合方法和紧密耦合方法。松散耦合的方法<sup>[7]</sup>一次处理辅助信息, 然后用它为协同过滤模型提供特征。由于信息流是单向的, 因此评级信息无法提供反馈来指导提取有用的功能。对于这个子类别, 改进通常需要依靠手动和乏味的特征工程过程。相反, 紧密耦合的方法允许双向交互<sup>[8]</sup>。一方面, 评分信息可以指导特征的学习。另一方面, 提取的特征可以进一步提高协同过滤模型的预测能力。通过双向交互, 紧密耦合的方法可以自动从辅助信息中学习特征, 并自然平衡评级和辅助信息的影响。

随着深度学习以及推荐系统的发展, 对于深度学习与推荐系统结合的研究也越来越多, 文献<sup>[9]</sup>选用了与本文相同的 SDAE 模型从用户自由文本标签中抽取项目的显式特征信息, 再使用改进隐因子模型算法对进行矩阵分解, 最终得出推荐矩阵, 但对于数据陷入局部最优的问题没有很好的解决。文献<sup>[10]</sup>中使用结合协同过滤推荐和基于内容推荐的混合模型, 其中基于内容推荐的部分使用了深度学习内容。但对于这种方法而言, 是将两种算法产生的结果按比例返回作为推荐结果。两种算法没有很好的融合在一起共同作用。按比例返回结果的比例也很难把控。文献<sup>[11]</sup>使用了两个堆叠降噪编码器来学习以得出推荐矩阵, 但预处理数据和训练模型时花费了较长的时间, 最优效果有待改进。文

献<sup>[12]</sup>引入深度学习模型提升算法性能, 但对于算法的调参提出了更高的要求, 并且算法的复杂度也相比传统算法提升不少。

本文介绍了一种称为协同深度学习模型 (Collaborative In Deep Learning, CIDL) 的分层贝叶斯模型, 作为推荐系统的一种新型紧耦合方法。协同深度学习模型基于一种称为堆叠降噪自动编码器 (Stacked Denoising AutoEncoder, SDAE) 的贝叶斯公式<sup>[13]</sup>。协同深度学习模型将内容信息的深度表示学习与评级 (反馈) 矩阵的协作过滤紧密结合在一起, 从而允许两者之间的双向交互。

## 2 本文算法

### 2.1 符号及公式说明

本文采用隐式反馈来训练和测试数据。整个项目数据  $J$  (电影) 的数据集合由一个  $j$  行  $S$  列的矩阵  $X_C$  来表示,  $J$  项目中的  $j$  行是由基于词汇库大小为  $S$  的词袋模型向量  $X_{C,j}$  组成的。对于整个用户数据集合  $I$ , 定义一个  $i$  行  $j$  列的二元评分矩阵  $R=R_{i,j}$ 。给定  $R$  中的部分评级和内容信息  $X_C$ , 用来预测  $R$  中的其他评级,  $R$  的维度为  $K$ 。  $X_0$  为噪声破坏输入, 同样是一个  $j$  行  $S$  列的矩阵,  $X_C$  的作用为净化输入到 SDAE 中的  $X_0$ 。  $X_1$  为 SDAE 的第 1 层的输出,  $X_1$  的  $j$  行表示为  $X_{1,j}$ 。  $W_l$  为 1 层的权重矩阵,  $b_l$  为偏差矢量。  $W^*$  表示所有层的权重矩阵和偏差矢量的集合。

### 2.2 算法流程

本文使用自动编码器学习数据并得到训练后的网络, 在利用训练后的网络生成用户和项目间的隐含特征矩阵。最后将该矩阵输入 CIDL 算法中训练学习, 最终得到新的评分矩阵。流程如图 1 所示。

图 1 中的具体步骤如下:

#### (1) 对数据进行预处理

本文中使用的数据由两部分组成。第一部分, 收视率和电影名称来自 Netflix 挑战数据集。第二部分, 本文从 IMDB (<http://www.imdb.com>) 收集了相应电影的情节。为了与前两个数据集的隐式反馈设置保持一致<sup>[14]</sup>, 本文中只提取正面评分 (评分 5) 进行培训和测试。在移除正面评级少于 3 个的用户和没有剧情的电影后, 最终数据集中有 407 261 个用户, 9228 部电影和 15 348 808 个评分。之后进行预处理<sup>[15]</sup>, 从电影的情节中提取文本信息 (项目内容)。在去除停止词后, 根据 tf-idf

(term frequency-inverse document frequency) 值选择最高的  $S$  个判别式词汇形成词汇表, 最后生成的词汇表大小为 20 000.

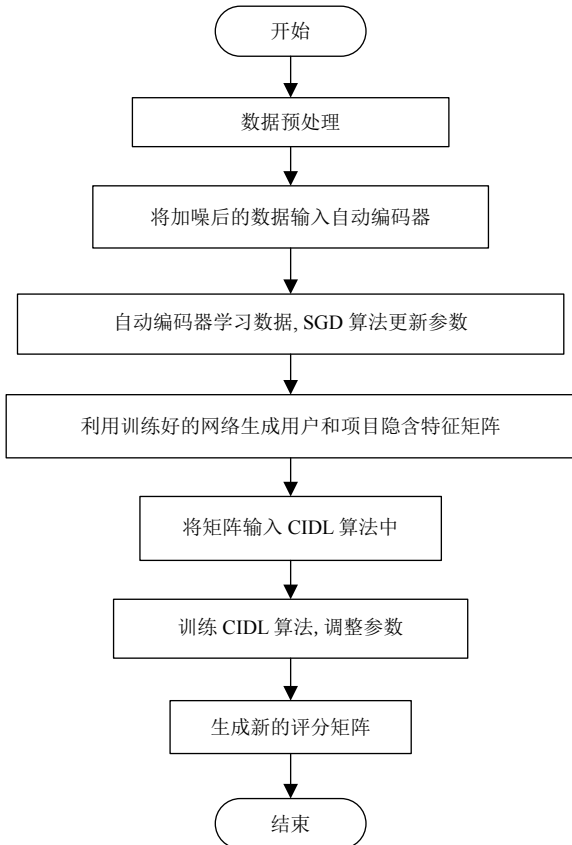


图1 算法流程图

(2) 训练 SDAE 自动编码器输出特征矩阵

对自动编码器中的每个用户输入  $R$  里的每一行、项目 ID 和项目特征向量,  $R$  里的每一行即每个用户对各个项目的打分作为其向量描述, 输入层个数等于项目数目  $J$ . 对于  $R$  加噪后的数据为:

$$\tilde{R}^{(u)} = (\tilde{R}_{u1}, \dots, \tilde{R}_{um}) \in \tilde{R}^n \quad (1)$$

训练用户评分矩阵的自动编码器的输出公式为:

$$\hat{R}^{(u)} = f(W'_1 g(W_1 \tilde{R}^{(u)} + b_1) + b'_1) \quad (2)$$

其中,  $W, W'$  分别是训练用户评分矩阵的自动编码器中输入层到隐藏层、隐藏层到输出层的权重矩阵,  $b, b'$  分别是偏置向量. 对于训练用户评分矩阵的自动编码器的最小损失函数为:

$$E_U = \frac{1}{2m} \sum_{u=1}^m \|R^{(u)} - \hat{R}^{(u)}\|^2 + \frac{\lambda}{2} \|W_1\|^2 + \frac{\lambda}{2} \|W'_1\|^2 \quad (3)$$

(3) 使用 CIDL 算法生成新的评分矩阵  
生成的具体过程如 2.5 小节所示.

2.3 堆叠降噪自动编码器

堆叠降噪自动编码器是一个前馈神经网络, 通过从叠加噪声的原始数据以及输出数据中学习得出未叠加噪声的入数据, 如图 2 所示.

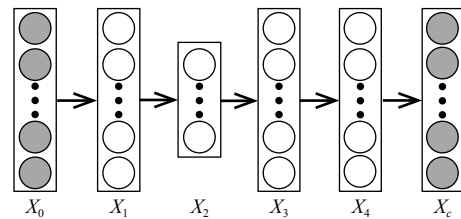


图2 SDAE 原理图

堆叠降噪自动编码器从叠加噪声的输入数据与未叠加噪声的输入数据中得出的特征几乎相同, 但从叠加噪声的数据中学习到的特征更具有鲁棒性. 通常情况下, 隐藏层在整个系统的中间 (即图 1 中的  $X_2$ ) 被约束为瓶颈<sup>[15]</sup>, 并且输入层  $X_0$  是叠加噪声的原始数据. 一个堆叠降噪编码器解决了去除噪声的优化问题, 即:

$$\min_{\{W_l, b_l\}} \|X_c - X_L\|_F^2 + \lambda \sum_l \|W_l\|_F^2 \quad (4)$$

其中,  $\lambda$  是正则化参数,  $\|\cdot\|$  是弗罗贝尼乌斯范数.

2.4 广义贝叶斯 SDAE

从  $X_L$  生成干净输入  $X_C$  是贝叶斯 SDAE 生成过程的一部分, 而从  $X_C$  生成噪声破坏输入  $X_0$  是一种人工噪声注入过程, 可帮助 SDAE 学习更强大的特征表示. 若假设干净输入  $X_C$  和损坏输入  $X_0$ , 可定义以下生成过程:

(1) 对于 SDAE 网络中的每一层  $l$

1) 对于权重矩阵  $NW_l$  的每一列  $n$

$$W_{l,*n} \sim \mathcal{N}(0, \lambda_W^{-1} I_{K_l}) \quad (5)$$

2) 绘制偏置向量

$$b_l \sim \mathcal{N}(0, \lambda_W^{-1} I_{K_l}) \quad (6)$$

3) 对于  $X_l$  的每一行  $j$

$$X_{l,j*} \sim \mathcal{N}(\delta(X_{l-1,j*}, W_l + b_l), \lambda_S^{-1} I_{K_l}) \quad (7)$$

(2) 对于每个  $j$ , 绘制一个干净的输入

$$X_{C,j*} \sim \mathcal{N}(X_{L,j*}, \lambda_n^{-1} I_j) \quad (8)$$

如果  $\lambda_S$  有效, 则方程 (4) 中的高斯分布将变为以

$\delta(X_{l-1,j^*}, W_l + b_l)$  为中心的狄拉克三角形分布<sup>[10]</sup>, 其中  $\delta(\cdot)$  是 S 形函数 (Sigmoid Function, SF). 该模型将退化 为 SDAE 的贝叶斯公式, 因此称之为广义 SDAE. 网络 中的第一个  $L/2$  层充当编码器, 最后的  $L/2$  层充当解码 器. 后验概率的最大化等同于考虑权重衰减时重建误 差的最小化.

### 2.5 协同深度学习

使用贝叶斯 SDAE 作为组件, CIDL 的原理图如 图 3 所示.

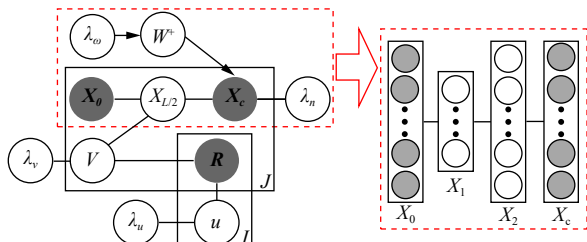


图 3 CIDL 原理图

CIDL 的生成过程如下:

(1) 对于 SDAE 网络的每一层  $l$ , 操作与前一部分 SDAE 中步骤相同.

(2) 对于每个  $j$ , 绘制一个干净的输入, 操作与 SDAE 中步骤相同.

1) 设置一个潜在的项目设置向量:

$$\varepsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K) \quad (9)$$

2) 并将潜在项目矢量设置为:

$$v_j = \varepsilon_j + X_{\frac{L}{2}, j^*}^T \quad (10)$$

3) 为每个用户设置一个潜在用户向量:

$$u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K) \quad (11)$$

4) 为每个用户项目对  $(i, j)$  绘制评分  $R_{i,j}$ :

$$R_{i,j} \sim \mathcal{N}(u_i^T v_j, C_{ij}^{-1}) \quad (12)$$

其中,  $\lambda_w, \lambda_n, \lambda_u, \lambda_s$  以及  $\lambda_v$  为超参数,  $C_{ij}$  是一个与 CTR 相似的置信参数, 中间层作为评级和内容信息之间的桥梁. 这个中间层, 以及潜在偏移量  $\varepsilon_j$  是使 CIDL 能够同时学习有效特征表示并捕获项目与用户之间相似性和隐含关系的关键.

### 2.6 最大后验估计

基于上述 CIDL 模型, 所有参数都可以作为随机变量处理, 以便可以应用完全贝叶斯方法, 如马尔科夫

链或变分近似方法<sup>[16]</sup>. 但是, 这种处理通常会导致高计算成本. 本文设计了一个期望最大算法来进行最大后验估计.

与协同主题回归模型类似, 最大化后验概率相当于最大化  $U, V, \{X_j\}, \{X_c\}, \{W_l\}, \{b_l\}$  的联合对数似然估计:

$$\begin{aligned} \mathcal{L} = & -\frac{\lambda_u}{2} \sum_i \|u_i\|_2^2 - \frac{\lambda_w}{2} \sum_l (\|W_l\|_F^2 + \|b_l\|_2^2) \\ & - \frac{\lambda_v}{2} \sum_j \|v_j - X_{\frac{L}{2}, j^*}^T\|_2^2 - \frac{\lambda_n}{2} \sum_j \|X_{L, j^*} - X_{c, j^*}\|_2^2 \\ & - \frac{\lambda_s}{2} \sum_l \sum_j \|\delta(X_{l-1, j^*} W_l + b_l) - X_{l, j^*}\|_2^2 \\ & - \sum_{i,j} \frac{C_{i,j}}{2} (R_{ij} - u_i^T v_j)^2 \end{aligned} \quad (13)$$

若  $\lambda_s$  趋近于无限, 此似然将会变为:

$$\begin{aligned} \mathcal{L} = & -\frac{\lambda_u}{2} \sum_i \|u_i\|_2^2 - \frac{\lambda_w}{2} \sum_l (\|W_l\|_F^2 + \|b_l\|_2^2) \\ & - \frac{\lambda_v}{2} \sum_j \|v_j - f_e(X_{0, j^*}, W^+)^T\|_2^2 \\ & - \frac{\lambda_n}{2} \sum_j \|f_r(X_{0, j^*}, W^+) - X_{c, j^*}\|_2^2 \\ & - \sum_{i,j} \frac{C_{ij}}{2} (R_{ij} - u_i^T v_j)^2 \end{aligned} \quad (14)$$

编码器函数  $f_e(\cdot, W^+)$  取未降噪的内容向量  $X_{0, j^*}$  作为输入并计算该项目的编码, 函数  $f_r(\cdot, W^+)$  同样取  $X_{0, j^*}$  作为输入并计算编码, 然后计算项目  $j$  的重构内容向量.

从优化的角度来看, 上述目标函数 (11) 中的第三项相当于一个多层感知器, 它使用潜在项向量  $v_j$  作为目标, 而第四项等价于一个使重建误差最小化的 SDAE. 从神经网络的角度来看, 当  $\lambda_s$  接近正无穷时, CIDL 的概率图模型的训练将退化为同时训练两个叠加在一起的共同输入层的神经网络. 由于参与评级矩阵, CIDL 网络比典型的神经网络复杂得多.

当  $\lambda_n/\lambda_v$  的比率接近正无穷时, CIDL 将退化为两步模型, 其中使用 SDAE 学习的潜在表示被直接放入协同主题回归模型中. 另一个临界点发生在  $\lambda_n/\lambda_v$  趋近于零时 SDAE 的解码器基本失去功能. 通过实验证实可得, 在这两种情况下, 模型预测性能都会大大提高.

对于  $u_i$  和  $v_j$ , 给定当前  $W^+$ , 计算  $\mathcal{L}$  相对于  $u_i$  和  $v_j$  的梯度并将它们设置为零, 得出以下结果:

$$u_i \leftarrow (VC_i V^T + \lambda_u I_k)^{-1} VC_i R_i \quad (15)$$

$$v_j \leftarrow (UC_i U^T + \lambda_v I_K)^{-1} (UC_j R_j + \lambda_v f_e(X_{0,j*}, W^+)^T) \quad (16)$$

其中,  $U = (u_i)_{i=1}^I, V = (v_j)_{j=1}^J, C_i = \text{diag}(C_{i1}, \dots, C_{iJ})$  为对角矩阵,  $R_i = (R_{i1}, \dots, R_{iJ})^T$  为包含用户所有评级的列向量.

对于给定  $U$  和  $V$ , 可以使用反向传播学习算法来学习权重  $W_l$  以及每个层的偏差矢量  $b_l$ . 关于  $W_l$  和  $b_l$  的可能性的梯度如下:

$$\begin{aligned} \nabla_{W_l} \mathcal{L} = & -\lambda_w W_l \\ & -\lambda_v \sum_j \nabla_{W_l} f_e(X_{0,j*}, W^+)^T (f_e(X_{0,j*}, W^+)^T - v_j) \\ & -\lambda_n \sum_j \nabla_{W_l} f_r(X_{0,j*}, W^+) (f_r(X_{0,j*}, W^+) - X_{c,j*}) \end{aligned} \quad (17)$$

$$\begin{aligned} \nabla_{b_l} \mathcal{L} = & -\lambda_w b_l \\ & -\lambda_v \sum_j \nabla_{b_l} f_e(X_{0,j*}, W^+)^T (f_e(X_{0,j*}, W^+)^T - v_j) \\ & -\lambda_n \sum_j \nabla_{b_l} f_r(X_{0,j*}, W^+) (f_r(X_{0,j*}, W^+) - X_{c,j*}) \end{aligned} \quad (18)$$

通过交替更新  $U, V, W_l$  和  $b_l$ , 可得到  $\mathcal{L}$  的局部最优值. 本文使用 SGD (Stochastic Gradient Descent, 随机梯度下降) [17] 算法也叫增量梯度下降法, 是在传统梯度下降法的基础上改进而来, 收敛速度快很多, 是常用的参数训练算法. SGD 通过最小化损失函数来训练模型参数, 根据每个样本来迭代更新一次, 可以控制下降速率和迭代次数, 使得参数的训练更可控, 以解决局部最优问题. 该算法过程如算法 1.

算法 1. 增量梯度下降法

Input: 评分矩阵  $R$ , 特征向量维度  $K$ , 学习率  $\eta$ , 比例参数  $a$ , 正则参数  $\lambda_u, \lambda_v, \lambda_Q$

Output:  $U, V$

- (1) 初始化: 用一个较小的值随机构造  $U, V$  和  $Q$ ;
- (2) while (error on validation set decrease);
- (3)  $\nabla_{U \mathcal{L}} = I(U^T V - R) V + a(U^T Q - H) Q + \lambda_u U$   
 $\nabla_{V \mathcal{L}} = [I(U^T V - R)]^T U + \lambda_v V$   
 $\nabla_{Q \mathcal{L}} = a(U^T Q - H)^T U + \lambda_Q Q$
- (4) set  $\eta = 0.1$
- (5) while ( $\mathcal{L}(U - \eta \nabla_{U \mathcal{L}}, V - \eta \nabla_{V \mathcal{L}}, Q - \eta \nabla_{Q \mathcal{L}}) > \mathcal{L}(U, V, Q)$ )
- (6) set  $\eta = \eta/2$
- (7) Endwhile
- (8)  $U = U - \eta \nabla_{U \mathcal{L}}$   
 $V = V - \eta \nabla_{V \mathcal{L}}$   
 $Q = Q - \eta \nabla_{Q \mathcal{L}}$
- (9) Endwhile
- (10) Return  $U, V$

本文对 SDAE 模型进行了改进, 并与协同过滤模型进行结合, 最后使用增量梯度下降算法解决了局部最优问题, 在数据处理与推荐准确性方面有了较大提升.

### 3 实验和分析

#### 3.1 评估计划

在数据集中随机选择与每个用户相关的  $P$  项以形成训练集 [18], 并将所有剩余的数据集用作测试集. 为了在稀疏和密集设置下评估和比较模型, 本文在实验中分别将  $P$  设置为 1 和 10. 对于  $P$  的每个值, 用不同的随机选择训练集重复评估五次, 并记录平均性能. 本文使用查全率作为性能度量, 由于评级信息采用隐式反馈的形式, 零条目可能是用户对该项目不感兴趣, 或者用户不知道其存在. 因此, 精度不是一个合适的性能指标. 像大多数推荐系统一样, 本文对候选项目的预测评级进行了排序 [19], 并向目标用户推荐前  $M$  项目. 每个用户的查全率 @  $M$  被定义为:

$$@M = \frac{\text{在 } M \text{ 中用户喜欢的项目数}}{\text{用户喜欢的项目总数}}$$

最终结果是所有用户的平均查全率.

#### 3.2 基线和实验设置

本文中协同深度学习算法 (CIDL) 同时与 CMF 算法 [20]、SVDFeature 算法 [21]、CTR 算法 [8] 以及 DeepMusic [22] 算法应用中的各项指标进行比较. 在实验中, 本文首先使用一个验证集来寻找 CMF、SVDFeature, CTR 以及 DeepMusic 的最佳超参数. CMF 算法中将不同上下文的潜在因子的正则化超参数设置为 10. 在网络搜索之后, CMF 在等级矩阵和内容矩阵的权重均为稀疏度为 5. 密度设置中的权重分别为 8 和 2. SVDFeature 算法中, 用户和项目的正则化超参数为 0.004 且学习率等于 0.005. DeepMusic 算法中, 使用具有两个卷积层的 CNN. CTR 算法中, 使用的参数为  $\lambda_u = 0.1, \lambda_v = 10, a = 1, b = 0.01, K = 50$ .

#### 3.3 平均查全率分析

图 4 和图 5 显示了在稀疏 ( $P = 1$ ) 和密集 ( $P = 10$ ) 设置下使用电影数据集比较 CIDL 算法, CTR 算法, DeepMusic 算法, CMF 算法以及 SVDFeature 算法的结果. 由图像可得, 无论在数据集密度稀疏还是密集, CIDL 算法都优于其他四种算法. 在数据密集的情况下虽然 DeepMusic 算法具有优秀的体系结构, 但 CTR 算

法在数据处理中更为优秀. 在稀疏设置中, CMF 算法大部分时间都比 SVDFeature 算法优秀, 有时可以达到与 CTR 算法相当的性能. DeepMusic 算法由于缺乏评分而表现不佳. 在密集的环境中, SVDFeature 算法对于 Netflix 数据来说, 效果不如 CMF. 在稀疏设置中 2 层 CIDL 的性能比 CTR 高出 1.9%, 在密集设置下高出 1.5%–2.0%. 由实验可以得出, 深度学习和推荐系统的整合算法是成功的, 对于查全率有显著的提升.

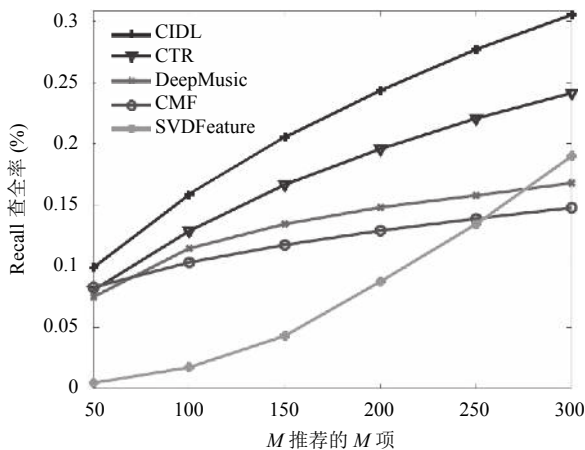


图4 各算法在  $P=1$  时的对比图

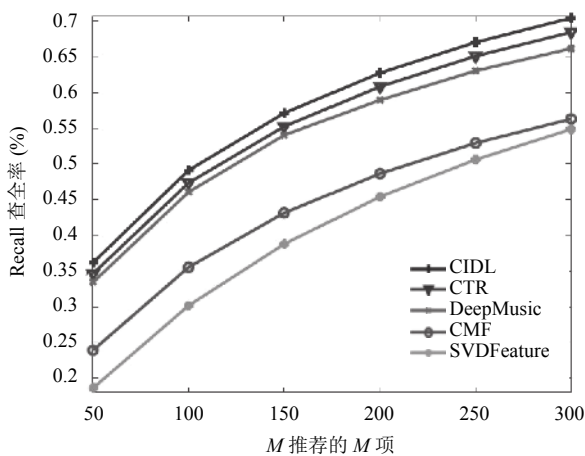


图5 各算法在  $P=10$  时的对比图

在稠密环境下使用不同  $\Delta n$  也有不同结果, 如图 6 所示.

当  $\lambda_n$  趋近于无限大时,  $\lambda_n/\lambda_v$  将会趋近正无穷, 此时 CIDL 将退化为两个单独的模型. 在这种情况下, SDAE 将以无监督方式学习潜在项目表示, 然后将其直接放入 CTR 的简化版本中. 这种情况下, 贝叶斯 SDAE 和基于矩阵分解的协同过滤组件之间没有交互作用,

预测性能将大大提高. 对于另一个极端, 当  $\lambda_n$  无限小时,  $\lambda_n/\lambda_v$  将趋近于近零, 此时 CIDL 退化为贝叶斯 SDAE 分量解码器基本消失的情况. 此时贝叶斯 SDAE 组件基本消失, 编码器将通过简单矩阵分解学习潜在项目向量. 如图 6 所示, 预测性能随着  $\lambda_n$  变化而变化. 当  $\lambda_n < 0.1$  时, 查全率@  $M$  已经非常接近 PMF 的结果 (或甚至比 PMF 结果还要差).

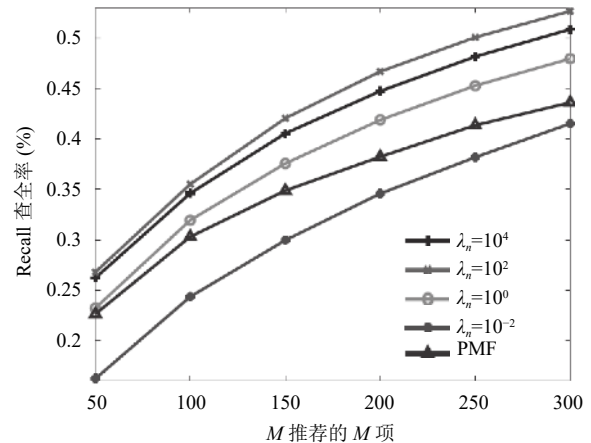


图6 在不同  $\Delta n$  的情况下各算法的比较

综合以上实验结果, 在大量数据的实验证明下可知, 不论在数据密集或是数据稀疏的情况下 CIDL 算法都优于 CMF 算法、SVDFeature 算法、DeepMusic 算法、CTR 算法以及 DeepMusic 算法. 而在 CIDL 算法中算法的预测性能会随着  $\lambda_n$  的变化而变化, 因此在 CIDL 算法中只有通过实验找到最合适的  $\lambda_n$  值, 才能更好的提高算法效率.

#### 4 结束语

本文中展示了通过联合对内容信息的深度学习和对评级 (反馈) 矩阵进行协作性过滤从而实现出色的性能的算法. CIDL 算法是一个用于弥合最先进的深度学习模型和推荐系统之间差距的等级贝叶斯模型. 在深度学习方面, 除了获得 MAP 估计的算法外, 本文还推导了一种基于采样的 CIDL 贝叶斯处理算法, 作为反向传播的贝叶斯广义版本.

#### 参考文献

1 Sarwar B, Karypis C, Konstan J, et al. Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on World

- Wide Web. Hong Kong, China. 2014. 285–295.
- 2 丁雪涛. 基于协同关系主题回归模型的推荐算法研究[硕士学位论文]. 北京: 清华大学, 2013.
  - 3 Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, MD, USA. 2014. 655–665.
  - 4 Vincent P, Larochelle H, Lajoie I, *et al.* Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research, 2010, 11: 3371–3408.
  - 5 孟祥武, 刘树栋, 张玉洁, 等. 社会化推荐系统研究. 软件学报, 2015, 26(6): 1356–1372. [doi: [10.13328/j.cnki.jos.004831](https://doi.org/10.13328/j.cnki.jos.004831)]
  - 6 Bobadilla J, Ortega F, Hernando A, *et al.* Recommender systems survey. Knowledge-Based Systems, 2013, 46: 109–132. [doi: [10.1016/j.knosys.2013.03.012](https://doi.org/10.1016/j.knosys.2013.03.012)]
  - 7 Sevil SG, Kucuktunc O, Duygulu P, *et al.* Automatic tag expansion using visual similarity for photo sharing websites. Multimedia Tools and Applications, 2010, 49(1): 81–99. [doi: [10.1007/s11042-009-0394-5](https://doi.org/10.1007/s11042-009-0394-5)]
  - 8 Wang H, Li WJ. Relational collaborative topic regression for recommender systems. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(5): 1343–1355. [doi: [10.1109/TKDE.2014.2365789](https://doi.org/10.1109/TKDE.2014.2365789)]
  - 9 杨帅, 王鹃. 基于堆栈降噪自编码器改进的混合推荐算法. 计算机应用, 2018, 38(7): 1866–1871.
  - 10 王妍, 唐杰. 基于深度学习的论文个性化推荐算法. 中文信息学报, 2018, 32(4): 114–119. [doi: [10.3969/j.issn.1003-0077.2018.04.014](https://doi.org/10.3969/j.issn.1003-0077.2018.04.014)]
  - 11 武玲梅, 陆建波, 刘春霞. 基于降噪自动编码器的推荐算法. 计算机与现代化, 2018, (3): 78–82. [doi: [10.3969/j.issn.1006-2475.2018.03.015](https://doi.org/10.3969/j.issn.1006-2475.2018.03.015)]
  - 12 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](https://doi.org/10.3969/j.issn.1000-1220.2018.01.003)]
  - 13 王宪保, 何文秀, 王辛刚, 等. 基于堆栈降噪自动编码器的胶囊缺陷检测方法. 计算机科学, 2016, 43(2): 64–67.
  - 14 Zhou K, Zha HY. Learning binary codes for collaborative filtering. Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Beijing, China. 2012. 498–506.
  - 15 许普乐, 王杨, 黄亚坤, 等. 大数据环境下基于贝叶斯推理的中文地名地址匹配方法. 计算机科学, 2017, 44(9): 266–271.
  - 16 余冲. 基于深度学习的协同过滤模型研究[硕士学位论文]. 深圳: 深圳大学, 2017.
  - 17 Zhang W, Zhuang JY, Yong X, *et al.* Personalized topic modeling for recommending user-generated content. Frontiers of Information Technology & Electronic Engineering, 2017, 18(5): 708–718.
  - 18 Guo YW, Cheng HR, Tang MS, *et al.* Kernel based collaborative topic regression for tag recommendation. Proceedings of the 2016 International Conference on Education, Sports, Arts and Management Engineering. 2016. 113–117.
  - 19 梁淑芬, 刘银华, 李立琛. 基于LBP和深度学习的非限制条件下人脸识别算法. 通信学报, 2014, 35(6): 154–160. [doi: [10.3969/j.issn.1000-436x.2014.06.020](https://doi.org/10.3969/j.issn.1000-436x.2014.06.020)]
  - 20 Singh AP, Gordon GJ. Relational learning via collective matrix factorization. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Las Vegas, NV, USA. 2008. 650–658.
  - 21 Chen TQ, Zhang WN, Lu QX, *et al.* SVDFeature: A toolkit for feature-based collaborative filtering. The Journal of Machine Learning Research, 2014, 13: 3619–3622.
  - 22 van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe, NV, USA. 2013. 2643–2651.