

DOI:10.16356/j.1005-2615.2019.05.003

基于深度混合模型评分推荐

钱付兰^{1,2} 李建红^{1,2} 赵 姝^{1,2} 张燕平^{1,2}

(1. 安徽大学计算机科学与技术学院, 合肥, 230601; 2. 安徽大学信息保障技术研究中心, 合肥, 230601)

摘要: 从用户-项目评分矩阵中学习用户对项目的个性化偏好, 对于评分推荐来说至关重要。许多推荐方法如潜在因子模型, 无法充分利用评分矩阵中的交互信息学到较好的个性化偏好而得到较差推荐效果。受深度学习中 Wide and Deep 模型应用于 APP 推荐启发, 本文提出一种深度混合模型并命名为 DeepHM 用于评分推荐。与 Wide and Deep 模型相比, 使用 DeepWide 和 DNN 部分重构 Wide 模型和 Deep 模型得到 DeepHM, 并且 DeepWide 和 DNN 部分共享交互信息输入。因此, DeepHM 可以更有效地使用评分矩阵中的用户和项目的交互信息学到个性化偏好信息。DeepHM 将评分推荐作为分类问题旨在提高推荐准确性。实验表明在公开的 Movielens 数据集上 DeepHM 算法相比现有的基于评分推荐模型具有更好的效果。

关键词: 深度学习; 推荐算法; 评分推荐

中图分类号: TP391 **文献标志码:** A **文章编号:** 1005-2615(2019)05-0592-07

Rating Recommendation Based on Deep Hybrid Model

QIAN Fulan^{1,2}, LI Jianhong^{1,2}, ZHAO Shu^{1,2}, ZHANG Yanping^{1,2}

(1. School of Computer Science and Technology, Anhui University, Hefei, 230601, China; 2. Information Support & Assurance Technology, Anhui University, Hefei, 230601, China)

Abstract: Learning individual preferences of users for items from user-item rating matrix is critical for rating recommendation. Many recommendation methods, such as the Latent Factor model, can not make full use of the interaction information from rating matrix to learning individual preferences, and achieve unsatisfying results. Inspired by wide and deep learning model of deep learning in APP recommendation, deep hybrid model is proposed and named DeepHM for rating recommendation. Compared with the wide and deep model, deep wide model and DNN model are used to reconstruct wide model and deep model, which can get DeepHM and make DeepHM become shared input to its deep wide and deep parts. Therefore, DeepHM uses interaction information of user and item from rating matrix more efficiently to obtain individual preferences information. Furthermore, DeepHM treats the rating recommendation as a multi-classification problem aiming to improve the accuracy of recommendation. Through comprehensive experiments on public Movielens datasets, it demonstrates that the efficiency of DeepHM based on rating recommendation is better than that of the existing models.

Key words: deep learning; recommendation algorithm; rating recommendation

近年来, 随着大量的互联网应用使得数据量越来越大。这些数据包巨大的商业价值但数据规模较大且冗余。如何从海量数据中获取有价值的信息对研究人员来说是一项挑战。作为信息过

基金项目: 国家重点研究和开发(2017YFB1401903)资助项目; 国家自然科学基金(61673020, 61702003, 61876001)资助项目; 安徽省自然科学基金(1808085MF175)资助项目。

收稿日期: 2019-07-10; **修订日期:** 2019-09-10

通信作者: 张燕平, 女, 博士生导师, 教授, E-mail: zhangyp2@gmail.com。

引用格式: 钱付兰, 李建红, 赵姝, 等. 基于深度混合模型评分推荐[J]. 南京航空航天大学学报, 2019, 51(5): 592-598.
QIAN Fulan, LI Jianhong, ZHAO Shu, et al. Rating Recommendation Based on Deep Hybrid Model[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2019, 51(5): 592-598.

滤的有效方法,推荐系统在个性化服务中发挥着重要作用,其核心技术是推荐算法,它决定推荐系统的质量和性能。协同过滤算法^[1]是推荐系统中使用最广泛的,思想是利用用户和项目交互信息来获取用户偏好,以获得准确的个性化推荐。该方法主要包括基于用户和基于项目两种类型。基于用户是计算用户间的相似性;基于项目的思想是计算所有历史用户偏好项目之间的相似性,然后推荐那些用户可能喜欢的项目。与项目总数相比,项目被用户的评分数据非常小,预测的准确性会受到影响。随着数据量的增长,获得个性化偏好的难度也增加。因此许多研究人员和学者提出了基于协同过滤算法的潜在因子模型等算法。例如,矩阵分解算法^[2],它通过将用户和项目交互矩阵分解为两个低维矩阵的乘积以便获得用户偏好信息。Tyco方法^[3]是在用户组的用户度中找到用户的“邻居”。其他方法还包括贝叶斯概率矩阵分解^[4]和SVD⁺⁺算法^[5]等等。这些方法虽然有效,但对推荐性能提升都不是很明显。而且数据的稀疏性限制了算法的性能。如何提升推荐效果成为推荐算法的核心问题。

近年来,深度学习^[6]证明了能够从数据中学习,从而获得较好信息特征表示。因此,许多研究人员尝试深度学习方法用于推荐。例如将受限玻尔兹曼机^[7]应用于个性化推荐当中,其做法是将项目类别信息集成为条件层,旨在优化模型参数。此外,自动编码器^[8]也被广泛用于个性化推荐中。这种方法的核心思想是将用户项矩阵输入到自动编码器并学习权重,编码器和解码器应用于输入以在学习权重时获得输出。类似方法包括基于自动编码器的协同过滤^[9]等。文献[10]提出了基于二次多项式的深度神经网络的推荐算法,其特点是通过二次多项式去学习数据中潜在的信息特征而非使用传统的矩阵分解方法,然后将这些信息特征放入深度神经网络中做输入学习输出预测评分。随着深度学习在推荐系统方面不断应用,针对多源异构数据,文献[11]设计了 Wide and Deep 模型,它不仅具有深度神经网络的特征表示能力,还有线性结构的记忆能力,但需要构建特征工程。由于该模型具有较好的推荐效果,因此在推荐中得到广泛应用。本质上,这些基于深度学习的方法都是利用深度学习强大的表示学习能力来分析评分信息以获得较好的推荐,当数据稀疏时却得不到令人满意的结果。故文献[12]提出了通过深度学习来分析附加信息(例如文本信息)的方法,从而达到提升推荐效果目的。虽然能有好的效果,但这些附加的信息往往并不容易获取。而且这些算法将评分问题作为

实数问题解决并不准确。例如,电影的实际评分为 3,模型的预测结果为 2.5 分,但 2.5 不能代表 3 或 2。由于 Wide and Deep 模型不能共享输入并且特征工程的构建还需要专业知识^[13]。这揭示了 Wide and Deep 模型构造是比较复杂的。

为了解决这些问题,提出了一种名为深度混合模型(Deep hybrid model, DeepHM)的新型深度学习框架,该模型扩展了 Wide and Deep 模型,使得模型在共享输入的同时避免了构建复杂的特征工程。其次,DeepHM 只对评分矩阵信息进行学习而不需要添加额外的信息。实验表明,相比较其他的评分推荐算法,DeepHM 算法可以取得较好的效果。

1 DeepHM 模型

1.1 Wide and Deep 模型

Cheng^[11]等通过分析用户的数据发现数据中存在一些关联,而这种关联关系被定义为记忆能力。因此他们将这个记忆能力与深度神经网络的特征表示能力相结合提出了 Wide and Deep 模型,其框架如图 1 所示。

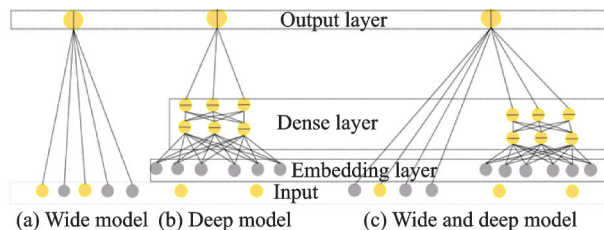


图 1 Wide and Deep 模型框架

Fig.1 Wide and Deep framework

图 1(a)是由线性结构组成的 Wide 模型,部分输入信息构建特征工程,通过线性结构一起训练。目的是获取数据中的记忆信息。式(1)和式(2)给出了计算方法为

$$y = \sigma(\mathbf{w}^T \mathbf{x} + b) \quad (1)$$

$$\varphi(\mathbf{x}) = \prod_{i=1}^d x_i^{c_{ki}}, c_{ki} \in \{0, 1\} \quad (2)$$

式中: \mathbf{x} 表示带有 d 维向量的输入, \mathbf{w} 是模型的权重参数, b 表示偏移量。 $\varphi(\mathbf{x})$ 是特征工程, σ 代表激活函数, y 是最终输出。 c_{ki} 表示布尔变量,如果有特征工程,则值为 1,否则值为 0。

图 1(b)是由深度神经网络组成的 Deep 模型。深度模型通过表示学习方式特征提取,计算过程为

$$a^{(l+1)} = f(a^{(l)}\mathbf{w}^{(l)} + b^{(l)}) \quad (3)$$

式中: \mathbf{w} , b 是激活函数层的权重和偏差, f 是激活函数。

图 1(c)是由前面 Wide 模型和 Deep 模型构建

成的Wide and Deep模型。计算过程为

$$P(Y=1|x) = \sigma(\mathbf{w}_{\text{wide}}^T [\mathbf{x}, \phi(\mathbf{x})] + \mathbf{w}_{\text{deep}}^T \mathbf{a}^{(l)} + b) \quad (4)$$

式中: \mathbf{x} 代表输入向量, y 是二分类标签,它根据模型预测的结果确认 x 是否被推荐。 $\mathbf{w}_{\text{wide}}, \mathbf{w}_{\text{deep}}$ 分别表示Wide模型权重和Deep模型权重, b 为偏差。

基于上面描述可知Wide and Deep模型旨在解决实数推荐,即根据预测值决定是否推荐。如果评分数据是整数,根据实数来处理整数的评分问题不够准确。此外,如图1所示Wide模型和Deep模型不共享输入,无法挖掘到较好的个性化用户偏好,式(2)显示了构造特征工程的复杂性,为此提出DeepHM来解决这些问题。

1.2 DeepHM模型

通常,许多评分推荐算法将评分根据实数结果

进行推荐,这对推荐的效果具有一定的影响。例如,电影的实际评分是4,模型的预测结果是3.5,但3.5不能代表3或4。本文为了提高推荐精度将评分推荐问题转换为多分类问题。式(5)给出了评分问题转换为多分类问题的方式,有

$$P(Y=1|2|3|\dots) = \sigma(\mathbf{x}) \quad (5)$$

式中: Y 是输出, \mathbf{x} 表示输入。

模型的框架如图2所示。通过图2可以看到,输入信息在送入DeepHM模型之前进行预处理。其方式是通过embedded层将输入信息映射到向量空间,因此输入信息转换为向量表示并作为DeepHM的输入。embedded层处理过程为

$$\begin{cases} \mathbf{x}_{\text{user}} = \text{embedded}(\mathbf{x}_{\text{user_input}}) \\ \mathbf{x}_{\text{item}} = \text{embedded}(\mathbf{x}_{\text{item_input}}) \end{cases} \quad (6)$$

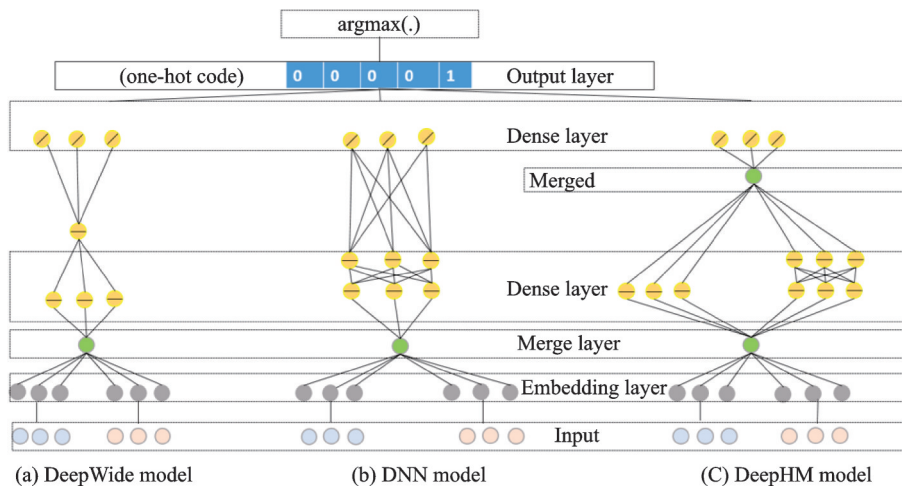


图2 DeepHM模型框架

Fig.2 DeepHM model framework

式中: $\mathbf{x}_{\text{user_input}}, \mathbf{x}_{\text{item_input}}$ 表示用户信息和主题信息原始输入,embedded表示嵌入函数, $\mathbf{x}_{\text{user}}, \mathbf{x}_{\text{item}}$ 分别表示embedded层处理之后的用户和主题嵌入输出。

通过merge层连接用户和项目信息表示表示用户与项目交互信息矩阵。merge层的处理过程为

$$\mathbf{x} = \text{concat}(\mathbf{x}_{\text{user}}, \mathbf{x}_{\text{item}}) \quad (7)$$

式中: $\mathbf{x}_{\text{user}}, \mathbf{x}_{\text{item}}, \mathbf{x}$ 分别代表用户信息、项目信息和两者合并后信息,concat表示拼接函数。现在可以在这些合并信息上训练模型。

对于输入的评分值将其转换为one_hot编码进行预处理,其做法是将评分值转换为对应标签值,那么评分问题则变为分类问题。假如一个评分值是3(5分制),那么对应的one_hot编码为00100。该过程为

$$y = \text{one_hot}(\text{rating}) \quad (8)$$

式中: y 是标签, rating表示输入的整数值,one_hot是整数转换为标签值函数。

DeepWide模型由Embedding层,Dense层和线性结构组成。与Wide模型不同的是,DeepWide模型中的所有输入信息通过merge层处理后,还通过一个Dense层的深度神经网络对信息进行学习,其目的是获得输入信息的特征表示作为Wide模型的输入。其次用Wide结构尝试从特征表示中挖掘其中存在用户偏好信息,最后利用反向传播更新对应的参数。该过程为

$$\mathbf{x}_{\text{wide}} = f_{\text{wide}}(\mathbf{w}_{\text{wide}} \mathbf{x} + b_{\text{wide}}) \quad (9)$$

$$\mathbf{w}_{\text{wide}} = \mathbf{w}_{\text{wide}} - \eta \frac{\partial J}{\partial \mathbf{w}_{\text{end}}} \frac{\partial \mathbf{w}_{\text{end}}}{\partial \mathbf{w}_{\text{wide}}} \quad (10)$$

式中: \mathbf{x}_{wide} 表示该DeepWide模型的输出; $\mathbf{x}, \mathbf{w}_{\text{wide}}, b_{\text{wide}}$ 分别表示DeepWide模型的输入;权重和偏差; η 是学习率, J 表示交叉熵损失; f_{wide} 代表激

活函数。

Deep neural network (DNN) 模型如图 2(b) 所示。它类似于图 1(b) 中的 Deep 模型, 中间全部由 Dense 层组成。输入数据被向量化且合并后, 将被送入到 DNN 模型中进行训练, 最后得到一个输出结果。处理过程如式(11—15)所示。

$$x_{l+1, \text{deep}, i} = f_{l+1, \text{deep}, i}(\mathbf{w}_{l, \text{deep}, i}^T x_{l, \text{deep}, i} + b_{l, \text{deep}, i}) \quad (11)$$

$$\text{input}_{l, \text{deep}, i} = b_{l, \text{deep}, i} + \sum_j \mathbf{w}_{l, \text{deep}, j} x_{l-1, j} \quad (12)$$

$$\frac{\partial J}{\partial \mathbf{w}_{l, \text{deep}, i}} = \frac{\partial J}{\partial \mathbf{w}_{\text{end}}} \frac{\partial \mathbf{w}_{\text{end}}}{\partial \text{input}_{l, \text{deep}, i}} \frac{\partial \text{input}_{l, \text{deep}, i}}{\partial \mathbf{w}_{l, \text{deep}, i}} \quad (13)$$

$$\mathbf{w}_{l, \text{deep}, i} = \mathbf{w}_{l, \text{deep}, i} - \eta \frac{\partial J}{\partial \mathbf{w}_{l, \text{deep}, i}} \quad (14)$$

$$\frac{\partial J}{\partial \text{input}_{l, \text{deep}, i}} = \sum_k \frac{\partial J}{\partial \mathbf{w}_{\text{end}}} \frac{\partial \text{end}}{\partial \text{input}_{l+1, \text{deep}, k}} \mathbf{w}_{l+1, ki} \quad (15)$$

式中: $x_{l, \text{deep}, i}$ 表示 DNN 模型第 l 层隐藏层的第 i 个输入, 第 1 层输入为 x , $\mathbf{w}_{l, \text{deep}, i}$ 是 DNN 模型表示 DNN 模型第 l 层隐藏层的第 i 个权重, $b_{l, \text{deep}, i}$ 表示 DNN 模型第 l 层的隐藏层第 i 个偏差。 $f_{l, \text{deep}, i}$ 表示第 $i+1$ 个隐藏层的激活函数。 $\text{input}_{l, \text{deep}, i}$ 表示 DNN 模型第 l 层隐藏层的第 i 个结果输入。

DeepHM 模型通过合并 DeepWide 和 DNN 模型所构成。如图 2(c) 所示, 该模型不仅具有 Wide and Deep 模型的优点, 并且共享输入同时也避免了复杂的特征工程构建。在输出之前添加 Dense 层的原因是输出节点的数量将由最后一个 Dense 层控制以达到分类的目的。更新模型中参数的过程为

$$\mathbf{x}_{\text{end}} = \text{concat}(\mathbf{x}_{\text{wide}}, \mathbf{x}_{\text{deep}}) \quad (16)$$

$$\hat{y} = f_{\text{end}}(\mathbf{w}_{\text{end}} \mathbf{x}_{\text{end}} + b_{\text{end}}) \quad (17)$$

$$\frac{\partial J}{\partial \mathbf{x}_{\text{end}}} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \mathbf{x}_{\text{end}}} \quad (18)$$

$$\frac{\partial J}{\partial \hat{y}_i} = \frac{\hat{y}_i - y_i}{\hat{y}_i(1 - \hat{y}_i)} \quad (19)$$

$$\frac{\partial J}{\partial \mathbf{w}_{\text{end}}} = \hat{y}_i - y_i \quad (20)$$

式中: \hat{y}_i 表示预测评分值, y_i 表示实际评分值, $f_{\text{end}}, \mathbf{w}_{\text{end}}, \mathbf{x}_{\text{end}}, b_{\text{end}}$ 分别表示激活函数、权重、输入和偏差。

训练损失函数是交叉熵函数, 其目的是计算实际输出值和实际值之间的误差。计算方式为

$$J = -\frac{1}{n} \sum_{i=1}^n (y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)) \quad (21)$$

式中: J 表示交叉熵损失; n 表示总数; y_i, \hat{y}_i 分别表示用户实际评分和预测评分。

与 Wide and Deep 模型直接输出结果不同的是, 在式(8)中, 评分值变成了 one_hot 编码的标签

值。故 DeepHM 中的输出评分值的形式是 one_hot 编码, 而不是直接输出评分。因此需要将其转换为整数值, 其表达式为

$$r_i = \text{argmax}(\hat{y}_i) \quad (22)$$

式中: \hat{y}_i 代表第 i 用户的以 one_hot 编码形式的预测评分, r_i 为 \hat{y}_i 整数评分值。argmax 将 one_hot 编码转换为整数函数。例如, 如果模型预测的评分 one_hot 编码形式如式(23)表示。整数评分则为 5, 3, ..., 5。

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ \vdots \\ 5 \end{bmatrix} \quad (23)$$

2 模型训练和实验结果分析

2.1 数据集

本文实验中使用数据集为公开数据集 Movielens 100 K 和 Movielens 1 M。Movielens 100 K 数据集包含 100 000 个评分记录, 一共有 943 个用户对 1 682 个项目的评分, 所有评分值均为整数且不大于 5。每个用户评分至少评价 20 部电影, 评分越高表示用户越喜欢这部电影。Movielens 1 M 数据集包含来自 6 040 位用户对 3 952 部电影的 1 000 209 个评分记录, 每位用户至少评价 20 部电影, 详细信息显示在表 1 中。根据表 1 可知数据集越大, 数据越稀疏。

表 1 Movielens 数据描述

Tab.1 Description of Movielens datasets

Features	100 K	1 M
Users	943	6 040
Items	1 682	3 952
Ratings	100 000	1 000 209
Ratings per user	106.4	165.6
Rating per item	59.5	253.09
Rating sparsity/%	93.7	95.8

2.2 评价指标与对比算法

平均绝对误差 (Mean absolute error, MAE) 是推荐系统中被广泛使用的评价标准, 因此使用 MAE 来评估 DeepHM 模型的性能。其计算表达式为

$$\text{MAE} = \frac{1}{T} \sum_{i=1}^T |P_i - R_i| \quad (24)$$

式中: T 表示测试集的大小, P_i 是第 i 个用户的预测评分, R_i 表示第 i 个用户的实际评分。

实验对比算法包括: 矩阵分解 (Matrix factorization, MF), 基于项目的协同过滤 (Item-based collaborative filtering, Item-Based), 贝叶斯概率矩

阵分解算法(Bayes probabilistic matrix factorization, BPMF), SVD算法的改进版SVD⁺⁺, 在用户组基于用户度中找到用户邻居的“邻居”算法(Typicality-based collaborative filtering, TyCo), 自构造聚类算法(Self-constructing clustering algorithm, SCC)^[14], 近端支持向量机算法(Proximal maximum margin matrix factorization, P MMMF)^[15], 序列回归矩阵分解算法(Double matrix factorization, DMF)^[16], 通过对偶自动编码器的表示学习方法(Representation learning via dual-autoencoder, ReDa)^[17], 基于二次多项式深度神经网络的推荐算法(Quadratic deep neural networks, Q-DNN), Wide and Deep 算法, 学习用户依赖关系的算法(Learning user dependencies for recommendation, LDUR)^[18](此算法是基于序列, 故归类为实数推荐), 基于Bias SVD组迭代的声誉排序算法(Iterative group-based ranking singular value decomposition, IGRS)^[19], 协同过滤的分层自动编码器算法(Hierarchical autoencoder for collaborative filtering, HACF), 协同减噪自动编码器(Collaborative denoising auto-encoders, CDAE), 基于推荐技术的隐式社交关系提取方法(Hell trust singular value decomposition, HellTrustSVD)^[20], 局部低秩矩阵逼近个性化选择算法(Local low-rank matrix approximation, LLRMA)^[21], 基于主题条件受限玻尔兹曼机算法(Hybrid item category aware conditional restricted boltzmann machine based recommendation, Hybrid IC-CRBMF)^[22]和基于矩阵填充算法(Deep conditional matrix completion, DC-MC)^[23]。

2.3 模型训练参数

DeepHM 和对比算法实验在 Movielens 100 K 和 Movielens 1 M 上进行。只使用用户 ID, 电影 ID 和评分的信息, 不会添加任何其他信息。实验参数设置使用 Adam 优化函数, Dense 层激活函数为 relu 激活函数, 最后一层使用 softmax 函数输出评分值, 程序语言是 python。DeepWide 模型为两层全连接层, 第 1 层有 128 个节点, 第 2 层只有 1 个节点; DNN 模型有 3 层全连接层, 每层均有 128 个节点。整个模型基于 Keras 构建, 后端使用 TensorFlow。为了减少运行时间, 使用 GPU 为 NVIDIA Tesla P100 来加速实验运行。实验进行 5 次, 每次随机抽取数据集中的 20% 数据作为测试集, 80% 数据作为训练集, 将 5 次结果的平均值作为实验的结果。

2.4 实验结果分析

实验由两部分组成。一部分正如表 2 中显示的基于实数推荐的算法进行比较。表 3 是与评分推荐中基于分类的算法对比, 黑体字表示结果最

佳。由于特征工程构造比较复杂, 需要用到专业的知识, 因而实现 Wide and Deep 模型较为困难, 且缺乏专业知识的合理性。使用共享输入和共享优化适当修改了 Wide and Deep 模型并命名为 Wide and Deep-I, 而用于多分类任务则命名为 Wide and Deep-II。

根据表 2 中 Movielens 100 K 的实验结果, DeepHM 模型在 MAE 指标上数值为 0.683, 相比其他算法至少提升 1.7%, 最多能提升 5.9%。而 MF, SVD⁺⁺, CDAE 和 CRBM 等算法的 MAE 值均高于 0.7。结合表 1 数据分析后, 认为 100 k 数据集相对来说较小而且数据还有一定的稀疏性, 导致用户和项目之间的交互信息较少。因此 MF, SVD⁺⁺ 和 Wide and Deep-I 等算法都无法获得有效的用户偏好信息致使算法推荐效果不佳。DC-MC 算法可以取得较好的结果是因为通过几何矩阵填充结合深度学习方法从而得到较好的结果。通过表 2 中 Movielens 1 M 的实验结果发现, DeepHM 模型在 MAE 指标上数值为 0.653, 相比其他算法最多提高了 4.7%, 最少提升了 0.7%。MF, SVD⁺⁺ 等对比算法中的 MAE 值也均未超过 0.7, 这表明数据集越大, 算法可以学习更多用户偏好信息从而使预测评分更准确。这就是 Movielens 1M 的实验结果优于 Movielens 100 K 的原因。另外根据表 1 对数据集的描述可知, Movielens 1 M 稀疏性要高于 100 K, 对算法推荐精度有一定的影响, 因此推荐精度提升不够明显。相比之下, DeepHM 的效果比其他方法有明显的提升。

从表 3 结果可以清楚地看到, DeepHM 模型与其他基于分类算法在 100 K 上提升最多 4.2%, 最少提升 1.1%, 平均提升 2.6%。在 Movielens 1 M 上, DeepHM 模型最多提升 2.8%, 最少提升 0.4%, 平均提升 1.6%。虽然 Wide 模型的结果不如其他深度模型好, 但它仍然优于表 3 中 MF, SVD⁺⁺ 等算法。此外, 提出的 DeepWide 模型, 它不仅有 Wide 特性, 同时还有表示学习特性。这增强了挖掘用户与项目交互信息的能力。从结果来看, DeepWide 效果比 Wide 模型要好, 而且也比 DNN 模型好。

结合表 2 和表 3 的数据不难发现, 表 3 的实验结果基本好于表 2, 所以基于分类评分推荐的方法要优于实数推荐, 而且 DeepHM 模型结果优于 Wide and Deep-I 和 Wide and Deep-II 模型。这也证实了不使用特征工程 DeepHM 在评分推荐上也能取得较好的效果。

通过以上实验可以得出结论, DeepHM 通过挖掘隐式交互信息来获得用户偏好, 并且将评分推荐

表 2 基于实数的评分推荐算法实验结果 MAE 值对比

Tab.2 Experimental results of MAE with different rating recommendation algorithms based on real number

Algorithm	100 K	1 M
MF	0.732	0.700
SVD ⁺⁺	0.739	0.698
IGRS	0.718	0.699
SCC	0.733	0.692
BPMF	0.742	0.690
DMF	0.712	0.679
PMMMf	0.723	0.677
TyCo	0.731	0.665
Item-Based	0.724	0.700
LUDR	0.721	0.673
DCMC	0.696	—
ReDa	0.720	0.665
HACF	0.717	0.678
CDAE	0.735	0.691
HellTrustSVD	0.716	0.665
LLRMA	0.700	0.660
Hybird IC-CRBMF	0.719	0.681
Wide and Deep-I	0.723	0.671
DeepHM	0.683	0.653

表 3 基于分类的评分推荐算法实验结果 MAE 值对比

Tab.3 Experimental results of MAE with different rating recommendation algorithms based on classification

Algorithm	100 K	1 M
Wide	0.725	0.681
DeepWide	0.694	0.657
DNN	0.715	0.662
Wide and Deep-II	0.706	0.669
Q-DNN	0.696	0.659
DeepHM	0.683	0.653

问题转换为分类问题后与其他传统的机器学习方法和深度学习算法相比,DeepHM具有更好的推荐结果。

3 结 论

本文提出了一种基于 Wide and Deep 模型的新型神经网络框架深度混合模型 DeepHM 并应用于评分推荐中。它可以充分利用用户-项目信息矩阵来获取用户个性化偏好信息,即使数据稀疏也具有较好的推荐能力。此外,将评分推荐问题转换为分类问题后使得推荐更加准确。在接下来的工作中将在 DeepHM 的基础上添加其他功能使其推荐效果更好。

参考文献:

- [1] SARWAR B, KARYPIS G, KONSTAN J, et al. Item - based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th International Conference on World Wide Web. Hong Kong :ACM, 2001: 285-295.
- [2] KOREN Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model[C]// Proceedings of the 14th ACM SIGKDD International Conference of Knowledge Discovery and Data Mining. Las Vegas:ACM, 2008: 426-434.
- [3] CAI Yi, LEUNG H H, LI Qing, et al. Typicality - based collaborative filtering recommendation[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(3): 766-779.
- [4] SALAKHUTDINOV R, MNIH A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo[C]//International Conference on Machine Learning. New York:ACM, 2008: 880-887.
- [5] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [6] BENGIO Y, GOODFELLOW I J, COURVILLE A. Deep learning [J]. Nature, 2015, 521(7553): 436 - 444.
- [7] SALAKHUTDINOV R, MNIH A, HINTON G. Restricted Boltzmann machines for collaborative filtering[C]//Proceedings of the 24th International Conference on Machine Learning. Corvalis: ACM, 2007: 791-798.
- [8] WU Yao, DUBOIS C, ESTER M, et al. Collaborative denoising auto - encoders for Top - N recommender systems[C]//International Conference on Web Search and Data Mining. San Francisco: ACM, 2016: 153-162.
- [9] SHUBHAM M, ANGSUL M. Hierarchical autoencoder for collaborative filtering[C]// Proceeding of International Joint Conference on Neural Networks. Rio :IEEE, 2018: 1-7.
- [10] ZHANG Libo, LUO Tiejian, ZHANG Fei, et al. A recommendation model based on deep Neural network [J]. IEEE Access, 2018, 6: 9454-9463.
- [11] CHENG H, KOC L, HARMSSEN J, et al. Wide & deep learning for recommender systems[C]// Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. Boston: ACM, 2016: 7 - 10.
- [12] DONG Xin, YU Lei, WU Zhonghuo, et al. A hybrid collaborative filtering model with deep structure for

- recommender systems[C]// Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. San Francisco: AAAI Press, 2017: 1309-1315.
- [13] GUO Huifeng, TANG Ruming, YE Yunming, et al. DeepFM: A factorization - machine based neural network for CTR prediction[C]//Proceedings of the 26th International Joint Conference on Artificial Intelligence. Melbourne: AAAI Press, 2017: 1725-1731.
- [14] LIAO C L, LEE S J. A clustering based approach to improving the efficiency of collaborative filtering recommendation [J]. Electronic Commerce Research & Applications, 2016, 18: 1-9.
- [15] KUMAR V, PUJARI A K, SAHU S K, et al. Proximal maximum margin matrix factorization for collaborative filtering[J]. Pattern Recognition Letters, 2017, 86: 62-67.
- [16] HU Jun, LI Ping. Collaborative filtering via additive ordinal regression[C]//Proceedings of the 11th ACM International Conference on Web Search and Data Mining. Marina Del Rey: ACM, 2018: 243-251.
- [17] ZHUANG Fuzhen, ZHANG Zhiqing, QIAN Mingda, et al. Representation learning via Dual-Autoencoder for recommendation[J]. Neural Networks, 2017, 90: 83-89.
- [18] LIU Yong, ZHAO Peilin, LIU Xin, et al. Learning user dependencies for recommendation[C]// Proceedings of the Twenty - Sixth International Joint Conference on Artificial Intelligence. Melbourne: AAAI Press, 2017: 2379-2385.
- [19] QIAN Fulan, YUE Ruxia, ZHAO Shu, et al. Robust recommendation algorithm using an iterative group - based reputation[C]// IEEE International Conference on Cloud Computing & Big Data Analysis. Chengdu: IEEE, 2018: 62-70.
- [20] TAHERI S M, MAHYAR H, FIROUZI M, et al. Extracting implicit social relation for social recommendation techniques in user rating prediction [C]//Proceedings of the 26th International Conference on World Wide Web Companion. Leipzig: International World Wide Web Conferences Steering Committee, 2017: 1343-1351.
- [21] ZHANG Menghao, HU Binbin, SHI Chuan, et al. Local low -rank matrix approximation with preference selection of anchor points [C]//Proceedings of the 26th International Conference on World Wide Web Companion. Leipzig: [s.n.], 2017: 1395-1403.
- [22] LIU Xiaomeng, OUYANG Yuanyin, RONG Wenge, et al. Item category aware conditional restricted Boltzmann machine based recommendation [C]//Conference on Neural Information Processing. Istanbul: Springer, 2015: 609-616.
- [23] NGUYEN D, CALDERBANK R, DELIGIANNIS N, et al. Geometric matrix completion with deep conditional random fields [EB/OL]. (2019-01-29). <https://arxiv.org/pdf/1901.10429.pdf>.

(编辑:陈璐)