

深度协同过滤推荐模型研究

纪佳琪 姜学东

(河北民族师范学院 河北 承德 067000)

摘 要 近些年深度学习在计算机视觉、语音识别、自然语言处理等领域取得了巨大成就,然而在推荐系统中应用深度学习的研究还处于起步阶段。为了充分利用深度学习的强大特征学习能力,设计一个基于深度神经网络的协同过滤推荐模型。利用隐式数据建立用户向量和物品向量,通过对用户向量和物品向量深度表征,非线性地学习了用户和物品间的内在联系。在两个不同数据集上的实验表明,该模型在 HR 和 NDCG 评价指标上比基线模型有较大提升。

关键词 推荐系统 深度学习 协同过滤 矩阵分解

中图分类号 TP3 文献标志码 A DOI: 10.3969/j.issn.1000-386x.2020.07.040

DEEP COLLABORATIVE FILTERING RECOMMENDER MODEL

Ji Jiaqi Jiang Xuedong

(Hebei Normal University for Nationalities, Chengde 067000, Hebei, China)

Abstract In recent years, deep learning has made great achievements in the field of computer vision, speech recognition and natural language processing. However, the application of deep learning in recommender system is at the initial stage. In order to fully utilize the powerful feature learning ability of deep learning, we design the collaborative filtering recommender model based on deep neural network. The implicit data was used to build up user vectors and item vectors, and the inner relationship between users and items was non-linear learned by deep representation of user vectors and item vectors. Experiments on two different datasets show that the proposed model has a greater improvement than the baseline model in the evaluation of HR and NDCG indicators.

Keywords Recommender system Deep learning Collaborative filtering Matrix factorization

0 引 言

随着大数据时代的到来,人们已经进入了信息爆炸时代,推荐系统对缓解信息过载问题起到了重要的作用,在电子商务、社交网络、在线新闻等领域得到了广泛应用。传统的推荐模型可以分为三类:协同过滤推荐模型、基于内容的推荐模型和混合推荐模型^[1]。协同过滤推荐模型又可以分为基于用户的协同过滤模型、基于物品的协同过滤模型和基于矩阵分解的协同过滤模型。其中,基于矩阵分解的协同过滤模型以其

较好的推荐效果得到了广泛的应用。其基本原理是把用户属性和物品属性分别映射到两个隐空间,使用两个隐向量来表示用户和物品,然后把这两个隐向量的内积作为用户对物品喜好程度的预测值。以此为基础,众多学者采用多种方法对其改进以增加性能,如与基于邻域模型进行集成^[2],考虑物品内容属性的主题模型^[3]和基于知识的模型^[4]等。然而由于矩阵分解中的内积操作是线性的,因此并不能很好地捕获用户和物品间复杂的交互信息,无法从根本上进一步提升推荐性能。

近些年,深度学习由于其强大的非线性表征能力,

收稿日期:2019-05-20。2019年度河北省引进国外智力项目(2019YX009B);河北省引进留学人员项目(C20190179);承德市科学技术研究与发展计划项目(201801A080)。纪佳琪,讲师,主研领域:机器学习与深度学习,大数据,推荐系统。姜学东,高级实验师。

在计算机视觉^[5-6]、语音识别^[7]和自然语言处理^[8]等领域获得了突破性进展。然而把深度学习应用到推荐系统领域的研究还处于起步阶段,并且这些研究主要是利用其对物品的文本描述信息^[9]、音乐的声音信息^[10]或图像信息^[11]进行建模后再结合用户和物品的隐向量做内积操作,其本质仍是线性的,并没有很好地利用深度学习强大的非线性表征能力。

本文将使用深度学习技术对用户和物品的交互数据进行学习,建立一个深度协同过滤推荐模型。该模型是非线性的,使用了多层神经网络进行特征抽象后得出推荐结果。模型很好地将深度学习与协同过滤算法进行了结合,有效地提升了推荐系统的性能。

1 相关工作

在推荐系统研究领域,经常用到显式数据和隐式数据。使用显式数据的推荐算法一般用于评分预测,而使用隐式数据的推荐算法一般用于推荐排序(也叫Top-K推荐)。本文使用矩阵分解和深度学习技术对隐式数据进行建模。

1.1 矩阵分解模型

设 $|m|$ 和 $|n|$ 分别代表用户数和物品数,定义用户和物品的交互矩阵 $R \in \mathbf{R}^{|m| \times |n|}$ 。该矩阵可以来源于用户评分矩阵,对于有评分的项设为1,无评分的项设为0,见图1。


	i_0	i_1	i_2			i_0	i_1	i_2
u_0	0	0	5		u_0	0	0	1
u_1	0	0	0		u_1	0	0	0
u_2	3	0	0		u_2	1	0	0

图1 用户评分矩阵的转换

图1右侧矩阵中的1表示用户 u 对用户 i 有交互(如:购买后评分)0表示 u 和 i 之间没有交互是缺失数据。因此该交互矩阵将会是非常稀疏的。矩阵分解模型把每一个用户和物品分别映射到隐向量空间,使用 p_u 和 q_i 分别表示用户和物品的隐向量,则用户对物品的喜好程度的预测值 \hat{r}_{ui} 可以表示为 p_u 与 q_i 的内积:

$$\hat{r}_{ui} = p_u^T q_i = \sum_{k=1}^K p_{uk} q_{ik} \quad (1)$$

式中: k 表示隐空间的维度。矩阵分解模型假设隐空间中每个维度之间都是相互独立的,并且使用相同的权重进行线性组合,因此矩阵分解模型可以看作是隐向量等权重的线性组合模型。

1.2 深度学习

深度学习是在二十世纪五六十年代神经网络的基础上发展起来的,拥有输入层、隐藏层和输出层。当隐藏层数目很多时称为深度神经网络。输入的特征向量通过隐含层变换达到输出层,在输出层得到分类结果。随着对深度学习研究的进展,很多深度模型可以应用于推荐系统领域。Cheng等^[12]利用多层感知机设计了一种深广学习(Wide&Deep Learning)模型,通过利用用户特征、情境特征和项目特征等多源异构数据进行收集App推荐,模型具有较高的记忆能力和泛化能力。Covington等^[13]通过利用用户信息、情境信息、历史行为数据和项目的特征信息等多源异构数据,提出一种深度神经网络模型用于YouTube视频推荐。此外,还可以利用卷积神经网络进行音乐推荐^[14],利用RNN进行新闻推荐^[15],利用协同降噪自编码器进行Top-K推荐^[16]等。

2 深度协同过滤推荐算法

本节将详细介绍深度协同过滤推荐算法(Deep Learning based Collaborative Filtering, DLCF)。首先给出整体架构,然后进行形式化表示,最后定义损失函数并给出训练方法。

2.1 整体架构

为了利用深度学习强大的特征学习能力,本文使用了一个深度神经网络来学习用户和物品隐向量间的内在联系,学习得到 \hat{r}_{ui} 值,其网络架构如图2所示。

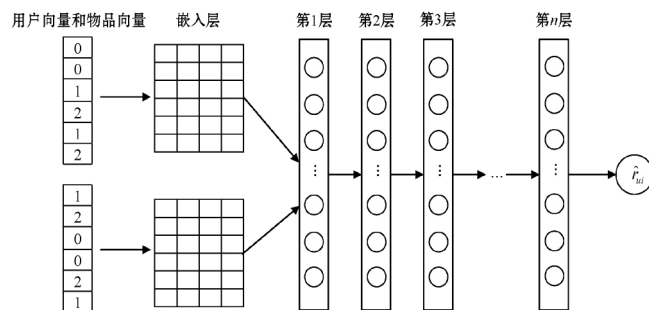


图2 DLCF 整体网络架构

首先分别把用户和物品的交互矩阵 R 转换为用户向量、物品向量和标签向量,其转换过程如图3所示。其中用户向量和物品向量是神经网络的输入,标签值用来和预测值做损失计算。然而由于 R 是 $|m| \times |n|$ 维的,因此用户向量和物品向量的维度也是 $|m| \times |n|$ 维的,而这个维度过大,从而计算开销过大。为了解决这个问题,本文将采用抽样方法对负例进行抽取,

即在 R 中抽取一定比例为 0 的值进入训练数据。图 4 模拟了抽取过程,其负例抽取的比例是正例的 2 倍。采用这样的抽样方法很好地解决了数据稀疏性所带来的问题。

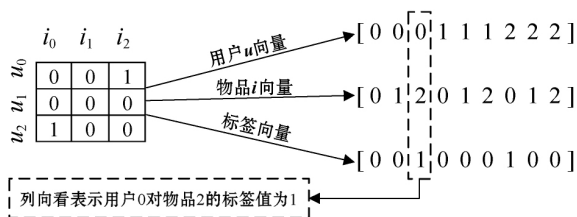


图3 R 转换为用户向量、物品向量和标签向量

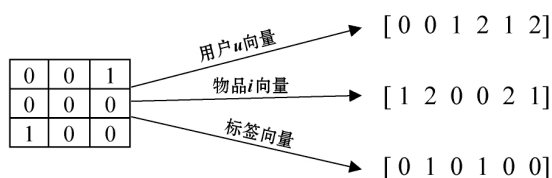


图4 对数据采样,负例数量是正例的 2 倍

在输入层之后是嵌入层,嵌入层是一个稠密矩阵,用户嵌入层矩阵和物品嵌入层矩阵可以分别代表用户隐空间和物品隐空间。然后把这两个矩阵分别拉伸为 2 个向量后,再连接成为 1 个向量作为后续网络的输入。接下来会进入一系列的网络层(共 n 层,是个调节参数),每一层都会发现用户和物品交互的内在关联。最后在输出层可以得到一个预测结果 \hat{r}_{ui} 。通过最小化 r_{ui} 和 \hat{r}_{ui} 之间的交叉熵损失,可以对该神经网络进行训练。

2.2 形式化表示

2.1 节所述的网络架构,可以形式化地表示如下:

$$\hat{r}_{ui} = f(P_u, Q_i | P, Q, \Theta) \quad (2)$$

式中: $P \in \mathbf{R}^{m \times k}$ 、 $Q \in \mathbf{R}^{n \times k}$ 分别表示用户隐空间和物品隐空间(对应图 2 中的嵌入层矩阵); Θ 是模型参数,本文使用的是深度神经网络,因此 Θ 就是各网络层参数。由此 f 函数可以表示为:

$$f(P_u, Q_i | P, Q, \Theta) = g_{out}(g_n(\dots g_2 g_1(w^T x + b))) \quad (3)$$

式中: g_n 表示第 n 层网络; g_{out} 表示应用某一激活函数(如 Sigmoid)后得到的预测结果。

2.3 模型学习

由于训练数据使用的是隐式数据,其值 r_{ui} 为 1 或 0,因此本文考虑使用概率的方法进行预测。我们把当前任务看成一个二分类问题,1 表示用户对物品感兴趣,0 表示不感兴趣。因此预测值 \hat{r}_{ui} 的取值范围是 $[0, 1]$,该值表示用户 u 对物品 i 的喜好程度。基于此,我们使用二分类交叉熵损失作为模型的损失函数,其定义如下:

$$L = - \sum_{(u,i) \in R^+ \cup R^-} r_{ui} \log \hat{r}_{ui} + (1 - r_{ui}) \log(1 - \hat{r}_{ui}) = - \sum_{(u,i) \in R^+} r_{ui} \log \hat{r}_{ui} - \sum_{(u,i) \in R^-} (1 - r_{ui}) \log(1 - \hat{r}_{ui}) \quad (4)$$

式中: R^+ 表示训练数据中的正例(即值为 1 的数据), R^- 表示训练数据中的负例(但不是全部,如 2.1 节所述是按一定比例进行采样)。

定义了损失函数后,就可以形式化地表示本文提出的深度神经网络结构。该架构具有较高的灵活性,通过非线性学习 p_u 和 q_i 的内在交互关系进而得到预测结果:

$$\begin{aligned} z_1 &= \begin{bmatrix} p_u \\ q_i \end{bmatrix} \\ z_2 &= a_2(w_2^T z_1 + b_2) \\ z_3 &= a_3(w_3^T z_2 + b_3) \\ &\vdots \\ z_{n-1} &= a_n(w_n^T z_{n-1} + b_n) \\ \hat{r}_{ui} &= \sigma(z_{n-1}) \end{aligned} \quad (5)$$

式中: w_i 、 b_i 分别表示第 i 层的权重、偏置;对于激活函数 a_i ,可以选择 Sigmoid 函数、tanh 函数和 ReLU 函数。Sigmoid 函数存在饱和区问题^[17],即在曲线平缓处产生梯度消失问题及当输出值接近 0 或 1 时学习停止,从而影响推荐效果。tanh 函数虽然可以在一定程度上缓解上述问题,但是无法从根本上避免,因为它可以看作是 Sigmoid 函数的重新缩放($\tanh(x) = 2\text{sigmoid}(2x) - 1$)。ReLU 函数是一个性能良好的函数并且不存在饱和区问题,因此我们选用 ReLU 作为激活函数。 σ 是输出层的激活函数,由于我们的输出结果要映射到 $[0, 1]$ 区间,因此选用 Sigmoid 函数。

3 实验

本文使用 Keras 进行模型编程实现,模型运行在 ubuntu 系统上。首先通过实验确定模型的超参数,然后再与其他基线算法进行比较分析。

3.1 数据集

本实验使用了两个公开的数据集: MovieLens 和 Book-Crossing。MovieLens 数据集广泛应用于对协同过滤推荐模型的评价,实验中使用的是 MovieLens-10m,它包含一百多万的评分数据并且每个用户至少存在 20 条评分记录。由于它是显式数据集,因此我们需要对其进行预处理,把它转换成隐式数据集,转换方

法是对有评分的项用 1 表示(无论评多少分),否则用 0 表示。Book-Crossing 数据集是 Cai-Nicolas Ziegler 博士使用爬虫程序从 Book-Crossing 图书社区爬取的图书评分数据。由于原始数据很大,并且有的用户对物品的评分数目很少,因此数据非常稀疏。为此首先抽取用户至少存在 20 条评分记录的数据,然后再把这些数据转换为隐式数据。表 1 列出了预处理后 2 个数据集的详细信息。

表 1 MovieLens 和 Book-Crossing 数据集详情

数据集	评分数	物品数	用户数	稀疏度
MovieLens-10m	1 000 209	3 700	6 040	95.53%
Book-Crossing	1 021 650	61 264	1 026	99.84%

3.2 评价指标

为了评价模型性能,使用了十折交叉验证,即把数据分割成 10 个子集,1 个子集被保留作为验证模型的数据,其他 9 个子集用来训练。交叉验证重复 10 次,每个子集验证一次,取 10 次的平均值作为最终的结果。使用命中率(Hit Ratio, HR)和归一化折损累积增益(Normalized Discounted Cumulative Gain, NDCG)作为评价指标。在 HR 和 NDCG 后加@K 符号表示推荐列表长度是 K 时(即做 Top-K 推荐),HR 和 NDCG 的值。HR 的计算公式如下:

$$HR@K = \frac{\sum_u \text{NumberofHits}@K}{|Test|} \quad (6)$$

式中:分子表示每个用户 Top-K 列表中出现在测试集中的个数总和,分母表示测试集大小。NDCG 的计算公式如下:

$$NDCG@K = \frac{DCG@K}{IDCG@K} = \frac{\sum_{i=1}^p \frac{2^{reli} - 1}{\log_2(i+1)}}{\sum_{i=1}^{|REL|} \frac{2^{reli} - 1}{\log_2(i+1)}} \quad (7)$$

式中:reli 表示在第 i 个位置时的“等级关联性”,一般可以用 0/1 处理,如果该位置的物品在测试集中,则 reli = 1,否则为 0。分母是归一化系数,|REL|表示按照相关性从大到小的顺序排序,取前 k 个组成的集合。

3.3 超参数设置

如 2.1 节所述,我们在训练阶段需要采样一定的负例,本文采取的方法是在每一轮迭代过程中采样 5 倍于正例的负例,这些负例和所有正例作为训练数据。使用 Adam 的优化方法, batch-size 设置为 256,学习率设置为 0.000 4。实验的重点是要确定神经网络的层数,以及每层中神经元的个数,为此我们分别选取 3 至

8 层不同网络结构进行实验(固定每层神经元个数都为 8),结果如表 2 所示。

表 2 不同网络层数下 HR 和 NDCG 值

网络层数	MovieLens-10m		Book-Crossing	
	HR@8	NDCG@8	HR@8	NDCG@8
3 层	0.54	0.298	0.627	0.389
4 层	0.572	0.313	0.685	0.409
5 层	0.601	0.322	0.711	0.435
6 层	0.601	0.323	0.712	0.437
7 层	0.600	0.322	0.712	0.437
8 层	0.601	0.323	0.712	0.436

从表 2 可以看出,无论是 MovieLens-10m 还是 Book-Crossing 数据集,当网络层数从 3 增长到 5 时,HR 和 NDCG 值都有明显的增长趋势,此时网络层数的增加有助于推荐效果的提升。当网络层数为 7、8 时,HR 和 NDCG 不再增加,甚至略有降低,说明无限地增加网络层数并不能无限制地提升推荐效果。当网络层数为 6 时,虽然 HR 和 NDCG 都在增加,但增幅不大。虽然本实验中具有 6 层网络比具有 5 层网推荐效果略有提升,但考虑到网络层数越多计算量越大,权衡计算量与推荐效果后,我们把网络层数设置为 5。

接下来考虑每个网络层中神经元的个数。大量实验表明,沿着网络正向传播的方向每层神经元个数递减的设置方式有利于性能的提高。给出 5 种神经元个数不同的设置方法,其结果见表 3。[32, 16, 8, 4, 2]表示每层神经元的个数分别是 32、16、8、4、2,以此类推。

表 3 不同神经元个数下 HR 和 NDCG 值

每层神经元个数	MovieLens-10m		Book-Crossing	
	HR@8	NDCG@8	HR@8	NDCG@8
[32, 16, 8, 4, 2]	0.603	0.325	0.714	0.439
[64, 32, 16, 8, 4]	0.619	0.342	0.732	0.452
[128, 64, 32, 16, 8]	0.641	0.361	0.753	0.471
[256, 128, 64, 32, 16]	0.642	0.362	0.754	0.472
[512, 256, 128, 64, 32]	0.642	0.363	0.755	0.473

从表 3 中可以看出,随着每层神经元个数的增加,HR 和 NDCG 总体呈上升趋势,但是在神经元为 [128, 64, 32, 16, 8]后,即使神经元个数增加了,HR 和 NDCG 上升量却不大,但计算开销量却成倍增加。因此权衡计算量与推荐效果后我们选择 [128, 64, 32, 16, 8]作

为每层的神经元个数。

通过上述实验,我们确立了模型超参数,其汇总见表4。后面的对比实验都是基于这些超参数得出的实验结果。

表4 模型超参数汇总

优化方法	Adam
批量大小(batch-size)	256
学习率	0.000 4
网络层数	5 层
每层神经元个数	[128 64 32 16 8]

3.4 对比实验

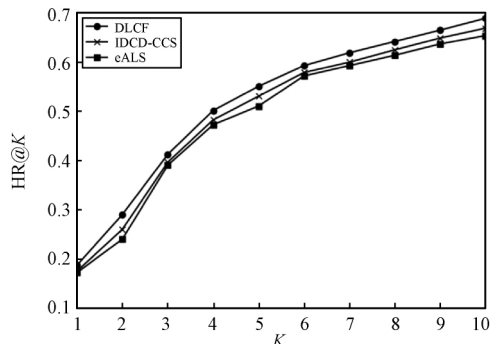
为了验证本文模型的有效性,与以下基线模型进行了对比实验。

(1) Item-based Collaborative Filtering^[18]: 基于物品的协同过滤模型,该模型目前在工业界有着广泛的应用,也是最早基于协同过滤推荐算法的模型之一。

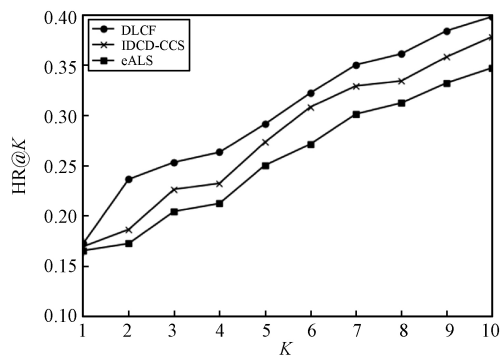
(2) IRCD-CCS^[19]: 该模型把深度学习方法和协同过滤方法相结合,使用显式数据解决物品评分问题,其经过显式数据转换为隐式数据后也能用于 Top-K 推荐。

(3) eALS^[20]: 一种使用矩阵分解对隐式数据进行建模的方法,该方法根据物品的流行度对缺失数据赋予不同的权重,然后使用加权平方作为损失函数进行训练,取得了良好效果。

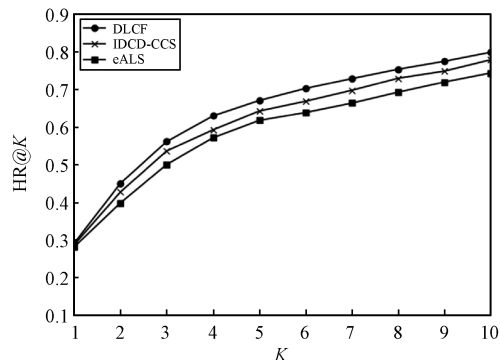
图5显示了4种模型分别在2个数据集上 HR@K 和 NDCG@K 的值(K取值从1到10)。可以看出,本文模型 DLCF 无论在哪个数据集上都远好于其他3个模型。这主要是因为基于物品的协同过滤其本质是线性的,无法对复杂的交互进行建模。IRCD-CCS 虽然使用 SDAE 方法进行非线性建模,但相比本文模型其层次不够深,无法深层次地进行特征抽取。本文使用了深度神经网络,能够更好地进行深度建模,取得良好的推荐效果。



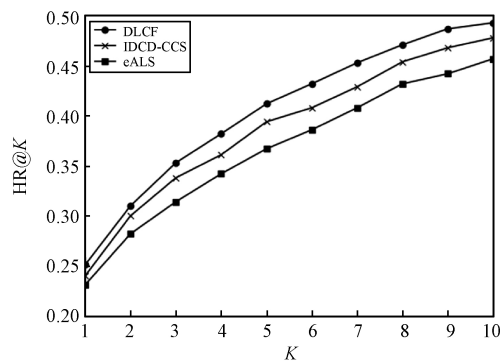
(a) MovieLens-10M HR@K



(b) MovieLens-10M NDCG@K



(c) Book-Crossing HR@K



(d) Book-Crossing NDCG@K

图5 当K从1到10时模型在2个数据集上 HR 和 NDCG 值

4 结 语

本文把深度学习技术与协同过滤推荐算法进行了结合,设计了使用深度神经网络的协同过滤推荐模型,该模型能够深度地非线性学习用户和物品间的内在联系,从而产生较好的推荐结果。下一步将继续关注深度学习在推荐系统中的应用,把物品的属性特征(如:描述、评论、标签等)通过深度学习技术学习后作为输入特征引入到推荐模型中来,从而产生更好的推荐结果。

参 考 文 献

- [1] Bajpai V, Yadav Y. Survey paper on dynamic recommendation system for e-commerce [J]. International Journal, 2018, 9(1): 774-777.

- [2] Yang C, Liu T, Liu L, et al. A nearest neighbor based personal rank algorithm for collaborator recommendation [C]//2018 15th International Conference on Service Systems and Service Management(ICSSSM). IEEE 2018.
- [3] Abdi M H, Okeyo G O, Mwangi R W. Matrix factorization techniques for context-aware collaborative filtering recommender systems: A survey [J]. Computer and information science 2018, 11(2): 1-11.
- [4] Tarus J K, Niu Z, Mustafa G. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning [J]. Artificial Intelligence Review 2018, 50: 21-48.
- [5] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection [EB]. arXiv: 1506.02640 2015.
- [6] Han X, Du Q D. Research on face recognition based on deep learning [C]//2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications(DINWC). IEEE 2018.
- [7] Ravanelli M, Brakel P, Omologo M, et al. Light gated recurrent units for speech recognition [J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2018, 2(2): 92-102.
- [8] Li H. Deep learning for natural language processing: advantages and challenges [J]. National Science Review, 2018, 5(1): 28-30.
- [9] Kim D, Park C, Oh J, et al. Convolutional matrix factorization for document context-aware recommendation [C]//Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016: 233-240.
- [10] Zhang S, Yao L, Sun A, et al. Deep learning based recommender system: A survey and new perspectives [J]. ACM Computing Surveys, 2019, 52(1): 5.
- [11] Tuan T X, Phuong T M. 3D Convolutional Networks for Session-based Recommendation with Content Features [C]//Proceedings of the Eleventh ACM Conference on Recommender Systems. ACM, 2017: 138-146.
- [12] Cheng H T, Koc L, Harmsen J, et al. Wide & Deep Learning for Recommender Systems [C]//Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016: 7-10.
- [13] Covington P, Adams J, Sargin E. Deep neural networks for youtube recommendations [C]//Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016: 191-198.
- [14] Oord A V D, Dieleman S, Schrauwen B. Deep content-based music recommendation [C]//Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013: 2643-2651.
- [15] Okura S, Tagami Y, Ono S, et al. Embedding-based news recommendation for millions of users [C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2017: 1933-1942.
- [16] Wu Y, Dubois C, Zheng A X, et al. Collaborative denoising auto-encoders for top-N recommender systems [C]//Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM, 2016: 153-162.
- [17] Li J C, Ng W W Y, Yeung D S, et al. Bi-firing deep neural networks [J]. International Journal of Machine Learning & Cybernetics, 2014, 5(1): 73-83.
- [18] Deshpande M, Karypis G. Item-based top-N recommendation algorithms [J]. ACM Transactions on Information Systems, 2004, 22(1): 143-177.
- [19] Wei J, He J, Chen K, et al. Collaborative filtering and deep learning based recommendation system for cold start items [J]. Expert Systems with Applications, 2017, 69: 29-39.
- [20] He X, Zhang H, Kan M Y, et al. Fast Matrix Factorization for Online Recommendation with Implicit Feedback [C]//Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM 2016: 549-558.
- ~~~~~
- (上接第 169 页)
- [9] Daniels J I, Ha L K, Ochotta T, et al. Robust smooth feature extraction from point clouds [C]//IEEE International Conference on Shape Modeling and Applications 2007. IEEE, 2007.
- [10] 张雨禾, 耿国华, 魏潇然, 等. 采用密度空间聚类的散乱点云特征提取方法 [J]. 西安电子科技大学学报, 2017, 44(2): 114-120.
- [11] Rusu R B, Blodow N, Beetz M. Fast point feature histograms(FPFH) for 3D registration [C]//2009 IEEE International Conference on Robotics and Automation. IEEE 2009.
- [12] 李宝, 程志全, 党岗, 等. 三维点云法向量估计综述 [J]. 计算机工程与应用, 2010, 46(23): 1-7.
- [13] Hoppe H, DeRose T, Duchamp T, et al. Surface reconstruction from unorganized points [J]. ACM SIGGRAPH Computer Graphics, 1992, 26(2): 71-78.
- [14] 刘增艺, 江开勇, 林俊义. 散乱点云特征边缘交互提取 [J]. 计算机工程与应用, 2016, 52(6): 186-190.
- ~~~~~
- (上接第 175 页)
- [15] 李海燕, 李洪达, 刘元龙, 等. 基于梯度方向场的掌纹三角点检测算法 [J]. 计算机应用与软件, 2016, 33(12): 148-151, 192.
- [16] 王云, 任雪婷, 张玲. 探析室内装饰设计中的软装饰 [J]. 山西建筑, 2017(3): 220-221, 227.
- [17] Huang X, Liu M Y, Belongie S, et al. Multimodal unsupervised image-to-image translation [C]//Proceedings of the European Conference on Computer Vision 2018: 172-189.