

基于双层注意力机制的联合深度推荐模型

刘慧婷 纪强 刘慧敏 赵鹏

(安徽大学 计算机科学与技术学院, 安徽 合肥 230601)

摘要: 许多电子商务网站中存在用户编写的大量评论信息, 大部分推荐系统虽然利用了评论信息, 但仅从单词级别而不是评论级别来评估评论的重要性。如果只考虑评论中的重要单词, 而忽略了真正有用的评论, 则会降低推荐模型的性能。基于此, 文中提出了一种基于双层注意力机制的联合深度推荐模型 (DLALSTM)。该模型首先利用双向长短期记忆网络 (BiLSTM) 分别对用户和项目评论进行词以及评论级别联合建模, 并通过两层注意力机制聚合为评论表示和用户/项目表示, 然后把从评论中学习的用户和项目的潜在表示融入由评分矩阵得到的用户偏好和项目特征, 实现评分预测。采用文中模型在 Yelp 和亚马逊的不同领域数据集上进行实验评估, 并与常用的推荐方法进行比较, 发现文中提出的模型性能超过目前常用的推荐方法, 同时该模型能够缓解数据稀疏问题, 且具有较好的可解释性。

关键词: 注意力机制; 双向长短期记忆网络; 推荐系统; 深度学习

中图分类号: TP391

文章编号: 1000-565X(2020)06-0097-09

随着在线购物网站的发展, 利用用户的个性偏好进行推荐变得越来越普遍。现有推荐系统可以分为3大类: 基于协同过滤 (CF) 的推荐、基于内容的推荐和混合推荐。推荐系统大多采用基于 CF 技术的推荐算法。这些算法的基本思想是: 具有相似偏好的用户往往会有类似的选择。简而言之, 协同过滤算法直接或间接依赖用户的评级, 以进行评级预测或者生成排名推荐。

尽管协同过滤算法在许多应用中表现出了良好的性能, 但是该算法有其自身的局限和缺点, 稀疏性问题是其面临的重大挑战之一^[1]。当用户对商品的评分非常稀疏时, 基于用户评级所得到的用户间的相似性可能不准确, 会出现稀疏性问题, 且该问题存在于许多实际应用中。

协同过滤技术的另一个缺点是可解释性差, 其

无法很好地解释用户的偏好。此技术只知道用户可能因为某个特定的特征而喜欢某个项目, 但不知道这个特征的含义。事实上, 用户喜欢某个项目的原因都是可以解释的, 例如餐厅的美食风格和价格等。使用评论文本进行推荐是解决上述问题的一种方法。

为了解决稀疏性问题, 研究人员还提出了基于内容的推荐系统。这些方法利用项目信息或者用户评论信息了解用户或项目的潜在特征。尽管基于内容的推荐系统可以有效地推荐新项目, 但是由于没有足够的有关用户个人信息, 这类方法无法生成个性化的预测。

现在越来越多的模型将 CF 和基于内容的方法结合起来进行推荐^[2-6], 取得了不错的效果。例如模型 sCVR^[2] 使用评论信息和社交关系来预测项目

收稿日期: 2019-11-12

基金项目: 国家自然科学基金资助项目 (61202227, 61602004); 安徽省高校自然科学基金项目 (KJ2018A0013)

Foundation items: Supported by the National Natural Science Foundation of China (61202227, 61602004) and the Natural Science Research Project of Colleges and Universities in Anhui Province (KJ2018A0013)

作者简介: 刘慧婷 (1978-), 女, 博士, 副教授, 主要从事自然语言处理和个性化推荐研究。E-mail: htliu@ahu.edu.cn

评分,实现了很好的可解释性。而文献[3]提出了深度学习框架 NRT,该模型可以在预测评分的同时生成摘要性的提示,提高了模型的推荐质量和可解释性。

目前,在大多数电子商务和评论服务网站如亚马逊和 Yelp 中,用户可以自由编写评论文本以及数字星级评分。评论文本中通常包含有关项目特征(例如质量、材料和颜色)的丰富信息,有时还会包含针对性的建议,这些建议对于那些打算作出购买决策的人来说具有很好的参考价值。但由于并不是所有的评论都是有价值的,所以用户从大量的评论中获取有用的信息是比较困难的。不太有用的评论会给推荐模型带来噪声,影响模型的性能,对用户的帮助也会有限,因此选择有用的评论对于提高模型的性能是非常重要的。

已有相关研究使用评论文本来提高评级预测模型的准确性,例如 CTR^[7]、HFT^[8]、RBLT^[9]等,它们证明了使用评论信息进行推荐的有效性,特别是对于评级较少的用户和项目。但是这些模型利用隐狄利克雷分配模型(LDA)^[10]或非负矩阵分解(NMF)^[11]从评论中提取潜在特征,其不足之处是它们使用词袋(BOW)表示来进行评论文本的处理,而忽略了评论的上下文语义信息。

由于深度神经网络(例如 RNN、CNN(卷积神经网络)等)对序列文本有着强大的建模能力,因而有研究利用深度神经网络提取用户评论中的有用信息来提高基于潜在因子建模的模型性能^[12-13]。例如,DeepCoNN 模型用 CNN 来编码评论信息,然后通过因子分解机(FM)^[14]耦合的两个并行部分对用户和项目进行联合建模,以进行评级预测。虽然该模型取得了很好的推荐性能,但它从用户或者项目的评论中无区别地提取信息,不能够很好地区分用户评论的差异性。

对于模型缺乏可解释性的问题,现有模型利用用户评论来提高推荐性能并生成解释^[2,15-16],但这些模型也存在一些局限性。首先,对于项目建模它们没有考虑每个评论的贡献程度,也没有区分发表无用评论的用户;其次,它们从评论中简单地提取单词或短语来为模型提供解释,缺乏评论中的上下文信息,这可能会扭曲原来句子的含义。

为了解决上述问题,提取有用的用户或者项目评论信息,本文提出了一种融合双层注意力机制的神经网络模型(DLALSTM),利用双向长短期记忆网络(BiLSTM)^[17]来学习用户和项目评论信息,以更好地提取评论中每个词的上下文信息;同时利

用双层注意力机制分别考虑词和评论的重要性,使得模型具有良好的可解释性。

1 双层注意力深度推荐模型

一个用户对不同的项目可能有多个评论信息,而项目也可能会接收不同用户对它的多个评论信息。一方面,在用户或者项目的多个评论信息中,有些评论信息在整个用户或者项目建模中起着非常重要的作用,而其他一些评论信息可能对于建模用户偏好和项目特征没有那么重要。另一方面,评论中不同单词的语义不同。一些单词能够体现出用户的偏好,而另一些则倾向于表明项目的特征。这意味着评论信息在词级别和评论级别具有不同的潜在语义表示。为了体现这两种信息在推荐模型中不同的作用,本文提出了一个新的框架 DLALSTM,来将用户和项目评论信息分别从用户视图和项目视图的角度,通过双层注意力机制并行地引入推荐模型,如图1所示,其中各符号的意义见下文。本文提出的双层注意力模型包含用户视图和项目视图两个并行的部分,分别实现用户建模和项目建模。整个模型由输入层(词嵌入层)、序列编码层、第1层注意力层、第2层注意力层、全连接层、评分预测层6部分组成。下文将从项目视图角度进行详细介绍,用户视图角度的建模和项目视图相似。

1.1 输入层

模型 DLALSTM 使用经过预训练的词向量作为输入,这些向量由 Google News^[18]中超过1000亿个单词训练得到。将用户的所有评论换为单词向量矩阵,表示为 $V_{u1}, V_{u2}, \dots, V_{uj}$, 其中 V_{uy} ($y = 1, 2, \dots, j$) 为用户 u 的第 y 条评论中的词 u_{yp} ($p = 1, 2, \dots, m$) 预训练后所构成的矩阵, j 为每个用户最大的评论数。同理,项目 i 的所有评论也转换成单词向量矩阵,表示为 $V_{i1}, V_{i2}, \dots, V_{ik}, V_{iz}$ ($z = 1, 2, \dots, k$) 为项目 i 的第 z 条评论中的词 i_{zq} ($q = 1, 2, \dots, n$) 预训练后所构成的矩阵,其中 k 为每个项目最大的评论数。然后将这些矩阵分别输入到两个并行 BiLSTM 中。

1.2 序列编码层

为更好地利用上下文信息,常见的方法是采用双向 LSTM(长短期记忆网络)来模拟正向和反向的文本语义。例如对于文本序列 $[x_1 \ x_2 \ \dots \ x_T]$, 正向 LSTM 是从 x_1 到 x_T 读取序列,反向 LSTM 则是从 x_T 到 x_1 读取序列。连接正向隐藏状态 \vec{h}_i ($i = 1, 2, \dots, T$) 和反向隐藏状态 \overleftarrow{h}_i ^[19-20], 即 $h_i = \vec{h}_i \parallel \overleftarrow{h}_i$ 。 h_i 包含了以 x_i 为中心的整个文本序列

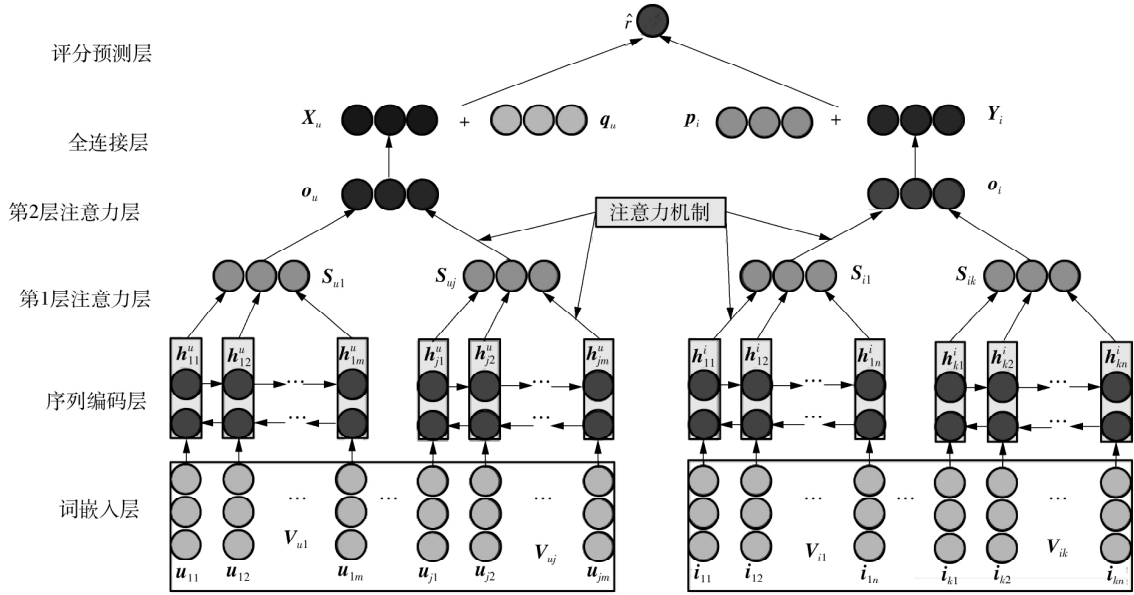


图1 双层注意力推荐框架图

Fig. 1 Double-layer attention recommendation frame

的信息, 因此对于项目评论中的词向量序列 \$[W_{L1} \ W_{L2} \ \dots \ W_{Ln}]\$, 本文采用 BiLSTM 来编码正向和反向的项目评论文本语义:

$$\overrightarrow{h}_{LM}^i = \text{BiLSTM}(\overrightarrow{W}_{LM} \ \overrightarrow{h}_{LM-1}^i) \quad (1)$$

$$\overleftarrow{h}_{LM}^i = \text{BiLSTM}(\overleftarrow{W}_{LM} \ \overleftarrow{h}_{LM+1}^i) \quad (2)$$

式中: \$\text{BiLSTM}\$ 表示正向的 LSTM, \$\overleftarrow{\text{BiLSTM}}\$ 表示反向的 LSTM; \$\overrightarrow{W}_{LM}\$ 表示评论文本中的词向量, \$L = 1, 2, \dots, k\$; \$M = 1, 2, \dots, n\$。

对于项目评论中第 \$L\$ 个评论中的第 \$M\$ 个词 \$i_{LM}\$, 首先训练为词向量 \$W_{LM} \in \mathbf{R}^d\$。对于词嵌入序列 \$V_{iL} = [W_{L1} \ W_{L2} \ \dots \ W_{Ln}]\$, 正向 LSTM 从 \$W_{L1}\$ 到 \$W_{Ln}\$ 读取序列, 而反向 LSTM 从 \$W_{Ln}\$ 到 \$W_{L1}\$ 读取序列。连接正向隐藏状态 \$\overrightarrow{h}_{LM}^i\$ 和反向隐藏状态 \$\overleftarrow{h}_{LM}^i\$, 即 \$h_{LM}^i = \overrightarrow{h}_{LM}^i \parallel \overleftarrow{h}_{LM}^i\$。最终项目视图中的 BiLSTM 网络生成隐藏状态 \$\{h_{L1}^i, h_{L2}^i, \dots, h_{Ln}^i\}\$。

1.3 第1层注意力层

从项目评论的角度来看, 评论中的每个单词在提取项目特征方面的贡献度不同。为了体现这种不同, 项目视图模型使用注意力机制来提取评论中对项目特征的提取有重要贡献的信息词, 然后, 评论则由这些信息词共同表示, 即评论表示 \$S_{iL}\$ 为项目评论中单词的隐藏状态的加权和:

$$S_{iL} = \sum_{M=1}^n \alpha_{LM}^i h_{LM}^i \quad (3)$$

式中: \$h_{LM}^i\$ 是项目 \$i\$ 的第 \$L\$ 个评论中第 \$M\$ 个单词的

隐藏状态; \$\alpha_{LM}^i\$ 是 \$h_{LM}^i\$ 对应的注意力权重, 用于衡量第 \$M\$ 个单词对当前项目评论的重要性。具体来说, 每个隐藏状态的注意力权重 \$\alpha_{LM}^i\$ 可以定义为

$$\alpha_{LM}^i = \frac{\exp(\alpha_{LM}^{i*})}{\sum_{M=1}^n \exp(\alpha_{LM}^{i*})} \quad (4)$$

$$\alpha_{LM}^{i*} = (v_w^i)^T \tanh(W_{wh}^i h_{LM}^i + b_w^i) \quad (5)$$

式中: \$v_w^i\$ 是权重向量, \$(v_w^i)^T\$ 表示它的转置; \$W_{wh}^i\$ 是权重矩阵; \$b_w^i\$ 是偏置项; \$\tanh\$ 是非线性激活函数。

1.4 第2层注意力层

项目可能有多个评论, 而每个评论具有不同的重要程度, 为了提取出有代表性的评论中的有用信息, 模型引入第2层注意力机制学习每个评论的权重。基于注意力的评论聚合的目标是选择有代表性的评论, 然后汇总这些代表性评论的信息来表示这个项目。该层的输入包含评论的特征向量 \$S_{iL}\$ 和编写该评论的用户 ID 嵌入 \$u_{iL}'\$。添加用户 ID 嵌入有助于识别发表无用的评论用户。具体来说, 第2层注意力网络被定义为

$$\alpha_{iL}^* = H^T \text{relu}(W_s S_{iL} + W_u u_{iL}' + b_1^i) + b_2^i \quad (6)$$

式中: \$H\$ 是权重向量, \$H^T\$ 是它的转置; \$W_s\$、\$W_u\$ 分别为 \$S_{iL}\$、\$u_{iL}'\$ 的权重矩阵; \$b_1^i\$、\$b_2^i\$ 是偏置项; \$\text{relu}\$ 是非线性激活函数。

本文使用 softmax 函数对上述注意力得分进行归一化来获得评论的最终权重, 可以解释为第 \$L\$ 个评论对项目 \$i\$ 的整个特征的贡献程度:

$$\alpha_{iL} = \frac{\exp(\alpha_{iL}^*)}{\sum_{L=1}^k \exp(\alpha_{iL}^*)} \quad (7)$$

在获得每个评论的注意力权重之后,项目*i*的特征向量的计算公式如下:

$$\mathbf{o}_i = \sum_{L=1}^k \alpha_{iL} \mathbf{S}_{iL} \quad (8)$$

1.5 全连接层

为了保证由 LFM^[1]得到的用户偏好和项目特征,与分别通过用户视图和项目视图建模得到的评论级别的用户和项目特征在同一特征空间,本文将第2层注意力层的输出输入到全连接层来获得一个*d*维向量,则项目*i*的最终表示为

$$\mathbf{Y}_i = \mathbf{W}_0 \mathbf{o}_i + \mathbf{b}_0^i \quad (9)$$

式中: $\mathbf{W}_0 \in \mathbf{R}^{d \times k}$, 是权重矩阵; $\mathbf{b}_0^i \in \mathbf{R}^d$, 是偏置项。

对于用户视图中的建模,和项目视图建模相似,并非所有单词都反映用户的偏好。为了解决这个问题,本文使用第1层用户注意力机制来提取对评论含义有重要贡献的用户特定词,同时使用第2层注意力机制提取对用户具有代表性的评论。在用户的视图建模中,用户的一个评论表示 S_{ug} 和用户评论集合表示 \mathbf{o}_u 的计算公式如下:

$$\mathbf{S}_{ug} = \sum_{f=1}^m \alpha_{gf}^u \mathbf{h}_{gf}^u \quad (10)$$

$$\mathbf{o}_u = \sum_{g=1}^m \alpha_{ug} \mathbf{S}_{ug} \quad (11)$$

$$\mathbf{X}_u = \mathbf{W}_1 \mathbf{o}_u + \mathbf{b}_3^i \quad (12)$$

式中, α_{gf}^u 是隐藏层 \mathbf{h}_{gf}^u 的单词级权重, $g=1, 2, \dots, j, f=1, 2, \dots, m, m$ 为用户评论中最大单词数, α_{ug} 是用户评论 S_{ug} 所对应的评论级权重, \mathbf{X}_u 是从用户视图中学到的用户偏好特征, \mathbf{W}_1 是 \mathbf{o}_u 的权重矩阵, \mathbf{b}_3^i 为偏置项。

1.6 评分预测层

在评分预测层,本文将 LFM 模型中的用户偏好分布和项目特征分布分别扩展为两个组件:基于评级的特征和基于评论的特征。具体而言,首先将用户和项目的基于评级的潜在因子映射到共享的隐藏空间,然后引入从评论中学习的基于评论的潜在表示,用户*u*和项目*i*之间的交互则被建模为

$$\mathbf{h}_0 = (\mathbf{q}_u + \mathbf{X}_u) \odot (\mathbf{p}_i + \mathbf{Y}_i) \quad (13)$$

根据式(13), \mathbf{q}_u 和 \mathbf{p}_i 是基于评级的用户偏好和项目特征; \mathbf{Y}_i 则是从评论中学习的基于评论的项目特征; \odot 表示向量的元素乘积。

式(13)的输出是一个*d*维向量,将其传递给预测层以得到预测评级 \hat{r}_{ui} :

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{W}_2^T \mathbf{h}_0 \quad (14)$$

式中: $\mathbf{W}_2 \in \mathbf{R}^d$, 表示预测层的权重; b_u 、 b_i 和 μ 分别表示用户偏差、项目偏差和全局平均偏差。

1.7 模型学习

由于本文关注的任务是评级预测,模型通过 Adam 优化器最小化以下目标函数来更新网络参数。这实际上是一个回归问题。对于回归,常用的目标函数是平方损失^[21]:

$$\text{Loss} = \sum_{u, i \in \mathcal{R}} (\hat{r}_{ui} - r_{ui})^2 + \Omega \quad (15)$$

式中: \mathcal{R} 表示训练集; r_{ui} 是用户*u*给项目*i*的真实评级; Ω 是正则化项,以防止过拟合。

2 实验

为了验证本文所提模型的性能,将在4个真实数据集上对基于双层注意力机制的推荐模型 DLAL-STM 和目前常用的推荐模型进行对比和分析。

2.1 数据集

实验将使用来自不同领域的4个可公开访问的数据集来评估本文的模型。其中3个数据集来自亚马逊,分别为 Toys_and_Games、Kindle_Store 和 Movies_and_TV; 另一个数据集来自 Yelp。这些数据集涵盖了不同的领域,包含不同个数的用户、项目和评论信息。其中, Movies_and_TV 是最大的数据集,它包含超过160万条评论,而 Toys_and_Games 是较小的数据集,仅包含约16万条评论。数据集的特征统计如表1所示。本文将数据集随机分成训练集(80%)、验证集(10%)和测试集(10%),验证集用于调整超参数,测试集则用于最终性能的比较。这些数据集的评分等级为整数1到5。本文对于亚马逊数据集预处理使用文献[22]公开的代码进行处理。由于评论的长度和数量具有长尾效应,本文保留的长度和数量为前*p*%的用户和项目的评论,对于 Toys_and_Games 和 Kindle_Store, *p* 设置为0.9,而对于 Movies_and_TV, *p* 则设置为0.85。Yelp 数据集包含来自 Yelp Challenge 2018 的商户评论。由于原始数据非常庞

表1 4个实验数据集统计

Table 1 Statistics of four datasets

数据集	用户数	项目数	评级数	评论数
Toys_and_Games	19412	11924	167597	167597
Kindle_Store	68223	61935	982619	982619
Movies_and_TV	123960	50052	1697533	1697533
Yelp	8744	14082	212922	212922

大且稀疏,因此本文遵循 Zhang 等^[15]对该数据集

进行的预处理。具体来说, 本文选择位于凤凰城 (Phoenix) 的商户, 并确保所有用户和项目至少有 10 个评论。

2.2 评价指标

RMSE (均方根误差) 是常用的评估协同过滤推荐模型性能的指标, 因此本文在实验中利用它来评估各推荐模型的推荐精度, RMSE 的值越小, 表示预测精度越高。给定预测评级 $\hat{r}_{u,i}$ 和用户 u 对于项目 i 的真实评级 $r_{u,i}$, RMSE 的计算公式如下:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{u,i} (\hat{r}_{u,i} - r_{u,i})^2} \quad (16)$$

其中 N 表示用户和项目之间的评级数目。

2.3 超参数

本文在 4 个数据集上进行了多组实验, 对比方法的参数设置如下: NMF 和 SVD++ 设置潜在因子数为 10, HFT 设置主题数为 50。对于模型 DeepCoNN, 本文使用了文献 [12] 中大多数的超参数设置, 其中, 卷积层中的神经元数为 100, 窗口大小为 3。在实验过程中, 本文模型 DLALSTM 的隐藏层大小的取值范围为 16、32、50、64、100、128、150。为防止过拟合, 本文在全连接层加入 dropout 方法, 将 dropout 在 0.1、0.3、0.5、0.7 和 0.9 中进行最佳参数搜索。批量大小分别取 16、32、50、64 和 100, 潜在因子维数则在 8、16、32、64 中选择并进行实验分析。根据实验结果, 本文将隐藏层大小设置为 128, dropout 值设置为 0.3, 学习率设置为 0.001, 批量大小设置为 64, 潜在因子维数设置为 32。

2.4 算法性能比较

为了评估模型评级预测的性能, 本文将模型 DLALSTM 与 5 种目前常用的推荐方法在 4 个不同领域的数据集上进行性能比较。5 种推荐模型分别为 PMF^[23] (引入高斯分布来模拟用户和项目的潜在因子)、NMF^[11] (仅使用评级矩阵作为输入)、SVD++^[24] (模型集成了隐式反馈数据, 有效地将显式和隐式数据进行结合)、HFT^[8] (将评论与评级相结合的有效方法) 和 DeepCoNN^[12] (利用深度学习技术从评论文本中分别模拟用户和项目)。与其他基于主题建模的方法相比, DeepCoNN 模型的推荐性能有了显著改进。本文实现了这个模型, 并将优化器由 RMSprop 更改为 Adam, 因为该推荐模型使用 Adam 优化器比使用 RMSprop 的推荐性能更加有效。DLALSTM 与 5 种对比方法的统计信息比较结果如表 2 所示。

表 2 算法统计信息比较¹⁾

特征	PMF	NMF	SVD++	HFT	DeepCoNN	DLALSTM
评分	✓	✓	✓	✓	✓	✓
评论文本				✓	✓	✓
深度学习					✓	✓
评论有用性						✓
注意力机制						✓

1) “✓”表示对应算法中使用到了该特征。

2.5 实验结果与分析

本节将从 3 个方面对模型的实验结果进行分析: 首先, 对模型 DLALSTM 和 5 个对比方法在 4 个数据集上的实验结果 (RMSE) 进行分析; 然后, 进一步分析基于深度学习的推荐模型对参数的敏感性; 最后, 分析双层注意力机制对模型推荐性能的影响以及进行模型的收敛性实验。

2.5.1 算法对比结果分析

本文将 6 组算法在 Toys_and_Games、Kindle_Store、Movies_and_TV 和 Yelp 数据集上进行实验, 分析评分预测结果的 RMSE 的不同, 表 3 给出了各个算法的 RMSE 的比较结果。

表 3 不同推荐算法的 RMSE 比较

Table 3 Comparison of RMSE obtained by different recommendation algorithms

模型	RMSE			
	Toys_and_Games	Kindle_Store	Movies_and_TV	Yelp
PMF	1.3076	0.9914	1.2920	1.2413
NMF	1.0399	0.9023	1.1125	1.2678
SVD++	0.8860	0.7928	1.0447	1.1561
HFT	0.8925	0.7917	1.0291	1.2738
DeepCoNN	0.8873	0.7872	1.0126	1.1503
DLALSTM	0.8781	0.7821	0.9995	1.1357

由表 3 可知, 模型 DLALSTM 在 Toys_and_Games、Kindle_Store、Movies_and_TV 和 Yelp 4 个数据集上的推荐性能超过了所有对比方法:

(1) 考虑评论的方法 (HFT、DeepCoNN 和 DLALSTM) 通常比仅考虑评级矩阵的协同过滤模型 (例如 PMF、NMF 和 SVD++) 表现更好。这是因为评论信息是对评级的补充, 它可以缓解协同过滤算法存在的稀疏问题, 提高潜在因子模型的推荐性能。高质量的建模则进一步提高了用户偏好和项目特征表示的准确性, 从而得到了更好的评级预测结果。

(2) 利用深度学习技术 (DeepCoNN 和 DLALSTM) 的方法通常优于传统方法 (包括 HFT), 尽管 HFT 也对用户和项目的评论进行建模。这主要是因为: ① 文献 [25-26] 已经表明, 像 CNN、RNN 这样的神经网络相比基于 LDA^[10] 的主题模型在分析文本信息时可获得更好的性能; ② 深度学习能够以非线性方式对用户和项目进行建模^[27], 并且采用一些深度学习策略, 如在神经网络中采用 dropout 来防止过拟合可能会进一步提高模型性能。

(3) DLALSTM 始终优于所有对比方法。尽管评论信息在推荐过程中很重要, 但评论信息的不同使用方式也会导致不同的推荐结果。本文提出的基于双层注意力的模型利用评论的同时又分别考虑了每个评论的重要性以及每个词的重要性, 使得推荐性能得到了进一步的提高。

2.5.2 参数敏感性分析

本节探索了潜在因子数和 dropout 对模型性能的影响。图 2 和图 3 展示了算法在 Toys_and_Games 和 Kindle_Store 两个数据集上的实验结果。为了更好地展现 DLALSTM 模型的性能和改进效果, 本文将 DeepCoNN 的评分预测层更改为表现更好的神经预测层 (参见式 (13) - (14)) 来扩展 DeepCoNN, 并将其命名为 DeepCoNN++^[22]。DeepCoNN++ 将和 DeepCoNN 以及本文的模型 DLALSTM 进一步进行性能的比较。

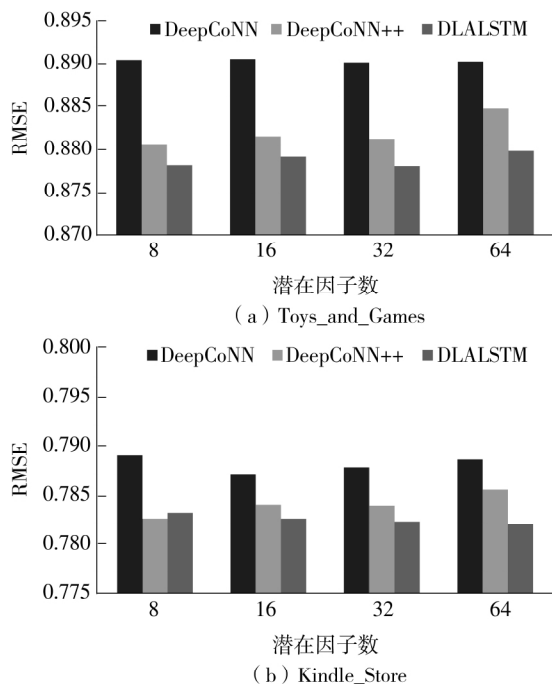


图2 模型在不同潜在因子数上的性能表现

Fig. 2 Performance of models on different latent factors

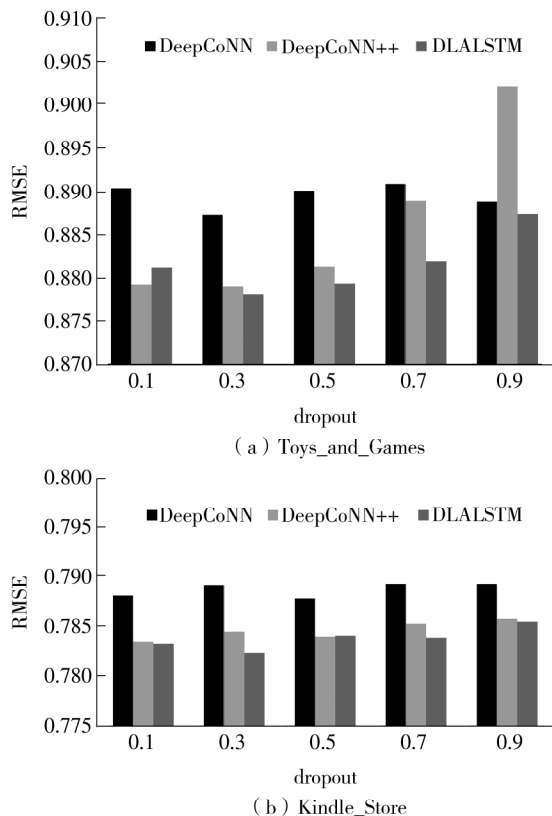


图3 dropout 对算法性能的影响

Fig. 3 Effect of dropout ratios on performance of algorithms

首先探索潜在因子数的影响。因为基于 MF 的模型 (PMF、NMF 和 SVD++) 以及 HFT 模型在推荐性能上和深度学习方法存在一定的差距, 本文只进一步对深度学习的方法进行比较。通过图 2 实验结果可以看出, 本文提出的模型以及对比模型的推荐性能在潜在因子数上的变化比较平稳, 在 Toys_and_Games 和 Kindle_Store 数据集上, 当预测因子数为 32 时本文提出的模型 (DLALSTM) 的 RMSE 取得了最优值, 比 DeepCoNN 和 DeepCoNN++ 模型的最优 RMSE 值分别降低了 1% 和 0.13%。

另外, 本文还通过实验讨论了 dropout 在基于深度学习模型上的影响。图 3 显示了 DeepCoNN 模型、DeepCoNN++ 模型以及本文的模型 DLALSTM 在不同 dropout 上的性能表现。从图 3 可以发现, 将 dropout 设置为适当的值时, 所有方法的性能都可以得到显著改善, 同时本文的模型在大部分情况下超过了所对比的方法, 当 dropout 为 0.3 时, DLALSTM 在 Toys_and_Games 和 Kindle_Store 数据集上取得最佳值, 超过了 DeepCoNN 和 DeepCoNN++ 模型的最优值。实验证明, 设置适当的 dropout 能够防止模型过拟合, 实现模型更好的性能。实验结果进一步表明, 在 dropout 的比较上, 推荐模型性

能在数据集 Toys_and_Games 上的变化比在 Kindle_Store 数据集上的变化更加明显。这是因为 Toys_and_Games 数据集相对较小, 更容易使模型出现过拟合情况, 而在较大的数据集上模型性能相对比较平稳。

2.5.3 注意力机制影响分析

本文分别在 Toys_and_Games 和 Kindle_Store 数据集上进行了注意力机制影响的实验, 3 个模型的 RMSE 比较如表 4 所示。在表 4 中, All 代表本文提出的应用了双层注意力机制的模型 DLALSTM; First attention 代表只使用了第 1 层注意力机制, 第 2 层用了恒等的权重; Second attention 代表在第 1 层用了恒等的权重, 第 2 层用了注意力机制。从表 4 可以看出, 应用两层注意力机制的评级预测的性能显著地超过了只用 1 层注意力机制的情况; 同时, 只用了 1 层注意力机制的模型无论是在第 1 层还是第 2 层使用都表现相当, 性能都低于用了双层注意力的模型性能。实验结果证明了本文的假设, 即不同的评论对用户偏好和项目特征具有不同的代表性, 同时每个词的作用也是不一样的, 设定不同的词的权重对用户和项目的建模是非常重要的。基于双层注意力机制的联合深度推荐模型可以很好地学习这种多样性, 从而可以提高推荐模型的性能。

表 4 注意力机制的影响
Table 4 Effect of attention mechanism

模型	RMSE	
	Toys_and_Games	Kindle_Store
All	0.8781	0.7821
First attention	0.8809	0.7839
Second attention	0.8806	0.7842

为了验证模型 DLALSTM 的第 1 层用户注意力层和第 1 层项目注意力层的有效性, 本文做了注意力可视化实验。表 5 提供了用户评论及其单词级别注意力可视化的例子, 以说明第 1 层注意力机制对用户偏好和项目特征识别的结果, 其中灰色较深的颜色表示单词具有较高的权重。

表 5 用户注意力和项目注意力在单词上的可视化
Table 5 Visualization of user attention and item attention over words

用户注意力	We love the ambiance and how cool this place is.
项目注意力	We love the ambiance and how cool this place is.

从表 5 可以看到 “love” 这个单词在用户注意力方面具有最高的注意力权重, 而 “cool” 这个单词在项目注意力方面具有最高的注意力权重。事实上, 单词 “love” 通常表达用户的偏好, 而单词

“cool” 则用于描述项目的特征。本文提出的模型不仅捕获了这些信息, 而且还给出了正确的预测。注意力的可视化表明本文提出的模型可以捕获用户的偏好和项目的特征。

为了验证模型 DLALSTM 的第 2 层用户注意力层和第 2 层项目注意力层的有效性, 本文做了评论注意力可视化实验。模型选择某个项目中的两个评论文本作为实验例子, 同时把这两个评论进行评论级别注意力权重可视化, 结果如表 6 所示。

表 6 DLALSTM 模型选择的高权重和低权重评论的例子
Table 6 Examples of the high-weight and low-weight reviews selected by DLALSTM

权重	评论
$\alpha_{iL} = 0.0432$	I like reading in my spare time, and I think this book is very suitable for me.
$\alpha_{iL} = 0.1923$	From beginning to end this book is a joy to read. Full of mystery, mayhem, and a bit of magic for good measure. Perfect flow with excellent writing and editing.

从表 6 中可以看出, 具有较高注意力权重的评论通常包含该项目更多的细节信息, 可以准确获得该项目的特征, 这对用户做出购买决策更有帮助。相比之下, 具有较低注意力的评论只包含用户的意见, 并没有该项目的细节, 这种评论对其他用户做出购买决策不具有说服力。

2.5.4 模型收敛性分析

在 Toys_and_Games 数据集上对 DLALSTM 模型进行了收敛性实验, 结果如图 4 所示。从图 4 可以看出, 当训练次数为 4 后损失趋于稳定, 因此可以看出模型 DLALSTM 是收敛的。

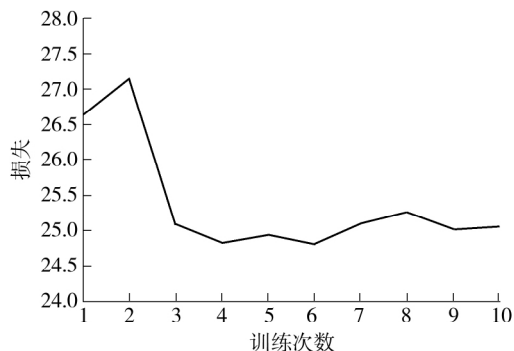


图 4 DLALSTM 模型在 Toys_and_Games 数据集上的验证收敛过程

Fig. 4 Convergence of DLALSTM model validation processes with Toys_and_Games dataset

3 结论

商品评论对用户做出购买决策起着非常重要的

作用,但用户很难从大量的评论中找到有用的信息。本文提出了一种基于双层注意力机制的深度推荐模型 DLALSTM,其采用基于双层注意力机制的双向 LSTM 来增强推荐性能,可以通过模型学习选择评论集中有用的评论以及评论中有用的词。相对于传统的基于评级的协同过滤方法,本文模型利用评论文本可以很大程度上缓解数据稀疏性问题。对来自亚马逊和 Yelp 4 个真实数据集的实验表明,在推荐性能方面,DLALSTM 始终优于所对比的常见推荐方法,因此,本文提出的模型将有助于构建一个更有效和可解释的推荐系统。模型 DLALSTM 建立在深度网络 BiLSTM 基础上,由于 BiLSTM 自身存在的效率较低的问题,利用模型 DLALSTM 进行推荐时较为耗时。在未来的工作中,我们将在保证推荐准确度的基础上,进一步提高模型运行的时间效率;同时也将继续探索利用基于注意力机制的 LSTM 来对用户长期的一般兴趣和近期的动态兴趣建模,以实现序列推荐。

参考文献:

- [1] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems [J]. *Computer*, 2009, 42(8): 30-37.
- [2] REN Z C, LIANG S S, LI P J, et al. Social collaborative viewpoint regression with explainable recommendations [C] // *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. Cambridge: ACM, 2017: 485-494.
- [3] LI P, WANG Z, REN Z, et al. Neural rating regression with abstractive tips generation for recommendation [C] // *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Shinjuku: ACM, 2017: 345-354.
- [4] WU L, QUAN C, LI C, et al. PARL: let strangers speak out what you like [C] // *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. Torino: ACM, 2018: 677-686.
- [5] HU G, ZHANG Y, YANG Q. Transfer meets hybrid: asynthetic approach for cross-domain collaborative filtering with text [C] // *Proceedings of the World Wide Web Conference*. San Francisco: ACM, 2019: 2822-2829.
- [6] FU W, PENG Z, WANG S, et al. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems [C] // *Proceedings of the AAAI Conference on Artificial Intelligence*. Honolulu: AAAI Press, 2019: 94-101.
- [7] WANG C, BLEI D M. Collaborative topic modeling for recommending scientific articles [C] // *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Diego: ACM, 2011: 448-456.
- [8] MCAULEY J J, LESKOVEC J. Hidden factors and hidden topics: understanding rating dimensions with review text [C] // *Proceedings of the 7th ACM Conference on Recommender Systems*. Hong Kong: ACM, 2013: 165-172.
- [9] TAN Y Z, ZHANG M, LIU Y Q, et al. Rating-boosted latent topics: understanding users and items with ratings and reviews [C] // *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. New York: IJCAI, 2016: 2640-2646.
- [10] BLEI D M, NG A Y, JORDAN M I. Latent dirichlet allocation [J]. *Journal of Machine Learning Research*, 2003, 3: 993-1022.
- [11] LEED D, SEUNG H S. Algorithms for non-negative matrix factorization [C] // *Proceedings of the 13th International Conference on Neural Information Processing Systems*. Denver: MIT Press, 2000: 556-562.
- [12] ZHENG L, NOROOZI V, YU P S. Joint deep modeling of users and items using reviews for recommendation [C] // *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. Cambridge: ACM, 2017: 425-434.
- [13] CATHERINE R, COHEN W. TransNets: learning to transform for recommendation [C] // *Proceedings of the Eleventh ACM Conference on Recommender Systems*. Como: ACM, 2017: 288-296.
- [14] RENDLE S. Factorization machines [C] // *Proceedings of the 10th IEEE International Conference on Data Mining*. Sydney: IEEE Computer Society, 2010: 995-1000.
- [15] ZHANG Y F, LAI G, ZHANG M, et al. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis [C] // *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Gold Coast: ACM, 2014: 83-92.
- [16] HE X N, CHEN T, KAN M Y, et al. Trirank: review-aware explainable recommendation by modeling aspects [C] // *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. Melbourne: ACM, 2015: 1661-1670.
- [17] GRAVES A, SCHMIDHUBER J. Framewise phoneme classification with bidirectional LSTM and other neural

- network architectures [J]. *Neural Networks*, 2005, 18(5/6): 602-610.
- [18] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality [C] // *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*. Lake Tahoe: NIPS, 2013: 3111-3119.
- [19] XIAO W Y, ZHAO H, PAN H J, et al. Beyond personalization: social content recommendation for creator equality and consumer satisfaction [C] // *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Anchorage: ACM, 2019: 235-245.
- [20] LI P J, WANG Z H, BING L D, et al. Persona-Aware tips generation [C] // *Proceedings of the World Wide Web Conference*. San Francisco: ACM, 2019: 1006-1016.
- [21] XIAO J, YE H, HE X N, et al. Attentional factorization machines: learning the weight of feature interactions via attention networks [C] // *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. Melbourne: IJCAI, 2017: 3119-3125.
- [22] CHEN C, ZHANG M, LIU Y, et al. Neural attentional rating regression with review-level explanations [C] // *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. Lyon: ACM, 2018: 1583-1592.
- [23] MNIH A, SALAKHUTDINOV R. Probabilistic matrix factorization [C] // *Proceedings of Advances in Neural Information Processing Systems*. Vancouver: Curran Associates, 2007: 1257-1264.
- [24] KOREN Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model [C] // *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas: ACM, 2008: 426-434.
- [25] KIM Y. Convolutional neural networks for sentence-classification [C] // *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Doha: ACL, 2014: 1746-1751.
- [26] LAI S W, XU L H, LIU K, et al. Recurrent convolutional neural networks for text classification [C] // *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Austin: AAAI Press, 2015: 2267-2273.
- [27] 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*. Perth: ACM, 2017: 173-182.

Joint Deep Recommendation Model Based on Double-Layer Attention Mechanism

LIU Huiting JI Qiang LIU Huimin ZHAO Peng

(School of Computer Science and Technology, Anhui University, Hefei 230601, Anhui, China)

Abstract: Many e-commerce websites keep a large amount of customer's reviews. The reviews was exploited by most recommendation systems by only considering their importance in the word-level rather than in the comment-level. The effectiveness of the recommendation model will be reduced by exclusively considering important words in the reviews and ignoring really useful reviews. Based on this, a joint deep recommendation model based on double-layer attention mechanism (DLALSTM) was proposed. First, DLALSTM uses bidirectional long short-term memory network (BiLSTM) to jointly model the customer and reviews from both word and customer levels, and aggregates the reviews representation and the customer/item representation by a double-layer attention mechanism. Then, the latent representation of customer and item learned from the reviews was incorporated into the customer preference and item feature obtained from the rating matrix to make rating prediction. DLALSTM was compared with the commonly used recommendation methods through experimental evaluation on different domain datasets of Yelp and Amazon. It finds that the performance of DLALSTM exceeds the state of art recommended methods. Meanwhile, the proposed model can alleviate the sparsity problem to some extent and has good interpretability.

Key words: attention mechanism; bidirectional long short-term memory network; recommendation system; deep learning