



基于矩阵分解和评论嵌入表示的推荐模型研究

张佳晖, 张 宇

(浙江理工大学信息学院, 杭州 310018)

摘 要: 协同过滤是目前最为成功的推荐技术之一,但它只利用了评分数据,忽视了大量可以利用的用户评论。针对该问题提出了一种基于概率图的深度神经网络推荐模型,即共享表示模型(Shared representation model, SRM),并在SRM的基础上提出一种基于多任务学习思想的改进模型,即隐因子共同学习模型(Joint learning model with latent factor, LF-JLM)。LF-JLM结合了基于矩阵分解的隐因子推荐算法和doc2vec语言模型,它在doc2vec和隐因子模型的映射层使用共享的用户、商品、评论文档的向量表示,因此能够学习到具有跨任务不变性的底层特征。在Amazon数据集上对提出的两种模型与作为基线的隐因子模型和HFT模型进行了对比实验,实验结果表明:LF-JLM能够有效地抽取评论中隐含的语义信息;与隐因子模型和HFT模型相比,该算法评分预测的均方误差分别减小了7.85%和1.19%。

关键词: 推荐系统;协同过滤;深度神经网络;多任务学习;词嵌入

中图分类号: TP181

文献标志码: A

文章编号: 1673-3851(2019)01-0079-13

Research on recommendation model based on matrix decomposition and reviews embedding

ZHANG Jiahui, ZHANG Yu

(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: Collaborative Filtering is one of the most successful recommendation technologies so far, but it only considers the rating information while ignoring a great amount of usable user comments. In order to solve this problem, a deep neural network recommendation model based on probabilistic graph was proposed, i.e. Shared representation model (SRM). And on the basis of SRM, an improved model based on multitask learning was proposed, i.e. Joint learning model with latent factor (LF-JLM). LF-JLM combined latent factor recommendation algorithm based on matrix decomposition and the doc2vec language model. By the method of introducing the shared vector representation of the user, item and comment document in the mapping layer of the doc2vec and the LFM, LF-JLM could learn the underlying features with the cross-task invariance. On the Amazon dataset, the proposed two models were compared with the LFM and the HFT model as the baseline. The experimental results showed that LF-JLM could effectively extract implied semantic information in the comments. Compared with LFM and the HFT (Hidden Factors as Topics) model, the MSE of rating of this algorithm decreased by 7.85% and 1.19% respectively.

Key words: recommendation system; collaborative filtering; deep neural network; multitask learning; word embedding

0 引言

随着互联网规模的迅速增大,信息过载问题愈发严峻。推荐系统可以针对不同用户提供个性化的推荐,被视为继搜索引擎后解决信息过载问题最有前景的方法^[1]。迄今为止,推荐系统已经在电子商务、音乐推荐、电影推荐等领域取得了巨大成功^[2]。然而,传统的协同过滤推荐方法存在诸多弊端,造成这些弊端的最主要原因是用于推荐的数据过于稀疏^[3]。协同过滤方法通过分析用户对产品的评分数据,寻找与目标用户相似的用户来做出推荐,但现实中各类产品的人气一般符合长尾分布^[4],即多数用户只会评价或购买少量产品,多数产品仅有很少人评价或购买。这意味着由用户对产品的评分组成的评分矩阵是稀疏矩阵。信息不足导致传统的协同过滤方法在这种情况下的推荐效果不尽如人意。

为了解决这些问题,一系列新的推荐技术不仅应用了用户对商品的评分信息,还试图结合其他辅助信息进行推荐,如用户注册信息、用户的社交网站信息、商品的描述文档信息等^[3, 5]。在这些辅助信息之中,用户评论文本是最为丰富且有待挖掘的信息之一。许多电子商务网站、电影书籍音乐推荐网站中有海量的用户评论,包含着大量非常有价值的信息,但由于这些评论在语法上的随意性和灵活性、语义上的模糊性和歧义性等原因,难以被推荐系统利用。如果人工逐条阅读这些评论进行归纳决策,则要耗费大量的时间和精力。

因此,为了有效利用用户评论文本数据,改进推荐系统的推荐效果,本文提出了一种采用深度神经网络、融合了评论文本数据和评分数据的推荐模型。首先从三个直观的假设出发,对评分和评论文本的生成过程建立概率图模型,然后证明了存在一个与概率图模型等价的深度神经网络模型,即共享表示模型;接着用隐因子模型和多任务学习的思想对共享表示模型进行改进,得到本文最终提出的模型,即隐因子共同学习模型;最后采用真实数据对隐因子共同学习模型与 HFT 模型、隐因子模型进行比较,以评价提出的推荐模型的推荐效果。

1 相关研究

1.1 结合用户评论推荐的研究现状

在结合用户评论来改进传统推荐方法的研究领域中,具有开创性的研究是 McAuley 等^[6]提出的 Hidden factors as topics(HFT)模型。HFT 用主题建

模算法隐狄利克雷分布(Latent dirichlet allocation, LDA)来建模评论文本。该模型首先聚合某种商品下的所有评论文本,对其进行主题建模后得到一个表示这种商品的概率分布向量(向量的各个维度对应文本的各个主题);然后用低秩矩阵分解,即 Koren 等^[7]提出的隐因子模型,分解评分矩阵,得到表示商品的隐因子;最后假设求解 LDA 得到的概率分布向量和矩阵分解得到的隐因子之间存在一个指数函数形式的映射函数,从而可以同时训练两个模型。实验证明,相比于仅使用评分数据,HFT 模型大大提升了推荐效果。

但是,HFT 模型有两个较大的缺陷:a)从矩阵分解得到的商品实值向量映射到 LDA 建模得到的商品概率向量的函数形式是人为定义的,函数中仅含有一个可调整参数,不够灵活,也缺乏理论依据。b)为了对评论文本建模,HFT 模型将同类商品下的所有评论文本聚合起来,从而丢失了这些评论文本各自评论者的对应信息。类似地,如果将同用户的所有评论聚合后进行主题建模,又会丢失关于商品的对应信息。HFT 在对评论文本进行建模时,无法同时兼顾商品因素和用户因素,只能考虑两者中的一个。

此后,在 HFT 的基础上,出现了诸多类似的用概率图模型来建模评论和评分生成过程的研究,也出现了结合深度学习和概率图模型的研究^[8-15]。其中,Bao 等^[8]提出的 TopicMF (Topic matrix factorization)模型是对 HFT 的改进之一。topicMF 在进行评分矩阵分解时,假设用户评分的方差和其对应评论的质量有关。topicMF 用矩阵分解处理评分,得到用户向量和物品向量;又用非负矩阵分解分解词共现矩阵,得到每条评论对应的评论-主题分布,并进一步认为这个分布的概率向量由用户向量和商品向量通过一个人工给定的映射函数共同决定。Ling 等^[9]提出的 RMR(Ratings meet reviews)模型也是对 HFT 的改进。RMR 不仅用 LDA 建模文本,而且将评分生成过程也用 LDA 来建模。因为在 RMR 模型中,利用评论文本数据求得的商品向量和利用评分数据求得的商品向量是同一个,从而解决了两者之间的对应关系问题。彭敏等^[10]提出了基于情感分析的推荐算法 SACF (Sentiment analysis for collaborative filtering)。SACF 利用 LDA 挖掘商品的潜在属性面,通过用户在各个属性面上的情感偏好计算用户相似度,从而构建推荐模型,找到最相似的 N 个用户进行推荐。实验表明属性特征和情感

词的提取对 SACF 最终的推荐精度影响较大。王建等^[11]提出了一种基于用户改进的 LDA 算法,即 LDA-RR(LDA based on ratings and reviews)算法。LDA-RR 算法假设每个用户下隐含着主题分布,主题下隐含着物品分布,同时评论词语的分布由主题和物品共同决定,根据潜在主题分布挖掘用户兴趣进行推荐。实验结果表明改进的算法有效提高了推荐质量。此外,还有 Wu 等^[12]提出的 FLAME(Factorized latent aspect model)模型等。

除以上利用概率图建模用户评分及评论的一类推荐模型外,还出现了利用深度学习来求解评论文本的深层表示并融入概率图的一类推荐模型。例如,Wang 等^[13]提出了基于堆叠降噪自编码器的 CDL(Collaborative deep learning)模型。CDL 模型没有采用 LDA 建模评论文本,而是将各个商品下的评论文本作为 SDAE 的输入以学习压缩编码;然后将编码作为对评分矩阵进行概率矩阵分解的概率图中的一个已知的观测结点,通过最大后验估计来求解这个模型。Kim 等^[14]提出了基于卷积神经网络的 ConvMF(Convolutional matrix factorization)模型。ConvMF 模型将文本输入一个具有线性映射层和卷积层的神经网络,输出为向量,该向量加上期望为 0 的高斯分布的偏移量后,对应为概率图中代表商品的结点,该概率图用概率矩阵分解方法对评分矩阵进行分解。模型的优化目标同样是最大化整个概率图的后验概率。Almahairi 等^[15]提出了分别基于词袋模型和循环神经网络的 BoWLF(Bag-of-words regularized latent factor model)和 LMLF(Language model regularized latent factor model)模型。前者由隐因子模型得到的商品向量,经过 Softmax 函数变换得到对应商品下评论的词频;后者应用了 Hochreiter 等^[16]提出的长短期记忆网络,由商品向量作为 LSTM 网络的参数,得到该商品下的评论集合的“预测下一个词词频的”语言模型。LMLF 的结构更复杂,但实验中 BoWLF 的表现更好。实验证明以上模型相比传统的协同过滤,都大幅提高了推荐效果。

但是,这些模型具有共同缺点:其一是对用户给商品撰写评论的生成过程有较多限于特定问题领域的、基于作者经验的启发式的假设,同时生成过程较为复杂,因而影响了这些模型的可解释性和泛化能力。其二是为了能将评论和评分数据的学习结合到同一个模型中,评论文本和评分矩阵都用概率生成过程建模,而无法结合已有的较为成熟的文本分

析技术和推荐技术。本文提出的基于深度神经网络的模型试图在简化基本假设的基础上,不用复杂的生成过程来建模评论文本信息,而是结合已有的较为成熟的 doc2vec 模型,潜在地将评论信息融入推荐,来达到结合评论推荐的目的。

1.2 与本文理论基础相关的研究

1.2.1 隐因子模型

2009 年,Koren 等^[7]提出了隐因子模型 LFM,夺得了 Netflix 推荐系统比赛(Netflix prize competition)的冠军。隐因子模型作为一种经典的协同过滤推荐算法,由于理论清晰,实现简单,推荐结果良好,因此通常作为推荐系统比较的基准模型。

该模型假设用户 i 对商品 j 的评分的预测 $rec(i, j)$ 可以由式(1)产生:

$$rec(i, j) = \alpha + \beta_i + \beta_j + \gamma_i \cdot \gamma_j \quad (1)$$

其中: α 为整体偏置项, β_i 和 β_j 分别为用户偏置和商品偏置, γ_i 和 γ_j 分别为 K 维的用户因子和商品因子。直观上, γ_i 向量的各个维度代表了用户 i 对各个商品属性的偏好,而 γ_j 的各个维度代表了商品 j 在各种属性上的权重,两者的内积加上用户偏置、商品偏置和整体偏置,得到用户对商品的评分。给定训练集 Γ ,令 $\Theta = \{\alpha, \beta_i, \beta_j, \gamma_i, \gamma_j\}$,可以最小化均方误差(Mean square error, MSE)求得参数:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{x_{ij} \in \Gamma} (x_{ij} - rec(i, j))^2 + \lambda \|\gamma\|_2^2 \quad (2)$$

其中: N 为训练集中的总评分数, x_{ij} 为第 i 个用户对第 j 个商品的评分, λ 为正则项系数。式(2)可以用交替最小二乘法或梯度下降方法求解。

隐因子模型可以视为是在将原评分矩阵分解为范数和维度约束下的用户向量映射矩阵和商品向量映射矩阵的乘积,因此基于该模型的方法及其改进方法常被称为基于矩阵分解的推荐方法。

1.2.2 word2vec 模型

Mikolov 等^[17]提出了一种学习词嵌入的深度神经网络模型,即词向量模型 word2vec。该模型利用一个词的上下文信息预测这个词,并试图在这个过程中学习到编码了该词语境信息的向量表示,每个词被映射为一个特定的向量(即词嵌入),字典中所有词对应的向量组成了一个映射矩阵。Mikolov 等^[17]提出了两种结构的 word2vec 模型:Continuous-bag-of-words(CBOW)和 Skip-gram。CBOW 模型通过特定大小窗口中的其他词来预测窗口中心词。Skip-gram 则与此相反,通过一个单

独的词来预测它的上下文窗口中的词。这两种结构在实验中取得了近似的结果,本文中主要讨论 CBOW 模型。

给定训练集的词序列: $w_1, w_2, w_3, \dots, w_T$, T 为训练集的总词数, CBOW 模型的训练目标是最大化训练文本的对数似然概率(即最小化交叉熵)。对数似然概率可用式(3)表示:

$$\frac{1}{T} \sum_{t=k}^{T-k} \log p(w_t | w_{t-k}, \dots, w_{t+k}) \quad (3)$$

其中: k 为调节上下文窗口的变量, $2k+1$ 为窗口的大小; p 为对窗口中心词 w_t 的概率预测函数, 可以用多类分类算法计算。模型利用 softmax 计算 p , 可用式(4)表示:

$$p(w_t | w_{t-k}, \dots, w_{t+k}) = \frac{e^{y_{wt}}}{\sum_i e^{y_i}} \quad (4)$$

其中: y_i 为未归一化的输出词的对数概率, 可以通过式(5)求得:

$$\mathbf{y} = \mathbf{b} + \mathbf{U}h(w_{t-k}, \dots, w_{t+k}; \mathbf{W}) \quad (5)$$

其中: \mathbf{U} 和 \mathbf{b} 为 softmax 函数的参数, \mathbf{W} 为词映射矩阵, \mathbf{y} 为词典中所有词对应的 y_i 组成的向量, h 函数的作用是从映射矩阵 \mathbf{W} 中抽出窗口中各个词对应的词向量, 并将其连接起来(或取均值)。

使用随机梯度下降和反向传播可以求解式(3), 可以求得词映射矩阵 \mathbf{W} , 即得到字典中每个词对应的词向量。实践中因为 softmax 分类器的输出类别过多(取决于字典的大小, 可能大至百万量级), 在反向传播计算梯度时计算量很大, Mikolov 等^[17] 提出了分层 softmax 和负抽样的方法来减少计算量, 并取得了较好的结果。本文提出的模型在实验中也采用了负抽样的技巧来简化 softmax 函数的计算。

1.2.3 doc2vec 模型

Le 等^[18] 基于 word2vec 的思想, 提出一种可以学习段落或文档的嵌入表示的神经网络模型。该模型的主要思路是: 语料中每个文档的序号通过文档映射矩阵被映射为一个独特的文档向量, 在预测该文档内的上下文窗口中的窗口中心词的时候, 这个文档向量和原本 CBOW 模型中上下文窗口中的词向量连接起来(或取平均), 作为输入来预测目标中心词。训练这个神经网络便可得到文档映射矩阵, 从而得到文档的嵌入表示。用类似的方法也可以得到段落的嵌入表示。

doc2vec 模型与 CBOW 模型非常相似, 唯一的区别是在预测任务中增加了文档的序号作为输入, 并增加了从文档的序号映射到文档向量的映射矩

阵。求得的文档向量直观上被认为是以某种方式编码了对应文档的主题和上下文信息。在情感分析等实验中, doc2vec 模型取得了比支持向量机和递归神经网络更好的结果^[18], 验证了该模型的有效性。

2 概率假设

为了将评论信息融入到推荐系统, 本文假设每个用户和商品都存在低维空间的嵌入表示, 它们共同决定了用户对商品的评论文本以及评分值。用户和商品的嵌入表示是隐含的、无法观测到的, 而评论和评分是含有噪声的、可观测到的。评论和评分各自由不同的生成过程得到。本文希望学习到的嵌入表示和生成过程满足如下直观假设:

a) 内在表示, 即用户和商品的低维空间嵌入表示, 应尽可能简单。

b) 生成过程, 即从嵌入表示到评论、评分的生成过程, 应尽可能简单。

c) 噪声和误差, 即预测的评论和真实的评论、预测的评分和真实的评分之间的差距, 应尽可能小。

以上 3 个条件是本文提出的假设, 也是约束条件。于是问题为已知评论和评分数据的情况下求嵌入表示和生成过程, 并满足以上 3 个约束条件。从问题的定义可以看出, 它和多传感器信息融合、多感觉整合等领域的研究有较大的相似性。贝叶斯估计在这些领域内被视为重要的方法^[19-20], 因此将以上约束条件下的评论和评分生成过程用一个简单的贝叶斯概率图模型表示, 如图 1 所示。

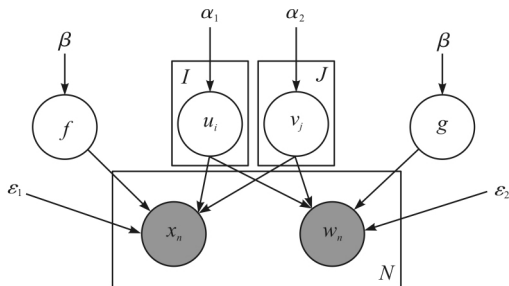


图1 评论文本和评分生成过程的概率图

用 K 维空间中的随机向量 $\mathbf{u}_i \in \mathbf{R}^K$ 代表序号为 i 的用户的嵌入表示, 称为用户向量。随机变量 \mathbf{u}_i 从期望为 0、方差为 α_1^2 的高斯分布中抽取, $\mathbf{u}_i \sim N(0, \alpha_1^2 \mathbf{I}_K)$, $i=1, 2, \dots, I$, I 为总用户数, \mathbf{I}_K 为 K 阶的单位矩阵。类似地, 对于每个商品, 商品向量 $\mathbf{v}_j \in \mathbf{R}^K$ 服从参数为 α_2 的先验分布, 即 $\mathbf{v}_j \sim N(0, \alpha_2^2 \mathbf{I}_K)$, $j=1, 2, \dots, J$, J 为总商品数。

再假设存在函数 f , 它将商品向量和用户向量对 $(\mathbf{u}_i, \mathbf{v}_j)$ 近似地映射到用户 i 对商品 j 的评分 $x_{i,j}$, 即 $x_{i,j} \sim N(f(\mathbf{u}_i, \mathbf{v}_j), \epsilon_1^2)$, ϵ_1 为分布的方差, 函数 f 服从参数为 β_1 的某特定形式的先验分布(该分布的支撑集为从 $(\mathbf{u}_i, \mathbf{v}_j)$ 到 $x_{i,j}$ 的函数空间)。类似地有函数 g , 它将 $(\mathbf{u}_i, \mathbf{v}_j)$ 映射到该用户对商品的评论的某种嵌入表示 $\mathbf{w}_{i,j}$, $\mathbf{w}_{i,j} \sim N(g(\mathbf{u}_i, \mathbf{v}_j), \epsilon_2^2 \mathbf{I}_w)$, $\epsilon_2^2 \mathbf{I}_w$ 为分布的方差, g 的先验分布的参数为 β_2 。

若用 i_n 代表第 n 条评论对应的用户的序号, j_n 代表第 n 条评论对应的商品的序号, 则评论集中第 n 条评论的嵌入表示为 \mathbf{w}_{i_n, j_n} , 简记为 \mathbf{w}_n 。第 n 条评论对应的评分为 x_{i_n, j_n} , 简记为 x_n 。记 $\mathbf{x} = (x_n)_{n=1}^N$, $\mathbf{W} = (\mathbf{w}_n)_{n=1}^N$, $\mathbf{U} = (\mathbf{u}_i)_{i=1}^I$, $\mathbf{V} = (\mathbf{v}_j)_{j=1}^J$ 。再令 $\Theta = \{f, g, \mathbf{U}, \mathbf{V}\}$, $\mathbf{A} = \{\alpha_1, \alpha_2, \beta_1, \beta_2, \epsilon_1, \epsilon_2\}$, $\mathbf{X} = \{\mathbf{x}, \mathbf{W}\}$, 则参数 Θ 的后验分布可用式(6)表示为:

$$p(\Theta | \mathbf{X}, \mathbf{A}) = \frac{p(\mathbf{X} | \Theta, \mathbf{A}) \cdot p(\Theta | \mathbf{A})}{p(\mathbf{X} | \mathbf{A})} \quad (6)$$

假设函数 f 和 g 可以用含有 t 个参数的前馈神经网络拟合, 并进一步假设神经网络的权重参数分别为向量 θ_1 和 θ_2 , 且 $\theta_1 \sim N(0, \beta_1^2 \mathbf{I}_t)$, $\theta_2 \sim N(0, \beta_2^2 \mathbf{I}_t)$, 代入式(6)求得 Θ 的最大后验估计为:

$$\begin{aligned} \Theta_{\text{MLE}} = \operatorname{argmin}_{\Theta} \frac{1}{\epsilon_1^2} \sum_n [x_n - f_{\theta_1}(\mathbf{u}_{i_n}, \mathbf{v}_{j_n})]^2 + \\ \frac{1}{\epsilon_2^2} \sum_n \|\mathbf{w}_n - g_{\theta_2}(\mathbf{u}_{i_n}, \mathbf{v}_{j_n})\|^2 + \\ \frac{1}{\alpha_1^2} \|\mathbf{U}\|_F^2 + \frac{1}{\alpha_2^2} \|\mathbf{V}\|_F^2 + \\ \frac{1}{\beta_1^2} \|\theta_1\|_2^2 + \frac{1}{\beta_2^2} \|\theta_2\|_2^2 \end{aligned} \quad (7)$$

其中: $\|\mathbf{U}\|$ 、 $\|\mathbf{V}\|$ 为矩阵 \mathbf{U} 和 \mathbf{V} 的 Frobenius 范数。

从式(7)可以看出, 该模型的目标函数是以下三项的加权和: 用于约束用户和商品嵌入表示复杂程度的正则项, 用于约束神经网络复杂程度的正则项, 以及评分和评论的均方误差。这三项分别对应上文提出的 3 个约束。式(7)关于 Θ 为凸, 于是只要给出 \mathbf{w}_n 的表示方法就可以用梯度下降求解。

3 模型推导

3.1 共享表示模型

由于 doc2vec 模型可以得到对段落的向量表示, 结合式(7), 一种直接的方法是对每条评论, 训练 doc2vec 来得到评论向量 \mathbf{w}_n 并代入式(7)。回顾式(7)以及第 2 节中对概率图模型的相关讨论, 因为假

设函数 f 和 g 可以用前馈神经网络拟合, 又假设神经网络的参数服从的先验分布是期望为 0、方差固定的高斯分布, 于是求解式(7)等价于训练如图 2 所示的前馈神经网络, 命名为共享表示模型(Shared representation model, SRM)。

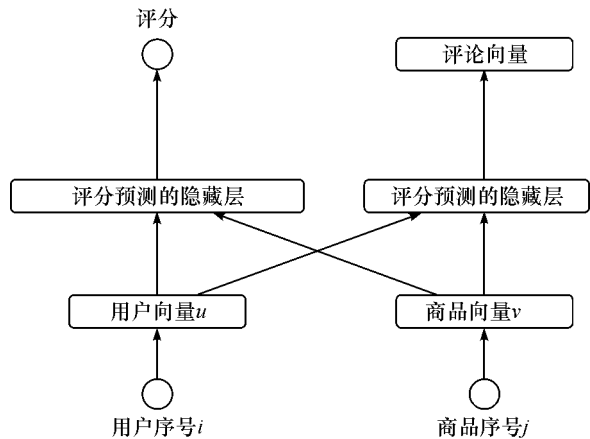


图 2 共享表示模型

图 2 左侧部分是预测评分的前馈神经网络回归模型, 用户序号和商品序号通过映射层, 映射为向量后, 两者连接起来共同作为隐藏层的输入, 用式(8)表示:

$$\mathbf{y}_1 = \mathbf{A}_1 h'(\mathbf{i}, \mathbf{j}; \mathbf{U}, \mathbf{V}) + \mathbf{b}_1 \quad (8)$$

其中: h' 函数输出 \mathbf{U} 矩阵第 i 列 \mathbf{u}_i 和 \mathbf{V} 矩阵第 j 列 \mathbf{v}_j 的连接; \mathbf{A}_1 和 \mathbf{b}_1 分别为从映射层连接到评分预测隐藏层的权重矩阵和偏置项, 隐藏层的输出 \mathbf{y}_1 经过激活函数的非线性变换后, 再利用线性回归进行评分拟合, 用式(9)表示:

$$\hat{x} = \mathbf{A}_2 \tanh(\mathbf{y}_1) + \mathbf{b}_2 \quad (9)$$

其中: \mathbf{A}_2 和 \mathbf{b}_2 是从隐藏层到评分输出的权重和偏置, \hat{x} 是预测的评分值。

与式(8)和式(9)类似, 对右侧的预测评论向量的回归模型, 同样有式(10)和式(11):

$$\mathbf{y}_2 = \mathbf{A}_3 h(\mathbf{i}, \mathbf{j}; \mathbf{U}, \mathbf{V}) + \mathbf{b}_3 \quad (10)$$

$$\hat{\mathbf{w}} = \mathbf{A}_4 \tanh(\mathbf{y}_2) + \mathbf{b}_4 \quad (11)$$

令 $\Theta = \{\mathbf{A}_i, \mathbf{b}_i \mid i=1, \dots, 4\} \cup \{\mathbf{U}, \mathbf{V}\}$, 整个模型的损失函数即为式(12):

$$\begin{aligned} \text{loss}(\Theta) = & \frac{\lambda_1}{2N} \sum_{n=1}^N (x_n - \hat{x}_n)^2 + \\ & \frac{\lambda_2}{2N} \sum_{n=1}^N \|\mathbf{w}_n - \hat{\mathbf{w}}_n\|^2 + \\ & \lambda_3 (\|\mathbf{U}\|) + \lambda_4 (\|\mathbf{V}\|) + \\ & \lambda_5 (\|\mathbf{A}_1\| + \|\mathbf{A}_2\|) + \lambda_6 (\|\mathbf{A}_3\| + \|\mathbf{A}_4\|) \end{aligned} \quad (12)$$

比较式(7)和式(12), 如果令式(12)中 $\lambda_1 =$

$2N/\epsilon_1^2, \lambda_2 = 2N/\epsilon_2^2, \lambda_3 = 1/\alpha_1^2, \lambda_4 = 1/\alpha_2^2, \lambda_5 = 1/\beta_1^2, \lambda_6 = 1/\beta_2^2$, 则两式是等价的。因此,图2中的神经网络模型具有概率解释:模型因为增加了评论向量预测任务,并在两个预测任务中使用同一个用户和商品映射矩阵,从而改变用户向量和商品向量的后验概率,即利用评论向量预测任务来对评分预测任务进行正则化。

因为预测评分和评论的神经网络共享了用户映射矩阵和商品映射矩阵作为参数,所以本文将该模型命名为共享表示模型。训练该模型前,先用 doc2vec 模型训练评论语料得到每条评论对应的评论向量 w_n 作为监督,使用标准的反向传播(Back propagation)算法可以求出用户映射矩阵、商品映射矩阵以及网络的权值。其中评论向量只在训练时被用到,而预测时没有使用任何评论数据,仅使用左侧预测评分的子网络。

3.2 隐因子共同学习模型

3.1 小节提出的共享表示模型直接源于式(7),较为简单。该模型实验中效果一般,主要原因是:

a) 该模型是三层前馈神经网络,在实验中发现其参数较多,容易过拟合,且超参数如隐藏层单元数也难以确定,因此难以训练。

b) 通过在概率图中增加代表评论文本的观测结点,模型改变了用户映射矩阵和商品映射矩阵的后验分布,使模型倾向于求得能够同时完成预测评论向量和预测评分两个任务的解。但由于作为训练目标之一的评论向量是通过 doc2vec 模型预先独立地训练出来的,和推荐任务的相关性不大,利用它来改进推荐效果比较困难。

本文提出了两种相应的解决方式来改进共享表示模型。

a) 改进一:针对模型参数太多容易过拟合的问题,本文的解决方式是采用 1.2.1 节中所述的隐因子模型来代替评分预测部分的三层前馈神经网络。隐因子模型的参数较少,具有更多合理的先验信息,更易于训练,且不容易过拟合。

b) 改进二:针对评论向量和任务相关性不大的问题,一个自然想法是让模型在进行评分预测任务的同时学习评论文档的嵌入向量。即不事先训练 doc2vec 模型以得到评论向量作为预测目标,而是将训练评论向量的过程融合进推荐模型,同时学习评论向量和用户、商品的嵌入表示,作为两个预测任务的共享表示。于是模型的学习目标应该是同时预测

评分值以及评论中的上下文窗口的中心词,完成第二个目标的时候应能够在中间层学习到评论文档的嵌入向量。评论向量和评分一样,应该由用户向量和商品向量共同决定。

以上方法采用的是多任务学习^[21]的思想。多任务学习是机器学习中的一种方法,即把多个相关的任务放在一起学习。这种方法非常适用于需要同时求解多个目标函数的情形。即使只有一个目标函数,对辅助任务的学习仍然能大幅改善主任务的性能。Caruana 等^[21]的研究表明,在多任务学习场景中,归纳偏置是由辅助任务来提供的,这会导致模型更倾向于那些可以同时解释多个任务的解,使其不容易在某一个特定的任务上过拟合,从而能够提高模型泛化能力。

多任务学习有两个特点:一是基于共享表示,二是同时学习多个任务之间有相关性。在本文对融合评论文本的推荐模型研究中,需要同时完成评论文本建模和评分预测这两个任务,虽然模型的主任务只有后者,但是加入前者可以改善后者的完成效果,这两个任务是高度相关的。因此,本文研究的场景非常适合多任务学习。

本文采用了基于硬参数共享的多任务学习方法。参数的硬共享机制是多任务学习中最常见的一种方式,即在各个任务之间共享隐藏层,同时保留部分任务间不同的隐藏层来学习各个任务的不同之处。例如,Yu 等^[22]的研究说明了词性标注,命名实体识别等自然语言处理中的基本任务可以被作为辅助任务使用参数的硬共享进行有监督学习,从而可以获得更好的语句嵌入表示。在本文研究中,共同学习的两个任务共享的硬参数包括用户的向量表示、商品的向量表示,评论文本中词的嵌入和评论文档嵌入。

本文将经过了两种改进得到的模型,即本文最终提出的模型,命名为隐因子共同学习模型(Joint learning model with latent factor, LF-JLM)。LF-JLM 的整体神经网络结构设计如图3所示。

在图3中,模型的左侧部分是输入用户序号和商品序号来预测评分的前馈神经网络回归模型,目标是 minimized 评分的均方误差;右侧是输入用户序号、商品序号以及评论中指定大小的上下文窗口中的词来预测窗口中心词的分类模型,目标是最大化目标词的似然概率。两个部分通过用户向量和商品向量到文档向量的一个非线性变换相联系。

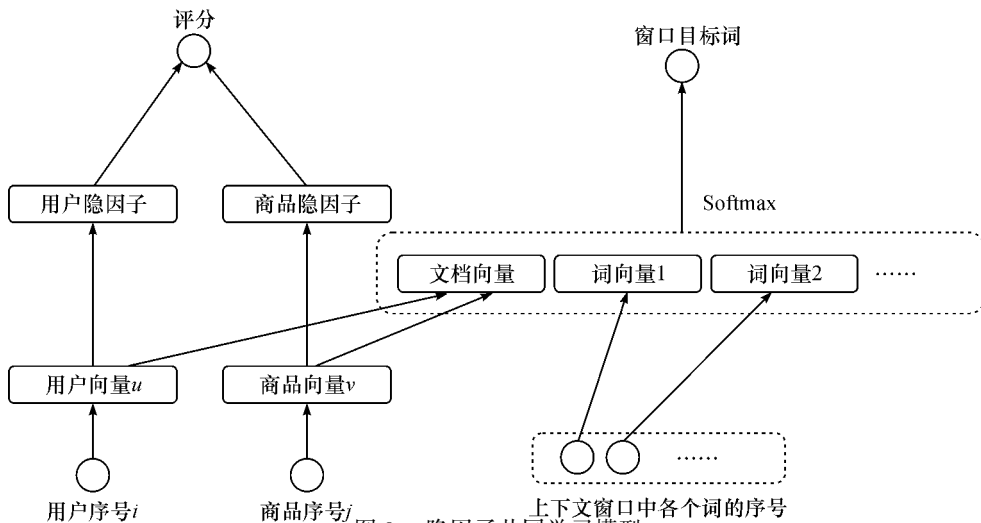


图 3 隐因子共同学习模型

在该网络的训练过程中,每条评论的每个上下文窗口都对应一个训练样本。其中用户序号 i 和商品序号 j 通过用户映射矩阵和商品映射矩阵分别映射为用户向量和商品向量。这两者经过线性变换各自作为隐藏层的输入,然后经过非线性激活函数 ReLU(Rectified linear units),输出为用户和商品的隐因子 $\mathbf{l}f_u$ 和 $\mathbf{l}f_v$:

$$\begin{cases} \mathbf{l}f_u = \text{ReLU}(\mathbf{A}'_1 h(i; \mathbf{U}) + \mathbf{b}'_1) \\ \mathbf{l}f_v = \text{ReLU}(\mathbf{A}'_2 h(j; \mathbf{V}) + \mathbf{b}'_2) \end{cases} \quad (13)$$

其中: \mathbf{A}'_1 和 \mathbf{A}'_2 分别为从用户映射层和商品映射层连接到用户隐因子层和商品隐因子层的权重矩阵, \mathbf{b}'_1 和 \mathbf{b}'_2 为相应的偏置。

按照隐因子模型的方法,计算两者的内积并加上偏置项,得到对应的评分 \hat{x} :

$$\hat{x} = [\mathbf{l}f_u, 1] \cdot [1, \mathbf{l}f_v] + \mathbf{b}'_3 \quad (14)$$

其中:用户隐因子 $\mathbf{l}f_u$ 的第一维隐含地代表用户偏置,商品隐因子 $\mathbf{l}f_v$ 的最后一维代表商品偏置, \mathbf{b}'_3 代表全局偏置。因此在(14)式计算中,在 $\mathbf{l}f_u$ 向量的末尾增加一位值为 1 的维度, $\mathbf{l}f_v$ 向量的开头增加一位值为 1 的维度,两者内积加上 \mathbf{b}'_3 得到评分预测值,和(1)式中由隐因子计算评分的方法相同。

另一方面,假设评论向量 ω 可以通过用户向量和商品向量连接后经过 ReLU 变换得到:

$$\omega = \text{ReLU}(\mathbf{A}'_3 \cdot h'(i, j; \mathbf{U}, \mathbf{V}) + \mathbf{b}'_4) \quad (15)$$

其中: \mathbf{A}'_3 为从用户向量和商品向量的连接层到文档向量层的权重矩阵, \mathbf{b}'_4 为相应的偏置。评论的上下文窗口中的各个词的序号通过词映射矩阵映射为词向量后,这些词向量和评论向量 ω 连接起来作为 softmax 分类器的输入,输出为窗口中心词的概率分布:

$$\mathbf{y} = \mathbf{A}'_4 \cdot h''(\omega_{t-k}, \dots, \omega_{t+k}; \mathbf{W}, \omega) + \mathbf{b}_5 \quad (16)$$

$$p(\omega_t | \omega_{t-k}, \dots, \omega_{t+k}) = \frac{e^{y_{\omega_t}}}{\sum_i e^{y_i}} \quad (17)$$

式(16)中: \mathbf{A}'_4 为输出层的权重矩阵, \mathbf{b}_5 为相应的偏置。 h'' 函数将上下文窗口中的各个词通过词映射矩阵 \mathbf{W} 映射为词向量,并返回它们和评论向量 ω 连接起来的结果。令

$\Theta = \{\mathbf{A}_i, \mathbf{b}_j | i=1, \dots, 4; j=1, \dots, 5\} \cup \{\mathbf{U}, \mathbf{V}, \mathbf{W}\}$, 整个模型的损失函数即为:

$$\begin{aligned} \text{loss}(\Theta) = & \frac{\lambda_1}{T} \sum_{t=k}^{T-k} \log p(\omega_t | \omega_{t-k}, \dots, \omega_{t+k}) + \\ & \frac{\lambda_2}{2T} \sum_{t=k}^{T-k} (x_t - \hat{x}_t)^2 + \lambda_3 (\|\mathbf{U}\|) + \\ & \lambda_4 (\|\mathbf{V}\|) + \lambda_5 (\|\mathbf{W}\|) + \\ & \lambda_6 (\|\mathbf{A}_1\| + \|\mathbf{A}_2\|) + \lambda_7 (\|\mathbf{A}_3\|) + \\ & \lambda_8 (\|\mathbf{A}_4\|) \end{aligned} \quad (18)$$

比较式(18)和式(7),可以发现两者类似,都是两个任务的损失函数以及网络权重正则项、映射矩阵正则项三者的加权和,权值为该模型的超参数。因此,基于多任务学习的 LF-JLM 虽然不是纯粹的概率模型,但具有类似的概率解释。

在该模型训练过程中,因为式(14)中输出单元是内积形式而不是非线性激活函数,导致一般的反向传播算法不可用。为了确保参数正常收敛,可以采用交替梯度下降法:在用反向传播迭代优化网络权值的每一步中,对奇数步固定用户隐因子,仅对商品隐因子求导后进行梯度下降;偶数步反之,固定商品隐因子,仅对用户隐因子求导后进行梯度下降。

4 实验

4.1 数据集

本文实验中使用了 McAuley 等^[6]收集的

Amazon.com 网站的评论数据集。该数据集包含了从 Amazon.com 的各个类别的商品下收集的用户 id、商品 id、评论文本、评分等。由于机器配置的限制(实验用机器配置为 2.6 GHz 4 核处理器, 8 GB 内存), 在本实验中仅使用了 Amazon.com 网站上 instant video 子类下的数据集, 它包含 312930 个用户, 22204 个商品以及 717651 条评论, 共 88958349 个单词, 平均每条评论长度约为 124 个单词。

4.2 基线

为了便于分析和说明实验结果, 将 3.2 节中仅采用了第一种改进的模型命名为隐因子共享表示模型(Shared representation model with latent factor, LF-SRM); 将仅采用了第二种改进的模型称为共同学习模型(Joint learning model, JLM)。实验中将本文提出的模型和如下几种模型进行了比较:

a) LFM。如 1.2.1 节所述, 这是 Koren 等^[7]提出的标准矩阵分解模型。训练该模型的过程中不使用任何评论数据。实验中按照 Koren 等^[7]的描述实现了该模型, 并采用了交替梯度下降方法进行训练。

b) HFT。这是 Mcauley 等^[6]提出的结合评论和评分的推荐模型。如 1.1 节所述, HFT 模型用矩阵分解建模评分, 用 LDA 建模评论, 并用一个指数函数变换来联系两者各自得到的商品的实数值向量和商品的概率向量。本文用 Mcauley 等^[6]发表的训练结果作为对比。

c) SRM。即 3.1 节中提出的共享表示模型。实验中先使用 python 的自然语言处理和主题建模开源库 gensim^[23]训练评论文本得到每条评论的评论向量, 然后分别训练两个神经网络来预测评分和评论向量, 两个神经网络使用相同的用户向量和商品向量映射矩阵。

d) LF-SRM。仅使用 3.2 一节中提出的改进 1 的模型。预测评分的部分不采用神经网络而是采用隐因子模型, 且使用三层前馈神经网络学习 doc2vec 事先训练得到的评论向量。

e) JLM。仅使用 3.2 一节中提出的改进 2 的模型。用三层神经网络的回归模型预测评分, 用共同学习的方式学习评论向量。

f) LF-JLM。同时使用改进 1 和改进 2 的模型, 也是本文最终提出的模型。

4.3 实验过程

实验中用 Google 的 tensorflow 库^[24]来构造神经网络模型。首先将数据集按照 8:1:1 的比例随

机地分为训练集、验证集和测试集三个部分。初始化时, 用均匀分布随机初始化用户向量、商品向量、词向量的映射矩阵, 其他的神经网络权值参数均用 Glorot 等^[25]提出的 Xavier 方法进行初始化。

实验中对同一种超参数配置试验 3 次, 每次将数据集中样本的顺序打乱, 并采用 mini-batch 梯度下降和 Adam 算法进行训练。每次训练当模型在验证集上的评分均方误差达到最小时, 就停止本次训练, 并将 3 次得到的在验证集上最小均方误差的平均值作为接下来超参数选择的标准。

因为模型中超参数较多(包括正则项和目标函数的权重、神经网络隐藏层单元数、各个映射矩阵的维度、batch 的大小等), 且多为连续变量或具有有序关系的离散变量, 实验中并未使用随机搜索或网格搜索, 而是在手工选择超参数之外辅以贝叶斯优化(Bayesian optimization)算法^[26]选择超参数。利用高斯过程回归对该模型的损失函数与超参数取值之间函数关系的后验分布进行建模, 并根据建模结果选择下一次采样的超参数, 重复这个循环, 当模型的表现不再进一步提升时, 或实验的次数达到预先设定的阈值时, 就认为得到了最优的超参数配置。之后, 将这个超参数配置在测试集上的评分均方误差表现作为最终的结果。利用贝叶斯优化算法对 LF-JLM 进行超参数选择时采样的样本分布的可视化如图 4 所示。

图 4 中显示了对 LF-JLM 的 X_0 到 X_6 共 7 个超参数的采样, 它们分别为:

X_0 : 用户嵌入向量和商品嵌入向量的维度;

X_1 : 词嵌入向量和文档嵌入向量的维度;

X_2 : 用户隐因子和商品隐因子的维度;

X_3 : 窗口中心词预测的交叉熵项权重 λ_1 ;

X_4 : 映射矩阵正则项的权重 λ_3 ($\lambda_3 = \lambda_4 = \lambda_5$);

X_5 : mini-batch 梯度下降的批大小(Batch size);

X_6 : 学习率(Learning rate)。

观察采样的分布, 可以看出参数 X_0 、 X_1 、 X_2 的采样比较集中, 表明它们能通过算法找到较好的解, 并且在决定模型表现的时候和其他参数的耦合不大。其余的参数没有这样优良的性质, 需要辅以手工调整。参数中没有出现用于调节非线性层的网络参数正则项权重的 λ_6 、 λ_7 、 λ_8 , 这是因为在实验中发现, 训练神经网络时对各个映射矩阵采用 L2 正则项, 而其余的神经网络隐藏层采用 dropout 方法代替 L2 正则能比单纯使用 L2 正则项取得更好的结果。

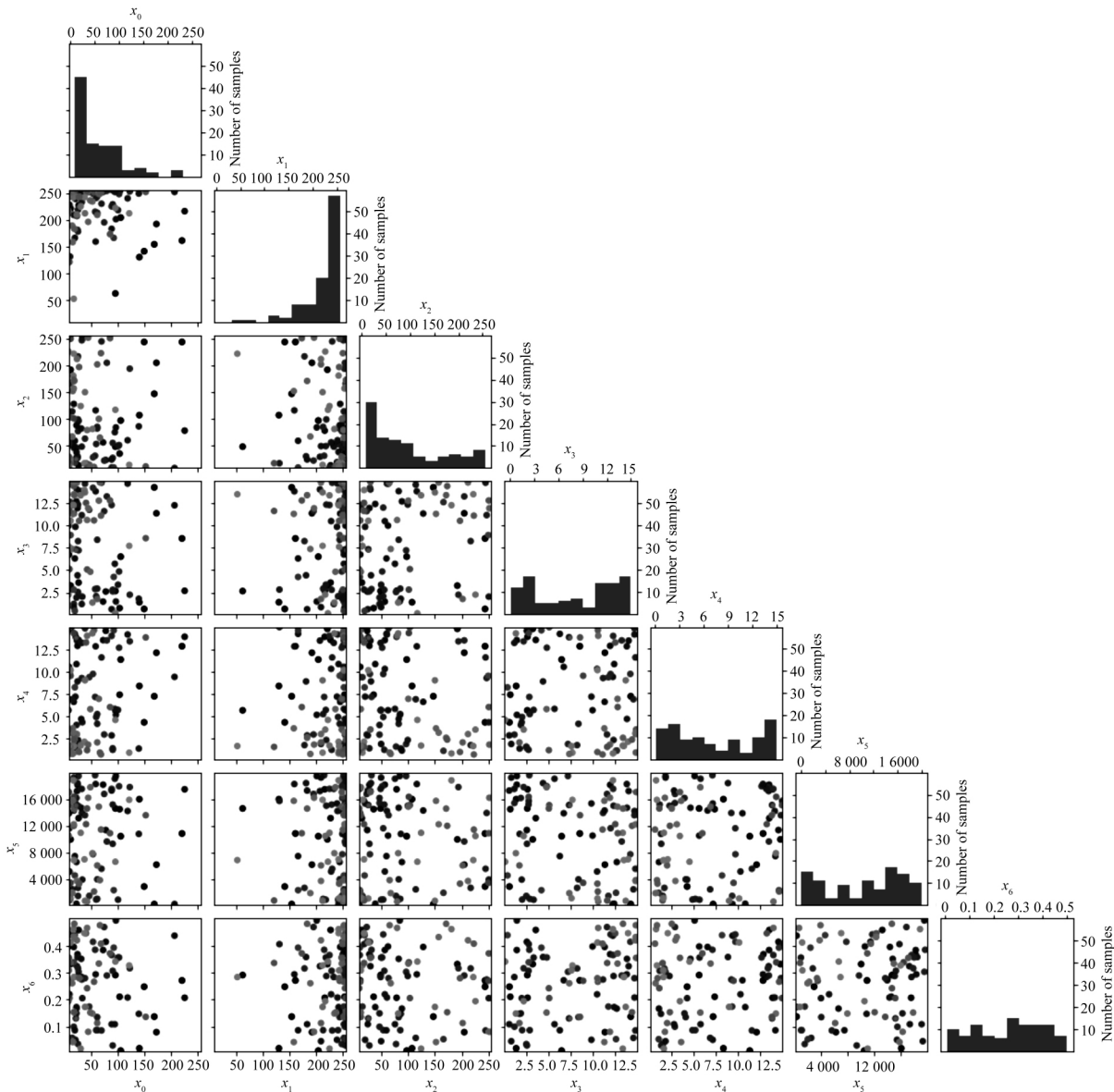


图 4 LF-JLM 超参数选择时的采样

Dropout 方法是 Srivastava 等^[27]提出的一种能有效避免神经网络过拟合的技巧,在训练深度神经网络的过程中,每训练一个 mini-batch 时网络中的每个结点有一定的概率被暂时从网络中删除,因此相当于每一个 mini-batch 都在训练不同的网络。在对本文提出的 LF-JLM 的训练中证实了这个技巧能够有效避免过拟合,且能够使训练过程更快收敛。

通过以上贝叶斯优化以及部分的人工选择超参数,最终模型的设置总结如下:

- a) 用户和商品的嵌入向量维度为 16,词向量的维度为 256。
- b) 用户隐因子、商品隐因子的维度为 32。
- c) 目标函数中评分的均方误差的权值为 1,评论窗口中心词预测的交叉熵项的权值为 10,映射矩

阵正则项的权值为 0.3。

d) Dropout 方法删除网络结点的概率为 0.8。

4.4 实验结果和分析

使用贝叶斯优化对 SRM、LF-SRM、JLM、LF-JLM 四种模型进行超参数选择的收敛曲线如图 5 所示。各个模型在测试集上的预测评分的平均均方误差见表 1。

表 1 对比实验结果

模型	平均均方 误差	均方误差提升/%	
		相对于 LFM	相对于 HFT
LFM	1.351	—	—
HFT	1.260	6.74	—
SRM	1.297	4.00	−2.93
LF-SRM	1.295	4.15	−2.78
JLM	1.248	7.77	1.11
LF-JLM	1.245	7.85	1.19

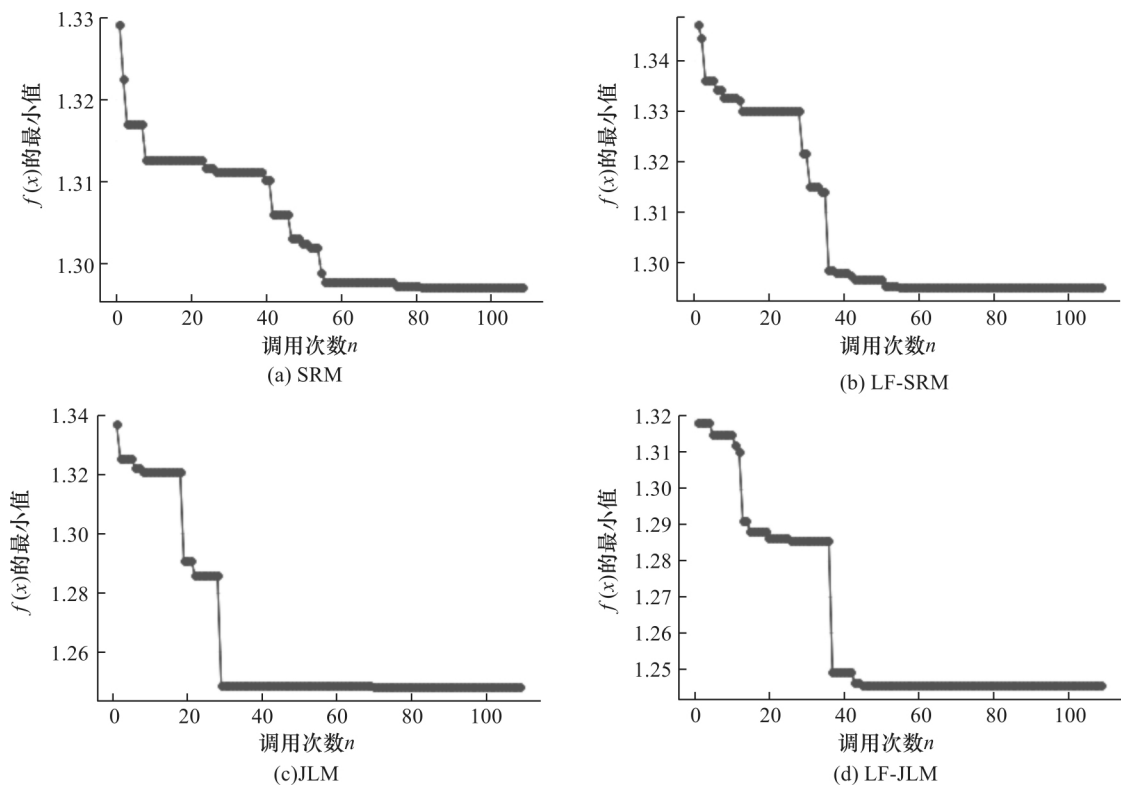


图5 超参数选择收敛曲线

从超参数选择的收敛曲线可以发现,基于共同学习的模型 JLM 和 LF-JLM 在超参数选择时收敛速度略快,这表明贝叶斯优化方法能较快地找到较好的超参数设置。而且 SRM 和 LF-SRM 的最优超参数设置相比随机初始的超参数设置,模型表现的差异不大;但 JLM 和 LF-JLM 的最优超参数设置相比随机初始的超参数设置,模型表现差异却较大。这表明共同学习模型对超参数设置比较敏感,不同的超参数设置会很大程度地影响模型的表现。因此超参数设置是该模型训练中至关重要、不可或缺的一步。

观察表 1 中对比实验的结果,可以发现:

- a) 对比 LFM、LF-SRM 和 SRM 的表现,发现 LF-SRM 的推荐效果比 LFM 更好;而且即使 SRM 没有使用任何一种改进方法,其效果也超过了未考虑评论文本数据的 LFM。
- b) 对比 SRM 和 LF-SRM、JLM 和 LF-JLM 的表现,发现在评分预测子任务中,基于隐因子模型的改进方法相对于简单的神经网络回归方法其表现仅有小幅度提升。
- c) 对比 SRM 和 JLM、LF-SRM 和 LF-JLM 的表现,发现在评分预测子任务中,基于共同学习的方法比较显著地提高了模型的效果,其在测试集上的均方误差减小了约 4%。

d) 同时两种改进方法、结合了隐因子模型和 doc2vec 模型的 LF-JLM 推荐模型相比本文实验中的其他推荐模型,其推荐结果最好,其均方误差比 HFT 模型减小了约 1.2%。

下面对以上实验结果进行详细分析。

a) 除了用于评论向量预测任务的子网络之外,LFM 和 LF-SRM 的结构几乎相同。这表明 LF-SRM 和 SRM 相比 LFM 取得效果提升的原因正是因为它确实能够有效地利用评论文本预测的任务来辅助评分预测任务,能潜在地将评论文本中的信息提取出来并应用到预测评分的推荐任务之中。

b) 基于隐因子模型的改进方法本可以显著减少原始模型的复杂度,避免过拟合现象;但在采用 dropout 方法后,过拟合现象已经被有效缓解,因此这种改进方法的效果并不显著。

c) 基于共同学习的改进要求神经网络在建模评论文本的过程中同时兼顾评分预测任务,因此模型在两个任务之间取得了较好的平衡;而没有采用共同学习的 SRM、LF-SRM 都是先训练出文档向量再将其作为目标拟合,其预测评分的神经网络需要去拟合已经得到的文档向量,网络在评论建模的部分受到较大的限制,没有前者的灵活性。因此基于共同学习的改进方法能显著地提高模型的推荐效果。

通过本文提出的模型得到的用户和商品的嵌入表示不同于矩阵分解模型得到的用户和商品的隐因子,虽然两者都以某种方式编码了用户和商品的语义信息,但后者存在清晰的解释,而前者类似词向量,难以解释各个维度的意义是什么。但是,正如词向量具有类比性质^[29],且能作为特征加入到传统的自然语言处理任务显著地提升效果,本文得到的用户向量和商品向量可能也有类似的性质,或具有改进其他推荐模型效果的潜力,这也是本文未来的研究方向之一。

参考文献:

- [1] 王国霞,刘贺平.个性化推荐系统综述[J].计算机工程与应用,2012,48(7):66-76.
- [2] 刘平峰,陈冬林.电子商务推荐系统研究综述[J].情报杂志,2007,26(9):46-50.
- [3] Ricci F, Rokach L, Shapira B, et al. Recommender Systems Handbook[M]. Springer,2011:1-35.
- [4] Shang M, Lu L, Zhang Y C, et al. Empirical analysis of web-based user-object bipartite networks[J]. Europhysics Letters,2012,90(4):1303-1324.
- [5] Purushotham S, Liu Y, Kuo C. Collaborative topic regression with social matrix factorization for recommendation systems[C]// International Conference on International Conference on Machine Learning. Omnipress, 2012:691-698.
- [6] McAuley J, Leskovec J. Hidden factors and hidden topics: Understanding rating dimensions with review text[C]// ACM Conference on Recommender Systems. ACM, 2013:165-172.
- [7] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems [J]. Computer, 2009,42(8):30-37.
- [8] Bao Y, Fang H, Zhang J. Topic M F: Simultaneously exploiting ratings and reviews for recommendation[C]// Twenty-Eighth AAAI Conference on Artificial Intelligence. AAAI Press,2014:2-8.
- [9] Ling G, Lyu M R, King I. Ratings meet reviews, a combined approach to recommend[C]// Proceedings of the 8th ACM conference on recommender systems. ACM,2014:105-112.
- [10] 彭敏,席俊杰,代心媛,等.基于情感分析和LDA主题模型的协同过滤推荐算法[J].中文信息学报,2017,31(2):194-203.
- [11] 王建,黄佳进.LDA-RR:一种基于评分和评论的推荐方法[J].计算机科学,2017,44(2):267-269.
- [12] Wu Y, Ester M. FLAME: A probabilistic model combining aspect based opinion mining and collaborative filtering[C]// Eighth ACM International Conference on Web Search and Data Mining. ACM, 2015:199-208.
- [13] Wang H, Wang N, Yeung D Y. Collaborative deep learning for recommender systems[C]// Twenty-First ACM SIGKDD Conference on Knowledge Discovery and Data Mining,2015:1235-1244.
- [14] Kim D, Park C, Oh J, et al. Convolutional matrix factorization for document context-aware recommendation [C]// ACM Conference on Recommender Systems. ACM,2016:233-240.
- [15] Almahairi A, Kastner K, Cho K, et al. Learning distributed representations from reviews for collaborative filtering [C]// ACM Conference on Recommender Systems. ACM,2015:147-154.
- [16] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation,1997,9(8):1735-1780.
- [17] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality [C]// International Conference on Neural Information Processing Systems. Curran Associates Inc,2013:3111-3119.
- [18] Le Q, Mikolov T. Distributed representations of sentences and documents[C]// International Conference on Machine Learning. New York: ACM,2014:873-882.
- [19] Dong J, Zhuang D, Huang Y, et al. Advances in multi-sensor data fusion: Algorithms and applications [J]. Sensors,2009,9(10):7771.
- [20] Deneve S, Pouget A. Bayesian multisensory integration and cross-modal spatial links[J]. Journal of physiology, Paris,2004,98(1/2/3):249.
- [21] Caruana R A. Multitask learning: A knowledge-based source of inductive bias [J]. Machine Learning Proceedings,1993,10(1):41-48.
- [22] Yu J, Jiang J. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification [C]// Conference on Empirical Methods in Natural Language Processing,2016:236-246.
- [23] Radim R. Scalability of semantic analysis in natural language processing [D]. Brno: Masaryk University,2011.
- [24] Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-scale machine learning on heterogeneous

- distributed systems[EB/OL]. (2016-03-16)[2018-05-17]. <https://arXiv.org/abs/1605.08695>.
- [25] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks [C]// Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), Chia Laguna Resort, Sardinia, Italy, 2010, 9: 249-256.
- [26] Mockus J. Bayesian Approach to Global Optimization [M]. Kluwer Academic Publishers, 1989: 7-60
- [27] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting [J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
- [28] Maaten L, Hinton G. Visualizing data using t-SNE[J]. Journal of Machine Learning Research, 2008 (9): 2579-2605.
- [29] Mikolov T, Yih W T, Zweig G. Linguistic regularities in continuous space word representations [J]. The North American Chapter of the Association for Computational Linguistics, 2013, 29(7): 71-79.

(责任编辑: 康 锋)