GitHub 中基于 CNN-LSTM 的开发者项目推荐模型

廖志芳[1],杨洪瑜[1,2],宋天惠[3],郁 松[1],齐笑斐[1]
(1. 中南大学计算机学院,湖南长沙 410083; 2. 中国航空无线电电子研究所,上海 200241;
3. 中南大学软件学院,湖南长沙 410083)

摘 要: 作为一个开源项目托管平台,GitHub 以多开发者协同参与进行开源项目的开发,开发者作为 GitHub 的核心元素,保证了整个系统的活跃性,然而,很多新项目在短时间内无法找到合适的协同开发者而被拖延开发周期. 针对这个问题,本文提出了一种基于 Word2Vec 的 CNN-LSTM 开发者项目推荐模型,该模型以 Word2Vec 训练开发者访问项目的序列,并将项目进行向量化表示,结合 CNN-LSTM 模型计算项目相似度并为开发者推荐合适的项目序列. 通过提取 GitHub 中 62,031 个开发者在 2015 全年的项目访问数据进行项目预测和相似项目发现实验,实验结果表明,该模型推荐效果较佳,并且可以帮助开发者发现感兴趣的相似项目.

关键词: GitHub; 项目预测; 项目推荐
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2020)11-2202-06
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2020.11.015

Developer Project Recommendation Model Based on CNN-LSTM in GitHub

LIAO Zhi-fang[1], YANG Hong-yu[1,2], SONG Tian-hui[3], YU Song[1], QI Xiao-fei[1]
(1. *School of Computer Science and Engineering, Central South University, Changsha, Hunan* 410083, *China*;
2. *China Aeronautical Radio Electronics Research Institute, Shanghai* 200241, *China*;
3. *School of Software, Central South University, Changsha, Hunan* 410083, *China*)

Abstract: As an open source project hosting platform, GitHub participates in the development of open source projects with multi-developers. As the core element of GitHub, developers ensure the activity of the whole system. However, many new projects can not find suitable collaborative developers in a short time and the development cycle gets delayed. To solve this problem, this paper proposes a CNN-LSTM developer project recommendation model based on Word2Vec, which trains developers to access the project sequence by Word2Vec, and vectorizes the project, calculates the project similarity with CNN-LSTM model, and recommends the appropriate project sequence for developers. Through the project prediction and similar project discovery experiments based on 62,031 developers' project access data in GitHub in 2015, the experimental results show that the model has better recommendation effect and can help developers find similar projects of interest.

Key words: GitHub; project forecast; project recommendation

1 引言

GitHub 是目前互联网中最大的开源软件社区及项目托管平台,自上线以来已拥有超过 2000 万开发者,托管项目数超过 7000 万.

本文通过调研发现目前 GitHub 中存在以下问题:(1)项目与开发者分布不均,少量的开发者参与了大部分的项目[1];(2)开发者往往经过漫长的关注和收藏项目以后,才可以判断自己是否具备贡献项目所需的能力并对项目作出贡献;(3)大量新创建项目在相当长一段时间内都不会收到 Pull Request,新项目无法及时被开发者们关注到,因此很多新的项目在启动初期难以快速发展壮大.

针对以上问题,为了帮助开发者快速发现感兴趣

收稿日期:2019-11-11;修回日期:2020-06-17;责任编辑:李勇锋
基金项目:科技部重点研发项目(No. 2018YFB003800);湖南省自然科学基金(No. HNNSF2018JJ2536)

的项目,本文提出了一种基于 Word2Vec 的 CNN-LSTM 开发者项目推荐模型,该模型首先通过 Word2Vec 模型将项目转化为基于项目序列上下文的向量形式,再通过 CNN-LSTM 模型获取开发者们基于上下文的开发者项目特征,从而对开发者们可能关注的陌生项目进行预测,将获得的项目预测列表推荐给合适的开发者,提高开发者查找项目的效率.

2　相关工作

在开发者项目推荐方面,传统方法[1~6]主要有基于内容的推荐和基于协同过滤方法的推荐传统的协同过滤方法.

Liao 等[7,8]提出了一种改进的分层粒子群算法的马尔可夫链模型,对预测开发者会话很有帮助,Wu 等[9]基于项目相似度矩阵和用户项目矩阵,使用需求度量来预测用户对未知项目的需求,为开发者们推荐 top-N 个项目. Zhang 等[10]通过分析开发者的 commit 行为和项目文本信息,采用协同过滤与文本匹配相结合的方式为开源项目推荐贡献者,能有效应对冷启动问题且准确率达到 63%.

目前,深度学习方法[11,12]也在推荐算法中有了广泛的运用,多位研究者分别通过 RBM、RNN、CNN 等深度学习模型获取特征,其应用到推荐系统中取得了较好推荐效果. Zhang[13]等在 2019 年提出了基于深度学习方法的 GitHub 开发者项目推荐模型 FunkR-pDAE,将推荐准确率提高到了 75.46%.

3　基于 Word2Vec 的 CNN-LSTM 开发者项目推荐模型

由于 GitHub 中项目有效描述信息非常短小,同时也缺乏对开发者的有效标注,这使得项目和开发者的匹配效率不高. 针对这个问题,本文提出了一种基于 Word2Vec 的 CNN-LSTM 项目预测模型对开发者可能感兴趣项目进行预测从而为其推荐相关的 top-k 个项目. 该模型通过 Word2Vec 算法将项目转化为项目向量,同时它结合 CNN-LSTM 模型对开发者可能访问的项目进行预测,得到针对开发者的项目推荐列表,这有利于提高用户发现感兴趣项目的效率.

模型整体设计如图 1 所示,①将开发者按照时间序列访问的项目集作为整个模型的输入;②通过 Word2Vec 模型确定项目在项目空间中的位置,并将项目转换为了向量的形式,从而将输入的开发者项目集转换为了矩阵的形式;③将开发者项目矩阵输入 CNN 模型通过卷积核池化操作得到开发者项目初级特征;④通过 LSTM 模型获取基于上下文的开发者项目特征;⑤最后通过 sigmod 函数将该特征向量与项目库进行比较,根据预测结果给开发者推荐 top-k 个项目.

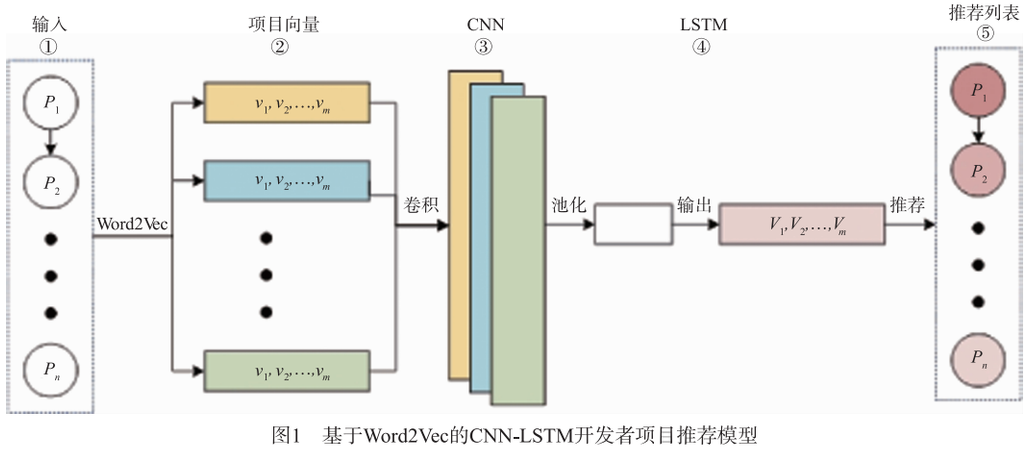

图1　基于Word2Vec的CNN-LSTM开发者项目推荐模型

3.1　基于 Word2Vec 的项目向量模型

Word2Vec 模型是 Google 提出的一种词嵌入模型,通过挖掘词与上下文的关系将词转为向量形式,从而方便度量词的相似度. 由于 GitHub 中开发者在某段时间内会持续对某一个或几个项目进行贡献,开发者访问项目的序列会有一定的上下文关系,因此项目之间也会像词一样具备相似关系,故本文采用 Word2Vec 模型挖掘项目在上下文中的关系,在将项目转换为项目空间中的向量的同时也保留了项目间的相关关系.

Word2Vec 模型首先需要基于词袋模型构建项目词典,初始的项目表示为其频率,然后通过不断的训练更新,得到最终的项目向量表示. Word2Vec 模型有两种常见训练模型 CBOW 和 Skip-Gram,由于 GitHub 中开发者和项目数量较多,而 Word2Vec 中的 Skip-Gram 模型更适合处理大型的数据集,其跳跃选择序列的机制也使得模型不受窗口大小限制,能很好地解决了开发者访问序列较长的问题. 因此,本文采用 Word2Vec 的 Skip-Gram 模型对项目向量模型进行训练,该模型的训练算法如图 2 所示.

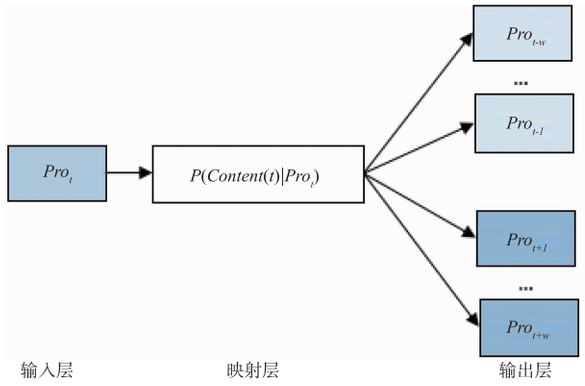

图2 Word2Vec模型图

Skip-Gram 模型的基本原理就是给定一个项目后，预测在它左右两边可能会出现什么项目. 对于每个项目 t 存在前后两个大小为 w 的窗口样本和，这两个窗口构成了项目 t 的上下文环境，因此项目 t 出现在上下文的项目集合的概率可用如下公式表示.

$$P(Content(t)|Pro_t) = P(Pro_{t-w},\cdots,Pro_{t-1},Pro_{t+1},\cdots,Pro_{t+w}|Pro_t) \tag{1}$$

因为 Word2Vec 模型基于词袋模型设计，故项目 t 出现上下文项目集合的概率可用式(2)计算.

$$P(Content(t)|Pro_t) = \prod_{i=1}^{N} p(u_i|Pro_t) \tag{2}$$

其中表示在项目库中的第 i 个项目，表示项目出现在项目 t 的上下文中的概率.

Word2Vec 模型训练时先为项目设置初始向量值，然后以项目访问序列构建训练样本，将样本依次输入到映射层 m 个神经元中，通过神经网络从每个神经元的计算获得了构成项目的向量矩阵，其中第 i 个项目的项目向量可以式(3)进行表示，m 为向量的维度数目.

$$\boldsymbol{V}_i = [v_1, v_2, \cdots, v_m] \tag{3}$$

通过对比项目向量与项目初始向量的差距不断调整项目的向量矩阵完成训练并输出概率. 通过训练模型过程中获得的项目向量矩阵即可将项目转换为其在项目空间中对应的向量形式，将项目与其最终得到的向量表示更新到项目词典中去.

3.2 CNN-LSTM 的开发者项目推荐模型

目前 CNN 模型在获取特征方面得到广泛的应用，通过 CNN 模型可以避免传统特征模型忽略项目上下关系的缺陷，因此对本文首先使用 CNN 模型对项目序列进行初级提取. 虽然 CNN 模型可以获取卷积核大小内的项目关联特征，但是获取的项目特征不具有长期的关联特征，而 LSTM 模型可以对开发者的项目获取长期基于项目上下文的特征，因此本文将再通过 LSTM 模型对开发者项目特征进行进一步地特征提取.

本文采用 CNN-LSTM 的项目预测模型进行项目特征提取，如图 3 所示，通过 CNN 模型获取开发者项目初级特征后，再利用 LSTM 进一步获取基于时间序列的项目上下文特征，进行项目相似度计算，并对开发者感兴趣的项目进行预测，将预测的 top-k 个项目推荐给开发者.

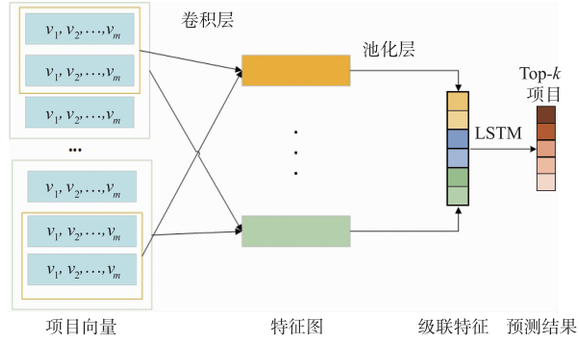

图3 CNN-LSTM的项目预测模型图

项目向量输入项目预测模型后，在卷积层先通过卷积操作提取项目信息，由于不同尺寸的卷积核得到的特征图的大小是不一样的，要完成各个卷积项目特征图的级联最常用的方法就是对特征图进行池化，并且提取出特征图的局部池化最大值，这样将每一个卷积核得到的特征对应为一个值，最后在级联起来，得到最终的特征向量. 步骤如下：

(1)在卷积层中，我们使用不同尺寸的卷积核来获取初级特征，每个卷积核包含不同权重矩阵. l 是窗口大小，这意味着卷积核对 l 个项目进行操作，d 是项目向量的维度.

$$\boldsymbol{v}'_i = \sigma(W^m \cdot \mathrm{v}_{i:i+l-1} + b) \tag{4}$$

其中 σ 为激活函数，b 为偏置. 当对开发者项目矩阵进行完卷积操作后，如式(5)表示出经过对开发者项目卷积操作生成的特征图. 通过卷积操作，分别对 l 个项目进行特征提取，可以充分表示开发者的项目特征.

$$\boldsymbol{v}' = [v'_1, v'_2, \cdots, v'_{n-l+1}] \tag{5}$$

(2)卷积操作之后是一个池化层，目的是对卷积得到的项目特征矩阵进行降维. 为了缓解神经网络训练过程中的过拟合问题，基于 GitHub 中项目的自身属性，本文使用最大池化方法对得到的特征矩阵进行池化操作. 另外，池化层的另一个目的将大小不同的卷积特征图通过不同方式的池化降维，转化为长宽相同的特征图，然后从深度进行特征图的连接，将连接后的特征图作为 LSTM 的待输入特征.

LSTM 模型神经元步骤如下：

(1)(LSTM 模型每个时刻 t 的神经元都会通过输入门输入上一个神经元的输出向量 $\boldsymbol{h}_{t-1}$ 和特征图中 t 时刻对应的向量 $\boldsymbol{x}_t$，如式(6)、(7)进行处理得到 i_t、$\tilde{C}_t$，这样既保留了此次的项目特征，也将上一个神经元的

信息进行了保留;

$$i_t = \sigma(W_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + b_i) \tag{6}$$

$$\tilde{C}_t = \tanh(W_C \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + b_C) \tag{7}$$

(2)然后通过遗忘门对输入门得到的向量信息进行处理,如式(8)得到向量 $\boldsymbol{f}_t$,同时根据得到的遗忘信息对神经元信息进行更新,如式(9)得到向量 $\boldsymbol{C}_t$;

$$\boldsymbol{f}_t = \sigma(W_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + b_f) \tag{8}$$

$$\boldsymbol{C}_t = \boldsymbol{f}_t * C_{t-1} + i_t * \tilde{C}_t \tag{9}$$

(3)最后,如式(10)、(11),输出门将该神经元处理得到的特征向量 $\boldsymbol{h}_t$ 传入下一个神经元中,同时传入最终的特征矩阵中,这样本神经元输出的特征向量既包含的此处的项目特征,也保留了上文的向量特征,也将此项目的特征传入了下个神经元.

$$o_t = \sigma(W_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + b_o) \tag{10}$$

$$\boldsymbol{h}_t = o_t \cdot \tanh(\boldsymbol{C}_t) \tag{11}$$

LSTM 模型每个时刻 t 的神经元都会输出一个特征,本文将序列长度 +1 时刻的神经元输出作为推荐项目的高级语义词向量,通过 sigmod 函数计算找到最相似的 top-k 个项目进行推荐.

3.3　训练

在训练阶段,本文采取 Logistic 回归的对数似然损失函数,衡量预测项目和真实项目的差距,通过反向传播算法更新神经网络参数值,完成网络的训练,训练公式如下所示.

$$L(Y, P(Y|X)) = -\log P(Y|X) \tag{12}$$

其中,Y 为输出变量,X 为输入变量,L 为损失函数. Logistic回归的对数似然损失函数可反映预测结果的损失率,损失率越小越好.

4　实验结果与验证数据集

4.1　数据

本文利用 GitHub Archive 提取实验数据集提取了 4331562 个开发者在 2015 年全年的项目操作数据. 由于 LSTM 模型要求序列的长短一致,故从访问数为大于 200 的 190904 个开发者中选取了 62031 位开发者,并对他们最近访问的 100 + k 个不重复的项目序列进行采样,k 为推荐的项目数量,将其分割为训练集 47871 条,测试集 14160 条.

4.2　实验说明

本实验的预测结果以准确率(Precision)、召回率(Recall)、F 值综合进行效果检验. 这三个的计算公式如下公式所示:

$$Precision = \frac{|\cap(PredictionSet, ReferenceSet)|}{|PredictionSet|} \tag{13}$$

$$Recall = \frac{|\cap(PredictionSet, ReferenceSet)|}{|Reference|} \tag{14}$$

$$F = \frac{Precision * Recall * 2}{Precision + Recall} \tag{15}$$

其中 $PredictionSet$ 为推荐项目集合,$ReferenceSet$ 为实际选择项目的集合.

本文将为开发者们推荐 top-k 个项目,准确率表示开发者参与项目在推荐列表中的比例,召回率表示推荐项目在开发者参与的项目列表中的比例.

4.3　结果与讨论

本节首先将传统的协同过滤方法与本文提出的模型进行实验对比;然后,对基于各种传统的深度学习方法和本文提出的基于 Word2Vec 的 CNN-LSTM 等开发者项目预测模型进行推荐结果分析,根据实验结果的对比分析确定模型中各个模块的有效性以及总体推荐效果的高效性.

4.3.1　损失值分析

在模型训练过程中,随着迭代次数加深,模型损失值如下图 4 所示,我们可以看出在当前模型训练中,模型最终完成收敛,并没有过多拟合.

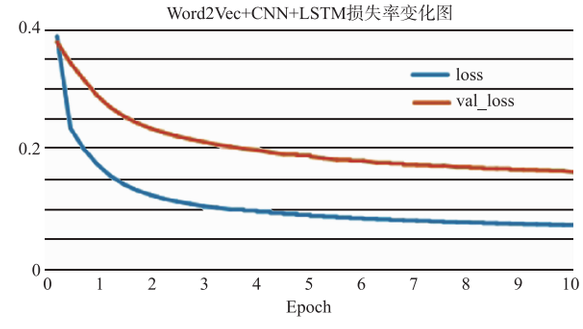

图4　模型训练损失值变化图

4.3.2　传统方法推荐结果对比

传统的项目推荐方法多采用协同过滤的方法,本文将基于之前的研究,设计多种对比实验进行推荐效果分析,top-5 新项目推荐结果如表 1 所示. 随机标签集代表通过随机方法生成的标签集;Actor 标签集是通过 LDALE 算法获得的标签集,LDALE 算法定义了一种开发者对项目熟悉度的计算方法,并结合分析项目行为文本获得的项目标签集获得开发者的个性化标签集;Profile 标签集是通过 PLCF 算法获得的标签集,PLCF 算法是一种基于 LDA 和协同过滤算法的标签集提取算法. 通过实验结果可见,本文提出的方法在各项指标上表现均优于传统的协同过滤方法.

表 1　不同标签集的协同过滤开发者项目推荐结果表

模型	准确率	召回率	F 值
基于随机标签集的推荐	37.79%	38.40%	38.09%
LDALE 算法	65.79%	66.85%	66.31%
PLCF 算法	70.49%	71.62%	71.05%
Word2Vec + CNN + LSTM	82.69%	84.02%	83.35%

4.3.3 深度学习方法推荐结果对比

近些年,深度学习方法在推荐系统中有很好的应用.传统的基于深度学习方法的项目推荐模型多采用CNN、LSTM 模型,在将项目转化为向量时候多采用随机向量的方法,忽略了开发者项目之间的联系,因此本文将 Word2Vec 模型训练向量和 LDA 训练向量分别作为传统 CNN 模型、LSTM 模型和本文提出的 CNN-LSTM 模型的输入进行实验,得到的 top-5 新项目推荐结果如表 2 所示.

表 2 基于项目特征向量表示的各模型推荐结果对比表

文本表示	特征模型	准确率	召回率	*F* 值
LDA	CNN	65.28%	66.33%	65.80%
	LSTM	68.39%	69.49%	68.94%
	CNN-LSTM	70.90%	72.04%	71.47%
Word2Vec	CNN	72.48%	73.65%	73.06%
	LSTM	76.67%	77.90%	77.28%
	CNN-LSTM	82.69%	84.02%	83.35%

实验结果表明:(1)基于 Word2Vec 模型提取的项目特征向量比基于 LDA 模型提取的项目特征向量更加适合作为开发者项目推荐模型的输入向量;(2)在基于 Word2Vec 的推荐模型中,本文提出的 CNN-LSTM 模型在各个指标上面都比 CNN、LSTM 模型的推荐效果更好,因此本文提出的模型推荐效果最佳.

最后,对实验结果进行进行总结,不管是同传统的协同过滤方法还是和传统的深度学习模型比较,本文提出的基于 Word2Vec 的 CNN-LSTM 开发者项目推荐模型项目推荐效果确实优于传统项目推荐模型和其他深度学习模型.

5 总结

本文提出了一种应用于开源软件社区 GitHub 上的基于 Word2Vec 的 CNN-LSTM 的开发者项目推荐模型.本模型能有效地预测开发者访问项目的行为,通过预测为开发者们推荐合适的项目,为开发者提高项目推荐效率,提高协作开发效率,促进 GitHub 社区的高效发展.未来希望采用更大的数据集来验证本文模型的扩展性.另外,通过分析开发者行为数据,本文并未关注开发者行为的时间戳信息,而开发者兴趣随时间将类似的周期性变化,未来希望结合马尔科夫链模型分析开发者在时间上的行为数据变化规律,提出更加准确的项目预测算法,使开发者能够更加快捷地发现感兴趣、可贡献的项目.

参考文献

[1] Liu C, Yang D, Zhang X, et al. Recommending GitHub projects for developer onboarding[J]. IEEE Access, 2018. 1 – 1.

[2] Thung F, Bissyande T F, Lo D, et al. Network structure of social coding in GitHub[A]. Proceedings of the 2013 European Conference on Software Maintenance and Reengineering[C]. Genova, Italy: IEEE, 2013. 323 – 326.

[3] Liao Z, Jin H, Li Y, et al. DevRank: Mining influential developers in Github[A]. Proceedings of the 2017 IEEE Global Communications Conference [C]. Piscataway, USA: IEEE, 2017. 1 – 6.

[4] Rahman M M, Roy C K. An insight into the pull requests of GitHub[A]. Proceedings of the 11th Working Conference on Mining Software Repositories[C]. NewYork, United States: Association for Computing Machinery, 2014. 364 – 367.

[5] Rahman M M, Roy C K, Collins J A. CORRECT: Code reviewer recommendation in GitHub based on bross-project and technology experience[A]. Proceedings of the 38th International Conference on Software Engineering Companion [C]. New York, United States: Association for Computing Machinery, 2016. 222 – 231.

[6] Zhao G, Da C, et al. Improving the pull requests review process using learning-to-rank algorithms [J]. Empirical Software Engineering, 2019, 24(4): 2140 – 2170.

[7] Liao Zhi-fang, Liu Min, et al. Markov chain-like model for prediction service based on improved hierarchical particle swarm optimization cluster algorithm [J]. International Journal of Software Engineering & Knowledge Engineering, 2016, 26(04): 653 – 674.

[8] 朱敬华,等. 传感器网络基于轨迹聚类的多目标跟踪算法[J]. 电子学报, 2017, 45(11): 2671 – 2676.
Zhu Jing-hua, et al. Multiple objects tracking algorithm by trajectories clustering in sensor network[J]. Acta Electronica Sinica, 2017, 45(11): 2671 – 2676. (in Chinese)

[9] Xu W, Sun X, Hu J, et al. REPERSP: Recommending personalized software projects on GitHub[A]. Proceedings of the 2017 IEEE International Conference on Software Maintenance and Evolution [C]. Los Alamitos, USA: IEEE Computer Society, 2017. 648 – 652.

[10] Zhang X, Wang T, Yin G, et al. Who will be interested in? A contributor recommendation approach for open source projects[A]. Proceedings of the 29th International Conference on Software Engineering and Knowledge Engineering[C]. Pittsburgh, USA: Knowledge Systems Institute Graduate School, 2017. 363 – 369.

[11] Dong Xin, Yu Lei, Wu Zhonghuo, et al. A hybrid collaborative filtering model with deep structure for recommender systems[A]. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence[C]. San Francisco, Cali-

fornia USA:AAAI Press,2017. 1309 – 1315.

[12] Dai H,Wang Y,Rakshit Trivedi,Song L. Recurrent coevolutionary latent feature processes for continuous-time recommendation[A]. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems[C]. New York, United States:Association for Computing Machinery,2016. 29 – 34.

[13] Zhang P, Xiong F, Leung H K N, et al. FunkR-pDAE: Personalized project recommendation using deep learning [J]. IEEE Transactions on Emerging Topics in Computing,2018:1 – 1. 10. 1109/TETC. 2018. 2870734.

作者简介

廖志芳　女,1968 年出生于湖南长沙,博士,教授,主要研究领域为数据挖掘、开源生态系统的分析与研究等.
E-mail:zfliao@ csu. edu. cn

杨洪瑜　女,硕士研究生,1994 年出生于山东烟台,研究方向为数据挖掘与可视化技术.
E-mail:174712091@ csu. edu. cn

郁　松(通讯作者)　男,1974 年出生,中南大学计算机学院软件工程系,博士,副教授,硕士生导师,主要研究领域为大数据、图像处理以及软件工程等领域.