

# 基于多级注意力机制网络的 app 流行度预测

张艺璇 郭斌 刘佳琪 欧阳逸 於志文

(西北工业大学计算机学院 西安 710029)

(zhangyixuan2014@mail.nwpu.edu.cn)

## app Popularity Prediction with Multi-Level Attention Networks

Zhang Yixuan, Guo Bin, Liu Jiaqi, Ouyang Yi, and Yu Zhiwen

(College of Computer Science, Northwestern Polytechnical University, Xi'an 710029)

**Abstract** The popularity prediction of mobile apps provides substantial value to a broad range of applications, ranging from operational strategy optimization to targeted advertising and investment. This work includes leveraging the rich data provided by the app market to mine the dynamic correlation between different factors and popularity, so as to predict the app popularity over the next period of time, which creates great value for developers, investors and the app market. However, the evolution of app popularity is highly dynamic, and its influence factors are very complex, including the iterative evolution of the app itself, user feedback, and competition for similar products and so on. At present, there are relatively few research studies on app popularity modeling and prediction. Most of them construct artificial features and capture its association with popularity, and there is room for improvement in terms of computational performance, prediction accuracy, and interpretability of results. In this paper, we propose DeePOP, an attention based neural network for app popularity modeling and prediction, which performs hierarchical modeling for complex influence factors. First, we propose the time-level self-sequence module to capture the long-term dependence on historical popularity, and propose the local and global feature level modules to capture the nonlinear relationship between features and app popularity. Second, the attention mechanisms provide adaptive capabilities for different modules to capture most relevant historical states and provide explanation for prediction. Last, the experimental results show that DeePOP outperforms the state-of-the-art methods and the root mean square error of prediction reaches up to 0.089.

**Key words** mobile app; popularity; prediction; attention mechanism; neural network

**摘要** 移动 app 流行度预测从运营策略优化到短期广告投资都具有重要意义。利用应用市场提供的丰富数据,挖掘不同特征指标与流行度之间的动态关联,从而预测未来一段时间内 app 的流行度变化过程及其原因,对于开发者、投资商和应用市场三方都具有应用价值。然而,app 流行度高度动态变化,其影响因素十分复杂,包括 app 自身迭代演化、用户反馈、同类产品的市场竞争等。目前,针对 app 流行度建模与预测的研究工作相对较少,大多通过构造人工特征并构建与流行度的关联来进行预测,在模型的计算性能、预测精度和结果可解释性等方面存在可提升的空间。因此针对 app 流行度建模与预测提出一种基于注意力机制的深度神经网络模型 DeePOP,并针对复杂影响因素进行分级建模。通过时间级自序列

收稿日期:2019-09-16;修回日期:2019-12-13

基金项目:国家重点研发计划项目(2017YFB1001803);国家自然科学基金项目(61772428,61725205)

This work was supported by the National Key Research and Development Program of China (2017YFB1001803) and the National Natural Science Foundation of China (61772428, 61725205).

通信作者:郭斌(guob@nwpu.edu.cn)

模块捕获对历史流行度的长期依赖,利用局部、全局特征层级模块构建影响特征与流行度的非线性关系。同时,注意力机制为不同模块提供自适应能力,以捕获与流行度变化最相关的历史状态并针对预测结果提供一定的解释。实验结果表明:与现有流行度预测方法相比,DeePOP 能够快速高效地进行 app 流行度建模与预测,预测均方根误差为 0.089。

**关键词** 移动 app; 流行度; 预测; 注意力机制; 神经网络

**中图法分类号** TP18; TP391.1

随着智能手机的普及,数以万计的移动 app 在应用市场中发布并服务于人们生活中的各个方面,例如社交通信、在线购物和日常娱乐等。有数据显示,截至 2018 年 8 月,Google Play 可供用户下载的 app 多达 280 万款<sup>[1]</sup>。而受手机内存和使用习惯的限制,每个用户在手机上安装并使用的 app 数量十分有限,1 项来自雅虎团队的研究表明每个移动设备平均安装 96 款 app<sup>[1]</sup>。面对应用市场中如此激烈的竞争与 app 日趋同质化的倾向,app 开发商需要不断提高各自产品的流行度和曝光率,从而吸引更多的用户下载使用,避免产品被淘汰而带来的巨大经济损失。所以针对 app 的流行度演化进行建模和预测具有较强的应用价值。其中,流行度作为抽象的指标具有多种量化方式,例如日下载量、市场排名和日活跃用户数等,本文采用日下载量这一直观反映用户接受程度的指标来量化流行度。

具体来说,对于 app 开发者而言,预先知道流行度的变化趋势能够助力运营决策,当未来流行度表现不佳时便于提前制定计划、采取措施以及提高 app 产品的人气。从长远来看,长期掌握 app 的流行度变化就相当于了解其在生命周期内的演化过程,这能为 app 的长期运营与维护提供积极导向,避免产品因各种原因被下架,尽可能延长 app 的上架时间。对于广告商和投资者来说,及时掌握目标 app 的未来流行度表现有利于正确选择短期合作或投资对象,促使收益最大化并有效规避投资风险。而持续关注流行度变化能够掌握 app 市场的风向变化,发掘潜在商机。对于应用市场来说,预知流行度有助于更新市场排名信息,提高用户对于推荐服务的体验。另外掌握 app 的流行度演化趋势有利于应用市场的管理和 app 生态系统的发展。

近年来,通过从应用市场获取丰富的用户交互和使用数据来进行 app 分析预测成为研究热点,相关工作包括:app 流行度预测<sup>[2-4]</sup>、使用预测<sup>[5]</sup>、使用推荐<sup>[6-7]</sup>和应用市场分析与挖掘<sup>[8]</sup>等。其中,流行度建模与预测并不是 app 领域的特有问题,它更广泛

地应用于社交媒体内容的领域,例如新闻<sup>[9]</sup>、微博<sup>[10]</sup>、图像<sup>[11]</sup>和视频<sup>[12]</sup>等。但是,app 的流行度预测工作仍然存在 2 个难点从而需要进一步探索研究方法。

1) app 流行度的影响因素非常复杂。现有的研究方法多是手动设计特征或针对原数据提取统计特征,再利用有限的模型构建特征与流行度之间的线性关系或是预定义的非线性关系。这无疑会在建模过程中丢失原数据中的隐含依赖关系。

2) app 个体差异大,生命周期没有共性规律。不同于线上社交内容的单一性和时效性,app 可能反复迭代更新并且长时间地存在于应用市场中,也可能因为经营不善在上架一段时间后被市场淘汰,且不同 app 所受影响因素及其侧重差异较大。所以线上内容的流行度预测方法并不适用于不断更新的 app。另外,现有研究方法的预测精度仍存在较大的提升空间,如何探索并设计更加强大的模型以全面建模众多影响因素是研究人员共同面对的挑战。

近年来深度神经网络以其强大的计算能力和良好的实验性能而被广泛应用于各个领域,其中包括时间序列预测工作<sup>[13]</sup>。在时序预测工作中最常见的内容包括循环神经网络(recurrent neural network, RNN)和注意力机制。RNN 独有的记忆单元可以捕获历史信息与当前输入的相关性,并且具有多种变体例如门控循环单元(gated recurrent unit, GRU)、长短期记忆网络(long short-term memory, LSTM)用于应对长期依赖问题等。注意力机制能够针对不同特征进行加权变换并为神经网络提供一定可解释性。由于 app 流行度建模本质为探索时序特征与流行度间的关系并完成多步时序预测,所以本文探索分析 app 流行度的多种影响因素,并提出基于注意力机制的多级 GRU 网络,预测未来一段时间 app 的流行度演化。

具体来说,本文设计 3 个模块用于捕获不同层级的相关性,并利用注意力机制来解释不同特征的影响力差异。

1) 时间级自序列相关性.设计基于注意力机制的 GRU 网络 Att-GRU 捕获先前时刻的流行度与目标预测时刻流行度的相关性,利用流行度时序进行时间依赖层级的预测.历史流行度决定了 app 的曝光率和累积人气,对未来流行度产生重要影响.

2) 特征级局部相关性.这是独立于时间层级的特征级模块之一:app 局部特征相关性 Inner-app. Inner-app 用于捕获 app 自身相关特征包括评论、评分和用户情感等与流行度的动态相关性,这些特征代表用户的使用反馈,很大程度代表了已有用户的使用感受,并影响正在浏览的潜在用户是否下载.

3) 特征级全局相关性.作为特征层级内的另一模块,Inner-Class 表征同类功能的其他 app 的流行度序列与目标预测 app 流行度间的相关性,从应用市场全局捕获与目标相关的流行度序列作为动态特征输入.该层级反映了市场竞争对于流行度的影响,同功能 app 中用户只会下载最具吸引力的产品,并且不会把这种情感倾向反映在他们没有下载的 app 的评论中.

概括来说,本文的主要贡献有 3 个方面:

1) 影响因素的分级表示.本文研究了 app 的流行度动态建模与预测问题,并探究了流行度的时间层级和局部、全局特征层级等复杂因素对流行度的动态相关性影响.

2) 多层级注意力网络 DeePOP 的提出.它从以上 2 个层级自适应地捕获最相关的历史状态(特征),并聚合不同的特征表示以预测动态流行度和揭示演化的本质.

3) 真实数据集的验证.实验结果显示,DeePOP 在预测精度上优于现有其他流行度预测方法,有效降低均方根误差至 0.089.

## 1 相关工作

本节回顾与 app 流行度建模与预测最相关的 3 类工作:移动 app 分析、流行度预测和深度学习中的时序预测.

### 1.1 移动 app 分析

app 应用市场提供关于 app 的丰富信息,包括类别、下载量、用户评论和评分等.移动 app 分析工作发现了 app 发展的重要影响特征,并为建模预测等工作提供特征提取的思路.这里着重介绍近年来几篇具有代表性的先进研究.

文献[8]通过分析来识别应用市场中不同 app

至关重要的更新版本,并发现更新版本的描述内容对 app 是否成功起决定性作用;文献[14]着力于优化 app 的新版本发布策略.他们使用更新时间间隔表征发布模式,并确定了包括市场排名、用户评分在内的影响更新结果的多种因素;文献[15]从 app 流行度演化的角度对评论进行分类,其中包括 4 个类别:错误报告、功能请求、用户体验和用户评分;通过分析应用市场中的用户评论和评分,文献[16]明确了用户喜欢或不喜欢某个 app 的原因.他们提出了分析方法以识别用户对不同类型 app 的主要关注点和偏好.

以上利用应用市场数据进行多角度分析的工作为本文开拓思路.上述工作已经阐述了应用市场中特征数据的重要性,但涉及的特征基本都是依靠人工构建,难以挖掘到原始数据之间的隐含关联关系.而对于上述工作中的传统机器学习方法,即使直接利用原始特征数据也存在预测效果不佳、结果不具有解释性等问题.为了更好地建模 app 流行度演化过程并得到优良的预测准确度,本文将特征工程与原始数据挖掘相结合运用,在参考以上工作中着重分析人工特征的同时,采用模型自动学习的方法探索原始数据中流行度(即下载量)在时间层级、个体层级和群组层级 3 方面的隐含关联关系,从而获得高预测精度并挖掘影响流行度演化的复杂因素.

### 1.2 流行度预测

流行度预测一直是研究社交内容演化的重要工作,而 app 流行度预测作为一个新的研究领域,近年来也有研究人员开展了一些开创性的工作.这里先回顾 app 流行度预测的已有工作,再总结有关线上社交内容流行度演化的典型研究方法.

文献[17]将 app 的流行度定义为市场排名、用户评分和用户评论,并提出基于流行度的隐马尔可夫模型 PHMM 来建模和预测 app 流行度量化的 3 个指标;文献[18]考虑不同 app 之间的竞争模式并提出了一个演化层级竞争模型 EHCM 来预测 app 的下载量;文献[19]将流行度定义为下载量的排名,旨在探索 app 的价格、评分与流行度间的关系.通过统计分析,他们发现用户评分与流行度之间存在强相关性,免费 app 的评分显著高于付费 app;文献[20]研究了影响 app 是否存在于苹果应用商店最受欢迎列表中的可能性因素.研究发现免费 app 更有可能在列表中存在,频繁的功能更新和积极正面的评论是保证其排名的关键因素.

关于线上社交内容的流行度预测工作十分丰

富,包括新闻<sup>[9,21]</sup>、微博或推文<sup>[22-24]</sup>和视频<sup>[12,25-26]</sup>。关于新闻,文献[9]使用从新闻内容中提取的特征预测新闻的流行度,其中包括 4 个特征:新闻来源、类别、语言的主观性和提及的命名实体;关于微博,文献[23]提出了一个带有时间映射的扩展加强泊松过程来模拟转发动力学并预测微博的流行度;关于视频,文献[25]提出一种划分流行度阶段的表示模型来描述视频的生命周期,并使用阶段信息来预测未来流行度。

本文所研究的工作区别于以上 2 类流行度预测.对比已有的 app 流行度预测方法,3 点原因促使这项工作还需继续推进:1)预测的准确度仍需提高.现有方法可能受模型的限制,其实验性能仍需进一步提升,力求得到更加接近真实值的预测结果.2)人工特征容易遗失重要信息.面对多模态数据,现有方法多是人工构建一些便于理解的影响特征用于预测,容易丢失原始数据中的隐含信息和关联关系,所以需要设计自动提取特征的方法来全面探索影响流行度演化的原因.3)现有模型方法表达能力不足.如若直接将多模态的海量数据作为输入,现有方法受模型计算能力限制往往难以全面捕获其中的非线性隐含关联,从而影响流行度建模过程的全面性.对比线上社交内容的流行度预测问题,app 具有 2 个特征而不能沿用已有方法的思路:1)app 不受时效约束.新闻、微博等社交内容生命周期十分短暂,随着时间推移其热度必定衰落,生命长度可观测得到.但是 app 生命周期长,随着用户规模的增大其流行度会逐渐上升.长时期的演化过程会受到更多不确定因素影响,因而为传统预测方法带来难度.2)app 自身不断变化.新闻、视频等一旦被发布其内容就无法改变,其流行度的变化更多取决于大众的偏好变化.但 app 只要在架就可以通过更新版本来修复问题、扩展功能,其自身也处于演化之中,所以固定的影响因素难以全面地刻画 app 演化过程,这为预测带来难度.

### 1.3 深度学习中的时序预测

近年来越来越多的研究尝试利用深度学习方法进行时间序列的预测工作,在诸如空气污染预测<sup>[13]</sup>、人群移动预测<sup>[27]</sup>和共享单车站桩流量预测<sup>[28-29]</sup>等方面取得成功.这里回顾深度学习中最常用于时序预测的神经网络 RNN 的应用实例.文献[30]设计考虑周期性的卷积循环神经网络以准确捕获时空相关性,学习并结合显式周期性表示,并且可以通过多步骤优化来优化预测;文献[27]使用残差

神经网络来模拟人群流量的时间接近度、周期和趋势属性,并为不同区域分配不同权重,从而实现不同区域间人群移动情况的预测;文献[31]提出了一个基于注意力机制的双阶段循环神经网络,用以适当地捕获长期时间依赖性并选择相关的输入特征,该模型在股票价格预测上显示出良好性能.

以上方法为 app 流行度预测提供了研究思路.首先,app 的流行度及其影响特征本身均为动态时间序列,当前时刻的表现必然受历史表现的影响,这体现从时间层级捕获相关性的重要性.其次,app 是否受欢迎既受自身功能、口碑影响,又因同类竞争 app 的表现而产生波动,此消彼长的现象广泛存在,这体现不同特征层级相关性的重要影响.另外,面对较长历史时间影响和众多特征影响时,引入注意力机制的 RNN 网络能够自适应地提取相关特征,解决了 RNN 难以捕获长时间依赖等问题.

## 2 问题概述

本节主要阐述 app 流行度建模的基本问题,提出 app 流行度和流行度预测的问题定义,分析不同影响特征与流行度序列的相关性.

### 2.1 问题设定

本节提出 app 流行度形式化定义,并介绍用以建模的影响特征的含义.

**定义 1.** app 流行度.给定 1 组 app  $A = \{a_1, a_2, \dots, a_N\}$ ,每个 app  $a_i (i = 1, 2, \dots, N)$  的流行度为具有  $N$  个时间点  $T = \{1, 2, \dots, N\}$  的序列.app  $a_i$  在第  $t (t \in T)$  天的流行度  $p_i(t)$  量化为第  $t$  天的下载量,则 app  $a_i$  的流行度演化序列为  $P_i = \{p_i(1), p_i(2), \dots, p_i(N)\}$ .

app 流行度受众多因素影响,包括 app 历史流行度表现、用户评论和市场竞争等,这些因素可以量化为具有  $N$  个时间点  $T = \{1, 2, \dots, N\}$  的不同特征序列.

1) 历史流行度.对于 app  $a_i$  的流行度演化序列  $P_i = \{p_i(1), p_i(2), \dots, p_i(N)\}$ , 第  $t$  天流行度  $p_i(t)$  可能受  $P_i^t = \{p_i(1), p_i(2), \dots, p_i(t-1)\}$  或其子序列的影响.如果 app 没有受到其他外部因素影响,之前受欢迎的 app 在未来也会受欢迎,遵循“富者更富”的思想.

2) 评论量.对于 app  $a_i$ ,其在  $T = \{1, 2, \dots, N\}$  的  $N$  天的每日评论数量可以表示为  $V_i = \{V_i(1), V_i(2), \dots, V_i(N)\}$ .评论数量在一定程度上反映了

app 的受关注度,一般来说评论数量越多,该 app 越受欢迎。

3) 评分.对于 app  $a_i$ ,其在  $T = \{1, 2, \dots, N\}$  的  $N$  天的每日平均评分可以表示为  $S_i = \{S_i(1), S_i(2), \dots, S_i(N)\}$ . 评分能够直观反映用户对 app 的态度和使用感受,高评分代表了高受欢迎程度。

4) 评论情感.对于 app  $a_i$ ,其在  $T = \{1, 2, \dots, N\}$  的  $N$  天的每天中的评论情感可以被量化为具体数值,表示为  $M_i = \{M_i(1), M_i(2), \dots, M_i(N)\}$ . 评论情感得分表达了用户对 app 的喜爱程度,得分越高代表用户对 app 的使用感受越具积极性和乐观性。

5) 市场竞争.app  $a_i$  在应用市场中的功能类别为  $l_j$ ,对于同类别标签的 1 组 app  $A = \{a_1, a_2, \dots, a_N\}$  内,  $a_i$  的流行度序列  $P_i$  受到其他 app 的流行度  $C = \{P_1, P_2, \dots, P_N\}$  的影响.当 app 的功能类别相同时,用户只会选择其中有限个下载使用.同类产品中某些 app 的流行度升高将为其他 app 带来竞争冲击。

**定义 2.** app 流行度预测.对于 app  $a_i$ ,已知其历史流行度和其他特征数据,预测  $a_i$  在未来  $t$  天的流行度序列  $P_i^t = \{p_i(N+1), p_i(N+2), \dots, p_i(N+t)\}$ .

在本文的 app 流行度建模中,上述不同特征被输入不同层级中以捕获特征与流行度间的依赖关系,为预测提供依据.具体来说,时间级自序列模块捕获历史流行度与待预测值的相关性,局部特征级模块捕获 app 自身影响因素评论量、评分和情感与流行度的隐含关联,全局特征级模块编码应用市场中同类产品间的竞争关系,捕获此消彼长的互斥关联。

## 2.2 特征相关性分析

为了验证不同特征与流行度之间的相关性,采用最大信息系数(maximal information coefficient, MIC)<sup>[32]</sup> 来衡量以上单特征与流行度之间的相关性大小.MIC 是 2011 年发表在“Science”上的一篇文章提出的一种可用于度量单变量与相应变量之间的非线性关系的算法。

### 2.2.1 局部特征相关性

通过计算评论量、评分和评论情感与流行度之间的 MIC,从而分析得到不同粒度下局部特征对流行度的影响.在应用市场中,app 具有不同的功能标签,功能标签分为一级标签和二级标签,一级标签指示更大范围的功能类别,其下包括若干二级功能标

签.表 1 是应用市场中的标签示例.一般来说,同一级标签的 app 具有类似的市场环境和应用需求,可能在流行度演化方式上类似,所以先根据一级标签对 app 分类,计算每类中局部特征的相关性均值.相关性计算结果如图 1 所示.

Table 1 Level 1 and Level 2 Labels in App Market

表 1 应用市场中一级、二级标签示例

Level 1 Label	Level 2 Label
社交	电话通讯、婚恋、聊天、社区
旅游	导航、购票、公交、租车、住宿、攻略
:	:

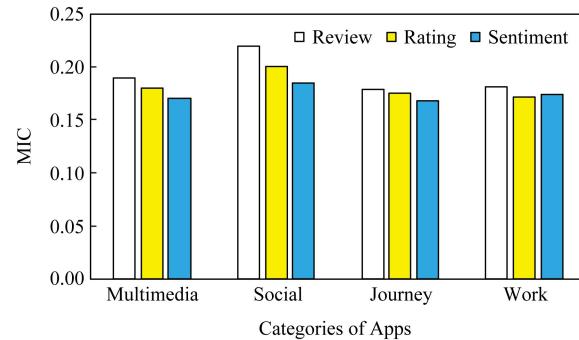


Fig. 1 Correlation between local features and popularity of first label apps

图 1 不同一级标签 app 的局部特征与流行度间相关性

从图 1 分析可得:1)对于不同一级标签,app 的局部特征相关性差异较小,基本都处于 0.17~0.21 之间.3 种特征均来自于用户的使用反馈,所反映的内容具有相似性.2)3 种局部特征在同一类 app 中相关性差异不大,评论量的相关系数略高于评分和情感,可能是因为评论量的增长可以间接理解为用户下载量增大,流行度呈上升趋势,而评分和情感是来自使用 app 一段时间的用户反馈,其对潜在用户的影响较小.3)局部特征的相关系数均较小,说明局部特征不足以刻画流行度演化过程,流行度的变化不止受用户使用感受影响.

为了细化局部特征与流行度之间的关联关系,图 2 以多媒体类 app 为例,详细表示了各二级标签之间局部特征相关系数的不同.图 2 中,横坐标为多媒体类 app 的二级标签,这里计算了各二级标签下所有 app 的平均每日流行度,并从左至右依次递增排列,即卡拉OK 类 app 流行度最低,视频类 app 的流行度最高.随着流行度的逐渐升高,局部特征的相关系数出现非线性变化,说明说明流行度高的 app 其特征相关性不一定强,流行度的建模需要考虑计

算性能良好的非线性模型.另外可观察得出,不同特征的相关系数在流行度变化时的变化趋势基本一致,这也表明流行度的变化影响因素复杂,只捕获用户反馈进行特征工程难以良好预测流行度演化过程.

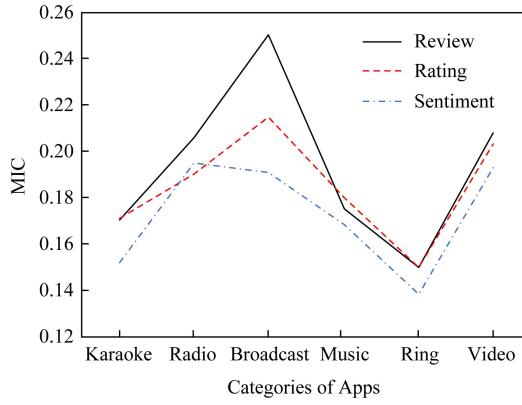


Fig. 2 Correlation between local features and popularity of second label apps

图 2 不同二级标签 app 的局部特征与流行度间相关性

### 2.2.2 全局特征相关性

通过计算全局特征即同一二级标签下其他 app 的流行度与某一特定 app 之间的 MIC 系数,从而分析全局特征对流行度的影响.为了与图 2 形成对比,同样选定多媒体类 app 下的二级标签 app,计算其与全局特征的 MIC 差异,并选定 MIC 最高的前 3 个特征呈现如图 3 所示.相较于局部特征的相关系数,全局特征中的前 3 名相关系数均有大幅度提升,但随着流行度的增加,变化没有规律性,不具备线性关系.可见本文设定的全局特征较好地捕获了市场竞争与流行度演化间的依赖关系.这也证实了相似功能的 app 间的市场竞争确实存在,且与 app 形成竞争的对象并不唯一.

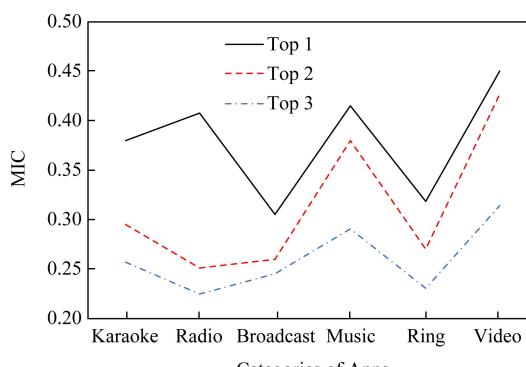


Fig. 3 Correlation between top 3 global features and popularity of second label apps

图 3 app 的前 3 个全局特征与流行度相关性

## 3 多层级注意力网络

本节将详细介绍 DeePOP 模型组成和各模块构成.图 4 显示了 DeePOP 的整体架构.它包含 3 类输入:历史流行度演化序列、局部特征和全局特征,分别对应于时间级自序列模块、特征级局部和全局模块.每个模块由含有注意力机制的 GRU 进行编码,并在输入中自适应地选择最相关的历史状态或者相关特征.在预测多步流行度时,连接先前 3 个模块的输出并通过多层感知机输出 app 流行度的预测结果.模型中的不同编码模块可以使用标准反向传播共同训练.

### 3.1 时间级自序列模块

该模块主要由具有时间注意力机制的 GRU 组成,从而能够在输入的时间步中自适应地选择最相关的历史状态.对于本节涉及的 app 流行度序列预测问题,经典的差分整合移动平均自回归模型<sup>[17]</sup>(autoregressive integrated moving average model, ARIMA)及其相关变体并不适用.ARIMA 算法适用于表现稳定的序列数据,本质上只能捕获数据间的线性关系.而 app 流行度受多种可变因素影响,自身序列变化复杂且目前尚未有研究得到其固定的变化规律,所以本文采用 GRU 与注意力机制结合的算法来捕获流行度序列的自相关性.GRU 作为 RNN 变体之一,既能够学习当前时间状态对于历史状态的非线性依赖,相比于 LSTM 又具有参数更少、能够较快收敛的优势.注意力机制的引入为当前状态对历史状态的依赖性大小提供一定的可解释性.所以本文在综合考虑以上问题后,设计了时间级子序列模块用以捕获历史状态对流行度的影响作用.具体来说,基于隐藏状态  $z_{t-1}$  和输入状态  $y_{t-1}$  可得当前时刻输入的注意力权重:

$$e_t^i = \mathbf{W}_P^T \tanh(\mathbf{W}_P z_{t-1} + \mathbf{U}_P y_i + \mathbf{b}_P), 1 \leq i \leq T. \quad (1)$$

为了保证所有注意力权重加和为 1,这里使用 Softmax 函数对式(1)进行归一化.

$$\alpha_t^i = \frac{\exp(e_t^i)}{\sum_{j=1}^T \exp(e_t^j)}. \quad (2)$$

在得到注意力权重之后,就可以计算得到时刻  $t$  的输出权重数值为

$$c_{yt} = \sum_{i=1}^T \alpha_i y_i. \quad (3)$$

之后采用 GRU 来学习时间自序列的表示.GRU 由重置门和更新门来控制信息的更新状态.重

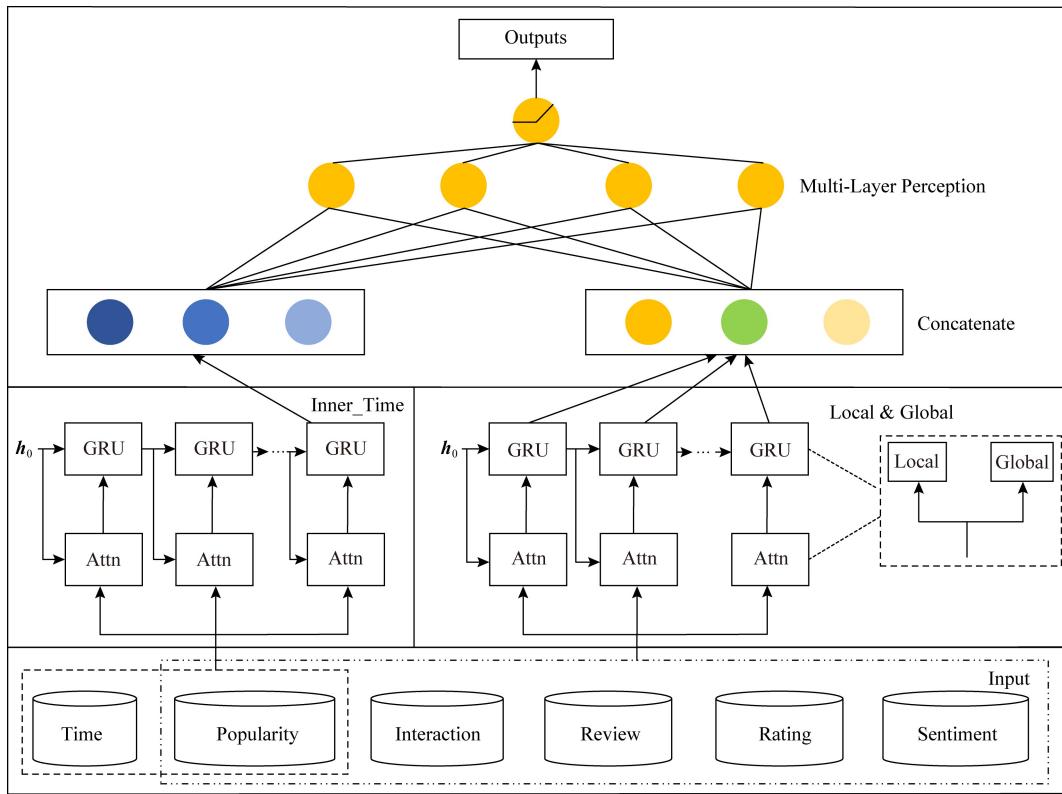


Fig. 4 Framework of DeePOP

图 4 DeePOP 架构图

置门控制历史状态对候选状态的贡献程度,更新门决定保留历史信息成分和添加新信息.GRU 的更新过程为

$$\begin{aligned} r_t &= \sigma(\mathbf{W}_r[\mathbf{h}_{t-1}, \mathbf{c}_t] + \mathbf{b}_r), \\ u_t &= \sigma(\mathbf{W}_u[\mathbf{h}_{t-1}, \mathbf{c}_t] + \mathbf{b}_u), \\ \tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_h[r_t \cdot \mathbf{h}_{t-1}, \mathbf{c}_t] + \mathbf{b}_h), \\ \mathbf{h}_t &= (1 - u_t) \cdot \mathbf{h}_{t-1} + u_t \cdot \tilde{\mathbf{h}}_t. \end{aligned} \quad (4)$$

### 3.2 特征级局部模块

为了捕获不同局部特征对于流行度影响程度的动态变化,该模块编码特征表示并通过 Att-GRU 自适应选择相关特征,以提高流行度的预测准确度.通过对隐藏状态  $\mathbf{z}_{t-1}$  和当前输入状态  $x_{t-1}$  编码,可得当前时刻局部特征的注意力权重:

$$\mathbf{u}_t^k = \mathbf{V}_L^T (\mathbf{W}_L \mathbf{z}_{t-1} + \mathbf{U}_L \mathbf{x}_t^k + \mathbf{b}_L), 1 \leq k \leq n. \quad (5)$$

归一化权重系数得:

$$\beta_t^k = \frac{\exp(u_t^k)}{\sum_{i=1}^n \exp(u_t^i)}. \quad (6)$$

在得到注意力权重之后,上下文向量可以计算得到:

$$\mathbf{c}_L = (\beta_t^1 \mathbf{x}_t^1, \beta_t^2 \mathbf{x}_t^2, \dots, \beta_t^n \mathbf{x}_t^n)^T. \quad (7)$$

同样,这里的输出向量将作为式(4)中 GRU 的输入进行计算,这里不再赘述.

### 3.3 特征级全局模块

对于全局特征,同样适用 Att-GRU 自适应地捕获当前时刻最相关的全局特征进行编码,通过对隐藏状态  $\mathbf{z}_{t-1}$  和当前输入状态  $x_{t-1}$  编码,可得当前时刻全局特征的注意力权重:

$$g_t^o = \mathbf{V}_G^T (\mathbf{W}_G \mathbf{z}_{t-1} + \mathbf{U}_G \mathbf{x}_t^o + \mathbf{b}_G), 1 \leq o \leq m. \quad (8)$$

对权重进行归一化得:

$$\gamma_t^o = \frac{\exp(g_t^o)}{\sum_{i=1}^m \exp(g_t^i)}. \quad (9)$$

在得到注意力权重之后,上下文向量可以计算得到:

$$\mathbf{c}_G = (\gamma_t^1 \mathbf{x}_t^1, \gamma_t^2 \mathbf{x}_t^2, \dots, \gamma_t^m \mathbf{x}_t^m)^T. \quad (10)$$

之后将得到的向量输入到式(4)中进行 GRU 计算.

### 3.4 模型输出与参数学习

在预测过程中,通过将以上多个模块的输出进行聚合并馈送至多层感知机(multi-layer perceptron, MLP)<sup>[33]</sup>中以聚合不同编码,从而获得多步流行度预测值,其中的参数均可通过学习获得.

此外,DeePOP 模型包含 2 类可学习参数:网络层连接系数( $\mathbf{W}_*$ , $\mathbf{b}_*$ )和注意力机制参数( $\mathbf{V}_*$ , $\mathbf{W}_*$ , $\mathbf{U}_*$ , $\mathbf{b}_*$ ),它们可以通过反向传播学习得到。实验中,可以通过调整一系列超参数包括学习率、网络层数和神经元数等来提高模型预测准确度。模型的损失函数为平均绝对误差(mean absolute error, MAE):

$$R_{\text{MAE}} = \frac{1}{T} \sum_{t=1}^T |\hat{y}_t - y_t|. \quad (11)$$

其中,  $\hat{y}_t$  表示流行度预测值,  $y_t$  表示流行度实际值。

## 4 实验与结果

本节将针对实验数据集、对比方法和评价指标等进行介绍,并通过实验结果分析评估 DeePOP 的预测性能。

### 4.1 数据集

本文从包括华为应用商店、应用宝、豌豆荚和小米应用商店等 9 个当下热门的应用市场获取了共 12 类一级标签的 app 各项数据<sup>①</sup>,包括名称、一级和二级标签、日下载量和用户评论等。为了综合考虑社交媒体中 app 的评论信息,这里以不同 app 名称作为关键词爬取了微博中不同 app 的评论信息,并按照时间标签和 app 标签与应用市场中的评论进行合并。各项数据的时间跨度为 2017-01-01—2018-01-01。表 2 是共计 1023 个 app 的数据统计。

Table 2 Parameter Index of the Dataset

表 2 数据集的参数指标

Level 1 Label	App Number	Level 2 Label Number	Average Download	Average Review
多媒体	94	6	48 025 253	2 344
社交	34	4	92 650 406	3 543
新闻	131	6	33 216 714	1 805
教育	61	6	7 628 510	2 232
金融	45	8	7 480 136	637
旅游	88	6	41 033 109	1 870
办公	139	5	5 768 262	494
照相	79	6	17 037 972	3 075
购物	120	5	38 901 915	1 371
娱乐	85	8	9 497 363	945
健康	66	5	3 577 440	428
育儿	81	6	3 368 246	793

对于应用市场获取的数据,需要预处理为能够输入至模型的标准格式。首先,针对 app 的下载量序列需要归一化成流行度演化序列的标准格式。而局部特征的构造则是针对 app 的评论文本进行计数统计、评分归一化和 SnowNLP<sup>②</sup> 情感分析,分别得到评论量、评分和情感 3 类局部特征。对于全局特征并不需要再进行特殊处理,只要提取与目标 app 同二级标签下其他 app 的流行度演化序列作为输入即可。

### 4.2 对比模型

这里选用了当前最新的 app 流行度预测模型和 DeePOP 模型的变体(部分架构)与 DeePOP 模型进行对比,这些方法包括:

1) 支持向量回归(support vector regression, SVR)模型<sup>[34]</sup>,使用影响特征训练支持向量机以完成回归任务。

2) 隐 Markov 模型 PHMM<sup>[17]</sup>是文献[17]建模 app 流行度信息的方法。要将 PHMM 应用于本文数据集,应首先将数据离散,用离散范围替换具体数值;然后应用 PHMM 来预测未来的流行度;最后,通过比较流行度的真实范围和预测的范围来测量 PHMM 的预测准确度。

3) 演化分级竞争模型 EHCM<sup>[18]</sup>,模拟 app 之间随时间演化的分级竞争并预测未来的下载量。为了将该方法应用于本文数据集,首先对 app 之间的演化竞争进行建模,其次预测未来的 app 下载量。

4) LSTM 模型,结构较 GRU 复杂,参数数量更多。这里采用较简单的 LSTM 模型,不包含 3 个层级模块,采用与 DeePOP 模型相同的输入特征。

5) DeePOP-Ti 模型,即不包含时间级自序列模块的模型。

6) DeePOP-Lo 模型,即不包含特征级局部模块的模型。

7) DeePOP-GI 模型,即不包含特征级全局模块的模型。

8) DeePOP-Se 模型,即不包含用户情感特征的模型。

另外,用于评价 DeePOP 模型和其他对比方法的预测性能,选取 2 种评估指标:

1) 均方根误差(root mean square error, RMSE)<sup>[35]</sup>。通过公式计算预测的流行度与实际的流行度之间的差异。RMSE 越小意味着流行度预测性能越好。

① <http://www.kuchuan.com/>

② <https://github.com/isnowfy/snownlp>

$$R_{\text{RMSE}} = \sqrt{\frac{1}{n} \sum_{t=1}^n (p_t - \hat{p}_t)^2}. \quad (12)$$

2) MAE<sup>[35]</sup>.通过公式计算预测的流行度和实际的流行度之间的 MAE.MAE 越小意味着预测性能越好.

### 4.3 实验结果

#### 4.3.1 不同模型的对比

所有 app 利用不同方法预测流行度的 RMSE 如表 3 所示.由图观察得知,DeePOP 模型的计算误差低于所有对比方法,预测性能最优.对比于其他方法可知:1)与现有 app 流行度预测方法相比,DeePOP 模型具有更强的计算性能,能够自适应捕获复杂的非线性关系,其中包括流行度演化的历史依赖和隐含特征的相关关系.2)与不含层级的 LSTM 模型相比,DeePOP 模型的优势在于它从不同层级捕获特征与流行度的相关性,并且能在不同时刻自适应提取相关性强的特征用于编码计算,而面对多特征输入,LSTM 模型在计算过程中容易遗忘和丢失信息,且受到运算时间的限制.

Table 3 Experimental Results of Different Models

表 3 不同模型的实验结果

Models	RMSE	MAE
SVR	0.138	0.095
EHCM	0.125	0.072
PHMM	0.116	0.065
LSTM	0.109	0.063
DeePOP-Ti	0.113	0.062
DeePOP-Gl	0.102	0.056
DeePOP-Lo	0.095	0.052
DeePOP-Se	0.091	0.050
DeePOP	0.089	0.047

DeePOP 模型变体的预测性能能够侧面反映不同层级对于建模流行度的重要性.可以看出,缺失全局特征模块的模型 DeePOP-Gl 和缺失局部特征模块的模型 DeePOP-Lo 的总体预测误差略低于缺失时间自序列模块的模型 DeePOP-Ti,而 DeePOP-Lo 模型的预测误差略低于 DeePOP-Gl 模型,更加接近 DeePOP 模型的预测性能.这与第 2 节中 MIC 的计算结果相吻合,说明对于逐渐同质化的 app 来说,市场竞争更加影响流行度,用户不选择下载使用某款 app 更可能是因为他们有更好的选择,而不是因为 app 自身性能的优劣等.其中 DeePOP-Se 模型变体的预测效果最贴近 DeePOP 模型,可能的原因是仅

仅去除用户情感得分特征并不会对最终的流行度预测结果产生重大影响,DeePOP 模型更加关注多粒度特征在多级建模后对流行度的影响.

#### 4.3.2 特征层级的注意力机制

这里以视频类 app 为例,利用全局特征的注意力机制解释不同特征在预测过程中的权重影响,为预测模型提供可解释性.图 5 显示了多媒体类中二级标签为视频类的 10 款 app 在参与预测其他 app 流行度时的注意力权重.如图 5 所示,纵坐标显示了二级标签中 10 款热门视频软件,即预测流行度的目标 app.对于每款软件其横轴代表了该软件在全局特征中注意力权重排在前 10 名的其他软件.所以横坐标的 app 序号只在对应每款视频软件(诸如爱奇艺)时特指某 app.观察图 5 可知,不同 app 的市场竞争现象各有不同.以爱奇艺 iQIYI 为例,其存在市场竞争现象,且竞争较为激烈的是 app 是优酷视频、哔哩哔哩动画和迅雷视频.对于 app 乐视视频 LETV 来说,其他同类 app 竞争较小,说明存在其他主要因素影响 app 的流行度演化过程.由于篇幅所限,这里不再列举其他 app 以及与其竞争的 app 的注意力权重.由此可见,注意力机制的引入能够在不同时步自适应地捕获最相关特征,从而为理解模型预测结果提供一定的解释性.

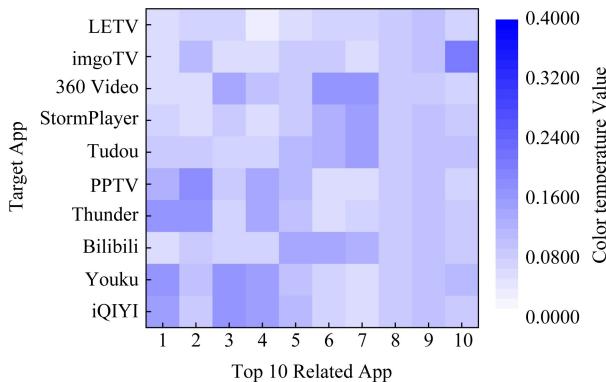


Fig. 5 Attention weight of video apps

图 5 视频类 app 的注意力权重

#### 4.3.3 DeePOP 模型不同变体的对比

为了更加细致地显示不同变体的作用和影响,分别计算不同一级标签的 app 在不同变体模型下的预测性能,结果如图 6 所示.

在图 6 中,不同 DeePOP 模型变体的模型在 12 类一级标签的 app 下预测性能差异稳定,对于新闻、教育等类别的 app,不同变体的预测性能都表现良好,低于平均值;而对于金融、购物等类别的 app,不

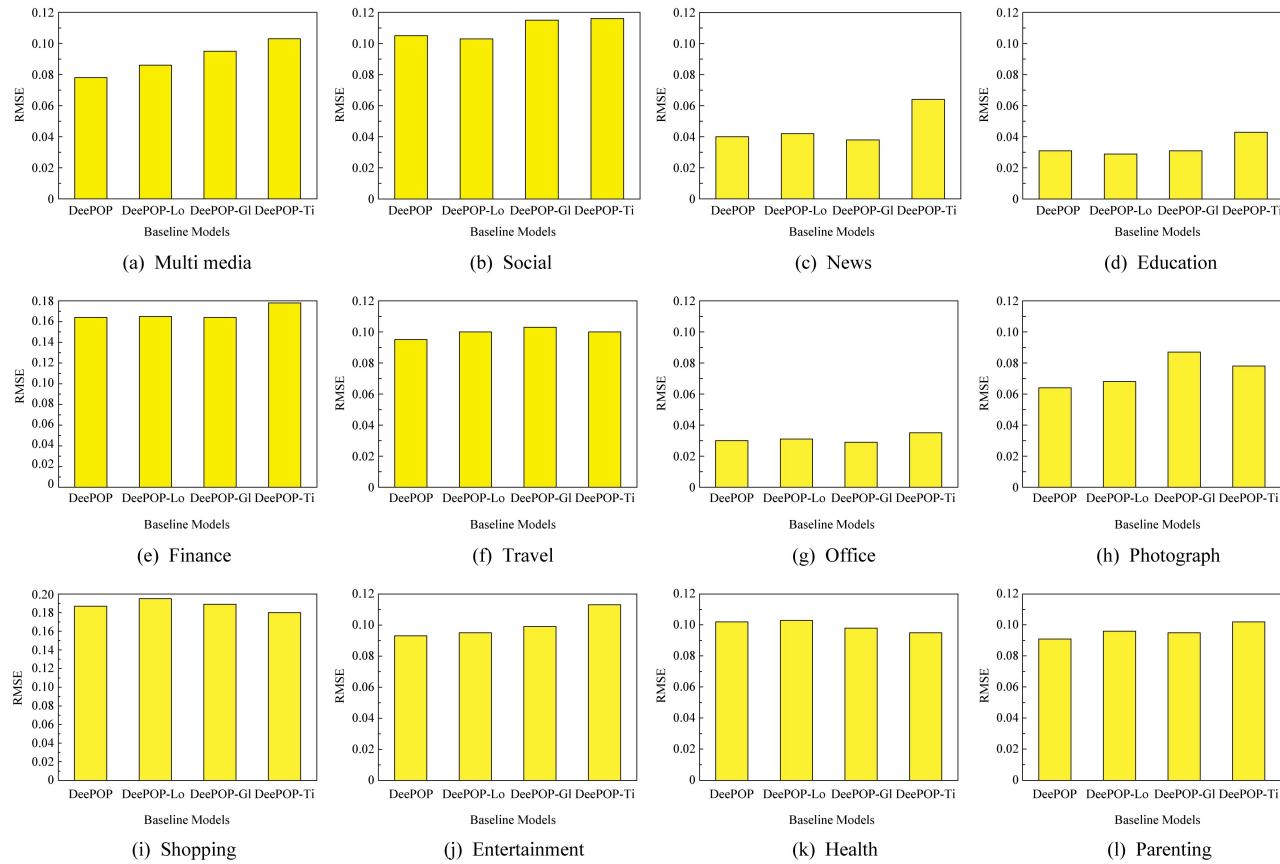


Fig. 6 The prediction performance of variants

图 6 DeePOP 变体模型的预测性能

同变体的性能都表现一般。另外,在不同类别的 app 中,变体之间的性能差异在一定程度上反映了不同特征对流行度演化的影响和作用不同。例如在照相类 app 中,性能较好的 DeePOP-GI 模型反而误差最高,这可能是因为照相类 app 产品间同质化现象较为严重,存在激烈的市场竞争,流行度演化的过程受 app 自身特征的影响较小。但从整体来看,DeePOP 模型仍然在各类 app 中获得最优预测性能。

## 5 总 结

本文提出了一种多层次注意力网络以进行 app 流行度预测,并针对复杂影响因素进行分级建模。通过时间级自序列模块捕获对历史流行度的长期依赖,利用特征级局部、全局模块构建影响特征与流行度的非线性关系。同时,注意力机制的引入为不同模块提供自适应能力,以捕获与流行度变化最相关的历史状态并针对预测结果提供一定的解释。实验结果表明,该方法能够快速高效地进行 app 流行度建模与预测。

## 参 考 文 献

- [1] Baeza-Yates R, Jiang Di, Silvestri F, et al. Predicting the next app that you are going to use [C] //Proc of the 8th ACM Int Conf on Web Search and Data Mining. New York: ACM, 2015: 285–294
- [2] Wang Yingzi, Yuan N, Sun Yu, et al. App download forecasting: An evolutionary hierarchical competition approach [C] //Proc of the Int Joint Conf on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2017: 2978–2984
- [3] Ouyang Yi, Guo Bin, Lu Xinjiang, et al. Competitivebike: Competitive analysis and popularity prediction of bike-sharing apps using multi-source data [J]. IEEE Transaction on Mobile Computing, 2019, 18(8): 1760–1773
- [4] Ouyang Yi, Guo Bin, Guo Tong, et al. Modeling and forecasting the popularity evolution of mobile apps: A multivariate hawkes process approach [C] //Proc of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. New York: ACM, 2018: 182:1–182:23
- [5] Lu Xuan, Chen Zhenpeng, Liu Xuanzhe, et al. Prado: Predicting app adoption by learning the correlation between developer-controllable properties and user behaviors [C] // Proc of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. New York: ACM, 2017: 79:1–79:30

- [6] Lin J, Sugiyama K, Kan M, et al. Addressing cold-start in app recommendation: Latent user models constructed from Twitter followers [C] //Proc of the 36th Int Conf on Research and Development in Information Retrieval. New York: ACM, 2013: 283–292
- [7] Yin Peifeng, Luo Ping, Lee W, et al. App recommendation: A contest between satisfaction and temptation [C] //Proc of the 6th ACM Int Conf on Web Search and Data Mining. New York: ACM, 2013: 395–404
- [8] Martin W, Sarro F, Harman M. Causal impact analysis for app releases in Google play [C] //Proc of the 24th ACM SIGSOFT Symp on Foundations of Software Engineering. New York: ACM, 2016: 435–446
- [9] Bandari R, Asur S, Huberman B. The pulse of news in social media: Forecasting popularity [C] //Proc of the 6th Int AAAI Conf on Web and Social Media. Menlo Park, CA: AAAI, 2012: 26–33
- [10] Zhao Qingyuan, Erdogdu M, He H, et al. Seismic: A self-exciting point process model for predicting tweet popularity [C] //Proc of the 21st ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2015: 1513–1522
- [11] Khosla A, Sarma A, Hamid R. What makes an image popular [C] //Proc of the 23rd Int Conf on World Wide Web. New York: ACM, 2014: 867–876
- [12] Rizoiu M, Xie Lexing, Sanner S, et al. Expecting to be hip: Hawkes intensity processes for social media popularity [C] //Proc of the 26th Int Conf on World Wide Web. New York: ACM, 2017: 735–744
- [13] Liang Yuxuan, Ke Songyu, Zhang Junbo, et al. Geoman: Multi-level attention networks for geo-sensory time series prediction [C] //Proc of the Int Joint Conf on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2018: 3428–3434
- [14] Shen Sheng, Lu Xuan, Hu Ziniu, et al. Towards release strategy optimization for apps in google play [C] //Proc of the 9th Asia-Pacific Symp on Internetwork. New York: ACM, 2017: 1:1–1:10
- [15] Maalej W, Nabil H. Bug report, feature request, or simply praise? On automatically classifying app reviews [C] //Proc of the 23rd IEEE Int Requirements Engineering Conf. Piscataway, NJ: IEEE, 2015: 116–125
- [16] Fu Bin, Lin Jiali, Li Lei, et al. Why people hate your app: Making sense of user feedback in a mobile app store [C] //Proc of the 19th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2013: 1276–1284
- [17] Zhu Hengshu, Liu Chuanren, Ge Yong, et al. Popularity modeling for mobile apps: A sequential approach [C] //Proc of the IEEE Transactions on Cybernetics. Piscataway, NJ: IEEE, 2015: 1303–1314
- [18] Wang Yingzi, Yuan N, Sun Yu, et al. App download forecasting: An evolutionary hierarchical competition approach [C] //Proc of the Int Joint Conf on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2017: 2978–2984
- [19] Finkelstein A, Harman M, Jia Yue, et al. Investigating the relationship between price, rating, and popularity in the blackberry world app store [J]. Information and Software Technology, 2017, 87: 119–139
- [20] Lee G, Raghu S. Determinants of mobile apps' success: Evidence from the app store market [J]. Journal of Management Information Systems, 2014, 31(2): 133–170
- [21] Castillo C, El-Haddad M, Pfeffer J, et al. Characterizing the life cycle of online news stories using social media reactions [C] //Proc of the 17th ACM Conf on Computer Supported Cooperative Work & Social Computing. New York: ACM, 2014: 211–223
- [22] Bao Peng. Modeling and predicting popularity dynamics via an influence-based self-excited hawkes process [C] //Proc of the 25th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2016: 1897–1900
- [23] Gao Shuai, Ma Jun, Chen Zhumin. Modeling and predicting retweeting dynamics on microblogging platforms [C] //Proc of the 8th ACM Int Conf on Web Search and Data Mining. New York: ACM, 2015: 107–116
- [24] Matsubara Y, Sakurai Y, Prakash B, et al. Rise and fall patterns of information diffusion: Model and implications [C] //Proc of the 18th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2012: 6–14
- [25] Yu Honglin, Xie Lexing, Sanner S, et al. The lifecycle of a Youtube video: Phases, content and popularity [C] //Proc of the 9th Int AAAI Conf on Web and Social Media. Menlo Park, CA: AAAI, 2015: 533–542
- [26] Pinto H, Almeida J, Gonçalves M. Using early view patterns to predict the popularity of Youtube videos [C] //Proc of the 6th ACM Int Conf on Web Search and Data Mining. New York: ACM, 2013: 365–374
- [27] Feng Jie, Li Yong, Zhang Chao, et al. Deepmove: Predicting human mobility with attentional recurrent networks [C] //Proc of the 23rd Int Conf on World Wide Web. New York: ACM, 2018: 1459–1468
- [28] Wang Shuai, He Tian, Zhang Desheng, et al. Bravo: Improving the rebalancing operation in bike sharing with rebalancing range prediction [C] //Proc of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. New York: ACM, 2018: 44:1–44:22
- [29] Liu Junming, Sun Leilei, Li Qiao, et al. Functional zone based hierarchical demand prediction for bike system expansion [C] //Proc of the 23rd ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2017: 957–966

- [30] Zonoozi A, Kim J, Li Xiaoli, et al. Periodic-CRN: A convolutional recurrent model for crowd density prediction with recurring periodic patterns [C] //Proc of the IEEE Int Joint Conf on Artificial Intelligence. Piscataway, NJ: IEEE, 2018: 3732–3738
- [31] Qin Yao, Song Dongjin, Chen Haifeng, et al. A dual-stage attention-based recurrent neural network for time series prediction [C] //Proc of the IEEE Int Joint Conf on Artificial Intelligence. Piscataway, NJ: IEEE, 2017: 2627–2633
- [32] Reshef N, Reshef A, Finucane K, et al. Detecting novel associations in large data sets [J]. Science, 2011, 334 (6062): 1518–1524
- [33] Yi Xiuwen, Zhang Junbo, Wang Zhaoyuan, et al. Deep distributed fusion network for air quality prediction [C] // Proc of the 24th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2018: 965–973
- [34] Smola J, Schölkopf B. A tutorial on support vector regression [J]. Statistics and Computing, 2004, 14(3): 199–222
- [35] Bao Yukun, Xiong Tao, Hu Zhongyi. Multi-step-ahead time series prediction using multiple-output support vector regression [J]. Neurocomputing, 2014, 129: 482–493



**Zhang Yixuan**, born in 1994. Master candidate at the college of Computer Science, Northwestern Polytechnical University. Her main research interests include mobile social network, ubiquitous computing.



**Guo Bin**, born in 1980. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include ubiquitous computing, social and community intelligence, urban big data mining, mobile crowd sensing and human-computer interaction.



**Liu Jiaqi**, born in 1991. PhD, associate professor. Member of CCF. Her main research interests include modeling, analysis and algorithm design in social networks.



**Ouyang Yi**, born in 1994. PhD candidate at the College of Computer Science, Northwestern Polytechnical University. His main research interests include mobile social network, big data processing and ubiquitous computing.



**Yu Zhiwen**, born in 1977. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include pervasive computing, context aware systems, personalization, recommendation technology, mobile social networks and multimedia intelligent service.