

单位代码: 10359  
学 号: 2014111090

分类号: TP301  
密 级: 公开



合肥工业大学

Hefei University of Technology

# 硕士学位论文

MASTER DEGREE THESIS

论文题目: 基于动态自适应权重的个性化微博推荐系统研究

学位类别: 学历硕士

学科专业:  
(工程领域) 管理科学与工程

作者姓名: 徐玉祥

导师姓名: 姜元春 副教授

完成时间: 2017 年 3 月

单位代码：10359

学 号：2014111090

密 级：公开

分类号：TP301

合肥工业大学  
Hefei University of Technology

# 硕士学位论文

## MASTER'S DISSERTATION

论文题目：基于动态自适应权重的个性化

微博推荐系统研究

学位类别：学历硕士

专业名称：管理科学与工程

作者姓名：徐玉祥

导师姓名：姜元春 副教授

完成时间：2017 年 3 月

合 肥 工 业 大 学



学历硕士学位论文

基于动态自适应权重的个性化微博推荐  
系统研究

作者姓名：\_\_\_\_\_徐玉祥\_\_\_\_\_

指导教师：\_\_\_\_\_姜元春 副教授\_\_\_\_\_

学科专业：\_\_\_\_\_管理科学与工程\_\_\_\_\_

研究方向：\_\_\_\_\_信息管理与信息系统\_\_\_\_\_

2017 年 3 月

A Dissertation Submitted for the Degree of Master

**Research on Dynamic Self-adaptive Feature Weighting  
for Personalized Micro-blog Recommendation System**

By

Xu Yuxiang

Hefei University of Technology

Hefei, Anhui, P.R.China

March, 2017

# 合 肥 工 业 大 学

本论文经答辩委员会全体委员审查，确认符合合肥工业大学学历硕士学位论文质量要求。

答辩委员会签名（工作单位、职称、姓名）

主席：汪传雷 安徽大学 教授

委员：胡一建 合肥工业大学 教授

江兵 合肥工业大学 教授

导师：姜元春 合肥工业大学 副教授

## 学位论文独创性声明

本人郑重声明：所呈交的学位论文是本人在导师指导下进行独立研究工作所取得的成果。据我所知，除了文中特别加以标注和致谢的内容外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得合肥工业大学或其他教育机构的学位或证书而使用过的材料。对本文成果做出贡献的个人和集体，本人已在论文中作了明确的说明，并表示谢意。

学位论文中表达的观点纯属作者本人观点，与合肥工业大学无关。

学位论文作者签名：

徐玉祥

签名日期：

2017年4月18日

## 学位论文版权使用授权书

本学位论文作者完全了解合肥工业大学有关保留、使用学位论文的规定，即：除保密期内的涉密学位论文外，学校有权保存并向国家有关部门或机构送交论文的复印件和电子光盘，允许论文被查阅或借阅。本人授权合肥工业大学可以将本学位论文的全部或部分内容编入有关数据库，允许采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：

徐玉祥

指导教师签名：

姜元春

签名日期：2017年4月18日

签名日期：2017年4月18日

论文作者毕业去向

工作单位：

联系电话：

通讯地址：

E-mail：

邮政编码：

# 致 谢

研究生阶段的学习，在快速而充实的节奏中不知不觉过去，令人唏嘘不已。春天来了，毕业的钟声也随之敲响，到了该说再见的时候了。此时此刻，总会勾起对往日回忆，回想起这三年学技术、做科研、定课题、写论文的日子，有过困难，有过迷茫，更有收获的喜悦。自己不仅在科研的道路上收获了成果，更重要的是学到了解决问题的思路以及为人处世的哲学，这些都让我受益良多。

首先，我要感谢敬爱的姜元春老师，给予我诸多指导和关怀！三年前刚成为老师学生的我对学术研究一知半解，多亏老师孜孜不倦地教导。姜老师言传身教，除教学工作外其他时间都在实验室忙科研，每当我在研究的过程中遇到困难时，老师总能及时地给予我帮助。老师严谨的治学态度、勤奋钻研的科研精神促使着我不断进取，经过研究生期间扎实的学习，自己的能力得到极大提升。后期大论文撰写过程中，老师从论文选题、撰写开题报告、构建论文框架到撰写和修改论文都对我悉心指导，在此向敬爱的姜老师表示衷心的感谢。

其次，我要感谢身边的老师同学们，每次遇到困难，都能在他们的帮助下走出泥潭。在这里，我想感谢刘业政老师、孙见山老师、凌海峰老师，感谢老师们在学术科研上的教导；感谢邵亮师兄在我技术上的指导，为论文和工作打好基础；感谢王锦坤博士、王佳佳博士和杜飞博士，谢谢你们对我论文写作的建议和指导，同时还要感谢李隆、何怡、陈雪娇、章旭、李荣岗、梁世全、谷俊辉等同学，谢谢你们热情的帮助和支持。

最后，我要感谢一直给予我无微不至的关爱的家人——爷爷、爸爸、妈妈和姐姐。爸爸、妈妈在我失望、低落的时候总是给予我鼓励，让我重拾信心，迈过了人生中一道又一道坎；爷爷以他人生的阅历经常教导我为人处世的道理；姐姐也总是时时刻刻关心着我。在此，我要谢谢你们一直以来的支持与陪伴！

研究生阶段的学习和生活将在不舍中画上句号，但我明白，这只是我人生中的一个逗号。我会在未来在工作中努力上进，把研究生期间学到的科研精神用到工作中去。

作者：徐玉祥

2017年3月

# 摘 要

作为一种通过关注机制分享简短实时信息的广播式社交网络平台，微博已经成为人们交流和获取信息的重要渠道。用户关注列表中好友发布的微博是用户获得的信息主要来源，但随着微博用户规模和活跃用户数量的不断增长，用户的关注列表变得越来越稠密，从而导致用户可能面临信息过载的问题。如何从繁多的信息流中挖掘出对用户有价值的微博是提高微博用户服务质量的关键问题。

本文研究了现有的微博推荐方法，在充分利用现有的方法基础上，结合实际应用场景，将微博个性化推荐问题转化为对用户接受到的微博信息重排序的问题。首先本文通过会话划分，确定了推荐范围；接着基于用户偏好、微博内容、发布者权威三种维度构建了多个微博特征；然后提出了动态自适应 (dynamic self-adaptive feature weighting, DAFW) 的权重融合方法，在实验验证其推荐有效性的基础上，最后设计了实时个性化微博推荐系统。本文的工作成果主要有以下几个方面：

(1) 本文建立了一套较为完备的微博语料处理方案，基于处理过的微博语料训练通用主题模型，再采用吉布斯采样 (Gibbs Sampling) 从通用主题模型中抽取出目标文本的主题分布，从而解决了短文本主题建模难的问题。

(2) 本文抽取用户转发过的但非其关注的博主，将其发布的微博作为待推荐内容不足情况下的补充，使得用户不用通过关注机制也可以接收到感兴趣的微博。

(3) 本文基于用户偏好、微博内容、发布者权威三种维度构建了十个微博特征，其中包括了很多分析特征，如微博热度、交互 TF-IDF 等，这些特征的加入，能够明显的提高推荐质量。

(4) 为解决多指标融合问题，本文将信息熵引入到权重调整环节，主要根据每一个特征值变异性大小来确定客观权重，并结合均值及参数得到排序函数，实验证明了本文方法的有效性。

(5) 研究并设计了实时推荐系统，将本文的理论研究成果应用到实际生活中，从而创造价值。

本文工作能够有效的处理社交网络时代普遍存在的信息过载问题。用户能够花更少的时间捕捉到朋友圈中比较有特色的信息，并能够实时的获得推荐结果，对于提高用户体验具有较好的理论和实际意义。

**关键词：**个性化推荐；LDA；动态自适应权重；分析特征；微博



# ABSTRACT

As a kind of broadcast social network platform which shares short real-time information through following mechanism, micro-blog has become an important channel for people to exchange and get information. The micro-blog which be published by friends of user's attention-list is the main source for user to acquire information, however, with the growing number of micro-blog users and active users, user's attention list has become longer, user maybe face the problem of information overload by these factors. How to dig out the value of micro-blog from a wide range of information flow is the key to improve the quality of service for micro-blog users.

This paper studied the existing method of micro-blog recommendation and made full use of them, then combed with the practical application scenarios, this paper transformed the problem of personalized micro-blog recommendation into the reordering of the micro-blog received by user. First of all, the recommended range is determined by session segmentation, then this paper constructed a lot of micro-blog features based on user preferences, micro-blog content and publisher authority, and then put forward a multi-feature fusion method based on dynamic self-adaptive feature weighting (dynamic self-adaptive feature weighting, DAFW), finally this paper designed a real-time personalized recommendation system of micro-blog after validating the validity of this scheme. The main results of this paper are as follows:

(1) This paper established a series of scheme which processing micro-blog corpus, then trained the general theme model based on the processed micro-blog, and then used method of Gibbs Sampling to estimate the theme distribution of the target text from general theme model, this method can solve the difficult of topic modeling for short text.

(2) This paper excavated the publisher who forwarded by but not followed by user, the micro-blog of publisher will be as a supplement to the recommended content in case of insufficient, so users can also receive interest micro-blog through other resources.

(3) This paper constructed ten features of micro-blog based on user preferences, micro-blog content and publisher authority, including many analysis features, such as micro-blog heat, interactive TF-IDF, etc., adding these features could significantly improve the quality of recommendation.

(4) In order to solve the multi-index fusion problem, this paper introduced the information entropy into multi-factor weighting, to calculate the objective weights

according to the variance of each features value, and combined features mean and parameters to get the sorting function, experimental results show the effectiveness of this method.

(5) This paper studied and designed a real-time recommendation system, to apply the theoretical research of this paper to real life which provides value.

The work of this paper can effectively deal with the information overload problem in the social network era, users can spend less time to capture more characteristic information in the circle of friends and get the recommended results in real time, and our work has better theoretical and practical significance for improving the user experience.

**KEYWORDS:** personalized recommendation; LDA; dynamic self-adaptive feature weighting; analysis feature; micro-blog

# 目 录

第一章 绪论 .....	1
1.1 研究背景和选题意义 .....	1
1.2 国内外研究现状 .....	2
1.2.1 社交网络的特征分析 .....	3
1.2.2 基于传统推荐算法的微博推荐 .....	4
1.2.3 基于机器学习的微博推荐 .....	4
1.3 本文研究工作 .....	5
1.4 论文的组织结构 .....	6
第二章 微博推荐的相关理论及技术 .....	8
2.1 微博结构 .....	8
2.2 微博网络爬虫及文本预处理 .....	9
2.2.1 微博网络爬虫 .....	9
2.2.2 文本预处理 .....	10
2.3 文本挖掘相关理论 .....	11
2.3.1 TF-IDF 算法及其原理 .....	12
2.3.2 LDA 算法及其原理 .....	12
2.3.3 熵理论 .....	14
2.4 推荐效果的评价指标 .....	15
2.5 本章小结 .....	16
第三章 基于动态自适应权重的个性化微博推荐 .....	17
3.1 用户会话划分 .....	17
3.2 主题模型构建 .....	19
3.3 微博特征构建 .....	22
3.3.1 基于用户偏好的特征构建 .....	22
3.3.2 基于微博内容的特征构建 .....	23
3.3.3 基于发布者权威的特征构建 .....	25
3.4 基于动态自适应的特征权重构建 .....	26
3.5 本章小结 .....	30
第四章 实验验证和原型系统 .....	31
4.1 数据收集和预处理 .....	31
4.2 实验及结果分析 .....	33
4.2.1 评价方法 .....	33
4.2.2 基准模型 .....	34

4.2.3 实验结果 ..... 34

4.3 原型系统 ..... 38

4.3.1 数据采集层 ..... 39

4.3.2 数据处理层 ..... 42

4.4 本章小结 ..... 46

第五章 总结与展望 ..... 47

5.1 工作总结 ..... 47

5.2 展望 ..... 48

参考文献 ..... 49

攻读硕士学位期间发表的论文 ..... 53

特别声明 ..... 54

# 插图清单

图 2.1	微博中用户关系 .....	8
图 2.2	微博信息结构图 .....	9
图 2.3	微博网页爬虫的流程示意图 .....	10
图 2.4	LDA 的图模型 .....	13
图 3.1	LDA 模型的矩阵示意图 .....	20
图 3.2	用户和微博主题分布示意图 .....	24
图 4.1	文本预处理流程图 .....	32
图 4.2	实验流程图 .....	32
图 4.3	动态自适应权重模型与基准模型的 P@N 对比 .....	35
图 4.4	动态自适应权重模型与基准模型的 MAP 对比 .....	36
图 4.5	动态自适应权重模型与基准模型的 ACC 对比 .....	36
图 4.6	动态自适应权重模型与基准模型的 D@N 对比 .....	37
图 4.7	实时推荐系统结构 .....	39
图 4.8	数据抓取的架构图 .....	40
图 4.9	爬虫程序主页 .....	40
图 4.10	系统提交抓取任务页面 .....	41
图 4.11	抓取过程中控制台信息 .....	41
图 4.12	数据表设计 .....	42
图 4.13	实时数据处理的流程图 .....	43
图 4.14	抓取下来的微博数据 .....	44
图 4.15	数据清洗、分词后的效果 .....	44
图 4.16	LDA 训练的文档集 .....	45
图 4.17	主题-单词排序文档 .....	45
图 4.18	系统推荐效果 .....	46

# 表格清单

表 2.1	正则表达式的元字符 .....	11
表 2.2	微博中使用的正则表达式 .....	11
表 2.3	基于 LDA 主题建模的参数解释 .....	14
表 3.1	用户会话划分实例 .....	19
表 3.2	用户微博中词的主题分布 .....	20
表 3.3	用户在各个主题下的概率与词频 .....	20
表 4.1	模型参数 .....	34
表 4.2	基于 DAFW 方法推荐的实际效果 .....	38
表 4.3	主题模型训练参数 .....	44

## 第一章 绪论

### 1.1 研究背景和选题意义

随着互联网技术和移动通信基础设施的更新换代,近几年来,QQ、微信、新浪微博等社交工具的覆盖率与日俱增,这些工具改变了用户获取信息和交流的方式。越来越多的人会在社交网络中发表自己的观点、记录生活的点滴、分享有趣的内容,社交媒体中所拥有的活跃用户日益增多。

微博的活跃用户也持续增长,截止 2016 年 12 月,统计显示微博月活跃用户数量已经达到了 3.13 亿,平均每天发布的微博总数超过了一亿<sup>[1]</sup>。微博已经逐渐成为一种社会化自媒体:社会名人通过微博发表最新动态及对一些热点事件的看法,实现与粉丝的直接互动;企业借助微博发布一些广告和新产品,提升企业曝光率;新闻媒体通过微博平台发布一些热点新闻,促成热点新闻传播;政务机关通过官方微博公开一些公务信息,提高政府透明度、发挥群众监督作用;作为普通用户可以使用微博近距离接触名人明星的生活百态,追踪一些社会热点以及朋友的最近动态等。每个用户在使用微博的过程中为了能够获取想要的信息,必须关注于特定的用户,这就导致了用户的关注列表不断增加。据统计,目前微博的平均关注数已经达到了 400 多人。随着用户的关注列表及活跃用户数的增加,相继产生了信息过载现象。而且人们获取信息的方式越来越多样化,除了微博,用户还可通过 QQ、微信朋友圈、多种新闻客户端、浏览器等获取丰富的信息,即对用户来说,由于精力有限,停留在微博上的时间大大压缩,微博用户更希望用零散的时间就可以获得更多的信息。而实际情况却不容乐观,浏览的微博中可能充斥着大量广告,一些异常活跃的关注用户,可能在短期内发了过多微博。目前微博信息主要按照发布的时间顺序推送给用户,那些发布时间比较靠前的微博就很可能被用户忽略掉,而这些微博中很可能有些是用户更加关心的。

除了上述的信息过载问题,不排除有些异常活跃的用户频繁的登陆微博,习惯性地刷朋友圈,但由于距离上次下线时间过短,在此期间用户关注的好友所发布的微博数量有限,朋友圈里发布的微博已经不能满足用户兴趣需求。为了解决上述问题,针对微博的推荐系统应运而生。

目前微博个性化推荐服务是根据用户以往的阅读记录来推测用户偏好,从而为用户推荐可能感兴趣的微博。例如根据用户以往转发的微博,预测用户可能对哪一方面感兴趣,接着分析发布微博的作者的信息分布,将兴趣相似的发布者微博推送给用户,这是基于协同过滤的思想。该方法能够有效地为用户推荐相关的内容,但在实际使用中却容易忽视用户体验,举例来说,如果用户经常转发一些体育方面的微博,那么基于该方法的推荐每次都会将有相同爱好的用户发微博推

荐给用户。在使用一段时间后，由于每天看到的内容很可能都是由同样的人发布的微博，从而导致用户浏览微博的热情会大减。产生上述现象主要有两方面的原因：一方面是描述微博的特征过少，不能够全面反映用户对该微博的兴趣；另一方面基于协同过滤思想会导致多样性的缺失。而从推荐主体来看，当待推荐的微博数量过少时，不管如何排序，推荐的整体质量很难提高，而目前的推荐系统很少考虑到推荐主体不足情况下的解决方案。

综上所述，本文微博个性化推荐系统需要解决以下几个问题：（1）为每条微博构建更加全面的度量指标；（2）有效地融合多种指标，在保证推荐精度的同时能够提高推荐的多样性；（3）在推荐主体不足的情况下，为用户推荐潜在感兴趣的微博。因此，本文针对上述问题，提出了基于动态自适应的权重的个性化微博推荐系统。首先，利用用户历史转发记录，将用户好友发布的微博进行会话划分，基于此确定了推荐主体，即每次会话期间产生的微博。然后，对用户转发过的博主进行提取，即将其中非用户关注的博主作为微博推荐的候选集，在推荐主体不足情况下采集近期这些候选集微博作为补充，确保最终的推荐质量。接着，对每一条微博，从用户偏好、微博内容、发布者权威三个角度量化 10 个特征，全面地反映每条微博的价值。为了有效融合多方面的特征，本文提出了一种动态自适应的特征权重计算方法。该方法能够实现在待推荐微博动态变化情况下对特征项权重进行自适应计算和动态调整，从而将在一些特征上有明显优势的微博推荐给用户。最后，本文基于实验验证方法的有效性，设计了实时个性化推荐系统，以期研究结果能够在实际生活中创造价值。

基于本文的工作，能够有效的处理社交网络时代普遍存在的信息过载问题，用户能够花更少的时间捕捉到朋友圈中比较有特色的信息，并能够实时的获得推荐结果，这对于提高用户体验具有较好的理论和实际意义。

## 1.2 国内外研究现状

目前推荐系统主要应用于信息检索<sup>[2]</sup>（如 MyYahoo、iGoogle、GroupLens、百度等）、在线电子商务<sup>[3]</sup>（如 Netflix、Amazon、eBay、阿里巴巴、豆瓣等）、生活服务<sup>[4]</sup>（如旅游服务 Compass、博客推送 M-CRS 等）、移动应用<sup>[5]</sup>（Daily Learner, Appjoy 等）等各个领域。其核心是推荐算法，主要分两大类：一类是传统的推荐算法如协同过滤推荐算法<sup>[6,7]</sup>、基于内容的推荐算法<sup>[8-10]</sup>、混合推荐算<sup>[11-13]</sup>；一类是将机器学习方法融合进推荐算法中，如排序学习技术<sup>[14]</sup>。因此传统推荐算法研究具有非常重要的指导作用，故本文主要从社交网络的特征、传统推荐方法的微博推荐、基于机器学习的微博推荐三个方面进行文献综述和现状分析。



### 1.2.1 社交网络的特征分析

当前研究中,有一部分是从用户使用社交网络的目的角度进行分析。例如 Akshay<sup>[15]</sup>指出人们使用 Twitter 的目的主要有三种,即分享信息、日常交流、浏览并评论一些新闻热点。该文首先利用 HITS 算法把用户分为不同的社区,然后在每个社区中又将用户分为三类:其一是朋友(Friends),是三类中占比最大的类;第二类是信息来源(Information source),文章认为有更多粉丝的用户发布微博相应的价值更高;剩下的是信息获取者(Information Seeker),这类用户特点就是关注了很多用户,却很少发布信息,针对每一类用户,其推荐的方法不尽相同。Naaman 等人<sup>[16]</sup>通过 Twitter 用户的历史数据统计分析,总结了用户使用 Twitter 主要基于社交需求及信息检索目的。

Jilin 等<sup>[17]</sup>指出在 Facebook、Twitter 等社交网络中,当向用户个性化推荐会话时主要面临两方面困难,即信息过载和用户兴趣的不同,解决问题的关键是分析造成用户不同偏好的因素。Jing 等人<sup>[18]</sup>比较了用户转发 Twitter 的因素,发现用户在转发 URL 时,其实主要出于其为好友的转发 URL,而不是出于这条 URL 的发布者,说明好友在转发中的重要性。Ibrahim 等<sup>[19]</sup>,提出了一种有效的信息过滤机制来处理微博信息流中信息过载问题,主要根据用户的转发记录得到用户对好友的微博偏好大小,根据这个偏好将其关注的好友分级,用户浏览到高级别的用户发布的消息后,用户浏览的兴趣会增加,而那些可能形成干扰的微博就相应靠后。Paek 等人<sup>[20]</sup>将支持向量机引入到 Facebook 中,以此计算接收的消息与用户的重要性大小,信息主要来源于好友最新状态及新闻热点。结果表明大量的信息中只有一部分对用户比较重要,而通过个性化方式为用户推送信息能够有效的区分信息重要性。

为了有效地区分好友推送的微博,研究人员从多角度构建了微博的特征。Zhao 等<sup>[21]</sup>从用户权威度角度度量信息,根据用户的粉丝数量与好友数量的比率来判断用户的权威度,值越大代表用户的影响力越大。Cha 等<sup>[22]</sup>则从入度、转发数和提及数度量微博,入度即用户的粉丝数量,转发数即微博被转发的数量,提及数指的是提到其他人的数量。Morgan 等<sup>[25]</sup>在构建用户兴趣模型时引入主题模型。Taiki<sup>[23]</sup>和卫冰洁等<sup>[24]</sup>认为微博具有时效性,故在微博排序时将时间作为一个重要因素。Victoria 等<sup>[26]</sup>指出用户在选择微博时主要出于四个要素:用户的价值观、用户的身份、用户的个性、微博质量。Rinkesh 等<sup>[29]</sup>将微博背后的社交元素考虑到微博排序中,如作者的出入度。Wouter 等<sup>[28]</sup>从博文可信度、微博内容两个维度线性融合 10 种特征,基于此为博客排序。Nasir 等<sup>[27]</sup>在评估微博时,考虑的因素有是否有 URL、标签内容、情感词分析、表情统计、微博主题等。Jan 等<sup>[30]</sup>在对微博质量进行比较时,利用 SVM 方法训练构造器,训练时主要基于微博良构性、真实性、导航质量。

本文在综合上述特征基础上,提出了一些需要建模和统计分析的分析特征,这些隐含在微博中的特征在度量微博与用户相关度方面起着重要作用。

### 1.2.2 基于传统推荐算法的微博推荐

在传统的微博推荐算法相关研究中,大量工作是基于用户兴趣相似的协同过滤思想。如 Wu 等<sup>[31]</sup>把用户的历史微博归集起来,通过 TF-IDF 的抽取关键词,这些关键词近似代表了用户的兴趣。Jilin 等<sup>[32]</sup>也基于 TF-IDF 抽取用户的兴趣标签,然后抽取待推荐微博关键词作为其兴趣标签,最后评估待推荐微博与用户兴趣标签的相似度,将相似的微博推荐给用户。闫强<sup>[33]</sup>等提出基于社交网络的动力学模型,该模型指出用户的兴趣在很大程度上能够影响用户的行为,在构建用户兴趣模型时,除了考虑一些显示信息,如用户资料、标签介绍等,还需要考虑一些隐式信息,如评论、转发过的历史微博等,最后基于此将兴趣分布相似的微博推荐给用户。

上述基于 TF-IDF 方法原理主要是其可以度量词语相对与文章重要程度,从而可以将重要的词作为语义的概括,但对于微博这种短文本来说,准确度却不高,无法区分一些歧义词。近年来基于主题模型的用户兴趣建模在微博推荐中大量运用,其中典型的代表如 PLSA 和 LDA。Weng 等<sup>[34]</sup>将用户历史微博整合成文档,然后通过 LDA 分析用户主题偏好。Ramage<sup>[36]</sup>等则将 Labeled-LDA 引入到对 Twitter 的用户和内容建模中去。Hong 等<sup>[35]</sup>比较三种方法下用 LDA 主题模型分析主题的准确度,其一是将每条微博作为一个单独的文档,第二种是把用户的以前微博合并作为一个文档,第三种是将包含相同标签的微博合并成一个文档,实验表明把用户的以前微博合并作为一个文档的主题模型最有效。高明等<sup>[37]</sup>也利用 LDA 主题模型预测微博用户的兴趣偏好,然后对实时热点微博做了提取,将其中与用户偏好比较吻合的微博推荐给用户。

一些研究者从矩阵分解角度来做微博推荐,Cui<sup>[38]</sup>等提出混合非矩阵分解的方法,该方法对随机游走模型进行了改进,构造了联合关注-兴趣模型,基于该模型能够实现文本推荐以及链接预测。Hong<sup>[39]</sup>等提出了通用矩阵分解模型,能够融合多种特征,从而全面地构建了用户的兴趣并根据兴趣来预测用户的行为。

### 1.2.3 基于机器学习的微博推荐

近几年,随着微博的研究的深入,越来越多表征微博的特征被提出来,而采用传统的推荐算法很难融合多方面因素,于是相关研究将机器学习的方法引入推荐算法中。机器学习区别与传统的推荐方法是其需要经过训练阶段,是一种监督性的学习,即通过训练数据集得到排序模型,并且通过调整模型的参数得到最优解,最后基于训练好的排序模型产生获得测试集的推荐结果。基于机器学习的排

序学习(Learn To Rank)方法能够分析诸多因素对微博用户的行为影响力度,如微博发布时间、用户兴趣偏好、用户间亲密度、用户权威度等因素。排序学习根据训练集形式的不同,一般可分为三类<sup>[40]</sup>: pointwise 方法、pairwise 方法和 listwise 方法。目前形成了一些比较经典的算法如 RankNet<sup>[41]</sup>,该算法原理是取训练集和测试集之间的相对熵及作为损失函数,并采用梯度下降算法来迭代训练神经网络,从而获得排序模型; Ranking SVM<sup>[42,43]</sup>,该方法是基于支持向量机的排序学习方法。RankBoost<sup>[44,45]</sup>,该方法通过提升策略(boosting)进行排序学习构建。

这些排序学习的方法已经广泛应用于搜索、推荐中,如 Zheng 等<sup>[46]</sup>收集了用户的点击数据,然后基于此构建了基于偏序关系的训练集,从而训练出排序函数,再根据排序函数对查询结果重排序、优化。通过该方法能够很好融合多特征,比仅仅依赖内容相似性的方法效果更好。彭泽环等<sup>[47]</sup>基于排序学习很好的融合了四大类影响用户推荐效果的模块,即用户个人信息模块、用户历史微博内容模块、用户的社交拓扑结构模块和交互信息模块,从而为用户推荐高质量的博主。Duan<sup>[48]</sup>提出一种基于排序学习方法的微博搜索模型,其中考虑的特征包括查询词与微博内容的相关性、微博文本特征、微博发布者的权威度,实验表明该方法比传统基于内容相关性的方法在排序准确率上提升了很多。

综上所述,相比于传统推荐方法,排序学习方法有诸多优势。目前对排序函数的学习主要采用半监督的方式,即用来训练参数的数据集需要人工标注相关性,在人工标注一条微博与用户是否相关或者相关程度时,结果会因人而异,从而可能导致真实值和主观值之间存在误差。因此实验中较难获取能够反映真实情况的训练数据。无论是基于传统推荐算法还是排序学习方法,都有其相应的应用场景,而本文的应用场景是将用户最近关注的好友产生的微博重排序推荐给用户。好友产生的微博集合是动态变化的,如果基于现有的排序学习方法得到排序函数可能出现如下情况,例如在排序学习中发现用户偏向于转发和自己主题相关的微博,因此系统赋予主题维度的特征更高权重。假如在某次会话中主题相关的微博特别多,但其中有一条微博的作者与用户主题不相关,但用户经常转发、关注其微博,历史行为表明用户更希望看到这条微博。但当表征这一行为的特征权重在训练过程中获得很低的权重时,这条微博在那么在总分就不具有优势。为此,本文在采用了上述工作者提出的部分微博特征度量指标,此外提出用户与发布者之间互动 TF-IDF 指标,从用户历史行为上度量其对发布者微博的偏好,提出一种动态自适应的特征权重计算方法。该方法能够根据待推荐微博在每个特征上分布的特点动态调整权重,使用户能够在更短的时间内浏览到有价值的微博。

### 1.3 本文研究工作

微博由于是一种短文本结构,语料中充斥着大量的噪音,为了能够实现面向

用户的实时个性化推荐微博，本文基于以下几个方面展开研究工作。

(1) 微博语料库的构建：由于本文需要训练出通用主题模型，需要大量的微博语料作为训练集，本文研究搭建了爬虫框架，获得未处理的微博语料，然后研究并编程实现了针对微博语料的中文语料处理，通过高频词统计把控处理的质量。

(2) 短文本主题建模：为了能够对微博这种短文本实现主题建模，本文基于语料库，研究了不同参数及迭代次数下基于 LDA 训练出的主题模型效果，通过比较选择出最优参数及合理的迭代次数，采用 Gibbs Sampling 方法从通用主题模型中抽取出短文本的主题分布，为微博的特征构建提供了基础。

(3) 微博特征提取：从用户偏好、微博内容、发布者状态三个角度的微博提取，构建了十个微博特征。研究并提出了一些分析特征，其能够有效表征微博的价值，例如改进的度量微博与用户的主题分布相似性的特征、能够度量用户与用户之间亲密度的特征。

(4) 微博特征融合：研究并提出动态自适应的特征权重模型。为解决多指标融合问题，本文将信息熵理论引入到权重调整环节，根据每一个微博特征指标变异性大小来确定客观权重，并结合均值及参数得到每个特征的动态权重，能够根据推荐内容的不同，自动调整权重参数，实验表明该方法能有效融合多方面特征。

(5) 实时推荐系统设计：为了将研究结果应用到实际中，本文研究并设计了实时推荐系统，通过数据采集层采集数据、数据处理层进行特征提取和重排序工作、数据展现层实现数据展示和交互工作，结合设计开发了大量功能接口，为后续应用做好基础。

## 1.4 论文的组织结构

本文的组织结构及内容如下：

第一章：绪论。首先阐述了本文的研究背景及意义，明确了当前的问题及研究目标；然后通过阅读文献分析目前针对微博个性化推荐的国内外研究现状；接着结合当前研究现状和应用背景提出本文的研究思路；最后介绍了本文主要的研究内容和论文的创新点。

第二章：微博推荐的相关理论及技术。首先分析了微博所包含的结构信息，然后针对其结构研究了爬虫和文本预处理技术，接着对本文用到的文本挖掘领域的方法及思想进行了介绍，即 TF-IDF、LDA、熵原理，最后对推荐系统常用的评价指标进行了归纳总结，为随后的研究奠定基础。

第三章：基于动态自适应权重的个性化微博推荐。首先通过用户会话划分近似得到用户的登陆时间，基于此将用户历史微博分段，得到每次登陆时好友推送的微博，即确定了微博的推荐范围；然后基于通用主题模型的训练，从三个维度

对体现用户个性化需求的特征进行量化；最后基于信息熵原理构建动态自适应的特征权重模型，得到排序函数。

第四章：实验验证和系统设计。首先介绍了微博数据采集及预处理流程；随后我们引入了四种全面度量模型有效性指标，即  $P@N$ 、MAP、HIR、 $D@N$ ；接着将本文提出的模型与默认的时间序列及基于 RankNet 排序学习方法、未引入分析特征的推荐模型进行效果对比，并以实际效果验证了模型在微博个性化推荐中的有效性；最后围绕数据采集层、数据处理层、数据展现层对实时个性化推荐做了一个系统设计和部分实现工作，以期研究结果能够在实际生活中创造价值。

第五章：总结与展望。该章首先对本文所获得的工作成果进行了归总，然后详细阐述了本文研究存在的局限性，最后基于此对下一步工作重点及研究方向做了介绍。

## 第二章 微博推荐的相关理论及技术

本文采集了大量用户历史微博及个人信息，通过文本预处理和主题模型的训练，获得研究所需要的数据，然后从用户兴趣、微博内容、发布者权威三个维度构建特征，最后基于信息熵原理提出动态适应的权重模型，经过实验验证了方法的有效性。本章主要将介绍上述过程中涉及的相关理论与技术，主要介绍了微博的结构、数据抓取和处理的技术方法、文本挖掘原理以及推荐系统评估的相关理论知识。

### 2.1 微博结构

微博作为新兴的社交网络，与传统的社交网络相比在结构上存在一些差异，例如接收消息是建立在关注关系基础上，而且关注关系是单向的。以新浪微博为例，如果用户 A 关注了用户 B，如图 2.1 (a) 所示，那么用户 B 就是用户 A 的关注好友，与此同时用户 A 成为了用户 B 的粉丝，这个过程中不需要 B 的审核，；当然用户 B 也不需要审核就能够关注用户 A，如图 2.1 (b)；当用户 A 和用户 B 彼此关注时，如图 2.1 (c)，就称为互粉。

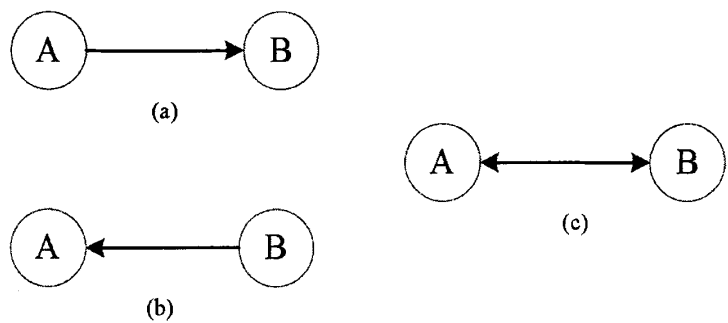


图 2.1 微博中用户关系

Fig 2.1 User relationship of micro-blog

通过上述方式，用户关注的所有好友就形成了其关注列表，关注的好友发布的实时微博会依照发布时间先后推送给用户。用户接受到的微博主体部分被限定在 140 字以内，是典型的短文本，用户可以通过图片、超链接、小视频等丰富内容。故在微博质量的相关研究中，除了将微博长度作为高质量微博的判断标准外，我们还可以通过判断是否包含图片、超链接、小视频等来确定微博质量，这类微博相比于纯文本微博内容更加丰富，更能吸引用户的浏览和转发。

用户发布的每条微博除了本身内容外，还携带了一些社交信息，主要是用户的个人信息以及微博的互动信息。如图 2.2 所示，其中用户个人信息又分为两类即

用户资料和指标类信息，用户资料包括职业、标签、昵称、性别等，指标类信息包括微博数、关注数、粉丝数、是否为认证用户等；微博的互动信息包括了微博的点赞数、转发数、评论即评论数等，反映了微博在用户之间的互动情况。

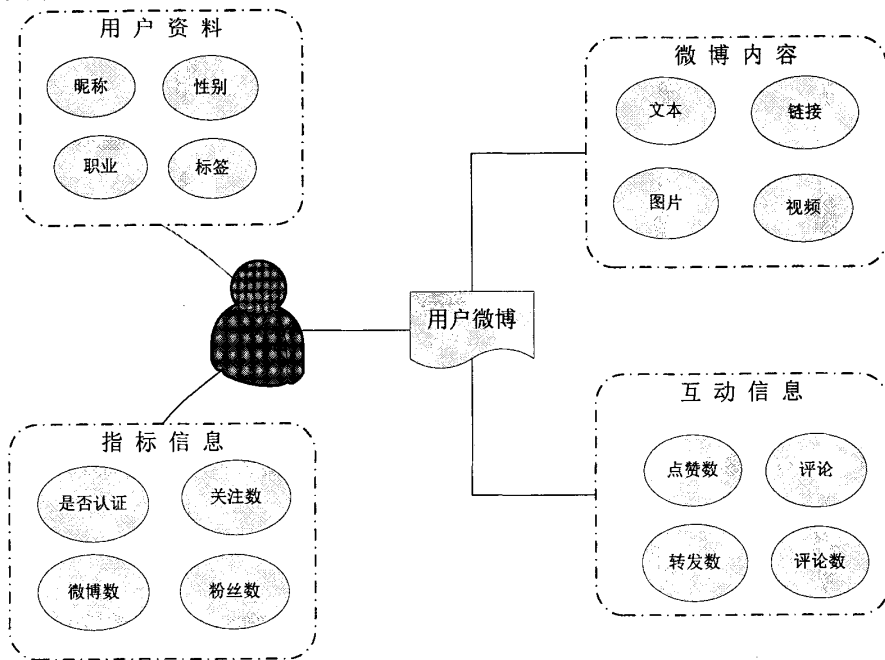


图 2.2 微博信息结构图

Fig 2.2 Information structure of Micro-blog

通过上述微博信息的整理，本文明确了影响微博总体质量的各方面因素，并将其归为四类，即微博内容、互动信息、微博发布者的用户资料及指标信息，为下一步数据采集和特征构建奠定了基础。

## 2.2 微博网络爬虫及文本预处理

虽然微博包含的信息种类很丰富，但通过新浪 API 接口获取数据的方法只能得到部分数据，无法获取到一些涉及用户个人信息但可能对推荐结果有重要影响的数据。因此本文采用垂直型爬虫技术获取用户数据，然后基于正则表达式及分词工具完成文本的预处理。

### 2.2.1 微博网络爬虫

网络爬虫是一种按照一定的要求，自动地抓取互联网信息的程序或者脚本，被广泛运用于一些搜索引擎中或其他应用中。本文获取数据的微博爬虫程序主要由模拟登陆、页面抓取、页面解析、任务调度四个关键模块组成，图 2.2 为本文的微博网络爬虫流程示意图。

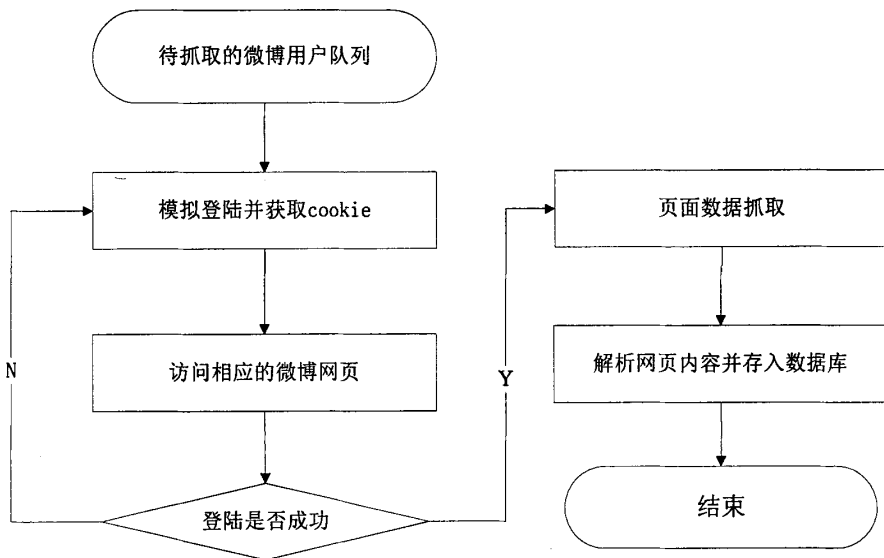


图 2.3 微博网页爬虫的流程示意图

Fig 2.3 The process diagram of micro-blog web crawler

其中页面抓取是通过 httpclient 发送请求，从而获得网页数据，数据解析采用 JSoup 抽取本文需要的内容，抽取的数据存入 MySQL 数据库中。为了能够连续高效率的爬取微博数据，文本采用 7 台电脑同时协作，轮流使用其中 6 台电脑抓取微博，有效减少了爬虫账号被封情况的发生。

### 2.2.2 文本预处理

文本预处理是分析数据的基础，通常经过数据清洗、分词、去除停用词等几个步骤。本文分词是基于 Hanlp 提供的分词包完成的，去停用词是基于常见停用词及高频噪音词过滤的，以下主要对微博数据清洗用到的方法进行介绍。

微博中除了文字外，还有很多噪音，如颜文字、表情符、URL 链接、标签等信息，需要在保证文本内容不被清理的同时清洗掉这些噪音。为了实现上述目标，本文主要使用正则表达式来处理这些噪音，主要基于其有极强的字符串表示和匹配能力，可以有效地对杂乱无章的微博数据进行降噪。

正则表达式原理是通过基本的元字符搭配组合，从而匹配一些复杂的字符串。表 2.1 列举了主要的元字符及其作用，表 2.2 列举了本文用来统计或者降噪微博的正则表达式。



表 2.1 正则表达式的元字符

Tab 2.1 The metacharacter of regular expression

元字符	作用
.	匹配一个换行之外的任何字符
+	匹配 1 次或多次
?	匹配 0 次或 1 次
\d	匹配一位数字
\s	匹配一个空白字符
\w	匹配一个字母, 数字或下划线
\D	匹配任意一位不是数据的字符
\b	匹配单词的边界
*	匹配 0 次或多次
{p,q}	匹配 p 次到 q 次, 包括 p,q
[^y]	匹配任意一位非 y 的字符

表 2.2 微博中使用的正则表达式

Tab 2.2 Regular expressions used in micro-blog

正则表达式	作用
<img\\src=(.??)[^>]*?>	匹配表情符
@.*?:[: ]	匹配转发者
http://t.cn\\w{6,7}	匹配超链接
<img.*render=(.??)[^>]*?>	匹配插图
#.[^#]{2,25}#	匹配标签
[\\u4e00-\\u9fa50-9]+	匹配汉字、字母及数字
^[0-9]+	匹配一串数字
\\s+{1,}	匹配一串空格

基于正则表达式, 首先去除表情符、超链接、标签、转发者等, 然后提取汉字、字母及数字, 接着实现繁体转简体, 最后分词、去停用词、词频统计, 完成本文预处理工作。

### 2.3 文本挖掘相关理论

微博作为一种简短而包含丰富的文本, 需要借助于一些文本挖掘的理论方法来分析文本中所包含的信息。下面主要介绍了一些本文所用到的模型与方法。

### 2.3.1 TF-IDF 算法及其原理

TF-IDF (term frequency - inverse document frequency) 是一种常用于资讯检索和资讯探勘的加权技术。其采用统计的方法, 来评估字词对于文件集或文档的重要程度。其核心思想是字词的重要性一方面随着它在文件中出现的次数累积而成正比增加, 但同时会随着它在语料库中出现的次数增长而成反比下降。

在一份给定的文件里, 词频 (term frequency, TF) 指的是每个对应的词语在该文件中出现的频率。目的是对词数 (term count) 的归一化, 防止出现它偏向长的文件。那么在某一特定文件  $i$  里的词语  $j$  来说, 它的重要性可表示为:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (2.1)$$

逆向文件频率 (inverse document frequency, IDF) 是从词语普遍重要性的角度度量。计算某一特定词语的 IDF, 首先由总文件数除以包含该词语的文件数, 再将得到的商取对数得到:

$$idf_i = \log \frac{|D|}{1 + |\{j : t_i \in d_j\}|} \quad (2.2)$$

其中  $|D|$  指的是语料库中的总文件数,  $|\{j : t_i \in d_j\}|$  指的是包含指定词语的文件数目, 为避免分母为零, 将其加一。最终得到了指定词的重要性度量指标为:

$$tfidf_{i,j} = tf_{i,j} \times idf_i \quad (2.3)$$

在微博中, 可以将用户所发布的微博看作一个文档, 采用 TF-IDF 算法计算每个词语在微博文档中的重要程度, 将权重高的词语作为用户兴趣标签, 代表用户的兴趣分布。

### 2.3.2 LDA 算法及其原理

用词语权重量化文档的方法虽然简单, 但却没有涉及文档的语义层面, 无法结合上下文来区分歧义词。为了从语义层面挖掘隐含的信息, Blei 等首次提出了一种概率生成模型 LDA (Latent Dirichlet Allocation) [49], 可用于通过无监督学习来学习估计文本主题分布。LDA 是利用文本建模中语义分析 (LSA) 的思路即在文本语料库中找到“主题”或“语义”的潜在结构, 本质是利用文本中词项的共现来发现主题结构。LDA 定义了如下所示的文档生成过程:

- (1) 对于每篇文档, 首先需要在主题分布中抽取一个主题;
- (2) 然后从上述抽到的主题所对应的单词分布中抽取一个单词;
- (3) 重复上述两个过程, 直到生成一篇文档所需要的所有单词。

具体来说,生成过程涉及两个分布,其一是语料库中的文档 $m$ 与 $K$ 个主题(经过反复试验得来,在计算主题模型前需要事先指定)存在一个多项分布(multinomial distribution),将这个多项分布记为 $\vec{\theta}_m$ 。其二是主题 $k$ 又与语料集中 $V$ 个单词存在一个多项分布,把这个多项分布记为 $\vec{\phi}_k$ 。 $\vec{\theta}_m$ 和 $\vec{\phi}_k$ 为分别有一个带有超参数 $\vec{\alpha}$ 和 $\vec{\beta}$ 的Dirilecht先验分布,即 $\vec{\theta}_m \sim \text{Dir}(\vec{\alpha}), \vec{\phi}_k \sim \text{Dir}(\vec{\beta})$ 。

那么基于LDA生成文档 $m$ 的过程如图2.3所示,首先确定文档长度 $M$ (服从 $\text{Poisson}(\zeta)$ 分布)即文档中单词的总数。然后对于文档 $m$ 中的每一个单词,我们从该文档对应的主题多项分布 $\vec{\theta}_m$ 中抽取一个主题 $z_{m,n}$ ,接着从主题 $z_{m,n}$ 所对应的词多项分布 $\vec{\phi}_k$ 中抽取一个单词 $w_{m,n}$ ,重复 $N_m$ 次就生成了文档 $m$ 。

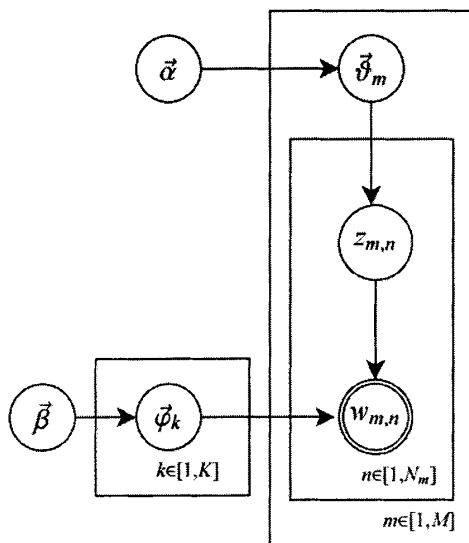


图 2.4 LDA 的图模型

Fig 2.4 Graph model of LDA

故生成文档的核心是训练得到文档-主题分布 $\vec{\theta}_m$ 和主题-词语分布 $\vec{\phi}_k$ ,用来训练的语料是现有的预处理后的语料集。目前主要采取 Gibbs Sampling 迭代获得这两个分布。训练过程中,设一个文档集 $D$ 由 $M$ 个文档组成,整个文档集 $D$ 中所有单词组成词汇表 $V$ ,其产生的参数和结果变量如表2.3所示:

表 2.3—基于 LDA 主题建模的参数解释

Tab 2.3 Parameter interpretation of LDA topic-model

参数名称	代表意义
$M$	文档集中的文档数
$K$	设定的主题数, 先验参数
$V$	词汇表中词汇总数
$\bar{\alpha}$	文档-主题 Dirichlet 分布的超参数, 先验参数
$\bar{\beta}$	主题-单词 Dirichlet 分布的超参数, 先验参数
$\bar{g}_m$	第 $m$ 个文档的主题分布, 表示成一个 $K$ 维的向量, 第 $i$ 项 ( $1 \leq i \leq K$ ) 表示该文档属于第 $i$ 个主题的概率
$\Theta$	是 $\bar{g}_m$ 的集合, 代表整个文档集的主题分布。即 $\Theta = \{\bar{g}_m\}_{m=1}^M$ , 是一个 $M * K$ 的矩阵, 行代表文档, 列代表主题, 矩阵元素为 $p\{z d=m\}$
$\bar{\varphi}_k$	第 $k$ 个主题的单词分布, 是一个 $V$ 维的向量, 第 $j$ 项 ( $1 \leq j \leq V$ ) 表示该主题属于第 $j$ 个单词的概率
$\Phi$	是 $\bar{\varphi}_k$ 的集合, 代表主题-单词层面的概率分布。即 $\Phi = \{\bar{\varphi}_k\}_{k=1}^K$ , 是一个 $K * V$ 的矩阵, 行代表主题, 列代表单词, 矩阵元素为 $p\{t z=k\}$

### 2.3.3 熵理论

熵的是从热力学中引入的概念, 物理学家香浓首次将这种概念引入到信息论中, 定义了信息熵以度量信息的不确定性, 所以信息熵也称为香浓熵<sup>[50]</sup>。在系统中也可以用信息熵来度量其有序化程度, 一个系统越无序, 那么它的信息熵就越高, 同理一个系统越有序那么信息熵相应也就越小。如果一个随机变量  $X$  的可能取值为  $X = \{x_1, x_2, \dots, x_n\}$ , 对应的概率为  $p = (X = x_i) \ (i=1, 2, \dots, n)$ , 则随机变量  $X$  的熵定义为

$$H = -\sum_{i=1}^n p(x_i) \log p(x_i) \quad (2.4)$$

基于信息熵的定义, 研究人员引入相互熵又称交叉熵, 是 Kullback-Leibler Divergence 的简称, 可以度量两个随机分布之间的相似程度。设  $p(x)$  和  $q(x)$  是  $X$  取值的两个概率概率分布, 则  $p$  对  $q$  的 KL-divergence 为

$$KL(p\|q) = \sum_{i=1}^n p(x_i) \log \frac{p(x_i)}{q(x_i)} \quad (2.5)$$

其物理意义表示的是已知概率分布  $p(x)$ ，若在相同的事件空间里，采用概率分布  $q(x)$  编码时，每个基本事件（符号）编码长度平均增加的幅度。那么若两个随机分布相同时，其相对熵则为零；若两个随机分布的差别越大时，它们的相对熵也会随之增大。

## 2.4 推荐效果的评价指标

为了评估推荐系统的优劣，研究人员主要根据推荐列表里用户实际的打分情况来衡量。目前的评价指标主要有查准率 (precision)<sup>[51]</sup>、查全率 (recall)<sup>[51]</sup>、MAP (mean average precision)<sup>[53]</sup>、NDCG (normalized discounted cumulative gain)<sup>[52]</sup> 等。

### (1) 查准率和查全率

设推荐系统向用户  $u$  推荐的  $M$  个物品记为  $R(u)$ ， $u$  在测试集上感兴趣的物品记为  $T(u)$ ，那么可以通过查准率、查全率来评估推荐算法的精度，公式为：

$$precision = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |R(u)|} \quad (2.6)$$

$$recall = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |T(u)|} \quad (2.7)$$

查准率体现了用户对被推荐物品感兴趣的程度，查准率越大，说明推荐的物品越容易引起用户的兴趣，其相应的推荐方法越好。而推荐查全率体现的是用户感兴趣的物品被列入推荐列表中的概率，查全率越大，说明该算法越可能为用户推荐感兴趣的物品。

### (2) 宏平均正确率 MAP

上述的两种评价指标只考虑了推荐的结果里用户感兴趣的物品数量，并没有考虑待推荐的物品之间排序。在实际体验中，推荐列表中用户感兴趣的物品越排序靠前，带来的用户体验越好，因此引入宏平均正确率 MAP。该指标由 Precision，Average Precision 和 Mean Average Precision 三个部分组成。首先计算得到推荐列表中位置  $k$  的查准率  $P@k(u)$ ，即推荐系统向用户  $u$  推荐的物品与  $u$  在测试集上感兴趣的物品交集的位置除以该物品在用户推荐列表  $R(u)$  中的位置，公式为：

$$P@k(u) = \frac{R(u) \cap T(u) @ k}{R(u) @ k} * n_k \quad (2.8)$$

其中， $n_k$  表示第  $k$  处推荐物品的得分，与用户相关为 1，否则为 0。

然后对用户  $u$  推荐列表中所有物品计算查准率得到 Average Precision：

$$AP(u) = \frac{\sum_{k=1}^{l_u} P@k(u)}{|R|} \quad (2.9)$$

其中,  $l_u$  表示推荐列表中所有物品,  $R$  为用户  $u$  推荐列表长度。

最后根据用户的  $AP$  求得宏平均正确率即为  $MAP$ , 公式为:

$$MAP = \frac{\sum_u AP(u)}{|U|} \quad (2.10)$$

通过  $MAP$  可以将推荐物品的排序位置考虑进来,  $MAP$  值越大说明算法能够将用户感兴趣的物品排序越靠前, 算法的推荐效果越好。

### (3) NDCG

计算  $NDCG$  前要计算用户  $u$  对推荐列表中位置  $k$  物品的  $DCG$ (discounted cumulative gain)值, 公式为:

$$DCG@k(u) = \sum_{i=1}^k G(u, i) D(i) \quad (2.11)$$

其中,  $G(u, i)$  反映的是物品和用户偏好的相关度程度, 由用户打分得到,  $D(i)$  为位置衰减函数。

由于不同的推荐方法推荐的数量很可能不同, 因此不能直接用  $DCG$  比较, 研究者计算了理想排序下的  $DCG$  值  $IDCG$ , 然后对所有用户求平均值, 获得算法的  $NDCG$  值, 公式为:

$$NDCG@k(u) = \frac{DCG@k(u)}{IDCG(u)} \quad (2.12)$$

## 2.5 本章小结

本章首先分析了微博的结构和特征, 然后针对其结构研究了爬虫和文本预处理技术。接着对本文用到的文本挖掘领域的方法及思想进行了介绍, 即  $TF-IDF$ 、 $LDA$ 、熵原理, 最后对推荐系统常用的评价指标进行了归纳总结, 为随后的研究奠定基础。

## 第三章 基于动态自适应权重的个性化微博推荐

本文的个性化推荐目的是为用户推荐朋友圈中有价值的信息，待推荐的内容主要为用户未浏览过的微博。本文提出了基于用户行为的会话划分方法，基于该方法获得用户未浏览过的微博，推荐系统推荐微博本质上就是对这些待推荐微博进行重排序，将用户可能感兴趣的微博往前排。

从目前研究来看，微博的重排序主要对三种对象构建特征，然后通过特征融合对待推荐微博进行重排序。这三类对象即待推荐的微博  $r$ 、微博的发布者  $v$  以及微博的接收者  $u$ ，研究人员从对象本身属性或者相互之间的关系构建了一些模型。例如基于用户  $u$  与发布者  $v$  之间的兴趣相似度建模，我们可以利用该模型将与用户  $u$  兴趣相似的发布者的微博推荐给用户；基于微博  $r$  质量的建模，我们可以将高质量的微博推荐给用户。本文在已有研究的基础上，从用户视角构建基于用户偏好的特征；从微博视角构建了基于微博内容的特征；基于发布者视角构建了基于发布者权威的特征。经过特征构建，待推荐的微博能够在各个特征上实现排序，但如何综合这些特征也是研究重点。本文结合应用场景，提出了动态自适应的特征权重计算方法，该方法能够根据微博在各个特征下的表现调整特征权重，从而将有特色的微博排在前面。用户基于该模型更容易捕捉到朋友圈中比较有特色的微博，下面将详细介绍本文的个性化推荐方法。

### 3.1 用户会话划分

用户登录微博后主要浏览最近离线期间好友推送的大量微博，因此本文研究的重点是对这些微博进行重排序，将有特色的微博往前排。用户的离线时间段即用户的会话可以根据用户登录和下线的时间来确定，但目前无法获取到用户的这些数据。为了得到用户的登录时间，从而确定推送范围，本文通过对用户点赞、转发、发布等行为发生的进行分析获得近似的登陆时间，依据登录时间能够对用户朋友圈微博进行会话划分。下面详细介绍本文用户会话划分方法。

假设用户  $u$  接收到的好友推送的微博信息集合表示为  $T = \{m_i \mid i = 1, 2, \dots, N\}$ 。会话的划分就是将  $T$  划分成连续的子集，其中每个子集包含的微博视为一个会话。那么对于第  $k$  次会话中的微博  $m$  表示为式 3.1：

$$T(k) = \{m \mid B(k) \leq t(m) < B(k+1), \forall m \in T\} \quad (3.1)$$

其中， $B(k)$  是第  $k$  次会话开始时间， $t(m)$  为微博  $m$  的发表时间。

会话划分的核心是获得  $B(k)$ ，本文通过用户历史行为数据分析近似得到  $B(k)$  的值。首先本文抓取了用户  $u$  转发或者发布过的微博，将这些微博表示为集合  $S = \{a_j \mid j = 1, 2, \dots, R\}$ ， $a_j$  是按照时间顺序由远及近排序的，如  $a_1$  表示的是能够

抓取的发布时间距离目前最远的微博。这些行为的发布时间可近似视为用户登录微博的时间，记  $a_j$  的发布时间为  $t(a_j)$ ，则用户发布或转发微博的时间用集合  $A = \{t(a_j) | j = 1, 2, \dots, R\}$  表示。在相邻两次用户发布或转发的时间段内，用户会收到好友推送的大量信息。但如果相邻时间差很小，用户很有可能一直在线，这个期间好友推送的微博往往会直接被用户浏览到，再对其推荐意义不大。因此本文将相邻时间差大于某个阈值的前节点作为划分会话的时间节点，式 3.2 表示了用户发布或转发的相邻的两条微博发布时间差。

$$H(a_{j+1}, a_j) = t(a_{j+1}) - t(a_j) \quad (3.2)$$

其中  $j$  的取值范围为  $j = 1, 2, \dots, R-1$ ，将  $H(a_{j+1}, a_j) > 60 \text{ min}$  的  $a_{j+1}$  筛选出来，记为式 3.3：

$$A_p = \{a_{j+1} | H(a_{j+1}, a_j) > 60 \text{ min}, \forall a_j, a_{j+1} \in A\} \quad (3.3)$$

将其近似作为用户登录时间，相邻的时间节点间产生的所有微博就属于同一会话。例如第  $k$  次会话中用户收到微博用  $m$  表示，那么下次会话开始时间  $B(k+1)$  表示为式 3.4：

$$B(k+1) = \min\{a | a \geq t(m), \forall a \in A_p\} \quad (3.4)$$

表 3.1 展示了通过用户发布或转发微博来划分会话的过程，其中  $K\text{-th session}$  表示会话编号， $m_i$  表示用户好友推送的信息。

如表 3.1 所示，本文通过会话划分将好友推送的微博信息片段化，每一个会话内的微博可作为用户登陆后的待推荐选项。实验中取研究对象近六个月的会话数据，将用户前四个月数据作为训练集，剩下的部分作为测试集。实验发现由于用户登录微博的频率是动态变化的，当用户登录频繁时，系统会出在用户登录后其对应的会话里待推荐的微博数量过少，不能够满足用户浏览需求。为解决该问题，本文基于假设——用户对转发的发布者其他文章也可能感兴趣，采集用户转发过但非其好友的微博作为推荐不足情况下的补充。例如用户  $u$  转发过的发布者集合记为  $P(u)$ ，用户  $u$  关注的好友集合记为  $G(u)$ ，则非用户关注的潜在感兴趣的发布者表示为式 3.5：

$$\hat{P}(u) = P(u) - P(u) \cap G(u) \quad (3.5)$$

确定了这些潜在感兴趣的用户后，本文对这部分用户的历史微博依据同样的指标  $A_p$  划分会话，从而将相同会话片段内微博作为补充。当系统需要为用户推荐  $N$  条微博时，为保证待推荐的微博中有足够优秀的微博，本文将待推荐的微博数设置为推荐目标数量  $p$  倍，即  $pN$  条。由于微博的实时性，待推荐的微博  $T'(k)$  应



尽可能属于一个会话区间里或数量上低于设定的阈值  $M$ ，表示为式 3.6:

表 3.1 用户会话划分实例

Tab 3.1 The instance of user session partition

微博列表	发布时间	$B(k+1)$	$K$ -th session
$m_1$	2015-08-10 08:10:30	2015-08-10 08:27:16	1
$m_2$	2015-08-10 08:17:31	2015-08-10 08:27:16	1
$m_3$	2015-08-10 08:20:27	2015-08-10 08:27:16	1
转发 $m_2$	2015-08-10 08:27:16		
$m_4$	2015-08-10 09:20:53	2015-08-10 10:31:22	2
$m_5$	2015-08-10 09:31:34	2015-08-10 10:31:22	2
$m_6$	2015-08-10 10:21:42	2015-08-10 10:31:22	2
发布微博	2015-08-10 10:31:22		
$m_7$	2015-08-10 10:40:16	2015-08-20 18:55:43	3
转发 $m_4$	2015-08-10 10:45:31		
$m_8$	2015-08-10 11:07:25	2015-08-20 18:55:43	3
...	...	...	...
$m_i$	2015-08-20 18:33:22	2015-08-20 18:55:43	3
转发 $m_j$	2015-08-20 18:55:43		

$$T'(k) = \begin{cases} T(k) & |T(k)| \geq \min\{pN, M\} \\ T(k) + \sum_{c=k-i}^k \hat{T}(c) & \left| T(k) + \sum_{c=k-i+1}^k \hat{T}(c) \right| \leq \min\{pN, M\} \leq \left| T(k) + \sum_{c=k-i}^k \hat{T}(c) \right| \end{cases} \quad (3.6)$$

其中， $|T(k)|$ 表示第  $k$  次待推荐微博的数量，当  $T(k)$  里微博数量不足时，模型则会从  $\hat{P}(u)$  中获取相应的会话片段的微博  $\hat{T}(k)$  作为补充，待推荐微博数量若依然不足则再取  $\hat{T}(k-1)$  会话数据作补充，以此类推，直到待推荐微博数量满足条件。该方法不仅能够保证待推荐的微博中有足够优质的微博，而且可以拓展用户信息来源，用户不通过关注机制也可以收到其他用户的微博，带来新的客户体验。

### 3.2 主题模型构建

如表 3.2 所示，微博的背后代表的是一些主题，微博的博主如果经常发布同一主题的微博则反映了用户对该主题有偏好。为了刻画这些主题偏好，本文采用 LDA 主题模型来推断用户的主题分布。如表 3.2、表 3.3 所示，用户发表的微博都由一些实体词构成，我们可以通过计算词的共线概率分析用户主题分布。

表 3.2 用户微博中词的主题分布

Tab 3.2 Topic distribution of user micro-blog

用户发表的微博:	主题: 1: 数码产品 2: 音乐 3: 美食 4: 娱乐八卦
小米 <sup>1</sup> 升级 <sup>1</sup> 了 V5 <sup>1</sup> 之后, 依然流畅 <sup>1</sup>	
张杰 <sup>4</sup> 哥哥 <sup>4</sup> . 不是突然爱你 <sup>4</sup> , 是一直都耐你 <sup>4</sup> . 期待杰锅 <sup>4</sup>	
林俊杰 <sup>2</sup> 新曲 <sup>2</sup> MV <sup>2</sup> 邀请滨崎步 <sup>2</sup> , 期待!	
烟台 <sup>3</sup> 苹果 <sup>3</sup> 真是好吃 <sup>3</sup>	
据说魅族 <sup>2</sup> 用户 <sup>2</sup> 忠诚度 <sup>2</sup> 仅次于小米 <sup>2</sup>	

表 3.3 用户在各个主题下的概率与词频

Tab 3.3 The probability and frequency of user under each topic

P(T2)=0.420		P(T1)=0.025		P(T3)=0.036		P(T4)=0.001		...	
T2: 数码产品		T1: 音乐		T3: 美食		T4: 娱乐八卦		...	
w	c(w)	w	c(w)	w	c(w)	w	c(w)	w	c(w)
小米	12	林俊杰	4	热干面	8	杰锅	5	...	...
魅族	8	MV	2	吃货	2	结婚	3	...	...
升级	5	滨崎步	2	苹果	1	汪峰	1	...	...

由于每篇微博字数不能超过 140 字, 属于短文本, 仅仅通过一条微博很难度量用户的主题, 因此本文将用户近期会话所发布的微博视为一个文档。为了抽取用户的主题分布, 本文采用采集了大量用户的微博进行文本预处理, 将处理后的语料集进行 LDA 训练。其中每位用户的微博语料对应到 LDA 模型中的一个文档, 采用 Gibbs Sampling 训练得到词项-主题概率分布  $\Phi$ 、文档-主题概率分布  $\Theta$ , 其矩阵表示如图 3.1 所示。

文档

$d_1 \ d_2 \ \cdots \ d_m$

主题

$z_1 \ z_2 \ \cdots \ z_k$

文档

$d_1 \ d_2 \ \cdots \ d_m$

词语

$\begin{Bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{Bmatrix}$

$\left\{ \begin{array}{c} \\ \\ \\ \end{array} \right\}$

$C$

$\left\{ \begin{array}{c} \\ \\ \\ \end{array} \right\}$

$=$

词语

$\begin{Bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{Bmatrix}$

$\left\{ \begin{array}{c} \\ \\ \\ \end{array} \right\}$

$\Phi$

$\left\{ \begin{array}{c} \\ \\ \\ \end{array} \right\}$

$\times$

主题

$\begin{Bmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{Bmatrix}$

$\left\{ \begin{array}{c} \\ \\ \\ \end{array} \right\}$

$\Theta$

$\left\{ \begin{array}{c} \\ \\ \\ \end{array} \right\}$

图 3.1 LDA 模型的矩阵示意图

Fig 3.1 Schematic diagram of matrix of LDA model

这里用到是 Gibbs Sampling 方法是一种典型的 Markov-Chain Monte Carlo 算法。

算法思想是每次选取概率向量的一个维度，然后通过其他维度的变量值  $\text{Sample}$  当前维度值，重复上述过程，直到分布收敛时输出待估参数。算法 3.1 展示了采用 Gibbs Sampling 训练通用主题模型的过程：第 1-10 步目的是随机为每个单词分配一个主题，统计单词  $l$  被分到主题  $k$  的次数  $n_m^{(k)}$ ，主题  $k$  被分配到文档  $m$  的次数；第 12-18 步计算排除当前词的主题分布，通过其他词主题分布估计当前词分配到各个主题概率。根据这个主题分布为该词  $\text{sample}$  一个新主题，依据同样方法更新下一个词的主题；第 19 步判断每个文档下主题分布  $\bar{\theta}_m$  及每个主题下词分布  $\bar{\phi}_m$  是否收敛，收敛则输出结果。

---

算法 3.1 Lda-Gibbs 过程

---

```

输入： 词索引  $\{\bar{w}\}$ ，主题数  $K$ ，超参数  $\alpha, \beta$ 

输出： 主题-词关联  $\{\bar{z}\}$ ，词项-主题概率分布  $\Phi$ ，文档-主题概率分布  $\Theta$ 

1:  初始化所有变量  $n_m^{(k)}, n_m, n_k^{(l)}, n_k$ 
2:  for all documents  $m \in [1, M]$  do
3:      for all words  $n \in [1, N_m]$  in document  $m$  do
4:          sample topic index  $z_{m,n} = k \sim \text{Mult}(1/K)$ 
5:          increment document-topic count:  $n_m^{(k)} += 1$ 
6:          increment document-topic sum:  $n_m += 1$ 
7:          increment topic-term count:  $n_k^{(l)} += 1$ 
8:          increment topic-term sum:  $n_k += 1$ 
9:      end for
10: end for
11: while not finished do //判断迭代次数是否超过设定次数
12:     for all documents  $m \in [1, M]$  do
13:         for all words  $n \in [1, N_m]$  in document  $m$  do
14:             decrement counts and sums:  $n_m^{(k)} -= 1; n_m -= 1; n_k^{(l)} -= 1; n_k -= 1$ 
15:             sample topic index  $k \sim P(z_i | \bar{z}_{-i}, \bar{w})$ 
16:             increment counts and sums:  $n_m^{(k)} += 1; n_m += 1; n_k^{(l)} += 1; n_k += 1$ 
17:         end for
18:     end for
19:     if converged read out result //收敛后读出训练的结果
20:     return  $\{\bar{z}\}, \Phi, \Theta$ 
21: end while

```

---

基于训练好的通用主题模型，采用 Gibbs Sampling 方法推断待分析的文档的词在主题下的分布：

$$p(z_w = i) \propto p(\phi_i | M) p(z_w | \phi_i) \quad (3.7)$$

其中  $w$  为文档  $M$  中的一个词， $z_w$  为该词的主题， $p(\phi_i | M)$  和  $p(z_w | \phi_i)$  则分别对应着 Dirichlet 分布在贝叶斯框架下的参数估计， $p(z_w | \phi_i)$  来自与上文自训练好的模型。本文根据词在主题下的分布统计了该文档的主题-词语共现概率矩阵，最后推导出待分析的文档的主题分布  $\theta_{w,k}$ 。

### 3.3 微博特征构建

本文通过用户会话的划分完成了推荐主体的构建，获得每次待推荐的微博信息集合  $Q = \{m_i | i=1,2,\dots,V, V \geq 10 * N\}$  ( $N$  为待推荐的微博数量)，微博个性化推荐的目标就是构建排序函数  $H(r,u,v)$  ( $r$  代表微博， $u$  代表用户， $v$  代表发布者) 对  $Q$  进行重排序，将  $Q$  中有特色的微博排在前面。为了得到排序函数，本文从用户偏好、微博内容、发布者权威三个维度建模，从而构建微博特征模型以实现微博的量化。本文将微博特征分为分析特征和直接特征，直接特征指的是从爬取的微博数据中能直接获取到，或者只需经过简单提取计算就能得到的特征；分析特征是指需要统计方法或者适当的模型才能获得的特征，而这些特征对微博推荐具有更加重要的作用。

#### 3.3.1 基于用户偏好的特征构建

当用户选择微博时，用户固有的偏好对用户选择往往具有很大影响。例如，有些用户对科技数码感兴趣，那么他往往会关注有共同爱好的博主，并分享转发这类微博。除此之外，用户对身边亲朋好友的最新微博也会格外的关注。为量化用户的偏好，本文从用户主题偏好和行为偏好两个视角分析用户偏好。

##### (1) 兴趣相似度

从用户主题偏好的视角，用户在选择浏览微博时，偏向于浏览和自己主题相似的作者的微博。例如，用户对汽车感兴趣，则会经常转发一些相关文章，而经常发表这方面文章的用户更容易吸引其注意。本文利用训练出来的通用主题模型 Gibbs 抽样得到用户  $u$  与博主  $v$  的主题分布  $\theta_u, \theta_v$ ，采用 KL-divergence 度量用户与用户之间的兴趣相似度，从而得到分析特征：

$$Similar(u,v) = 1 / KL(\theta_u || \theta_v) \quad (3.8)$$

##### (2) 交互 TF-IDF

从用户行为偏好的视角看，用户的评论、转发、提及这些行为表征了用户与

好友的亲密度，经常评论、转发、提及的好友微博可能是用户比较关心的。而当用户关注的好友列表中存在异常活跃的用户时，那些与用户亲密度高的用户微博很大可能被前者“刷屏”。

为了能将这些不活跃但与用户亲密度较高的微博推荐给用户，本文引入 2.3.1 里介绍的 TF-IDF 算法思想，即字词的重要性一方面随着它在文件中出现的次数累积而成正比增加，但同时会随着它在语料库中出现的次数增长而成反比下降。本文提出用户 TF-IDF：对用户来说，具有推荐意义的博主，应该是在过去一段时间内被用户转发、评论、提及频率高的，而在这段时间会话中，活跃度却比较低的博主。根据所提方法，首先统计研究片段中用户转发、评论、提及的博主次数  $\{n_v | v = 1, 2, \dots, M\}$ ，每位博主的 TF 值表示为：

$$tf_v = \frac{n_v}{\sum_{v=1}^M n_v} \quad (3.9)$$

通过会话划分计算这段时间的会话次数  $|D|$ ，统计被用户转发、评论、提及的博主在这段时间会话里出现的次数为  $\{l_v | v = 1, 2, \dots, M\}$ ，得到每位博主的 IDF 值：

$$idf_v = \log \frac{|D|}{l_v + 1} \quad (3.10)$$

由 3.9 和 3.10 可以计算出用户  $u$  对博主  $v$  交互 TF-IDF，从而得到分析特征：

$$Transmit(u, v) = tf_v \times idf_v \quad (3.11)$$

### 3.3.2 基于微博内容的特征构建

本文主要从微博与用户兴趣的关联度、微博质量、微博的传播力三个方面评估微博内容。从微博与用户兴趣的关联度角度，本文提出一种改进的计算微博与用户主题相关性的指标；从微博质量角度本文统计微博长度、附加连接、标签数量、@数量等直接特征作为评估指标；从微博的传播力角度，本文提出微博热度来衡量微博的社交性。

#### (1) 主题关联

由 3.2 的方法我们得到通用的主题模型，然后使用 Gibbs 抽样可以获取该条微博的主题分布。实验发现由于微博长度较短，有些微博经过数据预处理、分词后只剩个别词，那么经过 Gibbs 抽样得到的主题分布很可能接近均匀分布。本文在利用传统的主题相似性度量指标如 KL-divergence 距离并不能很好的度量一条微博与用户主题的匹配程度，例如图 3.2 中，对于用户会话中两条微博即 Weibo1、Weibo2 而言，从图上可以明显看出 Weibo1 与用户的主题偏好更相似，但 Weibo1、Weibo2 基于  $Similar(r, u)$  方法得到的结果分别是 0.519356、0.55295，即没有明确的主题的微博反而与用户主题更相关。

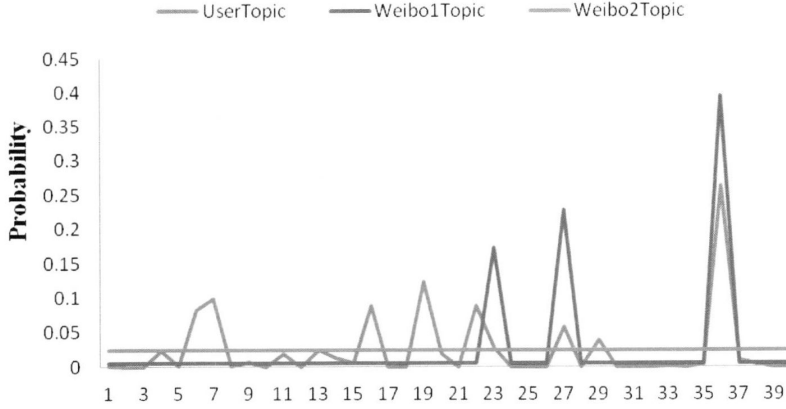


图 3.2 用户和微博主题分布示意图

Fig 3.2 User and micro-blog topic distribution diagram

为解决上述问题，本文提出一种改进的适用于超短文本的主题相似性度量方法。首先本文用标准差来度量微博主题的鲜明程度，即  $\sqrt{D(\theta_r)}$ ， $\theta_r$  为微博的主题分布，主题模糊的微博其  $D(\theta_r)$  就会相应偏小；然后结合 KL-divergence 距离计算出微博  $r$  与用户  $u$  的主题关联，得到如下分析特征，经计算 Weibo1、Weibo2 基于  $Relativity(r, u)$  方法得到的结果分别是 0.040159、0，即 weibo1 与用户主题更相关，说明改进的度量方法能够反映真实情况。

$$Relativity(r, u) = \sqrt{D(\theta_r)} \times \frac{1}{KL(\theta_r \| \theta_u)} \quad (3.12)$$

#### (2) 附加链接数量

微博的最大长度只有 140 字，用户附上其他连接地址能够传达更丰富的信息。因此链接的数量体现了微博所蕴含的其他信息量。本文通过正则匹配识别并统计微博中附加链接，从而得到直接特征：

$$Url(r) = \begin{cases} u(r) & \text{if result contains url} \\ 0 & \text{otherwise} \end{cases} \quad (3.13)$$

#### (3) @博主数量

如果微博中涉及到提及了其他博主，说明该微博具有这更丰富的社交信息，更容易被转发、评论、回复，其带有的话题也更容易被用户关注，本文通过正则匹配识别并统计微博中提交的博主数量，从而得到直接特征：

$$Refer(r) = \begin{cases} m(r) & \text{if result contains @} \\ 0 & \text{otherwise} \end{cases} \quad 3.14$$

#### (4) 标签数量

标签是对微博主题信息的有效概括，蕴含了大量有用的信息。用户通过标签

不仅可以对微博进行归类，而且表明其发布的微博带有鲜明的主题，更可能是用户感兴趣的。本文通过正则匹配识别并统计微博中标签，从而得到直接特征：

$$Hashtag(r) = \begin{cases} h(r) & \text{if result contains hashtag} \\ 0 & \text{otherwise} \end{cases} \quad (3.15)$$

#### (5) 微博长度

通常而言，微博长度  $l(r)$  越长说明用户撰写微博所花的精力就越多，微博传达的信息越丰富，其质量也越高。由于抓取的微博中包含了附加链接、@博主、标签等数据，这些数据在上文中已经统计了数量，故本文在统计微博长度前，先过滤掉附加链接、@博主、标签，再统计  $l(r)$ ，从而得到直接特征：

$$Length(r) = l(r) \quad (3.16)$$

#### (6) 微博热度

转发数量  $re(r)$  是微博质量的重要体现， $re(r)$  多说明微博  $r$  的受众广。与此同时，微博的评论数量  $d(r)$ 、点赞数量  $g(r)$  也表征了大众对这条微博的关注程度。本文将微博的转发数量、评论数量、点赞数量，作为该微博的总热度，由于总热度受用户粉丝数影响较大，本文将总热度除以该微博的发布者  $v$  的粉丝数。值的大小直接体现了微博传播的热度，值越大，说明大众对这条微博都普遍关注，故可作为推荐的依据，从而得到分析特征：

$$Heat(r, v) = \frac{re(r) + d(r) + g(r)}{fans(v)} \quad (3.17)$$

### 3.3.3 基于发布者权威的特征构建

微博相比其他社交媒体，具有更多的明星效应。被新浪认证的用户或者粉丝多的用户，其发布微博时会考虑到社会效应，故其微博更具有权威性、影响性。用户浏览微博时，会潜在受到微博发布者的这些因素影响，例如浏览的微博发布者中有多人都聊到算法领域的话题，用户可能更关注粉丝数相对较多并且被认证的发布者微博。本文采用用户影响力及新浪认证情况度量发布者的权威性。

#### (1) 影响力

判断一位发布者是否有影响力，最直观的指标就是看其粉丝的多少，如果其经常传播有用的信息或者是明星大咖，往往会有很多粉丝。但是当用户只是活跃于微博社交而不是发布高质量的微博，他的粉丝也会很多，但区别是这类用户的关注人数也很多。故本文评估用户影响力时将其关注数和粉丝数同时考虑进来，将该直接特征定义为：

$$Influence(v) = \frac{i(v)}{o(v) + i(v)} \quad (3.18)$$

而  $o(v)$  表示用户的出度，本文取用户的关注数， $i(v)$  表示用户的入度，本文

取用户的粉丝数，影响力越高的用户 *Influence* 值越接近于 1。

## (2) 认证名人

如果作者是名人，那么其会通过新浪的认证。用户发布的微博更具权威性，新浪提供为草根博主以及明星等提供多种角色的认证，本文通过抓取的个人信息判断是否是认证名人记，定义该直接特征为：

$$Famous(v) = \begin{cases} 1 & \text{if } a \text{ is public character} \\ 0 & \text{otherwise} \end{cases} \quad (3.19)$$

## 3.4 基于动态自适应的特征权重构建

基于上述微博特征的提取，本文得到 4 种分析特征即兴趣相似度、交互 TF-IDF、主题关联、微博热度，6 种直接特征即附加链接数量、@博主数量、标签数量、微博长度、影响力、认证名人。我们能够通过这些微博特征对待推荐的每条微博信息进行打分，但基于每种特征的打分得到的排序都不一样，如何融合这些指标获得排序函数是下一步工作的重点。

为解决多种指标融合的问题，研究人员考虑将排序学习技术融合进推荐算法之中。其思想是通过对某个博主历史数据进行 Learn To Rank，计算出排序函数，根据排序函数来计算微博得分。那么对于用户  $u$  浏览的一条由  $v$  发布的微博  $r$  来说，其微博得分表示为：

$$\begin{aligned} H(r, u, v) = & w_1 * Similar(u, v) + w_2 * Relativity(r, u) + w_3 * Transmit(u, v) \\ & + w_4 * Length(r) + w_5 * Url(r) + w_6 * Hashtag(r) + w_7 * Refer(r) \\ & + w_8 * Heat(r, v) + w_9 * Influence(v) + w_{10} * Famous(v) \end{aligned} \quad (3.20)$$

其中，参数  $w_i (i=1, 2, \dots, 10)$  为用于控制各个特征权重。

通过排序学习得到的  $\tilde{w}_i$  可以很好的融合多方面的特征，但实际操作可能面临一些困难：一方面采用有监督的学习方法费时费力，因为训练集中微博之间的偏好很难界定，通过人为打分可能存在主观上的偏差，并且当研究对象发生变化时，需要重新训练  $\tilde{w}_i$ ；另一方面对同一个用户而言，随着时间的推移，影响其浏览微博的主要因素可能也在变。例如，某段时间内用户  $v$  可能只倾向于某个关系亲密的好友微博，体现在用户可能经常转发和评论其微博。那么  $Transmit(u, v)$  指标相对于其他指标有更明显的区分度。而过了一段时间用户兴趣发生转移，用户只对一些主题感兴趣，那么  $Similar(u, v)$ ， $Relativity(r, u)$  指标相对于其他指标可能有更明显波动。如果采用训练好的固定权重指标，不能适用于这种转变，因此本文提出了一种动态自适应的特征权重计算方法 (dynamic self-adaptive feature weighting, DAFW)，处理待推荐微博动态变化情况下，对特征项权重进行动态自



适应计算和调整。

算法首先将这十个特征分组编号记为  $X^{(r)} = \{x_i^{(r)} | i=1,2,\dots,10\}$ ，表示微博  $r$  在特征上得分。然后运用 2.3.3 介绍的信息熵，根据指标变异性的的大小来确定客观权重，步骤如下：

#### (1) 数据标准化

将各个指标的数据进行标准化处理，这里用 Min-max 标准化，标准化后的值表示为：

$$Y(x_i^{(r)}) = \begin{cases} \frac{x_i^{(r)} - \min_{r' \in Q}(x_i^{(r')})}{\max_{r' \in Q}(x_i^{(r')}) - \min_{r' \in Q}(x_i^{(r')})} & \max_{r' \in Q}(x_i^{(r')}) > \min_{r' \in Q}(x_i^{(r')}) \\ 0 & \max_{r' \in Q}(x_i^{(r')}) = \min_{r' \in Q}(x_i^{(r')}) \end{cases} \quad (3.21)$$

#### (2) 求各指标的信息熵

根据信息论中信息熵的定义，求得各个特征的信息熵：

$$E_i = \begin{cases} -\ln(|Q|)^{-1} \sum_{r=1}^{|Q|} p_i^{(r)} \ln p_i^{(r)} & \max_{r' \in Q}(x_i^{(r')}) > \min_{r' \in Q}(x_i^{(r')}) \\ 0 & \max_{r' \in Q}(x_i^{(r')}) = \min_{r' \in Q}(x_i^{(r')}) \end{cases} \quad (3.22)$$

其中

$$p_i^{(r)} = \begin{cases} \frac{Y(x_i^{(r)})}{\sum_{r=1}^{|Q|} Y(x_i^{(r)})} & \max_{r' \in Q}(x_i^{(r')}) > \min_{r' \in Q}(x_i^{(r')}) \\ 0 & \max_{r' \in Q}(x_i^{(r')}) = \min_{r' \in Q}(x_i^{(r')}) \end{cases} \quad (3.23)$$

若  $p_i^{(r)} = 0$ ，则定义  $\lim_{p_i^{(r)} \rightarrow 0} p_i^{(r)} \ln p_i^{(r)} = 0$ ，信息熵  $E_i$  越小，表明指标值得变异程度越大，提供的信息量越多，在综合评价中所能起到的作用也越大，其权重也就越大。相反，某个指标的信息熵  $E_i$  越大，表明指标值得变异程度越小，提供的信息量也越少，在综合评价中所起到的作用也越小，其权重也就越小。

#### (3) 归一化指标权重

由信息熵计算得到各指标的差异系数  $1 - E_i$ ，从而确定各自特征的熵权

$$\hat{w}_i = \frac{1 - E_i}{10 - \sum_{i=1}^{10} E_i} \quad (3.24)$$

本文数据标准化采用用户在特征上的值  $x_i^{(r)}$  除以该特征下的平均值  $\bar{x}$ ，标准化后的特征值反映了在该特征值下微博偏离均值的水平。本文首先计算待推荐微博集合  $Q$  在各个指标下的均值：

$$\bar{x}_i = \frac{\sum_{r=1}^{|Q|} x_i^{(r)}}{|Q|} \quad (3.25)$$

其中,  $|Q|$  表示待推荐的微博数量, 将各个特征除以均值, 得到各个特征相对于均值的倍数:

$$\hat{x}_i^{(r)} = \frac{x_i^{(r)}}{\bar{x}_i} \quad (3.26)$$

由于特征与特征之间也存在主次之分。本文提取的分析特征可能对推荐的质量起主要作用, 而有些特征对推荐的质量起次要作用, 因此本文引入参数  $w_\theta$ , 用于控制各个特征的主次之分, 结合  $\hat{w}_i$ 、 $\hat{x}_i^{(r)}$ , 本文得到了微博  $r$  的排序函数:

$$H(r, u, v) = \sum_{i=1}^{10} w_\theta \hat{w}_i \hat{x}_i^{(r)} = \sum_{i=1}^{10} \frac{w_\theta \hat{w}_i}{\bar{x}_i} x_i^{(r)} \quad (3.27)$$

其中特征权重表示为  $w_i = w_{\theta i}(\bar{x}_i)^{-1} \hat{w}_i (i=1, 2, \dots, 10)$ , 当待推荐的微博数据变动后, 其特征权重也会动态自适应地变动, 从而将各个特征下有明显优势的微博推荐给用户。

本文构建的个性化微博推荐的整体算法如 3.2 所示, 输入为系统根据会话划分方法获得第  $k$  次会话期间用户朋友圈推送的微博, 即待推荐微博  $T'(k)$ 。同时用户历史微博  $W(u)$ 、通用主题模型  $\Phi$ 、发布者历史微博  $w(v)$  及个人信息也作为潜在的输入项。第 2-6 步遍历  $T'(k)$  的发布者集合  $V$ , 算法通过用户  $u$  与发布者  $v$  主题相似度和历史交互记录计算得到分析特征值  $Similar(u, v), Transmit(u, v)$ , 并根据发布者个人信息计算得到直接特征值  $Influence(v), Famous(v)$ 。这些特征值计算量大, 因此本文采用用户近期数据离线计算, 将结果存于数据库中。第 7-12 步对  $T'(k)$  中的微博  $r$  计算在线特征值。首先统计微博  $r$  中链接、标签、@用户、转发、评论等数量, 得到直接特征值  $Url(r), Hashtag(r), Refer(r), Heat(r, v)$ ; 然后算法对微博  $r$  进行文本预处理, 计算处理后的文本长度得到特征值  $Length(r)$ ; 最后算法对文本进行主题抽取, 计算得到文本与用户之间主题的相似度指标  $Relativity(r, u)$ 。第 13-22 步根据特征值在待推荐微博中的变异情况, 算法计算了待推荐微博各个特征的动态权重。第 23-25 步算法根据微博在各个特征下的打分  $x_i^{(r)}$  及动态权重  $w_i$  得到综合得分  $H(r, u, v)$ 。最后算法根据微博综合得分重排序, 将得分高的微博排在前面。

## 算法 3.2 基于 DAFW 的微博个性化推荐

---

输入: 待推荐微博  $T'(k)$

隐输入: 用户历史微博  $W(u)$ 、通用主题模型  $\Phi$ 、发布者历史微博  $w(v)$  及个人信息

输出: 重排序的微博  $R(T'(k))$

- 1:  $w(u)$  基于  $\Phi$  Gibbs Sampling 用户  $u$  兴趣分布  $\theta_u$
- 2: **for**  $v \in V$  **do** //遍历发布者
- 3:  $w(v)$  基于  $\Phi$  Gibbs Sampling 发布者  $v$  兴趣分布  $\theta_v$
- 4:  $x_1^{(r)} = \text{Similar}(u, v), x_3^{(r)} = \text{Transmit}(u, v)$
- 5:  $x_9^{(r)} = \text{Influence}(v), x_3^{(r)} = \text{Famous}(v)$
- 6: **end for**
- 7: **for**  $r \in T'(k)$  **do** //遍历待排序微博
- 8:  $x_5^{(r)} = \text{Url}(r), x_6^{(r)} = \text{Hashtag}(r), x_7^{(r)} = \text{Refer}(r), x_8^{(r)} = \text{Heat}(r, v)$
- 9: text preprocess,  $x_4^{(r)} = \text{Length}(r)$
- 10: 基于  $\Phi$  Gibbs Sampling  $\theta_r$
- 11:  $x_2^{(r)} = \text{Relativity}(r, u)$
- 12: **end for**
- 13: **for**  $w_i \in W$  **do** //遍历特征
- 14: **for**  $x_i^{(r)} \in X^{(r)}$  **do** //遍历微博的特征值
- 15: Max-Min 标准化特征值  $Y(x_i^{(r)})$
- 16:  $p_i^{(r)} = Y(x_i^{(r)}) (\sum_{i=1}^{|Q|} Y(x_i^{(r)}))^{-1}$
- 17:  $E_i += -\ln(|Q|)^{-1} p_i^{(r)} \ln p_i^{(r)}$
- 18: **end for**
- 19:  $\hat{w}_i = (1 - E_i) (10 - \sum_{i=1}^{10} E_i)^{-1}$
- 20:  $\hat{x}_i^{(r)} = x_i^{(r)} \bar{x}_i^{-1}$
- 21:  $w_i = w_{\partial i} (\bar{x}_i)^{-1} \hat{w}_i$
- 22: **end for**
- 23: **for**  $r \in T'(k)$  **do** //遍历待排序微博
- 24:  $H(r, u, v) = w_i x_i^{(r)} (i = 1, 2, \dots, 10)$
- 25: **end for**
- 26:  $R(T'(k)) = R(H(r, u, v))$
- 27: **return**  $R(T'(k))$

---

### 3.5 本章小结

本章详细介绍了基于动态自适应权重的个性化微博推荐方法。首先通过分析微博用户的转发、发布微博的时间，近似得到用户的登录时间，基于此本文确定了微博的推荐范围；其次，通过 LDA 训练，获得通用主题模型，采用 Gibbs 抽样获得文本的主题分布；然后从用户偏好、微博内容、发布者权威三个维度对微博特征进行量化；最后对本文提出的基于 DAFW 的个性化微博推荐方法做了详细介绍。

## 第四章 实验验证和原型系统

本文提出了基于动态自适应权重的个性化微博推荐方法，主要从用户兴趣、微博内容、发布者权威角度构建了微博特征，基于 DAFW 方法融合特征，实现对待推荐微博的重排序。为了证明方法的有效性，本文抓取了真实用户的微博数据，编码实现了本文的推荐算法。在证明方法的有效性后，本文设计并开发了个性化微博推荐的原型系统。本文的实验环境为（1）硬件配置：Intel(R) i5-3110M 3.2GHz, 12G 内存；（2）操作系统：Win7 64 位；（3）开发环境：Eclipse, Navicat for MySQL, R2012, EditPlus。本章详细介绍实验过程及原型系统。

### 4.1 数据收集和预处理

为了验证本文所提方法的有效性，本文通过爬虫采用模拟登陆方式采集新浪微博的相关数据。首先，我们通过热门话题下的评论列表随机选取了 6830 个用户，通过 7 台电脑合作抓取了约 311 万条微博用于通用 LDA 主题模型的训练；然后，选择了半年发布的微博数量多于 200 以及注册时间满一年的 100 位用户，通过网络爬虫获取其关注的好友列表及用户的基本信息；接着根据好友列表抓取好友所发的微博及好友的基本信息，用于微博特征提取；最后，统计用户转发过发布者，本文将不属于其关注的好友的发布者提取出来，再抓取这部分发布者的基本信息及其微博用于待推荐微博数量不足情况下的补充。

由于爬取的微博语料含有大量超链接、颜文字、标签等信息，因此在训练 LDA 模型前需要对文本数据进行预处理，减少文本噪声。第二章已介绍了文本预处理的方法，首先通过正则表达式清洗超链接、标签等无用内容；再通过正则提取中英文及数字微博信息并将其繁体转简体；采用上海林原信息科技有限公司开发的自然语言处理包 HanLP 来分词，并完成词性过滤、去停用词、词频统计，将高频词中噪音词放入停用词中；重复上一步操作直至没有高频噪音词；最终将处理好的用户微博数据以用户为单位，划分成文档，本文共获得 6824 篇文档用于 LDA 模型训练，图 3.2 展示了文本预处理的流程。

由于用户会话列表中微博信息无法直接获取，所以本文将用户及其关注的好友的历史微博信息按照时间序列组合在一起，形成用户的朋友圈。为保证用户关注好友的稳定性，本文取用户近半年的数据为研究对象。当为用户的第  $k+1$  次会话推荐微博时，首先根据公式 3.6 取得第  $k$  次会话期间需要待推荐的微博  $T'(k)$ ；然后取这些微博的发布者及用户前两个月微博作为语料；本文基于上面训练好的通用模型，获得近期研究对象的兴趣分布，并根据用户前两个月会话及基本信息为每条微博打分；我们利用 DAFW 得到动态后算的微博综合分数；最后根据分数

排序，将得分高的推荐给用户。图 4.2 展示了本次实验的完整流程。

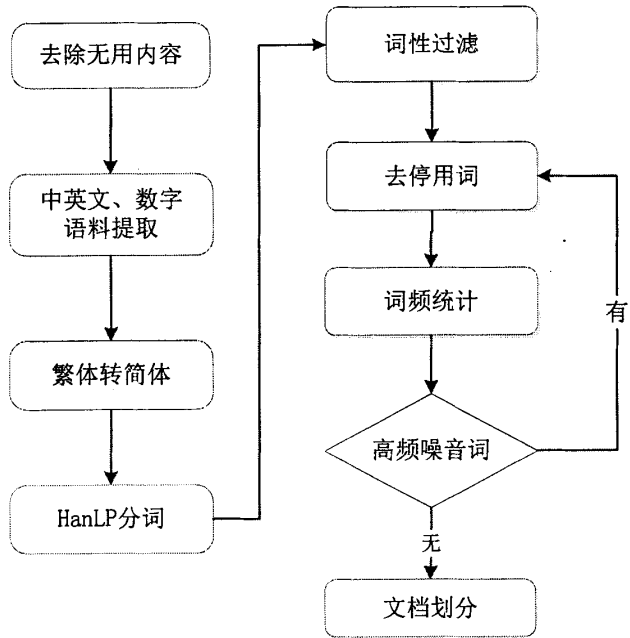


图 4.1 文本预处理流程图

Fig 4.1 The flowchart of text preprocessing

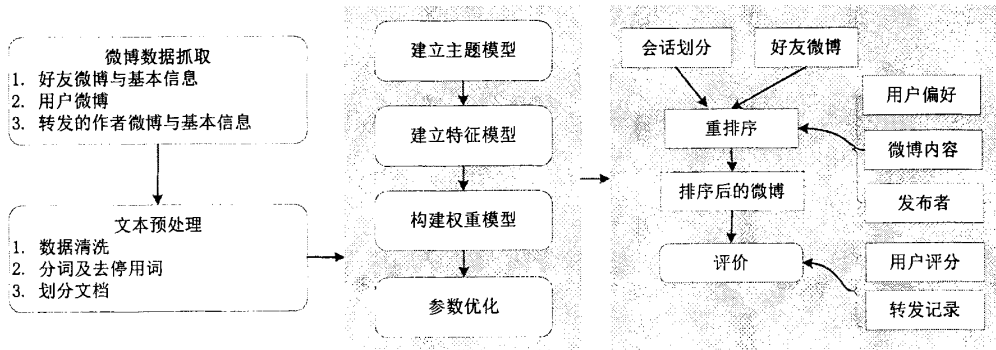


图 4.2 实验流程图

Fig 4.2 The flowchart of experimental

## 4.2 实验及结果分析

### 4.2.1 评价方法

为了评价推荐质量, 本文取用户最近会话里待推荐的微博, 将其随机排序交由用户打分。如果待推荐的微博是用户喜欢的, 用户打 1 分, 否则打 0 分。接着依据推荐模型为微博重排序, 由于微博为短文本, 用户浏览推荐结果比较快, 浏览的数量会比较多, 因而在本实验中截取重排序后得分排名前 50 的微博作为推荐结果。根据打分结果, 本文选用第二章介绍的  $P@N$  (查准率)、 $MAP$  (宏平均准确率) 指标进行评价, 计算公式分别为

(1)  $P@N$

$$P@N_r = \frac{\sum_{i=1}^L n_r^i}{N_r L} \quad (4.1)$$

其中  $n_r^i$  表示推荐的前  $N_r$  条微博中用户喜欢的微博数量,  $L$  为参与评分的用户数量。

(2)  $MAP(N)$

$$MAP(N_r) = \frac{\sum_{r=1}^{|N_r|} P@N_r}{|N_r|} \quad (4.2)$$

其中  $|N_r|$  表示截断水平种数,  $MAP$  越高, 说明推荐的质量就越高。

除了上述两种指标, 本文从推荐的精度和推荐多样性角度提出以下两种评价指标。

(3)  $HIR$

本文假定在一次用户会话里用户转发、评论的微博相比于其他的用户没有转发、评论的微博更相关。因此重排序后, 这样的微博在会话里被排在越靠前的位置, 代表用户越容易看到自己喜欢的微博, 指标计算公式如 4.3 所示:

$$HIR = \frac{1}{(e-s+1)L} \sum_{i=1}^L \sum_{k=s}^e \frac{rank(T^i(k))}{|T^i(k)|} \quad (4.3)$$

其中  $|T^i(k)|$  表示用户  $i$  的第  $k$  次会话待推荐列表数量,  $rank(T^i(k))$  表示重排序后, 被用户  $i$  转发的微博在会话  $k$  中位置,  $s, e$  分别表示统计的会话开始和结束的位置,  $HIR$  值小, 说明用户可以越早发现想要转发的微博。

(4)  $D@N$

从用户体验角度来说, 由于推荐的内容都来自于关注的好友, 若只追求推荐的精度, 可能造成推荐的多样性降低。例如每次只推荐某一个用户的微博, 或者

只推荐某个主题的微博，这样带来的用户体验也会大打折扣。本文假设在不降低推荐精度的情况下，用户更喜欢包含更多作者的推荐列表，描述推荐的多样性用公式 4.4 表示：

$$D@N_r = \frac{1}{(e-s+1)L} \sum_{i=1}^L \sum_{k=s}^e \frac{dis(T_{N_r}^i(k))}{N_r} \tag{4.4}$$

其中  $N_r$  表示推荐分数靠前的微博数量即截断水平， $dis(T_{N_r}^i(k))$  表示重排序后用户  $i$  的第  $k$  次会话中前  $N_r$  条微博的去重的发布者数量，例如前 10 条微博都是由一个作者发布的，那么  $dis(T_{N_r}^i(k))=1$ 。 $D@N$  越大，说明单位微博包含的作者数量越多，用户体验更好。

4.2.2 基准模型

为了验证本文提出的模型在微博推荐中的有效性，我们将其与一些基准模型进行对比，本文主要从以下四个模型来比较推荐效果。

Sina: 新浪依据发布时间来排序的方法，即发布时间越接近现在的排序越靠前，这是用户目前浏览微博时呈现的方式。

RankNet: 通过 RankLib 库实现了 RankNet 算法，该算法是基于 Pairwise 的排序学习方法，即通过训练集学习出排序函数，基于排序函数排序。

DAFW-D: 表示仅考虑直接特征的基于 DAFW 的微博推荐模型。

DAFW-A: 表示加入分析特征的基于 DAFW 的微博推荐模型，即本文提出的模型。

4.2.3 实验结果

为了对上述模型进行训练，本文对相关参数的设置如表 4.1 所示。

表 4.1 模型参数

Tab 4.1 Parameters of the model

参数	描述	值
M	待推荐微博数量上限	400
p	待推荐微博数量下限与推荐数量的比	5
$w_o$	调整特征权重	(2,2,2,1,1,1,1,2,1,1)
L	参与打分的用户数	100



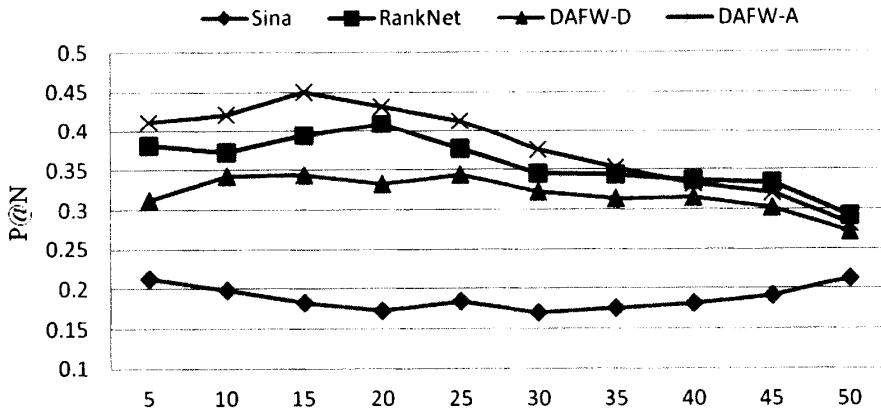


图 4.3 动态自适应权重模型与基准模型的 P@N 对比

Fig 4.3 P@N comparison of DAFW model and benchmark model

图 4.3 和图 4.4 给出了本方法与基准方法的效果对比。由图 4.3、图 4.4 可以看出，与新浪默认的按照时间排序的 Sina 方法相比，其他的三种模型都能明显的提高推荐质量，特别是在推荐的前 20 条微博内，效果更加明显。由于待推荐的微博数量一定，质量高的微博数量有限，这就形成当推荐数量慢慢增加时到 50 时，后面三种推荐模型准确率有所下降。而本文所提出的 DAFW-A 方法整体上也较 RankNet 方法好些。而且从图中可以明显看到，随着分析特征的加入，推荐的质量有了一个明显的提升，这表明，本文提出的多种分析特征能有效提高推荐质量，通过动态自适应的权重模型能够有效的融合多种指标，对提高微博个性化推荐效果具有较好的作用。

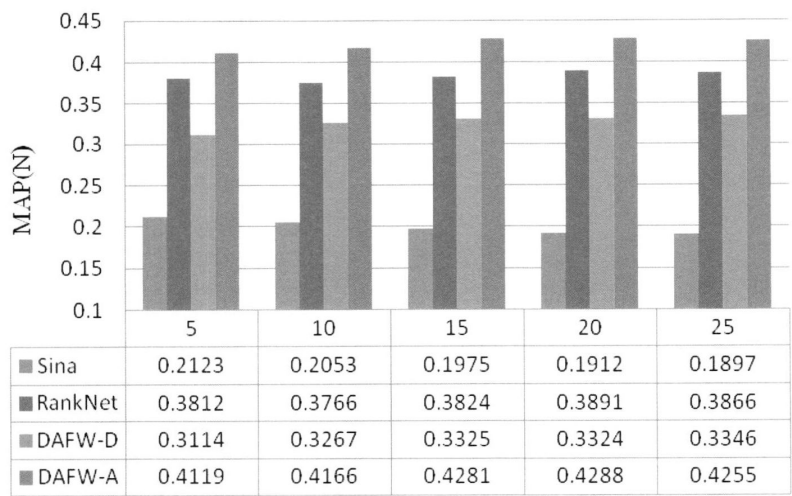


图 4.4 动态自适应权重模型与基准模型的 MAP 对比

Fig 4.4 MAP comparison of DAFW model and benchmark model

从图 4.5 中可以看出，基于时间排序的 Sina 方法 HIR 值最高，说明用户看到自己想要转发的微博花的时间最长。而本文提出方法能将用户想要转发的微博明显往前排，用户更容易获取到自己感兴趣的内容。

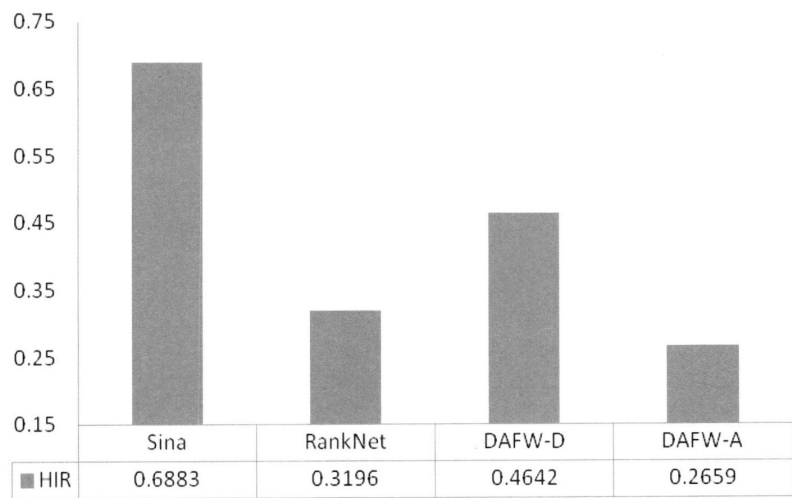


图 4.5 动态自适应权重模型与基准模型的 ACC 对比

Fig 4.5 ACC comparison of DAFW model and benchmark model

图 4.6 显示了加入分析特征相比于未加入分析特征的推荐方法，在推荐的多样性上有明显的提升。而基于时间序列的 Sina 方法在 D@5 上也较本文提出的方法低，

原因是好友中存在一些活跃的公众号，连续发布一些微博，从而导致在短区间里该值低。而基于 RankNet 方法的推荐列表，对于排名前 10 的微博多样性稍差。

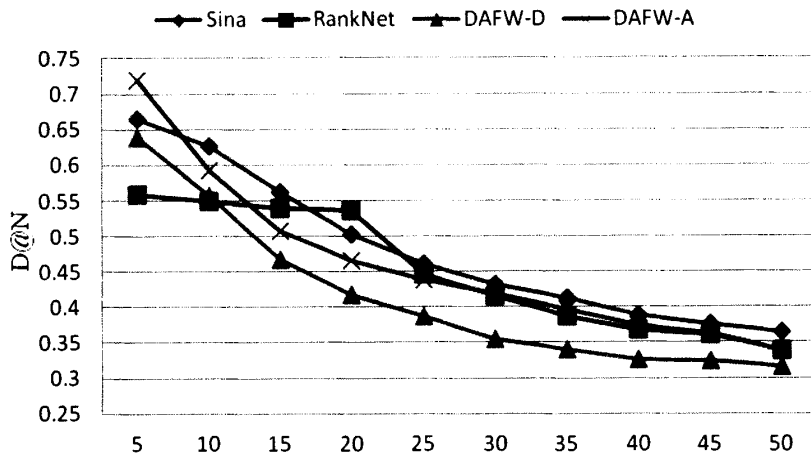


图 4.6 动态自适应权重模型与基准模型的 D@N 对比

Fig 4.6 D@N comparison of DAFW model and benchmark model

在实际效果上，如表 4.2 所示，用户@何小台 RMadrid 在第 420 会话期间关注的好友产生了 777 条微博，我们利用模型对微博进行了重排序。基于对该用户的建模，我们发现用户对体育足球主题兴趣较大，因此可以看到推荐的前五条微博中有很多都是关于体育足球的内容，并且微博质量相对较高，有大量的社交信息，而排名第一的微博主要是用户与发布者@五朵 a 的交互 TF-IDF 指标得分异常高，其他会话中只要有@五朵 a 的微博就能明显往前排，进入@五朵 a 主页发现其发布频率很低，但是@何小台 RMadrid 对其发布的内容经常评论、点赞、转发等，这两方面促成了交互 TF-IDF 指标得分高，进一步我们惊喜发现原来二者是亲人关系。

用户@NvL-TLF 在第 376 会话期间关注的好友只产生了 21 条微博，待推荐的微博数量过少，基于用户以前转发记录，本文获得转发过的非关注好友的微博，合成了 156 条微博，从而推荐潜在感兴趣微博，其中第一条和第三条来自转发的好友，第一条基于微博热度比较高，且微博质量较高，第三条是来自 YouTube 精选，可靠、内容丰富，基于用户对计算机技术方面的兴趣，其他几条都推荐了这方面内容并且微博质量也很高，含有标签、@博主、附近链接等信息。

表 4.2 基于 DAFW 方法推荐的实际效果

Tab 4.2 The practical effect of recommendation based on DAFW method

基本信息	发布者	微博来源	排名前五的微博
用户： 何小台 RMadrid  会话编号： 420  微博数量： 777/777	五朵 a	关注 好友	还有 60 没去@美食和旅行:有机会,我也要跑遍这 186 块钱,你想和谁一起去?
	广州边锋谢俊辉	关注 好友	心情很糟 //@枪迷平原君@别别的兵工厂:→_→...纳九七:2011 年阿森纳 3:4 巴萨,2012 年阿森纳 3:4 米兰,2013 年阿森纳 3:3 拜仁,心情好些没?
	梦幻超级佳	关注 好友	严重推荐#虎扑篮球#强图《科比失散在中国的兄弟》(分享自 @虎扑体育)<网页链接>
	广州恒大淘宝足球俱乐部	关注 好友	3 月 18 日 20:00, 亚冠小组赛第三轮, 中国广州恒大淘宝主场迎战日本鹿岛鹿角, 这一战#鹿死谁手#?!<网页链接>广州恒大淘宝足球俱乐部
	南大周志华	关注 好友	//@愚形妙手:适合于各种大规模网络的向量表示:有向无向, weighted or binary。欢迎大家试用。@余凯_西二旗民工@唐杰 THU...<网页链接>...
用户： NvL-TLF  会话编号： 376  微博数量： 21/156	统计之都	转发 记录	【COS 招聘实习】微博平台及大数据招实习生啦,感兴趣的速速看过来,致所有有需要的人们。网页链接<招微博平台及大数据部实习生>
	ICTCLAS 张华平博士	关注 好友	【SMP2014】第三届全国社会媒体处理大会最新版本的会议手册, ...下载地址: <网页链接>透露一个抢书秘笈: Panel 环节提问的前五位朋友将获赠作者签名的《大数据搜索与挖掘》一本//@武卫东@刘挺@唐杰
	YouTube 精选	转发 记录	OK GO 这个乐队的特点就是, 嗯, 怎么说呢, MV, 非常, 非常, 奇怪[doge]。... (youtube 两天播放量超 400 万) ;Let You Down 查看图片...
	叶阳爱吃鱼	关注 好友	编程到底难在哪里? 是的, 有的时候吃个饭回来刚写的代码就忘了什么意思 //@悦诗风眠: @wangleineo: 程序员在编写代码时, 要在头脑中构建...
	数据堂	关注 好友	【大数据资讯】大数据的未来在物联网, 百度、阿里、腾讯谁第一个尝到甜食——说起大数据, 可能很多人都知道这是未来互联网时代发...<网页链接>

4.3 原型系统

上述实验验证了推荐方法的有效性, 为了将方法应用到实时的个性化推荐中, 本文设计了个性化微博推荐的原型系统。该系统能够实现对信息实时采集、分析、

重排序，并将重排序的结果展示给用户。系统的整体结构如图 4.7 所示，主要由数据采集层、数据处理层、数据展现层 3 个模块组成，下面详细介绍数据采集模和数据处理两个核心模块。

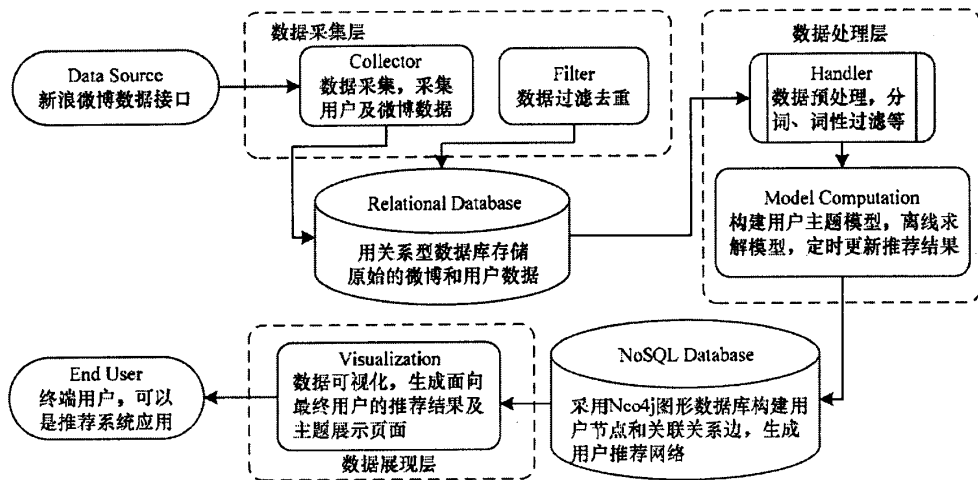


Fig 4.7 Real-time recommendation system architecture

### 4.3.1 数据采集层

本文提出的推荐方法需要用到大量的用户微博及用户的个人信息，由于新浪授权的 API 对部分数据的权限是封闭的，本文开发了通过模拟用户登录来采集相关数据的爬虫系统。开发环境是基于 MyEclipse, Redis, Tomcat, MySQL 等工具搭建起来的。本文的爬虫系统获取用户 id 后，能够实现对用户个人的基本信息、用户的历史微博及用户关注的好友列表等信息的采集。图 4.8 为数据抓取的架构图，我们取出存放在数据库中的用户账号及密码，通过 httpclient 请求验证登录新浪微博，系统收到提交的用户账号或者 url 时会模拟访问者抓取用户微博、用户基本信息、用户关注列表等网页数据。接着我们利用 JSoup 包解析网页数据，将解析过的数据存入 MySQL 数据库中，用于后续研究。

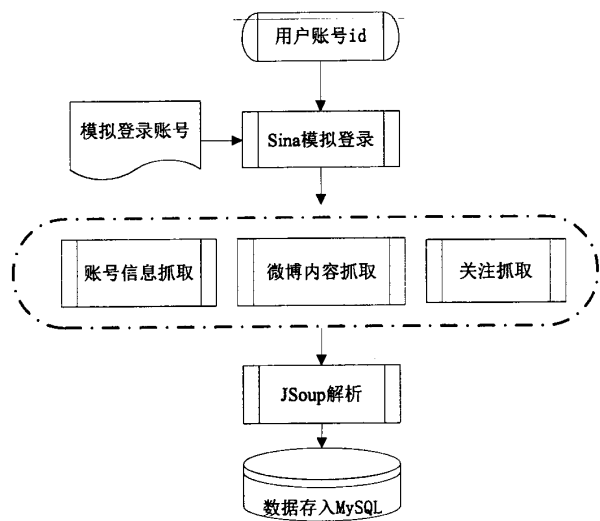


图 4.8 数据抓取的架构图

Fig 4.8 Architecture of data crawler

为了提高可操作性，本文对爬虫程序进行了简单封装，可以通过浏览器提交抓取任务，如图 4.10 所示，点击“开启抓取”后系统后台会实现验证登录。在验证成功后，我们可以点击“添加抓取任务”提交抓取需求，如图 4.11 所示，将待抓取的用户账号或者 url 放入任务框中，我们提交不同类型的任务后系统会抓取相应的数据。

# 系统操作

开启抓取系统

暂停抓取系统

添加抓取任务

图 4.9 爬虫程序主页

Fig 4.9 Homepage of crawler

# 添加任务

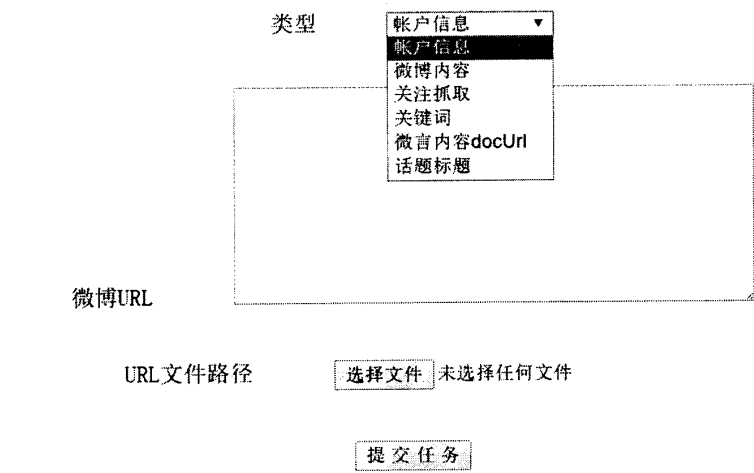


图 4.10 系统提交抓取任务页面

Fig 4.10 Page of system submit task

下图为开发的爬虫程序运行过程中控制台打印的信息，可以基于此监控爬虫运行状态。

```
2017-03-06 13:26:20,222 [Timer-1] INFO [com.weibo.common.tasks.CircleTimerTask] - 开始执行周期任务!
2017-03-06 13:26:20,222 [Timer-1] INFO [com.weibo.common.tasks.CircleTimerTask] - 现在的circleUrls size-----0
2017-03-06 13:26:20,222 [Timer-1] INFO [com.weibo.common.tasks.CircleTimerTask] - 现在的circle keywords size-----0
2017-03-06 13:26:24,824 [守护线程] INFO [com.weibo.common.threads.DaemonThread] - 守护线程运行中
2017-03-06 13:26:34,824 [守护线程] INFO [com.weibo.common.threads.DaemonThread] - 守护线程运行中
2017-03-06 13:26:44,825 [守护线程] INFO [com.weibo.common.threads.DaemonThread] - 守护线程运行中
2017-03-06 13:26:53,850 [http-8090-4] INFO [org.apache.struts2.dispatcher.Dispatcher] - Unable to find 'struts.multipart.
txtFile--null
2017-03-06 13:26:54,825 [守护线程] INFO [com.weibo.common.threads.DaemonThread] - 守护线程运行中
2017-03-06 13:26:55,842 [cookie抓取--2] INFO [com.weibo.sina.cookie.person.GrabContentInfo] - 个人版---正在抓取http://weib
Hibernate: insert into doc (article, author, discuss, doc article id, docUrl, extAt, forwardDocUrl, grab_method, insertTim
person doc save successful---DocInfo [article=<a title="微博会员特权" href="http://vip.weibo.com/privdesc?priv=11074from=
Hibernate: insert into doc (article, author, discuss, doc article id, docUrl, extAt, forwardDocUrl, grab_method, insertTim
person doc save successful---DocInfo [article=Google人工智能学习语言的新方式: 多看情色小说<a target="blank" render="ext" :
Hibernate: insert into doc (article, author, discuss, doc article id, docUrl, extAt, forwardDocUrl, grab_method, insertTim
person doc save successful---DocInfo [article=使用Klearn优雅地进行数据挖掘-一起大数据<a target="blank" render="ext" extra
Hibernate: insert into doc (article, author, discuss, doc article id, docUrl, extAt, forwardDocUrl, grab_method, insertTim
person doc save successful---DocInfo [article=转发微博, author=一起大数据, discuss=1, docArticleId=4081604609977929, docUrl
Hibernate: insert into doc (article, author, discuss, doc article id, docUrl, extAt, forwardDocUrl, grab_method, insertTim
person doc save successful---DocInfo [article=转发微博, author=一起大数据, discuss=1, docArticleId=4081604538234741, docUrl
Hibernate: insert into doc (article, author, discuss, doc article id, docUrl, extAt, forwardDocUrl, grab_method, insertTim
person doc save successful---DocInfo [article=PG DBA at 探探 / 数据分析招聘<a target="blank" render="ext" extra-data="type
Hibernate: insert into doc (article, author, discuss, doc article id, docUrl, extAt, forwardDocUrl, grab_method, insertTim
person doc save successful---DocInfo [article=转发微博, author=一起大数据, discuss=0, docArticleId=4081233816843031, docUrl
Hibernate: insert into doc (article, author, discuss, doc article id, docUrl, extAt, forwardDocUrl, grab_method, insertTim
person doc save successful---DocInfo [article=转发微博, author=一起大数据, discuss=0, docArticleId=408123380965262, docUrl
```

图 4.11 抓取过程中控制台信息

Fig 4.11 Console information during the crawl

抓取的微博数据写入事先设计好的数据库表中，文本根据研究需要设计了一些关键表，如（1）基本信息表：sina\_person；（2）微博数据表：doc；（3）关注的好友列表：person\_attention；（4）主题分布表：topic\_distribution；（5）特征分布表：feature；表的主要结构如图 4.12 所示。

```

    graph LR
      SP[sina_person] -- doc_id --> D[doc]
      D -- doc_id --> F[feature]
      PA[person_attention] -- doc_id --> D
      PA -- feature_id --> F
      SP -- person_id --> PA
  
```

**sina\_person**

- id: bigint
- grab\_method: varchar(20)
- uid: varchar(20)
- url: varchar(255)
- name: varchar(64)
- sex: varchar(128)
- brithday: varchar(16)
- address: varchar(255)
- fansNum: int UNSIGNED
- summary: varchar(1024)
- wbNum: int UNSIGNED
- gzNum: int UNSIGNED
- weiboType: varchar(10)
- blogUrl: varchar(255)
- realName: varchar(100)
- email: varchar(64)
- qq: varchar(50)
- msn: varchar(64)
- edu: varchar(2047)
- work: varchar(2047)
- renZh: int
- updateTime: datetime
- uid\_long: bigint
- tags: varchar(200)
- verifyInfo: varchar(1000)
- accountType: varchar(100)

**doc**

- id: bigint
- grab\_method: varchar(20)
- sendUrl: varchar(100)
- docUrl: varchar(100)
- article: varchar(2000)
- person\_id: varchar(30)
- publishtime: datetime
- origin: varchar(30)
- discuss: varchar(1000)
- insertTime: datetime
- author: varchar(30)
- weibo\_type: int
- transmit: int
- doc\_article\_id: varchar(50)
- sina\_doc\_param: varchar(100)
- person\_id\_long: bigint
- isForward: int
- forwardDocUrl: varchar(50)
- extAt: varchar(255)
- likes: int
- verify\_status: int

**feature**

- id\_score: double
- ren\_zhen\_score: double
- trans\_score: double
- at\_score: double
- docId\_score: double
- hashtag\_score: double
- url\_score: double
- lengh\_score: double
- influence\_score: double
- tf\_idf\_score: double
- id: bigint
- doc\_url: varchar(100)
- article: varchar(1000)
- person\_id: varchar(30)
- publish\_time: datetime
- insert\_time: datetime
- author: varchar(30)
- weibo\_type: int
- doc\_article\_id: varchar(50)
- sina\_doc\_param: varchar(100)
- is\_forward: int
- author\_url: varchar(50)
- old\_article: varchar(2000)

**person\_attention**

- id: bigint
- uid: varchar(20)
- attentionUrl: varchar(100)
- attentionIdList: longtext
- attentionSize: int
- attentionTime: datetime

**topic\_distribution**

- person\_id: varchar(20)
- strenght: double
- model: varchar(30)
- topic0: double
- topic1: double
- topic2: double
- topic3: double
- topic4: double
- topic5: double
- topic6: double
- topic7: double
- topic8: double
- topic9: double

图 4.12 数据表设计

**Fig 4.12** Design of data table

### 4.3.2 数据处理层

本文通过数据的抓取获得了大量用户真实数据，结合本文的个性化推荐算法，我们使用 JAVA 语言，基于 MVC+Mybatis+Maven 的框架设计并实现了对实时微博数据的重排序及微博数据的处理。数据处理的功能流程如图 4.13 所示，其中语料预处理功能模块 TextPreprocess 和通用主题训练功能模块 LDATrain 是程序的核心功能点，对推荐性能有直接影响；LDAPredictor 模块则通过通用主题模型对短文本进行主题抽取，得到短文本的主题分布；FeattrueCal 模块实现了对推荐微博的特征量化，输出为每条微博在各个特征上的打分，其中离线特征是事先计算得到的；FeatureCom 模块通过 DAFW 方法计算得到特征的动态权重，从而计算得到每条微博的综合得分，然后根据综合得分对微博进行重排序并将排序结果作为输出封装到模型中。



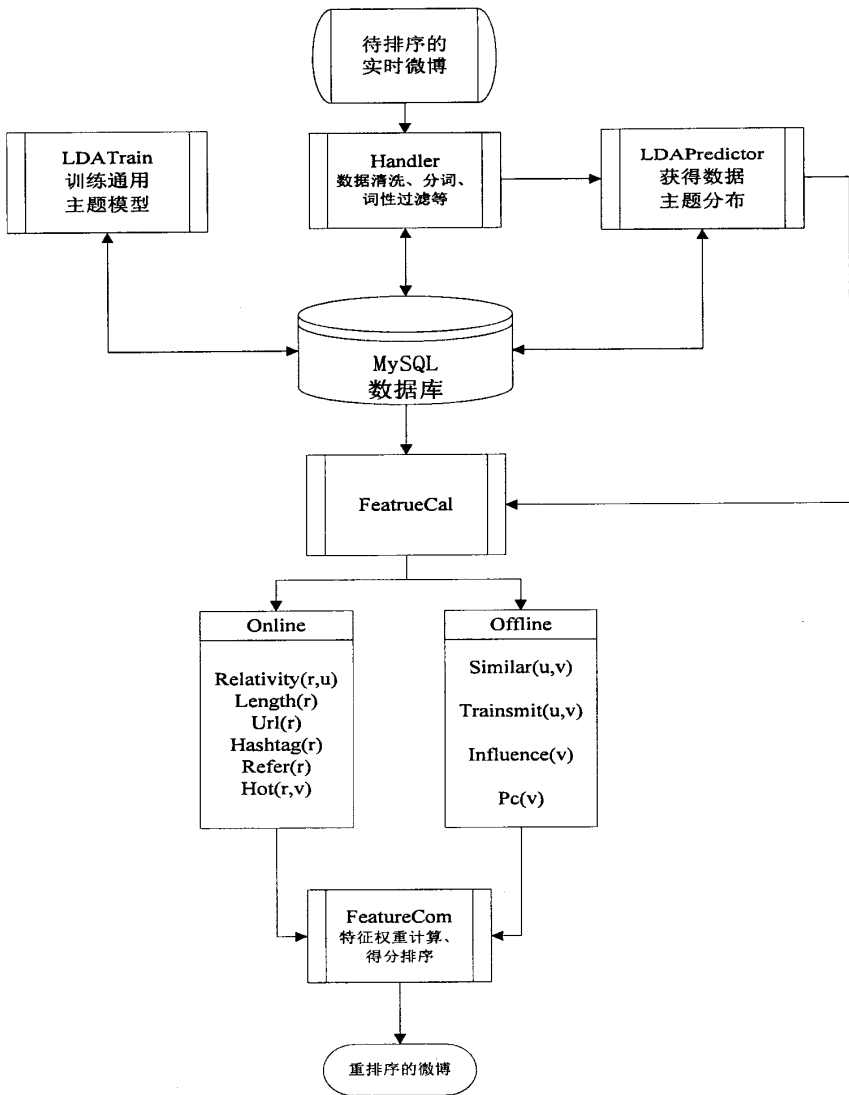


图 4.13 实时数据处理的流程图

Fig 4.13 The flowchart of real-time data processing

图 4.14 展示了未处理的微博语料，其中含有大量噪音，直接基于这样的数据研究很难获得有用结果。图 4.15 展示了利用本文开发的 Handler 模块处理后的效果。可以看出处理后的数据主要由一些实体词组成，该模块能够为后续通用主题训练及实时微博分析提供了高质量的数据。






```
勿忘初心。只是好伤心。 <b>lda.params</b>  | 2016-12-16 0:02  | PARAMS 文件  | 1 KB       |
|  <b>lda.phi</b>     | 2016-12-16 0:04  | PHI 文件     | 413,192 KB |
|  <b>lda.tassign</b> | 2016-12-16 0:04  | TASSIGN 文件 | 241,565 KB |
|  <b>lda.theta</b>   | 2016-12-16 0:04  | THETA 文件   | 5,615 KB   |
|  <b>lda.twords</b>  | 2016-12-16 0:04  | TWORDS 文件  | 23 KB      |
|  <b>wordmap.txt</b> | 2016-12-15 20:39 | 文本文档       | 8,048 KB   |

图 4.16 LDA 训练的文档集

Fig 4.16 The document set of LDA training

图 4.17 为训练出的主题-单词排序文档  $\text{lda.twords}$  的部分内容, 通过该图我们能够看出通用主题模型训练到达了预期效果, 即主题下词实现了聚类而主题与主题之间有明显的语义区别。

|            |                          |                          |                          |                            |
|------------|--------------------------|--------------------------|--------------------------|----------------------------|
| topic 3 :  | 阅读 0.01187950589604378   | 作者 0.008560267440974712  | 故事 0.008242002688946293  | 作品 0.0060738553293049335   |
| topic 4 :  | 汽车 0.019020646810531616  | 宝马 0.013091503642499447  | 车型 0.007132763508707285  | 车主 0.0064224498346447945   |
| topic 5 :  | 市场 0.010858028195798397  | 证监会 0.009074696399666843 | 投资 0.008024144917726517  | 金融 0.006874614729307643    |
| topic 6 :  | 动作 0.04861002415418625   | 运动 0.01580145582567245   | 健身 0.013864264619704985  | 训练 0.011712626987893799    |
| topic 7 :  | 旅行 0.01916636712284914   | 杭州 0.021244426344670911  | 旅游 0.009229680513276329  | 攻略 0.007661943789571524    |
| topic 8 :  | 搭配 0.01752799180950085   | 时尚 0.01061641238629818   | 系列 0.010109621100227697  | 单品 0.006832466542720795    |
| topic 9 :  | 中国 0.022346816956996918  | 美国 0.008996455655463656  | 习近平 0.005553813525289297 | 国家 0.005340928211809205    |
| topic 10 : | 招牌 0.052790723741054535  | 实习 0.0232906546437008    | 工作 0.0220007624924183    | 校园 0.0214634258300066      |
| topic 11 : | 电影 0.032135673451278597  | 故事 0.0073388010645346165 | 高清 0.006648669019341469  | 字幕 0.006041018757969141    |
| topic 12 : | 比赛 0.016767892986536026  | 球迷 0.009207378141582012  | 球员 0.00662796579901357   | 足球 0.008582325652241707    |
| topic 13 : | 数据 0.013883874362111092  | 互联网 0.012494663707912368 | 公司 0.010921105742454529  | 产品 0.010752067901194096    |
| topic 14 : | 作品 0.009458797052502632  | 摄影 0.007839910244313221  | 设计 0.007333309395650822  | 摄影师 0.00694832229492711    |
| topic 15 : | 手机 0.03205680547167969   | 小米 0.012981314540898273  | 苹果 0.0100345494362097645 | iphone 0.00973375787851509 |
| topic 16 : | 生活 0.014634227380116517  | 人生 0.02248851270973682   | 时间 0.009385976009070873  | 世界 0.009168971329927444    |
| topic 17 : | 考研 0.029337190091609555  | 雅思 0.024028176441786673  | 英语 0.019474871456623077  | 口语 0.01721409521996975     |
| topic 18 : | 音乐 0.016350915465493736  | 歌手 0.008184028787478707  | mv 0.0065167006105184555 | 节目 0.0059536295011639595   |
| topic 19 : | 韩国 0.03229605406522751   | 太阳 0.006945021450519562  | 首尔 0.006703411694616079  | 免费 0.006602849023491144    |
| topic 20 : | 东西 0.0047463891096412136 | 朋友 0.00463756313547492   | 好像 0.00415112404152751   | 手机 0.003283575875684619    |
| topic 21 : | 日本 0.018144585660910606  | 加多宝 0.012312324717640677 | 动物 0.01102634280308628   | 台湾 0.008858513087034225    |
| topic 22 : | 研究 0.01081613693118095   | 教授 0.008677029982209206  | 中国 0.00743155739323047   | 学生 0.007144850213089645    |
| topic 23 : | 宝宝 0.018997149541874068  | 孩子 0.01049575343132018   | 妈咪 0.009721679612994194  | 医生 0.0072676013223826885   |
| topic 24 : | vs 0.01960044354200363   | 比赛 0.01591330559423714   | 女排 0.01479502022146388   | 决赛 0.01100427544239415     |
| topic 25 : | 江西 0.00971829324112892   | 双盘 0.0148287583142519    | 双鱼座 0.011670123390578952 | 运势 0.011569911614040402    |
| topic 26 : | 美食 0.025771356097276688  | 赣州 0.00595066758823395   | 密团 0.008271585412605762  | 工作 0.0080461839174509      |
| topic 27 : | 设计 0.05269630425844002   | 创业 0.013782232579827309  | 美味 0.0109235094616019611 | 吃货 0.00989024203751564     |
| topic 28 : | app 0.015981536358594694 | 白蜡树 0.025779519812965593 | 建筑 0.0196777824324485    | 新歌 0.01667504757624761     |
| topic 29 : | 小伙伴 0.012623802092075335 | 开发者 0.013538874695764313 | 技术 0.011106903664767742  | 下载 0.009795242920517921    |
| topic 30 : | 考试 0.026287570598741272  | 机会 0.010771911591291428  | 福利 0.007612589406967163  | 参与 0.00776309058915241     |
| topic 31 : |                          | 公务员 0.022292301058769226 | 报名 0.016033307045698166  | 面试 0.01637214794754982     |

图 4.17 主题-单词排序文档

Fig 4.17 Topic-words document

本文通过上述各个模块的开发实现了面向用户的实时个性化推荐系统, 图 4.18 展示了系统推荐的效果, 结合后续数据表现层功能的开发完善, 系统能够实时为用户推送朋友圈中有特色的信息。

```
76 2495587332 第420次会话关注的好友推送了36条微博
77 2495587332 向用户推荐20条微博
78 2495587332 从转发过的非好友的微博中获得74条微博
79 2495587332 待排序的微博数量112条微博
80 2495587332 动态特征权重参数{"atStrength":0.018922754701933622,"dockStrength":0.3334692043573748,"hashtagStrength":0.0011459153622011694}
81
82 author: 数据化管理 url: http://www.weibo.com/1424710394/Dqd1O6qBT
83 gzDoc: 郭跟 纸牌屋 竞选 新闻 特朗普 老婆 老公 英雄救美 克蕾兹 傅志 没想到 老底 彻底 现在 新闻网 全面 跟进 真是 机
84 author: 数据化管理 Score:10.491606790677519 Refer:1.0 Relativity:0.0011459153622011694 Hashtag:0.0 Url:0.0 Length:
85 RenZhen:1.0 Transmit:5.0 Heat:0.02 Influence:0.9991195436725651 Similar:1.2654385929348567
86
87 author: 前端大全 url: http://www.weibo.com/5261893810/Dc8wEhBAs
88 zfDoc: 浅析 正则表达式 模式匹配 方法 代码 使用 正则表达式 进行 模式匹配 经常 用到 对象 对象 方法 方法 以下 方法 4
89 author: 前端大全 Score:4.96148278234538 Refer:1.0 Relativity:0.007331574044402119 Hashtag:0.0 Url:2.0 Length:72.0
90 RenZhen:0.0 Transmit:0.0 Heat:0.00948717 Influence:0.9996794871794872 Similar:0.2653127550639308
91
92 author: 视界2016 url: http://www.weibo.com/5821486185/DcaLJhGV
93 zfDoc: 国外 网友 狗狗 洗澡 生气 乱窜 有心 主人 家下 配上 对话 狗狗 喜感 视频
94 author: 视界2016 Score:1.8881379969380705 Refer:0.0 Relativity:0.003102270322233709 Hashtag:0.0 Url:1.0 Length:28.
95 RenZhen:0.0 Transmit:0.0 Heat:0.001123 Influence:0.9825388981236054 Similar:0.12607575556791958
96
97 author: 好东西传进门 url: http://www.weibo.com/5220650532/Dc8XJpUeH
98 gzDoc: 开发 日报 网页 链接 源代码 解析 带有 爆炸 动画 效果 菜单 安卓 应用 反编译 常用工具 使用 方法 统一 请求 设4
99 author: 好东西传进门 Score:4.209874448100869 Refer:0.0 Relativity:0.004503101120441468 Hashtag:0.0 Url:2.0 Length:
100 RenZhen:0.0 Transmit:0.0 Heat:0.3823 Influence:0.9737782353297402 Similar:0.24635271283934804
101
102 author: 锤子科技 url: http://www.weibo.com/2968634427/Dc6Z0Dq4H
103 gzDoc: 今天 西方 传统 节日 复活节 锤子 科技 官网 商城 首页 商品价格 彩蛋 彩蛋 彩蛋 可能 蕴含着 惊喜 降临 时刻 降4
104 author: 锤子科技 Score:3.62142624706935 Refer:0.0 Relativity:0.0016248844662461845 Hashtag:0.0 Url:0.0 Length:58.0
105 RenZhen:0.0 Transmit:0.0 Heat:0.105407 Influence:0.9995602999292891 Similar:0.1479205407785201
106
```

图 4.18 系统推荐效果

Fig 4.18 The effect of system recommended

4.4 本章小结

本章首先介绍了实验数据的获取和预处理。随后，我们引入了四种全面度量模型有效性的指标，即 P@N、MAP、HIR、D@N。围绕四个指标确定要对比的 3 个基准模型，即 Sina、RankNet、DAFW。接着通过实验结果分析了模型的有效性，并以实际结果展示了模型的推荐效果。最后围绕数据采集层、数据处理层、数据展现层对实时个性化推荐做了一个系统设计和部分实现工作，以期研究结果能够在实际生活中创造价值。

## 第五章 总结与展望

随着社交网络的飞速发展, 用户获取信息变得越来越便捷, 但随着活跃用户的增加, 每天好友推送的信息量将变得异常丰富, 微博就是典型的代表。微博的用户会收到关注的各种信息源推送的丰富信息, 例如一些名人、亲朋好友、政务微博、新闻媒体、商业群体等不同账号发布的大量信息。这些信息按照时间先后融合在一起, 形成庞大的信息流, 而随着用户的关注列表及活跃用户数的增加, 相继产生了信息过载现象。对于精力有限的用户来说, 希望能用在很短的时间里获取有价值的信息; 而对于频繁使用微博的用户来说, 关注的用户发布的微博可能不能够满足其需求, 他们需要对其推送用户潜在感兴趣的微博信息。解决上述两种问题, 为用户提供个性化推荐服务, 成了本文研究的主要内容。

### 5.1 工作总结

本文分析了微博推荐的国内外研究现状, 在充分利用现有的方法基础上, 结合实际应用场景, 提出了基于动态自适应权重的个性化微博推荐系统。本文通过用户评价及真实微博数据的实验, 验证推荐系统的有效性。本文的工作成果总结如下:

(1) 为了构建微博语料集, 本文建立了一套完备的微博语料处理方案。该方案能够对内容元素丰富的微博实现层层过滤, 在去除噪音去的同时保留了重要的文本内容。并且本文经过多次实验整理出针对微博的停用词库, 提高了过滤后的数据质量。

(2) 针对短文本主题建模难度大的问题, 本文构建了通用主题模型, 即基于 LDA 方法使用大量微博语料训练出通用主题模型。通过训练结果的对比, 本文训练出 40 个主题维度, 再采用 Gibbs Sampling 方法从通用主题模型中抽取出短文本的主题分布, 本文训练出的优质通用主题模型可以为相关研究工作提供辅助。

(3) 由于用户的登陆和注销时间难以获取, 为了能够明确推荐范围, 本文提出了基于用户行为的用户会话划分方法。基于该工作不仅明确了下一次需要待推荐的微博, 而且可以将用户历史转发的微博及其所在的会话片段获取到。评估推荐算法时, 可以基于推荐算法将用户转发的微博往前排的位置来度量, 从而为评估推荐效果提供了一种方法。

(4) 每条微博背后都存在社交属性, 为了全面的量化微博的各方面特征, 本文首先对现有的特征量化方法做了分析总结, 然后从用户偏好、微博内容、发布者状态三种不同视角构建了十个微博特征, 其中融入了很多分析特征, 如微博热度、交互 TF-IDF 等, 实验表明, 这些特征的加入能够明显的提高推荐质量。

(5)为解决多指标融合问题,本文将信息熵理论引入到权重调整环节。首先根据每一个微博特征指标变异性大小来确定客观权重,并结合均值及参数得到每个特征的动态权重,能够根据推荐内容的不同,自动调整权重参数。实验表明该方法能有效融合多方面特征。

(6)研究并设计了实时推荐系统。首先该系统能够通过数据采集层采集用户数据,然后利用数据处理层对待推荐微博进行实时特征提取和重排序工作,最后通过数据展现层实现数据展示和交互工作,目前结合设计已开发了一些功能接口,为个性化推荐系统应用打下基础。

## 5.2 展望

本文为了给用户提供实时的个性化推荐服务,解决普遍存在信息过载问题,在前人研究基础上改进并构建了一批微博特征,并提出 DAFW 方法融合多个特征,实现微博重排序,将有价值的微博推荐给用户,虽然实验验证了方法的有效性,但仍存在很多有待改进的空间,需要在后续研究中深入探讨,具体有:

(1)在计算用户主题偏好时,本文通过用户近两个月所发的历史微博提取用户的主题分布,然而每个用户的兴趣可能随着时间变化的速度不一致,对于个别活跃的用户而言,可能存在度量的兴趣与当前兴趣不一致。如何取用户历史微博使其计算得到的用户兴趣分布与用户当前真实兴趣分布误差尽可能小,这将是未来的一个研究方向。

(2)在划分用户会话时,本文是通过用户登录后转发、发布微博行为来计算开始时间的。但是在实际情况中,用户在某次登录后存在只浏览不转发和发布微博的行为,可能导致多个会话的合并,那么待推荐的微博就包含了用户浏览过的微博,其价值就降低了。因此,下一步工作中需要对会话划分方法做进一步改进,本文可以考虑开发使用移动端的微博推荐 APP,方便用户登陆数据获取,从而明确推荐范围。

(3)在特征量化中,本文提出了一些分析特征,实验证明了其对提高推荐质量有明显作用。因此在接下来研究中将围绕微博挖掘更多分析特征,并分析其在推荐中发挥的作用。

(4)本文主要基于通用主题模型抽取出短文本的主题分布,由于微博中新词、热词层出不穷,基于通用模型无法计算未出现在词的主题分布,因此下一步工作中,本文需要研究如何基于通用主题模型计算未出现的新词的主题分布,并迭代更新通用主题模型。

## 参考文献

- [1] 中国互联网信息中心. 第39次中国互联网发展状况统计报告[R]. 北京: 中国互联网信息中心, 2017.
- [2] Gupta Y, Saini A, Saxena A K. A new fuzzy logic based ranking function for efficient Information Retrieval system[J]. Expert Systems with Applications, 2015, 42(3):1223-1234.
- [3] Xu H, Zhang R, Lin C, Gan W. Construction of E-commerce recommendation system based on semantic annotation of ontology and user preference[J]. Indonesian Journal of Electrical Engineering and Computer Science, 2014, 12(3): 2028-2035.
- [4] Gavalas D, Kenteris M. A web-based pervasive recommendation system for mobile tourist guides[J]. Personal and Ubiquitous Computing, 2011, 15(7):759-770.
- [5] Colombo-Mendoza L O, Valencia-García R, Rodríguez-González A, Samper-Zapater JJ. RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes[J]. Expert Systems with Applications, 2015, 42(3):1202-1222.
- [6] Popescul A, Ungar L H, Pennock D M, Lawrence S. Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments[J]. Eprint Arxiv, 2013:437-444.
- [7] Arora G, Kumar A, Devre G S, et al. MOVIE RECOMMENDATION SYSTEM BASED ON USERS'SIMILARITY[J]. International Journal of Computer Science and Mobile Computing, 2014, 3(4): 765-770.
- [8] Ekstrand M D, Riedl J T, Konstan J A. Collaborative filtering recommender systems[J]. Foundations and Trends in Human-Computer Interaction, 2011, 4(2): 81-173.
- [9] Linden G, Smith B, York J. Amazon. com recommendations: Item-to-item collaborative filtering[J]. IEEE Internet computing, 2003, 7(1): 76-80.
- [10] Cai Y, Leung H, Li Q, Min H, Tang J, Li H. Typicality-Based Collaborative Filtering Recommendation[J]. IEEE Transactions on Knowledge & Data Engineering, 2014, 26(3):766-779.
- [11] Burke R. Hybrid recommender systems: Survey and experiments[J]. User modeling and user-adapted interaction, 2002, 12(4): 331-370.
- [12] Kagita V R, Pujari A K, Padmanabhan V. Virtual user approach for group recommender systems using precedence relations[J]. Information Sciences, 2015, 294(294):15-30.
- [13] Chen W, Niu Z, Zhao X, Li Y. A hybrid recommendation algorithm adapted in e-learning environments[J]. World Wide Web, 2014, 17(2):271-284.

- [14] Li H. Learning to Rank for Information Retrieval and Natural Language Processing[J]. Synthesis Lectures on Human Language Technologies, 2014, 4(1):113.
- [15] Java A, Song X, Finin T, Tseng B. Why we twitter: understanding microblogging usage and communities[C]//Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis. ACM, 2007: 56-65.
- [16] Naaman M, Boase J, Lai C H. Is it really about me?: message content in social awareness streams[C]// ACM Conference on Computer Supported Cooperative Work. ACM, 2010:189-192.
- [17] Chen J, Nairn R, Chi E. Speak little and well: recommending conversations in online social streams[C]//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2011: 217-226.
- [18] Jiang J, Chen P, Wang X, Dai Y. Who drive people to forward information: publisher or spreader?[C]//Proceedings of the Fifth Workshop on Social Network Systems. ACM, 2012: 11.
- [19] Uysal I, Croft W B. User oriented tweet ranking: a filtering approach to microblogs[C]//Proceedings of the 20th ACM international conference on Information and knowledge management. ACM, 2011: 2261-2264.
- [20] Paek T, Gamon M, Counts S, Chickering D M, Dhesi A. Predicting the Importance of Newsfeed Posts and Social Network Friends[C]//AAAI. 2010, 10: 1419-1424.
- [21] Zhao L, Zeng Y, Zhong N. A weighted multi-factor algorithm for microblog search[C]//International Conference on Active Media Technology. Springer Berlin Heidelberg, 2011: 153-161.
- [22] Cha M, Haddadi H, Benevenuto F, et al. Measuring user influence in twitter: The million follower fallacy[J]. Icwsm, 2010, 10(10-17): 30.
- [23] Taiki M, Kazuhiro S, Kuniaki U. Improving Pseudo-relevance Feedback via Micro-document Selection[J]. Ipsj Journal, 2014, 55:1585-1594.
- [24] 卫冰洁, 王斌. 面向微博搜索的时间感知的混合语言模型[J]. 计算机学报, 2014, 37(1):229-237.
- [25] Harvey M, Crestani F, Carman M J. Building user profiles from topic models for personalised search[C]//Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, 2013: 2309-2314.
- [26] Mccallum A K. MALLET: A machine learning for language toolkit[J]. 2002.
- [27] Rubin V L, Liddy E D. Assessing Credibility of Weblogs[C]//AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs. 2006: 187-190.



- [28] Weerkamp W, De Rijke M. Credibility Improves Topical Blog Post Retrieval[C]//ACL. 2008, 8: 923-931.
- [29] Nagmoti R, Teredesai A, De Cock M. Ranking approaches for microblog search[C]//Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. IEEE, 2010, 1: 153-157.
- [30] Vosecky J, Leung K W T, Ng W. Searching for quality microblog posts: Filtering and ranking based on content analysis and implicit links[C]//International Conference on Database Systems for Advanced Applications. Springer Berlin Heidelberg, 2012: 397-413.
- [31] Wu W, Zhang B, Ostendorf M. Automatic generation of personalized annotation tags for twitter users[C]//Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics. Association for Computational Linguistics, 2010: 689-692.
- [32] Chen J, Nairn R, Nelson L, Bernstein M, Chi E. Short and tweet: experiments on recommending content from information streams[C]//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2010: 1185-1194.
- [33] 闫强, 吴联仁, 郑兰. 微博社区中用户行为特征及其机理研究[J]. 电子科技大学学报, 2013, 42(3): 328-333.
- [34] Weng J, Lim E P, Jiang J, He Q. Twitterrank: finding topic-sensitive influential twitterers[C]//Proceedings of the third ACM international conference on Web search and data mining. ACM, 2010: 261-270.
- [35] Hong L, Davison B D. Empirical study of topic modeling in twitter[C]//Proceedings of the first workshop on social media analytics. ACM, 2010: 80-88.
- [36] Ramage D, Dumais S T, Liebling D J. Characterizing microblogs with topic models[C]//ICWSM, 2010, 5(4): 130-137.
- [37] 高明, 金澈清, 钱卫宁, 等. 面向微博系统的实时个性化推荐[J]. 计算机学报, 2014, 37(4): 963-975.
- [38] Cui P, Wang F, Liu S, Ou M, Yang S, Sun L. Who should share what?: item-level social influence prediction for users and posts ranking[C]//Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. ACM, 2011: 185-194.
- [39] Hong L, Doumith A S, Davison B D. Co-factorization machines: modeling user interests and predicting individual decisions in twitter[C]//Proceedings of the sixth ACM international conference on Web search and data mining. ACM, 2013: 557-566.
- [40] Li H. Learning to rank for information retrieval and natural language processing[J]. Synthesis

Lectures on Human Language Technologies, 2014, 7(3): 1-121.

- [41] Song Y, Wang H, He X. Adapting deep ranknet for personalized search[C]//Proceedings of the 7th ACM international conference on Web search and data mining. ACM, 2014: 83-92.
- [42] Cao Y, Xu J, Liu T Y, Li H, Huang Y L, Hon H W. Adapting ranking SVM to document retrieval[C]//Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006: 186-193.
- [43] Cao H, Verma R, Nenkova A. Speaker-sensitive emotion recognition via ranking: Studies on acted and spontaneous speech[J]. Computer speech & language, 2015, 29(1): 186-202.
- [44] Freund Y, Iyer R, Schapire R E, Singer Y. An efficient boosting algorithm for combining preferences[J]. Journal of machine learning research, 2003, 4(Nov): 933-969.
- [45] Miao Z, Wang J, Zhou A, Tang K. Regularized boost for semi-supervised ranking[C]//Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1. Springer International Publishing, 2015: 643-651.
- [46] Zheng Z, Chen K, Sun G, Zha H. A regression framework for learning ranking functions using relative relevance judgments[C]//Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2007: 287-294.
- [47] 彭泽环, 孙 乐, 韩先培, 石贝. 基于排序学习的微博用户推荐[J]. 中文信息学报, 2013, 27(4): 96-103.
- [48] Duan Y, Jiang L, Qin T, Zhou M, Shum H Y. An empirical study on learning to rank of tweets[C]//Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, 2010: 295-303.
- [49] Blei D M. Probabilistic models of text and images[D]. University of California, Berkeley, 2004.
- [50] Shannon C E. A mathematical theory of communication [J]. Bell System Technical Journal, 1948, 27(1-2): 623-656
- [51] Xiang L. Recommender System Practice. Beijing: Posts and Telecom Press, 2012. 40-55.
- [52] Bordes C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, et al. Learning to rank using gradient descent[C]//Proceedings of the 22nd international conference on Machine learning. ACM, 2005: 89-96.
- [53] Shi Y, Karatzoglou A, Baltrunas L, Larson M, Hanjalic A, Oliver N. TFMAP: optimizing MAP for top-n context-aware recommendation[C]//Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2012: 155-164.

# 攻读硕士学位期间发表的论文

## 1) 参加科研项目

- (1) CCF-腾讯犀牛鸟基金“社会化推荐与算法研究”(编号: CCF-TencentRAGR20140109), 2014.10-2015.10
- (2) 面向大数据的商务分析与计算方法以及支撑平台(编号: 71490725), 国家自然科学基金重大项目课题, 2015.01-2019.12

## 2) 发表的学术论文

- [1] Jiang Y, Xu Y, Shao L. A Personalized Weibo Search Model Considering User - Publisher Relationship[C]// 2016 IEEE First International Conference on Data Science in Cyberspace, 2016.
- [2] Xu Y. Optimization of LDA Text Microblogging Recommendation Algorithm Based Learning To Rank[C]// International Conference on Advances in Mechanical Engineering and Industrial Informatics. 2016.

## 特别声明

本学位论文是在我的导师指导下独立完成的。在研究生学习期间，我的导师要求我坚决抵制学术不端行为。在此，我郑重声明，本论文无任何学术不端行为，如果被发现有任何学术不端行为，一切责任完全由本人承担。

学位论文作者签名：徐玉祥

签字日期：2017 年 04 月 10 日