

清华大学

综合论文训练

题目：面向社区媒体的用户分享行为预测

系 别：计算机科学与技术系

专 业：计算机科学与技术

姓 名：蒋 煜

指导教师：杨 士 强 教 授

2010 年 6 月 18 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名: 蒋缘 导师签名: 杨晓凡 日 期: 2010.6.29

中文摘要

随着社区媒体和社交网络的迅猛发展，用户的网络行为成为了信息传播、信息安全等诸多领域的重要研究对象。对社交网络中用户的行为进行建模和尽可能准确地预测，不仅有利于对信息的传播途径进行研究和预测，更是能够进一步对社交网络进行督导和管理。

在采用 2009 年 9 月~2009 年 12 月自行收集的人人网用户关系数据和分享信息的基础上，构建媒体内容的话题模型、用户的偏好模型和影响力模型，从而通过归纳学习的方法，融合上述模型计算用户对媒体信息的兴趣度，预测用户的行为，并与过去的预测方法进行比较，验证合理性和优越性，并提出媒体信息个性化推荐方法。

关键词： 用户行为；话题模型；偏好模型；影响力模型；信息传播

ABSTRACT

With the fast development of social media and networks, user behaviors on web become the important research objects in fields such as information spreading and security. It is not only the best way to research and predict the dissemination of information, but also that to supervise and manage social networks, if our system based on user behavior model can predict what will happen on social web as correctly as possible.

Based on data collected from September to December, 2009 that includes both relationships and sharing information of users on Renren, we build up topic model for media content, preference model and influence model for users. Then we combine all these models with inductive learning methods to calculate the preference on given media information for given user. This system is able to predict user behavior and comparable to the past methods of prediction. Based on our reasonable and effective system, we provide a recommendation method on media information for each user.

Keywords: user behavior;topic model;preference model;influence model;information spreading

目 录

第 1 章 引言	1
1.1 研究背景	1
1.2 相关研究工作	3
1.2.1 媒体主题挖掘	3
1.2.2 网络结构分析	4
1.2.3 网络节点影响力建模	4
1.2.4 用户兴趣建模	4
1.3 论文工作	5
1.3.1 工作目标	5
1.3.2 工作难点和重点	6
1.4 论文组织结构	6
第 2 章 数据获取与分析	8
2.1 数据获取准备	8
2.1.1 数据内容及获取对象	8
2.1.2 数据获取工具	8
2.1.3 数据解析工具	11
2.1.4 数据规模描述	12
2.2 数据的结构化	12
2.2.1 社区网络结构数据结构化	12
2.2.2 社区媒体数据结构化	13
2.3 数据调研	14
2.3.1 可预测的若干特征规律	14
2.3.2 特征规律的验证	15
第 3 章 话题挖掘与用户偏好分析	20
3.1 中文分词工具的简介	20
3.2 话题索引和词库的建立	20
3.3 标题内容语义聚类	22
3.4 话题模型的建立	27

3.5 社交网络偏好模型	28
3.5.1 常见的偏好模型	28
3.5.2 适用于社交网络的用户偏好模型	29
3.6 用户偏好检验话题模型	31
第 4 章 用户偏好与影响力融合的行为预测	36
4.1 网络结构图的建立及可视化	36
4.2 用户影响力的概述	39
4.3 用户影响力模型的建立	40
4.3.1 用户的偏好相似度	40
4.3.2 用户的交流频度	41
4.3.3 用户的网络结构紧密度	42
4.4 用户兴趣度模型	44
4.4.1 兴趣度相关的变量定义	44
4.4.2 兴趣度模型的建立	45
4.5 用户行为预测及验证	48
4.5.1 用户行为预测简介和意义	48
4.5.2 用户行为预测的验证方法	48
4.5.3 用户行为预测的验证结果和结论	51
4.6 媒体数据的个性化推荐	57
4.6.1 媒体数据个性化推荐的简介	57
4.6.2 媒体数据个性化推荐的方法	57
第 5 章 总结展望	59
5.1 工作总结	59
5.2 未来研究工作	59
插图索引	61
表格索引	62
参考文献	64
致 谢	66
声 明	67

附录 A 外文资料的调研阅读报告（或书面翻译）68

第1章 引言

1.1 研究背景

二十一世纪以来，网络技术和多媒体技术的进步，方便了人们更多地使用网络进行沟通交流，这带来的是社区媒体网络的迅猛发展、规模快速膨胀的现状。传统的媒体共享网站，如图片网站 Flickr，视频网站 YouTube，还有中文视频网站优酷、土豆等，提供了大量个人拍摄、编辑并上传的图片和视频，基于这些内容的分享、传播和浏览已成为互联网上最受用户欢迎的服务之一。需要注意的是，基于互联网的传播平台上的媒体信息和网络用户的联系非常紧密。不仅媒体内容本身大多由用户创造产生，而且用户还能够随意浏览、评价、转载、标注媒体内容等。因此，为了区别于传统的媒体形式，学者们将这种用户高度参与的媒体形式称为新媒体，也称为社区媒体。社区媒体已经成为当前互联网上最热门的应用之一。

由于 Web 2.0 技术的发展，各类社交网站，例如 Facebook、人人网等，吸引了数量庞大的用户群。在这些网站上，用户不仅能够和虚拟生活中的好友建立连接，形成规模巨大的社会网络；还可以在好友中传播、共享各种网络信息，加强社会交流。由于互联网技术的进步，分享的网络信息的表现形式逐渐呈多模态形式发展，信息内容的类型由文字为主逐步发展为文字、图片和视频共同主导的局面。如 Facebook 用户现在每月可分享数以亿计的照片；人人网上，用户不仅能共享上传的照片，还可以转载不同社区媒体网站上的有趣图片或视频。因此，社区媒体已经成为社交网络上信息的主题形式，社会网络与媒体内容的融合是当前社会网络发展的重要趋势。

社会网络与社区媒体的融合与渗透，形成了以互联网为依托、用户高度参与的、信息与知识快速传播的大规模社区媒体网络。社区媒体网络在经济、政治、文化生活中发挥着越来越重要的影响力。例如奥巴马是第一位借助互联网和新媒体的力量成功当选的“网络总统”，这是美国总统大选新的里程碑。美国大选期间，奥巴马利用信息技术，运用富有想象力的新媒体攻略，建立竞选门户网站和通过互动社交网站，“奥巴马答复中心”，以及搜索引擎的高调介入，让年轻人能

够与奥巴马在网上讨论政治话题，能够收看与竞选相关的视频，从而了解奥巴马的竞选政策，成功地树立了诚恳、谦卑的形象。

互联网和用户是社区媒体网络的基础；通过互联网和用户的广泛交互，产生了大量社区媒体与社会网络；而社区媒体内容与社会网络的融合与渗透，形成了社区媒体网络。因此，在用户行为的影响下，社区媒体网络中媒体内容在产生形式、传播方式以及影响的深度和广度方面都具有社会化的特点。社会网络中特有的用户高度参与性和互动性，导致了大量用户产生内容以及因此衍生出来的大量用户评价信息。相对于主流媒体内容，用户产生内容具有噪声大、数据稀疏、结构松散以及潜在主题突发性的特点，因此要求在传统语义分析的基础上加强信息解析以及主题发现与追踪方面的理论方法研究。用户在信息传播方面逐渐由传统互联网下的被动接收角色转变为社会网络下的主动参与角色，以信息接收—信息消化—信息转发的形式对媒体内容进行主动传播。用户不仅可能在互联网上“广播”媒体内容，例如上传图片、视频等。也会向自己的好友分享、推荐有趣的媒体内容。体现用户社会关系的社会网络成为媒体内容重要传播途径。媒体内容带来的影响同样有社会化特征。在社会网络条件下，媒体内容除了自由的语义属性，在产生和传播阶段被赋予了附加的社会属性，如产生形式、热度、时间相关性及用户偏好与影响力作用等，共同构成了媒体内容的多尺度描述。媒体内容对用户的影响取决于用户在信息语义及信息社会属性方面的综合偏好。

社区媒体网络分析与挖掘技术的研究，吸引了当前学术界和工业界的广泛关注，具有重要的研究价值和应用前景。在社会媒体网络中，用户作为信息接收、消化和传播的主体，由于其特有的社会属性和个人特质，形成了对不同信息内容的不同关注度。如何通过对用户的属性信息以及信息浏览历史进行归纳、统计和抽象，进而在信息层面形成与用户信息相关的模型，对于研究信息的传播及社会网络演化机制具有重要的研究意义，同时对于信息定向投放，特定路径信息传播及信息传播泛化等应用模式具有广泛的现实意义。网站系统在运营过程中运用分析挖掘技术来合理预测用户行为，实现社区媒体个性化推荐，用户可以更舒适更便捷的享受网络生活。用户行为预测是监管社区媒体内容及其动态变化，从而维护社会稳定的过程中必不可少的步骤，这是社区媒体网络分析与挖掘技术中很重要的研究领域。

1.2 相关研究工作

本工作的研究主体在于媒体主题挖掘、网络结构分析、网络节点影响建模和用户兴趣建模。国内外在这四个方面的研究都已经有了一定的成果，现状如下所述。

1.2.1 媒体主题挖掘

主题是和人类感知一致的抽象性、概念化信息，反映了媒体的高层语义。媒体主题挖掘有利于媒体内容的组织管理，方便用户访问和搜索媒体内容。媒体内容分析常常利用向量空间模型（vector space model）或统计模型（statistical model）。向量空间模式将媒体内容表示成文本关键词的向量，利用 TF-IDF 方法区分特征权重，并且基于特征空间举例函数度量媒体内容相似性。统计模型深入挖掘特征维度之间的相关性，统计媒体与特征之间的分布规律，建立概率随机模型，度量媒体内容相似性。其中，概率图模型（probabilistic graphical model）是近年来机器学习领域研究的热点。概率图模型是概率理论和图论结合的产物。在概率图模型上，每个节点表示一个随机变量，图的结构表示随机变量之间的独立性关系。基于概率图模型理论，研究者提出和建立了很多主题模型（topic model），挖掘媒体的潜在主题。

Probabilistic Latent Semantic Analysis (PLSA)^[1] 和 Latent Dirichlet Allocation (LDA)^[2] 是最经典的主题模型。主题模型假设数据集合上存在指定数目的潜在主题，表示成关键词上的多项式分布，并且关键词由潜在主题生成。PLSA 模型假设文档是潜在主题的混合，通过模拟文档的生成式过程，得到文档和关键词的联合概率分布为

$$P(d, w) = P(d) \sum_{z \in Z} P(w | z) P(z | d) \quad (1-1)$$

通过最大化训练数据集合的后验概率，求解模型，得到潜在主题以及文档在主题上的多项式分布。

为了避免模型过拟合的问题，LDA 模型^[2]假设所有文档在主题的混合满足狄利赫雷（Dirichlet）先验参数的多项式分布。因此，模型的生成式假设是：每个文档 d 的主题分布 θ 基于狄利赫雷先验分布 α 生成；对于文档 d 中每个关键词 w ，

从多项式分布 θ 随机生成一个主题 z , 关键词从 z 相关的主题分布 β 生成。因此, 模型的后验概率为

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^N \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta \quad (1-2)$$

利用变分推到或者 Gibbs 采样^[3]求解模型, 挖掘数据集合中的潜在主题。

基于主题模型的基本思想, 针对不同的应用, 研究者引入其他信息, 提出了很多新的主题模型, 例如有监督学习的主题模型^[4]、分层结构的主题模型^[5]以及研究主题之间的关联^[6]和动态变化^[7]。

1.2.2 网络结构分析

在过去的几十年里, 一些学者致力于研究社会网络的结构和动态物理特性。社会学家提出“六度空间理论”(Six Degrees of Separation)^[8]和“小世界理论”(small world)^[9,10], 为社交网络的定性特征奠定了理论基础。网络结构分析有利于个体或群体用户行为特性建模, 例如对用户节点的度量可以表示其在社会网络中与其他用户的关联情况。

网络结构与文本内容的结合有利于网络数据的结构分析、内容理解和主题建模。例如, 一些研究者提出假设相邻节点的主题分布具有相似或相关性^[11,12], 还有的学者提出生成式的主题模型^[13], 利用文本信息标注网络节点之间的关系。

1.2.3 网络节点影响力建模

从社交网络用户之间影响力的角度来描述网络节点之间的关系是研究者常用的思路。一些在学术论文引述数据挖掘上所做的工作中, 综合考虑表示引用关系的网络结构和文本内容两部分信息, 预测论文引用行为。研究者还会关注一般化社会网络中用户之间影响力强度的建模, 他们基于节点上已知的主题分布, 提出了 Topical Affinity Propagation (TAP)^[14]方法, 模拟社会网络上主题层面影响力的传递过程, 挖掘节点之间具体的影响力。

1.2.4 用户兴趣建模

用户兴趣偏好建模反映了用户对网络媒体数量的兴趣程度。研究者提出了基于模型的方法来挖掘用户兴趣模式, 实现个性化推荐。主要的推荐方法包括协

同过滤(collaborative filtering)^[15]和基于内容的推荐(content-based recommendation)^[16]两类。基于内容的推荐方法提取用户访问过的内容特征，表示用户兴趣偏好模式，从而预测新项目被用户访问的概率。协同过滤又分成基于项目(item-based)^[17]或基于用户(user-based)^[18]的协同过滤。协同过滤不提取具体的内容信息，只基于访问相同项目的用户来挖掘用户相似兴趣，或者根据被相同用户访问的项目来从相似内容的角度关联项目，实现用户个性化推荐。

1.3 论文工作

1.3.1 工作目标

社区媒体网络中由不同类型的节点与连边组成，并且相同类型与不同类型节点之间都存在不同含义的连边，例如，用户与用户之间的连边，代表了用户之间的好友关系；用户与媒体之间的连边，代表了用户与媒体之间的交互关系。不同类型节点之间存在着大量的关联。媒体与媒体之间的关联，体现在社区媒体之间特征相似性、内容相关性；媒体与用户之间的关联，反应了用户对社区媒体内容的兴趣偏好；用户与用户之间的管理，体现了用户对用户的行为影响力等。

工作的核心目标是能够在社会媒体网络中合理进行用户行为预测。用户的行为不仅和社会网络结构(用户好友关系)相关，也与用户关联的媒体主题相关。从主题层面描述用户行为更加符合实际应用情况。例如，不同主题上同一用户的影响力大小不同。为用户行为进行合理建模，是提高用户行为预测性能的主要方法。

基于建立用户行为模型以预测用户行为的目标，首先需要为媒体内容建立合理的话题模型，这需要对媒体的上下文文本信息进行语义分析，从而在主题层面上理解社区媒体。在媒体内容的话题模型的基础上，分析与用户行为相关的媒体内容，构建用户的兴趣偏好模型，从而描述用户对媒体内容的兴趣程度、用户与用户的偏好相似度。接着，从社交网络结构以及用户行为特征的角度来构建用户影响力模型，以此描述用户与用户之间在行为偏好及话题特征上的相互影响程度。最后，融合媒体内容的话题模型、用户的兴趣偏好模型以及用户的在社交网络上的影响力模型，来构建用户对特定媒体信息的兴趣度模型，这样就可以根据这一算法结果对用户行为进行合理预测，并拓展出为特定用户做媒体信息的个性化推荐等其他应用。

1.3.2 工作难点和重点

由于社区媒体数据大多由非专业的网络用户生成，不具备统一的生成框架与尺度标准，因此真实的社区媒体网络具有噪声大、结构松散、数据稀疏、不平衡的特点，给我们的工作带来了较大的困难，具体体现在以下几个方面：

网络用户不仅制作的社区媒体本身，还会生成其上下文文本信息，包括媒体的标题、标签、分类等。例如，用户会给照片、视频添加标题，评价交流媒体内容。社区媒体的上下文文本信息，体现了用户对媒体内容的理解，隐含了媒体潜在主题。然而，由于网络用户大多是非专业人员，用户行为不是基于统一的框架与标准，具有较强的随意性；甚至有些用户为了吸引眼球，可能会有意的标注错误信息，如添加过于赞美的标题等，因此社区媒体的上下文文本信息存在大量噪声，增加了社区媒体的隐含主题发现与内容监管的困难，也使该工作具有重要的学术研究价值；

体现用户行为模式的用户与社区媒体交互数据具有稀疏与不平衡的特点。一方面，用户浏览某个社区媒体，可能是因为该社区媒体内容主题和用户兴趣模式相符合，也可能是该用户受某个好友影响或者推荐，和用户兴趣模式关系不大。另一方面，用户没有和某社区媒体发生关联，可能是因为用户本身对该社区媒体内容不感兴趣，也可能是用户没有发现该社区媒体，不代表用户本身不喜欢其内容。因此用户行为内在驱动因素非常复杂，且受随机因素影响较大。稀疏、不平衡的用户与社区媒体交互数据是用户行为建模与提高互联网服务质量的阻碍。

1.4 论文组织结构

针对以上研究内容，本文后续章节安排如下：

第2章是为整个工作提供了数据基础。准确分析工作中的数据需求，诸如内容、规模等各个方面。研究获取数据的途径和方法，包括获取和解析等。对不规则的复杂数据结构化成整齐统一的格式。合理运用统计学对数据进行调研，从网络结构数据和媒体数据中验证一些特征规律。

第3章研究社区媒体的话题模型建立和用户偏好分析。整合媒体的上下文文本信息，建立词库和话题索引，继而对文本进行语义分析聚类，得到媒体的话题模型。讨论常见的用户偏好模型，并提出一种适用于社交网络的偏好模型，能够在预测用户行为的性能方面达到最佳，同时检验了媒体话题模型的合理性。

第4章研究了用户影响力模型的建立方法和用户兴趣度模型的建立。在用户的兴趣和影响力模型的基础上，预测社区媒体网络上的用户行为，并验证了融合用户偏好与影响力的行为预测方法效果比起单纯使用其中一种更为优越。在用户行为预测的效果上，我们继而提出了对特定用户进行媒体数据个性化推荐的方法。

第5章总结全文，并展望进一步研究工作的方向。

第2章 数据获取与分析

2.1 数据获取准备

2.1.1 数据内容及获取对象

鉴于研究目标是用户网络行为偏好的建模，我们需要兼顾用户网络行为产生的两大因素：一是与用户行为相关的社会网络媒体数据，包括用户新建、编辑、删除、分享以及评论的各种信息；二是用户所处的社会网络结构，即用户的好友关系以及其好友的网络行为数据。媒体数据经过语义分析可以对其进行模型建立，同时这一模型可以合理描述用户的网络行为偏好；网络结构数据可以归纳得到影响力模型，媒体信息在网络上的传播与用户之间的影响力息息相关。

我们需要选择较为容易获取这两类数据的社会网络，人人网（renren.com）成为最佳选择。人人网是中国最大、最具影响力的 SNS 网站，以实名制为基础，为用户提供日志、群、即时通讯、相册、集市等丰富强大的互联网功能体验，满足用户对社交、资讯、娱乐、交易多方面需求。目前，人人网已经拥有真实注册用户超过 7000 万、PV4 亿、日登录 2200 万人次。2008 年 7 月，人人网正式开放平台，本着开放的态度与所有第三方公司、独立开发人员一起，跨入互联网的“开放”时代，成为人与人联系朋友的互动沟通平台。

人人网所能提供的与用户行为相关的社会网络媒体数据，来源于用户的分享行为，用户可以分享包括日志、照片、相册、视频、链接、市场商品、音乐、个人主页、群组在内的各种信息。同时，人人网所提供的网络结构数据，来源于用户的好友关系，由此形成社区网络的无向图表示。在这一无向图中，结点表示社区网络用户，结点之间的边关系表示用户之间的好友关系。由此，需要访问人人网，以获取的与研究内容相关的两大类数据，其具体内容已经确定。

2.1.2 数据获取工具

数据需要通过访问人人网来获得，最佳获取工具就是网络爬虫。网络爬虫，又称为网页蜘蛛，是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。网络爬虫的实际应用在于很多搜索引擎都依赖它提供最新的数据：网络爬虫提供它访问过页面的一个副本，然后搜索引擎就可以对得到的页面进行索引，以提供快速地访问。另外，网络爬虫也可以在 Web 上用来自动执行一些任务，例如

检查链接，确认 html 代码等；也可以用来抓取网页上某种特定类型信息，例如抓取电子邮件地址等。

网络爬虫的工作原理是它从一组要访问的 URL 链接开始，可以称这些 URL 为种子。爬虫访问这些链接，它辨认出这些页面的所有超链接，然后添加到这个 URL 列表，可以称作检索前沿。这些 URL 按照一定的策略反复访问。

鉴于我们的数据获取对象明确，数据内容清晰，在设计网络爬虫时可以将 URL 规范化，有选择的访问页面，并合理运用并行化策略。

按照本实验的数据要求，自行开发的网络爬虫的抓取目标描述如下：

首先，明确人人网用户信息架构。每个用户在注册的时候获得一个人人 id，其作用是可以准确地唯一地标识这一用户。

接着，通过向人人网 URL 链接 (<http://passport.renren.com/PLogin.do>) 发送登录请求，即发送含有作者的用户名及密码的数据包，获取作者的人人 id，由此我们可以进一步获取作者的人人网信息，包括其行为相关的媒体数据（分享）和网络结构数据（好友关系）。

第三，获取用户的媒体数据，即获取人人网上用户的分享内容。对于人人 id 为 1234 的用户，其分享列表的 URL 链接为：

<http://share.renren.com/share/ShareList.do?id=1234>。人人网的分享类型包括视频、音乐、链接、日志、商品和照片。用户的每一条分享内容都拥有特定的一个分享 id，因此对于上述用户的分享 id 为 5678 的分享内容 URL 链接为：

<http://blog.renren.com/share/1234/5678>。用户分享列表是分页显示的，每页 20 条，用户分享内容数量可以从分享列表首页中解析出来的，由此计算得到分享列表的页面数，例如访问上述用户页号为 2 的分享列表的 URL 链接为：

<http://share.renren.com/share/1234?curpage=2>；页号从 0 开始，因此这一页中含有用户的第 41 条至第 60 条分享内容，倘若用户分享数量不到 60，则至用户最后一条分享内容。

分享内容中包含的信息中基本部分包括类型，分享者和分享时间，其他的信息是需要分类型讨论的。这一部分信息，对于日志、照片和相册来说，指标题和创建者；对于视频、链接来说，指标题和原始链接地址。

第四，获取用户的网络结构数据，即获取人人网上用户的好友关系。对于人人 id 为 1234 的用户，其好友列表的 URL 链接为：

<http://friend.renren.com/GetFriendList.do?id=1234>。用户好友列表是分页显示的，每页 20 条，用户的好友数量是可以从好友列表首页中解析出来的，由此计算得

到好友列表的页面数，例如访问页号为 2 的好友列表的 URL 链接为 <http://friend.renren.com/GetFriendList.do?curpage=2&id=1234>；页号从 0 开始，因此这一页中含有用户的第 41 个至第 60 个好友，倘若用户好友数量不到 60，则至用户最后一位好友。好友的信息包括好友的人人 ID 以及好友姓名，同时可以获取好友就读学校名称。

最后，需要制定进一步获取机制及结束条件。为获取足够数量的网络结构数据，我们采用逐层展开的策略，即从登录用户起，获取其所有好友，作为第一层结点，继而访问第一层每个结点的好友列表，获取第二层结点，以此类推，直至满足结束条件为止。结束条件有两种，一是达到指定层数为止，好处是结构鲜明清晰，但用户数量无法控制，因此第二种结束条件是被访问的用户数量达到要求为止，这种方法利于控制网络结构数据的规模。

根据上述的抓取目标，这一网络爬虫的设计方案和工作原理如下：

网络爬虫的开发平台为 Microsoft Visual Studio 2008，开发语言为 Visual C#。

- (1) 将用户注册电子邮件地址 email 和密码 password 采用 POST 方法发送至登录链接，登录成功后返回个人首页，同时保留返回的一组 Cookie，来保证下次以登录状态访问链接；
- (2) 获取网络结构数据，形成用户列表：
 - a) 将登录用户放入待访问结点链表中，已访问结点链表为空；
 - b) 判断已访问结点链表中结点数量是否满足规模要求，满足则执行步骤 e，不满足继续执行；
 - c) 取出待访问结点链表中第一个结点，采用 GET 方法访问该结点的好友列表链接，使用正则表达式解析出好友数量，逐页访问以获取其所有好友，并全部作为结点插入待访问结点链表尾部；
 - d) 将取出这一结点放入已访问结点链表，回到步骤 b；
 - e) 将已访问结点链表结构化形成用户列表。
- (3) 根据用户列表获取用户行为相关的媒体数据。遍历用户列表中每个结点，采用 GET 方法访问该结点的分享列表链接，使用正则表达式解析出分享数量，逐页访问以获取其所有分享，并解析成结构化数据保存。

2.1.3 数据解析工具

解析获取的 Html 网页以获取结构化数据时，仅仅使用正则表达式是远远不够的。在这里我们使用的数据解析工具是 Html Agility Pack (HAP)，这是一个允许用户解析 Html 网页文件的 .Net 代码库，它构建可读可写的 DOM 接口并支持 XPath 和 XSLT，使用 `HtmlNodeCollection` 和 `HtmlAttributeCollection` 来打开一个对 XPath 的等价 Html 树。

解析网络结构数据，即好友列表网页。好友列表中每个好友的对应网页源代码格式为：（假设该好友的人人 id 为 1234，姓名为 Karl）

```
<dd>
  <a href="http://www.renren.com/profile.do?id=1234"">
    Karl
  </a>
</dd>
```

只需要获取并简单处理该好友的个人首页链接，即可获得人人 ID，姓名更为容易得到。

解析用户行为相关的媒体数据，即分享列表网页。分享列表中每个分享内容的对应网页源代码格式为：

(1) 解析分享类型，以及分享者人人 id 和其姓名，例如：

```
<strong>
  <a href="http://www.renren.com/profile.do?id=1234"">
    Karl
  </a>
  分享照片
</strong>
```

(2) 解析分享时间，例如：

```
<span class="timestamp">
  2010-01-01 00:00
</span>
```

(3) 解析分享链接和标题，例如：

```
<h4>
  <a href="http://share.renren.com/share/1234/5678" target="_blank">
    标题
  </a>
</h4>
```


</h4>

这样就可以得到分享内容的各种信息，并进行结构化。

2.1.4 数据规模描述

网络结构数据。完整展开登录用户的好友，得到第一层结点；接着完整展开第一层结点，得到第二层结点；展开第二层的部分结点，得到第三层的部分结点；这样，我们得到了一定规模的用户结点数量。其中包括 5,141 个被展开的用户结点，85,436 个用户结点。被展开的用户结点的所有好友关系，以边的形式表现在描述用户网络结构的图中。用户结点信息包括用户人人 id，用户名，用户好友关系。

用户行为相关的媒体数据。这一数据涵盖了 5,141 个被展开用户结点的分享信息，共 656,234 条；同时获取了其他部分用户结点的分享信息，媒体数据的总条数为 9,839,677。分享信息包括分享类型、标题、发布者（对于视频等内容，可把原来的链接地址视为发布者）、分享者和分享时间。考查分享信息的分享时间，其时间跨度为 2007 年 2 月 1 日至 2009 年 12 月 20 日。

我们可以按照后面阐述的结构化方法，将处理后的数据录入 MySQL 数据库，以便进一步操作。

2.2 数据的结构化

2.2.1 社区网络结构数据结构化

对于包含 5,141 个被完整展开的 85,436 个用户结点，我们可以结构化结点信息为如下格式（表 2.1）。

表2.1 结构化结点信息

数据结构 User	用户列表中的位置	人人 id	用户名	用户好友关系
变量名称	pos	id	name	friends
变量类型	int	string	string	HashSet<User>
用户例 1	0	1234	Lucas	1,2,3,4
用户例 2	1	1235	Nathan	2,3,5,6

这样的结构优势在于不仅完整地阐明网络结构的图表示，同时便于查询用户列表中结点信息，便于在图中搜索符合某条件的结点，还可以以特定结点为根结点，由此展开其好友关系，并完成后续操作。

2.2.2 社区媒体数据结构化

对于 656,234 条分享记录，我们可以结构化分享信息为如下格式（表 2.2）。

表2.2 结构化分享信息格式

数据结构	在分享列	分享类	分享标	分享记	发布者	分享者	分享时间
ShareRecord	表中位置	型	题	录链接	人人 id	人人 id	
变量名称	pos	type	title	link	owner	sharer	time
变量类型	int	string	string	string	string	string	DateTime

补充说明一下变量特征。`type` 描述的是分享记录的类型，主要包括 `blog`（日志）、`photo`（照片）、`album`（相册）和 `video`（视频）。分享标题 `title` 是一条文本记录，后续会进行语义分析。分享记录链接与分享记录的实际内容是一一映射的关系，对于 `video` 来说，它指代视频来源的链接地址；对于其他三类信息来说，它涵盖了发布者的人人 id 及其在发布者的发布内容 id，因此也可以唯一标识分享记录。分享时间指代分享者分享这一条信息的时间。

举例说明分享信息格式：（用逗号间隔一条分享记录中的上述 7 部分内容）

例 1: 0, blog, 北京奥运志愿者招募开始, <http://blog.renren.com/blog/1234/56789>, 1234, 2345, 2009-7-1 20:00:00

例 2: 12, photo, 暑期赴甘肃民勤社会实践调研土地沙漠化活动海报, <http://photo.renren.com/getphoto.do?id=56789&owner=1234>, 1234, 2345, 2009-7-14 22:10:00

例 3: 74, album, 精美家居装饰设计~一定要分享哦, <http://photo.renren.com/getalbum.do?id=56789&owner=1234>, 1234, 2345, 2009-8-1 12:29:00

例 4: 145, video, 2006 年世界杯决赛录像, http://v.youku.com/v_show/id_ABCDE.html, -1, 2345, 2009-9-15 20:30:00

这样的结构优势在于不仅涵盖了分享记录中足够的信息，同时可以很方便地根据分享类型、发布者、分享者进行筛选，根据分享时间进行排序分类。

2.3 数据调研

2.3.1 可预测的若干特征规律

针对用户网络结构数据、网络行为相关的媒体数据以及其相结合的内容进行合理地数理统计和概率分析，我们可以很容易地验证一些社交网络中的可预测的特征规律。常见的规律表现形式有以下几种：“二八”定律以及数据之间的乘方、指数关系。

“二八”定律，又称为帕累托法则（Pareto principle），是 19 世纪末、20 世纪初意大利经济学家帕累托发明的。他认为，在任何事物中，最重要、起决定性作用的只占其中的小部分，约 20%；其余 80% 的尽管是多数，却是次要的、非决定性的。

可以预见的，社交网络中的现象也是符合这一定律的，更准确的说是长尾效应（Heavy tail）。长尾效应可以形象地用函数描述为： $y_i \propto i^{-\alpha}$ ，其中 i 指代变量的贡献（如数值等）排行， y_i 指代变量所作出的贡献； y 随着 i 值的变化曲线图如图 2.1：

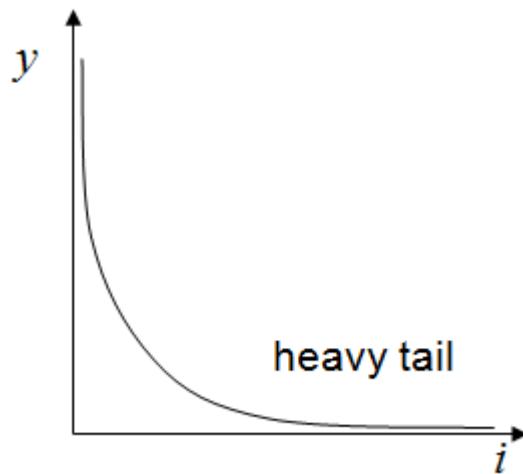


图2.1 长尾效应的曲线图

我们可以提出这样一些假设并加以验证：分享者较多的媒体数据占其总量的小部分；分享内容较多的用户占用户总量的小部分。

另外，我们可以通过统计学方法分析出用户好友的数量与用户分享数量之间的关系，以及对于不同的分享内容类型，分享次数会存在的差异。

2.3.2 特征规律的验证

我们依据已经结构化的数据，可以对上述预测的特征规律进行验证。

(1) 分享者较多的媒体数据占其总量的小部分。

结构化后的媒体数据按照分享者的数量进行排序，将分享者数量作为纵轴，媒体数据在排序后序列中的位置作为横轴，我们可以发现生成的曲线图是符合上述规律的。其中，拥有最多分享者的媒体数据被分享过 733 次，而绝大部分信息的分享次数不足 10 次。以下是针对序列中前 50/200/500/2000 分享内容生成的曲线图，如图 2.2。

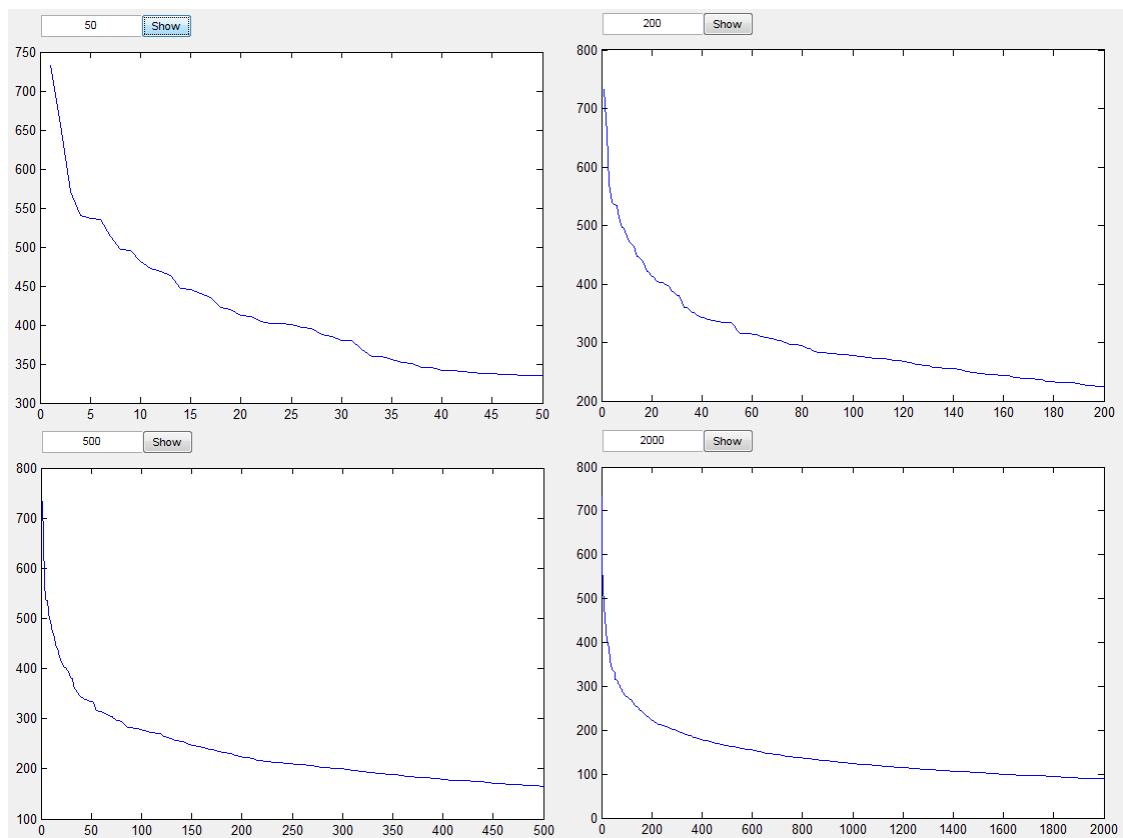


图2.2 分享者较多的媒体数据占其总量的小部分

(2) 分享内容较多的用户占用户总量的小部分。

结构化媒体数据按照分享者进行分类，再对每个用户（分享者）所分享的信息数量进行排序，将用户分享信息数量作为纵轴，用户在排序后序列中的位置作为横轴，我们可以发现生成的曲线图是符合上述规律的。其中，分享内容最多的用户分享过 6929 条记录，而绝大部分用户的分享数量不足 100 条。以下是针对序列中前 50/200/500/2000 个用户生成的曲线图，如图 2.3。

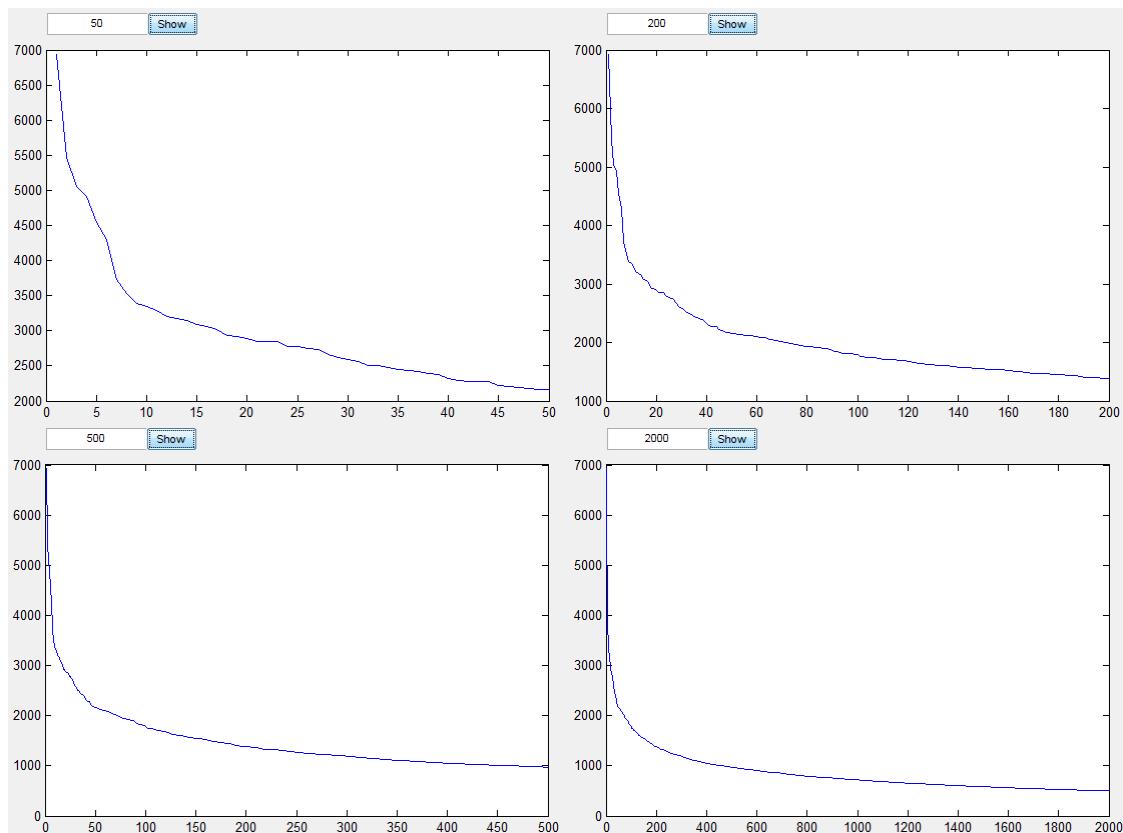


图2.3 分享内容较多的用户占用户总量的小部分

(3) 用户好友数量与用户分享数量之间的关系。

对结构化媒体数据按照分享者（即用户）进行分类，我们可以得到每个用户的分享记录数量；同时对结构化的网络结构数据进行统计，我们可以得到每个用户的好友数量。将用户作为横轴，用户的好友数量和用户的分享数量作为两条折线的纵轴，生成的折线图可以合理地描述出这两者之间存在的关系。

下图（图 2.4）是将用户按照好友数量进行排序后的结果。

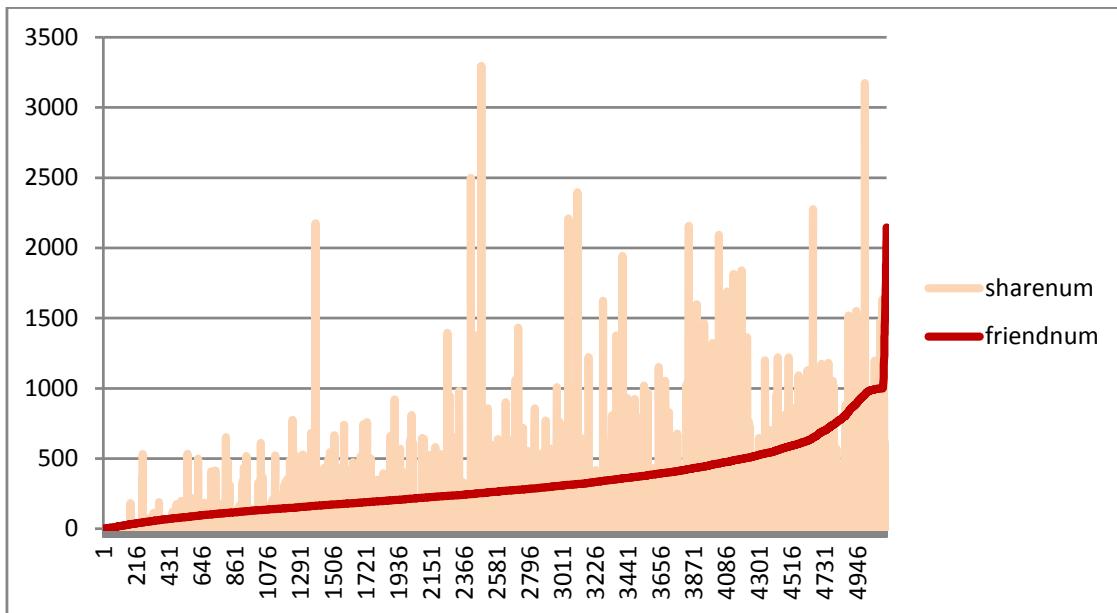


图2.4 用户分享数量随其好友数量变化的曲线

由此可以看出，用户的分享数量从一定程度上说，是随着用户好友数量的增多也越来越多的。究其原因在于，用户的分享数量和用户的好友数量都可以用来描述用户的活跃度：分享数量的多少，能够表现出用户发生网络行为的多少，从而反映用户是否活跃；用户好友数量的多少，能够表现出用户在网络结构中的地位和影响力，这反映了用户在交际领域是否活跃。而上图所反映的现象说明这两种描述活跃度的方式之间是有联系的，即对于一个虚拟社交网络中的活跃用户来说，往往既拥有较多的好友，也产生较多的网络行为；并且，随着好友增多，好友对其的影响会促进网络行为的增多，同时，网络行为的增多又会反作用于其好友关系，能够结交更多的朋友。

下图（图 2.5）是将用户按照分享数量进行排序后的结果。

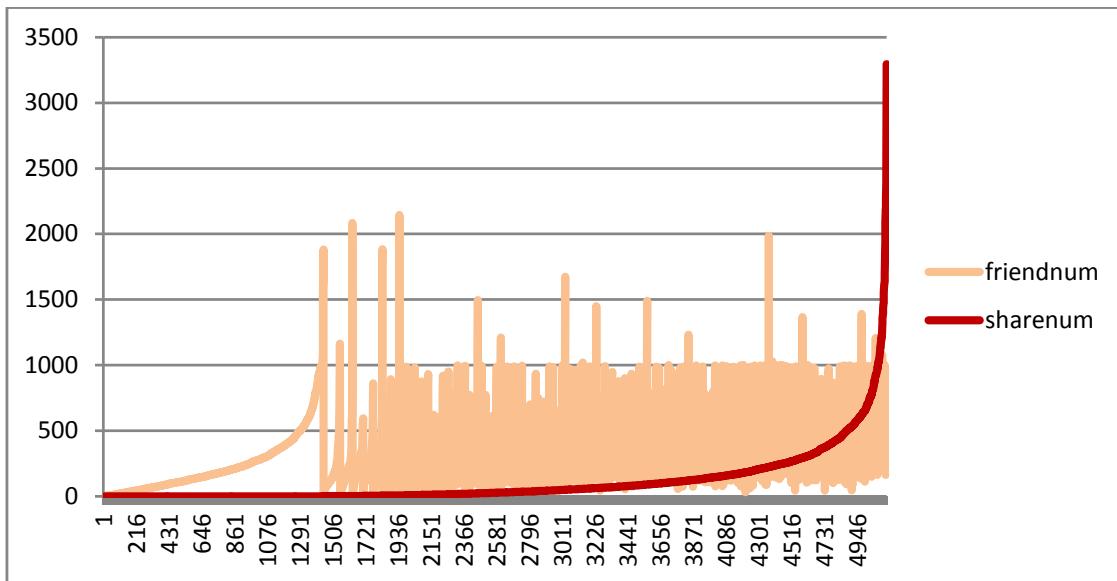


图2.5 用户好友数量随其分享数量变化的曲线

由此可以看出，用户网络行为的活跃程度并不完全由用户的好友数量决定，而是本身反映了用户在网络上的行为特征，这种行为的触发往往与用户所接触到的媒体信息内容息息相关，用户对其偏好的内容会有较大的兴趣，继而产生了分享行为。因此，研究用户的网络行为，尤其是网络行为偏好，必须要研究行为对象，即从语义上分析用户所接触到的媒体数据内容。

(4) 对于不同的分享内容类型，分享次数存在差异。

用户对不同的媒体信息类型，包括日志、照片、相册、视频、链接、商品、音乐等，具有不同的行为偏好，其分享次数存在普遍规律上的差异。下表(表 2.3)描述了日志、照片、相册这三类媒体信息在结构化数据中的数量、以及被分享次数较多的 6 个。

表2.3 三类媒体信息的分享次数

	日志	照片	相册
数量	717,301	444,708	273,185
被分享次数较多的	657	156	404
前 6 个	571	147	388
	540	141	321
	536	131	307
	535	121	283
	514	118	280

由此可以看出，日志是最常见的用户分享行为对象，在数量上远远超过照片和相册，同时其分享次数也较多。相册与照片是同一类型的媒体数据，均是用图像来记录和描述信息。相册比起照片来说，数量较少，但内容上相对较为精华，用户对其发生的分享行为更为频繁。

第3章 话题挖掘与用户偏好分析

3.1 中文分词工具的简介

对于结构化的用户行为相关媒体数据，即用户分享记录，其标题内容很容易被提取，同时形式统一，均为语义文本数据。对文本数据进行语义分析，进而对其进行语义聚类，最后达到合理描述媒体数据特征，建立媒体数据模型的目的。语义聚类的前提在于拥有可聚类文本和词库，因此必须要对海量的语义文本数据进行分词处理。由于人人网用户大多为中华人民共和国公民，因此其中的媒体数据多数为中文。经过一定的调研，实验中采用的中文分词工具为 ICTCLAS 汉语分词开源系统。

中文词法分析是中文信息处理的基础与关键。中国科学院计算技术研究所研制出汉语词法分析系统 ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System)，主要功能包括中文分词、词性标注、命名实体识别、新词识别，同时支持用户词典，支持繁体中文，支持 GBK、UTF-8、UTF-7、UNICODE 等多种编码格式。

由于本实验开发环境为 Microsoft Visual Studio 2008，开发语言为 Visual C#，因此采用 ICTCLAS 提供的动态库 SharpICTCLAS.dll 来进行中文分词。

3.2 话题索引和词库的建立

从海量数据中，提取 9,839,677 条用户分享记录，提取标题内容，去除为空的部分，建立标题内容索引库。索引库（表 3.1）由 6,742,258 条语义文本数据组成。

表3.1 标题内容索引库示例

索引号	标题内容
173	你有真正的朋友吗？（转）
174	转自凤凰网：妈妈别哭，我去了天堂[组图]感动死了…
175	山河失色草木同悲
176	那一刻，中国静止了！
177	清华……悼
178	国务院公告：5月19日至21日为全国哀悼日
179	小学生上网吧被温总理当场抓住
180	TECC 暑期支教及少数民族文化保护项目志愿者火热招募中
181	清华大学学生献血和募捐的一些最终数据
182	情侣钻戒~总会有你心动哒！
183	小测试：你在古代是什么身份 很惨的
184	我们永远是朋友
185	训练归来
186	(转) 看食指，断你的爱情观（好准的说！~）
187	给所有女孩（转）
188	傻瓜！女生这样问你…你要怎么回答？？（很有用哦~）
189	老公与老婆（看完不许哭，但我做不到）
190	女子当如林徽因，情人当若金岳霖

对索引库中每条文本数据进行中文分词。鉴于目标是对文本数据进行语义聚类，我们在分词过程中只保留能够产生聚类意义的词，即名词和动词，这样得到了一个含有 228,486 个词的词库。接着，对词库需要完成一系列的筛选过滤的工作：

第一步，去除词长为 1 的词。这类词的特征是虽然高频，但大多数没有实际的语义，例如：是 (713,207, 括号中为词频，即在索引库分词结果中出现的次数，后面相同)，转 (627,862)，看 (610,759)。词库剩余词数为 224,027。

第二步，去除词频为 1 的词。这类词在索引库中出现次数过少，无法被用来进行语义聚类，例如：Saitou (1)，北条高史 (1)，丘疹 (1)。词库剩余词数为 126,374。

第三步，仅保留词频至少为 10 的词。原因与第二步相同。词库剩余次数为 48,585。

第四步，去除中英文的无用词（stop word）。这类词在文本中起到起承转合的作用，但没有实际语义，例如：可以（112,985），出来（27,298），还有（19,871）。词库剩余词数为 48,299。

最后，根据英文词汇表，以及手动筛选，去除无意义的词汇，得到最终词库。词库中含有 37,633 个词。

这时，重新对原标题索引库中的标题内容进行过滤，即去除分词结果的词都没有在词库中出现过的标题内容，这时得到新的标题索引库（表 3.2），不考虑空记录，共有 883,790 条文本信息。

表3.2 标题内容分词结果示例

索引号	标题内容分词结果
1725	徐州 美食 收藏 上学
1726	日本 组织 曝光
1727	兔斯基 表情 包子
1728	清华大学 学期 节假日 课程
1729	动物 羊驼

分词后的标题索引库中每条索引可以看成一则文本，同时又有了词库，我们可以开始语义聚类的工作。

3.3 标题内容语义聚类

选择话题聚类作为描述媒体数据特征方法的原因在于，分类操作是需要提供已分类词库的，输入要求过强，而聚类操作是根据既有词汇进行合理划分。当前常见的分类词库有《人民日报》中文词库、拼音输入法词库等，但是人人网的媒体数据特点是文本较短，规模巨大，日常用语、网络用语繁多，因此使用分类词库无法达到预期效果。采用浅层狄利赫雷方法（LDA）可以解决这一话题语义聚类模型问题。

对文章的建模过程中，词频特征及词的分布规律特征都是非常重要的。浅层语义索引（Latent Semantic Indexing, LSI）方法认为，每一篇文本都是一个主题

的产物，反映了这个主题。概率浅层语义索引（pLSI）通过引入概率放松了这一限制，它认为文本中可以有多个 topic，每个词从 topic 中产生出来，而用 topic 的分布来表征这篇文本。这样表征文本的维数就从词汇集的量级降到了 topic 数的量级。在此之后，由 David Blei, Andrew Ng 和 Michael Jordan 于 2002 年提出的浅层狄利赫雷（Latent Dirichlet Allocation, LDA）方法引起了人们的重视，它表现为一个主题发现与挖掘的生成模型。LDA 是一种层级贝叶斯生成概率模型，它把文本语料看作离散数据，数据中的每一个元素看作是由底层的有限个混杂在一起的话题（topic）产生出来的，而每一个 topic 又被看作是从一个更底层的 topic 的概率模型中产生出来的。LDA 克服了 pLSI 的理论缺陷，并且继承了 PLSI 的降维优势。

本实验中采用 LDA 方法来对标题内容索引库进行语义聚类，所采纳的 Windows 应用程序为 lda.exe。

输入数据为存储标题内容索引库的文件，其格式为第一行是索引库中的标题内容条数，其余各行为每条标题内容的分词结果，这些词汇均在最终词库中出现，例如“清华大学 学生 社会 实践 表彰会 节目 校庆 校友 访谈”等。

运行 lda.exe 后可以看到调用时采用的命令行参数，如图 3.1 所示。

```
Please specify the task you would like to perform (-est/-estc/-inf)?
Command line usage:
    lda -est -alpha <double> -beta <double> -ntopics <int> -niters <int> -savestep <int> -twords <int> -dfile <string>
        lda -estc -dir <string> -model <string> -niters <int> -savestep <int> -twords <int>
        lda -inf -dir <string> -model <string> -niters <int> -twords <int> -dfile <string>
        lda -batch#data file name must be nips-lda-train.txt and nips-lda-test.txt
```

图3.1 运行lda.exe时命令行参数提示

解释格式 lda -est -alpha <double> -beta <double> -ntopics <int> -niters <int> -savestep <int> -twords <int> -dfile <string> 中的参数：

-est, 是指在输入相关参数之后，根据文本内容集合生成词汇的话题分类结果，以及每条文本的话题模型分布；

-alpha 和-beta, 是两个双精度型参数，在本实验中不妨分别设定为 0.1 和 0.01；
-ntopics, 是整数参数，指代话题模型中话题数量；

-niters, 是整数参数, 指代 lda 运行中的迭代次数;

-savestep, 是整数参数, 指代 lda 运行过程中输出文件保存中间结果的迭代次数间隔;

-twords, 是整数参数, 指代语义聚类后每个话题中的词汇数量;

-dfile, 是字符串, 指代存储标题内容索引库的文件路径。

解释格式 `lda -estc -dir <string> -model <string> -niters <int> -savestep <int> -twords <int>` 中的参数:

-estc, 是指输入相关参数之后, 根据保存的中间结果, 继续运行得到最终结果: 词汇的话题分类, 以及文本的话题模型分布;

-dir, 是指输入文件的目录路径;

-model, 是指输入中间结果的文件路径。

解释格式 `lda -inf -dir <string> -model <string> -niters <int> -twords <int> -dfile <string>` 中的参数:

-inf, 是指输入相关参数之后, 根据输入的 model (可以是-est 或-estc 运行生成的模型中间结果或者最终结果), 以及新输入的标题内容, 生成这一标题内容的话题模型分布;

-dfile, 是指存储新输入的标题内容的文件路径, 这些标题内容将会根据 model 来生成话题模型分布。

将本实验中的标题内容索引库作为输入数据, 使用-est 方法, 调整参数, 多次运行 LDA 生成聚类结果, 并进行人工分析, 以期得到最佳效果 (表 3.3)。

表3.3 LDA调整参数话题聚类

话题数量 ntopics	话题词汇数量 twords	迭代次数 niters	聚类结果 result
50	500	5000	较差
10	2500	10000	一般
20	1000	5000	一般
10	3000	5000	较好
20	1500	2000	较好
10	4000	10000	好

由此可见，当话题数量为 10，每个话题中含有 4000 个词汇，运行迭代次数为 10000 时，聚类结果好。调整参数时发现：话题数量过少，不利于对标题内容话题模型分布的合理性；话题数量过多，导致无法从语义上根据话题中词汇集合合理地总结出话题意义。总词库中词汇数量为 37,633，当仅考虑到词汇的语义鱼类结果，允许其重复出现于不同的话题中时，总词库中的大部分词汇都能归类在至少一个话题中，并且考虑到了某些词汇具有的语义广泛特征。程序运行的迭代次数足够大时，运行结果稳定，效果较好。

首先，词汇在 10 个话题中的分类结果，按照词汇与话题的相似度（这里主要考虑的是词频）排序，列出每个话题的前 12 个词汇（表 3.4）。

表3.4 词汇在10个话题中部分分类结果I

话题 0	话题 1	话题 2	话题 3	话题 4	话题 5	话题 6	话题 7	话题 8	话题 9
中国	男人	视频	英语	时尚	北京	大学	中国	语录	手机
美国	女人	电影	考试	广告	城市	学生	世界	老师	电脑
新闻	星座	音乐	学习	摄影	中国	学院	视频	高考	生活
日本	女生	歌曲	网站	世界	上海	毕业	北京	人生	减肥
总理	男生	明星	大学生	收藏	地震	活动	照片	同学	朋友
世界	女孩	专辑	大学	创意	家乡	中国	开幕式	作文	图片
韩国	爱情	娱乐	专业	love	南京	通知	比赛	高中	游戏
国家	朋友	歌词	工作	品牌	小吃	同学	nba	笑话	照片
历史	老婆	世界	同学	搭配	美食	北京	足球	生活	食物
经济	恋爱	照片	资料	插画	四川	校园	图片	爱情	身体
台湾	生活	日本	英文	日本	旅游	大学生	国庆	兄弟	技巧
人民	老公	台词	面试	欣赏	方言	志愿者	篮球	奋斗	世界

对每个话题中的上述词汇进行语义归纳，可以总结每个话题的内容特征为如下表（表 3.5）。

表3.5 语义归纳的话题特征

话题 0	时政新闻	话题 5	地区特色
话题 1	内心情感	话题 6	校园文化
话题 2	娱乐媒体	话题 7	体育赛事
话题 3	考场职场	话题 8	网络流行
话题 4	时尚潮流	话题 9	生活百态

上述话题的语义归纳是否合理，需要进行检验测试。我们将每个话题分类结果的词汇中，按照词汇与话题的相似度排列的第 13 个至第 24 个（表 3.6）。

表3.6 词汇在10个话题中部分分类结果II

话题 0	话题 1	话题 2	话题 3	话题 4	话题 5	话题 6	话题 7	话题 8	话题 9
危机	文章	演唱会	美国	艺术	全国	学校	冠军	怀念	同学
金融	男孩	mv	答案	韩国	火炬	周年	建筑	文章	眼睛
演讲	孩子	英文	经验	代码	杭州	晚会	体育	中学	女生
社会	幸福	美女	留学	生日	天津	校长	cctv	青春	流感
地震	结婚	周杰伦	中国	衣服	山东	清华	直播	纪念	笔记本
文化	性格	游戏	公司	相册	新疆	演讲	欧洲	孩子	水果
媒体	献给	演员	招聘	大师	西安	招募	国家	照片	生命
记者	女孩子	中文	词汇	女装	山西	上海	美国	天涯	桌面
论坛	学会	姐姐	论文	china	大连	校庆	新闻	姐妹	动物
文章	单身	动画	翻译	流行	照片	文化	抵制	小学	google
凤凰	心理	乐队	实习	化妆品	成都	复旦	刘翔	bbs	系统
政府	生日	摇滚	复习	美女	重庆	教授	家乐福	献给	睡觉

对照话题命名以及上表内容，可以看出对话题的语义归纳是合理的。

3.4 话题模型的建立

运行 LDA 的结果中含有每个 document，即词汇构成的文本，在话题模型上的分布。在这里，document 指代的是标题内容的分词结果。模型分布结果可以引申到对应的标题，继而是对应的分享记录内容，即媒体数据。下表（表 3.7）为标题内容索引库中的若干条标题内容在上述的 10 个话题上的概率分布（话题需用 0 到 9 表示）。

表3.7 标题内容在话题上的概率分布

索引	0	1	2	3	4	5	6	7	8	9
3893	0.033	0.033	0.033	0.700	0.033	0.033	0.033	0.033	0.033	0.033
3894	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.550
3895	0.033	0.033	0.033	0.033	0.033	0.700	0.033	0.033	0.033	0.033
3896	0.025	0.025	0.025	0.025	0.025	0.025	0.775	0.025	0.025	0.025
3897	0.700	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033
3898	0.033	0.033	0.033	0.700	0.033	0.033	0.033	0.033	0.033	0.033
3899	0.025	0.025	0.775	0.025	0.025	0.025	0.025	0.025	0.025	0.025

由此，我们得到了标题内容索引库中每条文本的话题模型结果，同时可以引申至分享记录的索引库，对每一条分享记录建立了对应的话题模型。

我们拥有的话题（topic）数量为 $M = 10$ ，后续均将从时政新闻、内心情感到生活百态的 10 个话题称为话题 0，话题 1 到话题 9。对于一条在索引库中索引为 i 的分享记录（share），话题模型表现为一个话题向量

$\text{share}(i) = [s(i, 0) \dots s(i, M - 1)]$ ，其中 $s(i, j)$ 表示索引为 i 的分享记录在话题 j 上的概率分布值。话题向量符合条件： $s(i, j) \in [0, 1]$ 且 $\sum_j s(i, j) = 1$ 。这也保证了后续操作中用户的偏好向量也符合这一条件。话题向量合理地描述了分享记录的语义特征，由此我们获得了从分享记录到话题向量的一一映射。

3.5 社交网络偏好模型

3.5.1 常见的偏好模型

用户偏好模型的最广泛应用在于推荐技术，不同的推荐技术从根本上是建立在对应的用户偏好模型之上的。目前较为常见的推荐技术有协同过滤推荐（Collaborative filtering recommendation），基于内容推荐（Content-based recommendation），基于效用推荐（Utility-based recommendation）和基于知识推荐（Knowledge-based recommendation）等。基于效用、基于知识的推荐技术在电子商务系统中更为常用，它们对输入数据提出了要能够描述用户需求和建议的要求，在社交网络领域中，难以对这类信息内容从数据上进行合理地描述。因此下面简要介绍前两种推荐技术，及其对应的用户偏好模型特征，并讨论其优点和缺点。

协同过滤推荐技术在个性化推荐系统中应用最广，常见的有基于用户（User-based）和基于项目（Item-based）两种协同过滤技术。

基于用户的协同过滤是到目前为止实际应用中最为成功的个性化推荐技术。它根据其他用户的观点产生对目标用户的推荐列表，基于这样一个假设：如果用户对一些项目的评分比较相似，则他们对其他项目的评分也比较相似。其算法核心是系统使用统计技术搜索目标用户的若干最近邻居，然后根据最近邻居对项目的评分预测目标用户对未评分项目的评分，然后选择预测评分最高的前若干项作为推荐结果反馈给用户。其优点在于所收集的信息大多来自具有相同兴趣偏好的用户，继而根据他们对信息的评价产生推荐结果，但同时存在以下缺点：

- (1) 没有考虑到项目的具体内容，仅在无法分析资源内容的情况下比较适用；
- (2) 社交网络与电子商务领域存在一定差异，用户与用户的关系仅用是否有相同兴趣偏好来描述强弱是不合适的，他们是否存在好友关系、是否有足够的交流频度、是否存在一定的依赖关系都是需要考虑到的。

基于项目的协同过滤根据用户对相似项目的评分预测对目标项目的评分，基于这样一个假设：如果大部分用户对一些项目的评分比较相似，则当前用户对这些项目也比较相似。其算法核心是系统使用统计技术找到目标项目的若干最近邻居，由于当前用户对最近邻居的评分与对目标项的评分比较类似，所以可以根据当前用户对最近评分预测当前用户对目标项目的评分，然后选择预测评分最高的前若干项作为推荐结果反馈给用户。这种方法能够发现内容上完全不同的资源，对新奇信息进行合理的处理，但同时有着一定劣势：

- (1) 没有考虑到用户之间关系在推荐过程中，对用户选择产生的影响；
- (2) 这种对项目进行相似性的归类是建立在评分的数据上的，而并不是内容的分析，缺乏合理性。

基于内容的推荐方法利用信息检索技术（如自然语言处理、文本语义分析）来分析项目的内容，通常应用邻居函数和分类技术来分析和聚类项目的文本内容，并基于项目特征与用户的档案产生推荐。基于内容的信息推荐主要集中在文本信息推荐领域，近些年涌现了大量使用标签完成其他类型媒体信息推荐的技术。其优势在于对信息内容的分析能够从语义上合理描述用户的兴趣偏好，但也存在几个基本的限制：

- (1) 仅仅能够获得项目特征的部分信息，通常为文本信息，其他的内容信息如图形、图像、音频、视频内容都忽略了；
- (2) 这种方法仅对有相似特征的项目进行推荐，并且仅使用目标用户的反馈，尽管用户的兴趣也可能被其他用户的兴趣所影响。

总结上述的推荐技术和所对应的用户偏好模型，我们需要构建一个能够同时考虑到用户与用户之间关系、用户与项目之间关系和项目与项目之间关系的混合用户偏好模型。这种模型克服了上述的缺点，较为适用于社交网络领域。

3.5.2 适用于社交网络的用户偏好模型

在构建适用于社交网络的用户偏好模型之前，我们分析一下用户在社交网络中网络行为的触发特征。在社交网络（如人人网）中，用户对媒体信息发生分享行为的触发原因可以分为以下两种情况：

- (1) 用户注意到好友对某条媒体信息的分享推荐，同时用户浏览后对该媒体信息产生了兴趣；
- (2) 用户仅因兴趣而主动对媒体信息产生了分享行为，而非受到了好友的影响。

下面通过一个样例来说明对于一条媒体信息，用户如何在社交网络中发生分享行为。

人人网用户在新鲜事列表中关注到其好友对标题为“2006年德国世界杯十大精彩进球”视频的分享消息，点击进去后观赏该视频，并决定分享。这种分享行为的触发属于情况一。情况二指的是该用户在人人网的热门分享、优酷网土豆网或其他视频网站发现了该视频，观赏后进行分享，这一行为会成为一条消息出现在其好友的新鲜事中。

基于用户的协同过滤仅仅考虑到该用户与这一好友是否有共同兴趣，与之有共同兴趣的好友对该视频的评价如何，即用户与媒体信息之间的关系；基于项目的协同过滤仅仅考虑该用户之前的网络行为是否能说明他喜欢这一类的媒体信息，也就是他是否分享过与之相关联的内容，即媒体信息与媒体信息之间的关系；基于内容的方法只考虑到这一媒体信息的文本语义，从用户之前的网络行为中总结其兴趣偏好是否与之相吻合，即媒体信息的语义内容。

适用于社交网络的用户偏好模型必然需要考虑到上述三方面的因素，同时需要兼顾两种触发分享行为情况。其优势在于充分利用用户网络结构特征，将用户与用户之间的影响关系考虑进来，这与人们实际的行为习惯相吻合。因此，与这一偏好模型相关的有以下四个因素：

- (1) 媒体信息的语义内容。我们已经从结构化的媒体信息数据中提取出了标题内容，进行语义分析聚类，建立了媒体信息的话题模型，并且得到了合理性的检验测试。
- (2) 媒体信息与媒体信息之间的关系。媒体信息的话题模型相似度的测量方法可以简单描述为媒体信息的话题向量之间的夹角余弦值。当然，媒体信息的话题模型本身就可以用来对其在语义上进行归类。
- (3) 用户与媒体信息之间的关系。基于用户所分享的媒体信息构建的用户偏好向量可以合理描述用户所感兴趣的话题分布特征。结合媒体信息的话题模型，可以得到用户与媒体模型之间的契合程度。
- (4) 用户与用户之间的影响关系。这是社会网络中信息传播的最突出特点，我们可以用影响力模型来描述。影响力模型中的三个要素的体现形式为：在用户偏好向量基础上求解用户偏好的相似度；使用统计技术分析用户的交流频度；利用社区结构挖掘技术分析用户的网络结构紧密度。

利用适用于社交网络的用户偏好模型，结合媒体信息的话题模型，社交网络结构的影响力模型，我们可以构建用户的兴趣度模型，即在已知给定用户与给定媒体信息特征的条件下，合理测量用户对于媒体信息的兴趣度大小，从而实现对用户行为进行预测、为用户推荐媒体信息等应用。

在进行话题模型检验测试时，我们提出了这样的假设：实验中用户的偏好，仅与其分享记录的内容有关。我们可以利用分享记录的话题分布，可以合理地描述用户的偏好。我们可以通过在分享记录索引库中筛选出分享者为某个用户 p ，继而按照时间进行排序，就可以获得用户 p 的分享记录列表。假设该分享记录列表的长度为 N ，即用户 p 共分享过 N 条内容，那么 p 的分享记录列表可以表示为

$\text{sharelist}(p) = \{\text{share}(0) \dots \text{share}(N - 1)\}$, 其中 $\text{share}(i)$ 指代用户 p 所分享的第 i 条内容。基于之前的假设, 我们可以得到用户的偏好向量 $\text{person}(p) = \sum_{i=0}^{N-1} \text{share}(i)/N$ 。

实际上, 用户的偏好向量描述了其偏好在已知话题上的概率分布, 因此, 用户偏好向量可以表示为 $\text{person}(p) = [p(0) \dots p(M - 1)]$, 其中话题数量为 $M = 10$ 。对于每个分享记录 $\text{share}(i)$, 对应得到它的话题向量为 $\text{share}(i) = [s(i, 0) \dots s(i, M - 1)]$, 那么用户 p 的偏好在话题 j 上的概率分布值 $p(j) = \sum_{i=0}^{N-1} s(i, j)/N$ 。易知, $p(j) \in [0, 1]$ 且 $\sum_j p(j) = 1$ 。

3.6 用户偏好检验话题模型

检验测试分享记录的话题模型是否合理, 可以人为地从语义上判断分享记录的内容是否与得出的话题模型相符合, 例如:

对于标题内容为“(转) 清华大学学生社会实践表彰会节目——校庆期间校友访谈”的分享记录, 我们经过分词后得到文本为“清华大学 学生 社会 实践 表彰会 节目 校庆 校友 访谈”, 在 LDA 方法语义聚类后, 这条分享记录在 10 个话题上的分布如下 (表 3.8)。

表3.8 分享记录样例在话题上分布

话题 0	话题 1	话题 2	话题 3	话题 4	话题 5	话题 6	话题 7	话题 8	话题 9
0.033	0.033	0.033	0.033	0.033	0.033	0.700	0.033	0.033	0.033

由此可见, 该分享记录在话题 6, 即“校园文化”方面的概率分布值明显比其他话题上的分布值要高。我们从语义上也可以看出这条分享记录是与大学校园及社会实践相关的, 是属于“校园文化”范畴。同理, 我们可以随机地抽取若干条分享记录的话题模型建立结果进行人为地检验测试。结果证明, 这一话题模型的建立对于分享记录来说是合理有效的。

检验测试仅仅针对分享记录进行测试, 具有数据量大、判断难度大等缺点, 因此我们还需要针对用户的网络行为进行检验测试。这一检验测试提出了两点假设。

假设一，用户的网络行为偏好，这里指分享行为偏好，仅与其分享记录的内容有关。我们可以利用分享记录的话题分布，可以合理地描述用户的网络行为偏好。

假设二，用户的现实社会生活和兴趣爱好，与其网络行为偏好相关。我们可以根据用户在现实生活中的社会身份以及特长爱好，预测出用户在网络行为上的偏好。

在这两点假设的基础上，对于每个已知其社会身份和特长爱好的用户，将其所有分享记录在 10 个既定话题上的概率分布，进行平均处理，然后将话题按照概率分布值排序，挑选较大的前三个进行归一化后，列举如下（表 3.9）。

表3.9 实验分析考察用户关心的话题及分布值

用户 编号	用户描述 (均为清华大学学 生或老师)	最关 心话 题	最关心分 布值	次关 心话 题	次关心分 布值	较关 心话 题	较关心分 布值
1	校团委某负责人	校园	0.556485	时政	0.221757	赛事	0.221757
		文化		新闻		体育	
2	校团委某负责人	校园	0.393667	时政	0.304715	考场	0.301618
		文化		新闻		职场	
3	毕业出国的大四学 生, 酷爱篮球	体育	0.366759	娱乐	0.316621	考场	0.316621
		赛事		媒体		职场	
4	校教育扶贫公益协 会会长, 酷爱篮球	校园	0.419525	体育	0.320585	网络	0.25989
		文化		赛事		流行	
5	计算机系七字班本 科生辅导员	校园	0.51706	地区	0.350831	生活	0.132109
		文化		特色		相关	
6	计算机系六字班本 科生辅导员	校园	0.401535	时政	0.375959	考场	0.222506
		文化		新闻		职场	
7	计算机系六字班本 科生辅导员	校园	0.447326	时政	0.303716	考场	0.248958
		文化		新闻		职场	
8	计 61 班 (国防班) 学生, 党员	时政	0.403929	内心	0.298035	生活	0.298035
		新闻		情感		相关	
9	计 63 班学生, 党员	时政	0.345774	网络	0.333934	生活	0.320293
		新闻		流行		相关	
10	计 63 班学生, 毕业 出国	考场	0.392833	网络	0.350289	地区	0.256878
		职场		流行		特色	
11	计 62 班女生	内心	0.541875	考场	0.277125	时尚	0.181
		情感		职场		潮流	
12	计 73 班女生, 备考 出国	内心	0.377299	考场	0.341327	网络	0.281375
		情感		职场		流行	
13	计 80 班女生	内心	0.38235	时尚	0.369928	校园	0.247722
		情感		潮流		文化	
14	计 64 班女生	时尚	0.391386	内心	0.319705	娱乐	0.288909
		潮流		情感		媒体	

对于上述测试结果，我们可以通过身份分析发现其合理性：

在清华大学的校团委、教育扶贫公益协会、计算机系学生工作组担任社会工作的老师和学生，关心的词条往往含有学校、校园、文化、大学、学院、校长、教授、大学生、学生、同学、志愿者、招募、演讲、毕业、校庆、周年、晚会、活动、通知、中国、北京等词汇，这类词条往往属于“校园文化”这一话题。因此，他们最为关注的话题为“校园文化”这一现象是可以预见的。

计划毕业出国的学生，需要准备应对 GRE、托福等考试，他们对于“考场职场”这一话题的内容更为关注；酷爱篮球的学生，往往会分享与“体育赛事”相关的日志、视频等内容；担任社会工作或者拥有党员身份的学生或老师，关心“时政新闻”的话题；女生往往比较关注与“内心情感”、“时尚潮流”方面的内容。

数值上的差异反映了不同用户在话题上的偏好差异。例如，我们可以发现不同的女生对“内心情感”内容的关心程度并不相同。

我们通过用户调查（User Study）来检验其合理性。调查人群为认识考察用户尽可能多的同学，主体为计算机系学生；设计调查问卷，发放填写，共 24 份。调查问卷（表 3.10）待填写的表格如下：

表3.10 调查问卷表格

请填写你所认识的考察用户在你看来，他们第一、第二、第三关心的话题类型对应的序号：			
0. 时政新闻 1. 内心情感 2. 娱乐媒体 3. 考场职场 4. 时尚潮流 5. 地区特色 6. 校园文化 7. 体育赛事 8. 网络流行 9. 生活百态			
(不认识考察用户，则空行)			
考察用户名	第一关心的话题序号	第二关心的话题序号	第三关心的话题序号
(14位考察用户的姓名，他们的社会身份在前面的表中给出，问卷中给出具体姓名)			

问卷填写结果（表 3.11）经过统计分析如下：

我们定义序号 1、2、3 为第一、第二、第三关心的话题序号。问卷共收回 20 份，问卷中会有空行，也就是我们统计实验中结果的每个序号可以得到多少分数。分数是这样定义的：被调查者提供的考察用户关心话题序号，如果认为是其第一

关心，则得 3 分，第二关心为 2 分，第三关心为 1 分。由此我们可以得到实验结果的每个序号得到的调查者所给分数的总和。

表3.11 用户调查结果统计

考察用户	实验中的结果			给分人数	调查结果			得分/总分
	序号 1	序号 2	序号 3		分数 1	分数 2	分数 3	
1	6	0	7	5	11	7	1	19/30
2	6	0	3	3	6	7	3	16/18
3	7	2	3	11	28	15	6	49/66
4	6	7	8	15	34	28	8	70/90
5	6	5	9	7	14	6	8	28/42
6	6	0	3	9	13	20	6	39/54
7	6	0	3	9	10	21	8	39/54
8	0	1	9	6	16	5	6	27/36
9	0	8	9	13	35	10	6	51/78
10	3	8	5	14	28	10	10	48/64
11	1	3	4	7	17	8	13	38/42
12	1	3	8	5	14	6	4	24/30
13	1	4	6	2	5	3	0	8/12
14	4	1	2	12	34	22	10	66/72

我们发现绝大部分的情况下，分数总和都能够达到足够的大小来证明这三个话题是用户所关心的；且分数呈现递减，用户评价与实验结果相吻合。

我们从用户网络行为偏好的角度检验得知，上述为分享记录建立的话题模型是合理的，进而得知，由话题模型生成的用户偏好模型是合理的。

第4章 用户偏好与影响力融合的行为预测

4.1 网络结构图的建立及可视化

结合社区网络结构数据，即用户好友关系数据，我们可以用点描述社区网络用户，边描述用户之间的好友关系，由此生成的无向图就是社区网路结构图。

图中的点包含的信息有特定用户的人人 id 和姓名，以及其他可扩展信息。可扩展部分指代后续操作中可以添加的用户的分享记录信息、用户网络行为偏好模型的概率分布值等。图中的边表示用户的好友关系，如果将用户之间影响力视为无向关系，可以添加边的权值，形成影响力图。

社区网络结构图规模庞大，为便于研究和开发，可以搭建平台来进行可视化表示。我们使用图探索系统 GUESS^①来导入数据，并可视化社区网络结构图，并发掘其中的规律。GUESS 采用支持图形感知的嵌入式语言 Gython（Python 的扩展），系统能够解析数据并形成结构化的图形。GUESS 所展示的可放缩的界面中，用户可以关注到图中点和边的特征信息，同时可以选择性的聚焦到某个区域来观察结构关系。最可贵的一点是系统提供了多种图的点布局方式，能够利用布局算法挖掘出图中的可视化特征，从而了解社区网络图的结构特征。

提供给 GUESS 系统的数据时 gdf 格式的，共分成两个部分：

- (1) 节点信息。以“nodedef> name, prop VARCHAR(10)”行开头，后面每行是节点的编号和节点信息，这里是指用户姓名。
- (2) 边关系信息。以“edgedef> node1,node2”行开头，后面每行是边的两个端点的编号。

我们导入数据后，希望能够看到反映社区网络结构特征规律的布局，这里选用的是 Fruchterman-Rheingold Layout 算法。下面我们将对三个不同的用户的网络结构图进行比较，总结出其中的差异与相同点。

^① GUESS 是一种图探索系统，网站首页 <http://graphexploration.cond.org/>

下图（图 4.1）是用户 1 的社区网络结构图。

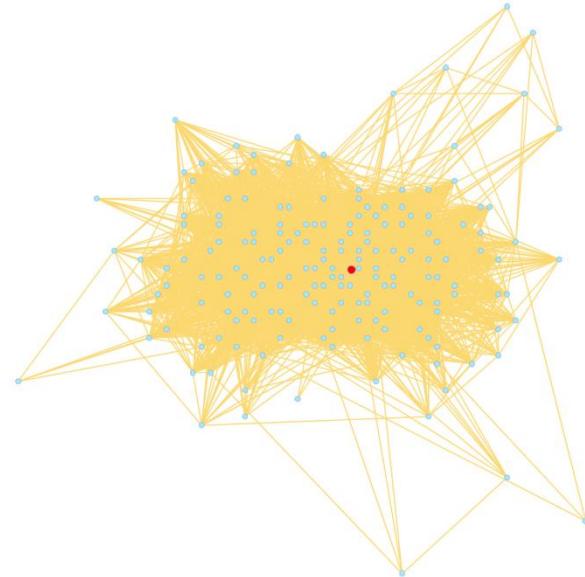


图4.1 用户1的社区网络结构图

下图（图 4.2）是用户 2 的社区网络结构图。

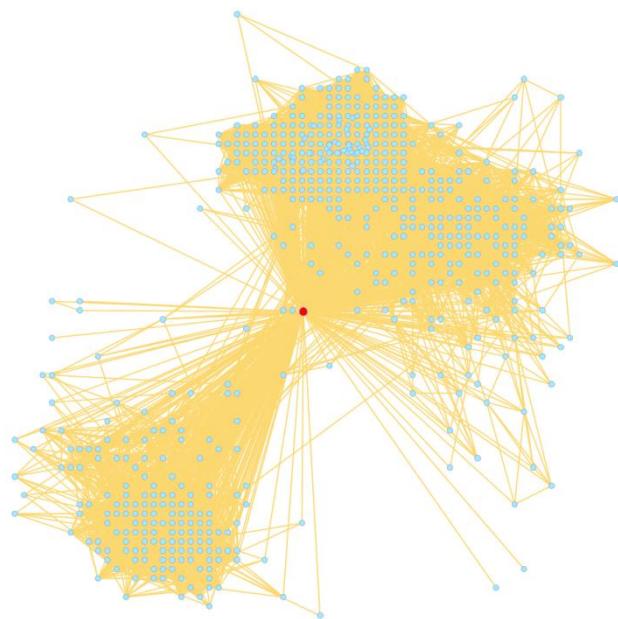


图4.2 用户2的社区网络结构图

下图（图 4.3）是用户 3 的社区网络结构图。

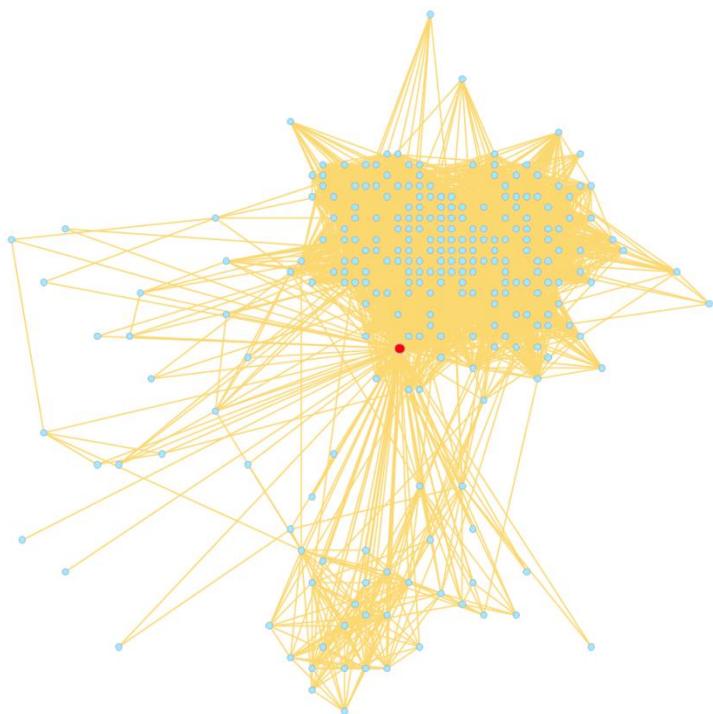


图4.3 用户3的社区网络结构图

我们可以发现，用户 1 的好友群呈现团状，而用户 2 与用户 3 的好友群都分隔成两个部分。这在社会学中认为三个用户所在的“小社区”数量分别为 1, 2, 2。用户 1 所认识的人群社会身份较为接近，例如，他所认识的人都在他所在的公司或者地区。用户 2 结识的好友可以分为两个人群，不同人群之间彼此的联系很少，并且用户 2 处于中间位置，也就是说他与两个人群的联系较为平衡，例如，他与他的大学同学和高中同学之间都有着频繁的交流。用户 3 同样与两个人群结识，但他所在的位置偏向于更大的人群，说明他与这片人群交流较多，例如，他与高中同学的联系渐渐淡去，而更多的是生活在大学同学中间。

这样的可视化结果不仅可以反映出社会网络结构特征，更是可以发现用户之间的不同特点，从而在对用户模式有所了解的基础上实现进一步的应用。

4.2 用户影响力的概述

在现实社会网络中，每个人是通过各种关系联系在一起的。社会网络分析领域里，一些得到广泛研究的关系有：

- (1) 个人之间的评价关系，如喜欢、尊重等
- (2) 物质资本的传递，如商业往来、物资交流
- (3) 非物质资源的转换关系，如人与人之间的交往、信息的交换等
- (4) 隶属关系，如属于某一个组织
- (5) 行为上的互动关系，如人与人之间的谈话、拜访等
- (6) 正式关系、权威关系，如教师学生、医生病人、老板职员关系等
- (7) 生物意义上的关系，如遗传关系、亲属关系以及继承关系等

人与人之间的“多元关系”，也就是联系，是值得重点关注的。例如，两个学生之间可能同时存在同学关系、友谊关系和恋爱关系等，但可以综合为他们之间的联系。这些联系的强弱，就称之为人与人之间的影响力。联系按照联系的强弱可以分为强联系和弱联系。人们与其较为紧密、经常联络的社会关系之间形成的是强联系。与之相对应的，与其不紧密联络或者间接联络的社会关系之间形成的是弱联系。

在传递资源、信息、知识过程中，一般认为弱联系更具有重要性。强联系之间由于彼此很了解，知识结构、经验、背景等相似之处颇多，并不能带来进一步的新的资源信息和知识，所增加的部分大多是冗余的。而弱联系所提供的资源信息或知识会比较具有差异性，如果在弱联系之间搭起某种形式的桥梁，就可以传递多种多样的资源信息和知识。

在虚拟社会网络中，用户与用户之间的联系既存在强联系也存在弱联系，我们需要一种统一的方法来描述用户与用户之间的影响力。如果我们把每一个用户看作一个点，他们之间的联系看作边关系，由此形成的图需要考虑到联系的特征来确定为无向图或是有向图。联系中的常见关系往往是具有单向特征的，尤其是在虚拟社会网络中。这与网络结构图是有区别的。仅采用用户好友关系建立的网络结构图中，点指代的是用户，但边关系指代用户之间的好友关系。由于其特定的行为模式，这种关系往往式无向的。例如，在人人网中，好友关系必须通过发送请求并经过同意才能建立，由此建立的关系对于双方是等价的。然而，某一用户 A 对用户 B 感兴趣，于是向 B 发送添加好友申请，B 同意之后，A、B 之间就建立了好友关系，这样造成的结果是如果 A 是 B 的好友，那么 B 也一定是 A 的

好友，但 B 有可能不认识 A。同样的，用户 A 因为对用户 B 感兴趣，很关心 B 所分享的内容，而 B 有可能完全不关心用户 A 的分享内容。这样形成的关系就是具有单向特征的。

考虑到上述原因，我们需要在网络结构图的基础上，将边关系调整为有向带权边，用以描述某用户对另一用户的影响力。同时，影响力值可以用 0 到 1 之间的双精度浮点数来表示。

4.3 用户影响力模型的建立

由上所述，我们可以假设用户 A 对用户 B 的影响力（联系的强弱）为 $I(A \rightarrow B) \in [0,1]$ 。影响力值的确定必须综合与之相关的各种因素，主要有以下几点：

4.3.1 用户的偏好相似度

描述朋友的成语里常常会见到“志同道合”、“志趣相投”、“知音难求”，朋友关系往往建立在“共同语言”的基础上。这一常见的社会现象引申到虚拟社会网络中同样适用：拥有某一方面的兴趣偏好的用户，在关注社交网络传播的这一领域内容时，往往会优先浏览他所了解的同样拥有这一方面兴趣偏好的用户所传播的信息。一是“专家效应”。对某一领域涉猎颇深的用户所传播的内容往往更为经典、精华、引领潮流，更为切合当前的需要，有更多的浏览价值，这会直接导致其他关心这一领域的用户的青睐。二是两者之间好友关系的形成可能是建立这一领域的信息交流之上的。两个用户之间相互了解，清楚对方对某一领域有很强烈的兴趣；或者恰恰因为现实生活中了解对方与自己有共同爱好，而申请好友形成了好友关系。这都决定了用户之间的影响力与用户偏好的相似度相关。在研究用户偏好相似度的过程中，我们假设这一影响力不存在方向，即从用户 A 观察用户 B 与其的偏好相似度与从 B 观察 A 所得的是相同的。

根据用户偏好模型的设定，样本用户 p_1 和 p_2 的偏好向量分别为 $\text{person}(p_1) = [p_1(0) \dots p_1(M-1)]$ 和 $\text{person}(p_2) = [p_2(0) \dots p_2(M-1)]$ ，这里用向量间的夹角余弦值来计算用户偏好之间的相似度。假设样本 $\text{person}(p_1)$ 和 $\text{person}(p_2)$ 之间的夹角为 $\theta(p_1, p_2)$ ，显然 $\theta(p_1, p_2) \in [0, \pi/2]$ 。用户偏好相似度为：

$$\begin{aligned}
simi(p_1, p_2) &= \cos[\theta(p_1, p_2)] \\
&= \frac{person(p_1) \cdot person(p_2)}{\|person(p_1)\| \cdot \|person(p_2)\|} \\
&= \sum_{j=0}^{M-1} p_1(j) \cdot p_2(j)
\end{aligned} \tag{4-1}$$

用户 p_1 对用户 p_2 、用户 p_2 对用户 p_1 建立在偏好相似度上的影响力为：

$$\begin{aligned}
infl(p_1 \rightarrow p_2) &= infl(p_2 \rightarrow p_1) \\
&= simi(p_1, p_2) \\
&= \sum_{j=0}^{M-1} p_1(j) \cdot p_2(j)
\end{aligned} \tag{4-2}$$

4.3.2 用户的交流频度

20世纪70年代以来全球最知名的社会学家之一，格兰诺维特（Mark Granovetter）在其1973年发表的著名论文《弱关系的力量》中提到：人际关系强度的测量可能是对时间累积量（包括频度和持续时间）、情感紧密度、彼此熟识程度、信任程度以及互惠服务的线性综合。

也就是说，如果两个人花在关系上的时间越多、情感越紧密、相互间的信任和关照越多，这种关系就越强，反之则越弱。因此，在虚拟社会网络中分析用户之间影响力，他们的交流频度是不可忽视的。

社交网络上的交流是双向进行的。我们可以通过交流频度、每次交流的持续时间、数次交流的总时长以及两次交流之间的时间间隔来衡量两个人在网络接触方面的关系强度。同时交流是具有方向性特征，即谁是某次交流的发起者，谁是响应者，我们可以认为响应者从某种程度上说，在此次交流过程中收到了发起者的影响更多。由于受到数据资源的限制，我们在如下假设下建立基于用户交流频度的影响力模型：一是不考虑发起者与响应者之间的差异；二是不考虑每次交流持续时间的长度，仅使用交流频度的数据。用户之间的一次交流可以有很多表现形式，包括页面的访问，在留言板上的留言，在发布的信息（如状态、日志、照片、相册以及分享内容）上的回复等。它们所产生的交流强度不同，因此我们需要采用加权的方法。

假设在对所有用户相同的时间段内，用户 p2 访问用户 p1 页面、p1 访问 p2 页面的次数分别为 $n_{visit}(p1 \rightarrow p2)$ 和 $n_{visit}(p2 \rightarrow p1)$; p2 在 p1 的留言板上留言、p1 在 p2 的留言板上留言的数量分别为 $n_{note}(p1 \rightarrow p2)$ 和 $n_{note}(p2 \rightarrow p1)$; 同理的，p2 在 p1 发布的信息上的回复、p1 在 p2 发布的信息上的回复总数分别 $n_{message}(p1 \rightarrow p2)$ 和 $n_{message}(p2 \rightarrow p1)$ 。继而调整参数以设定页面访问、留言板留言、发布信息的回复在决定影响力方面的权值分别为 μ_{visit} 、 μ_{note} 和 $\mu_{message}$ 。当然，对于发布信息回复的部分，可以根据情况拆解成 n_{state} 、 n_{blog} 、 n_{photo} 、 n_{album} 和 n_{share} 。那么，用户 p1 与用户 p2 之间建立在用户交流频度上的影响力为：

$$infl(p_1 \rightarrow p_2) = \sum_{type \in \{visit, note, message\}} \mu_{type} \cdot n_{type}(p_1 \rightarrow p_2) \quad (4-3)$$

和

$$infl(p_2 \rightarrow p_1) = \sum_{type \in \{visit, note, message\}} \mu_{type} \cdot n_{type}(p_2 \rightarrow p_1) \quad (4-4)$$

然而，社会网络的稀疏特征以及用户交流的互动性直接导致用户之间建立在交流频度上的影响力的方向性是无法清晰区分的，我们不妨假设影响力结果为：

$$\begin{aligned} infl'(p_1 \rightarrow p_2) &= infl'(p_2 \rightarrow p_1) \\ &= infl(p_1 \rightarrow p_2) + infl(p_2 \rightarrow p_1) \end{aligned} \quad (4-5)$$

4.3.3 用户的网络结构紧密度

大规模社会网络的社区结构具有一些特殊的性质，如小社区稳定性、个体中心性等。这样的性质使得用户之间影响力的作用会与网络结构有关。

小社区稳定性。在现实的社会网络中，一个社区或者成为一个团体的规模在一个范围内时，这个社区就处于相对稳定的状态中，但是如果社区的规模过大或者过小，社区就容易发生变化，例如大的社区可能会分裂为小的社区，而小的社区则可能会继续发展直到它处于一个相对稳定的状态中。著名的“Dunbar 数字”（150 法则）称人类大脑认知能力将允许人类拥有稳定的社交网络人数为 148，

四舍五入大约为 150 人。很多机构和社团的构建似乎都合乎这一规律。因此，有些社会学家认为应该将一个人宽广的社交范围和他的社交“核心”区分开来。换句话说，社交“核心”区的好友与其他关系好友对用户的影响力是截然不同的。

个体中心性。社区中的成员是频繁变化的，但是社区中核心成员的变化频率相对要小的多，而且整个社区多半都是围绕一个或多个核心成员自发建立的。所以，越处于中心地位的用户，其对周围人的影响力越大。同时引申而知，处于相同社区的数量越多，两个人之间的相互关系应当越紧密。

建立在上述理论基础上的一个思路，就是挖掘用户的小社区结构，判断小社区的重复性和用户的中心性。而对两个用户之间在网络结构方面影响力测量的更为简单的处理，是分析两个用户共同好友的数量。

假设用户 p_1 、 p_2 的好友集合分别为 F_1 和 F_2 ，他们的共同好友集合为 $F_{common} = F_1 \cap F_2$ ，那么可以定义用户 p_1 与 p_2 之间的影响力为其网络结构体现在共同好友关系上的紧密度：

$$\begin{aligned} infl(p_1 \rightarrow p_2) &= infl(p_2 \rightarrow p_1) \\ &= common(p_1, p_2) \\ &= \frac{\|F_{common}\|}{\|F_1\| \cdot \|F_2\|} \end{aligned} \tag{4-6}$$

在构建用户影响力模型时，上述因素都是需要考虑到的，但因为数据来源等限制条件，我们需要先行分析因素被采纳带来的优势和劣势（表 4.1），继而进行合理的选择。

表4.1 三种用户影响力相关因素的讨论

用户影响力相关因素	优势	劣势
偏好相似度	结合与用户网络行为相关的媒体数据，分析语义内容，考虑到用户行为偏好	将好友关系视为均等，仅用是否存在边关系来表示，无法体现网络结构特征
交流频度	合理地描述人际关系强度，最为接近现实社会生活人与人之间影响力特征	数据获取的难度较大，模型构建中参数难以确定
网络结构紧密度	结合用户关系数据，分析用户网	对用户网络结构挖掘的准确度

络的结构特征，考虑到社交网络 难以保证，同时不同用户拥有不同的社区特点 同的网络结构特征

综上，兼顾获得的数据特征，我们在后续实验中仅采用第一个因素来描述用户之间的影响力，即结合用户的分享记录，分析用户偏好相似度。

4.4 用户兴趣度模型

4.4.1 兴趣度相关的变量定义

用户 user 对媒体信息 media 的兴趣度的度量是在某一确定时刻 t 进行的。

该用户的网络结构中存在 N 个好友关系；媒体信息和用户的偏好都表示为建立在 M = 10 个话题上的概率分布模型，即话题模型和偏好模型；定义用户好友列表中第 i 个好友为 u_i，第 j 个话题为 t_j，显然有 0 ≤ i < N 且 0 ≤ j < M。

构建媒体信息的话题模型，即话题向量为 S = [s₀ … s_{M-1}]，其中 s_j 表示该媒体信息或者说分享记录在话题 j 上的概率分布值。

构建用户的兴趣偏好模型，即偏好向量为 P = [p₀ … p_{M-1}]，其中 p_j 表示该用户在话题 j 上兴趣偏好的概率分布值。

构建用户好友对该媒体信息的分享向量为 F = [f₀ … f_{N-1}]，其中定义 f_i 为好友是否在 t 之前的时刻分享过该媒体信息，即

$$f_i = \begin{cases} 0 & \text{用户好友 } u_i \text{ 没有分享过该媒体信息} \\ 1 & \text{用户好友 } u_i \text{ 分享过该媒体信息} \end{cases}.$$

构建用户好友在话题上对用户网络行为的影响因子矩阵为

$$L = \begin{bmatrix} l_{0,0} & \cdots & l_{0,M-1} \\ \vdots & \ddots & \vdots \\ l_{N-1,0} & \cdots & l_{N-1,M-1} \end{bmatrix}$$

其中定义 l_{ij} 为用户好友列表中第 i 个好友 u_i 在第 j 个话题 t_j 上对用户的网络行为产生的影响因子，其数值由前面所述的影响力模型决定，即 l_{ij} = infl_on_t_j(u_i → user)。

本实验中采用的是使用用户偏好的相似度来描述用户之间的影响力，因此有 l_{ij} = infl_on_t_j(u_i → user) = simi_on_t_j(u_i, user) = u_i(j) · user(j)。

为了区分两种网络行为的触发情况对特定用户的影响程度的不同，定义 α 为用户的网络行为受到网络结构（好友关系）的影响程度，β 为用户的网络行为受到媒体信息内容的影响程度。

4.4.2 兴趣度模型的建立

对于用户分享行为触发原因的两种情况，讨论起对应的影响力计算方法。

第一种，用户看到其好友分享该媒体信息的消息，并且对该信息有兴趣，因而产生分享行为。社交网络中在该媒体信息被分享的新鲜事里，用户可以看到这条信息被哪些好友所分享。因此我们可以假设好友的分享与否和他是否对该用户产生影响是紧密相关的。

在某时刻，用户好友对媒体信息的分享消息对触发用户分享行为的影响力为

$$\begin{aligned} \text{pref}_{\text{friend}} &= F \cdot L \cdot S^T \\ &= [f_0 \ \dots \ f_{N-1}] \begin{bmatrix} l_{0,0} & \cdots & l_{0,M-1} \\ \vdots & \ddots & \vdots \\ l_{N-1,0} & \cdots & l_{N-1,M-1} \end{bmatrix} [s_0 \ \dots \ s_{M-1}]^T \quad (4-7) \\ &= \sum_{j=0}^{M-1} s_j \left(\sum_{i=0}^{N-1} f_i \cdot l_{i,j} \right) = \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} f_i \cdot l_{i,j} \cdot s_j = \sum_{i,j} (f_i s_j) \cdot l_{i,j} \end{aligned}$$

第二种，用户主动观看到某条媒体信息，由于他对其有兴趣而产生了分享行为。用户对媒体信息的偏好相似度对触发用户分享行为的影响力为

$$\begin{aligned} \text{pref}_{\text{media}} &= P \cdot S^T \\ &= [p_0 \ \dots \ p_{M-1}] [s_0 \ \dots \ s_{M-1}]^T \quad (4-8) \\ &= \sum_{j=0}^{M-1} p_j \cdot s_j \end{aligned}$$

综上，用户对某媒体信息的兴趣度为

$$\text{pref} = \begin{cases} \alpha \cdot \text{pref}_{\text{friend}} & \text{if } \text{pref}_{\text{friend}} > 0 \\ \beta \cdot \text{pref}_{\text{media}} & \text{if } \text{pref}_{\text{friend}} = 0 \end{cases} \quad (4-9)$$

设定阈值 VALUE 为某一个合适的数，当兴趣度达到阈值时，用户对该媒体信息发生网络行为（分享之），即 $\text{pref} \geq \text{VALUE}$ 。这一处理方法建立在假设同一用户的网络行为在兴趣度方面等价。实际上，VALUE 值与用户的活跃程度紧密相

关，例如，有些用户的行为习惯是只看帖（媒体信息），不进行分享，也就是说网络行为方面具有惰性。

用户好友在话题上对用户网络行为的影响因子矩阵在本实验中以用户与其好友的偏好相似度来决定。如果用户的偏好向量为 $P = [p_0 \dots p_{M-1}]$ ，用户的好友 u_i 的偏好向量为的 $P_i = [p_{i,0} \dots p_{i,M-1}]$ ，由此构造的影响因子矩阵为

$$L = \begin{bmatrix} l_{0,0} & \dots & l_{0,M-1} \\ \vdots & \ddots & \vdots \\ l_{N-1,0} & \dots & l_{N-1,M-1} \end{bmatrix} = \begin{bmatrix} p_0 \cdot p_{0,0} & \dots & p_{M-1} \cdot p_{0,M-1} \\ \vdots & \ddots & \vdots \\ p_0 \cdot p_{N-1,0} & \dots & p_{M-1} \cdot p_{N-1,M-1} \end{bmatrix}, \text{ 其中用户好友 } u_i \text{ 在话题 } t_j \text{ 上对用户的网络行为产生的影响因子 } l_{i,j} = p_j \cdot p_{i,j}.$$

综上可以总结得出用户对某媒体信息的兴趣度公式为

$$\text{pref} = \begin{cases} \alpha \cdot \text{pref}_{\text{friend}} = \alpha \cdot (\sum_{j=0}^{M-1} \sum_{i=0}^{N-1} f_i \cdot p_j \cdot p_{i,j} \cdot s_j) & \text{if } \text{pref}_{\text{friend}} > 0 \\ \beta \cdot \text{pref}_{\text{media}} = \beta \cdot (\sum_{j=0}^{M-1} p_j \cdot s_j) & \text{if } \text{pref}_{\text{friend}} = 0 \end{cases} \quad (4-10)$$

用户兴趣度计算算法的主要框架（表 4.2）如下所示：

表4.2 用户兴趣度计算算法

算法 用户兴趣度计算算法

1: 输入：用户及好友关系列表，分享记录索引库，给定时刻 t 、用户 $user$ 、媒体信息 $media$
2: 输出：在给定时刻 t ，给定用户 $user$ 对给定媒体信息 $media$ 的兴趣度 $pref$
3: 初始化社交用户网络结构图
遍历用户及好友关系列表：更新用户的人人 id，姓名，好友关系
遍历分享记录索引库：更新用户的分享记录列表，包含分享记录对应的话题向量
初始化用户的偏好向量为空数组
4: 遍历用户网络结构图中每个用户：对所有分享记录的话题向量平均得到偏好向量
5: 获取给定媒体信息的话题向量 $s[j]$ ；获取给定用户的偏好向量 $p[j]$ 及其好友 $u[i]$ 的偏好向量 $p[i,j]$ ；初始化好友分享向量 $f[i]$ ，遍历用户的每个好友 $u[i]$ ，判断其分享记录列表中是否存在时刻 t 之前分享的给定媒体信息 $media$
6: 用户受好友影响产生的兴趣度 $pref_friend \leftarrow 0$ ，
媒体信息内容产生的兴趣度 $pref_media \leftarrow 0$
7: for $i=0$ to $N-1$
8: for $j=0$ to $M-1$
9: $pref_friend \leftarrow pref_friend + f[i] * p[j] * p[i,j] * s[j]$
10: end
11: end
12: for $j=0$ to $M-1$
13: $pref_media \leftarrow pref_media + p[j] * s[j]$
14: end
15: if $pref_friend > 0$
16: 用户对媒体信息的兴趣度 $pref = alpha * pref_friend$
17: else
18: $pref = beta * pref_media$
19: end

其中 $i \in [0, N)$, $j \in [0, M)$, N 为给定用户的好友数量, M 为话题数量

4.5 用户行为预测及验证

4.5.1 用户行为预测简介和意义

尽管我们从常识上明白人们的行爲是带有很强的随机性的，但其中依旧存在可以发觉的模式和特征。用户的网络行爲与人类日常生活的行为一样，一旦养成了某种习惯，就很难被改变，而且众所周知，人类的生活习性具有一定的普遍性，网络行爲也是如此。我们从观察者的角度来分析用户的行为，合理地归纳总结出习惯的表现形式，继而可以对用户的下一步网络行爲做出较为正确地预测，这对以最大程度满足用户需求的社交网络系统来说是一种莫大的帮助。

如果社交网络系统能够自行有效地预测用户的行为方向，智能地选择、提供用户感兴趣的信息，则可以有效地提高用户使用社交网络的效率。相关的研究成果可以用来解决很多问题，如在线推荐，减少网络延时，改善网站结构等等。另一个角度来看，了解网络用户的当前行为并预测用户行为可以及时掌握信息在社交网络上的传播速率、途径和方向以及这一类信息的特征，从而可以采取有效措施引导信息传播、监督管理网络秩序和维护网络安全。

4.5.2 用户行为预测的验证方法

我们所获取的媒体数据（分享记录）的时间跨度是从 2007 年 2 月 1 日至 2009 年 12 月 20 日。这样的时间条件允许我们采用“时间窗”的方式来预测用户的分享行为，从而验证用户兴趣度模型的合理性。

首先，明确验证过程中时间窗的涵义，从而明确验证步骤。

训练时间：利用处于这段时间内社交网络中用户的分享记录来训练得到用户的偏好模型，如 2009 年 3 月 1 日至 9 月 1 日。

测试时间：收集在这段时间内的用户好友及其本身分享的媒体信息，对该数据集中每条媒体信息逐一计算得到用户对其的兴趣度，如 2009 年 9 月 1 日至 9 月 30 日。

验证时间：验证用户是否在后续时间里分享过该媒体信息，同时也可以得到分享该信息的具体时间，如 2009 年 9 月 1 日至 12 月 20 日。

时间窗将数据处理过程划分成三个步骤，如下图 4.4 所示。



图4.4 验证过程的示意图

第二，明确兴趣度的计算方法。

用户兴趣度公式中可以根据既定数据准确计算得到的是 $\text{pref}_{\text{friend}} = \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} f_i \cdot p_j \cdot p_{i,j} \cdot s_j$ 和 $\text{pref}_{\text{media}} = \sum_{j=0}^{M-1} p_j \cdot s_j$ 。这里需要确定描述用户兴趣度受好友影响和媒体信息内容影响的因子 α 与 β ，以得到准确的 pref 公式。在这一假设的基础上，我们可以定义用户兴趣度公式为 $\text{pref} = \gamma * \text{pref}_{\text{friend}} + \text{pref}_{\text{media}}$ ，其中 $\gamma = \alpha / \beta$ 。这里， γ 越大，则用户兴趣度受好友影响程度与媒体信息内容本身相比较越强。

本实验中，我们会将融合了用户影响力和行为偏好的模型与单纯基于内容的兴趣度模型在行为预测上的效果。基于内容的兴趣度模型实际上就是不考虑 $\text{pref}_{\text{friend}}$ ，即定义 $\gamma = 0$

另外，我们在预测和验证阶段都需要对每条媒体信息是否被用户或用户好友实际分享进行标注，标注方法是用 0 或 1 来表示，0 表示没有分享过，1 表示分享过。

第三，确定验证指标。

测试数据集为在测试时间段内，用户及其好友所分享的所有媒体信息，用 T 来表示。按照上述的计算方法得到的用户对测试数据集中每条媒体信息的兴趣度 pref 由大到小，对 T 中数据进行排序。用户实际分享数据集 S 含有且仅含有测试数据集 T 中用户在验证时间段里实际分享的所有媒体信息。

我们设定一个可调参数 n 和待求参数 x，表示实际分享数据集 S 中的前 $x \cdot |S|$ 条媒体信息全部位于测试数据集 T 中的前 $n \cdot |T|$ 条。不妨使得 n 为 0.1 至 1.0 不等，每隔 0.1 计算出对应的 x 数值。倘若兴趣度模型合理，则必有 $x > n$ ，且 x 的数值越大越适宜；容易知道，x 随着 n 的增大而增大，并且增大幅度随着 n 的增大而逐步减小。

n 值的涵义是实际推荐给用户的信息量占用户可接触信息量的百分比，也就是说如果希望用户可以从较少的推荐信息中找到自己会去分享的内容，n 值就会

较小。一般的，系统实际给用推荐的信息数量应该满足 n 值较小，例如 10%、20% 及 30%，我们关心这时的 x 数值大小。

信息检索系统性能的两个基本客观指标是召回率（Recall Rate）和准确率（Precision Rate）。

$$\text{召回率 (recall)} = \frac{\text{检索到的相关文档}}{\text{数据库中所有的相关文档}}$$

$$\text{准确率 (precision)} = \frac{\text{检索到的相关文档}}{\text{所有被检索到的文档}}$$

召回率越大越好，因为我们希望数据库中相关的文档，被检索到的越多越好；准确率越大越好，因为我们希望检索到的文档中，相关的越多越好，不相关的越少越好。这两者之间虽然没有必然的关系，然而在大规模数据集合中，这两个指标却是相互制约的。

这样的指标在推荐系统中也经常被采纳，这里定义的召回率和准确率为：

$$\text{召回率 (recall)} = \frac{\text{推荐给用户且用户分享的媒体信息数量}}{\text{用户分享的媒体信息数量}}$$

$$\text{准确率 (precision)} = \frac{\text{推荐给用户且用户分享的媒体信息数量}}{\text{推荐给用户的媒体信息数量}}$$

在本实验中，召回率（recall）和准确率（precision）的公式为：

$$\text{recall} = \frac{x \cdot |S|}{|S|} = x \quad (4-11)$$

$$\text{precision} = \frac{x \cdot |S|}{n \cdot |T|} \quad (4-12)$$

在实验验证过程中，我们主要考察如下情形和规律：

- (1) 在 n 为 10%、20% 和 30% 的情况下，调整 γ 值，使得对于社区中用户群来说，平均召回率和平均准确率最高；

- (2) 将融合影响力和行为偏好的模型与基于内容模型 ($\gamma = 0$) 进行比较召回率和准确率;
- (3) 召回率和准确率随着 n 值变化的变化曲线: 考察数值大小与变化以反映算法的效果, 增幅随着 n 值增大的变化;
- (4) 考察训练数据的时间长度以及数据的新鲜度对召回率和准确率的影响。

4.5.3 用户行为预测的验证结果和结论

用户选择。实验准备数据中, 可以用来进行行为预测的社交网络用户必须满足这样两个条件: 一是拥有其完整的好友关系数据; 二是拥有其完整的分享记录列表。符合这样要求的用户数量是 5141 个, 我们从中随机挑选 400 个用户进行行为预测。

时间窗选择。实验中训练时间的长短设置为 2 年, 6 个月和 3 个月。测试时间为 2009 年 7 月、8 月或 9 月。验证时间为从测试时间起至最终。

首先, 绘制召回率和准确率随着 n 值变化的变化曲线, 考察数值大小与变化以反映算法的效果, 增幅随着 n 值增大的变化。

训练数据的时间长度为 3 个月, 对测试时间段为 2009 年 9 月的数据, 选择使平均召回率达到最大值的 γ 值, 下表 (表 4.3) 反映了召回率、准确率随着 n 值变化的情况。

表4.3 召回率、准确率随着n值的变化情况

n	召回率 recall	准确率 precision
0.1	0.36290	0.049632
0.2	0.49858	0.034049
0.3	0.57522	0.026706
0.4	0.64209	0.022726
0.5	0.71486	0.020660
0.6	0.78846	0.019483
0.7	0.84704	0.018198
0.8	0.91752	0.018322
0.9	0.97215	0.017514

用散点图 (图 4.5) 来表示这样变化规律如下:

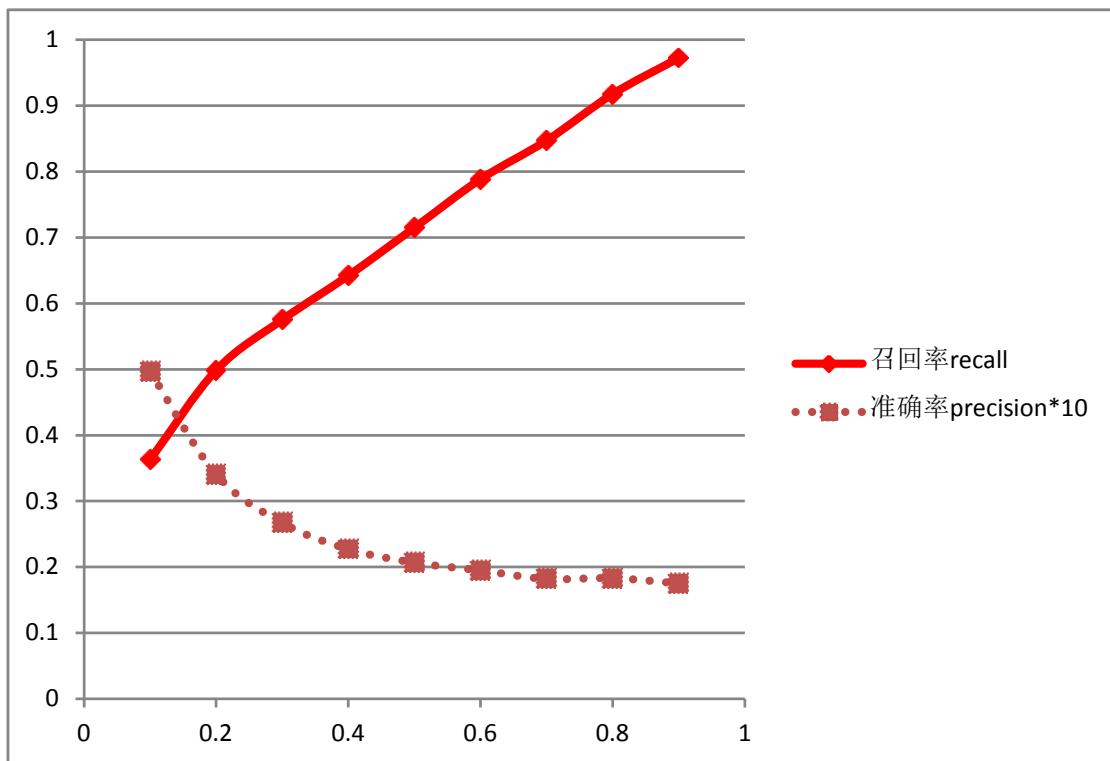


图4.5 召回率和准确率随n值的变化曲线

我们可以从这一实验结论中看到这样的特征规律：

规律 I：对于不同的 n 值，召回率都大于 n，且随着 n 值增大而增大；准确率随着 n 值增大而减小。

召回率大于 n，这从一方面证明了融合影响力和用户偏好的模型在用户行为预测方面是起到较强的积极作用的，比起随机推荐的方法，在推荐信息量适合的条件下，召回率提高两倍到三倍。准确率取值在 1% 到 5% 之间，这体现出用户在社交网络上对推荐信息采取分享行为的可能性。采取一次分享行为，用户需要完成浏览、点击、观赏、评价、思考是否推荐给好友（分享行为的本因之一）等心理活动和行为活动，可以发现人人网用户在分享行为上活跃度并不高。

召回率随着 n 值增大而增大的原因在于推荐信息量的增大，保证了用户能更多的接受到他可能采取分享行为的内容；准确率随着 n 值增大而减小的原因在于计算出的用户对推荐给他的信息的兴趣度越大，用户的行为随机性越小，系统的误差也就越小，推荐更为准确。

规律 II：随着 n 值的增大，召回率的增大幅度逐渐变小，准确率的减小幅度也逐渐变小。

这样的变化规律体现了推荐功能的本质。在实际的推荐系统中，推荐给用户的信息量一定是有限的，否则既造成信息冗余，同时也给用户带来不便。当推荐信息量较小时，推荐效果较好，这是系统和用户都乐于见到的。

第二，在 n 为 10%、20% 和 30% 的情况下，调整 γ 值，使得对于社区中用户群来说，平均召回率和平均准确率最高。

训练数据的时间长度为 3 个月，分别对测试时间段为 2009 年 7 月、8 月和 9 月的数据，调整 γ 值，选择指标值达到最大时的 γ 最小值（表 4.4）。

表4.4 三个测试时间段下召回率最大时的 γ 值及结果

2009 年 7 月			2009 年 8 月			2009 年 9 月			
n	γ	recall	precision	γ	recall	precision	γ	recall	precision
0.1	4.623	0.3611	0.03794	6.128	0.3282	0.03742	6.049	0.3652	0.04985
0.2	1.252	0.4865	0.02917	3.099	0.4475	0.02923	3.990	0.4987	0.03406
0.3	1.791	0.5645	0.02355	4.128	0.5257	0.02409	3.198	0.5755	0.02671

我们可以从这一实验结论中看到这样的特征规律：

规律 III： γ 的取值范围在 [1, 10]。 γ 的意思是用户兴趣度受好友影响程度与媒体信息内容本身影响程度之比；当 $\gamma = 0$ 时，兴趣度仅受媒体信息内容的影响；当 γ 趋近于无穷大时，兴趣度仅受用户好友的影响。由此可以发现，用户的兴趣度是同时受到其好友和媒体信息内容影响的，并且受到用户好友影响的程度更多一些。

规律 IV：当 n 值为 0.1 时， γ 的取值相对于 n 为 0.2 和 0.3 时要大很多。当 n 值较小时，推荐信息量较少，推荐准确率较高，推荐效果较好，用户分享行为更容易受到好友的影响。这与社交网络中的实际情况很符合。用户所接触的网络信息大多是从好友新鲜事里获得，然后去关注和分享。

第三，将融合影响力和行为偏好的模型分别与基于内容的模型和协同过滤模型进行召回率和准确率的比较。

基于内容的模型在本实验中的表现形式就是调整参数使得 $\gamma = 0$ ，这种模型可以应用在完整数据集上，因为对于数据集中每一条媒体信息都可以进行语义内容分析。

协同过滤模型尽可以在存在着其他用户对给定媒体信息有评价或者发生网络行为的条件下适用。因此在本实验中，对数据集中不含用户影响力部分的媒体信息无法进行合理的行为预测。在过滤掉这部分媒体信息后，我们可以应用协同过滤模型合理预测用户对剩余媒体信息的兴趣度，并进行预测。

训练数据的时间长度为3个月，对测试时间段为2009年9月的数据，调整 γ 值，选择指标值达到最高值时的 γ 最小值（表4.5）。

表4.5 本方法与基于内容和协同过滤方法的对比结果

model	content-based: $\gamma=0$		social (after filter)			best	
	n	recall	precision	recall	precision	γ	recall
0.1	0.30482	0.044408	0.16703	0.17108	6.0490	0.36522	0.049849
0.2	0.47960	0.032852	0.31322	0.18109	3.9898	0.49865	0.034055
0.3	0.56720	0.026382	0.46594	0.19157	3.1978	0.57551	0.026709

融合影响力和用户偏好的行为预测方法带来的效果，与单纯基于内容的和协同过滤的方法进行比较，如下表（表4.6）。

表4.6 本方法与基于内容和协同过滤方法的效果提高程度

提高程度	召回率（recall）提高程度		准确率（precision）提高程度
	n	比基于内容高	比协同过滤高
0.1	19.82%	118.7%	12.25%
0.2	3.972%	59.20%	3.662%
0.3	1.465%	23.52%	1.240%

我们可以从这一实验结论中看到这样的规律V：实验中采用的行为预测方法在召回率比起基于内容和协同过滤的方法都有提高，并且由于协同过滤方法不考虑内容特征，在推荐信息量较少时会损失掉大量有用的媒体信息。但是，协同过滤方法的优势在于准确率非常高，因为用户在社交网络中所接触到的信息更多的是来自好友的新鲜事。随着推荐信息量的增大，本方法相对于基于内容方法在召回率和准确率上，以及相对于协同过滤方法在召回率上的优势渐渐变小。

第四，考察训练数据的时间长度以及数据的新鲜度对召回率和准确率的影响。

采用距离测试时间不同时间长度的数据进行训练，也就是观察用户最近两年、6个月和3个月的网络行为来做出预测，我们仅选用n为30%时的结果，如下表所示（表4.7）。

表4.7 不同的训练时长对测试结果的影响

训练时长 测试时间	两年	6个月	3个月
2009年7月	0.542143495	0.54240407	0.560653035
2009年8月	0.475611917	0.514034344	0.525869094
2009年9月	0.477536885	0.5670612	0.57548834

在训练时间长度为相同的3个月（6个月）、测试时间为2009年8月和9月的情况下，将距离测试时间9个月之久开始算起的3个月（6个月）和就在测试时间之前的3个月（6个月）里媒体信息分别作为训练数据，这样预测的用户网络行为的结果如下（表4.8）。

表4.8 不同的训练时间段对测试结果的影响

测试时间	2009 年 8 月			
训练时间	2008 年 11 月至 2009 年 1 月		2009 年 5 月至 7 月	
n	recall	precision	recall	precision
0.2	0.45443	0.030395	0.44752	0.029232
0.3	0.52394	0.025050	0.52573	0.024086
测试时间	2009 年 9 月			
训练时间	2008 年 12 月至 2009 年 2 月		2009 年 6 月至 8 月	
n	recall	precision	recall	precision
0.2	0.49370	0.034602	0.49865	0.034055
0.3	0.56970	0.026985	0.57551	0.026709
测试时间	2009 年 8 月			
训练时间	2008 年 8 月至 2009 年 1 月		2009 年 2 月至 7 月	
n	recall	precision	recall	precision
0.2	0.42756	0.029756	0.43304	0.028045
0.3	0.50675	0.024793	0.51374	0.023618
测试时间	2009 年 9 月			
训练时间	2008 年 9 月至 2009 年 2 月		2009 年 3 月至 8 月	
n	recall	precision	recall	precision
0.2	0.46351	0.033322	0.47658	0.032824
0.3	0.54665	0.026311	0.56692	0.026597

我们可以从这一实验结论中从两个角度都体现了这样的规律 VI：当训练数据越新鲜，越接近测试时间，召回率和准确率就越高。究其原因是在于用户的网络行为偏好是存在时间演变的特征的，如果依照用户近期的分享内容进行分析会更为准确，但这种提高训练数据新鲜度的方法在提高预测效果方面是存在瓶颈的。当训练数据过于新鲜、时间跨度过短，随机误差就越明显，因此提高程度会逐渐减小。

4.6 媒体数据的个性化推荐

4.6.1 媒体数据个性化推荐的简介

推荐是日常生活以及网络生活中人们常常获取到的服务，这种服务可以来自他人，也可以是某些掌握了大量知识的系统所提供的。我们常常依赖于一些所知所闻来挑选自己感兴趣的的商品或是做出决定。例如，当我们在买一张 CD 时，我们会听从在音乐上有同样品味的人的见解。还有的时候，我们的购买评价来自于产品本身的特点以及我们自己的兴趣偏好。所以，一个能够为用户提供很好的推荐服务的系统，必须要在构建模型时尽可能多地考虑到各方面的因素。在这个问题上，已经存在着一些常见的方法。第一种方法是系统根据所提供的它的其他用户对商品的评价，来猜测用户会喜欢什么样的商品，这是一种协同过滤的方法。第二种方法是系统以用户偏好为基础，通过归纳学习来猜测用户会喜欢什么样的商品。这种方法不依赖与其他用户，所以称之为基于内容的推荐方法。两种情况下，目标都是在于去发现一种能够通过对所有用户和商品进行学习后，预测用户对商品的兴趣度以及评价的函数。我们希望在对社交网络的用户行为预测的基础上，实现为特定用户，根据其个性和品位，以及周围人群对信息的接收情况，来推荐最为适合其浏览的媒体数据。我们的系统在处理数据和计算方法上会同时考虑到协同过滤特征和内容特征^[19]，并且将社交网络本身的网络结构特性融合进去，能够提高推荐的效果。

4.6.2 媒体数据个性化推荐的方法

我们之前的工作是通过构建媒体数据的话题模型、用户的偏好模型、用户之间的影响力模型，以及从兴趣度模型的角度诠释用户的预测行为。因此我们可以根据融合了用户影响力和行为偏好的模型计算出的兴趣度值，产生的对用户推荐数值较高的媒体内容的推荐方法是比起协同过滤以及基于内容的方法更为优越的。

具体的媒体数据个性化推荐方法按照以下几个步骤进行：

- (1) 收集在某一时刻用户所能接触到的媒体信息集合。信息来源包括用户好友所传播的信息，用户好友所产生的新鲜媒体内容，以及从用户及其好友圈子以外传播的热门信息；
- (2) 计算媒体信息集合中媒体数据的话题模型，用户个人的偏好模型，以及用户与其好友之间的影响力模型；

- (3) 采用用户对媒体信息的兴趣度算法，计算用户对媒体信息集合中的每一条信息的兴趣度；
- (4) 将媒体信息集合中内容按照兴趣度从大到小排序，选取排在若干条的内容作为推荐列表返回给用户。

第5章 总结展望

5.1 工作总结

在这篇文章里，提出了一种实现社会媒体网络中用户行为预测的归纳学习方法。这种方法是通过在大量的现实社交网络数据上的实现得到了验证。这种预测方法的基础是为媒体内容建立合理的话题模型，得到用户的偏好模型和影响力模型，从而计算出用户对媒体内容的兴趣度大小。这种方法的特点在于它融合了社交网络的结构特征，也就是用户之间的影响力，以及从媒体内容的语义信息总结得到的用户行为偏好的特征。

这种用户行为预测的方法相对于其他的协同过滤和基于内容的方法更为灵活，它能够将这两方面的特征融入到模型中进行合理的处理。我们将社交数据划分为训练数据与测试数据两部分，来校验这种方法的合理性和优越性。在这种用户行为预测方法的基础上，进一步提出了对特定用户进行个性化推荐媒体内容的方法。

5.2 未来研究工作

在未来的研究工作中，我们可以在这种预测方法的基础上，从三个角度来进一步提升用户行为预测的效果。

首先是改进媒体信息的话题模型。当前的话题模型是建立在媒体信息的上下文文本信息，即标题文本的语义分析与聚类的结果上。提升的方法是分情况讨论不同类型的媒体信息的处理方式。例如，对于日志，我们需要分析日志的内容和评论；对于照片和相册，我们需要分析其中含有的图像信息；对于视频，我们需要从视频内容的角度运用计算机视觉方面的知识进行分析等。在这种改进方法的基础上，我们可以更好地理解媒体信息内容，获得更为准确的话题模型。

第二是改进用户的偏好模型。当前的用户偏好模型仅仅是从用户所分享的媒体信息的语义内容总结得出的。实际上，用户的偏好体现在诸多方面，包括用户创造的媒体信息内容，用户的实际浏览内容，用户对媒体信息给予的评价等，其中更为准确的是用户为自己所标注的兴趣爱好。如果能掌握这些信息，偏好模型会更为准确。

第三是改进用户的影响力模型。前面我们提出了三种建立用户影响力模型的方法，包括用户网络行为偏好的相似度、用户交流频度和用户网络结构的亲密程

度。这里我们由于受到数据源的限制，仅仅采用了第一种方法，即讨论用户网络行为偏好的相似度建立了影响力模型。实质上，倘若能融合后两种方法来计算影响力，用户行为预测的结果会更有准确。

这三种改进方法能提升用户对媒体内容的兴趣度计算的效果，更好地进行用户行为预测，同样地，也可以提升对给定用户进行个性化推荐媒体信息的效果。我们如果能够在这三个方向对这个工作进行很好的改进，社会媒体网络系统在采纳了这项成果之后，可以更好地为用户提供服务，方便其在因特网上的社交生活。

插图索引

图 2.1 长尾效应的曲线图	14
图 2.2 分享者较多的媒体数据占其总量的小部分	15
图 2.3 分享内容较多的用户占用户总量的小部分	16
图 2.4 用户分享数量随其好友数量变化的曲线	17
图 2.5 用户好友数量随其分享数量变化的曲线	18
图 3.1 运行 lda.exe 时命令行参数提示	23
图 4.1 用户 1 的社区网络结构图	37
图 4.2 用户 2 的社区网络结构图	37
图 4.3 用户 3 的社区网络结构图	38
图 4.4 验证过程的示意图	49
图 4.5 召回率和准确率随 n 值的变化曲线	52

表格索引

表 2.1 结构化结点信息	12
表 2.2 结构化分享信息格式	13
表 2.3 三类媒体信息的分享次数	19
表 3.1 标题内容索引库示例	21
表 3.2 标题内容分词结果示例	22
表 3.3 LDA 调整参数话题聚类	24
表 3.4 词汇在 10 个话题中部分分类结果 I	25
表 3.5 语义归纳的话题特征	26
表 3.6 词汇在 10 个话题中部分分类结果 II	26
表 3.7 标题内容在话题上的概率分布	27
表 3.8 分享记录样例在话题上分布	31
表 3.9 实验分析考察用户关心的话题及分布值	33
表 3.10 调查问卷表格	34
表 3.11 用户调查结果统计	35
表 4.1 三种用户影响力相关因素的讨论	43
表 4.2 用户兴趣度计算算法	47
表 4.3 召回率、准确率随着 n 值的变化情况	51
表 4.4 三个测试时间段下召回率最大时的 γ 值及结果	53
表 4.5 本方法与基于内容和协同过滤方法的对比结果	54
表 4.6 本方法与基于内容和协同过滤方法的效果提高程度	54
表 4.7 不同的训练时长对测试结果的影响	55

表 4.8 不同的训练时间段对测试结果的影响 56

参考文献

- [1] Hofmann T. Probabilistic latent semantic indexing. Proceedings of SIGIR'99: the 22nd International Conference on Research and Development in Information Retrieval, 1999. 50-57.
- [2] Blei D, Ng A, Jordan M. Latent dirichlet allocation. Journal of Machine Learning Research, 2003, 3: 993-1022.
- [3] Griffiths T. L, Steyvers M. Finding scientific topics. Proceedings of the National Academy of Sciences, 2004, 101(suppl. 1):5228–5235.
- [4] Blei D. M, McAuliffe J. D. Supervised topic models. Proceedings of NIPS'07: Advances in Neural Information Processing Systems, 2007. 248–256.
- [5] Teh Y, Jordan M, Beal M, et al. Hierarchical dirichlet processes. The Journal of the Acoustical Society of America, 2006, 101(476):1566–1581.
- [6] Blei D. M, Lafferty J. D. A correlated topic model of science. The Annal of Applied Statistics, 2007, 1(1):17–35.
- [7] Blei D. M, Lafferty J. D. Dynamic topic models. Proceedings of ICML'06: the 23th International Conference on Machine Learning, 2006. 113–120.
- [8] Milgram S. The small world problem. Psychology Today, 1967, 2:60–67.
- [9] Kleinberg J. The small-world phenomenon: an algorithm perspective. Proceedings of STOC'00: the thirty-second annual ACM symposium on Theory of computing, New York, NY, USA: ACM, 2000. 163–170.
- [10] Watts D. J, Strogatz S. H. Collective dynamics of ‘small-world’ networks. Nature, 1998, 393(6684):440–442.
- [11] Mei Q, Cai D, Zhang D, et al. Topic modeling with network regularization. Proceedings of WWW'06=8: the 17th international conference on World Wide Web, 2008. 101–110.
- [12] Sun Y, Han J, Gao J, et al. iTopicModel: information network-integrated topic modeling. Proceedings of ICDM'09: IEEE International Conference on Data Mining, 2009. 493-502.
- [13] Chang J, Boyd-Graber J, Blei D. M. Connections between the lines: augmenting social networks with text. Proceedings of KDD '09: the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009. 169–178.
- [14] Tang J, Sun J, Wang C, et al. Social influence analysis in large-scale networks. Proceedings of KDD '09: the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009. 807–816.

- [15] Balabanovic M, Shoham Y. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 1997, 40(3):66–72.
- [16] Koren Y. Collaborative filtering with temporal dynamics. *Proceedings of KDD'09: the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008. 531–538.
- [17] JamaliM, EsterM. TrustWalker: a randomwalk model for combining trust-based and itembased recommendation. *Proceedings of KDD '09: the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009. 397–406.
- [18] Wang J, Vries A. P, Reinders M. J. T. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. *Proceedings of SIGIR '06: the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA: ACM, 2006. 501–508.
- [19] Chumki Basu, Haym Hirsh, William Cohen: Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 1998. 714-720.

致 谢

我谨向我的指导老师杨士强教授致以衷心的谢意！杨老师给予我悉心关怀和谆谆教诲，使我顺利完成这一研究工作。

感谢孙立峰副教授对我在理论知识和研究方向上的细心指导，使我在较短时间里确认了研究工作的目标。

感谢崔鹏师兄和刘璐师姐对我在大到研究思路和分析的角度，小到基本理论的知识细节，都给了我很多宝贵的意见和建议。

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，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名: 蒋琳 日 期: 2010.6.29

附录 A 外文资料的书面翻译

书面翻译题目

写出至少 5000 外文印刷字符的调研阅读报告或者书面翻译 1-2 篇(不少于 2 万外文印刷符)。

书面翻译对应的原文索引

Chumki Basu, Haym Hirsh, William Cohen: Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, 1998. 714-720.

Recommendation as Classification: Using Social and Content-Based Information in Recommendation

Chumki Basu*

Bell Communications Research
445 South Street
Morristown, NJ 07960-6438
cbasu@bellcore.com

Haym Hirsh

Department of Computer Science
Rutgers University
Piscataway, NJ 08855
hirsh@cs.rutgers.edu

William Cohen

AT&T Laboratories
180 Park Ave, Room A207
Florham Park, NJ 07932
wcohen@research.att.com

Abstract

Recommendation systems make suggestions about artifacts to a user. For instance, they may predict whether a user would be interested in seeing a particular movie. Social recommendation methods collect ratings of artifacts from many individuals, and use nearest-neighbor techniques to make recommendations to a user concerning new artifacts. However, these methods do not use the significant amount of other information that is often available about the nature of each artifact — such as cast lists or movie reviews, for example. This paper presents an inductive learning approach to recommendation that is able to use both ratings information and other forms of information about each artifact in predicting user preferences. We show that our method outperforms an existing social-filtering method in the domain of movie recommendations on a dataset of more than 45,000 movie ratings collected from a community of over 250 users.

Introduction

Recommendations are a part of everyday life. We usually rely on some external knowledge to make informed decisions about an artifact of interest or a course of action, for instance when we are going to see a movie or going to see a doctor. This knowledge can be derived from social processes. When we are buying a CD, we can rely on the judgment of a person who shares similar tastes in music. At other times, our judgments may be based on available information about the artifact itself and our known preferences. There are many factors which may influence a person in making these choices, and ideally one would like to model as many of these factors as possible in a recommendation system.

There are some general approaches to this problem. In one approach, the user of the system provides ratings of some artifacts or items and the system makes informed guesses about what other items the

user may like. It bases these decisions on the ratings other users have provided. This is the framework for *social-filtering* methods (Hill, Stead, Rosenstein & Furnas 1995; Shardanand & Maes 1995). In a second approach, the system accepts information describing the nature of an item, and based on a sample of the user's preferences, learns to predict which items the user will like (Lang 1995; Pazzani, Muramatsu, & Billsus 1996). We will call this approach *content-based filtering*, as it does not rely on social information (in the form of other user's ratings). Both social and content-based filtering can be cast as learning problems: in both cases, the objective is to learn a function that can take a description of a user and an artifact and predict the user's preferences concerning the artifact.

Well-known recommendation systems like *Recommender* (Hill, Stead, Rosenstein & Furnas 1995) and *Firefly* (<http://www.firefly.net>) (Shardanand & Maes 1995) are based on social-filtering principles. *Recommender*, the baseline system used in the work reported here, recommends as yet unseen movies to a user based on his prior ratings of movies and their similarity to the ratings of other users. Social-filtering systems perform well using only numeric assessments of worth, i.e., ratings. However, there is often readily available information concerning the content of each artifact. Social-filtering methods leave open the question of what role content can play in the recommendation process.

For many types of artifacts, there is already a substantial store of information that is becoming more and more readily accessible while at the same time growing at a healthy rate. Let's take, for instance, a sample of the information a person can obtain about a favorite movie on the Web alone: a complete breakdown of cast/crew, plot, movie production details, reviews, trailer, film and audio clips, (and ratings too) and the list goes on. When users decide on a movie to see, they are likely to be influenced by data provided by one or more of these sources. Social-filtering may be characterized as a generic approach, unbiased by the regularities exhibited by properties associated with the items of interest (Hill, Stead, Rosenstein & Furnas 1995). (Indeed, a significant motivation for some of the work on such systems is to explore the utility of recognizing communities of users based solely on similarities in their preferences.) However, the fact that

*Department of Computer Science, Rutgers University, Piscataway, NJ 08855

We would like to thank Susan Dumais for useful discussions during the early stages of this work.

Copyright ©1998, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

content-based properties can be identified at low cost (with no additional user effort and that people are influenced by these regularities make a compelling reason to investigate how best to use them.

In what situations are ratings alone insufficient? Social-filtering makes sense when there are enough other users known to the system with overlapping characteristics. Typically, the requirement for overlap in most of these systems is that the users of the system rate the same items in order to be judged similar/dissimilar to each other. It is dependent upon the current state of the system — the number of users and the number and selection of movies that have been rated.

As an example of the limitations of using ratings alone, consider the case of an artifact for which no ratings are available, such as when a new movie comes out. Since there will be a period of time when a recommendation system will have little ratings data for this movie, the recommendation system will initially not be able to recommend this movie reliably. However, a system which makes use of content might be able to make predictions for this movie even in the absence of ratings.

In this paper, we present a new, inductive learning approach to recommendation. We show how pure social-filtering can be accomplished using this approach, how the naive introduction of content-based information does not help — and indeed harms — the recommendation process, and finally, how the use of hybrid features that combine elements of social and content-based information makes it possible to achieve more accurate recommendations. We use the problem of movie recommendation as our exploratory domain for this work since it provides a domain with a large amount of data (over 45,000 movie evaluations across more than 250 people), as well as a baseline social-filtering method to which we can compare our results (Hill, Stead, Rosenstein & Furnas 1995).

The Movie Recommendation Problem

As noted above, in the social filtering approach, a recommendation system is given as input a set of ratings of specific artifacts for a particular user. In recommending movies, for instance, this input would be a set of movies that the user had seen, with some numerical rating associated with each of these movies. The output of the recommendation system is another set of artifacts, not yet rated by the user, which the recommendation system predicts the user will rate highly.

Social-filtering systems would solve this problem by focusing solely on the movie ratings for each user, and by computing from these ratings a function that can give a rating to a user for a movie that others have rated but the user has not. These systems have traditionally output ratings for movies, rather than a binary label. They compute ratings for unseen objects by finding similarities between peoples' preferences about the rated items. Similarity assessments are made amongst

individual users of a system and are computed using a variety of statistical techniques. For example, *Recommender* computes for a user a smaller group of reference users known as recommenders. These recommenders are other members of the community most similar to the user. Using regression techniques, these recommenders' ratings are then used to predict ratings for new movies. In this social recommendation approach recommended movies are usually presented to the user as a rank-ordered list.

Content-based recommendation systems, on the other hand, would reflect solely the non-ratings information. For each user they would take a description of each liked and disliked movie, and learn a procedure that would take the description of a new movie and predict whether it will be liked or disliked by the user. For each user a separate recommendation procedure would be used.

Our Approach

The goal of our work is to develop an approach to recommendation that can exploit both ratings and content information. We depart from the traditional social-filtering approach to recommendation by framing the problem as one of classification, rather than artifact rating. On the other hand, we differ from content-based filtering methods in that social information, in the form of other users' ratings, will be used in the inductive learning process.

In particular, we will formalize the movie recommendation problem as a learning problem—specifically, the problem of learning a function that takes as its input a user and a movie and produces as output a label indicating whether the movie would be liked (and therefore recommended) or disliked:

$$f((user, movie)) \rightarrow \{liked, disliked\}$$

As a problem in classification, we also are interested in predicting whether a movie is liked or disliked, not an exact rating. Our output is also not an ordered list of movies, but a set of movies which we predict will be liked by the user. Most importantly, we are now able to generalize our inputs to the problem to other information describing both users and movies.

The information we have available for this process is a collection of user/movie ratings (on a scale of 1-10), and certain additional information concerning each movie.¹ To present the results as sets of movies predicted to be liked or disliked by a user we compute a ratings threshold for each user such that 1/4 of all the user's ratings exceed and the remaining 3/4 do not, and we return as recommended any movie whose predicted rating is above the training-data-based threshold on movies.

¹It would be desirable to make the recommendation process a function of user attributes such as age or gender, but since that information is not available in the data we are using in this paper, we are forced to neglect it here.

Below we will outline a number of alternative ways that a user/movie rating might be represented for the learning system. We will first describe how we represent social recommendation information, which we call “collaborative” features, then how we represent “content” features, and finally describe the hybrid features that form the basis for our most successful recommendation system.

Collaborative Features

As an initial representation we use a set of features that take into account, separately, user characteristics and movie characteristics. For instance, perhaps a group of users were identified as liking a specific movie:

Mary, Bob, and Jill liked *Titanic*.

We defined an attribute called *users who liked movie X* to group users like these into a single feature, the value of which is a set. (*E.g.*, { Mary, Bob, Jill} would be the value of the feature *users who liked movie X* for the movie *Titanic*). Since our ground ratings data contain numerical ratings, we say a user likes a movie if it is rated in the top-quartile of all movies rated by that user.²

We also found it important to note that a particular user was interested in a set of movies, namely the ones which appeared in his top-quartile:

Tim liked the movies, *Twister*, *Eraser*, and *Face/Off*.

This led us to develop an attribute, *movies liked by user*, which encoded a user’s favorite movies as another set-valued feature. We called these attributes *collaborative features* because they made use of the data known to social-filtering systems: users, movies, and ratings.

The result of this is that every user/movie rating gets converted into a tuple of two set-valued features. The first attribute is a set containing the movies the given user liked, and can be thought of as a single attribute describing the user. The second attribute is a set containing the users who like the given movie, and can be thought of as a single attribute describing the movie. Each such tuple is labeled by whether it was liked or disliked by the user, according to whether it was in the top-quartile for the user.

The use of set-valued features led naturally to use of *Ripper*, an inductive learning system that is able to learn from data with set-valued attributes (Cohen 1995; 1996). *Ripper* learns a set of rules, each rule containing a conjunction of several tests. In the case of a set-valued feature *f*, a test may be of the form “ $e_i \in f$ ” where e_i is some constant that is an element of *f* in some example. As an example, *Ripper* might learn a rule containing the test *Jaws* \in *movies-liked-by-user*.

²The value of 1/4 was chosen rather arbitrarily, and our results are similar when this value was changed to 20% or 30%.

Content Features

Content features are more naturally available in a form suitable for learning, since much of the information concerning a movie are available from (semi-) structured online repositories of information. An example of such a resource which we found very useful for movie recommendation is the Internet Movie Database (IMDb) (<http://www.imdb.com>). The IMDb contains an extensive collection of movies and factual information relating to movies. All of our content features were extracted from this resource. In particular, the features we used in our experiments using “naive” content features were: Actors, Actresses, Directors, Writers, Producers, Production Designers, Production Companies, Editors, Cinematographers, Composers, Costume Designers, Genres, Genre Keywords, User-submitted Keywords, Words in Title, Aka (also-known-as) Titles, Taglines, MPAA rating, MPAA reason for rating, Language, Country, Locations, Color, Soundmix, Running Times, and Special Effects Companies.

Hybrid Features

Our final set of features reflect the common human-engineering effort that involves inventing good features to enable successful learning. Here this resulted in *hybrid features*, arising from our attempts to merge data that was not purely content-based nor collaborative. We looked for content that was frequently associated with the movies in our data and that is often used when choosing a movie. One such content feature turned out to be a movie’s *genre*. However, to make effective use of the *genre* feature, it turned out to be necessary to relax an apparently natural assumption: that a $\langle user, movie \rangle$ pair would be encoded as a set of collaborative features, plus a set of content features describing the movie. Instead, it turned out to be more effective to define new collaborative features that are influenced by content. We call these features *hybrid* features.

We isolated three of the most frequently occurring genres in our data — comedy, drama, and action. We then introduced features that isolated groups of users who liked movies of the same genre, such as *users who liked dramas*. Similar features were defined for comedy and action movies. These hybrid features combine knowledge about users who liked a set of movies with knowledge of a particular content feature associated with the movies in a set. Definitions concerning what it means for a user to like a movie remain the same (top-quartile) as in the earlier parts of this paper.

Experiments and Results

We conducted a number of experiments using different sets of features. Below we will report on some of the significant results.

Training and Test Data

Our data set consists of more than 45,000 movie ratings collected from approximately 260 users. This data originated from a data set that was used to evaluate *Recommender*. However, over the course of our work we discovered that the training and test distributions in this data were distributed very differently. We therefore generated a new partition of data into a training set which contained 90% of the data and a testing set which contained the remaining 10%, for which the two distributions would be more similar. Unfortunately, for some of the users *Recommender* failed to run correctly, and those few users were dropped from this study. Note that this was the only reason for dropping users. No users were dropped due to the performance of our own methods.

We generated a testing set by taking a *stratified random sample* of the data, in the following way:

- For every user, separate and group his movie/rating pairs into intervals defined by the ratings. Movies are rated on a scale from 1 to 10.
- For each interval, take a random sample of 10% of the data and combine the results.

Among the advantages of using stratified random sampling (Moore 1985), the primary one for us is that we have clearly defined intervals where all the units in an interval share a common property, the rating. Therefore, the holdout set we computed is more representative of the distribution of ratings for the entire data set than it would have been if we had used simple random sampling.

Evaluation Criteria

As mentioned earlier, we differ from other approaches in the output that we desire. This stems from how we compare ratings of different movies to deal with similarity. Rather than getting the exact rating right, we are interested in predicting whether a movie would be amongst the user's favorites. This has the nice effect of dealing with the fact that the intervals on the ratings scale are *not equidistant*. For instance, given a scale of 1 to 10 where 1 indicates low preference and 10, high preference, the "qualitative" difference between a rating of 1 and a rating of 2 is less when compared to the difference between 6 and 7, for any user whose ratings are mostly 7 and above. Our evaluating a movie as being liked if it is in the top-quartile reflects our belief that knowing the actual rating of a movie is not as important as knowing where the rating was relative to other ratings for a given user.

Both (Hill, Stead, Rosenstein & Furnas 1995) and (Karunanithi & Alspector 1996) evaluate the recommendations returned by their respective systems using *correlation* of ratings. For instance, they compared how well their results correlated with actual user ratings and the ratings of movie critics. Strong positive correlations are indicative of good recommendations.

However, since we are not predicting exact ratings, we cannot use this method of evaluation.

We instead use two metrics commonly used in information retrieval — *precision* and *recall*. Precision gives us an estimate of how many of the movies predicted to be in the top-quartile for a user really belong to that group. Recall estimates how many of all the movies in the user's top-quartile were predicted correctly. A system that returns all movies as liked can achieve high recall. On the other hand, if we are more generous and consider all movies except those in the lowest quartile as liked, then we would expect precision estimates to increase. Therefore, we cannot consider any one measure in isolation.

However, we feel that when recommending movies, the user is more interested in examining a small set of recommended movies rather than a long list of candidates. Unlike document retrieval, where the user can narrow a list of retrieved items by actually reading some of the documents, here, the user is really interested in seeing just one movie. Therefore, our objective for movie recommendation is to maximize precision without letting recall drop below a specified limit. Precision represents the fact that a movie selected from the returned set will be liked, and the recall cutoff reflects the fact that there should be a non-trivial number of movies returned (for example, in case a video store is out of some of the recommended titles).

Baseline Results

In our initial experiment, we use *Recommender's* social-filtering methods to compute predictions for $\langle \text{user}, \text{movie} \rangle$ pairs on the holdout data set. To do this, for every user, we separate his data from the holdout set. The rest of the data is made available to *Recommender's* analysis routines, which means that every other user's test data serves as part of the training data for a given hold-out-user's test data. Then, for every movie in the user's holdout data, we apply *Recommender's* evaluation routines to compute a rating. These routines look for a set of recommenders correlated with the user and compute a rating for a movie using a prediction equation with the recommenders as variables.

For every rating computed by *Recommender*, we need to determine whether it is in the top-quartile. To do this, we precompute *thresholds* for every user corresponding to the ratings which separate the top from the lower quartiles. To convert a rating, we use this rule:

- If a *predicted rating* \geq *user's threshold*, set the rating to "+".
- Otherwise, set the rating to "-".

These thresholds are set individually for each user, using only the training data ratings for the training data threshold, but the full set of data for a user is used to set the testing data threshold.

Our precision estimates are *microaveraged* (Lewis 1991). Microaveraging meant that our prediction decisions were made from a single group and an overall precision estimate was computed. This is preferable to *macroaveraging*, in which one computes results on a per individual basis and averages them at the end, giving equal weight to each user. Unfortunately, in some cases (due to the small amount of data for some users) no movies were recommended, leaving precision ill-defined in these cases. Microaveraging does not suffer this problem (unless no movies are returned for any users). As shown in Table 1, the *Recommender* achieved microaveraged values of 78% for precision and 33% for recall.

Inductive Learning Results

In the first of our inductive learning recommendation experiments using *Ripper*, we use the same training and holdout sets described above. However, now every data point is represented by a collaborative feature vector. The collaborative features we used were:

- Users who liked the movie
- Users who disliked the movie
- Movies liked by the user

The ratings are converted to the appropriate binary classification as described earlier. The entire training set and holdout set are made available to *Ripper* in two separate files. We then ran *Ripper* on this data and generated a classification for each example in the holdout set. *Ripper* produces a set of rules that it learns for this data which it uses to make predictions about the class of an example.

When running *Ripper*, we have the choice of setting a number of parameters. The parameters we found most useful in adjusting from the default settings allow *negative tests in set-valued attributes* and varying the *loss ratio*. The first parameter allows the tests in rules to check for non-containment of attribute values within a set-valued feature. (*E.g.*, tests like *Jaws* \notin *movies-liked-by-user* are allowed.) The *loss ratio* is the ratio of the perceived cost of a false positive to the cost of a false negative; increasing this parameter encourages *Ripper* to improve precision, generally at the expense of recall. In most of the experiments, we varied the loss ratio until we achieved a high value of precision with a reasonable recall. At a loss ratio of 1.9, we achieved a microaveraged precision of 77% and a recall of 27% (see Table 1). This level of precision is comparable to *Recommender*, but at a lower level of recall.

In the second set of experiments, we replaced the collaborative feature vector with a new set of features. In our studies, we extracted 26 different features from the IMDb. The features we chose ranged from common attributes such as *actors* and *actresses* to lesser known features such as *taglines*. We also chose a few features which were assigned to movies by users, such as *keyword* descriptors.

We began by adding the 26 content features to the collaborative features. With these new features, we were not able to improve precision and recall at the same time (see Table 1). Recalling that high precision was more important to us than high recall, we find these results generally inferior to that of *Recommender*. Furthermore, examining the rules that *Ripper* generated, we found that content features were seldom used.

Two points should be noted from this experiment. First, the collaborative data appear to be better predictors of user preferences than our initial encoding of content; as a result, *Ripper* learned rules which ignored all but a few of the content features. Secondly, given the high dimensionality of our feature space, it appears to be difficult to make reasonable associations amongst the examples in our problem.

In our next attempt, we created features that combined collaborative with content information relating to the genre of a movie. These hybrid features were:

- Comedies liked by user
- Dramas liked by user
- Action movies liked by user

Although the movies in our data set are not limited to these three genres, we took a conservative approach to adding new features and began with the most popular genres as determined by the data.

To introduce the next set of collaborative features, we face a new issue. For example, we want a feature to represent the set of users who liked comedies. Although we have defined what it means to like a movie, we have not defined what it means to like movies of a particular genre. How many of the movies in the user's top-quartile need to be of a particular genre in order for the user to like movies of that genre?

Surveying the data, we found that the proportion of movies of any particular genre appearing in a user's top-quartile usually fall into some broad clusters. As a first cut, we divided the proportions of movies of different genres into four groups and created features to reflect the degree to which the user liked a particular genre. For each of the popular genres, *comedy*, *drama*, and *action*, we defined the following features:

- Users who liked many movies of genre *X*
- Users who liked some movies of genre *X*
- Users who liked few movies of genre *X*
- Users who disliked movies of genre *X*

We also add features including, for example, the genre of a particular movie. Running *Ripper* on this data with a loss ratio of 1.5, we achieved a microaveraged precision of 83% with a recall of 34%. These results are summarized in Table 1.

Using the standard test for a difference in proportions (Mendenhall, Scheaffer, & Wackerly 1981, pages 311-315) it can be determined that *Ripper* with hybrid features attains a statistically significant improvement

<i>Method</i>	<i>Precision</i>	<i>Recall</i>
Recommender	78%	33%
Ripper (no content)	77%	27%
Ripper (simple content)	73%	33%
Ripper (hybrid features)	83%	34%

Table 1: Results of the different recommendation approaches.

over the baseline *Recommender* system with respect to precision ($z = 2.25, p > 0.97$), while maintaining a statistically indistinguishable level of recall.³ *Ripper* with hybrid features also attains a statistically significant improvement over *Ripper* without content features with respect to both precision ($z = 2.61, p > 0.99$) and recall ($z = 2.61, p > 0.998$).

Observations

Our results indicate that an inductive approach to learning how to recommend can perform reasonably well when compared to social-filtering methods, evaluated on the same data. We have also shown that by formulating recommendation as a problem in classification, we are able to combine meaningfully information from multiple sources, from ratings to content.

At equal levels of recall, our evaluation criteria would favor results with higher precision. Our results using hybrid features show that even with high precision, we also have a slight edge over recall as well.

We can comment on our features in terms of their effects on recall and precision. When we try to improve recall we are trying to be more inclusive — to add more items in our pot at the expense of unwanted items. On the flip side, when we improve precision, we are being more selective about those items we add. Features like *users who like comedies* help to increase recall. They are a generalization of simple collaborative features like *users who liked movie X*. Features like *comedies liked by user* have the reverse effect. They are a specialization of the collaborative feature, *movies liked by user*, and thereby focus our attention on a subset of a larger space of examples and increase precision.

Related Work

We have already described previous work on recommendation in our discussion of the *Recommender* system. There has also been work which explored the use of content features in selecting movies, in the context of another system designed on social-filtering principles. This previous study compared *clique-based* and *feature-based* models for movie selection (Karunanithi & Alspector 1996). A clique is a set of users whose movie ratings are similar, comparable to the set of

³More precisely, one can be highly confident that there is no practically important loss in recall relative to the baseline; with confidence 95%, the recall rate for *Ripper* with hybrid features is at least 32.8%.

recommenders in (Hill, Stead, Rosenstein & Furnas 1995). Those members of the clique who have rated a movie that the user has not seen predict a rating for that movie. Clique formation is dependent upon two parameters. The first is a correlation threshold which is the minimum correlation necessary to become a member of another user's clique. The second is a size threshold which defines a lower limit on the number of movies that a user must see and rate to become a member of that clique. In their implementation, the authors set the size parameter to a constant value of 10 and set the correlation threshold such that the number of users in the clique is held at a constant 40. After a clique is formed, a movie rating is estimated by calculating the arithmetic mean of the ratings of the members of the clique. This mean serves as the predicted rating for the user. The authors also outline a general algorithm for a feature-based approach to recommendation:

1. Given a collection of rated movies, extract features for those movies.
2. Build a model for the user where the features serve as input and the ratings as output.
3. For every new movie not seen by the user, estimate the rating based on the features of the movie.

They used a neural-network model which associated these features (inputs to the model) with movie ratings (outputs of the model).

In this study, the authors isolated six features describing movies (not necessarily gathered from the IMDb): MPAA ratings, Category (genre), Maltin (critic's) rating, Academy Award, Length of movie, and Origin (related to the country of origin). They justified their choice of features on the grounds that they wanted to start with as small a set of features as possible, and that they found these features easiest to encode for their model.

The category and MPAA ratings were first fed into hidden units. Unlike the other features, which were nominal valued, these two features had a 1-of-N unary encoding. In other words, the feature is encoded as a N-bit vector where each bit represents one of the feature's possible values. (Only one bit corresponding to the feature's value in the example is set.) Although this representation is suited for their model and allows the feature to take multiple values, it is limited to the extent that all the values need to be enumerated at the outset.

In our case, the majority of features, content as well as collaborative, turned out to be set-valued. Set-valued features are more flexible than the 1-of-N unary features. These values can grow over time and need not be predetermined. Computationally, set-valued features are also much more efficient to work with than the corresponding 1-of-N encoding, particularly in cases for which N is large.

In the feature-based study, the authors found that by using features, in most cases, they outperformed a

human critic but almost consistently did worse than the clique method. Our initial results with content features supported these findings. However, we also demonstrated that content information can lead to improved recommendations, if encoded in an appropriate manner.

Fab (Balabanovic & Shoham 1997) is a system which tackles both issues of content-based filtering and social-filtering. In the *Fab* system, content information is maintained by two types of agents: *user agents* associated with individuals and *collection agents* associated with sets of documents. Each collection agent represents a different topic of interest. Each of these agent-types maintains its own profiles, consisting of terms extracted from documents, and uses these profiles to filter new documents. These profiles are reinforced over time with user feedback, in the form of ratings, for new documents. In so doing, the goal is to evolve the agents to better serve the interests of the user and the larger community of users (who receive documents from the collection agents). There are some key differences in our approach. Ours is not an agent-based framework. We do not have access to topics of interest information, which in *Fab*, were collected from the users. We also do not use ratings as relevance feedback for updating profile information. Since we are not dealing with documents, we do not employ IR techniques for feature extraction.

Another well known social-filtering system is *Firefly*. *Firefly*, which has since expanded beyond the domain of music recommendation, is a descendant of *Ringo* (Shardanand & Maes 1995), a music recommendation system. *Ringo* presents the user with a list of artists and albums to rate. This system maintains this information on behalf of every user, in the form of a user profile. The profile is a record of the user's likes and dislikes and is updated over time as the user submits new ratings. The profile is used to compare an individual user with others who share similar tastes. During similarity assessment, the system selects profiles of other users with the highest correlation with an individual user. In the *Ringo* system, two of the metrics used to determine similarity are *mean-squared difference* and the *Pearson-R measure*. In the first case, *Ringo* makes predictions by thresholding with respect to how dissimilar two profiles are based on their mean-squared difference. Then, it computes a weighted average of the ratings provided by the most similar users. In the second case, *Ringo* makes predictions by using Pearson-R coefficients as weights in a weighted-average of other users' ratings.

Final Remarks

In this paper, we have presented an inductive approach to recommendation. This approach has been evaluated via experiments on a large, realistic set of ratings. One advantage of the inductive approach, relative to other social-filtering methods, is that it is far more flexible; in particular it is possible encode collaborative and

content information as part of the problem representation, without making any algorithmic modifications. Exploiting this flexibility, we have evaluated a number of representations for recommendation, including two types of representations that make use of content features. One of these representations, based on hybrid features, significantly improves performance over the purely collaborative approach. We have thus begun to realize the impact of multiple information sources, including sources that exploit a limited amount of content. We believe that this work provides a basis for further work in this area, particularly in harnessing other types of information content.

References

- Balabanovic, M.; and Shoham Y. 1997. Content-Based, Collaborative Recommendation. *Communications of the ACM* Vol. 40, No. 3. March, 1997.
- Cohen, W. 1995. Fast Effective Rule Induction. In *Proceedings of the Twelfth Conference on Machine Learning*. Lake Tahoe, California.
- Cohen, W. 1996. Learning Trees and Rules with Set-valued Features. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*.
- Hill, W.; Stead, L.; Rosenstein, M.; and Furnas, G. 1995. Recommending and Evaluating Choices in a Virtual Community of Use. In *Proceedings of the CHI-95 Conference*. Denver, CO.
- Karunanithi, N.; and Alspector, J. 1996. Feature-Based and Clique-Based User Models for Movie Selection. In *Proceedings of the Fifth International Conference on User Modeling*. Kailua-Kona, HI.
- Lang, K. 1995. NewsWeeder: Learning to filter netnews. In *Machine Learning: Proceedings of the Twelfth International Conference*. Lake Tahoe, California: Morgan Kaufmann.
- Lewis, D. 1991. Evaluating Text Categorization. In *Proceedings of the Speech and Natural Language Workshop*. Asilomar, CA.
- Mendenhall, W.; Scheaffer, R.; and Wackerly, D., eds. 1981. *Mathematical Statistics with Applications*. Duxbury Press, second edition.
- Moore, D. 1985. *Statistics: concepts and controversies*. W. H. Freeman.
- Pazzani, M.; Muramatsu, J.; and Billsus, D. 1996. Syskill & Webert: identifying interesting web sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*.
- Shardanand, U.; and Maes, P. 1995. Social Information Filtering: Algorithms for Automating "Word of Mouth". In *Proceedings of the CHI-95 Conference*. Denver, CO.

分类型推荐:

推荐系统中运用社交的和基于内容的信息

Chumki Basu

Bell Communications Research
445 South Street
Morristown, NJ 07960-6438
cbsu@bellcore.com

Haym Hirsh

Department of Computer Science
Piscataway, NJ 08855
hirsh@cs.rutgers.edu

William Cohen

AT&T Laboratories
180 Park Ave, Room A207
Florham Park, NJ 07932
wcohen@research.att.com

摘要

推荐系统能够就产品为用户提出建议。例如，系统可以预测用户是否会对看某个特定的影片感兴趣。社会化推荐方法向许多人收集他们对产品的评价，使用最近邻技术给关心新产品的用户进行推荐。然而，这些方法并没有用到与产品本质相关的大量有用的信息，例如演员表或者电影预览。这篇论文能够同时运用评价信息和其他形式的信息来预测用户对每一个产品的兴趣，这在推荐中做出了感应学习上的进步。我们使用从超过 250 个用户的社区中收集到的多达 45,000 个影片评价的数据集来展现这一方法在电影推荐领域里胜过现存的社交过滤方法。

介绍

推荐是日常生活的一部分。我们常常会要依赖于一些特定的知识来做出对感兴趣的产品以及行动步骤的知情决策，比如当我们要去看电影或者看医生的时候。这种只是可能源自社会化过程。当我们在买一张 CD 时，我们会听从在音乐上有同样品味的人的见解。还有的时候，我们的评价来自于产品本身的有用信息和我们已有的兴趣偏好。影响一个人做出这些选择的因素有很多，理想的情况是在推荐系统中建立模型时尽可能多

地考虑到这些因素。

在这一问题上有着一些根本的方法。有一种方法是，系统中的用户提供对一些产品和项目的评价，系统就用户会喜欢什么样的其他物品做出知情猜测。这些决定是以所提供的其他用户的评价为基础的。这就是社会化过滤方法的框架 (Hill, Stead, Rosenstein & Furnas 1995; Shardanand & Maes 1995)。第二种方法是，系统接收到描述项目本质的信息，以用户偏好的样例为基础，通过学习来猜测出用户会喜欢什么样的项目 (Lang 1995; Pazzani, Muramatsu, & Billsus 1996)。因为这种方法不依赖于社会化信息(以其他用户评价的形式)，我们称之为基于内容的过滤方法。社交花过滤和基于内容过滤的方法都可以被看作学习的问题：两种情况下，目标都在于去学习一个能够采纳对用户和产品的描述来预测用户对产品的兴趣偏好的函数。知名的推荐系统，像 Recommender (Hill, Stead, Rosenstein & Furnas 1995) 和 Firefly (<http://www.firefly.net>) (Shardanand & Maes 1995) 是建立在社会化过滤的原则上的。Recommender，这一在这里被用作基准系统，基于用户过去对电影的评价和与其他用户评价的相似度来为他推荐还没有看过的电影。仅使用像评价等价值的数值估计的社会化过滤系统性能很好。然而，经常存在着与每一个产品内容相关的准备好的有用信息。社会化过滤方法保留着内容在推荐过程中扮演什么样角色这一问题的开放性。

对于不同类型的产品，已经有了越来越可用

的大量信息储备，同时还在以健康的速率增长中。让我们举这样一个例子，单独在 Web 中的人能够获得的关于一个最为喜欢的影片的信息样例为：对演员阵容和剧组的全面分析，影片情节，影片出品信息，影片预览，预告片，影片和视频片段，（还有评价），还有很多很多。当用户决定要看一部电影时，他们很可能是被上述的一种或多种信息来源所提供的数据所影响。社交化过滤是一种被与感兴趣项目相关的性质所展现的规律所调整得到的一般性的解决途径 (Hill, Stead, Rosenstein & Furnas 1995)。（实际上，在这类系统的工作中有意义的研究动机是探寻由用户兴趣偏好相似形成的可识别的用户社区的实体性。）然而，基于内容的性质可以被低成本地识别（不需要用户额外的贡献）以及人们会被这些规律影响的事实激发了人们去研究如何能最好地使用。

在什么情况下仅使用评价并不充足呢？如果系统能够知道足够多的有重叠特征的其他用户，社交化过滤会起作用。大多数系统提出的代表性的需求是通过用户们对相同产品的评价来判断出用户彼此之间的相似或是不同。这依赖于系统的当前状态——用户的数量和被评价影片的数量和选择。

一个仅仅使用评价带来限制的例子是这样，一种产品还没有得到任何可用的评价，例如一个新的电影上映的情况。这样的话，在一段时间内，推荐系统几乎没有对于这部电影的评价数据，推荐系统无法可靠地为推荐这部电影做好准备。然而，一个使用了内容来做预测的系统即使缺乏电影的评价，已然能够为这部电影进行预测。

在这篇文章里，我们提供了一种新的归纳学习算法来实现推荐。我们阐述如何使用这一方法来实现纯社交过滤，基于内容的信息如何无法帮助、甚至损害到推荐过程的简介，以及如何结合社交元素和基于内容的信息的混合特征来使推荐更为准确。鉴于电影推荐这一问题能够提供大量的数据（多达 250 个人的超过 45,000 条影片评估），我们的工作里探究领域就在于此，同时用社交过滤方法 (Hill, Stead, Rosenstein & Furnas 1995) 作为基准来与我们的结果进行比较。

影片推荐问题

就像前面所述，使用社交过滤方法的推荐系统的输入数据是特定用户对明确的产品给出的评价集合。例如，在影片推荐系统中，这种输入可以是用户看过的影片集合以及一些与这些影片相关的评价数值。推荐系统的输出是其预测出的用户对他还没有做出评价的产品可能会评价较高的产品集合。

社交过滤系统能够通过关注每一个用户的影片评价来解决这个问题，它用这些评价来计算出用户会为其他用户评价过但这个用户还没有的影片给出什么样的评价。传统的系统输出对用户的评价，而不是一个二进制的标签。系统能够通过找寻用户对已评价产品的兴趣偏好与还未看过物品之间的相似度来计算用户评价。系统运用大量的统计方法，为每一个用户个人做相似度的评估。例如，Recommender 为用户计算出较小的可参考他们意见的用户群。这些向用户做推荐的人是社区里与用户相似度最高的。回归分析方法使用这些推荐人的评价来预测用户对新电影的评价。在社交推荐方法中，被推荐的影片以按照评价排名的列表形式展现给用户。

另一方面，基于内容的推荐系统会反映出与评价毫不相干的信息。每一个用户会为他们喜欢或者不喜欢的影片写一段描述，推荐系统会通过对新电影描述的学习步骤来预测用户是否会喜欢这部电影。对每个用户都有着独立的推荐步骤。

我们的方法

我们的工作目标在于开发一种同时利用了评价信息和内容信息的推荐方法。我们同传统的社交过滤方法的区别在于我们把这种方法而不是产品评价视作问题架构里的一个类别。另一方面，我们同基于内容的过滤方法的区别在于在归纳学习的过程中使用了以其他用户评价形式存在的社会化信息。详细地说，我们会把影片推荐问题描述为一种学习问题——这个学习函数的问题是输

入为一个用户和一个电影，输出为一个刻画用户喜欢（所以推荐给他）或是不喜欢这个电影的标签：

$$f(<\text{user}, \text{movie}>) \rightarrow \{\text{liked}, \text{disliked}\}$$

由于这是一个分类问题，我们也是对影片是否被喜欢的预测结果感兴趣，而并不是准确的评价值。我们的输出也就不是一个电影的有序列表，而是我们预测的用户会喜欢的电影集合。更重要的是，我们现在能够将对问题的输入，生成出既描述用户也描述影片的其他信息。

我们的程序可用的信息是一个用户对影片评价（数值从 1 到 10）的集合，以及关于影片的一定的额外信息。（我们希望推荐过程中能够用到年龄和性别的用户属性，但由于在我们这篇论文里这类数据是无法得到的，我们不得不在这里忽视掉它。）为了描述预测出的用户喜欢或者不喜欢的电影集合，我们去对每个用户计算出一个评价阈值，这个阈值比起用户评价中的 $1/4$ 都要大，而小于剩余的 $3/4$ ，然后我们再去推荐那些预测评价值要超过基于训练数据得到的阈值的那类影片。

下面我们将要概述一些用于提供学习系统以用户对影片评价的可选择的方法。首先我们描述“协同”特征，也就是社交推荐信息，然后介绍“内容”特征，最后描述混合特征，这个我们的最为成功的推荐系统的基础。

协同特征

正如前面所述，我们使用一组特征来分别考虑用户特点和影片特点。例如，可以察觉到有一组用户都喜欢一个明确的影片：

Mary, Bob 和 Jill 喜欢《Titanic》。

我们定义一个包含有上述特征的“用户喜欢的电影”集合 X 。（例如， $\{\text{Mary}, \text{Bob}, \text{Jill}\}$ 就是影片《Titanic》对应的“用户喜欢的电影”集合 X ）。由于我们的评价数据是数值形式，我们认为如果用户给影片的评价排在前 $1/4$ ，那么用户就喜欢这个影片。（ $1/4$ 这个数的选择是任意的，而且当这个数在为 20% 或者 30% 时，结果是相似的。）

我们也知道记录一个特定用户喜欢的影片，

也就是那些在他评价列表里排在前 $1/4$ 的影片集合，是很重要的：

Tim 喜欢影片，《Twister》，《Eraser》和《Face/Off》。

这样我们就得到了一个用集合特征来描述用户最喜欢的电影的量“用户喜欢的电影”。因为这个量使得社交过滤系统用到了用户、影片和评价数据，我们称之为“协同特征”。这样带来的结果是每一个用户对影片的评价会转化为一个集合二元组来表示。第一个成员是包含有给定用户所喜欢电影的集合，这可以被看成是描述用户的独立属性。第二个成员是喜欢给定影片的用户集合，这可以被看成是描述影片的一个独立属性。每一个二元组通过考察电影是否出现在用户评价列表的前 $1/4$ 来标记出影片是用户喜欢还是不喜欢的。

自然地使用这种集合特征的是 Ripper 这一归纳学习系统，它能够去学习集合属性的数据（Cohen 1995; 1996）。Ripper 学会一个规则集合，其中每个规则都是由一些测试的结合得来。对于一个集合表示的特征 f ，一个测试的形式为 “ $e_i \in f$ ”，其中 e_i 是样例中 f 的某个元素的一些常数表示。例如，Ripper 可能会学得一个规则包含有测试：《Jaws》 \in 用户喜欢的影片集合。

内容特征

由于影片相关的许多信息是可以从（半）结构化的在线信息知识库中得到，内容特征是可以更自然的找到适宜学习的形式。我们能够找到的在影片推荐方面很有用的资源是因特网影片数据库（IMDb）（<http://www.imdb.com>）。IMDb 包含了大量的影片以及与影片相关的真实信息。我们所有的内容特征都可以从这个资源里抽取出。我们在实验里用的特定特征是很简单的内容特征：男演员，女演员，导演，编剧，制片人，制片设计，制片公司，剪辑师，摄影师，配乐师，服装师，电影流派，风格关键词，用户创作的关键词，标题文字，也就是标题，标语，MPAA 评价，MPAA 评价理由，语言，国家，位置，色彩，混音，播放

时长和特效公司。

混合特征

我们的贡献在于最终的特征选择是去创造了能够成功学习的好的特征。我们尝试融合基于内容和协同特征的数据来得到混合特征。我们寻找与影片频繁相关、在选择影片时常被使用的内客。这样内容特征的一种就是影片的风格流派。然而，想更有效的使用风格特征，及需要放宽这种自然的表面上的假设：一个`<user, movie>`组是用协同特征的集合以及描述影片的一组内容特征来表示的。定义出受到内容影响的新的协同特征会更加有效。我们称之为“混合”特征。

我们提取出在数据中最为频繁出现的三种风格特征——喜剧片，戏剧和动作片。然后我们介绍提取出的喜欢同一风格影片的用户群特征，比如“喜欢看戏剧的用户”。在喜剧片和动作片中也可以定义相似的特征。这种混合特征结合了喜欢某些影片的用户是谁的知识，以及与影片相关的特定内容是什么的知识。就像在论文前面所说一样，如何定义一个用户是否喜欢一个影片保持不变（评价排在前 $1/4$ ）。

实验和结果我们在不同的特征上做了许多实验。我们把一些有意义的结果报告如下。

训练和测试数据

我们的数据集包含了从接近 260 个用户手机来的超过 45,000 条影片评价。来自这个数据集的数据可以用来评估 Recommender。然而，在我们的工作中会发现训练和测试数据的分布是非常不同的。所以，我们将数据的 90%作为训练数据集合，剩余的 10%作为测试集合，这样两者的分布就会更近似。不幸的是，Recommender 对一些用户无法正确运行，这些很少的用户在研究中不考虑到。要记住这是放弃某些用户的唯一原因。我们自己方法的效果不会丢弃任何用户。

我们如下采用分层随机样例数据作为测试集合：

(1) 对每个用户，按照评价值对他评价的影片进行分级。影片评价值从 1 到 10。

(2) 对每个分级，提取随机的 10%的样例数据来校验结果。

采用分层随机采样 (Moore 1985) 的好处中对我们来说最重要的是可以清楚地定义一个等级中的所有单元具有相同的属性，也就是评价。所以，我们计算出的这个集合比起使用简单随机采样要更能代表整个数据集的评价值分布情况。

评价标准

如之前所述，我们要把我们想要的输出同其他的方法区分开来。这来源于我们决定影片相似度时是通过比较不同影片的评价如何。我们感兴趣的是预测影片是否是用户所喜爱的，而不是得到准确的评价数值。这在面对评价等级并不等距的情况下会有好的效果。例如，给定 1 到 10 的数值范围，其中 1 表示兴趣低，10 表示兴趣高；对于给出评价大多为 7 或者更高的用户来说，评价值为 1 和 2 之间的区别比起 6 和 7 之间的就显得小了。我们评价一个影片是否被喜欢的方法是看它是否在评价列表的前 $1/4$ ，这反映了我们认为一个影片的评价具体数值并没有与给定用户对其他影片的评价的对比度重要。

(Hill, Stead, Rosenstein & Furnas 1995) 和 (Karunanithi & Alspector 1996) 都在他们各自的系统中使用了评价值的关联关系来估计推荐结果。例如，他们会用他们的结果和用户实际的评价值以及影片评论家对影片的评价做比对。有更为优异的关联就意味着是好的推荐方法。然而由于我们无法预测准确的评价值，就不能使用这种评估方法。

我们使用两个信息检索领域常采纳的度量标注——准确率和召回率。准确率是指预测出的评价值在前 $1/4$ 的影片有多少是用户真正喜欢的。召回率是指预测的用户评价在前 $1/4$ 的影片有多少是预测正确的。一个认为所有影片都是用户喜欢的，能够使得召回率达到很高。另一方面，如果我们更为大方地

认为除了评价极其低的影片外的所有影片都是用户喜欢的，那么我们就能期待准确率的值增加。所以，我们不能够孤立地采纳其中任何一个测量方法。

然而，我们还知道用户更希望检验的可推荐影片是少量的，而不是一个长长的列表。不像文档检索可以实际地看一组检索出的文档来进行判断，这里，用户必须真的看了影片并且感兴趣才行。所以，影片推荐的目标就是尽可能不让召回率掉到给定的限定以下的条件下，最大化准确率。准确率代表了返回集合中选出的影片是用户喜欢的，召回率的减小是说返回的影片中含有该被喜欢的影片（例如，一个录像店都在被推荐的标题之外）。

基准结果

在实验的初始阶段，我们用 Recommender 的社交过滤方法来计算在所给数据机上的预测 $\langle user, movie \rangle$ 对组。实验中，对每一个用户，我们把他的数据从数据集里提出来。剩余的数据就可以提供给 Recommender 的分析方法，每一个其他用户的测试数据都可以作为给定用户的训练数据。然后，对用户数据中每一个影片，我们使用 Recommender 的评估方法来计算出评价值。这种方法可以找到和用户相关联的一组推荐者，然后以推荐者作为变量，使用预测公式来计算影片的评价值。

对每一个由 Recommender 计算的评价值，我们需要决定它是否在前 $1/4$ 。要做到这个，我们要预先计算出每一个用户的评价阈值以期分来排在前面的影片与排在后面的。我们用这样的规则来转化评价值：

- (1) 如果预测的评价数值 \geq 用户阈值，就设定评价位“+”。
- (2) 否则，就设定为“-”。

这些阈值对于每一个用户都要使用训练评价数据来得到特定的训练数据阈值，但是用户的完整数据集是用来设定一个测试数据阈值的。

我们的准确率估计是微平均的 (Lewis 1991)。微平均是说我们从单独的组数据中计算出

整体的准确率估计值。在微平均方法中，计算每一个独立偏差并最后做平均，得到每一个用户相同的权重，这是可取的。不幸的是，在一些情况（原因是一些用户数据量少）下，并没有推荐的电影，造成了这些情况下准确率定义不正确。微平均方法不会受到这个问题的影响（除非对任何用户都没有影片可返回）。如表 1 所示，Recommender 能够使得准确率达到 78% 并且召回率达到 33%。

归纳学习结果

我们使用 Ripper 做的第一个归纳学习实验中，采用与上述相同的训练数据。然而，现在每一个数据点都被协同特征向量所表示。我们使用的协同特征有：

- (1) 喜欢某影片的用户有哪些
- (2) 不喜欢某影片的用户有哪些
- (3) 被用户所喜欢的影片有哪些

这些评价可以如之前所说转化成合适的二进制分类。用两个独立的文件来给 Ripper 提供可用的完整训练数据集合。然后我们在数据上运行 Ripper 并且对每一个样例生成一个分类结果。Ripper 产生一组从数据中学来的规则，能够用来预测样例的类别。当运行 Ripper 时，我们在设置许多参数时有着选择。我们发现调整默认设置，允许在数据集上的进行消极测试并且改变丢失比，这样的参数更有用。第一个参数允许规则测试来检查在集合特征里的非包含关系。（例如，像“大白鲨不属于用户所喜欢的影片集合”是允许的。）丢失比是指从错误肯定到错误否定的感知代价，提升整个参数是鼓励 Ripper 以消耗召回率的代价去提升准确率。在大多数的试验中，我们直到用一个有意义的召回率得到了高数值的准确率，就改变了丢失比。当丢失比为 1.9 时，我们得到微平均的准确率为 77% 以及召回率为 27%（如表 1）。这个等级的准确率可以和 Recommender 做比较，但召回率更低。

在第二组的实验中，我们用一组心的特征来替代协同特征向量。在我们的研究中，我们从 IMDb 里抽取 26 个不同的特征。我们所选择的特征从常见属性如男演员和女演员到

较少知道的属性，如标志性语言。我们也还选择一些用户为影片创作的特征，例如关键词描述等。

我们开始给协同特征添加进 26 个内容特征。有了这些新的特征，我们不能同时提高准确率和召回率（如表 1）。我们知道高的准确率比起高的召回率更重要，我们发现这些结果比起 Recommender 的要差。另外，通过检验 Ripper 生成的规则，我们发现内容特征几乎没有被用上。

这个实验必须注意两点。第一点，协同数据比起我们刚刚采用的内容数据在预测用户偏好上更好；结果是 Ripper 学习出特征时只注意到一点内容特征。第二点，对给定的高维特征空间，在我们问题的样例中，做出有意义的关系是困难的。

在我们的下一次尝试中，我们结合协同信息和关于影片风格的内容信息的特征。这个混合特征是：

- (1) 用户所喜欢的喜剧
- (2) 用户所喜欢的戏剧
- (3) 用户喜欢的动作片

虽然我们数据集中的影片不限定于这样三个风格，我们采用一种保守的方法来添加新的特征并且由数据决定的最为流行的风格。要介绍下一个组协同特征，我们要面对一个新的问题。例如，我们想要一个代表用户喜欢的喜剧的特征。虽然我们已经定义了什么是喜欢一个影片，但我们还没有定义什么是喜欢一个特定风格的影片。用户有多少个某一个特定风格的影片需要被排在他所喜欢的影片中的前 $1/4$ ，才能说明他喜欢这一风格的影片？

调查了数据，我们发现任何在用户最为喜欢的电影列表前 $1/4$ 的特定风格的影片通常会变成一些集群。如同第一次分离，我们把不同风格的影片按比例分成四个组，创造出可以反映用户喜欢一个特定风格的程度的特征。对每一种流行的风格，喜剧，戏剧和动作片，我们定义如下的特征：

- (1) 喜欢风格 X 的许多影片的用户
- (2) 喜欢风格 X 的一些影片的用户
- (3) 几乎不喜欢风格 X 的影片的用户
- (4) 讨厌风格 X 的影片的用户

我们也可以添加诸如特定影片风格的特征。在丢失比为 1.5 的数据上运行 Ripper，我们得到微平均的准确率为 83%，并且召回率为 34%。这些结果总结在表 1 中。

在比例的差异上做标准化测试 (Mendenhall, Scheaffer, & Wackerly 1981, pages 311-315)，能够确认用混合特征的 Ripper 在统计学上与基准 Recommender 相比具有准确率 ($z=2.25$, $p>0.97$) 方面有意义的提高，同时维护者统计学上并没有太大区别的召回率（更准确地说，相比于基准情况，那一点召回率的损失实际上远远不够重要；当确信度为 95% 是，混合特征下 Ripper 的召回率至少达到 32.8%）。混合特征的 Ripper 也在与统计学上和不含内容特征的 Ripper 相比具有准确率 ($z=2.61$, $p>0.99$) 和召回率 ($z=2.61$, $p>0.998$) 方面有意义的提高。

方法	准确率	召回率
Recommender	78%	33%
Ripper (不含内容)	77%	27%
Ripper (简单内容)	73%	33%
Ripper (混合特征)	83%	34%

表 1：不同推荐方法的结果

观测

我们的结果显示着，当和社交过滤方法作比较，在同一数据上评估的时候，一个学习如何推荐的归纳方法可以运作得相当的好。我们也可以发现用形式化的推荐来处理分类问题，我们能够将从多个源里的评价到内容的有意义的信息相结合

在同等的召回率下，我们的评估标准可以有更高的准确率。使用混合特征的结果显示即使有高准确率，我们也只是在召回率上有轻微的偏移。

我们能够按照召回率和准确率的形式来评论我们的特征。当我们想要去提高召回率时，我们会尽力去做得更全面——冒着添加了不想要的内容的风险，在实验中加进更多的

东西。另一方面，当我们提高准确率时，我们在添加项目时会更有选择性。诸如“喜欢喜剧的用户”的特征能够帮助提高召回率。这些特征是从像“喜欢影片 x 的用户”这些简单的协同特征里生成的。像“用户喜欢的喜剧”的特征会有相反的效果。这是从“用户喜欢的影片”的协同特征里特殊化得到的，从而注意我们在样例的更大的子集上的意图来提高准确率。

相关工作

在我们对 Recommender 系统的讨论中，我们已经描述了在推荐上以前的工作。也有着挑选影片时采纳内容特征的工作，有着在社交过滤原则设计的另一个系统。这个之前的研究在影片挑选上同基于圈子和基于特征的模型相对比。一个圈子和在(Karunanithi & Alspector 1996) 的推荐者进行对比，是指对影片的评价相似的一组人。已经评价了用户不曾看过的影片的圈子里的成员为电影预测评价。圈子的形成依赖于两个参数。第一个是关联阈值，也就是成为另一个用户圈子中一员所必须的最小关系。第二个是大小阈值，这定义了用户成为圈子一员所必须观看和评价的影片数量的下线。在他们的提高部分，作者设置了大小参数为常量 10，设置了关联阈值为圈子里的用户数量保持在常量 40。在一个圈子形成之后，一个电影的评价就可以通过计算这个圈子里成员们所给评价的算术平均数来估计得到。这个平均数作为用户的预测评价。作者还给出一个基于特征的推荐方法的综合算法：

- (1) 给定一组被评价的影片，抽取这些影片的特征。
- (2) 为用户建立模型，其中特征作为输入，评价作为输出。
- (3) 对于每一个用户没有看过的新影片，从影片的特征上估计评价。

他们使用一种中立网络模型来关联这些特征（输入到模型中）和影片的评价（模型的输出）。

在这个研究中，作者把六个描述影片的特征独立出来（不是必须从 IMDb 中收集来）：

MPAA 评价，分类（流派），Maltin（评论家的）评价，影片奖项，电影时长和来源（来自的国家）。他们想从尽量小的一组特征出发，来校正对基本特征的选择，他们发现这些特征对于他们的模型最为容易。

分类和 MPAA 评价被最先引入为隐式单元。不像其他的用名词衡量的特征，这两个特征是用 N 分之一为单位编码的一元值。另一方面，被编码成 N 位向量的特征中，每位表示了特征可能的数值中一个。（在集合样例中与特征值相关的位是唯一的。）虽然这种表示适合他们的模型并且允许其采纳多个数值，但它也受到所有数值需要在开始时就计算出来的限制。

在我们的例子中，大多数的特征，包括内容还有协同数据，都可以用集合表示。集合表示的特征比起 N 分之一为单位的一元值表示要灵活得多。这些值可以随着时间变化，也不需要早早决定。集合特征值在计算方面也比起与 N 分之一单位编码要更有效得多，尤其是当 N 很大的时候。

在基于特征的研究中，作者发现，在大多数情况下使用特征比起人们的评价要效果更好，但是几乎还是要都是要不如圈子方法。我们用内容特征的初始结果支持这些发现。然而，我们也声明，如果按照适当的方式编码，内容信息能够带来推荐上的提高。

Fab (Balabanovic & Shoham 1997) 是一个既处理了基于内容过滤，又处理了社交过滤的系统。在 Fab 系统中，内容信息被两种类型的代理所维护：用户代理与个人相关联，集合代理与文档集合关联。每一个集合代理代表着一个不同的兴趣话题。代理类型中每一个都维护着它自己的架构，包括从文档里提取的词语，并且用这些架构来过滤新的文档。这些架构对新文档，以评价的形式，使用用户反馈进行时间上的加强。这么做的目的是引进能更好地为用户和更大的用户群兴趣的兴趣服务的代理（从集合代理中接收文档）。这与我们的方法截然不同。我们的不是一个基于代理的架构。我们不需要接触到在 Fab 中需要收集的兴趣话题的信息。我们也不需要通过相关反馈的评价来更新结构信息。由于我们不用处理文档，我们也就不能

需要为特征提取而运用信息检索的技术。另一个知名的社交过滤系统是 Firefly。拓展到音乐领域的推荐系统 Firefly，是音乐推荐系统 Ringo (Shardanand & Maes 1995) 的衍生物。Ringo 用一组用来评价的音乐家和专辑来表示用户。这个系统以用户个人介绍的形式，维护每一个用户的特征。这个个人介绍是描述用户喜欢什么讨厌什么的记录，而且会随着用户提交新的东西而随着时间不断更新。这个个人介绍用来比较一个用户和其他人是否有相似的品味。在相似度的评估中，系统选择和这个用户紧密相关的其他人的个人介绍。在 Ringo 系统中，用两个标准来决定相似度，分别是均方误差和皮尔逊-R 测量方法。在第一种里，Ringo 基于两个个人介绍的均方误差来限定其不相似程度，从而做出预测。然后它计算由大多数相似的用户提供的带权评价平均值。在第二种情况下，Ringo 用皮尔逊-R 系数作为其他用户评价的平均带权值，来预测结果。

最后的话

在这篇文章里，我们提出了一种推荐方面的归纳方法。这种方法已经通过大量的现实的评价数据集上的实验进行了评估。归纳方法的一个优势在于，相对于其他的社交过滤方法，更为灵活；特别的，它能够在不做任何的算法变动情况下，将协同信息和内容信息编码为问题中的部分表示。得力于这种灵活性，我们评估了推荐的许多表示方法，包括使用了内容特征的两种类型表示。一种表示是基于混合特征的，在纯协同方法上提高效果很有意义。这样我们也开始意识到多种信息资源的影响，包括从有限数量的内容上获利。我们相信本项工作为这个领域得未来工作提供了基础，特别是在治理其他类型的内容信息上。

参考文献

Balabanovic, M.; and Shoham Y. 1997.

- Content-Based, Collaborative Recommendation. Communications of the ACM Vol.40, No.3. March, 1997.
- Cohen, W. 1995. Fast Effective Rule Induction. In Proceedings of the Twelfth Conference on Machine Learning. Lake Taho, California.
- Cohen, W. 1996. Learning Trees and Rules with Set-valued Features. In Proceedings of the Thirteenth National Conference on Artificial Intelligence.
- Hill, W.; Stead, L.; Rosenstein, M.; and Furnas, G. 1995. Recommending and Evaluating Choices in a Virtual Community of Use. In Proceedings of the CHI-95 Conference. Denver, CO.
- Karunanithi, N.; and Alspector, J. 1996. Feature-Based and Clique-Based User Models for Movie Selection. In Proceedings of the Fifth International Conference on User Modeling. Kailua-Kona, HI.
- Lang, K. 1995. NewsWeeder: Learning to filter netnews. In Machine Learning: Proceedings of the Twelfth International Conference. Lake Taho, California: Morgan Kaufmann.
- Lewis, D. 1991 。 Evaluating Text Categorization. In Proceedings of the Speech and Natural Language Workshop. Asilomar, CA.
- Mendenhall, W.; Scheaffer, R.; and Wackerly, D., eds. 1981. Mathematical Statistics with Applications. Duxbury Press, second edition.
- Moore, D. 1985. Statistics: concepts and controversies. W.H.Freeman.
- Pazzani, M.; Muramatsu, J.; and Billsus, D. 1996. Syskill & Webert: identifying interesting web sites. In Proceedings of the Thirteenth National Conference on Artificial Intelligence.
- Shardanand, U.; and Maes, P. 1995. Social Information Filtering: Algorithms for Automating "Word of Mouth". In Proceedings of the CHI-95 Conference. Denver, CO.