# 大规模通用网页检索与网络新闻检索的智能排序算法研究

# Research of Intelligent Ranking Algorithms for Large Scale General Web Search and Web News Search

（申请清华大学工学硕士学位论文）


培 养 单 位 ： 计算机科学与技术系

学　　科 ： 计算机科学与技术

研 究 生 ： 王 珏

指 导 教 师 ： 孙 增 圻 教 授


二〇〇六年五月

大规模通用网页检索与网络新闻检索的智能排序算法研究　王珏

# 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：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）根据《中华人民共和国学位条例暂行实施办法》，向国家图书馆报送可以公开的学位论文。

本人保证遵守上述规定。

**（保密的论文在解密后遵守此规定）**

作者签名：_____ 导师签名：_____

日　　期：_____ 日　　期：_____

# 摘　要

本文介绍了作者在大规模通用网页检索与网络新闻检索的智能排序算法方面的研究工作。本文分为三个部分。

第一部分[1]介绍了作者在大规模通用网页检索的排序函数学习方面的工作。作者针对当前排序函数的学习中，有标注训练样本普遍不足的情况，提出了基于流形排序算法的半监督学习机制，将少量已标注样本的排序值传递到其他未标注样本上，进而大大增加了学习可用样本数。基于此种相关度传递机制——MRBRP（Manifold Ranking Based Relevance Propagation），作者提出了一套完整的排序函数的学习框架，在主流商用搜索引擎的海量数据集合上的实验表明，此种学习框架可以显著提高排序函数的性能。

第二部分介绍了作者对新闻流行度排序[2]问题的研究。新闻流行度排序即不考虑单个用户个人爱好的排序算法，适用于大型公开的新闻发布（如 Google News 等）。此算法是基于新闻在新闻网站首页的视觉分布所代表的重要性和多新闻站点之间相关新闻的投票来确定新闻事件的重要性。在对新闻进行排序的同时，也得到了对新闻站点的排序，新闻和站点之间构成了相关增强的关系。作者通过一个三分图模型对新闻站点，新闻和事件之间的关系进行了建模，该三分图模型上的平衡解为最终的排序结果。同时，新闻的时效性问题在本算法中也有所反映。在多个实际新闻站点数据上的实验结果表明此算法具有良好的效果和用户体验。

第三部分介绍了作者在用户个人兴趣模型以及相似用户群体的协同滤波的用户个性化新闻推介排序[3]方面的研究。此方法适用于个性化新闻阅读器等。作者针对现有的主流用户兴趣模型的不足，提出了一个对应着事件具有动态节点层的全新的层级结构用户兴趣模型；同时，作者又提出了基于此模型的改进的协同滤波算法以更精确的利用相似用户群体的信息。在实际的用户新闻浏览数据上的实验结果表明，此算法可显著的提高用户个性化新闻推介的准确度。

**关键词：**排序函数　流形排序　半监督学习　网络新闻　新闻流行度　三分图　首页视觉分布　个性化推介　用户兴趣模型　协同滤波

# Abstract

In this paper, we presented our work on intelligent ranking algorithms for large scale general web search and web news search. This paper consists of three parts.

In the first part, we described our work on ranking function learning for large scale web search [1]. Due to the unaffordable time and money cost, the labeling process could only be applied on very few samples, which is far from enough for ranking function learning. This is an essential problem for all large scale web search engines. To deal with this problem, we proposed a relevance propagation scheme which could propagate label score (or relevant score) from labeled samples to unlabeled ones by Manifold Ranking algorithm. Thus, the training set with good labels is extremely augmented. We proposed a general learning framework based on this Manifold Ranking Based Relevance Propagation (MRBRP). Any compatible learning algorithm could be incorporated in our framework with MRBRP. Experimental results on commercial search engine data show that the framework attains a significant improvement over existing ranking function learning algorithms.

In the second part and third part, we describe our work on web news ranking problem.

In the second part, we describe our work on news popularity ranking. This ranking problem considers no user personal interests. It is quite similar to Google News and Baidu news, and this approach is suitable for creating public news top stories. In our approach, the importance of a news event is determined by the visual layout information of news site homepage and cross-site voting for similar pages. We use a tripartite graph to model the reinforcement relationships among news pages, news sites, and news events. This approach also deals with the decaying effect of news. Experimental study

on news data from multiple commercial news sites indicates the effectiveness of the proposed approach.

In the third part, we describe our work on personalized news recommendation based on adaptive user profile model and collaborative filtering. This approach is suitable for personalized news list delivery. In this part, we proposed a two-level hierarchical user profile model, with could model user interest in both the fixed concept category level and dynamic event level. Based on this profile model, we proposed a modified collaborative filtering algorithm to better utilize similar users' information. Experimental results on real news browsing data shows the advantage of our approach over existing profile model and collaborative filtering algorithms.

**Keywords:** Ranking Function Manifold Ranking Semi-supervised Learning Web News News Importance Visual Layout Personalized Recommendation User Profile Model Collaborative Filtering

# 目 录

# 第1章　引言

在当今信息时代，用户通过网络来获取信息的行为日益普遍。来自 iResearch 的数据显示[1]，2005 年美国网民的数目是 1.33 亿，占总人口比例的 66%，而在 2006 年 4 月，这一数字已增长到 1.47 亿，占总人口比例的 73%。而作为世界上第二大互联网市场的中国，现在拥有 1.1 亿的互联网用户，这一数字将在未来的 5 年内增长至 2.3 亿[2]。互联网用户的数目在随着互联网本身的飞速发展而增长，网络用户对互联网的使用也是越来越广泛。在 Cerberian 和 SonicWall 于 2004 年组织的一次网络用户使用情况调查报告显示[3]，个人用户对网络的使用主要分布在如下的类别中：

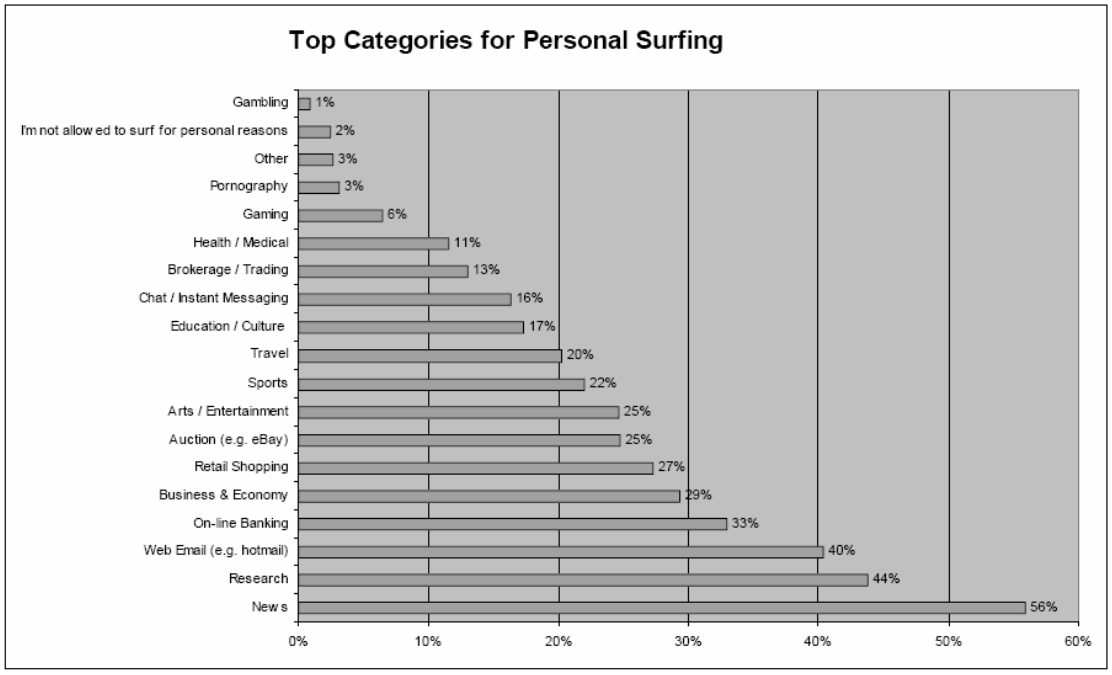


图 1.1 互联网用户的使用类别

从上图可以看出，用户对网络的使用主要是围绕着信息获取这一主题进行的。然而，当今互联网的规模已逾 200 亿页面之巨（仅被 Google 索引的页面就有 80 亿），这还是没有考虑到可以产生无数新网页的动态数据库网络（Deep Web）

的结果。据不完全统计[4]，仅在中国互联网上网站的数目就已达 1382130 之巨。网络上存在的如此海量信息及其爆炸性的增长速度，使得由 Yahoo 等早期互联网公司所倡导的传统信息获取过程——目录式分类式导航浏览（Classified Directory Navigation），无法满足人们不断增长的信息需求。此时，通过搜索（Search）的方式来进行信息获取成为了当前的主流。人们通过搜索在网络上查找信息主要借助于搜索引擎这一工具。有调查显示[5]，当前的主流门户网站（如新浪，雅虎，搜狐等）的访问量有 47%以上来自于搜索引擎（即用户从搜索引擎的搜索结果中链接到门户网站中的页面）；同时，百度的流量已经超过了新浪，成为了中文网站流量之冠，而 Google 的流量也已经接近 Yahoo 的总体流量。这也从另一侧面反映了搜索引擎已经成为网络用户获取信息的最主要的入口。所以，对搜索相关技术的研究是具有十分重要的现实意义的。

另外，从图 1.1 我们可以看到通过网络阅读新闻，已经成为用户在网上最重要的行为之一。所以，研究如何利用搜索技术改进当前网络新闻的发布方式，以摆脱传统新闻媒介的发布模式的束缚，创建真正的网络搜索时代的新闻模式，具有非常重要的意义和应用价值。

## 1.1　通用网页检索和新闻检索的定义和比较

本文中通用网页检索和新闻检索有其特定的含义和使用范围，作者将在本节中阐明二者的定义和关系，以明确本文研究工作的应用背景。

**定义 1.1 ---- 通用网页检索：**根据用户提供的一个或一系列的关键词，在万维网（World Wide Web）上查找相关的文档（以 HTML 文档为主，也包括其他被索引的文件类型），并将查找到的文档列表按照一定规则排序返回给用户的搜索服务。

**定义 1.2 ---- 新闻检索：**包括两种工作模式，1）根据用户提供的关键字或浏览记录等信息，在万维网上的新闻源中，检出符合用户兴趣需求的新闻文档，将其按照一定规则组织排序返回给用户；2）不考虑具体用户的兴趣需求，将网络上的新闻按照对所有用户群体通用的规则组织排序展现给用户。

通用网页检索（General Web Search）是所有的通用搜索引擎都要提供的基本服务。其技术和系统也往往是新闻检索，图像检索以及其他的限定领域（domain specific）的检索的基础。当前的通用网页检索通常包括各种可以索引

的非多媒体文本文档信息，如 HTML，TXT，PDF，PS，DOC，PPT 等。

通用网页检索往往是来满足如下三类用户信息需求[6]：

1. **导航性的（Navigational）。**此类查询的目的是查找某个确定的网站链接（用户默认该网站是可能存在的）。

   例如，输入查询 *compaq*，用户可能想查找的结果是康柏的网站 http://www.compaq.com。

2. **资料性的（Informational）。**此类查询的目的是在网络上查找确定存在的一些“静态”信息。经过查询之后，用户只需阅读即可，不需要其他的交互性操作。所谓“静态”是指目标文档不是根据用户查询动态生成的。

   例如，查询 *cars*，*San Francisco* 等等。

3. **事务性的（Transactional）。**此类查询的目的是找到需要进一步进行用户交互的网页，该网页中包含用户需要从事的“事务”。

   例如，搜索购物站点，文件下载站点，或者查找游戏服务器等等。

而一般用户阅读新闻的需求包括如下三大类：

1. **流行度。**即阅读由编辑确定的最为“热门”（Popular）的新闻。比如阅读新闻站点的头版头条。这反映了用户的从众心理。
2. **个人兴趣。**即阅读用户本人比较感兴趣的新闻类别或者话题。
3. **相似用户群体的兴趣。**即阅读与用户相似的用户群体共同感兴趣的新闻。

由于用户在阅读新闻方面的信息需求和对通用网页检索的需求完全不同，同时由于新闻本身具有时效性，快速更新等特性以及新闻事件这一潜在概念实体的存在，新闻检索的排序算法研究和通用网页检索的排序算法研究所采用的方法和出发角度都有很大不同。二者在模型和研究方法上的区别具体请参见 1.4。

## 1.2　网络搜索引擎原理

网络搜索涉及到的计算机科学中的各个领域，包括：系统结构，数据库，人工智能（信息检索，数据挖掘，机器学习），人机交互，甚至包括心理学等其他学科的交叉内容。此处只能对其原理作一简要介绍，而作者本人的工作则集中在研究搜索结果的排序算法方面，属于信息检索，机器学习领域。

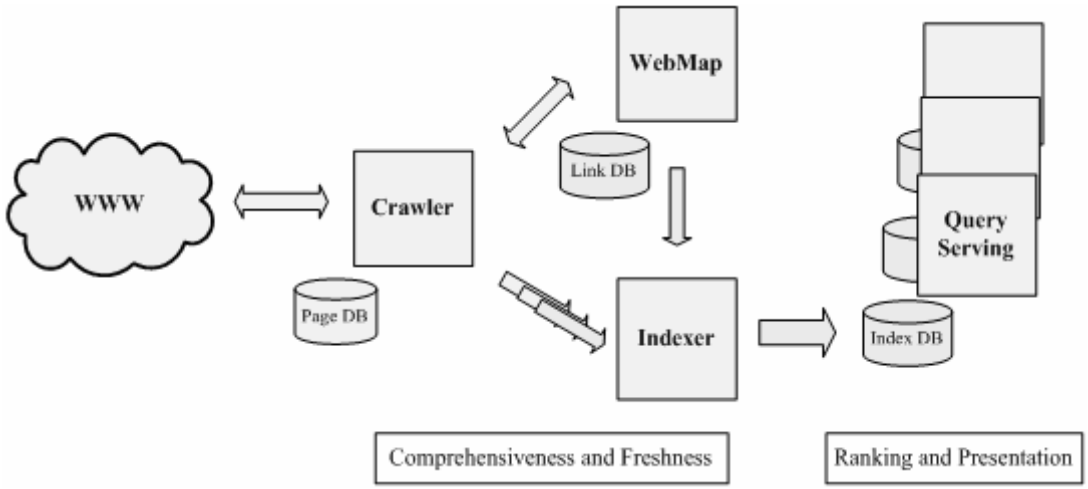


图 1.2 搜索引擎的系统架构

上图是一般网络搜索引擎的系统架构。由上图我们可以看到，现在的第二代网络搜索引擎有四个主要的组成部分：**下载模块**（Crawler），**链接分析模块**（WebMap），**索引模块**（Indexer），和**查询服务模块**（Query Serving）。

下面简要介绍一下搜索引擎的工作流程，以说明排序函数是如何发挥作用的。首先，下载模块从万维网上下载包括普通网页在内的所有可以索引的文件，保存在文件数据库中。链接分析模块从文件服务器中的网页文件中抽取链接，构建页面之间的连接关系，并进行一系列的运算（例如 PageRank[9]的计算）。接下来，索引模块利用文件数据库中的文件以及由链接分析模块得到的网络图信息，为页面建立倒排索引，索引的结果保存在索引数据库中。当查询服务模块接收到用户查询（query）时，就会按照某种匹配方法从索引数据库中查找出一系列的相关文档，最后使用排序函数对这些相关文档进行排序，将排序的结果展现给用户。

需要说明的是，查询服务模块从索引数据库中通过匹配查找出相关文档的同时，会为每一个文档生成一组特征值。这些特征值包括与查询相关的特征，如查询关键词匹配次数（hit number），关键词距离（proximity）等；还包括与查询无关的特征，如页面质量（page quality），是否垃圾页面（spam），PageRank 值等。排序函数就是根据这些特征值得到每一篇文档的排序值，进而得出最终的排序结果。

搜索结果排序的优劣将直接影响用户的体验，排序性能是决定搜索引擎质

量的四条因素之一，或者称之为搜索引擎的第一要素也不为过。这四条因素包括：

**排序**（Ranking）。即能严格按照与用户查询的相关程度进行排序。

**完备性**（Comprehensiveness）。即能尽量覆盖万维网上的信息内容，以保证用户可以查到需要查询的内容。

**信息新鲜程度**（Freshness）。即尽量保证页面数据库和索引数据库中的数据是最新的，能跟随网络上不断发生的变化（如新页面的加入，原有页面的撤销和改变等）。

**信息表达**（Presentation）。以用户友好的方式将搜索的结果显示给用户，尽量缩短用户浏览结果所需的时间。

## 1.3　网络搜索引擎发展简史

1990 年以前，没有任何人能搜索互联网。

所有搜索引擎的祖先，是 1990 年由 Montreal 的 McGill University 学生 Alan Emtage、Peter Deutsch、Bill Wheelan 发明的 Archie(Archie FAQ)。虽然当时 World Wide Web 还未出现，但网络中文件传输还是相当频繁的，由于大量的文件散布在各个分散的 FTP 主机中，查询起来非常不便，因此 Alan Emtage 等想到了开发一个可以用文件名查找文件的系统，于是便有了 Archie。Archie 是第一个自动索引互联网上匿名 FTP 网站文件的程序，但它还不是真正的网络搜索引擎(Web Search Engine)。Archie 是一个可搜索的 FTP 文件名列表，用户必须输入精确的文件名搜索，然后 Archie 会告诉用户哪一个 FTP 地址可以下载该文件。

由于 Archie 深受欢迎，受其启发，Nevada System Computing Services 大学于 1993 年开发了一个 Gopher（Gopher FAQ）搜索工具 Veronica（Veronica FAQ）。Jughead 是后来另一个 Gopher 搜索工具。

Robot（机器人）一词对编程者有特殊的意义。Computer Robot 是指某个能以人类无法达到的速度不断重复执行某项任务的自动程序。由于专门用于检索信息的 Robot 程序象蜘蛛(spider)一样在网络间爬来爬去，因此，搜索引擎的 Robot 程序被称为 spider(Spider FAQ)程序。世界上第一个 Spider 程序，是 MIT Matthew Gray 的 World wide Web Wanderer，用于追踪互联网发展规模。刚开始它只用来统计互联网上的服务器数量，后来则发展为也能够捕获网址（URL）。

与 Wanderer 相对应，1993 年 10 月 Martijn Koster 创建了 ALIWEB（Martijn Koster Annouces the Availability of Aliweb），它相当于 Archie 的 HTTP 版本。ALIWEB 不使用网络搜寻 Robot，如果网站主管们希望自己的网页被 ALIWEB 收录，需要自己提交每一个网页的简介索引信息，类似于后来大家熟知的 Yahoo。

随着互联网的迅速发展，使得检索所有新出现的网页变得越来越困难，因此，在 Wanderer 基础上，一些编程者将传统的 Spider 程序工作原理作了些改进。其设想是，既然所有网页都可能有连向其他网站的链接，那么从一个网站开始，跟踪所有网页上的所有链接，就有可能检索整个互联网。到 1993 年底，一些基于此原理的搜索引擎开始纷纷涌现，其中最负盛名的三个是：Scotland 的 JumpStation、Colorado 大学 Oliver McBryan 的 The World Wide Web Worm（First Mention of McBryan's World Wide Web Worm）、NASA 的 Repository-Based Software Engineering (RBSE) spider。JumpStation 和 WWW Worm 只是以搜索工具在数据库中找到匹配信息的先后次序排列搜索结果，因此毫无信息关联度可言。而 RBSE 是第一个索引 Html 文件正文的搜索引擎，也是第一个在搜索结果排列中引入关键字串匹配程度概念的引擎。

Excite 的历史可以上溯到 1993 年 2 月，6 个 Stanford（斯坦福）大学生的想法是分析字词关系，以对互联网上的大量信息作更有效的检索。到 1993 年中，这已是一个完全投资项目 Architext，他们还发布了一个供 webmasters 在自己网站上使用的搜索软件版本，后来被叫做 Excite for Web Servers。（注：Excite 后来曾以概念搜索闻名，2002 年 5 月，被 Infospace 收购的 Excite 停止自己的搜索引擎，改用元搜索引擎 Dogpile）

1994 年 1 月，第一个既可搜索又可浏览的分类目录 EINet Galaxy（Tradewave Galaxy）上线。除了网站搜索，它还支持 Gopher 和 Telnet 搜索。

1994 年 4 月，Stanford University 的两名博士生，美籍华人 Jerry Yang（杨致远）和 David Filo 共同创办了 Yahoo（Jerry Yang Alerts a Usenet group to the Yahoo Database ，1996 年的 Yahoo）。随着访问量和收录链接数的增长，Yahoo 目录开始支持简单的数据库搜索。因为 Yahoo!的数据是手工输入的，所以不能真正被归为搜索引擎，事实上只是一个可搜索的目录。Wanderer 只抓取 URL，但 URL 信息含量太小，很多信息难以单靠 URL 说清楚，搜索效率很低。Yahoo!中收录的网站，因为都附有简介信息，所以搜索效率明显提高。（注：Yahoo 以后陆续使用 Altavista、Inktomi、Google 提供搜索引擎服务）

1994 年初，Washington 大学 CS 学生 Brian Pinkerton 开始了他的小项目 WebCrawler（Brian Pinkerton Announces the Availability of Webcrawler）。1994 年 4 月 20 日，WebCrawler 正式亮相时仅包含来自 6000 个服务器的内容。WebCrawler 是互联网上第一个支持搜索文件全部文字的全文搜索引擎，在它之前，用户只能通过 URL 和摘要搜索，摘要一般来自人工评论或程序自动取正文的前 100 个字。（后来 webcrawler 陆续被 AOL 和 Excite 收购，现在和 excite 一样改用元搜索引擎 Dogpile）

Lycos（Carnegie Mellon University Center for Machine Translation Announces Lycos ）是搜索引擎史上又一个重要的进步。Carnegie Mellon University 的 Michael Mauldin 将 John Leavitt 的 spider 程序接入到其索引程序中，创建了 Lycos。1994 年 7 月 20 日，数据量为 54,000 的 Lycos 正式发布。除了相关性排序外，Lycos 还提供了前缀匹配和字符相近限制，Lycos 第一个在搜索结果中使用了网页自动摘要，而最大的优势还是它远胜过其它搜索引擎的数据量：1994 年 8 月－－394,000 documents；1995 年 1 月－－1.5 million documents；1996 年 11 月－－over 60 million documents。（注：1999 年 4 月，Lycos 停止自己的 Spider，改由 Fast 提供搜索引擎服务）

Infoseek（Steve Kirsch Announces Free Demos Of the Infoseek Search Engine）是另一个重要的搜索引擎，虽然公司声称 1994 年 1 月已创立，但直到年底它的搜索引擎才与公众见面。起初，Infoseek 只是一个不起眼的搜索引擎，它沿袭 Yahoo!和 Lycos 的概念，并没有什么独特的革新。但是它的发展史和后来受到的众口称赞证明，起初第一个登台并不总是很重要。Infoseek 友善的用户界面、大量附加服务（such as UPS tracking, News, a directory, and the like) 使它声望日隆。而 1995 年 12 月与 Netscape 的战略性协议，使它成为一个强势搜索引擎：当用户点击 Netscape 浏览器上的搜索按钮时，弹出 Infoseek 的搜索服务，而此前由 Yahoo!提供该服务。（注：Infoseek 后来曾以相关性闻名，2001 年 2 月，Infoseek 停止了自己的搜索引擎，开始改用 Overture 的搜索结果）

1995 年，一种新的搜索引擎形式出现了——元搜索引擎（A Meta Search Engine Roundup）。用户只需提交一次搜索请求，由元搜索引擎负责转换处理后提交给多个预先选定的独立搜索引擎，并将从各独立搜索引擎返回的所有查询结果，集中起来处理后再返回给用户。第一个元搜索引擎，是 Washington 大学硕士生 Eric Selberg 和 Oren Etzioni 的 Metacrawler。元搜索引擎概念上好听，

但搜索效果始终不理想，所以没有哪个元搜索引擎有过强势地位。

DEC 的 AltaVista(2001 年夏季起部分网友需通过 p-roxy 访问，无 p-roxy 可用 qbseach 单选 altavista 搜索，只能显示第一页搜索结果)是一个迟到者，1995 年 12 月才登场亮相（AltaVista Public Beta Press Release ）。但是，大量的创新功能使它迅速到达当时搜索引擎的顶峰。Altavista 最突出的优势是它的速度（搜索引擎 9238：比较搞笑，设计 altavista 的目的，据说只是为了展示 DEC Alpha 芯片的强大运算能力）。

而 Altavista 的另一些新功能，则永远改变了搜索引擎的定义。

AltaVista 是第一个支持自然语言搜索的搜索引擎，AltaVista 是第一个实现高级搜索语法的搜索引擎（如 AND, OR, NOT 等)。用户可以用 AltaVista 搜索 Newsgroups（新闻组）的内容并从互联网上获得文章，还可以搜索图片名称中的文字、搜索 Titles、搜索 Java applets、搜索 ActiveX objects。AltaVista 也声称是第一个支持用户自己向网页索引库提交或删除 URL 的搜索引擎，并能在 24 小时内上线。AltaVista 最有趣的新功能之一，是搜索有链接指向某个 URL 的所有网站。在面向用户的界面上，AltaVista 也作了大量革新。它在搜索框区域下放了“tips”以帮助用户更好的表达搜索式，这些小 tip 经常更新，这样，在搜索过几次以后，用户会看到很多他们可能从来不知道的的有趣功能。这系列功能，逐渐被其它搜索引擎广泛采用。1997 年，AltaVista 发布了一个图形演示系统 LiveTopics，帮助用户从成千上万的搜索结果中找到想要的。

然后到来的是 HotBot。1995 年 9 月 26 日，加州伯克利分校 CS 助教 Eric Brewer、博士生 Paul Gauthier 创立了 Inktomi（UC Berkeley Announces Inktomi），1996 年 5 月 20 日，Inktomi 公司成立，强大的 HotBot 出现在世人面前。声称每天能抓取索引 1 千万页以上，所以有远超过其它搜索引擎的新内容。HotBot 也大量运用 cookie 储存用户的个人搜索喜好设置。(Hotbot 曾是随后几年最受欢迎的搜索引擎之一，后被 Lycos 收购）

Northernlight 公司于 1995 年 9 月成立于马萨诸塞州剑桥，1997 年 8 月，Northernlight 搜索引擎正式现身。它曾是拥有最大数据库的搜索引擎之一，它没有 Stop Words，它有出色的 Current News、7,100 多出版物组成的 Special Collection、良好的高级搜索语法，第一个支持对搜索结果进行简单的自动分类。（2002 年 1 月 16 日，Northernlight 公共搜索引擎关闭，随后被 divine 收购，但在 Nlresearch，选中"World Wide Web only"，仍可使用 Northernlight 搜索引擎）

1998 年 10 月之前，Google 只是 Stanford 大学的一个小项目 BackRub。1995 年博士生 Larry Page 开始学习搜索引擎设计，于 1997 年 9 月 15 日注册了 google.com 的域名，1997 年底，在 Sergey Brin 和 Scott Hassan、Alan Steremberg 的共同参与下，BachRub 开始提供 Demo。1999 年 2 月，Google 完成了从 Alpha 版到 Beta 版的蜕变。Google 公司则把 1998 年 9 月 27 日认作自己的生日。

Google 在 Pagerank、动态摘要、网页快照、DailyRefresh、多文档格式支持、地图股票词典寻人等集成搜索、多语言支持、用户界面等功能上的革新，象 Altavista 一样，再一次永远改变了搜索引擎的定义。

在 2000 年中以前，Google 虽然以搜索准确性备受赞誉，但因为数据库不如其它搜索引擎大，缺乏高级搜索语法，所以使用价值不是很高，推广并不快。直到 2000 年中数据库升级后，又借被 Yahoo 选作搜索引擎的东风，才一飞冲天。

Fast（Alltheweb）公司创立于 1997 年，是挪威科技大学(NTNU)学术研究的副产品。1999 年 5 月，发布了自己的搜索引擎 AllTheWeb。Fast 创立的目标是做世界上最大和最快的搜索引擎，几年来庶几近之。Fast（Alltheweb）的网页搜索可利用 ODP 自动分类，支持 Flash 和 pdf 搜索，支持多语言搜索，还提供新闻搜索、图像搜索、视频、MP3、和 FTP 搜索，拥有极其强大的高级搜索功能。

Teoma 起源于 1998 年 Rutgers 大学的一个项目。Apostolos Gerasoulis 教授带领华裔 Tao Yang 教授等人创立 Teoma 于新泽西 Piscataway，2001 年春初次登场，2001 年 9 月被提问式搜索引擎 Ask Jeeves 收购，2002 年 4 月再次发布。Teoma 的数据库目前仍偏小，但有两个出彩的功能：支持类似自动分类的 Refine；同时提供专业链接目录的 Resources。

Wisenut 由韩裔 Yeogirl Yun 创立。2001 年春季发布 Beta 版，2001 年 9 月 5 日发布正式版，2002 年 4 月被分类目录提供商 looksmart 收购。wisenut 也有两个出彩的功能：包含类似自动分类和相关检索词的 WiseGuide；预览搜索结果的 Sneak-a-Peek。

Gigablast 由前 Infoseek 工程师 Matt Wells 创立，2002 年 3 月展示 pre-beta 版，2002 年 7 月 21 日发布 Beta 版。Gigablast 的数据库目前仍偏小，但也提供网页快照，一个特色功能是即时索引网页，你的网页刚提交它就能搜索（注：这个 spammers 的肉包子功能暂已关闭）。

Openfind 创立于 1998 年 1 月，其技术源自台湾中正大学吴升教授所领导的 GAIS 实验室。Openfind 起先只做中文搜索引擎，曾经是最好的中文搜索引擎，

鼎盛时期同时为三大著名门户新浪、奇摩、雅虎提供中文搜索引擎，但 2000 年后市场逐渐被 Baidu 和 Google 瓜分。2002 年 6 月，Openfind 重新发布基于 GAIS30 Project 的 Openfind 搜索引擎 Beta 版，推出多元排序(PolyRankTM)，宣布累计抓取网页 35 亿，开始进入英文搜索领域，此后技术升级明显加快。

北大天网 是国家"九五"重点科技攻关项目"中文编码和分布式中英文信息发现"的研究成果，由北大计算机系网络与分布式系统研究室开发，于 1997 年 10 月 29 日正式在 CERNET 上提供服务。2000 年初成立天网搜索引擎新课题组，由国家 973 重点基础研究发展规划项目基金资助开发，收录网页约 6000 万，利用教育网优势，有强大的 ftp 搜索功能。

Baidu 2000 年 1 月，超链分析专利发明人、前 Infoseek 资深工程师李彦宏与好友徐勇（加州伯克利分校博士）在北京中关村创立了百度（Baidu）公司。2001 年 8 月发布 Baidu.com 搜索引擎 Beta 版（此前 Baidu 只为其它门户网站搜狐新浪 Tom 等提供搜索引擎），2001 年 10 月 22 日正式发布 Baidu 搜索引擎。Baidu 虽然只提供中文搜索，但目前收录中文网页超过 9000 万，可能是最大的的中文数据库。Baidu 搜索引擎的其它特色包括：网页快照、网页预览/预览全部网页、相关搜索词、错别字纠正提示、新闻搜索、Flash 搜索、信息快递搜索。2002 年 3 月闪电计划（Blitzen Project）开始后，技术升级明显加快。

2002 年 12 月 24 日，雅虎称公司同意以大约 2.35 亿美元的价格收购搜索软件公司 Inktomi。

2003 年 1 月 18 日，Google 收购博客网站 Blogger.com 开发团队——网上出版软件开发商 Pyra Labs。

2003 年 2 月 19 日，Overture 服务公司表示，计划以 1.4 亿美元现金加股票从 CMGI 公司手中收购门户网站 AtaVista。

2003 年 2 月 26 日，Overture 同意以 1 亿美元收购位于挪威的 Fast Search and Transfer 公司的网络搜索部门。

2003 年 4 月 15 日，新浪与中国搜索联盟结成战略同盟，至此，中国已有数百家网站结成搜索联盟，以迎接国际巨头 Google 挺进国内市场后的巨大压力。

2003 年 4 月 21 日,第二大互联网搜索引擎提供商 Ask Jeeves 公司宣布对其 Ask.com 网站进行升级。Ask Jeeves 是仅次于 Google 的第二大搜索引擎，也是互联网上第五大搜索基地 (Google、雅虎、微软、AOL、Askjeeves) 。

2003 年 6 月 18 日，微软公司表示其正在加大研发新型互联网搜索引擎技术

的力度，包括对一款功能更先进的技术原型进行测试。

2003 年 7 月 12 日，从加利福里亚传来消息，Google 即将把总部从 Bayshore Parkway 搬迁至半里之遥的一个有四栋楼房的复式结构建筑中去，而这个建筑是由鼎鼎大名但目前却陷入困境的硅谷图象（Silicon Graphics）公司腾出来的。大卫.奎恩（David Krane）证实了这个消息，并解释说，这样能让公司现有的 800 多员工更好的分工合作和管理。

2003 年 7 月 13 日，百度推出图象搜索，新闻搜索两大搜索功能，以此来带动搜索流量。同时，辅以百度的搜索风云榜，使得百度的信息搜索及信息评估的作用更加突出

2003 年 7 月 15 日，全球最大的互联网公司雅虎宣布，以 16．3 亿美元收购在网络搜索服务上的竞争对手—Overture 公司，以期在同 Google 的竞争中取得优势。

2005 年 9 月，Google 正式进入中国市场。

## 1.4　通用网页检索和新闻检索的排序问题模型及研究方法

在本节中，作者将以数学模型或者符号对这些排序问题进行描述，同时说明这些问题所属的研究领域和常用方法，以期给读者一个宏观上的把握。相关研究的具体细节内容将在下面各章节的相关工作部分介绍

### 1.4.1　通用网页检索排序

#### 1.4.1.1　　问题描述

如 1.2 中所述，查询系统为每个页面生成了 $t$ 个特征值（包括查询相关特征和查询无关特征）。这样，每个页面可以表示成为一个 $t$ 维的特征向量 $x$，这一向量就构成了欧氏空间中的一个点。对于每个查询 $q$，查询系统会返回一组相关页面集：$\chi = \{x_{l1}, x_{l2}, ...x_{lm}, x_{u1}, x_{u2}, ...x_{un}\} \subset \mathbb{R}^t$，其中，前 $m$ 个点是有用户标注页面，后 $n$ 个点对应着没有用户标注的页面。与该页面集对应的标注向量为 $\gamma = [y_{l1}, y_{l2}, ..., y_{lm}, 0, 0, ..., 0]^T$，这里我们用 0 来代表未标注的页面。为方便表示起见，本文只考虑标注值为正的情况（很容易推广到标注为负值时的情况）。这些标注的取值通常为离散值，例如页面的分档和对应的标注可取为：

*完美，非常好，好，一般好，一般，不好，未标注*

*31　　15　　7　　3　　1　　0　　0*

这种 $r(k)=2^k-1$ 的取法是比较常见的一种做法。

定义 $d:\chi\times\chi\to\mathbb{R}$ 为 $\chi$ 上的两点 $x_i$ 与 $x_j$ 之间距离的度量标准：$d(x_i,x_j)$。

定义函数 $f:\chi\to\mathbb{R}$，$f$ 将每个点 $x_i$ 映射到一个实值 $f_i$ 上。

则函数 $f$ 就是一个排序函数，$f_i$ 就是 $x_i$ 的排序值。排序函数学习的问题就是从一系列查询集合的页面特征集 $X=\{\chi_q\}$ 和标注集 $Y=\{\gamma_q\}$ 中学习出函数映射 $f:\chi\to\mathbb{R}$。集合中没有标注的页面不参与训练或者做为负例参与训练。

#### 1.4.1.2　排序函数学习方法

解决此类问题的机器学习算法很多，最直观的方法就是基于回归（regression）的种种方法，包括：logistic regression, SVM based regression, neural network regression 等等。然而，影响最终搜索效果的只是页面之间的相对顺序关系，与具体的排序值无关。所以，另一类学习算法的思路是避开相对较困难的回归方法，采用基于点对的学习方法，有代表性的是 Rankboost[12]， RankerNet[13]等。

我们也可以从传统的 IR 角度出发，采用一些启发式，经验性的规则，直接生成排序函数，如采用 BM25 排序函数[14]等。

### 1.4.2　网络新闻流行度排序

#### 1.4.2.1　基本概念和讨论

我们首先给出事件的定义：

**定义 1.3 ---- 事件：**按照美国 DARPA 的 TDT（Topic Detection and Tracking）组织的定义[8]，事件指的是在某一（几）个地点，某一特定时间（段）内，由某些人参与的行为，以及这些行为所需的一切必要条件。

下面是流行度的定义。

**定义 1.4 ---- 流行度：**对新闻而言，其流行度是网络上多个站点对其认可的程度的综合。对新闻站点而言，其流行度是其上发布的新闻的流行

度所反向产生的。

新闻和站点之间的流行度存在着互增强（reinforcement）的关系，即：

在排名较高的站点上发布的新闻，其流行度高的可能性较大；而发布高度流行新闻比较多的站点，其流行度排名也应该较高。

新闻的流行度可以反映在两方面：

1)　新闻流行度越高，越多的站点会同时报导该新闻，反之亦然。

2)　如果新闻在新闻首页上占据的视觉区域越重要，则该站点对该新闻的推介程度越高，反之亦然。

1)中的规律已经被这方面前人的工作所发现和证实了[7]，而 2 ）中的规律则是我们的独创。我们的方法是基于如下的假设。

**假设 1.1：**每条新闻在新闻首页上所占据的视觉区域属性（位置，大小，字体，有无图片等）反映了该新闻在该首页上的重要程度。

这个假设是明显成立的。例如，新浪新闻的头条明显会比排在下方的新闻的流行度强。所以我们可以设法利用这些视觉信息而不是像前人那样[7]简单的认为来自于同一新闻站点的新闻会受到站点相同的推介。关于这方面技术的细节我们将在第三章中详细讨论。

#### 1.4.2.2　基本模型和改进模型

基于以上的讨论，读者已经对新闻站点的发布模式有所了解。下面我们将先介绍在前人工作中[7]使用较多的新闻站点和新闻关系的模型，然后在其基础上提出我们的改进模型。

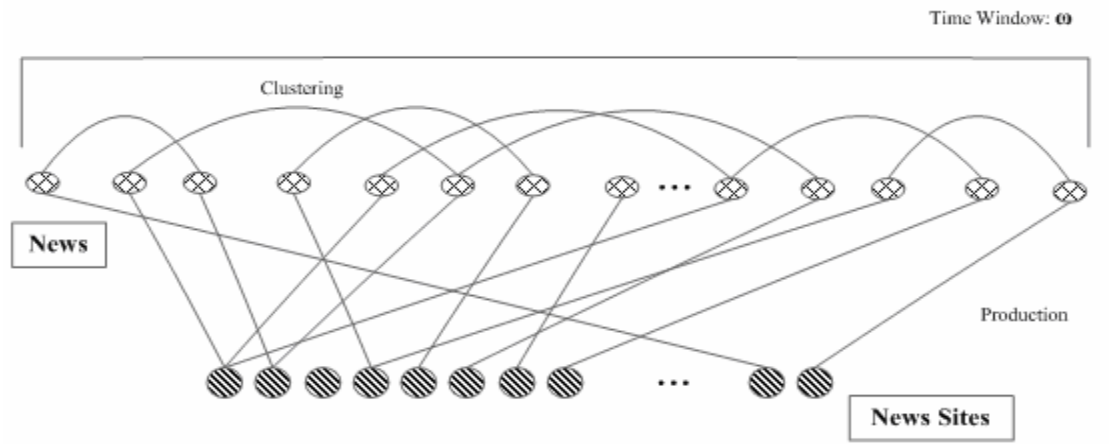


图 1.3 新闻站点与新闻排序模型

如图 1.3 所示，在给定的时间窗 ω 内，新闻发布的过程可以用一个无向图 $G=(V,E)$ 来表示。其中顶点集 $V=S\bigcup N$，$S$（蓝色）是代表新闻站点的顶点，而 $N$（红色）是代表时间窗 ω 内的新闻。同样，边的集合 $E$ 也可以分为两个无交连集 $E_1$ 和 $E_2$。$E_1$ 是连接顶点集 $S$ 和 $N$ 的无向边的集合，它代表了新闻发布和推介的关系，其上的权重代表了新闻站点对新闻的推介程度。$E_2$ 是新闻顶点之间的无向边的集合，代表了相似新闻的聚类过程，其上的权重代表了新闻两两之间的相似度。$S$ 中的顶点完全覆盖了 $N$ 中的顶点，即 $\forall n\in N,\exists s\in S$，使得 $(s,n)\in E_1$。这样，通过这个模型，我们可以得到新闻和新闻站点的排序。

由于网络上新闻站点发布新闻的模式是实时的将所有的新闻发布在**新闻首页**上，同时相似新闻实际上是在报导同一新闻事件，所以我们对上述模型进行如下改进。

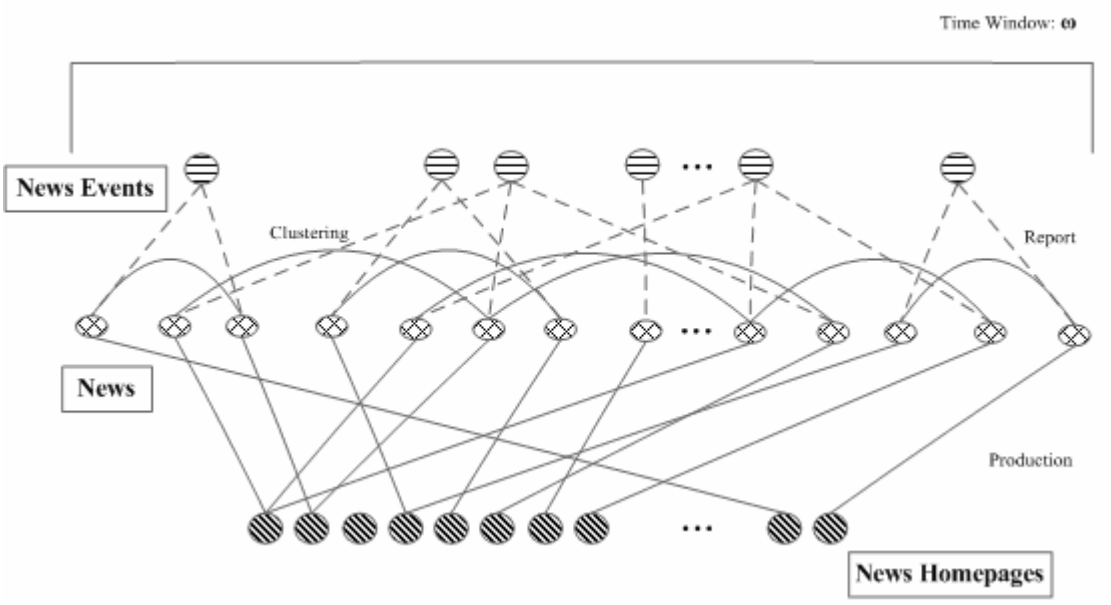


图 1.4 新闻首页，新闻与新闻事件的三层排序模型

如图 1.4 所示，我们将新闻发布的节点替换成为新闻首页，这样更符和实际情况，而且对新闻首页的排序比单纯的新闻站点排序更加合理。例如，新浪的“世界新闻”首页可能是所有站点中排名最高的，但是雅虎的

财经类新闻比新浪的排名更高[1]。通过对新闻首页的排序，我们可以得到每一类别中最热门的网站，同时也可以用来指导网站建设，以提高排名较低的栏目首页。

同时，由于每一篇新闻都是在报导某一新闻事件，内容相近的新闻实际上是在报导相同的新闻事件。我们在排序模型中引入事件节点（红色），虚线代表着报导的关系。这样，我们在得到新闻排序的同时也将得到新闻事件的排序，这显然有着非常重要的语义价值和实用价值。

#### 1.4.2.3　排序算法

新闻和首页之间的流行度排序存在着互增强（reinforcement）的关系，即：

在排名较高的首页上发布的新闻，其流行度高的可能性较大；而发布流行度高的新闻比较多的首页，其流行度排名也应该较高。

这样，基于单边增强的 PageRank[9]（L. Page, 1998）算法不适用于我们的问题。而由 Cornell 的 J. Kleinberg 提出的 HITS（Hyperlink-Induced Topic Search）算法[10]则可以比较好的解决这一问题。

### 1.4.3　网络新闻的用户个性化定制排序

#### 1.4.3.1　问题描述

给定一组用户 $U = \{u_1, u_2, ...u_p\}$，对任一用户 $u_i$，我们有其新闻浏览历史记录 $h_i = \{n_{i_1}, n_{i_2}, ..., n_{i_s}\}$。其中，$n_{i_k}$ 代表一篇新闻文档，拥有 3 个属性：

1) 用户阅读发生时间 $t_{i_k}$。即用户在 $t_{i_k}$ 时间阅读了该新闻。
2) 内容描述 $v_{i_k}$。新闻内容的描述方式取决于所采用的语言模型，比较常用的方法是采用向量空间模型（Vector Space Model）[11]，将新闻内容用一个或多个向量表示。
3) 新闻所属类别。这是网络新闻的一个重要特点，即每一条新闻都有确定的类别，如政治，体育等。新闻的类别反映了人们在对新闻知识分类的固定模式。新闻类别是以树的形式分层嵌套的，例如体育下包含篮球，足球等，篮球下又有可能包括 NBA，

[1] 此处是为了说明问题假设举例，可能与实际情况不符

国内篮球等。

每个周期 $T$（通常是 1 天）内，外部新闻源会产生一系列新的新闻文档 $N=\{n_1,n_2,...n_q\}$，个性化推介排序算法的任务就是由这些用户的历史浏览记录，为每个用户生成 $N$ 的最优排序，以期符合用户的兴趣需求。实际上，这一排序问题也是一个过滤得过程，由于用户的总阅读量有限，排序较低的新闻相当于被过滤掉了。所以，本文题属于信息过滤（Information Filtering）领域。

#### 1.4.3.2　单用户的排序算法

所谓的单用户问题就是指只有当前用户的浏览记录可得的情况下，如何生成个性化推介排序。这里我们指出一个假设。

**假设 1.2**：用户的浏览历史，反映了用户的兴趣和兴趣的变迁。

基于这一假设，我们就可以设法从用户的浏览记录中对用户兴趣进行建模（User Profile Model），将新的新闻与此模型进行对比，以判断其符合用户兴趣的程度，进而根据这一相关度进行排序。这一类方法属于信息过滤下的自适应过滤领域（Adaptive Filtering），也叫基于内容的过滤（Content based Filtering）。

用户的兴趣一般分为长期兴趣和短期兴趣两种。所谓长期兴趣是指用户兴趣模型中比较稳定，变化较为缓慢的部分，比如某爱好运动用户对体育新闻的兴趣；短期兴趣是指会随着当前热点事件的产生而改变的兴趣，比如上述用户的兴趣会从世界杯转向亚运会。现有的一些用户兴趣模型大都考虑了从两个层次上对用户兴趣进行建模，当前的主流用户兴趣模型也基本都是层级型结构。然而，这些模型对用户的短期兴趣建模的精确程度并不高，因而在用户端起兴趣变化比较快的情况下，算法的准确度并不理想。

#### 1.4.3.3　多用户的排序算法

所谓的多用户问题就是除当前用户之外还有其他多个用户的浏览记录可得的情况下，如何生成个性化推介排序。这里我们指出另外一个假设。

**假设 1.3**：当前用户感兴趣的内容，可由与其相似的用户推断得出。

基于这一假设，我们如果能设法找到与当前用户相似的用户群体，就

可以从整个群体的感兴趣内容中来推断当前用户的感兴趣内容。这一类方法属于信息过滤下的协同过滤（Collaborative Filtering）领域。

在此类方法中，很关键的一步就是确定与当前用户相似的用户群体。由于个性化新闻排序这一问题的特殊性，使得用户的浏览记录以及浏览过的新闻文档内容都可以被用来计算用户相似度。现有的主流协同过滤算法可以做到这一点。然而，我们还可以设法利用用户兴趣模型（User Profile Model）中蕴含的信息，进一步辅助用户相似度的计算。这也是作者在多用户排序问题下的创新点所在。

## 1.5　作者的工作和贡献

本文中所涉及的作者的工作主要在如下三个方面。

### 1.5.1　通用网页检索排序

考虑到人力成本和时间成本，当前通用网页检索的排序函数学习中，有标注训练样本普遍不足。在实际的商用搜索引擎[2]的大规模的训练样本集中，每个查询的相关文档集中，有标注的文档只有 20 个左右，而未标注的文档约有 2000 个。因而，排序函数的学习训练受到了很大的限制。为解决这一问题，作者提出了通过基于流形排序算法的半监督学习机制，将少量已标注样本的相关值传递到其他未标注样本上，进而大大增加排序函数学习可用样本数。基于此种相关度传递机制——MRBRP（Manifold Ranking Based Relevance Propagation），作者提出了一套完整的排序函数的学习框架，将相关度传递的结果用于任意排序学习算法，进而改进训练的效果。此框架具有极大的灵活性，可使用 MRBRP 来配合任何最先进的排序函数学习算法。

在主流商用搜索引擎的海量数据集合上的实验表明，此种学习框架可以显著提高排序函数的性能。

### 1.5.2　网络新闻流行度排序

在新闻流行度排序算法的研究中，作者通过一个三分图模型对新闻站点，新闻和事件之间的关系进行了建模，新闻，站点和新闻事件之间构成了相关增

[2] 考虑知识产权的问题，此处隐去了具体名称

强的关系，该三分图模型上的平衡解为最终的排序结果。作者创造性的提出了用新闻在新闻网站首页的视觉分布来表示该新闻在该新闻站点中的重要程度；同时考虑多站点之间对新闻报道的同一性，采用了多新闻站点之间相关新闻的投票最终得到新闻和站点的关系。该算法可以同时得到新闻，新闻事件，以及新闻站点的排序。同时，新闻的时效性问题在本算法中也有反映。

在多个实际新闻站点数据上的实验结果表明此算法具有良好的效果和用户体验。

### 1.5.3　网络新闻的用户个性化定制排序

考虑到用户的个人兴趣，现有的做法是用一个用户兴趣模型进行建模描述。作者针对现有的主流用户兴趣模型在描述用户在新闻事件层次上的兴趣的不足，提出了一个对应固定概念分类有着固定的节点层，对应着事件具有动态节点层的全新的层级结构用户兴趣模型，可以同时描述用户在概念类别和具体事件上的兴趣。此层级模型的节点结构和节点内容还可以随着用户的阅读内容变化而改变，能更好的跟踪用户的兴趣。

为了利用相似用户群体的兴趣信息，作者又提出了一种基于此模型的层级结构的改进协同滤波算法，能更好的发现相似用户群体。

在实际的用户新闻浏览数据上的实验结果表明，本用户兴趣模型和协同滤波算法可显著的提高用户个性化新闻推介的准确度。

## 1.6　本文的组织

第 2 章将介绍作者在大规模通用网页检索排序算法方面的工作。第 3 章将介绍作者在网络新闻流行度排序方面的工作。第 4 章将介绍作者在用户兴趣模型和协同滤波方面的工作。第 5 章对全文做一总结。

# 第2章　基于相关度传递增广的排序函数学习

## 2.1　本章引论

如同 1.2 和 1.5.1 中所述，在大规模网页检索的环境下，针对每个用户查询，搜索引擎的查询服务模块会返回大量的文档。为了生成最终的搜索结果，我们需要一个效用函数来评价每篇文档的相关分值。所有的文档将按照它们的分值进行排序。我们称这一效用函数为排序函数。

如 1.2 中所述，搜索引擎为每篇文档生成一系列的特征，包括查询相关特征和查询无关特征，并以之来表示每一篇文档。由这些特征生成排序函数的方法有很多。我们可以采用 IR 领域中传统的启发式方法 BM25[14]，我们也可以采用机器学习的算法从这些特征值和其对应的排

序标注中学习出排序函数。基于机器学习理论的方法可以分为如下两类：基于回归的方法和基于成对分解的方法。

排序函数实际上可以看作从特征值到排序值的一个影射，所以函数回归的方法是一个很直接的解决方案，下面介绍前人在这方面的工作。在[15]中，Herbrich 等人将排序函数的学习转化为一个有序回归的问题，即排序值是取自某个特定的有序序列，其算法学习从特征向量到此类排序值的映射。此时，排序值序列中的排序分界对最终排序函数的有着决定性的影响，而合理的选取排序分界则比较困难。Crammer 和 Singer[16]提出了一种在线算法 PRank，该算法采用了一个加权线性的感知器将特征向量影射到一个实值。算法的关键问题是权值的学习。 基于 PRank 方法，Harrington[17]提出了一个很简单但是非常有效的扩展，即通过对多个 PRank 模型求平均的方法来估计 Bayes 点，相当于用 version space 的重心来近似表示 Bayes 点。这些基于回归的方法的一个最主要的问题是：在排序问题中，最重要的是相对的排序关系而不是绝对的排序值取值。将排序转化为回归实际上是增加了问题的复杂程度。由于解决精确映射问题的难度要大于排序关系的学习，所以另一种思路是通过学习成对样本的顺序关系来确定排序函数。

基于样本对的学习方法不考虑每一个样本的排序取值，而是集中考虑样本对之间的排序值差别。在[18]中，Cohen 等人提出了一个两阶段的学习算法：在

阶段一，通过给定的样本对学习出一个概率排序函数。在阶段二，通过最优化排序价值函数，将阶段一中的排序函数应用到新的样本上。在[19]中，Freund 等人将 boosting 方法和样本对的排序问题结合起来，提出了 RankBoost 方法。Chis[20]等人提出了一个定义在样本对上的价值函数。他们提出了一种通过价值函数梯度下降的方式学习最终的排序函数的方法----RankerNet。此类方法最主要的问题是样本对组合爆炸。在网页检索问题中，由于样本对的构建是在各个查询内部发生的，而每个查询所包含的文档数目是有限的，所以不会出现组合爆炸的问题。

从上述讨论我们可以看出，文档的特征值和用户标注的相关排序值是排序函数学习问题的两大基本要素。如同 1.4.1 中所述，，为每篇文档生成一系列的特征值相对很容易，然而由于每个查询的文档数目（数千）远远的超出了人们可以承受的标注能力（数十）。所以，所有的基于机器学习理论的学习方法都面临训练样本不足的问题，这将极大的影响学得的排序函数的泛化能力。

为了解决标注不足的问题，Brinker[21]等人提出了一种主动学习（active learning）算法，通过某些启发式规则选取最具信息量的训练样本进行训练。他们的实验结果表明这样的方法的确可以显著的减少训练所需样本数。然而，启发式规则的选去限制了此方法在大规模通用网页检索领域的应用，同时也限制了其与其他学习方法的结合使用。

我们对训练样本不足这一问题的解决方案是受到了基于内容的图像检索领域（CBIR）中 MRBIR（Manifold Ranking Based Image Retrieval）[27]算法的启发。MRBIR 是一个很有效的相关度反馈框架。同过借助某些半监督学习的方法如流行排序（Manifold Ranking），我们可以将少量的用户标注传递到所有的未标注文档上，这样，所有的数据都可以用做训练样本。基于此，我们的排序函数学习框架如下所述：

首先，通过基于流行排序的相关度传递（Manifold Ranking Based Relevance Propagation, MRBRP）为所有的未标注样本估计排序值。这一过程可由如下两幅图说明。

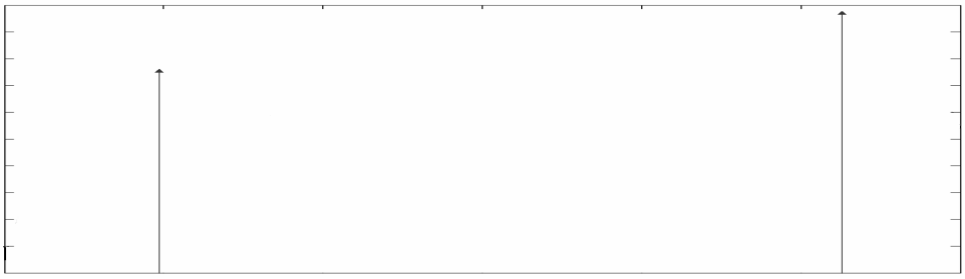

图 2.1 进行相关度传递之前，只有 2 个已标注样本

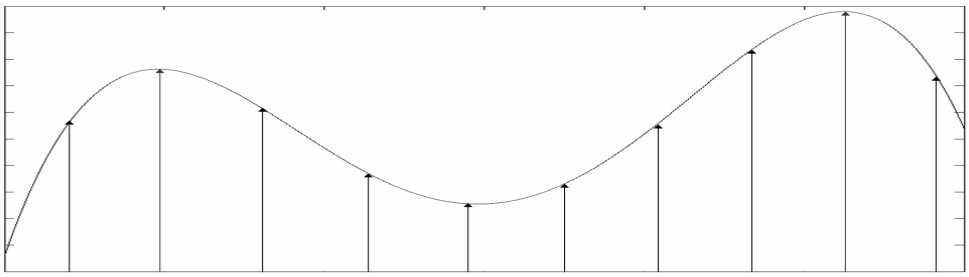

图 2.2 相关度传递之后，所有的未标注样本都得到了估计的排序值

接下来，我们就可以通过任何排序函数学习算法从得到增广的训练样本中学得最终的排序函数。当得到最终的排序函数之后，我们可以直接将其应用到未知的新查询上进行排序。

我们的工作主要贡献是：

1. 提出了一个全新的半监督学习机制通过相关度传递来增广训练集合。
2. 我们提出的学习框架非常灵活，可以采用任何学习算法。对相关度传递和学习算法中任意一步的改进都会改善最终的学习效果。

本章的剩余内容将如下组织：2.2 节给出由相关度传递而增广的排序函数学习框架，包括相关度传递方案，排序函数学习算法和最终的排序计算；2.3 节讨论在大规模网页索引环境下算法实现的一些问题；2.4 节给出实验数据和结果以证明我们方法的效果；最后是本章小结。

## 2.2　由相关度传递而增广的排序函数学习框架

### 2.2.1　总述

本章剩余部分将使用 1.3.1.1 中排序问题描述所采用的符号表示。

传统的学习算法只采用有标注样本的特征 $X_l$ 和标注 $Y_l$ 或者简单的将未标注样本集 $Y_u$ 中的样本当作负例来处理。我们的方法与此不同，会使用 $X_l$ ， $X_u$ 和 $Y_l$ 中所有的信息。我们不直接采用 $Y_u$，而是通过 MRBRP 为 $Y_u$ 中的样本合理的估计排序值，这样，我们可以得到一个更精确的从 $X$ 到 $Y$ 的映射。图 2.3 中说明了我们的学习框架和传统学习算法框架的区别。

我们提出的学习框架包括如下 3 个主要过程：1）相关单独传递（目前是 MRBRP）；2）排序函数学习；3）排序预测。通过相关度传递，所有未标注样本得到了估计标记 $Y_e$。这样，我们得到了一个从 $X$ 到 $Y$ 完全映射，所有的数据都可以在学习过程中被用作训练样本。通过排序函数学习过程，我们可以得到排序函数。接下来，我们将这一排序函数应用在新的未知数据上进行排序预测。对新的未知数据而言，我们只知道其对应的 $X$ ，即特征值信息，而不知道排序值 $Y$ 。

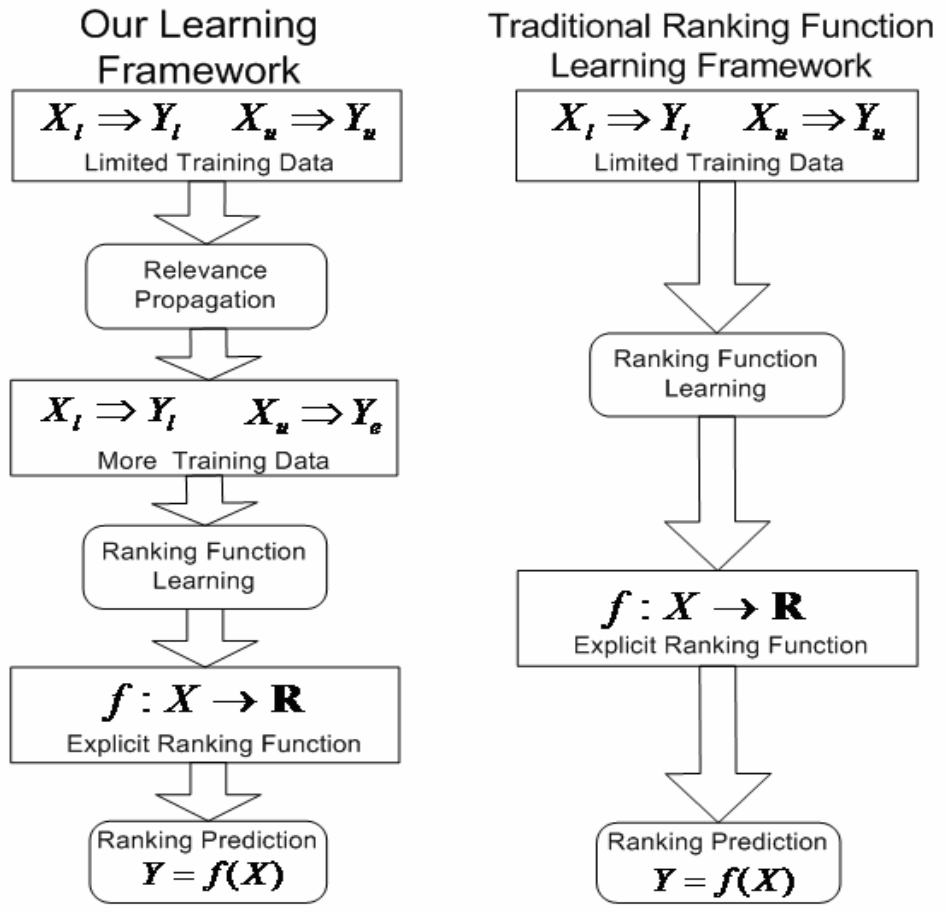


图 2.3 我们的排序函数学习框架和传统的排序函数学习框架对比

从图 2.3 中可以看出，在我们的学习框架中，相关度传递过程和排序函数学习过程是完全独立的。所以，对任一步骤的改进都可以改进框架最终的学习效果。

### 2.2.2　相关度传递

在这一过程中，我们研究文档特征空间中点对之间的关系来描述已标注文档和未标文档之间的相关程度。为了实现这一目的，我们采用了流行排序算法[22][23]。这一算法通常被用作对数据点沿着数据内部的流形（Manifold）进行排序。数据的流形是由所有数据点之间的相互关系所决定的。每一篇文档，不管是已标定的还是未标定的，构成了特征空间中的一个点。我们可以通过研究这种所有数据点之间的关系来衡量标定文档和未标定文档之间的相关程度。我们将在本节中介绍这一算法。

#### 2.2.2.1　　基于流行排序的相关度传递

我们这里讨论的是在单个查询内部做相关度传递问题，很容易将这一算法推广到多个查询之间传递的情况。

图 2.4 是 MRBRP 算法的流程。

1. 按照某种规则连接某些顶点生成图 $G$。$G$ 的生成规则参见 2.2.2.3
2．根据某种距离函数确定图 $G$ 的关联矩阵 $W$，关于相似度函数的细节参见 2.2.2.3。令 $W_{ii}=0$ 。
3．用 $S=D^{-1/2}WD^{-1/2}$ 来对称的正则化 $W$。$D$ 是一个对角阵，其对角线上处在 $(i,i)$ 位置的元素等于 $W$ 的第 $i$ 行之和。
4．叠代 $f(t+1)=\alpha Sf(t)+(1-\alpha)y$ 直至收敛，$\alpha$ 是取值在 $[0,1)$ 之间的参数，$y$ 是原始的标注向量, $f(0)=y$。
5．令序列 $\{f(t)\}$ 的极限为 $f^*$，$f^*$ 就是相关度传递的结果。

图 2.4 基于流形排序的相关度传递算法流程

对这个算法的直观解释是：首先建立一个加权图，每一篇文档对应图中的一个顶点，边的权值代表顶点文档的相似度；为每个顶点分配其标注值，未标注的顶点取值为 0。所有的顶点通过与其相连的边，将其取值传播到相邻的顶点上。这一传播过程不断叠代，直到全图达到稳定状态。此时所有的未标注点都

会得到一个最终的取值。我们通过将关联矩阵的对角线元素设为 0，避免了自增强的问题。排序值传递的结果反映了所有的数据点之间的关系。通常来说，相距较远的点会有不同的排序值，除非他们处于同一个聚类（cluster）中，通过类中的其他点相连接。相距较近的点通常会有相似的排序值，除非他们处在不同的聚类中。在我们的问题中，未标注文档得到的传播结果反映了它相似于正例的概率，其分值越高，该文档为正例的可能就越大。

#### 2.2.2.2　传递算法分析

[10]中的定理保证了序列$\{f(t)\}$会收敛到：

$$f^* = (1-\alpha)(I-\alpha S)^{-1} y \tag{2-1}$$

对上式进行泰勒展开并忽略掉常数乘子，可得：

$$\begin{aligned} f^* &= (I-\alpha S)^{-1} y \\ &= (I+\alpha S+\alpha^2 S^2+\ldots) y \\ &= y+\alpha S y+\alpha S(\alpha S y)+\ldots \end{aligned} \tag{2-2}$$

通过上式，我们可以从另外一个角度来理解 MRBRP。$f^*$可以理解为一个无限序列的求和。序列的第一项是原始的标注$y$，第二项是标注值经过一次传递的结果，第三项是标注值又经过一次传递的结果。这样，通过这一系列的传递，数据中所包含的流形信息就会被引入到最终的排序结果里来。

这些文档在特征空间中构成了无向的加权图。通过对此图作归一化处理，我们可以估计出一个调和函数。通过此调和函数，可以为每个未标注的点计算出排序值。这一调和函数和 MRBRP 的收敛解相同。这给了我们对 MRBRP 的另一种理解：通过相关度传递的迭代过程，我们最终得到的是一个调和函数（无显示函数表示），我们称之为隐式的排序函数。在接下来的函数学习过程将把此函数的显示表达学习出来。

#### 2.2.2.3　加权图的构建

流刑排序算法的一个关键问题就是关联矩阵 $W$ 的定义[24]。$W$ 的定义包括两个基本要素：1）加权无向图的建立方式；2）点对间相似度的度量方式。

构建加权图的关键是保证图的稀疏性的同时为每个顶点保留尽量多的连接

度。这样得道的图可以揭示内嵌的主要流形，同时不会引入太多噪声。我们尝试了如下几种构建加权图的方式：全联通（Full Connected Graph, FCG），最小生成树（Minimum Spanning Tree, MST），k 近邻（k Nearest Neighbour, kNN）。通过比较，我们最终选择了 k 近邻来构建图。我们的实验结果表明参数 k 取 10 的时候，相关度传递可以得到比较好的结果。

通过 k 近邻的方式得到的图未必是连通的，可能由几个独立的聚类组成。这种情况下，一个不可避免的问题是图上的某些点经过传播后得不到任何相关值。这种情况是合理的，因为这些点的出现是因为其没有连接到正例的顶点上，所以其最终被划分到负利是非常合理的。然而在我们的实验中，这种情况很少发生，因为有标注的文档涵盖了标注的各个层级，在 k 近邻形成的图中，不同的聚类实际上是对应着不同的标注层级。所以，通过相关度传递，每个聚类中所有的文档都能得到一定的排序值。

我们尝试了多种文档相似度的计算方法，包括：余弦相似度（Cosine Similarity），曼哈顿距离（Manhattan Distance），欧几里德距离（Euclidean Distance）等。实验的结果表明曼哈顿距离的效果要优于其它两种度量方式，这与 MRBIR[27]中的实验结果和[25][26]中的结论是相符的。我们最终采用了拉普拉斯核来定义曼哈顿距离，以表示图中顶点的相似度：

$$k_L(x_i, x_j) = \prod_{l=1}^{t} \frac{1}{2\sigma_l} \exp(-\left|x_{il} - x_{jl}\right| / \sigma_l) \qquad (2\text{-}3)$$

其中，$x_{il}$ 和 $x_{jl}$ 是 $x_i$ 和 $x_j$ 的第 $l$ 维特征，$t$ 是特征空间的维度，$\sigma_l$ 是反映了不同维度特征取值范围的参数。由于顶点的相似度就是边的权重，所以我们有：

$$W_{ij} = k_L(x_i, x_j) = \prod_{l=1}^{t} \exp(-\left|x_{il} - x_{jl}\right| / \sigma_l) \qquad (2\text{-}4)$$

此处忽略了常数系数 $1/2\sigma_l$，因为由此引起的在关联矩阵 $W$ 上的效果会被归一化的步骤 $S = D^{-1/2}WD^{-1/2}$ 所抵消，所以不会影响最终的传播结果。

#### 2.2.2.4　查询内传递和查询间传递

如同 1.3.1.1 中所述，在排序函数学习的问题中，每篇文档由一个 $t$ 维的特征向量描述，构成了欧式空间 $\mathbb{R}^t$ 中的一个点。排序函数的学习实际上是从欧式空间 $\mathbb{R}^t$ 到实值排序值映射的学习。如此建模实际上包含了如下两个假设：

**假设 2.1**：所有的点都处在同一个欧式空间中，也就是说，对所有点而言，

每一维特征的定义都相同。由于对所有文档而言，特征的定义都是一致的，所以这一假设是成立的。

**假设 2.2**：由所有的文档所反映出的排序规律是恒定的，换句话说，排序函数的学习实际上是探究多个查询共有的排序规律。

基于假设 2.1，单个查询所包含的文档实际上构成了由多个查询的文档所对应的图上的一个子图。当单个查询内部的文档数目较少时，这一子图有可能无法覆盖含有排序规则的主要流形。在这一子图上进行相关度传递（我们称之为查询内传递）无法为函数学习训练提供足够好的数据。在这种情况下，我们需要更多的文档加入到子图中，以揭示全局的排序规则。为达到这一目的，我们可以在多个查询的文档共同组成的集合上进行相关度传递操作，我们称此为查询间传递。

基于假设 2.2，查询间传递的操作是合理的，因为这一操作的目的是查找多个查询中所共有的排序规律。然而，由于相关度传递的实际上是用户的标注信息，所以我们还需另一个假设来保证查询间传递的合理性。

**假设 2.3**：用户标定对所有的查询都是一致的，也就是说，不同查询中相同的标定等级意味着同样的排序值。

由于标定级别的定义对所有的查询都一样，所以假设 2.3 是成立的。然而，考虑到标定者个人的偏好不同，所以不同标定者的的标定值的分布未必相同。可以通过设定严格的标定规则和进行一些培训来解决这一问题。根据我们的实验结果，当使用的文档数目较多时，少量的噪声标定对相关度传递的结果影响很小。

相关度传递很容易由单个查询推广到多个查询的情况，算法不必作任何的改动。我们只需选取由多个查询的文档所构成的无向加权图作为相关度传递的对象既可。然而，我们很难确定应该选取多少个查询来构建此图。而且，由于随着查询数目的增加，查询间传递的计算量将增长的非常快。所以，查询内传递更加实用。

### 2.2.3　排序函数的学习

在上一步中，我们通过相关度传递得到了完整的排序值向量，接下来可以使用学习算法从这些被增广的训练数据中显示的学习出排序函数。

一般来说，排序函数的学习过程可以理解为图 2.3 中的黑盒过程。黑盒的输

入是文档的特征值数据和对应的排序值，黑盒的输出是排序函数。由于通过相关度传递得到的排序值是连续的实数，一些排序算法需要对这些值进行离散化处理，或者这些算法本身需要调整以能够处理连续值输入。至于选择哪种方式取决于具体的学习算法。

相关度传递算法本身对学习算法没有任何限制。任何排序函数学习算法都可以被用来配合使用。我们主要研究了两类算法的性能：

1. 回归算法。这类算法相对比较直接和简单，我们采用了 BP 神经网络来进行回归学习。
2. 样本对学习。RankerNet，RankBoost 和 RankSVM 等算法属于此类。我们研究了 RankerNet 和 RankBoost 算法在排序函数学习方面的应用。

除了灵活性这一优点，我们的学习框架的一大优势是相关度传递过程和排序学习过程是完全独立的。基于本框架的这一特性，我们相信对任一过程的改进都可以改进最终的学习效果。具体说来就是，采用更好的版监督学习方法来实现相关度传递可以生成更加精确的预估排序值，这样就为学习过程提供了更好的训练样本。而由于最终的排序结果是有学习过程产生的排序函数决定的，所以，更好的学习算法必然能改进最终排序结果。

### 2.2.4　排序计算

当排序函数的显式表达被学习出来之后，排序计算就非常直接了。排序函数可直接应用于仅特征值可得的新的未知文档。预测的过程可用下式表达：

$$Y = f(X) \qquad (2\text{-}5)$$

其中 $f$ 是学得的排序函数，$X$ 代表新的未知文档的特征值，$Y$ 是排序值的预测结果。

## 2.3　算法实现的一些问题

我们提出的学习框架可以完全离线运行，即给定训练数据集之后，离线的方式学习出排序函数，再将排序函数应用到在线排序中去。所以，从这个角度来说，本学习框架是没有可扩展性（scalability）问题的。这也是本方法和 MRBIR 相比的一大优势[27]。

然而，考虑到通用网页检索问题的规模通常都非常巨大，例如，训练集的文档数目通常在百万的级别。所以，时间上的效率还是很重要的，尤其是对超大规模的训练而言。

本算法的时间耗费通常包括两个方面：1）在相关度传递 MRBRP 阶段的耗费；2）在排序函数学习阶段的耗费。

对 MRBRP 而言，大规模的矩阵乘法和大规模无向图的构建占据了大部分的计算量。当采用了查询间传递的方式后，这一问题更加显著。下面是我们对这一问题的分析和解决方案：

1. 在进行 MRBRP 的时候，采用迭代的方式，从而避免大规模矩阵求逆的问题。因为矩阵求逆的最优解法 Gauss-Jordan 法需要更多的空间和时间消耗。采用迭代的方式和矩阵求逆的复杂度分别是 $O(I*n^2)$ 和 $O(n^3)$，$n$ 是文档的数目，$I$ 是迭代的次数。在我们的实验中，10 次迭代就可以保证收敛性了。
2. 前述几种建立无向图的方法的复杂度分析如下：
   全联通（FCG）：时间复杂度 -- $O(n^2)$；空间复杂度 -- $O(n^2)$。
   最小生成树（MST）：时间复杂度 -- $O(n^2)$；空间复杂度 -- $O(n)$。
   K 近邻（kNN）：时间复杂度 -- $O(kn)$；空间复杂度 -- $O(kn)$。
   由于 $k$ 远小于 $n$，所以采用 kNN 算法，我们可以得到一个在时间和空间上都比较经济的方案。同时，关联矩阵 $W$ 会变得很稀疏。这样，我们可以采用稀疏矩阵计算方法来进一步的提高效率。
3. 采用查询间传递的方式时，文档的数目随着查询数目的增加迅速的增加，迭代过程会变得非常慢。为了解决这一问题，我们只选择一定数量的查询进行查询间传递，而不是在所有的文档间进行传递。这样，我们在计算量和由多查询文档提供的信息量之间取了一个 trade off。实验证明查询间传递的效果比查询内传递要好。然而，在进行查询间传递时，现在还没有很好的办法来确定合并查询的数目。

## 2.4　实验

### 2.4.1　实验数据

我们试验中使用的数据来自一个实际的著名商用搜索引擎。每个查询内部

的文档数据来自于索引数据库。文档数据中所包含的查询相关特征来自于网页的四个域：链接描述文字（Anchor Text），URL，文档标题（title），和文本内容主体（body of text）。数据还包含一些附加的查询无关的特征如 PageRank 值等。每篇文档共有 640 个特征。在数据预处理阶段，我们将这些特征数据进行了归一化，以限制数据的分布范围，同时更便于学习算法的训练。实验数据共包括来自 12，000 个查询的文档。我们将这些查询随机打乱，从中选择了 2/3（8000 个查询）作为训练集，1/6（2000 个查询）作为校验集（validation set）以选择最优排序函数，1/6（2000 个查询）作为测试集用来测试排序函数的性能。为了减小数据量，我们为训练集中每个查询保留了 100 个未标注的文档，为校验集中每个查询保留了 137 个未标注文档，为测试集中每个查询保留了所有的未标注文档（2371）。

表 2.1 是关于实验数据的一些统计。

表 2.1 实验数据统计

| | Query Num | Doc Num | Labeled/Query | Unlabeled /Query |
|---|---|---|---|---|
| **Training Set** | 8000 | 9494945 | 18.7 | 100 |
| **Validation Set** | 2000 | 311684 | 18.8 | 137 |
| **Testing Set** | 2000 | 4779431 | 18.7 | 2371 |

每个查询中，有标注文档的排序值取值范围从 0（“not match at all”）到 5（“perfect match”），未标注文档的排序值取值为 0。这两个位置分别对应着用户最关注的区域和用户会完全浏览查找相关项的区域。

### 2.4.2 评价标准

我们采用 NDCG（Normalized Discounted Cumulative Gain）[28]指标来衡量排序的精确度。我们选择了在排名第 3 和排名第 10 位置上的 NDCG 值（NDCG@3 和 NDCG@10）作为评价指标。对给定查询 $q_i$，搜索的结果一般是按照排序函数输出的排序值降序排列。此时 NDCG 值将按照下式计算：

$$\mathcal{N}_i \equiv N_i \sum_{j=1}^{10} (2^{y_j} - 1) / \log(1 + j) \qquad (2\text{-}6)$$

其中 $y_j$ 是排名第 $j$ 的文档的原始用户标定值（如果为未标定文档，$y_j$ 为 0），

$N_i$ 是归一化常数，用来保证最优情况下（按照原始的用户标注从大到小排序）的 NDCG 取值为 1。需要注意的是未标注文档对 NDCG 计算中的求和没有影响，然而由于其的存在将会降低有标注文档的排序，所以会导致整体的 NDCG 值降低。另一点需要注意的是，$M_i = 1$ 基本不可能发生，即使对好的排序函数也是如此；因为未标注的文档中有可能有非常相关的样本被排序函数选中，排在了前 10 名。

### 2.4.3　实验设计

我们的实验中使用了 3 中排序函数学习算法：基于 BP 神经网络的回归，以及基于样本对的 RankerNet[13]和 RankBoost 算法[12]。

在训练的阶段，我们随机选取了 20 个未标注样本参加相关度传递。这些样本通过相关度传递得到的排序值和原始标注样本的标注值被用直接用于BP网络回归算法。对于 RankerNet 和 RankBoost 这两种基于点对的算法而言，我们引入了如下的参数——“成对域值”（pair constructing threshold）。训练的样本对根据下列规则进行创建。

$$\text{If } (|y_i - y_j| > pair\_construct\_th),$$
$$\text{Then insert } (x_i, y_i; x_j, y_j) \text{ into } training\_pair\_stack$$

其中 $x_i, x_j; y_i, y_j$ 分别为文档 *i* 和 *j* 的特征值向量和排序值。

对 BP 回归和 RankerNet 算法，我们令训练的过程循环 500 个周期（或者是循环尽量多的周期以保证训练误差在 10 个周期内没有变化）。每个周期的结果模型（神经网络）都被保存下来，以供校验和测试。在校验的过程中，前面提到的所有模型都在校验集上进行评估。我们选取具有最高 NDCG@10 值的模型作为最终的模型，将其在测试集上进行测试。我们保留测试集上的 NDCG@3 和 NDCG@10 作为最终的评价结果。

对于 RankBoost 而言，学习的过程同样循环 500 个周期，最终的学习结果是对弱分类器的一个线性组合（只有一个模型）。我们将训练集和校验集合并在一起，然后均分成 5 等份，在其上进行了一次 5 重交叉验证（5-fold cross validation）以选择最优模型。

为了验证我们的学习框架的有效性。我们也在同样的数据集合上对没采用相关度传递的模型进行了实验。在这些实验的训练过程中，我们将 20 个未标注

的文档视作负例加入到训练集中。

在上述所有实验之前，我们通过一个线性分类器对所有的特征进行了特征选取（feature selection），最终实验中保留的特征维数是 300 维。

### 2.4.4　实验结果

此处我们称由 BP 神经网络回归得到的结果为 *BP Net*，采用了我们的学习框架的结果为 *MRBRP + BP Net*，同样，采用 RankerNet 的结果是 *RankeNet* 和 *MRBRP + RankerNet*，采用了 *RankBoost* 的结果是 *RankBoost* 和 *MRBRP + RankBoost*。

#### 2.4.4.1　有相关度传递与无相关度传递的对比

首先，我们在 2.4.1 所述的完整的数据集上，按照 2.4.3 中的方案进行了实验。相关度传递的方式为采用了 k 近邻的查询内传递。k 近邻参数 $k$ 的取值为 10，相关度传递的强度参数 $\alpha$ 取值为 0.6。相关度传递的结果被归一化到[0,1]范围内，成对域值取值为 0.1。这也是下面所有实验的默认参数设置。实验的结果如下表。

表 2.2 有无相关度模型结果对比

| | **NDCG@3** | **NDCG@10** |
|---|---|---|
| **BP Net** | 0.437 | 0.452 |
| **MRBRP + BP Net** | 0.436 | 0.453 |
| **RankerNet** | 0.442 | 0.461 |
| **MRBRP + RankerNet** | 0.453 | 0.478 |
| **RankBoost** | 0.439 | 0.457 |
| **MRBRP + RankBoost** | 0.446 | 0.465 |

从表 2.2 中，我们可以看出通过相关度传递的帮助，RankerNet 和 RankBoost 的性能有了明显的提高。使用了相关度传递的 RankerNet 和未始用相关度传递的 RankerNet 相比，其 NDCG@3 提高了 2.5%，NDCG@10 提高了 2.2%。对 RankBoost 而言，这一提高分别是 1.6%和 1.8%。然而，基于 BP 网络的回归方法没有显著提高。这意味着查询内传递的结果对 BP 回归算法而言过于复杂。

#### 2.4.4.2　训练集规模敏感度

我们也测试了这些算法在不同规模训练集上的性能。改变训练集规模的第一种方式是固定每个查询中文档的数目而改变训练集中查询的数目。我们从初始训练集的 8000 个查询中选则了 1000，2000 和 5000 个查询作为新的训练集，同时我们使用了和前述实验同样的校验集和测试集。实验结果如下表，我们用 NDCG@10 来描述最终的结果。

表 2.3 算法对查询数目的敏感度（NDCG@10）

| **Query Number for Training** | **1000** | **2000** | **5000** | **8000** |
|---|---|---|---|---|
| **BP Net** | 0.433 | 0.444 | 0.453 | 0.452 |
| **MRBRP + BP Net** | 0.441 | 0.446 | 0.450 | 0.453 |
| **RankerNet** | 0.443 | 0.452 | 0.458 | 0.461 |
| **MRBRP + RankerNet** | 0.453 | 0.460 | 0.471 | 0.478 |
| **RankBoost** | 0.447 | 0.455 | 0.459 | 0.457 |
| **MRBRP + RankBoost** | 0.450 | 0.459 | 0.461 | 0.465 |

从上表可以看出，超过 5000 个查询的训练集对基于 BP 网络的回归没有改进。然而，当进行了相关度传递之后，更多的训练样本对最终的结果还是有帮助的（表中第二行）。当使用较少的查询作训练集时，相关度传递对最终的结果帮助更大。RankerNet 算法可以利用更多的查询作为训练样本。当加入了相关度传递后，RankerNet 的性能有了显著的提升。对于 RankBoost 而言，相关度查询的加入使得排序性能随着训练集的增加而提高，而这一特性在没采用相关度查询时是 RankBoost 所没有的。表 2.3 中，性能最优的模型是 *MRBRP + RankerNet*。

第二种变换训练集规模的方法是固定查询的数目，改变每个查询采用的未标定文档数。实验结果如下表。

表 2.4 算法对未标定文档数的敏感度（NDCG@10）

| **Unlabeled Doc Num/Query** | **10** | **20** | **50** | **100** |
|---|---|---|---|---|
| **BP Net** | 0.441 | 0.452 | 0.455 | 0.457 |
| **MRBRP + BP Net** | 0.446 | 0.453 | 0.456 | 0.460 |

| | | | | |
|---|---|---|---|---|
| **RankerNet** | 0.445 | 0.461 | 0.467 | 0.468 |
| **MRBRP + RankerNet** | 0.452 | 0.478 | 0.481 | 0.483 |
| **RankBoost** | 0.449 | 0.457 | 0.459 | 0.455 |
| **MRBRP + RankBoost** | 0.451 | 0.465 | 0.464 | 0.465 |

对这六种方法而言，采用了 100 个未标定文档得到的结果相对使用了 10 个未标定文档的改进分别是：3.6%, 3.2%, 5.2%, 6.9%, 1.2%, 3.1%。显然，引入 MRBRP 之后，RankerNet 和 RankBoost 的性能改进更大。表 2.3 和 2.4 的结果还说明了增加每个查询中文档的数目对最终的结果的贡献比单纯增加查询数目要来得大。

#### 2.4.4.3　变换成对域值的影响

成对域值的不同会导致产生不同数目的样本对。我们选择了 RankerNet 来验证不同成对域值的效果，结果如下表。

表 2.5 不同成对域值的实验结果

| Pair-Constr Th | NDCG@3 | NDCG@10 | Pair Number |
|---|---|---|---|
| **0.05** | 0.439 | 0.465 | 6235042 |
| **0.10** | 0.453 | 0.478 | 4733636 |
| **0.15** | 0.458 | 0.481 | 4362128 |
| **0.30** | 0.437 | 0.454 | 2109814 |
| **N/A** | 0.442 | 0.461 | 3369882 |

从上表可以看出，当成对域值的取值过大（如 0.3）时，产生的样本对数量比不采用相关度传递时（最后一行）还要少很多。此时训练样本的减少必然导致信息量的减少，所以最终的性能下降是合理的。当成对域值的取值过小（如 0.05）时，样本对的数量剧增。然而此时的训练样本噪声很多，而且弱样本较多，所以最终的结果比不采用相关度传递改进很小。在这个实验中，成对域值取值为 0.15 可以得到最好的结果，此时相关度传递为学习的过程提供了足够多的优良训练样本。

#### 2.4.4.4　查询内传递和查询间传递的对比

上述所有实验都是采用了查询内传递的方式。为了对比两种传递方式的效果，我们在完整的训练集上采用查询间传递进行了实验。两种传递方式的实验结果对比如下表（我们采用了 NDCG@10 作为评价标准）。

表 2.6 查询间传递 v.s. 查询内传递

| | **Inter-query** | **Intra-query** |
|---|---|---|
| **MRBRP + BP Net** | 0.476 | 0.453 |
| **MRBRP + RankerNet** | 0.475 | 0.478 |
| **MRBRP + RankBoost** | 0.459 | 0.465 |

从上表可以看到，采用了查询间传递时，*MRBRP + BPNet* 回归算法的结果有了明显的改进，然而另外两种基于点对的方法效果确稍微有些下降。这是因为相关度传递的过程起到了函数平滑的效果[22]。通过查询间传递得到的效果要比查询内传递的效果更加平滑，因为前者的平滑过程是在一个更大无向图上进行的。这一更佳的平滑性使得回归问题更加简单，同时保留了主要的排序规律。然而，过度的平滑会导致丢失一些点对间的排序关系，所以对 RankerNet 和 RankBoost 会有负面的影响。解决这一问题的一个 tradeoff 是选择更少的查询进行查询间传递。然而最优的查询数不易确定。

这两种传递方式需要的运算时间如下表所示，此处的算法和程序已经按照 2.3 中的内容进行了优化。

表 2.7 两种相关度传递算法所需时间

| | **Time Cost** |
|---|---|
| **Intra-query** | 0 hr 12 min |
| **Inter-query** | 14hr 50 min |

我们可以看出查询内传递的速度要比查询间传递快的多，所以更适用于对排序函数训练时间要求比较苛刻的场合。

#### 2.4.4.5　相关度传递直观结果举例

为了能让读者对相关度传递的效果有一个直观的印象，我们在下表中给出

了某查询的 20 篇未标注文档经相关度传递之后得到的排序值结果。本查询为“chef schools”，含义为厨师学校。此处用户的目的是在网上查找一些关于厨师学校的信息或厨师学校的网站相关度传递的结果排序值被归一化到了[0,1]。排名最高的三篇文档以斜体字标出。其中，得到了“1”的排序值（意味着最好的匹配）的文档（网页）是一个厨师学校的网站列表，非常符合用户需求。排名第二和第三的文档也是厨师学校的网站或介绍厨师学校的网页。这个例子证明了我们的方法的假设基础：具有较高排序值的未标注文档有更大的可能性是相关文档。

表 2.8 未标注文档的 URL 和由相关度传递得到的排序值

| URL | Propagated Score |
|---|---|
| http://www.theshillongtimes.com/A-19-july.html | 0.0387 |
| http://www.osca.ca/TAP/GR12TA-C.doc | 0.0698 |
| http://www.mergedigital.com/technology/games/chi-madiganrnc,0,6595206.story?coll=me-gameshead-hed | 0.0262 |
| http://www.booksm.com/phil.htm | 0.0162 |
| http://www.vigile.net/dossier-101/2.html | 0.1995 |
| http://freerepublic.com/focus/keyword?k=oprah | 0.0486 |
| http://english.hk.yahoo.com/Society_and_Culture/Food_and_Drink/Cooking/Culinary_Education | 0.1733 |
| http://www.gokis.net/self-service/archives/cat_retail_and_retail_stores.html | 0.0087 |
| http://radio-flyer.my-szukamy.com/ | 0.0162 |
| ***http://www.offshoreadventures.tv/OAS3_E2.html*** | ***0.5125*** |
| http://www.auctiongoat.com/sell/index.cfm?cat=790403775 | 0.0761 |
| http://www2.clearclick.net/directory/Head-Lamp.html | 0.1658 |
| ***http://www.dementedcards.com/cooking-classes.htm*** | ***0.4564*** |
| http://www.intlconnections.com/cookingarchive.html | 0.0212 |
| http://foodservice.chef2chef.net/directory/Business/Major_Food_Companies/W/index.html | 0.2406 |
| http://www.eurosurveillance.org/em/v03n03/v03n03.pdf | 0.0324 |
| http://www.jccc.net/home/depts/003100/site/tocaboutjccc/history | 0.2756 |
| http://www.thisissouthend.co.uk/essex/southend/news/NEWS168.html | 0.1347 |
| ***http://www.thepamperedchef.com/cooking-class.htm*** | ***1.0000*** |
| http://lk.tiptopjob.com/tiptop/links/JB/management_executive_jobs.htm | 0.0125 |

## 2.5　本章小结

在本章中，我们提出了一个全新的排序函数学习框架。其中，基于流形排序的相关度传递算法可以有效的将未标注样本转化为可用的训练样本，这为后续的排序函数学习提供了增广的样本集。由于本框架具有显著的灵活性，任何有效地半监督学习机制以及任何先进的学习算法都可以在本框架下配合使用。对其中任一部分的改进都可以改进最终的排序效果。我们尝试了将 MRBRP 和 BP 神经网络的回归学习，RankerNet 以及 RankBoost 组合。最终得到的排序函数在实际的搜索引擎数据集上有着非常好的排序效果和改进。

我们下一步的工作是要在本框架下尝试更多的机器学习算法，同时将此学习框架应用到更广泛的排序问题中去。另外，在训练的过程中，我们也会尝试将用户的原始标注和相关度传递的结果区别对待，因为其可信度有所区别。同时，我们也将在本框架的相关度传递之前对查询进行分类的，以期对每类查询得到最优的参数配置。

# 第3章 基于新闻首页视觉分布和多站点投票的新闻流行度排序

## 3.1 本章引论

如同在第一章开始所述，Cerberian 在 2004 年的一次调查结果显示，阅读新闻被 56%的网络用户评为最受欢迎的 5 种网络行为之一。网络上独立新闻源的迅速增长为我们提供了越来越多的信息通道，这是一大进步。然而，当今网络用户要面对的是从大量站点发布的远远超出用户接受能力的新闻。阅读网络上最流行的新闻，是绝大多数用户的需求。所以，如何从网络上如此大量的新闻中确定最流行的一些新闻，有着非常重要的意义。关于新闻流行度的定义请参见 1.3.2。同时，由于不同新闻的流行度以及不同的新闻事件的流行度都不相同，如何根据其流行度进行排序也是信息检索领域中需要解决的重要问题之一。在本章，我们将讨论如何从多个独立的新闻源中确定新闻，新闻事件以及新闻站点的流行度，及其排序算法。

如 1.3.2.1 所述，新闻文的流行度可以反映在两方面：

1) 新闻流行度越高，越多的站点会同时报导该新闻，反之亦然。
2) 如果新闻在新闻首页上占据的视觉区域越重要，则该站点对该新闻的推介程度越高，反之亦然。

在本章中，我们将介绍检测具有此两点属性的新闻。新闻在首页上占据的视觉重要程度可以视为新闻首页对新闻的推介强度。我们发现有些新闻站点会把具有最高流行度的新闻放在最显著的位置，而有些站点则具有显著的地域性偏好。我们定义了“可信度（credibility）”的概念用来描述我们对新闻首页推介程度的信任程度。可信度实际上反映了新闻站点的流行度，和新闻的流行度呈相关增强的关系。这类似于 J. Kleinberg 提出的 Hub 页面和 Authoritive 页面的关系[10]。同样，新闻流行度和新闻事件流行度也呈现相关增强的关系。我们采用了一个三分图模型对新闻首页，新闻，和新闻事件进行建模。同时根据这两种相关增强关系在此图上的平衡解，得出最终的排序结果。

此方面的相关工作主要分为两方面。第一类是重要事件检测。这方面的研

究主要是由美国的 TDT 组织和马萨诸塞州立大学（Umass, Amerst）的 CIIR 实验室进行的[29][30][31]。此类研究主要是从广播新闻流中检测重要事件。我们的工作和他们不同的是我们的应用范围是网络环境，可以利用更多的信息。另一类相关工作是通用网页检索中基于图的链接分析[32][33][34][35]。我们对新闻，新闻首页，和新闻事件的建模也是受了这方面的启发。

本章的剩余内容将如下组织：3.2 节给出我们的模型；3.3 节讨论流行度传递模型，包括首页投票模型，多站点投票模型和混合模型；3.4 节介绍我们实现的新闻排序系统 TopStory 的技术细节以及由首页视觉决定推介强度的方法；3.5 节给出实验数据和结果；最后是本章小结。

## 3.2 网络新闻模型

为了描述符合前述两个特点的重要新闻，我们将从新闻首页和新闻页面挖掘出两类必要的信息。

新闻首页提供的不仅仅是链向新闻页面的链接。新闻首页实际上是读者阅读新闻的入口，往往是经过精心设计以帮助用户最快速的浏览信息，例如头版头条新闻，热门新闻推荐等。这种种特点可以归纳为新闻首页上的新闻分别占据了不同的视觉重要性。最重要的新闻往往被放在页面的最上端，辅以新闻图片和说明性文字。不怎么重要的新闻往往就是通过一个配有链接的标题来表示。而此类信息反映了新闻首页（实际上是新闻站点的编辑）对新闻的推介程度，这些信息对于我们确定新闻的流行度是非常有帮助的。

新闻页面通常包括标题，摘要和内容。我们可以根据这些信息来比较两篇新闻是否报道同一新闻事件。另外，从多个新闻站点的报道中，我们还可以得出多少新闻是在报道同一事件。

我们对这两类信息有如下的观察。

### 3.2.1 关于网络新闻的观察

新闻首页的可信度（流行度）也不尽相同。每个新闻站点通常包含两类新闻首页：入口首页和分类首页。入口首页通常是将各类新闻中最重要的选取出来报道，而分类首页则报道属于特殊类别的所有新闻，例如，世界新闻，体育新闻，娱乐新闻等。通常而言，入口首页的流行度要比分类首页来的高。同时，

不同的新闻站点的同类首页的流行度也有所不同。新闻首页的流行度和新闻的流行度有着相关增强的关系。

**观察 1** 新闻首页和新闻页面

- 在流行度高的新闻首页上占据了重要的视觉地位的新闻的流行度高的可能性较大。
- 流行度高的新闻首页报道流行度高的新闻的可能性较大。

所有的新闻报道都是由新闻事件驱动的。新闻页面的流行度和新闻事件的流行度同样呈相关增强的关系。

**观察 2** 新闻页面和新闻事件

- 重要的新闻事件被更多新闻页面报道的可能性较大。
- 报道重要新闻事件的新闻页面其流行度也较高。

### 3.2.2 网络新闻的三分图模型

我们采用了一个三分图模型对网络新闻的前述属性关系进行建模。此处有三类对象：新闻首页，新闻页面和新闻事件。该三分图为一个五元组 $G=\{F,N,E,Q,P\}$，其中 $F=\{F_1\cdots F_m\}$，$N=\{N_1\cdots N_n\}$，$E=\{E_1\cdots E_d\}$，分别对应着新闻首页，新闻页面和新闻事件。我们定义 $Q$ 为一个 $F\times N$ 的矩阵，$Q_{ij}$ 代表着新闻首页 $F_i$ 对新闻 $N_j$ 的推荐强度。我们假定所有新闻首页的最大推荐强度是一致的。所以，$Q$ 的行要归一化以保证 $\forall i,\max_j Q_{ij}=1$。$P$ 是一个 $N\times E$ 的矩阵，$P_{ij}$ 代表新闻 $N_i$ 在报道事件 $E_j$ 的可能性。同时，我们有 $\forall i,\sum_j P_{ij}=1$ 成立。这里的 $P$ 和 $E$ 不是直接观测得到的。下图是本三分图模型的形象表达。

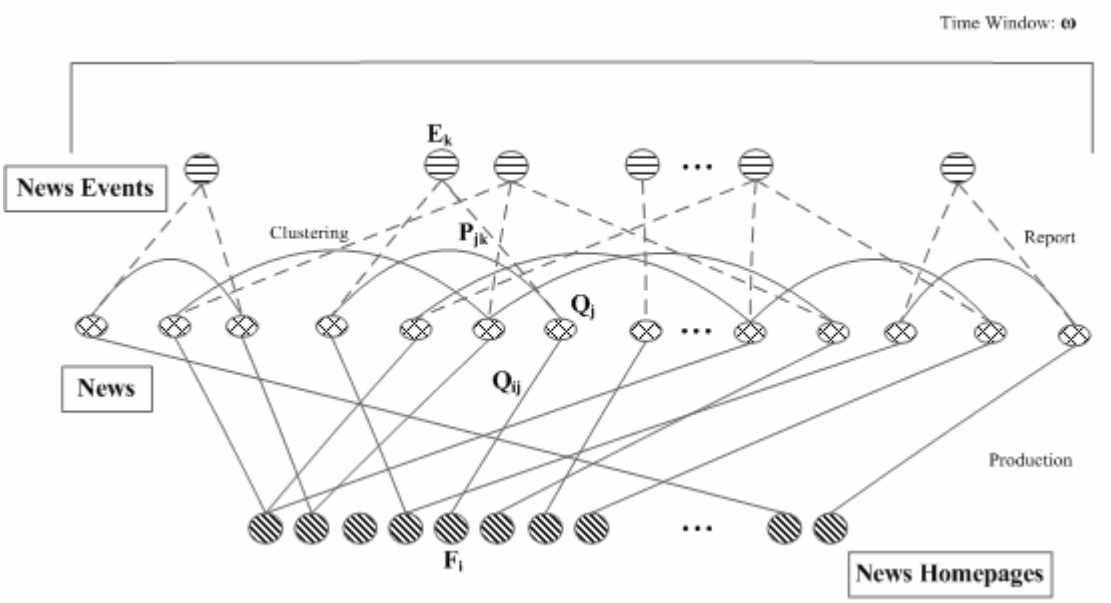


图 3.1 新闻首页，新闻页面，和新闻事件的三分图模型

我们为每个新闻首页 $F_i$ 定义了首页流行度 $w_i^f$，为每个新闻页面 $N_j$ 定义了新闻流行度 $w_j^n$，为每个新闻事件 $E_k$ 定义了事件流行度 $w_k^e$。同时，每类流行度都将归一化如下：

$$\sum_{i=1}^{F}(w_i^f)^2=1,\ \sum_{j=1}^{N}(w_j^n)^2=1,\ \sum_{k=1}^{E}(w_k^e)^2=1$$

## 3.3 流行度传递模型

基于观察 1 和观察 2，我们通过在图上求得两种相关增强关系的平衡状态，从而得出三者的流行度。平衡状态的求取可以通过类似于 HITS 算法的迭代过程实现[10]。

我们首先以独立的方式分别研究这两种相关增强的关系。通过进一步的分析，我们将发现这两种关系之间是有联系的。更好的方式是将二者综合考虑，找到一个共同的平衡解。

### 3.3.1 首页投票模型及分析

根据观察 1，我们定义如下操作（$Q^T$ 是 $Q$ 的转置）：

$$w^n \leftarrow Q^T \times w^f \tag{3-1}$$

$$w^f \leftarrow K_q \times Q \times w^n \tag{3-2}$$

这里的 $K_q$ 是一对角阵，$K_q(i,i)=1/\sum_j Q_{ij}^2$。此处引入 $K_q$ 是必要的，因为新闻首页的流行度并不取决于其上报道的新闻篇数。所以当给定 $w^n$ 时，我们可以用下式来估计 $w^f$。

$$w_i^f = \frac{\sum_j Q_{ij} w_j^n}{\sum_j Q_{ij}^2} \tag{3-3}$$

3-1 和 3-2 是 $w^f$ 和 $w^n$ 之间相关增强关系的体现。通过对 3-1 和 3-2 的不停交替迭代，我们可以得到 $w^f$ 和 $w^n$ 稳态平衡解。此时，$w^f$ 收敛到 $w^{f^*}$，$w^{f^*}$ 是 $K_q \times Q \times Q^T$ 的主特征根。$w^n$ 收敛到 $w^{n^*}$，$w^{n^*}$ 是 $Q^T \times K_q \times Q$ 的主特征根，同时 $w^{n^*}$ 正比于 $Q^T \times w^{f*}$

基于线性代数理论，我们有如下结论：如果 $M$ 是一 $n \times n$ 的矩阵，$v$ 是任一不与 $M$ 的主特征向量正交的向量，则随着 $k$ 的增加，$M^k v$ 会收敛到 $M$ 的主特征向量。这一结论成立的一个必要前提是 $M$ 的主特征根大于 $M$ 其它的所有特征根。

我们定义矩阵 $B = Q^T \times K_q \times Q$，有 $B_{ij} = \sum_{l=1}^{m} Q_{li} \times Q_{lj} \times K_l$。所以，如果两篇新闻出现在同一个新闻首页，他们的流行度是相关增强的，相关系数正比于该首页对这两篇新闻的推介程度。

此模型的主要缺点是无法提供不同新闻首页的流行度排序。这是因为，每个新闻首页（新闻站点的入口首页除外）提供独立的新闻报导，本模型无法对来自不同首页的不同新闻之间的联系进行建模，只有同时出现在同一新闻首页的新闻之间才能相互影响。这意味着 $B$ 将成为一个块对角阵，每一块对应着一个新闻首页。所以，在其主特征向量 $w^{n*}$ 中，只有来自某个特定首页的新闻会对应着非 0 值。这一首页对应着 $Q$ 中最密集的字图。由于对 $w^{f*}$ 与 $w^{n*}$ 正比，所以 $w^{f*}$ 也是同样的情况。

这一缺点会导致另一个问题。每个新闻首页都有其特定的推介偏好。所以，$w^{n*}$ 得结果会受 $w^{f*}$ 中具有最大值的新闻首页偏好的影响。当然如果我们能够合理的计算新闻首页流行度的相对值，将会很好的平滑由单个首页的新闻偏好带

来的影响。

实际上，来自于不同新闻站点的新闻首页的流行度相互之间并不是完全独立的。由于多个站点之间报道的新闻会有相当的重复，所以我们可以利用这一信息来更好的计算新闻首业之间的相对流行度。下面的模型将对这一问题进行讨论。

### 3.3.2 多站点投票模型

基于观察 2，我们定义如下操作：

$$w^n \leftarrow P \times w^e \tag{3-4}$$

$$w^e \leftarrow P^T \times w^n \tag{3-5}$$

显然，这一过程中的数学规律和首页投票模型完全一样。所以此处的 $w^{n*}$ 是 $P \times P^T$ 的主特征向量。

我们定义矩阵 $A = P \times P^T$，有 $A_{ij} = \sum_{l=1}^{d} P_{il} \times P_{jl}$。所以 $A_{ij}$ 意味着新闻页面 $N_i$ 和 $N_j$ 在报道同一新闻事件的概率。我们可以通过计算两片文档内容上的相似度来估计 $A_{ij}$。文档相似度的计算方法的细节将在 3.4.2 中讨论。此处，我们假定我们可以不需要 $P$ 而得到这一相似度（实际情况也是如此）。这样，我们就可以推导出 $w^{n*}$ 的解答而不需要知道 $E$ 和 $P$ 的取值。

这一模型的性能非常依赖于相似度矩阵 $A$ 的近似计算。为了说明这一点，我们考虑如下极端情况。假定有一些话题（topic），每个话题包含一些事件，此处的话题是比事件更为广阔的一个概念。自于两个不同话题的新闻页面应该是完全无关，它们在 $A$ 中的对应取值应该为 0。然而，由于文档相似度的计算本身是一不精确的过程，无法保证两片新闻页面的相似度一定为 0。同样，处于同一话题的新闻往往在内容上有着相当大的相似度，然而，很有可能不是报道同一时间。这是当前事件检测技术的极不完善所导致的。

假定我们在理想的情况下得到了 $A$，此时 $A$ 应该为一个块对角阵，每一块对应着一个话题。在其主特征向量中，只有某一个话题的新闻会获得非 0 值。这样意味着这一话题中的任一新闻要比其他话题中的所有新闻都要来得重要，这显然是不合理的。因为每个话题都有重要新闻和不太重要的新闻。

在 HITS 算法中[10]，J.Kleinberg 也讨论了对具有多个 Hub 和 Authority 顶点的图的同样问题。在 HITS 算法中，一个查询字符串对应着此处的一个话题。但

是 J.Kleinberg 也没有给出好的解决方法。

本模型的另一个问题就是，当某条重要新闻首次出现而没有太多首页同时报道时，本模型永远无法将该重要新闻检测出来。由于不同的新闻站点在报道同一新闻事件必然会有先后性，所以这一问题非常致命。只有当越来越多的新闻站点开始报道这一新闻时，本模型才能发挥作用。然而首页投票模型就没有这个问题。所以，我们可以考虑将这两个模型搭配使用以互补不足，这就是我们下面要讨论的混合模型。

### 3.3.3 混合模型

根据前文所述，首页推介模型和多站点投票模型都有其优势和不足，而且这两个模型正好是互补的关系。所以我们将它们混合起来考虑，混合的规则如下：

$$w^n \leftarrow A \times w^n \leftarrow A \times Q^T \times w^f \tag{3-6}$$

$$w^f \leftarrow K_q \times Q \times w^n \tag{3-7}$$

同样，我们可以通过对上述两步的迭代而得到稳定解。此处的 $w^{f*}$ 是 $K_q \times Q \times A \times Q^T$ 的主特征向量，$w^{n*}$ 是 $A \times Q^T \times K_q \times Q$ 的主特征向量。用来计算稳态解的算法流程如图 3.2。这一算法收敛很快，通常迭代周期 $k = 20$ 就可以保证收敛。

**Hotness and Credibility Iteration Procedure:**

Iterate $(Q, A, k)$

$Q$: a front-page to news matrix
$A$: a news to news matrix
$k$: a natural number

Let $z$ denote the vector $(1,1,\cdots,1) \in R^m$.

Set $w_0^n = z$.

For $i = 1,2,\cdots,k$

$w_i^n = A \times Q^T \times K_q \times Q \times w_{i-1}^n$.

Normalize $w_i^n$

End

Set $w_k^f = K_q \times Q \times w_n^f$.

Normalize $w_k^f$

Normalize $w_k^n$

Return $(w_k^n, w_k^f)$

图 3.2 流行度迭代算法

我们的三分图模型是作为连接的整体而存在的。直观的说来，来自于不同新闻站点是通过其报道的相同事件产生联系的；报道不同事件的新闻也会通过处在同一新闻首页而发生联系。唯一例外的情况是关于某一新闻事件的新闻仅仅由唯一的一个新闻首页报道，但是这种情况在实际中不可能发生，所以我们在此忽略了此类情况。

考虑到三分图上每一条边都被分配了正的权值，所以经过迭代 $w^{f*}$ 和 $w^{n*}$ 中每一个元素的取值都是正值。这意味着每一条新闻和新闻首页都有其流行度取值，我们可以比较任意两条新闻或新闻首页的流行度，这一特性是前两种模型所不具备的。

## 3.4 TopStory 系统

基于上述算法，我们设计实现了一个实际多站点新闻流行度排序系统：TOPSTORY（见图 3.3）。本系统检控一系列的新闻站点，并周期性的下载其上的新闻首业。由程序解析出所有由首页指向的新闻，并将最新出现的新闻下载下来。经过分析程序，新闻页面和新闻首页中的内容都将被保存至数据库。

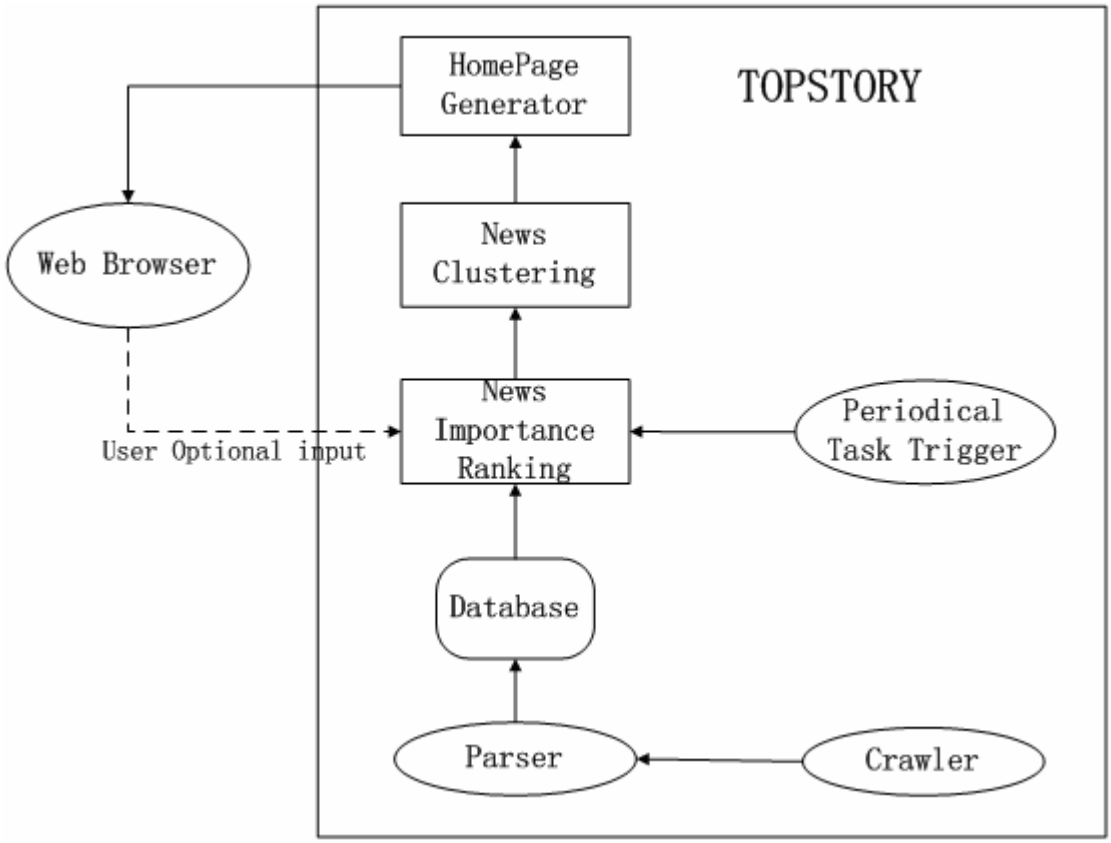


图 3.3 TOPSTORY 系统结构图

本系统有两种用户交互方式。第一种是系统周期性的计算出当前流行度最高的新闻，将结果以新闻事件的方式组织返回给用户观看，类似于 Google News 的效果。第二种方式是由用户查询驱动。这种方式下系统可以动态检测出任意给定时间段内的重要新闻，同时返回用户指定的数目的新闻事件。我们实现了一个简单的新闻聚类算法以将新闻页面聚类成新闻事件。最终生成新闻首页中的新闻是按照事件流行度进行排序的，非常易于阅读。

在下面的章节中，我们将介绍从新闻首页和新闻页面中抽取信息来计算推

介强度矩阵 $Q$ 和相似度矩阵 $A$ 的方法。

### 3.4.1 由首页视觉决定的推介强度

在 TOPSTORY 系统中，每个新闻首页都以一系列的快照来表示 $\{S_{t1},S_{t2},\cdots\}$。此处快照 $S_t$ 代表处在时刻 $t$ 的新闻首页。每个 snapshot 都表示了一些具有不同时决强度的新闻。我们可以将某一新闻首页在某一时间段内的所有快照合并起来，就可以得到该首页对每条新闻的推介强度。

我们采用了基于视觉的页面分割算法[36]（VIsion-based Page Segmentation algorithm）来分析首页快照的视觉结构。图 3.4 是一个例子。


图 3.4 某新闻首页的快照

由上图可以看出，每一篇快照都被分为了一些矩形的块，每个块由一个新闻页面的链接支配。我们采用了针对新闻页面链接训练的分类器将那些没指向新闻页面的链接剔除了出去。

某块的视觉强度是由其大小，位置和是否包含图片决定的。我们采用了下

面的规则来计算：

$$q(S,N)=BlockSize/MaxBlockSize+(1-top/PageHeight)+0.5*(ContainImage?1:0) \quad (3\text{-}8)$$

这里的 $q(S,N)$ 代表新闻 $N$ 所对应的块在 $S$ 中占据的视觉强度。*BlockSize* 是快的面积。*top* 是该块最高点的位置。*ContainImage* 是一个布尔型变量用来判断块是否包含图片。*MaxBlockSize* 是 $S$ 中最大块的面积。*PageHeight* 是快照页面的高度。

一条新闻在首页上的视觉强度可能会随着时间而改变。我们将对一个时间段内的所有快照作一合并，得出该首页某一新闻的推介强度。合并的规则视用户的需求而定。举例来说，如果用户是想阅读一周内的所有重要新闻，那么本周内的所有快照具有同样的重要程度。假如用户想要阅读当前的重要新闻，此时最近的快照要比早些时间的快照更加重要，这是新闻时效性的体现。我们通过对快照加权的方式进行调整。在时效性模型下，我们采用了 sigmoid 函数来模拟新闻的流行度随着时间衰弱的情况。这里的 $a$ 和 $t_0$ 是用来调整时间的参数。

$$w(S_t)=\begin{cases}1 & \text{, for the first case}\\ \dfrac{1}{1+e^{a(t-t_0)}} & \text{, for the second case}\end{cases} \quad (3\text{-}9)$$

对于新闻 $N$ 来说，其最终得到的推介强度是它在每个快照上所占据视觉强度的加权平均。

$$q(F,N)=\frac{\sum_{S_t} w(S_t)*q(S_t,N)}{\sum_{S_t} w(S_t)} \quad (3\text{-}10)$$

在式 3-10 中，只有包含新闻 $N$ 的快照才参加了最后的计算。同时我们对 $q(F,N)$ 进行了归一化，以保证每个新闻首页的最大推介强度为 1。

### 3.4.2 新闻文档相似度算法

为了计算两篇新闻报道同一事件的可能性，我们采用了常用的向量空间模型（VSM）来对新闻文档内容进行建模，进而计算文档的相似度。对于每篇新闻而言，我们定义了两种关键词：命名实体（Named Entity）和普通关键词（general keyword）。我们通过自然语言处理工具从每篇文档中抽取出如人名，地名，组

织名称，时间等等具有描述事件信息的命名实体关键词。另外我们对文档内容中剩余的关键词采用了过滤词过滤（stop words）和词干化（stemming）技术进行处理，以得到普通关键词。

对于每个关键词，我们从如下三个方面考虑其权重。

1．经典的 TF-IDF 算法[11]。

2．新闻页面中的字体信息。例如，以黑体字，粗体字或比较大的字体出现的关键词将获得更大的权重。

3．命名实体将比普通关键词更加重要。

我们用于计算新闻页面 $N$ 中关键词 $e$ 的规则如下：

$$w(N,e)=\begin{cases} idf_e * \sum_{\omega\in N \text{ and } \omega=e} f_N(\omega) & \text{, if e is a feature word} \\ idf_e * \sum_{\omega\in N \text{ and } \omega=e} f_N(\omega)*5 & \text{, if e is a name entity} \end{cases} \quad (3\text{-}11)$$

这里的 $f_N(\omega)$ 表示词 $\omega$ 在 $N$ 中的出现处的字体大小。

接下来我们采用了经典的余弦相似度算法来计算两篇新闻文档的相似度。

$$S(N_1,N_2)=\cos(V_1,V_2)=\frac{V_1\times V_2}{\|V_1\|\bullet\|V_2\|}=\frac{\sum V_{1i}V_{2i}}{\sqrt{\sum V_{1i}^2}\bullet\sqrt{\sum V_{2i}^2}} \quad (3\text{-}12)$$

其中 $V_1$，$V_2$ 分别代表新闻 $N_1$，$N_2$ 的 VSM 向量。$V_{1i}$ 表示向量中的第 $i$ 个关键词。

## 3.5 实验

本节中，我们首先介绍实验数据。然后我们将在 TOPSTORY 系统上进行一系列的实验来验证我们的算法。

### 3.5.1 实验数据

我们监测了 8 个著名的新闻站点。监测过程持续了一周时间，所有的新闻首页快照以 10 分钟为周期被保存了下来，所有的新闻页面也被保存下来。。我们对“世界新闻”这一栏目尤其感兴趣因为所有的新闻站点都有此栏目，可比性较好。表 3.1 是对实验数据的一些统计。

表 3.1 实验数据统计

| News Sites | Homepage Num | News Page Num | World News Num |
|---|---|---|---|
| BBC | 7 | 1262 | 348 |
| CNN | 8 | 331 | 133 |
| CBC | 5 | 498 | 258 |
| NEWSDAY | 6 | 922 | 336 |
| CBSNEWS | 5 | 197 | 50 |
| YAHOO | 8 | 1219 | 365 |
| ABCNEWS | 7 | 2601 | 463 |
| REUTERS | 6 | 624 | 289 |

下面我们将解释我们如何设计用户标定数据的。实际上，主观上很难确定每篇新闻的流行度。这里存在两个问题。第一个问题是，用户能够确切进行标定的是新闻事件而不是每篇新闻报道本身。所以，我们先要将新闻页面划归到一定的新闻事件中去。我们采用了聚类算法来自动实现这一过程。虽然这一聚类算法未必非常精确，但是它为不同的三种模型提供了同样的标定数据。另一个问题是用户标定是用户的主观反映。

我们请来 5 个用户对新闻事件的流行度按照下表所示的取值进行了标定。

表 3.2 流行度标定分级

| Importance Level | Weight |
|---|---|
| Very important | 10 |
| Important | 5 |
| Normal | 0 |

我们用 5 个用户标定的平均值作为每一事件的流行度。需要说明的是，这 5 个标定者都来自于我们的研究组，所以标定结果可能存在者一定的用户群体偏好。然而，对于前述 3 个模型来说，同样的用户标定是公平的。所以，在这一标定上比较 3 个模型的实验结果是有意义的。

### 3.5.2 实验结果

#### 3.5.2.1 新闻首页流行度

首先，我们比较新闻首页的流行度。我们采用了世界新闻栏目，主要考虑到该栏目的新闻具有前述的优点，而其他栏目的新闻在数量上随着站点的不同区别过大，很难进行有意义的对比。此处为了表述得方便，我们用新闻站点的名称来指代各自世界新闻首页。而且，这样的比较也能更加直观的反应站点之间的区别。

为了在 3 个模型中保持一致，该流行度通过 $K_q \times Q \times w^{n^*}$ 进行计算（参见方程 3.2）。表 3.3 是最终的结果。

表 3.3 关于世界新闻的首页流行度

| News Sites | Hybrid Model | Homepage Model | Similarity Model |
|---|---|---|---|
| BBC | 0.3914 | 0.0000 | 0.3066 |
| CNN | 0.3405 | 0.0000 | 0.2586 |
| CBC | 0.1356 | 0.0000 | 0.1451 |
| NEWSDAY | 0.1329 | 0.0000 | 0.1182 |
| CBSNEWS | 0.1252 | 0.5290 | 0.1373 |
| YAHOO | 0.0750 | 0.0000 | 0.0917 |
| ABCNEWS | 0.2110 | 0.0000 | 0.3777 |
| REUTERS | 0.4541 | 0.0000 | 0.6276 |

有趣的是混合模型得道的结果比多站点投票模型（Similarity model）更加平衡。二者的方差分别是 0.0203 和 0.0325。从结果看来，多站点投票模型对新闻首页的偏好程度更加严重。而首页投票模型无法实现多站点的首页排序，再次模型中，只有来自一个站点的首页才能存活下来。但是需要指出的是，在一个站点内部，首页投票模型是可以得出不同分类首页之间的流行度排序的。

#### 3.5.2.2 基于 Scope-Average 的流行度

我们采用了类似于 scope-precision 的方法来对前述三个模型的性能进行比较。Scope 的含义是由算法生成的排序中事件的个数。我们将这些事件的标定流

行度取值求平均，与理想情况下的事件排序进行对比。 图 3.5 给出了 3 个模型的对比。

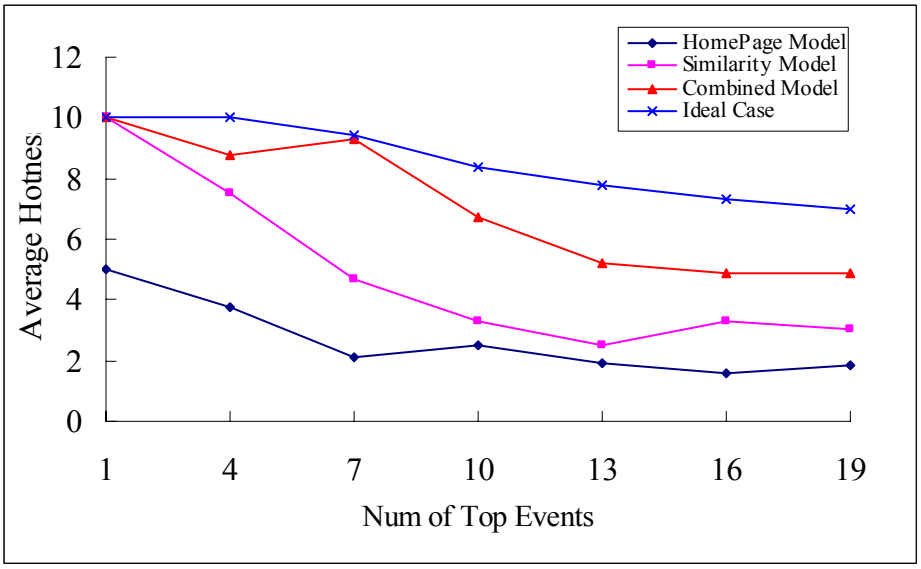


图 3.5 算法的 Scope-Average 流行度取值

从上图可以看出，混合模型的性能要明显优于多站点投票模型和首页投票模型。对多站点投票模型而言，有一些内容上和重要事件相近的普通事件被算法误认为重要事件加入排序，导致了整体 Scope-Average 分值的下降。对首页投票模型而言，只有来自一个站点的新闻事件被保留了下来（参见表 3.3）。由于单个站点新闻的偏好和不全面性，必然导致最终的效果变差。

#### 3.5.2.3 时延问题

评价这三个模型的另一个标准是对重要新闻事件检测的时延。此处时延的确切定义是从对某重要事件的第一次报道出现开始，到我们的 TOPSTORY 系统能将其作为重要新闻挑选出来的这段时间。我们希望这一时延尽量的小以便能够将重要新闻第一时间提供给用户。

我们任选了一组重要事件，在这三种模型中，这些事件都可以被检测为重要事件。这些事件的列表如下。左边一栏是这 8 个站点中最早出现关于这一事件报道的时间，右边是事件的简要描述。

| | |
|---|---|
| 2005-01-02 14:44:39 | Car bomb attack kills 19 Iraqis |
| 2005-01-03 02:45:02 | Peru rebels surrend |
| 2005-01-02 23:39:46 | Powell warns of more Iraq attacks |
| 2005-01-04 06:53:55 | Governor of Baghdad assassinated |
| 2005-01-06 11:03:08 | Nelson Mandela's eldest son dies |
| 2005-01-05 05:35:40 | Aid plea for 'tsunami generation' |
| 2005-01-06 06:36:10 | Jordan rallies support for Iraq poll |
| 2005-01-02 14:36:06 | Peter Molyneux has been made an OBE. |
| 2005-01-03 14:47:06 | Uzbeks promise smelter clean-up |

时延的结果在图 3.6 中给出。

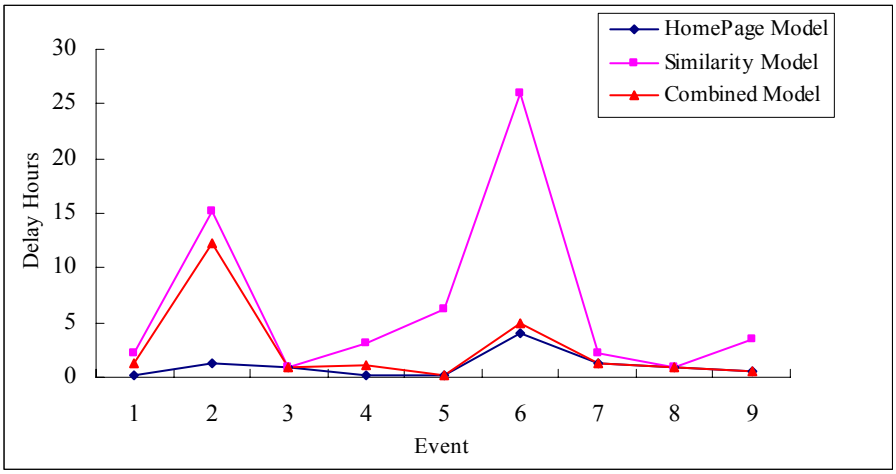


图 3.6 各种模型的时延

对首页投票模型，多站点投票模型，和混合模型而言，平均时延分别为 1.0 小时，6.7 小时，和 2.6 小时。由首页投票模型得到的时延结果要比其他两个模型小的多，这是因为当新闻站点第一次报道重要新闻时，往往就将其放在首页上的显要位置。而多站点投票模型需要大得多的时延，这是因为只有当多个站点都报道了关于此事件的新闻之后，系统才能监测到该新闻。而混合模型在大部分情况下都很接近首页投票模型的时延。

#### 3.5.2.4 多站点和单站点比较

我们认为从多站点阅读新闻比仅仅阅读来自单个站点的新闻要更有优势。由于每个新闻站点都有其自己的偏好，更倾向于报道其本地的新闻而不是最重

要的新闻。尽管一些著名的新闻站点会尽量满足全球用户的阅读需要，然而还是或多或少的存在这一问题。所以，通过综合多站点的解果，我们可以满足更多的用户。

为了验证这一想法，我们进行了如下实验。我们仅仅使用了来自单个站点的新闻数据进行了计算，将得到的新闻事件排序的 Scope-Average 值和所有新闻站点数据的结果相比较，如图 3.7 所示。

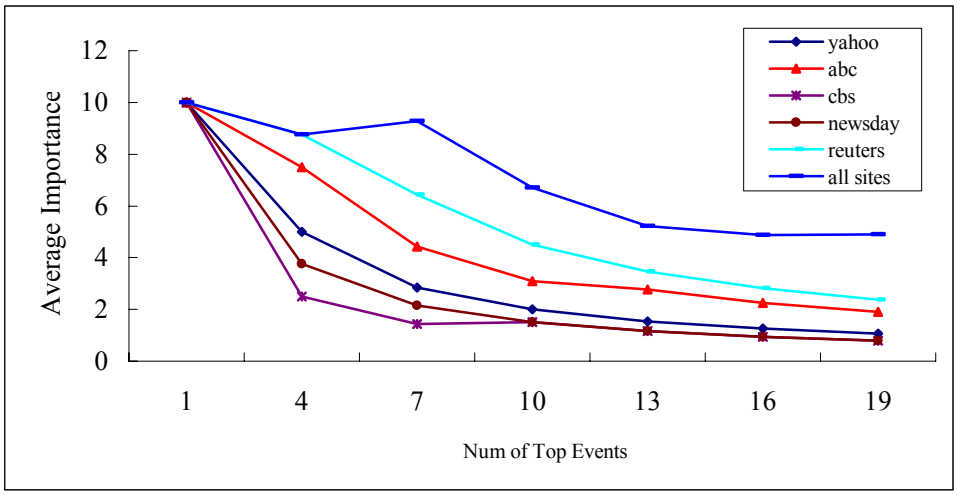


图 3.6 单站点和多站点的 Scope-Average 值

从上图可以看出，基于多个站点数据的结果远远超出任何来自单站点的结果。这也和我们的标注用户对这些新闻站点没有自己的区域偏好有关。

## 3.6 本章小结

在本章中，我们提出了在网络环境中检测重要新闻的算法，可以依据新闻，新闻首页，和新闻事件的流行度进行排序。我们探究了新闻首页对新闻的视觉推介和多站点对同样新闻事件的共同推介关系。我们采用了一个三分图模型对这三者进行了建模，同时我们提出了基于特征向量的算法以求出该图的稳态解作为流行度排序结果。我们实现了一个新闻排序系统 TOPSTORY，以帮助用户更好的阅读重要新闻。实验结果验证了我们的模型和算法的有效性。我们尝试了不同的投票模型，最终的结果证明混合模型优于单一投票模型。

这项研究的后续工作包括几个方面。首先，网络上蕴含的信息比我们现在

用到的视觉信息和文本内容信息要丰富的多，如何利用其他相关信息来确定新闻流行度是一个非常有意思的问题。第二，我们现在采取的解法是基于特征向量的方法，然而对这一方法，我们还没有很好的理论上的解释。我们可以尝试将此图当作随机图（random graph）进行建模。最后，由于新闻事件是本模型的一个重要组成部分，而且也是展现给用户的最终结果的最小单位。我们需要更好的事件检测算法和新闻聚类算法来改善这方面的性能。

# 第4章 基于自适应用户兴趣模型和改进的协同过滤算法的新闻推介排序

## 4.1 本章引论

随着互联网上新闻站点的爆炸性增长，在网络上阅读新闻越来越成为一种耗费时间的行为。用户需要花费大量的时间从海量的新闻中挑选出他/她感兴趣的少量内容。所以，按照用户的个人兴趣进行个性化新闻推介是十分必要的。很多公司和组织在这方面已经有所尝试，例如 Google News Alert 等。然而，这些方法流于简单，并没有取得好的效果。

根据以往的研究[37][38]，用户兴趣一般包括多个固定的概念类别，如世界新闻，体育新闻等。在每个概念类别中，用户兴趣又可以分为长期兴趣和短期兴趣。这两类兴趣实际上又各自包括多个层面。

个性化新闻推介的关键问题之一是如何找到一个合适的方法来对用户兴趣和兴趣变化进行建模。如 1.3.3.2 中所述，这一问题属于基于内容的过滤领域（Content Based Filtering，CBF，下同）。在 CBF 领域，被广泛证明行之有效的一类方法是采用层级结构的用户兴趣模型进行建模[37][38][39][40]。然而，这些方法都不能描述用户在事件这一层面的兴趣。

个性化新闻推介的另一大解决思路是通过相似用户群体进行推介。这属于协同过滤领域（Collaborative Filtering，CF，下同）。然而在当前的 CF 研究中，不管是传统的 Memory-based Collaborative Filtering （MBCF，下同）还是现在流行的 Model-based Collaborative Filtering 方法，都不能在计算用户相似度时将用户不同层面的兴趣考虑进去。

在本章中，我们将从如下两个方面讨论个性化新闻推介问题。首先，我们提出了一个层级结构的用户兴趣模型。在这一模型的比较高的层面上是对应着概念分类的固定节点，在比较低的层面上是对应着事件的动态节点（事件的定义请参照 1.3.2.1）。这一用户模型的节点内容和动态事件节点结构会随着用户的兴趣改变而变化。第二，利用[41]中的方法，我们使用用户兴趣模型中多个层面上的信息来计算用户相似度，这样在我们的 CBF 和 CF 方法中都考虑到了用户

兴趣的多层面性。另外，通过采用用户兴趣模型对新闻排序进行预测，我们还解决了传统 CF 算法的稀疏性（sparsity）问题和 first-rating 问题。

本章的剩余内容将如下组织：4.2 节介绍 CBF 和 CF 领域的相关工作及其局限；4.3 节讨论自适应的用户兴趣模型；4.4 节介绍基于自适应用户兴趣模型的协同过滤算法；4.5 节给出统一的个性化推介框架；4.6 节给出实验结果；最后是本章小结。

## 4.2 相关工作介绍

本节中，我们将分别介绍 CBF 和 CF 两个领域的相关工作。

### 4.2.1 用户兴趣模型建模（CBF）

这方面的研究包括在两层的框架下对用户兴趣建模[37][38]和采用了多层的层级结构建模[39][40][42]。

在[37]中，作者首先利用 k 近邻算法来判定新闻是否属于代表用户短期兴趣的新闻集合。如果不是，再将该新闻和一个用于表示用户长期兴趣的固定关键词向量对比，以判断是否属于用户长期兴趣内容。在[38]中，作者采用了两个关键词向量来描述用户短期兴趣中感兴趣的内容和不感兴趣的内容；用另外一个关键词向量来描述用户长期兴趣的感兴趣内容。然而，这两种模型对用户兴趣的简单二分类导致他们无法更加精确的对用户兴趣进行建模。

[39][42]中的多层级用户兴趣模型一定程度上体现了用户兴趣中从一般到特殊的关系。在[42]中，Pretschner 提出了一种具有概念分类节点的层级模型结构，这实际上是来自于类似 Yahoo Directory 的一种公开分类。这一模型的结构是固定的，只有概念分类节点上的内容可以随着用户兴趣的变化而改变。[39]的作者提出了一种具有类似概念分类节点的层级结构模型 PVA。这一模型的改进是可以同时改变节点的内容和结构以跟踪用户兴趣变化。然而，这一层级结构的创建和节点关系变化都要按照某预定义的全局层级结构进行。此处的全局层级结构就是类似[42]中采用的比较成熟的概念分类，如下图所示。

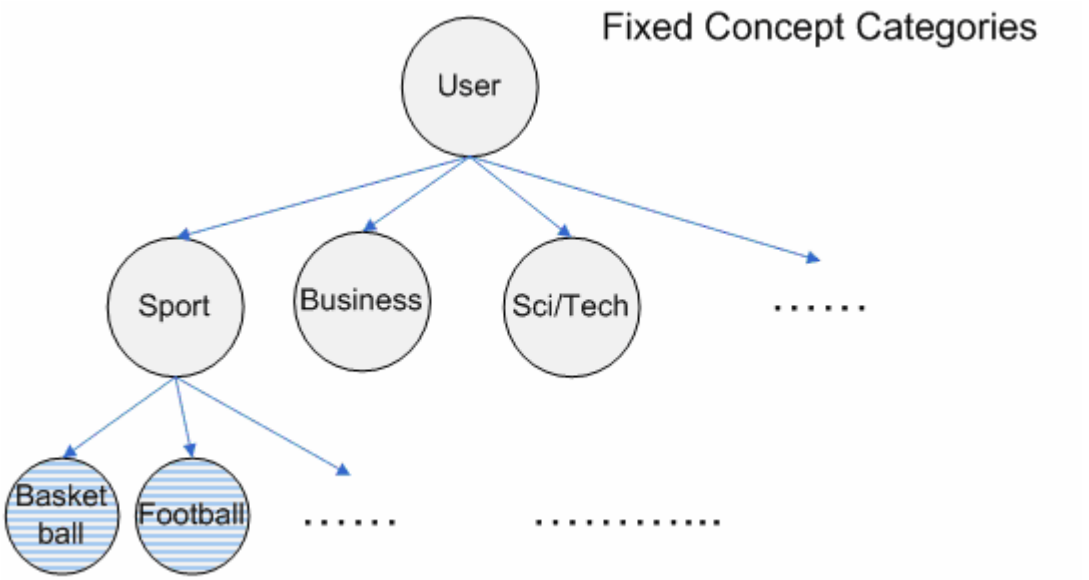


图 4.1 概念分类固定层级结构

具体说来，假设用户 A 原本对篮球很感兴趣而对足球不感兴趣，则 A 的兴趣模型上就没有足球节点。然而如果对足球新闻产生了兴趣，则 A 的模型上就会按照上图的结构加入足球节点。当 A 对足球的兴趣消失之后，模型上的足球节点就会被取消。

这类模型的最大问题是无法在比固定的概念分类更精确的层面上对用户兴趣建模，比如，这一类模型通常无法区分用户对 NBA 决赛的兴趣和用户对奥运会篮球决赛的兴趣。对新闻读者来说，新闻事件通常是其兴趣内容的最小单位。

Nanas 等人[39][40]提出了一种全新的用于构建层级式兴趣模型的方法。他们从正例文档中抽取了一系列有代表行的关键词，将其按照层级结构的形式组织起来。这一模型可以反应关键词之间的语义关系，然而，这一模型的性能一般，而且由于其无法对用户在固定概念分类和事件上的兴趣进行显式的描述，所以不太适用于新闻推介[43]。

### 4.2.2 协同过滤（Collaborative Filtering）

#### 4.2.2.1 Memory-based CF 算法

MBCF 算法[44]是 CF 领域中最经典也是应用最为广泛的一种方法。其基本的想法是以相似用户群体对某文档的评定的加权平均来代替当前用户对该文档

的评价。算法的过程如下：

1. 通过用户评价向量间的 Pearson 相关性来计算用户相似度。
2. 选择和当前用户最为相似的 $n$ 个用户。
3. 对选择的用户群体进行加权平均，得到对当前用户评价的预测。

在第 1 步中的用户评价向量间的 Pearson 相关性定义如下：

$$w_{a,b}=\frac{\sum_{j=1}^{M}(v_{a,j}-\bar{v}_a)(v_{b,j}-\bar{v}_b)}{\sqrt{\sum_{j=1}^{M}(v_{a,j}-\bar{v}_a)^2\sum_{j=1}^{M}(v_{b,j}-\bar{v}_b)^2}} \tag{4-1}$$

其中，$M$ 是所有文档的数目，$v_{a,j}$ 是用户 a 对文档 j 的评价，$\bar{v}_a$ 是用户 a 所有评价的平均值，$w_{a,b}$ 是用户 a 和 b 的相似度。

最终的评价预测为：

$$P_{a,j}=\bar{v}_a+\frac{\sum_{b=1}^{N}(v_{b,j}-\bar{v}_b)\times w_{a,b}}{\sum_{b=1}^{N}w_{a,b}} \tag{4-2}$$

其中，$N$ 是用户的数目。$P_{a,j}$ 是对当前用户 $a$ 于文档 $j$ 的评价预测。

MBCF 的一大优势是可以迅速的将用户最新的信息引入排名预测，然而它有两个致命的弱点[45]：

1. **稀疏性**（Sparsity）。简单说来就是大部分用户只会对非常少量的文档作出评价，所以用户－文档的评价矩阵十分稀疏。所以，通过 Pearson 相关系数计算出相似用户的可能非常小。这个问题在新闻推介系统这种文档－用户比很大的系统中尤其突出；或者在推介系统刚开始运行，用户评价还很少的时候也比较严重。
2. **First-rating 问题**。如果没有任何用户对某一文档进行过评价，那么对所有用户都无法作出关于这一文档的评价预测。当不断有新的文档加入时，这一问题尤其显著。

为了解决上述种种问题，这一领域的研究人员做了很多尝试和改进。例如在[45]中，R.J.Mooney 提出了一种由文档内容增强的 CF 算法（Content Boosted Collaborative Filtering，CBCF）。这种方法可以通过已有文档的内容预测用户对未知文档的评价值，一定程度上解决了 Sparsity 和 First-rating 的问题。

#### 4.2.2.2 CF 相关研究

协同过滤算法的关键步骤是相似用户群体的确定。可以说，当前用户的相似群体决定了 CF 算法最终的输出。在 MBCF 中，确定相似用户群体的算法是 Pearson 相关性（Pearson Correlation）或向量空间相似度[46]（Vector Space Similarity）。为了找到一种更好的用来确定用户相似度的方法，研究人员做了如下的尝试。在[45]中，作者提出了一种 hybrid correlation weighting 和 self weighting 机制，用来做最终的评价预测。Hoffman 在[7]中和 Si 在[8]中提出了一种基于模型的 CF 算法（Model-based CF），这种算法可以通过隐元素分析（Latent aspect analysis）将用户聚类成不同的群体。在[39]中，Ganesan 等人证明了在层级结构上计算用户相似度要优于传统方法，同时也能更加符合人们的直观理解。然而在 CF 领域，还没有人考虑到把[39]中的方法应用进来。

## 4.3 自适应用户兴趣模型

#### 4.3.1.1.1.1 模型描述

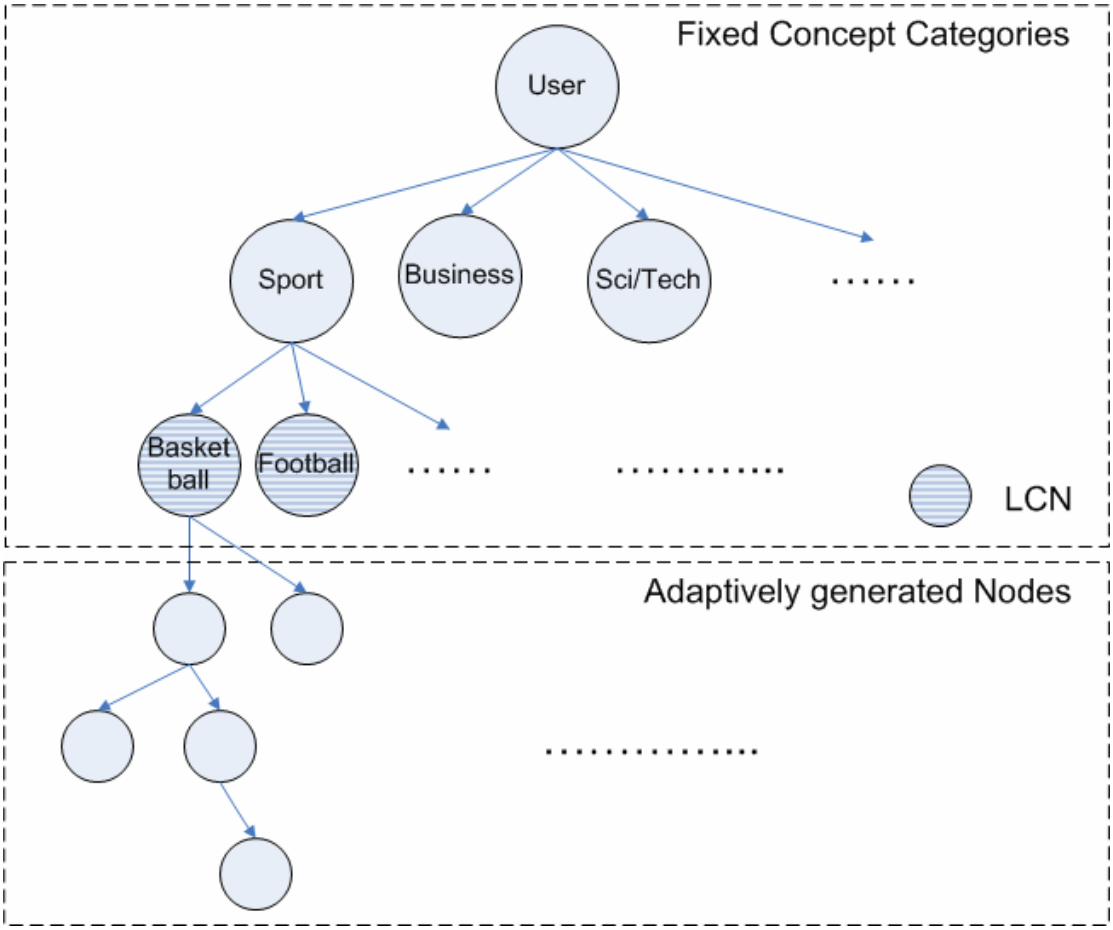


图 4.2 自适应的用户兴趣模型

如上图所示，这一用户兴趣模型包含两个层面。第一个层面（上半部）是固定的概念类别节点，第二个层面是动态的事件节点。

新闻站点通常按照同样的规则对新闻进行分类。我们训练了一个分类器，从多个新闻站点的数据中学习到这一规则，最终得到了一个共同的概念类别层级结构。我们定义此层级的叶子节点为 *Lowest -Category-Nodes*（LCN）。当一个 LCN 节点接收到足够夺得新闻文档之后，处于此节点的用户兴趣可以在更细节的层面上进行描述。为了实现这一目的，我们将此节点分裂，产生新的节点属于下面的动态事件层面。动态事件层面的节点对应着新闻事件。这些事件节点

的内容和层级结构可以随着用户兴趣的变化而改变，这也是我们称之为动态事件层的原因。当用户对某一事件的关注大到一定程度，动态事件层中的节点将按照与 LCN 同样的规则分裂。当用户对某一事件不在感兴趣，该节点将被合并到父节点中，其对用户兴趣的影响将逐渐消失。

### 4.3.2 用户兴趣描述

我们的模型将考虑用户兴趣的两个重要方面：用户所感兴趣的内容（content）和用户感兴趣的强度（vitality）。

#### 4.3.2.1 内容描述

我们采用了经典的 VSM 模型来描述用户兴趣模型中的节点和单个新闻文档，当然，对这二者采用了不同的权重计算方法。

考虑到 TF－IDF 算法的在线计算性能，我们用其来确定新闻文档的关键词权重。

$$W_{i,d} = tf_{i,d} * idf_i \tag{4-3}$$

这里的 $idf_i = \log(N/df_i)$，是关键词 $i$ 的逆文档频率，$N$ 是已处理文档数目，$tf_{i,d}$ 是文档 $d$ 中关键词 $i$ 出现的频率。所有的关键词权重都按照下式进行了归一化。

$$w_{i,d} = w_{i,d} \Big/ \max_i(w_{i,d}) \tag{4-4}$$

模型节点的关键词权重是通过分析划归其上的正例文档而得到的。由于新闻文档存在实效性，每个节点的关键词向量必须周期性的更新，以反映用户兴趣的衰减效果。我们采用了经典的 Rocchio 算法[49]来实现节点关键词权重的更新：

$$V_i^k = (1-\alpha) * \frac{\sum_{d \in D_k} V_d}{\left|D_k\right|} + \alpha * V_i^{k-1} \tag{4-5}$$

其中，$D_k$ 是在周期 $k$ 加入到节点 $i$ 的文档集合。$\left|D_k\right|$ 是 $D_k$ 中文档的数目。$V_d$ 是文档 $d$ 的关键词向量。$V_i^{k-1}$ 和 $V_i^k$ 是节点 $i$ 在周期 $k$-1 和周期 $k$ 的关键词向量。$\alpha$（$\alpha \in [0,1]$）是节点内容的衰减系数。

式 4-5 的非迭代模式如下：

$$V_i^k = (1-\alpha) * \sum_{j=0}^{k} \alpha^{k-j} \frac{\sum_{d \in D_j} V_d}{\left| D_j \right|} \tag{4-6}$$

从 4-6 可以看出，节点的关键词向量是该节点原有内容和新加入文档的加权和。衰减系数 $\alpha$ 表示旧的文档对用户当前兴趣内容影响的损失速度。

#### 4.3.2.2 强度（Vitality）描述

我们为每个节点定义了一个实数来衡量用户对该节点内容的感兴趣程度。我们称之为“能量值”（energy value）[39]

$$E_i^k = \beta * E_i^{k-1} + \sum_{d \in D_k} \cos(V_i^k, V_d) \tag{4-7}$$

这里的 $E_i^{k-1}$ 和 $E_i^k$ 分别表示节点 $i$ 在周期 $k$-1 周期 $k$ 的能量值。$\cos(V_i^k, V_d)$ 表示向量 $V_i^k$ 和 $V_d$ 之间的余弦相似度。$\beta$ （$\beta \in [0,1]$）是节点能量的衰减系数。

式 4-7 的非迭代模式如下：

$$E_i^k = \sum_{j=0}^{k} \beta^{k-j} \sum_{d \in D_j} \cos(V_i^j, V_d) \tag{4-8}$$

从上式可以看出，这里的能量值类似于人工生命理论中的生命指数，控制着用户在该节点上兴趣的生命周期。当有新鲜的文档被分配到该节点时，其能量值会增大，这意味着用户在节点上的兴趣正在增强。由于能量衰减系数的存在，没有新文档加入的节点的能量会逐渐减弱，意味着用户正在失去对该节点的兴趣。

从式 4-8 可以看出，节点的能量 $E_i^k$ 是该节点所有文档与节点向量的余弦相似度之和。所以，能量值既可以表示用户兴趣的强度，又可以表示节点内容的差异程度。然而，决定节点内容差异程度的量还包括节点的文档数目 $|D_{N_i}|$（$D_{N_i}$ 表是属于节点 $N_i$ 的文档集合）。如果某节点的能量值很高而单个文档的平均能量值 $E_i^k / \left| D_{N_i} \right|$ 较低，我们可以推断出用户对该节点对应的事件很感兴趣，然而他感兴趣的内容较为分散。所以我们会很自然的想到将该节点分裂一下，在更细节的层面上来描述用户兴趣内容。如果单文档的平均能量值很高，这意味着用户

对在该节点的兴趣内容很集中，此时尽管节点的能量值可能会很高，我们也不应将节点分裂。如果此时对节点进行了分裂，将会损害节点内容的一致性和削弱用户在节点的兴趣表示。关于分裂和合并操作的更多细节将在 4.3.3 节中阐述。

### 4.3.3 兴趣模型学习算法

每个周期内，在接收到用户评价反馈之后，我们的用户兴趣模型将从用户反馈中进行学习以跟踪用户兴趣变化。我们的研究中只考虑了正例用于学习的情况。学习过程包括两个主要步骤：分配（Assign）和更新（Update）。用户评出的正例首先被分配到对应的节点上，然后整个模型的内容和结构将被更新。

#### 4.3.3.1 分配（Assign）过程

在分配过程中，我们首先判断新加入的新闻文档 $d$ 是否属于用户当前感兴趣的某一细节事件，也就是模型中动态事件节点层的叶子节点。如果不属于任何上述事件，我们将该文档和用户更加综合的兴趣节点进行比较。我们不停重复这一过程，直到该文档被划归到某一事件节点或与其对应的 LCN 节点。分配的过程如图 4.3 所示。

**Assign process:**
For incoming document *d*

0. In the dynamic layer, start from leaf node layer $L_f$, set threshold value *Th*
1. $N_r = \arg\max_{N_i \in L_f} \cos(V_d, V_i)$
2. If $\cos(V_d, V_r) > Th$, then assign $d$ to $N_r$, return

   Else $Th = Th * 0.8$; Set nearest layer above as $L_f$; goto 2
3. Assign *d* to LCN(*d*)

图 4.3 分配过程

#### 4.3.3.2 更新（Update）过程

当所有的新加文档被划归到各自对应的节点之后，整个模型的节点内容和节点结构都将被更新。更新的过程如图 4.4 所示。

**Update Process:**
1. Update every node's *V* and *E* by equation (1) and (3)
2. Perform split or merge when needed

图 4.4 更新过程

当节点 $i$ 的能量 $E_i$ 和差异程度 $E_i^k \Big/ \left|D_{N_i}\right|$ 增长到一定程度时，学习算法将对这一节点进行分裂操作以期更精确的描述用户兴趣。

**Split Process:**
For a given node $N_i$
1. If $E_i > Th_{high}$ and $\bar{E}_i < Th_{average}$, then goto 2, otherwise return.
2. Perform clustering on node $N_i$ by *k*-means, the sub clusters are $\{N_{sub}\}$
3. $N_{sub}^* = \arg\max_{N_{sub} \in \{N_{sub}\}} E_{sub}$; create $N_{sub}^*$ as $N_i$'s child
4. Update vector and energy of $N_{sub}^*$ and $N_i$ according to equation (4-5) and (4-7)

图 4.5 分裂过程

我们采用了 k-均值聚类（k-means clustering）算法对原始节点进行聚类，选择具有最大能量的子类分裂出来作为新的事件节点。我们可以采用最新的事件检测算法如[50]，以改进分裂的精确度。

当一个节点没有新鲜的文档加入时，其能量值将逐渐减弱，意味着用户在逐渐丧失对此节点的兴趣。当某一节点的能量低于某一域值并持续一段时间之后，该节点应该从用户兴趣模型中删除。在删除该节点之前，它对用户更高层面兴趣的影响要通过合并到父亲节点中保留下来。

**Merge Process:**
For a given node $N_i$, suppose its life span is $T_{N_i}$
If $E_i < Th_{low}$ and $T_{N_i} > T_{th}$
Then assign all document in $N_i$ to $N_i$'s parent: $N_{parent}$; update vector and energy of $N_{parent}$ by equation (4-5) and (4-7)

图 4.6 合并过程

### 4.3.4 预测机制

通过学习，拥有了兴趣模型之后，我们可以借助它为每篇新的文档分配一个分值以表示其对用户兴趣的符合程度。然后所有的新闻按照这个分值排序，排名最高的 $N$ 篇新闻被返回给用户。

对符合程度的判断既要考虑文档和节点内容的相似程度也要考虑节点的能量值。具体说来，对于一篇新的新闻文档 $d$,我们首先通过类似与 assign 过程中的方法找到与 $d$ 最相关的节点，然后通过下式计算分值：

$$P_p(u,d) = E_n * \cos(V_d, V_n) \tag{4-9}$$

$P_p(u,d)$ 代表由用户 $u$ 的兴趣模型对文档 $d$ 的预测分值。对于第 $k$ 周期的新文档集合 $D_k$ 而言，每篇文档的预测分值按照下式进行归一化：

$$P_p(u,d) = \frac{P_p(u,d)}{\underset{d \in D_k}{MAX}\{P_p(u,d)\}} \tag{4-10}$$

## 4.4 基于自适应用户兴趣模型的协同过滤

当每个用户的兴趣模型被建立起来之后，我们可以利用这些信息进行协同过滤。我们将用 MBCF 来表示 memory-based collaborative filtering，用 CBCF 来表示 content-boosted collaborative filtering，用 PBCF 来表示基于我们的兴趣模型的 CF 算法（profile-based collaborative filtering）。

### 4.4.1 基于兴趣模型的 CF 算法（PBCF）

为了解决 sparsity 的问题，我们为每个用户 $u$ 定义一个伪评价向量（pseudo user-rating vector）：

$$v_{u,d} = \begin{cases} 1 & : \text{if user } u \text{ browsed document } d \\ P_p(u,d) & : \text{otherwise} \end{cases} \tag{4-11}$$

$P_p(u,d)$ 代表由用户兴趣模型得出的预测值。把所有用户的伪评价向量合并在一起就可以得到密集的伪评价矩阵 $V$。然后我们在这一矩阵上进行 CF 操作。

接下来的一个关键问题是用户相似度的计算。在我们的兴趣模型中，概念

分类层级对所有用户有着共同的固定结构，所以我们很自然的想到了通过利用这一共同结构来计算用户相似度。

计算树性结构相似度的很常用的方法是编辑距离[51]（Edit Distance）。然而，编辑距离的一个很重要的缺陷是很难确定单位操作（替换，插入，删除等）的损耗。我们采用了 Ganesan 等人提出的 Optimistic Genealogy Measure（OGM）[13]来在此层级结构上计算用户相似度。

假设有两个用户，其阅读的集合是$C_1$和$C_2$。二者之间的相似度可由下式求得：

$$sim(C_1,C_2)=\frac{\sum_{d_i\in C_1} leafsim_{T_1,T_2}(d_i)*W(d_i)}{\sum_{d_i\in C_1} W(d_i)} \quad (4\text{-}12)$$

这里的$T_k$是用户 $k$ 的概念分类层级的 induced tree。$leafsim_{T_1,T_2}(d_i)$的取值衡量了新闻$d_i$和$T_2$的相似程度，由前述基于兴趣模型的预测方法计算。$W(d_i)$是文档$d_i$的先验权重，在本研究中我们取逆用户频率作为$W(d_i)$。

因为 OGM 算法是非对称的，最终的用户相似度用采用下式计算：

$$sim(u_1,u_2)=\frac{sim(C_1,C_2)+sim(C_2,C_1)}{2} \quad (4\text{-}13)$$

我们选择与当前用户 $a$ 最为相似的 $M$ 个用户做最终的预测。将当前用户 $a$ 的伪评价向量与相似的 $M$ 个用户混合得到最终的结果。然而，我们想要加强来自 $a$ 本身的兴趣模型的预测，像[45]中的做法一样，我们引入了 SelfWeighting 因子：

$$sw_a=\begin{cases}\frac{n_a}{50}\times max : \text{if } n_a<50\\ max : \text{otherwise}\end{cases} \quad (4\text{-}14)$$

这里的$n_a$是当前用户 $a$ 浏览过得新闻数目。参数 $max$ 代表对兴趣模型预测结果的置信度。对用户 $a$ 的最终预测结果按照下式计算：

$$P(a,d)=\bar{v}_a+\frac{sw_a(v_{a,d}-\bar{v}_a)+\sum_{\substack{u=1\\u\neq a}}^{M} sim(a,u)P_{a,u}(v_{u,d}-\bar{v}_u)}{sw_a+\sum_{\substack{u=1\\u\neq a}}^{M} sim(a,u)P_{a,u}} \quad (4\text{-}15)$$

在上式中，$\bar{v}_a$ 是预测平均值。$v_{a,d}$ 是用户 $a$ 的伪评价值。$sw_a$ 是 SelfWeighting 因子。$sim(a,u)$ 采用 4-13 计算得到。$P_{a,u}$ 是用户 $a$ 和 $u$ 之间的 Pearson 相关系数。

### 4.4.2 MBCF，CBCF 和 PBCF 的比较

通过将 PBCF 和传统的 MBCF，CBCF 方法对比，我们能更好的看清我们的方法的优越性。

表 4.1 不同 CF 算法的比较

| | **MBCF** | **CBCF** | **PBCF** |
|---|---|---|---|
| **User Similarity** | By rating | By rating | By profile |
| **Rating Prediction** | N/A | By page | By profile |
| **Sparsity Problem** | Y | N | N |
| **First-rating Problem** | Y | N | N |

从表 4.1 我们可以看出 MBCF 不具备评价预测的能力，而且有着 Sparsity 和 First-rating 问题。 CBCF 可以通过页面文档内容的预测解决这两个问题。而在我们的 PBCF 方法中，用户相似度和评价预测都是来自于用户兴趣模型，更加精确。所以通过 PBCF，我们可以得到更好的预测结果，而且本方法能够更好的跟踪用户兴趣变化。4.6.4.3 中的实验结果验证了我们方法的优势。

## 4.5 统一的新闻个性化推介框架

下图展示了我们统一的推介框架结构。它可以根据用户自身的兴趣和相似用户群体的信息，为用户提供经过排序后的新闻，以尽可能的满足用户的兴趣需求。这一框架属于推介系统中 *Find Good Items* 类别[52]。此类系统是个性化推介领域的核心系统，在学术界引起了广泛的关注，在工业界也有很多实际的应用。

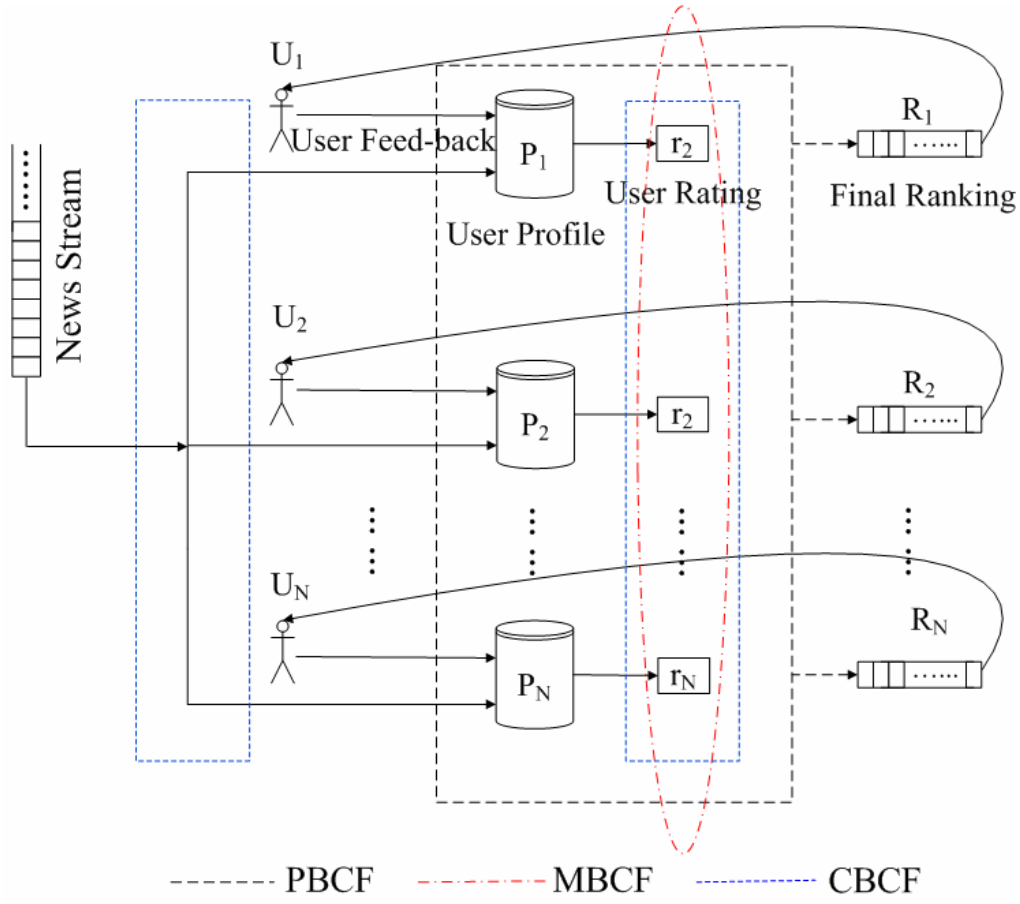


图 4.7 统一的新闻个性化推介框架

新的文档首先被分配的各个用户的兴趣模型中($P_1$~ $P_n$)，然后系统根据用户兴趣模型生成评价预测。在这之后，再有我们的 PBCF 算法生成最终的评价值预测。所有的新加文档都将按照其预测评价值排序。排名最高的 *N*（我们取 *N*=20）篇文档将被返回给用户。然后，用户的阅读反馈给系统，用来更新兴趣模型。这样，我们得到了一个将 CBF 和 CF 统一起来的新闻推介框架。如果没有 CF 的过程，我们的系统将退化成类似于 PVA 的单用户推介系统。

如图 4.7 所示，在 PBCF（黑色矩形框）中，兴趣模型不光被用来对新加文档进行预测，同时还参与了用户相似度的计算。而传统的 MBCF（红色椭圆框）之采用了纯粹的用户评价。CBCF（蓝色矩形框）利用了用户评价和文档内容。

## 4.6 实验

### 4.6.1 实验数据

我们使用了来自美国某大学联盟的 10 个 http 代理服务器上的访问纪录。这些代理服务器的用户主要来自于联盟中的大学。考虑到我们的需求，我们将用户阅读新闻的条目从这些访问纪录中抽取出来。这里借助了一个我们在第三章提到的 URL 分类器，用来判断访问纪录是否为新闻站点。此处我们做了一个假设，即每个独立 IP 对应着一个用户，这在实际情况中是合理的。下表是对新闻访问纪录的一些统计。

表 4.2 新闻访问纪录数据统计

| **Time Span** | 2002 3.20 ~ 2002 3.31 |
|---|---|
| **Total user (IP) number** | 10765 |
| **News browsing log number (all users)** | 112060 |
| **Unique news page number (all users)** | 38922 |
| **Users* number (who browsed for more than 8 days)** | 342 |
| **News browsing log number (Users*)** | 80615 |
| **Unique news page number (Users*)** | 21788 |
| **Access number per day per person (Users*)** | 19.64 |

从上表我们可以看出，新闻访问纪录来自于 2002 年 3 月中的 12 天，大约两周的时间。在这 12 天内，共有 10765 个用户有阅读新闻的行为，共阅读新闻共计 112060 次。这些新闻包含了 38922 篇独立的页面，这样每篇新闻平均被 112060/38922=2.9 个用户访问过。从这一数据，我们可以看出，这一用户群体在新闻阅读上有着相当的共同偏好。我们进而可以利用这一信息进行协同过滤。为了避免数据过于稀疏，我们选取了 12 天内有 8 天以上的时间会阅读新闻的用户作为实验数据源（表 4.2 中的 User*），我们认为这些数据反映的用户有着稳定的新闻阅读习惯，个性化新闻推介对其的作用会比较明显。同时，在这一用户群体中，阅读偏好的重叠更加明显，每篇新闻被不同用户访问的次数为 80615/21788=3.7，明显高于前面所有用户的 2.9。所以，我们有理由相信 CF 算法在这一用户群体中更能发挥作用。

### 4.6.2 实验方法和评价标准

由于在本文的新闻推介问题中，计算上所需的时间耗费与推介周期（天）相比是微不足道的。所以，我们的实验将集中在考察预测算法的准确度上。

我们使用了传统的 IR 评价标准查准率（precision）来衡量我们算法的性能。计算公式如下：

$$Precsion=\frac{\left|D_{visited}\right|}{N_s} \tag{4-16}$$

这里的 $N_s$ 是由我们的推介系统推荐的新闻总数，$D_{visited}$ 是这 $N_s$ 个文档中用户实际访问过的页面集合。

依据 4.5 中的框架，我们实现了一个系统 MyNews 来验证我们的方法。为了显示我们的方法比前人方法的改进，我们设计了如下实验。

首先，我们在 MyNews 上进行了在线的模拟实验，没有采用 PBCF 算法，将实验结果和采用了静态 PVA 兴趣模型的结果进行比较。

其次，我们将 PBCF 加入进来，将加入 PBCF 后的实验结果和加入 PBCF 之前的实验结果进行对比，以展示 PBCF 算法的功效。

最后，我们将 PBCF 算法的结果和传统 MBCF，CBCF 的结果比较，以说明 PBCF 算法更适用于层级结构的兴趣模型。

### 4.6.3 参数的选取

在进行最终的实验之前，我们考察了各种方法对参数的敏感度。我们在各种算法的敏感度曲线上确定了各自最优的参数，在余下的实验中都使用了这组参数。

为了确定参数敏感度，我们选择了前 10 天的数据，并将其分成了训练集和测试集两部分。我们在其上进行了 10 重交叉验证，用查准率作为评价指标，以最终确定参数选取。

### 4.6.4 实验结果

本节中将用 MyNews 代指我们的用户兴趣模型。

#### 4.6.4.1 动态兴趣模型 v.s. 静态兴趣模型

我们实现了 PVA 算法并将其与 MyNews 进行了对比。实验的过程如下：我们随机的选出 50 个用户。我们从代理记录数据中抽取出每天 50 条新闻纪录作为测试集。我们将这 50 个用户的实际浏览记录作为用户反馈，系统在这一基础上学习出用户的兴趣模型，同时将第二天的预测排序结果返回给用户，同时再将用户的实际浏览记录反馈给系统，如是循环。我们将此实验在连续的 10 天数据上运行了 10 个周期，同时以系统对这 50 个用户的平均查准率作为性能指标。在本实验中，我们将 $N_s$ 分别取为 5，10，20 以考察算法在不同 scope 下的表现。实验结果如下图所示。

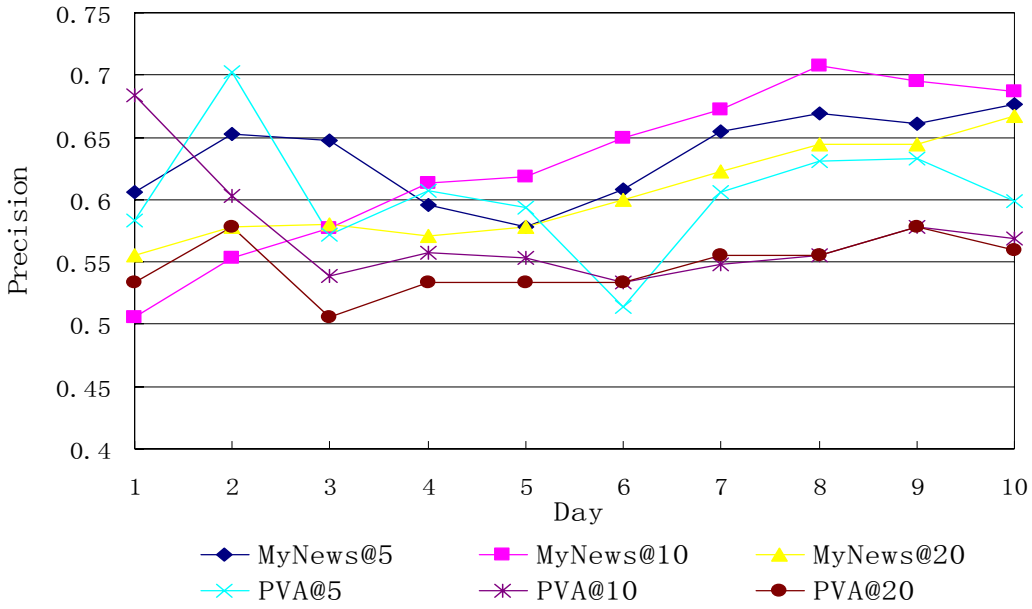


图 4.8 MyNews 和 PVA 的比较

从上图可以看出，在任何一个 scope，MyNews 都显著的超过了 PVA。这主要是因为 MyNews 可以在事件的层面上对用户兴趣进行建模，所以，它比 PVA 能更好的跟踪用户兴趣细微的变化。

在第三天的时候，PVA 在这 3 个 scope 的性能都有明显下降，而 MyNew 则维持了比较好的表现。这是因为在第三天的时候一个热门的体育事件（NFL Final）结束了，大多数用户都转向了体育类另一个热门事件（NBA Play offs）。而 PVA 的兴趣描述向量是对应着整个体育新闻类别的，无法从内容上描述这一

剧烈变化。而 MyNews 为每一个热门事件生成了一个独立的结点，所以一个事件结点的消失不会对另一个事件结点造成太大的影响。

#### 4.6.4.2 单纯兴趣模型 v.s. 兴趣模型+协同过滤

为了展示 PBCF 的功效。我们将加入 PBCF 后的实验结果和加入 PBCF 之前的实验结果进行了对比。此处的 $N_s$ 取为 20，实验结果如下图所示。

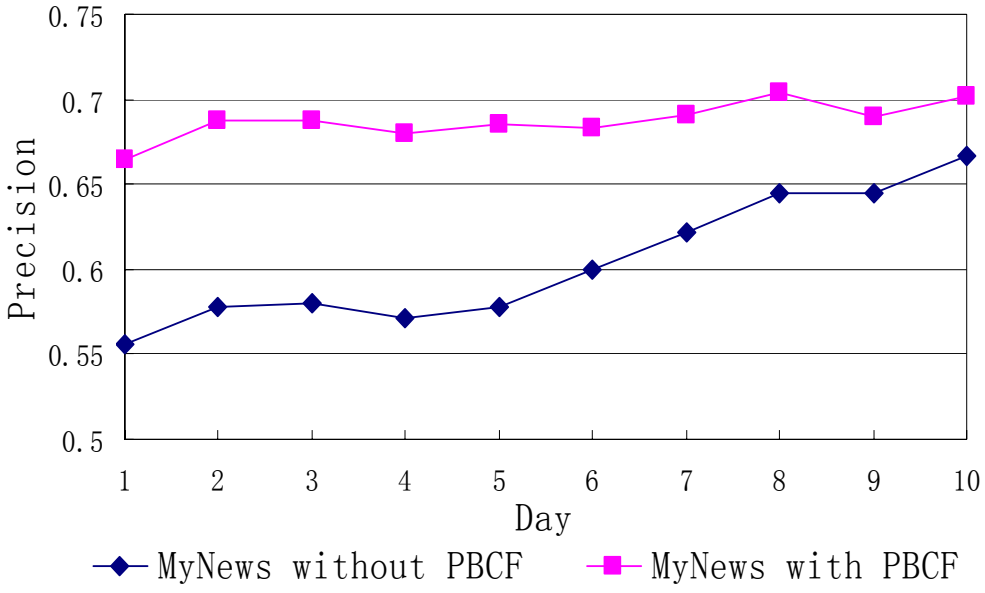


图 4.9 有无 PBCF 的结果对比

从图 4.9 可以看出，加入了 PBCF 的 MyNews 性能远远超出为加入 PBCF 的 MyNews。加入 PBCF 之后的查准率平均提高了 8.2%。这是因为实验用户群体间的兴趣重叠比较明显，所以 CF 算法有着很好的效果。随着训练过程的进行，这两种算法之间的差距在逐渐缩小。这一现象有两个原因。首先，基于单纯兴趣模型的 MyNews 随着训练数据的增加其精确度必然会提高；其次，由于 PBCF 采用的是来自多个用户的预测结果来进行协同过滤，所以当单个用户兴趣模型的精确度提高时，由协同过滤带来的改进会相对减少。

#### 4.6.4.3 几种协同过滤算法的比较

我们实现了 MBCF 算法和 CBCF 算法，并将其与 MyNews 兴趣模型组合在一起进行了实验，实验结果与 PBCF 的对比如下图。（$N_s$=20）

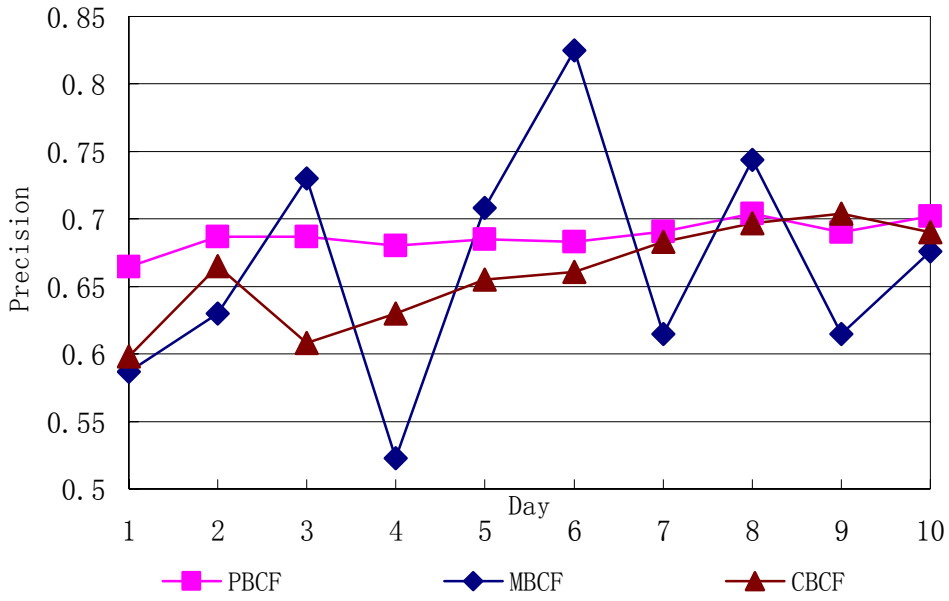


图 4.10 MBCF，CBCF 和 PBCF 的比较

从上图可以看出，MBCF 性能的方差要远大于 CBCF 和 PBCF。这是由于 MBCF 算法能利用到的用户评价数据非常稀疏，增加了算法的随机性。

这三者的平均查准率非常接近，PBCF 稍高于 MBCF 和 CBCF。

在第三天用户兴趣发生了明显变化时，我们的 PBCF 算法能更流畅的跟踪用户兴趣变化，而 CBCF 算法需要更多的时间来调整跟踪。

## 4.7 本章小结

在本章中，我们要解决的是新闻个性化推介问题。我们提出了一个全新的自适应兴趣模型。这一模型包括两大层面。其中，上层由固定的概念分类节点构成，下层由动态事件节点构成。这一结构可以自适应的对用户在概念分类和事件上的兴趣及兴趣变化进行建模。基于这一模型中固定的概念分类层级结构，我们提出了一个改进的 CF 算法。通过利用蕴含在层级结构中的信息计算用户相似度，我们的 CF 算法比传统的 MBCF 和 CBCF 能找到更好的相似用户群体。实验结果证明了我们的方法在推介准确度上的改进。

下一步的工作主要包括四方面内容：

首先，我们将采用更先进的事件检测算法来改进节点分裂过程。这将进一

步的改进兴趣模型的预测精度。

其次，我们现在的方法利用了概念分类层面的固定层级信息，将来我们考虑利用事件层面的动态层级结构信息来改进 PBCF 算法。

第三，我们考虑采用层级贝叶斯模型（Hierarchical Bayesian Model）来对我们当前的推介框架重新进行建模。

最后，我们将在这一问题中引入主动学习的方法（active learning）以辅助用户兴趣模型的学习过程。

# 第5章 结论和展望

在本文中，作者针对大规模通用网页检索的排序函数学习问题，网络新闻流行度排序问题以及网络新闻个性化推介问题，介绍了自己的解决方法。作者工作的贡献主要包括如下三个方面。

首先，作者针对当前排序函数的学习中，有标注训练样本普遍不足的情况，提出了通过基于流形排序算法的半监督学习机制，将少量已标注样本的相关值传递到其他未标注样本上，进而大大增加排序函数学习可用样本数。基于此种相关度传递机制——MRBRP（Manifold Ranking Based Relevance Propagation），作者提出了一套完整的排序函数的学习框架，在主流商用搜索引擎的海量数据集合上的实验表明，此种学习框架可以显著提高排序函数的性能；同时此框架具有极大的灵活性，可使用 MRBRP 来配合任何最先进的排序函数学习算法。

接下来，作者分析了新闻首页，新闻页面，和新闻事件之间的相关增强关系。作者使用了一个加权无向三分图模型对三者进行了建模。通过对新闻首页视觉推介强度分析和多站点间新闻内容相似度分析，作者得到了该三分图的权重的一种构建方法。基于此三分图，作者提出了利用关联矩阵特征向量的平衡解求解方法。从这一平衡解，我们可以得的三者分别的排序。同时，新闻的时效性问题在本算法中也有所反映。在多个实际新闻站点数据上的实验结果表明此算法具有良好的效果和用户体验。

最后，作者针对现有的主流用户兴趣模型在描述事件层面上用户兴趣的不足，提出了一个具有上下两个层面的层级模型。其中，上层由固定的概念分类节点构成，下层由动态事件节点构成。这一结构可以自适应的对用户在概念分类和事件上的兴趣及兴趣变化进行建模。同时此层级模型的节点结构和节点内容还可以随着用户的阅读内容变化而改变，能更好的跟踪用户的兴趣。为了利用相似用户群体的兴趣信息，作者又提出了一种基于此模型上层层级结构的改进协同滤波算法，能更好的发现相似用户群体。在实际用户数据上的实验结果表明，本用户兴趣模型和协同滤波算法可显著的提高用户个性化新闻推介的准确度。

由于作者的时间有限，这三个问题分别还有很多值得继续研究的内容有待进一步发掘。

在通用网页检索排序函数学习问题中，我们可以进一步的考察更多的半监督学习机制来实现相关度传递。我们还需进一步研究配套排序算法的选择和改进。另外，查询分类也是更好的解决此问题的一个必要步骤。

在新闻流行度排序问题中，我们还需继续挖掘网络上蕴含的其他相关信息来辅助新闻流行度的确定。我们还需考察更先进的事件检测算法和新闻聚类算法来改进最终展示给用户的结果。

在新闻个性化推介问题中，我们可引入更先进的事件检测算法来改进节点分裂过程。以改进兴趣模型的预测精度。我们也可采用主动学习的方法（active learning）以辅助用户兴趣模型的学习过程。另外我们考虑进一步的利用事件层面的动态层级结构信息来改进协同过滤算法。我们也可以采用层级贝叶斯模型（Hierarchical Bayesian Model）来对我们当前的推介框架重新进行建模。

此外，多站点新闻流行度问题中涉及到的网络版权问题和个性化新闻推介中涉及到的用户隐私问题也十分值得我们去探讨。

# 参考文献

[1] Jue Wang, Jinyi Yao, Zengqi Sun, et al. Adaptive User Profile Model and Collaborative Filtering for Personalized News. In APWeb 2006, LNCS 3841, pp. 474 – 485, 2006. APWeb 2006 Best Student Paper Award. (SCI Indexed)

[2] Jinyi Yao, Jue Wang, Zengqi Sun, et al. Ranking Web News via Homepage Visual Layout and Cross-site Voting. In ECIR 2006, LNCS 3936, pp. 131 – 142, 2006. (SCI Indexed)

[3] Jue Wang, Mingjing Li, et al. Augmenting Ranking Function by Label Propagation. Accepted by AIRS 2006, in LNCS series. (SCI Indexed)

[4] http://www.iresearch.com.cn/html/online_users/detail_news_id_28895.html

[5] http://www.iresearch.com.cn/html/online_users/detail_news_id_28746.html

[6] 2004 Web Usage Survey Results. Sponsored by Cerberian and SonicWall. http://www.cerberian.com/content/CerberianSonicWallSurveyResults.pdf

[7] http://bbs.yt165.com/printpage.asp?BoardID=6&ID=19743

[8] www.alex.com

[9] A. Broder. A taxonomy of web search. ACM SIGIR Forum, 2002.

[10] Ranking a stream of News, WWW 2005

[11] TDT 2004: Annotation Manual

[12] L Page, S Brin, R Motwani, T Winograd. The PageRank Citation Ranking: Bringing Order to the Web.Stanford Digital Library working paper SIDL-WP-1999-0120.

[13] J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. Journal of the ACM, Vol. 46, No. 5, September 1999, pp. 604 –632.

[14] A vector space model for automatic indexing. G Salton, A Wong, CS Yang. Communications of the ACM, 1975

[15] Yoav Freund, Raj Iyer, Robert E. Schapire, Yoram Singer. An Efficient Boosting Algo-rithm for Combining Preferences. Journal of Machine Learning Research 4 (2003) 933-969, 2003.

[16] Chris B., et al. Learning to Rank using Gradient Descent. In Proceedings of the Twenty-second International Conference on Machine Learning, ICML 2005.

[17] D. K. Harman. Overview of the Third Text REtrieval Conference (TREC-3). In Proceed-ings of the Third Text REtrieval Conference (TREC-3). NIST Special Publication 500-226, 1995.

[18] R. Herbrich, T. Graepel & K. Obermayer. Large margin rank boundaries for ordinal regression. Advances in Large Margin Classifiers, MIT Press. (pp. 115-132

[19] K. Crammer & Y. Singer. Pranking with ranking. NIPS, 2002.

[20] E. Harrington. Online ranking/collaborative filtering using the Perceptron algorithm. In Proceedings of the Twentieth International Conference on Machine Learning, ICML 2003.

[21] W. Cohen, R. E. Schapire & Y. Singer. Learning to order things. Journal of Artifi-cial Intelligence Research, 10, 243–270, 1999.

[22] Yoav Freund, Raj Iyer, Robert E. Schapire, Yoram Singer. An Efficient Boosting Algorithm for Combining Preferences. Journal of Machine Learning Research 4 (2003) 933-969, 2003.

[23] B. Chris, et al. Learning to Rank using Gradient Descent. In Proceedings of the Twenty-second International Conference on Machine Learning, ICML 2005.

[24] Klaus Brinker. Active Learning of Label Ranking Functions. In Proceedings of the Twenty-first International Conference on Machine Learning, ICML 2004.

[25] D. Zhou, et al. Learning with local and global consistency. NIPS, 2003.

[26] D. Zhou, et al. Ranking on data manifolds. NIPS, 2003.

[27] J. Shi, and J. Malik. Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, pp. 888-905, 2000.

[28] Pass, G. Comparing images using color coherence vectors. In Proceedings of the Fourth ACM International Conference on Multimedia, ACMM 1997.

[29] M. Stricker, and Orengo, M. Similarity of color images. Storage and Retrieval for Image and Video Databases, Proceedings of SPIE 2420, pp 381-392, 1995.

[30] Jingrui He, Mingjing Li, Hong-Jiang Zhang, et al. Manifold-ranking based image retrieval. In Proceedings of the 12th Annual ACM International Conference on Multimedia, ACMM 2004.

[31] Jarvelin, K., Kekalainen, J. IR evaluation methods for retrieving highly relevant documents. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000

[32] J. Allan, etc.(Ed.) Topic Detection and Tracking. Springer 2002

[33] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In Proceedings of the Broadcast News Understanding and Transcription Workshop, pages 194{218, 1998.

[34] J. Allan, V. Lavrenko, and H. Jin. First story detection in TDT is hard. In Proceedings of the Ninth International Conference on Information and Knowledge Management, pages 374{381, 2000.

[35] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan and A.S. Tomkins. The Web as a graph: measurements, models and methods, In Proc. 5th Int. Computing and Combinatorics. 1999

[36] R. Cooley, B. Mobasher, J.Srivastava. Web Mining: Information and Pattern Discovery on the World Wide Web. In the Ninth International Conference on Tools with Artificial Intelligence (ICTAI'97). 1997

[37] R. Kosala and H. Blockeel. Web Mining Research: A Survey(2000). SIGKDD Explorations – Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining 2(1), pp. 1-15, July, 2000

[38] S. Chakrabarti, BE Dom, SR Kumar, P.Raghavan, S. Rajagopalan, A. S. Tomkins, D. Gibson, J. M. Kleinberg. Mining the Web's Link Structure. IEEE Comp, 1999

[39] Deng Cai, Shipeng Yu, Ji-Rong Wen and Wei-Ying Ma. VIPS: a Vision-based Page Segementation Algorithm. Microsoft Technical Report, MSR-TR-2003-79, 2003.

[40] D. Billsus, and M. J. Pazzani. A Personal News Agent that Talks, Learns and Explains. In Proceedings of the Third International Conference on Autonomous Agents, 1999.

[41] D. H. Widyantoro, T. R. Ioerger, and J.Yen. An Adaptive Algorithm for Learning Changes in User Interests. In Proceedings of the Eighth International Conference on Information and Knowledge Management (CIKM'99), 1999.

[42] C. C. Chen, M. Chen, and Y. Sun. PVA: A Self-Adaptive Personal View Agent. In Proceedings of the Seventh ACM SIGKDD international conference on Knowledge discovery and data mining, 2001

[43] N. Nanas, V. Uren, and A. D. Roeck. Building and applying a concept hierarchy representation of a user profile. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, 2003

[44] P. Ganesan. H. Garcia-Molina, and J. Widom. Exploiting hierarchical domain structure to compute similarity. ACM Transactions on Information Systems (TOIS) archive Volume 21 , Issue 1, January 2003

[45] A. Pretschner, and S. Gauch. Ontology Based Personalized Search. In 11th IEEE Intl. Conf. On Tools with Artificial Intelligence, 1999

[46] P. Ganesan. H. Garcia-Molina, and J. Widom. Exploiting hierarchical domain structure to compute similarity. ACM Transactions on Information Systems (TOIS) archive Volume 21 , Issue 1, January 2003

[47] Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., and Riedl, J. GroupLens: An Open Architecture for Collaborative Filtering of Netnews, In Proceedings of the Conference on Computer Supported Collaborative Work, 1994. 175-186.

[48] Prem Melville and Raymond J. Mooney and Ramadass Nagarajan. Content-Boosted Collaborative Filtering for Improved Recommendations. AAAI 2002

[49] J. S. Breese, D. Heckerman and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceeding of the Fourteenth Conference on Uncertainty in

Artificial Intelligence (UAI), 1998.

[50] T. Hofmann. Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, 2003

[51] L. Si, and R. Jin. Flexible mixture model for collaborative filtering. In Proceedings of the Twentieth International Conference on Machine Learning, 2003

[52] T. Joachims. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In Proceedings 14th International Conference on Machine Learning, 1997

[53] Z. Li, B. Wang, and M. Li. Probabilistic Model of Retrospective News Event Detection. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, 2005

[54] P. N. Klein. Computing the Edit-Distance between Unrooted Ordered Trees. Lecture Notes in Computer Science, Volume 1461, Chapter p. 91, 2003

[55] J. Herlocker, J. Konstan, and L. G. Terveen. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), Volume 22 Issue 1, 2004

[56] K. Yu, V. Tresp, and S.Yu. A Nonparametric Hierarchical Bayesian Framework for In-formation Filtering. In Proceedings of the 27th annual international conference on Research and development in information retrieval, 2004

[57] Baum, E., & Wilczek, F. (1988). Supervised learning of probability distributions by neural networks. Neural Information Processing Systems (pp. 52-61).

[58] Bradley, R., & Terry, M. (1952). The Rank Analysis of Incomplete Block Designs 1: The Method of Paired Comparisons. Biometrika, 39, 324-245.

[59] Bromley, J., Bentz, J.W., Bottou, L., Guyon, I., Le-Cun, Y., Moore, C., Sackinger, E., & Shah, R.(1993). Signature Verication Using a "Siamese" Time Delay Neural Network. Advances in Pattern Recognition Systems using Neural Network Technologies, World Scientific (pp. 25-44)

[60] Burges, C. (1996). Simplified support vector decision rules. Proc. International Conference on Machine Learning (ICML) 13 (pp. 71-77).

[61] Caruana, R., Baluja, S., & Mitchell, T. (1996). Using the future to \sort out" the present: Rankprop and multitask learning for medical risk evaluation. Advances in Neural Information Processing Systems (NIPS) 8 (pp. 959-965).

[62] Dekel, O., Manning, C., & Singer, Y. (2004). Loglinear models for label-ranking. NIPS 16.

[63] Kimeldorf, G. S., &Wahba, G. (1971). Some results on Tchebychefian Spline Functions. J. Mathematical Analysis and Applications, 33, 82-95.

[64] LeCun, Y., Bottou, L., Orr, G. B., & Mfiuller, K.-R.(1998). Efficient backprop. Neural Networks: Tricks of the Trade, Springer (pp. 9-50).

[65] Mason, L., Baxter, J., Bartlett, P., & Frean, M. (2000). Boosting algorithms as gradient descent. NIPS 12 (pp. 512-518).

[66] Mitchell, T. M. (1997). Machine learning. New York: McGraw-Hill.

[67] Refregier, P., & Vallet, F. (1991). Probabilistic approaches for multiclass classification with neural networks. International Conference on Artificial Neural Networks (pp. 1003-1006).

[68] Scholkopf, B., & Smola, A. (2002). Learning with kernels. MIT Press.

[69] C. C. Aggarwal, J. Han, J. Wang, and P. Yu. Clustream: A framework for clustering evolving data streams. In Proceedings of the 29th International Conference on Very Large Data Bases, 2003.

[70] M. Atallah and R. Gwadera. Detection of significant sets of episodes in event sequences. In Proceedings of the International Data Mining Conference, pages 3–10, 2004.

[71] S. Chung and D. McLeod. Dynamic topic mining from news stream data. In Proceedings of International Conference on Ontologies, Databases and Applications of Semantics, pages 653–670, 2003.

[72] S. Blair-Goldensohn, D. R. Radev, Z. Zhang, and R. S. Raghavan. Interactive, domain-independent identification and summarization of topically related news articles. In Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries, pages 225–238, 2001.

[73] S. Blair-Goldensohn D. R. Radev, Z. Zhang, and R. S. Raghavan. Newsinessence: A system for domain-independent, real-time news clustering and multi-document summarization. In Proceedings of the Human Language Technology Conference, 2001.

[74] D. de Castro Reis, P. Golgher, A. da Silva, and A. Laender. Automatic web news extraction using tree edit distance. In Proceedings of the 13th International WWW Conference, pages 502–511, 2004.

[75] D. M. Dunlavy, J. P. Conroy, and D. P. O'Leary. Qcs: A tool for querying, clustering, and summarizing documents. In Proceedings of the Human Language Technology Conference-NAACL, pages 11–12, 2003.

[76] E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: Providing personalized newsfeeds via analysis of information novelty. In Proceedings of the 13th International WWW Conference, pages 482–490, 2004.

[77] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams: Theory and practice. In IEEE Transactions on Knowledge and Data Engineering, pages 515–528, 2003.

[78] M. Henzinger, B. Chang, B. Milch, and S. Brin. Query-free news search. In Proceedings of the 12th International WWW Conference, pages 1–10, 2003.

[79] M. G. Kendall. A new measure of rank correlation. Biometrika, 430:81–93, 1938.

[80] H. Mannilla, H. Toivonen, and A. Verkamo. Discovery of frequent episodes in event sequences. In Proceedings of the Data Mining and Knowledge Discovery, pages 259–289, 1997.

[81] S. Muthukrishnan. Data streams: algorithms and applications. In Proceedings of the ACM-SIAM 14th Symposium on Discrete Algorithms, page 413, 2003.

[82] C. Spearman. The proof and measurement of association between two things. American Journal of Psychology, 15:72–101, 1904.

[83] T. Cover, P. Hart, "Nearest neighbor pattern classification," IEEE Trans. Information Theory IT-13(1967), pp. 21-27.

[84] R. Ehrich, J. Foith, "Representation of Random Waveforms by Relational Trees," IEEE Trans. Computers, C25:7(1976).

[85] S. Fine, Y. Singer, N. Tishby, "The hierarchical hidden Markov model: Analysis and applications," Machine Learning 32(1998).

[86] V. Guralnik, J. Srivastava, "Event detection from time series data," Intl. Conf. Knowledge Discovery and Data Mining, 1999.

[87] D. Hand, H. Mannila, P. Smyth, Principles of Data Mining, MIT Press, 2001.

[88] S. Havre, B. Hetzler, L. Nowell, “ThemeRiver: Visualizing Theme Changes over Time," Proc. IEEE Symposium on Information Visualization, 2000.

[89] D. Hawkins, ”Point estimation of the parameters of piecewise regression models," Applied Statistics 25(1976)

[90] J. Helfman, C. Isbell, "Ishmail: Immediate identification of important information," AT&T Labs Technical Report, 1995.

[91] E. Keogh, P. Smyth, "A probabilistic approach to fast pattern matching in time series databases," Proc. Intl. Conf. Knowledge Discovery and Data Mining, 1997.

[92] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, J. Allan, "Mining of Concurrent Text and Time-Series," KDD-2000 Workshop on Text Mining, 2000.

[93] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz. "A Bayesian approach to filtering junk email," Proc. AAAI Workshop on Learning for Text Categorization, 1998.

[94] S.L. Scott, Bayesian Methods and Extensions for the Two State Markov Modulated Poisson Process, Ph.D. Thesis, Harvard University, Dept. of Statistics, 1998.

[95] R. Swan, J. Allan, "Extracting significant time-varying features from text," Proc. 8th Intl. Conf. on Information Knowledge Management, 1999.

[96] R. Swan, J. Allan, "Automatic generation of overview timelines," Proc. SIGIR Intl. Conf. Information Retrieval, 2000.

[97] S. Whittaker, C. Sidner, \E-mail overload: Exploring personal information management of

e-mail," Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems, 1996.

[98] Y. Yang, T. Ault, T. Pierce, C.W. Lattimer, "Improving text categorization methods for event tracking," Proc. SIGIR Intl. Conf. Information Retrieval, 2000.

[99] Y. Yang, T. Pierce, J.G. Carbonell, "A Study on Retrospective and On-line Event Detection," Proc. SIGIR Intl. Conf. Information Retrieval, 1998

[100] http://www.bbcworld.com/.

[101] http://www.boston.com/.

[102] http://www.cnn.com/.

[103] http://www.e-marketing-news.co.uk/Aug04-Greg.html.

[104] http://www.findory.com/.

[105] http://news.google.com/.

[106] http://newsbot.msnbc.msn.com/.

[107] http://www.nielsen-netratings.com/.

[108] http://www.channelnewsasia.com/.

[109] http://www.newsinessence.com/.

[110] http://www.rednova.com/.

[111] http://www.reuters.com/.

[112] http://searchenginewatch.com/.

[113] http://news.yahoo.com/.

[114] http://news.baidu.com/.

# 致　谢

衷心感谢导师孙增圻教授对本人的精心指导和倾力支持。他的言传身教将使我终生受益。

在微软亚洲研究院的实习期间，承蒙李明镜教授热心指导与帮助，不胜感激。

感谢实验室的钱宗华等其他老师同学对本文研究工作的支持和帮助。

本课题承蒙国家自然科学基金资助，特此致谢。

最后我要谢谢我的女朋友，是她让我心情愉快的写下了本文最后一句话。

# 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：　　日　期：

# 个人简历、在学期间发表的学术论文与研究成果

## 个人简历

1999 年 9 月考入清华大学自动化系自动化专业，2003 年 7 月本科毕业并获得工学学士学位。

2003 年 9 月免试进入清华大学计算机科学与技术系攻读计算机应用硕士至今。

## 发表的学术论文

[1] Jue Wang, Jinyi Yao, Zengqi Sun, et al. Adaptive User Profile Model and Collaborative Filtering for Personalized News. In APWeb 2006, LNCS 3841, pp. 474 – 485, 2006. APWeb 2006 Best Student Paper Award. (SCI Indexed)

[2] Jinyi Yao, Jue Wang, Zengqi Sun, et al. Ranking Web News via Homepage Visual Layout and Cross-site Voting. In ECIR 2006, LNCS 3936, pp. 131 – 142, 2006. (SCI Indexed)

[3] Jue Wang, Mingjing Li, et al. Augmenting Ranking Function by Label Propagation. Accepted by AIRS 2006, in LNCS series. (SCI Indexed)