

一个基于用户行为的影视节目分类和推荐系统

摘要

信息爆炸使得人们很难快速地从海量的数据中获取自己需要的信息。推荐系统以其精准定位和个性化服务受到广大用户的青睐和研究者的重视。

本论文详细讨论了如何构建一个基于用户行为的影视节目分类和推荐系统，分析了它的各个组成部分。为了解决影视节目描述信息获取困难的问题，系统开发了基于链接长度比例的网络信息抽取工具，首先从因特网上抽取大量影视节目的文本信息。然后对这些文本构建倒排索引，同时建立向量空间，进行文本分析。系统还引入了潜在语义索引，从而部分缓解一义多词的问题。接着采用文本聚类的相关方法将影视节目分类。同时，系统为每个用户维护一个能动态变化的个性向量空间，用以表征用户喜好，并由此进行推荐。最后，针对于影视节目的生命周期短、时间敏感度高的特点运用了时间矩阵的方法，以达到跟踪用户收视习惯的目的，并更有针对性地进行推荐。

关键词：推荐系统，网络信息提取，向量空间，潜在语义索引，聚类，时间矩阵

A PERSONALIZED TV PROGRAM RECOMMENDATION SYSTEM BASED ON VIEWERS' PREFERENCES AND TEXT CATEGORIZATION

ABSTRACT

The plethora of information has made it an exhausting task for consumers to target the data that fits their preferences. And recommendation systems have won the approval of many consumers due to the precision and personalized services. Therefore, it has become a promising research field.

In this paper, we mainly describe the design of a personalized TV program recommendation system based on viewers' preferences and text categorization. We have designed a web extraction tool analyzing the link length ratio, in order to solve the difficulty of getting TV programs' description. With the help of this tool, we can extract large amount of document data from the web. We then construct inverted index of all the documents and a vector space to analyze them. We also introduce Latent Semantic Indexing to lessen the influence of synonymy. Then we categorize the programs using the method of text classification. Meanwhile, the system maintains a dynamic personalized vector for each consumer, based on which recommendation is made. At the end of this paper, we introduce time-matrix to take time factor into consideration and track consumers' habits. With the help of it, the system can then perform better.

Key words: personalized recommendation system, web information extraction, vector space, Latent Semantic Indexing, clustering, time-matrix

目 录

第一章 绪论	1
1.1 推荐系统的研究现状	1
1.1.1 基于内容的推荐算法	1
1.1.2 协同推荐算法	2
1.1.3 混合算法	2
1.2 影视节目推荐系统	2
1.2.1 相关工作	2
1.2.2 我们的系统	3
1.3 本章小结	3
第二章 系统概述	4
2.1 系统简介	4
2.2 本章小结	5
第三章 网络信息提取	6
3.1 相关工作	6
3.2 基于链接长度比例的算法	7
3.2.1 分块标签	7
3.2.2 三个观察结果	7
3.2.3 算法详解	7
3.2.4 其他机制	9
3.2.5 引入链接长度比例的原因	9
3.2.6 算法复杂度	9
3.3 本章小结	9
第四章 倒排索引和词频统计	10
4.1 文档集	10
4.2 中文分词	10
4.3 倒排索引	10
4.4 词频统计	11
4.5 向量空间	12
4.6 本章小结	13
第五章 潜在语义索引	14
5.1 矩阵奇异值分解	14
5.2 矩阵低秩逼近	15
5.3 LSI 的应用	15
5.4 本章小结	15
第六章 文本分类	16
6.1 基于机器学习的方法	16
6.1.1 监督学习分类	16

6.1.2 无监督学习分类-----	16
6.1.3 算法简介-----	17
6.1.4 affinity propagation 算法-----	18
6.1.5 本系统的选择-----	19
6.2 本章小结-----	19
第七章 推荐算法-----	20
7.1 用户的个性向量空间-----	20
7.2 衰减公式-----	20
7.3 推荐方式-----	20
7.4 用户行为挖掘-----	20
7.5 时间矩阵-----	22
7.6 本章小结-----	24
第八章 系统测试-----	25
8.1 网络信息抽取的测试-----	25
8.2 聚类的评价-----	28
8.2.1 聚类测试一-----	29
8.2.2 聚类测试二-----	29
8.3 推荐部分的评价-----	30
8.3.1 推荐测试一-----	30
8.3.2 推荐测试二-----	30
8.3.3 数据分析-----	30
8.4 系统缺点-----	31
8.5 本章小结-----	31
第九章 结论-----	32
参考文献-----	33
谢辞-----	35

第一章 绪论

自从20世纪中期,推荐系统已经成为一个非常重要和活跃的研究领域。由于推荐系统涉及的学术领域非常广,包括信息检索、产品分类、行为建模、实时推荐,并且它能够进行个性化推荐,所以对于它的研究热情经久不衰。推荐系统的主要职能有:跟踪用户行为、筛选合适的内容。这样就免去了用户在海量数据中盲目搜索的负担,并直接提供用户最有可能偏爱的内容。如今,推荐系统已经深入商业营销的方方面面,主流的应用如Amazon.com上推荐图书、CD和其他相关产品;优酷视频网站上根据观看历史推荐类似的视频、音频;淘宝上相似产品的推荐等等。

可以说,推荐系统已经渗透入人们生活的方方面面,对人们日常生活产生了潜移默化的影响。人们在浏览网页、收看节目、网络购物的时候,不可避免地会受到推荐系统的导引。

推荐系统在数字电视上的应用也非常引人注目。随着现代通讯技术的飞速发展以及数字电视机顶盒的逐步普及,数字电视已经成为广大家庭获取信息的主要途径。技术的变革使得机顶盒能够接收和存储大量的影视节目,但这同时也让用户难以快速准确地从中选取适合自己的节目。推荐系统的介入,能赋予数字电视更强的竞争力,从而取得商业上的成功。

尽管推荐系统已经取得了很大的成果,并已为广大用户所接受甚至依赖,但研究人员仍对新一代的推荐系统提出了更高的要求。人们希望能够有更高的准确率,更广的覆盖率。为了达成这些目标,我们必须提出新的方法,包括如何获取更多的产品信息,如何为用户行为建立更加灵活的模型等等。

1.1 推荐系统的研究现状

推荐系统按照推荐方法一般可以分为三大类^[1]:

- (1) 基于内容的推荐算法(Content-based Recommendation);
- (2) 协同推荐算法(Collaborative Recommendation);
- (3) 混合算法(Hybrid Approaches)。

1.1.1 基于内容的推荐算法

基于内容的推荐算法会通过对用户过去的喜好记录,推荐相似的产品。这种推荐方法来源于信息检索领域,比较适用于推荐那些有文字标签分类的商品,比如书籍、网页、新闻等等。

一般情况下,系统会记录用户过去的行为历史,从中抽取他喜好的商品的关键词,建立一个针对于该用户的个性档案。档案中包括用户偏爱的关键词及其权重等等。在进行推荐时从候选商品中取出最匹配用户档案的前几位进行推荐。

基于内容的推荐系统的优点是简单、有效,其缺点是过分细化用户的特征。系统只能推荐和用户过去的档案相匹配的商品,但无法推荐用户以前从未接触过的领域,这一点对于商业拓展业务是十分不利的。因此,一些基于内容的推荐系统不仅排除与用户的历史记录极不相似的商品,还排除那些太相似的,因为这些商品就算不推荐,用户也会去关注。总的来说,基于内容的推荐算法一般会推荐一些不同种类的候选者,而不是局限在一个相似的种类中。

基于内容的推荐系统还有一个显著的问题,那就是对于一个新用户,没有很多历史记录,系统并不了解他的喜好,因此,一开始,系统无法准确地进行推荐。当然,在经过一段时间的历史累积,接着就能够较为准确地进行推荐了。

1.1.2 协同推荐算法

协同推荐算法与基于内容的推荐算法不同,协同推荐算法试图根据其他用户的行为历史来进行推荐。为此,系统通过分析用户兴趣,在用户群中找到该用户的相似用户,综合这些相似用户对某一信息的评价,形成系统对该指定用户对此信息的喜好程度预测。

与传统基于内容的推荐系统相比,协同推荐算法有如下优点^[1]:

- (1) 能够处理那些难以用文本内容来描述的信息。如艺术品、音乐等等。
- (2) 能够同时考虑一些难以表达的复杂的概念。如商品质量、品位等等。
- (3) 能够推荐一些用户以前未接触过的商品,从而拓展业务,提高销量。

协同推荐系统也存在着不足^[1]:

- (1) 对于新用户无法准确定位,这和基于内容的推荐系统一样。
- (2) 对于新商品无法推荐。因为新商品刚上线,还没有被很多用户评价过,所以协同推荐系统无法进行推荐。
- (3) 协同推荐系统的推荐准确度依赖于一个很大的用户群,对于一些受众群体较小的商品,推荐的力度就不够了。另外,用户对于商品的评价非常稀疏,这样基于用户的评价所得到的用户间的相似度可能不准确。
- (4) 对于一些品位较独特的用户,他们的数量占全体用户的比例并不高,可能并没有与之相近的其他用户,这样就会导致推荐准确率的下降。

1.1.3 混合算法

混合算法将基于内容的推荐算法和协同推荐算法有机地结合在一起,以避免二者单独使用所带来的缺陷。有多种方式来合并两种算法^[1]:

- (1) 分别实现基于内容的推荐算法和协同推荐算法,将它们的推荐结果最后合并后呈现。
- (2) 将一些基于内容的推荐算法中的特性融合到协同算法中。
- (3) 将一些协同推荐算法的特性加入到基于内容的推荐算法中。
- (4) 构建一个统一的系统同时包含两者的特性。

1.2 影视节目推荐系统

对于影视节目的推荐,主体方式和其他推荐系统没有很大的区别。但是,影视节目推荐系统有几个与众不同的地方:

- (1) 影视节目的描述信息获取困难;
- (2) 用户没有明显的反馈;
- (3) 影视节目的生命周期短,时间敏感度高。

所以,由于这些特殊性,针对于影视节目的推荐系统也成为了研究领域的热门。

1.2.1 相关工作

Srinivas Gutta et al.讨论了如何通过用户的喜好来推荐电视节目^[2]。他们的系统能够让用户搜索和浏览电视节目数据库,致力于成为一个“smart electronic program guide (smart EPG)”。系统维护了用户的历史档案,并按照用户个人信息计算出推荐结果。

James Davidson et al.讨论了在 YouTube 视频网站上的推荐系统^[3]。该系统主要针对于注

册用户的推荐。系统致力于向用户推荐高质量的、符合他们兴趣爱好的视频。该系统面临的主要挑战是 YouTube 上的视频大都由网友上传，很多情况下没有类别标签或者描述信息质量不高，同时，大多数视频都很短，小于 10 分钟，导致观看用户的交互短暂而混乱，用户偏好难以捕捉。而对于一些购物网站，如 Amazon，买与不买是最重要的信息，用户的意图就很容易捕捉。另一个挑战是大多数视频的生命周期很短，所以推荐系统必须跟上节奏。系统的主要思想是建立视频之间的相关联系图，推荐时取最合适的 N 个，而不是预测用户的行为。

SeungGwan Lee et al. 讨论了在云计算环境中的个性化数字电视节目推荐系统^[4]。系统记录用户的观看历史推断出喜好模式，并利用计算资源进行高效的推荐。作者将云计算很好地融入推荐系统，为推荐系统大规模付诸应用提出了非常好的方案。

Zhiwen Yu et al. 讨论了如何将多用户的历史记录合并从而进行推荐^[5]。该系统设计初衷是针对于家庭或学生宿舍这样的小型群体。文中设计了一个用户档案合并算法，该算法基于总距离最小的想法。当得到了一个统一的用户模型后，再使用传统的方式进行推荐。这种推荐策略尤其适用于多个用户共同观看一个电视机的情形。

Claire Laudy et al. 讨论了推荐系统中影视节目的信息整合^[6]。影视节目信息的来源不同，如何将它们整合成一份冗余少，内容全的节目最终信息，对于推荐系统后续的所有开发应用是至关重要的。该文章提出了很好的解决方法。

Kazuki Ikawa et al. 讨论了如何评估一个基于 Electronic Program Guide (EPG) 和用户观看历史日志的电视节目推荐系统^[7]。

1.2.2 我们的系统

很多推荐系统的设计都将精力花在对于用户行为模式的分析上，这本无可厚非。但是对于影视节目的预处理工作却力度不够。国外一些关于推荐系统的论文大都假设已获取了大量影视节目的基本信息。这是由于国外对于电视节目的信息整合、收集工作很完善，一般能找到权威的来源。相比之下，国内的影视市场就稍显滞后，权威信息的来源不足。为了解决这个问题，我们开发了一个能自动从网络上搜集信息并进行分类的推荐系统。

本系统结合前人理论和方法，采用基于内容的推荐算法，并将因特网信息抽取和文本分类知识融入了推荐系统。从因特网上获取信息保证了信息的充裕度和完整性。而文本分类的引入使对于影视节目的分类更具灵活性，免除了人工标注类别、标签所带来的人力浪费和错误。

为了能提供个性化服务，我们也建立了用户的个性模型，针对于不同用户有不同的个性化推荐。同时，我们充分考虑到电视节目的时间敏感性，设计了时间矩阵，这样不仅能够推荐相似的节目，还能够适应用户的收看习惯。

1.3 本章小结

本章主要介绍了推荐系统的发展前沿，并简单阐述了我们的系统特色。

第二章 系统概述



图 2-1 系统结构

2.1 系统简介

本系统主要应用于影视节目推荐。系统结构如图 2-1 所示。

本系统首先致力于对因特网上的影视节目信息做信息搜索和提取，从而把握住节目的主旨，这样就省去了人工标注类别的繁琐，并且由于拥有了大量的描述信息，正确率也有所保证。然而网页上通常包含大量噪声，如广告、图片、视频、无相关链接等等，网页的主体内容通常被这些噪声所包围。如何从这些网页上获取所需信息，而将噪声的影响降到最低在很多领域都有不容忽视的应用价值。本系统使用基于链接长度比例的算法实现了一个简单有效的网页信息抽取工具，为后续的分类分析提供信息，打下基础。互联网上的信息经过网页过滤和抽取之后，就可以作为各影视节目的特征描述文档。

对于中文文档的分析，首要的任务就是分词。本系统采用中科院计算所的 ICTCLAS 工具进行分词，得到一个一个词项。接着我们建立倒排索引，并引入潜在语义索引，同时为每一个节目建立向量。有了向量空间后，我们可以采用文本聚类的方式将影视节目分类。

为了能提供个性化服务，我们也类似地建立了用户的个性向量，来表征用户的偏好。通过寻找与用户的个性向量最相似的影视节目来进行推荐。同时，为了加入电视节目的时间敏感性度量，设计了时间矩阵，这样不仅能够推荐相似的节目，还能够适应用户的收看习惯。

系统的流程如图 2-2 所示。

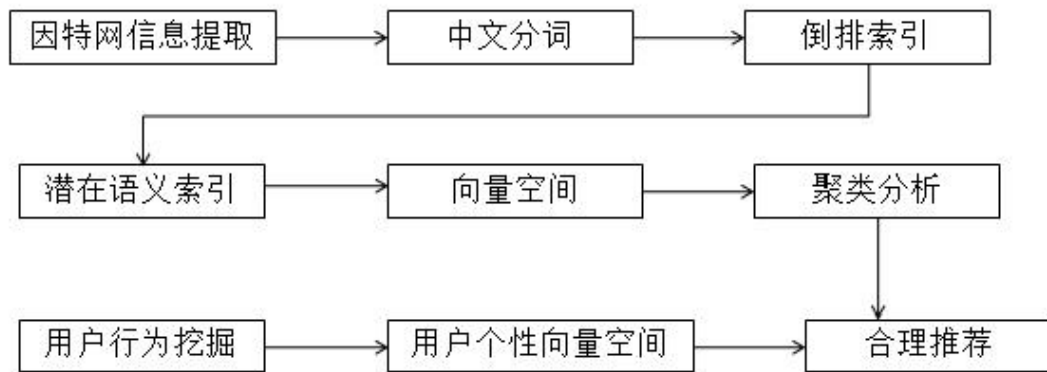


图 2-2 系统流程

2.2 本章小结

本章主要介绍了本系统的大致结构和流程。后续章节将会详细介绍。

第三章 网络信息提取

网络信息提取是指从网页上提取出目标信息^{[8][9][10][11][12][13]}。网页上通常包含大量的噪声，如广告、图片、无用链接等等。而最重要的主体信息常常被这些噪声所包围。如何从其中准确高效地抽取有关信息，同时最大限度降低噪声的影响，是一个非常重要的课题。针对于网页主体信息的自动化抽取在很多学术领域都有不容忽视的应用价值。最常见的应用之一就是在搜索引擎的网页采集器中。

3.1 相关工作

从 20 世纪中期以来，研究人员一直都在研究如何高效准确地抽取网络信息。

主流的方法有三种^[15]：

(1) 人工提取

由程序员通过观察网页界面，分析它的 HTML 源码，发现一些规律和模式，从而编写一个程序进行提取。但是这种方法适用范围小，扩展性较弱。

(2) 半自动提取

这是一种监督学习的方法。在这种方法中，从一些已经人工标注好的网页中推导出规则，进而利用这些规则从其他格式相近的网页中提取信息。

(3) 自动提取

这是一种无监督的自动提取方法。对于任意一个网页，它会自动分析、定位、提取。由于无需人工，所以这种方法可以适用于大范围的网页。

在本系统中，网页都是从搜索引擎返回的结果链接而来，范围很广，无法或难以找到一种普适的规则。于是，自动提取是最佳选择。由于从网络中收集到的影视节目信息将直接影响到分类和推荐，所以它的有效性是整个系统的基础。

自动提取的方法一般有：

(1) 基于视觉(visual cues)^[15]

这种方法使用视觉信息来帮助寻找网页中的主体信息。网页的视觉布局信息可以从网页浏览器中得到。比如，IE 提供了一个接口，输出每个网页元素的坐标。一般人们使用机器学习的方式来构建提取算法，此时，大量的训练数据是必要的。

(2) 树匹配算法(tree alignment)^{[14][15][16]}

这种方法是基于这样一种观察：大多数的商业网站的网页都是使用固定的模板生成的。于是，这种方法致力于寻找出这些隐藏的模板。同时，由于 HTML 是一种嵌套的结构，很容易构建一棵树来帮助分析。通过计算子树之间的匹配度，来发现模板，进而发现主体内容。

在本系统中，由于没有大量的训练集，所以基于视觉的算法在应用上有难度。另外，树匹配算法更加适用于发现重复的数据元组(data record)^[15]。因此，对于本系统中涉及的各式各样、毫无规律的网页集，以上这些方法都没有提供最简洁有效的途径。

3.2 基于链接长度比例的算法

在 3.1 节中介绍的自动提取方式不适用于我们的系统。我们经过多次测试和计算，设计了基于链接长度比例的算法。

该算法基于剔除无用信息，而不是找出有效信息，所以是反其道而行之。

为了能更方便地分析 HTML 文件，首先将网页转化为 Document Object Model(DOM)树。事实上，HTML 的树型结构能够为分析提供很多便捷。树型结构为我们的遍历提供了更大的灵活度，同时，我们还可以方便地删除和修改节点，最后留下包含主体内容的节点^[8]。图 3-1 是 DOM 树的示意图。

HTML 文件中包含普通文本，标签，指向图片、音频、视频、其他页面的链接等等。大多数标签都是成对出现的。每一对都包含开始标签和结束标签。在标签对内部还可以有其他的标签对，这样就形成了嵌套结构。因此，HTML 文件是自然嵌套的。

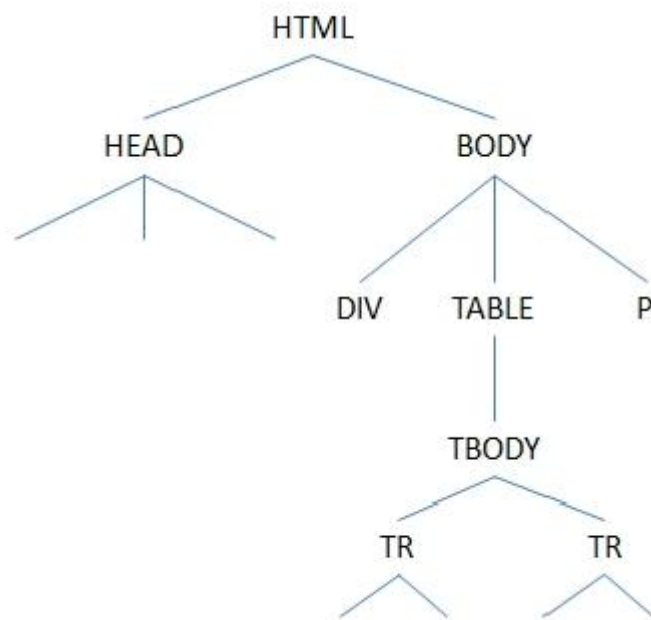


图 3-1 DOM 树范例

3.2.1 分块标签

HTML 语言中有一些特别的标签，如<p>、<div>、<table>、<form>、、等，它们标志了一整块内容。我们定义这些标签为分块标签，它们标志的范围称为块。

3.2.2 三个观察结果

我们观察到了下列现象：

- (1) 绝大多数噪声都是链接，比如广告、滚动条、音频视频、站内导航等。
- (2) 相似或相关的内容一般都在同一块或相近的块中。

(3) 主体内容中普通文本相比链接文本所占的长度比例更大，相反地，在噪声中，链接文本所占的比例更大。

3.2.3 算法详解

3.2.2 中的三个观察结果启发了我们的算法。

由观察(2)和(3)可知，如果在某一块中普通文本相比链接文本所占的长度比例更大，那这一块是主体内容的可能性也更大。于是，很自然地想到统计每一块中的普通文本和链接文本的长度，计算出链接文本长度和普通文本长度的比值 r ，并设定一个阈值 T ，当 $r > T$ 时，判为噪声，否则有可能为主体内容。

为了达到这个目的，我们后序遍历 DOM 树，而不是从上至下依次访问。这就保证了嵌套最内层的块能最先被发现。

伪代码如图 3-1、3-2、3-3 所示。

```
Algorithm WebExtraction(Root,  $\tau$ )
1  Candidates :=  $\phi$ 
2  Traverse(Root, Candidates,  $\tau$ )
3  Output all the texts still in the tree
```

图 3-2 WebExtraction 函数

```
Algorithm Traverse(Node, Candidates,  $\tau$ )
1  for each Child in Node.Children
2  do Traverse(Child, Candidates,  $\tau$ )
3  end for
4  Judge(Node, Candidates,  $\tau$ )
```

图 3-3 Traverse 函数

```
Algorithm Judge(Node, Candidates,  $\tau$ )
1  if Node is not a block tag
2  then add Node into Candidates
3  else if Node is a block tag
4  then if Node has no children
5       then return
6       else if Node has children and children are not in Candidates
7       then return
8       else if Node has children and some children are in Candidates
9       then calculate the ratio  $r$  of Node according to Candidates
10      if  $r > \tau$  then remove Node and all its children from the tree
11      else keep Node
12      remove Node and its children from Candidates
```

图 3-4 Judge 函数

整个算法是一个树后序遍历算法。那三个观察结果在本算法中有所体现。首先初始化 Candidates，它是用来存放待定的节点的，初始为空。函数 Traverse() 执行了后序遍历，从内至外分解嵌套。

Judge()函数负责计算和判断的工作。它的第 1、2 行表示如果该节点不是分块标签，则将它加入 Candidates 待定。第 3 至 12 行处理了分块标签。第 4、5 行表示如果该节点没有子节点，则说明包含噪声的可能性较小，事实上，从观察结果(1)中可知，如果包含噪声，则应该包含链接，链接在 HTML 文件中是以子节点的形式出现的。

第 6、7 行表明，该节点存在子节点，但都不属于 Candidates，这就说明这是一个嵌套结构，即分块节点中嵌套着其他分块节点，而内部的分块节点已经通过了检测，是主体内容，故该节点也应该保留。

第 8 至 12 行处理了最后一种情况，该节点存在子节点，且至少有一个子节点属于 Candidates，于是我们利用观察结果(2)和(3)，统计出该节点内部所有的后代的普通文本长度 t ，以及链接文本长度 a ，计算比例 $r=a/t$ ，如果 r 大于阈值 τ ，则说明这是一个有丰富链接文本的块，它是噪声的可能性很大，于是删除该节点及其所有属于 Candidates 的子节点。否则，保留之。最后要清理 Candidates，将已经处理过的节点从中删除。

3.2.4 其他机制

事实上，在算法的实现上，我们还可以引入其他的一些机制来更好地过滤无用信息。比如引入黑名单。

在 HTML 中，有一些特别的标签，它们完全不可能是主体的文本内容，如图片，滚动文本选取框<marquee>，表单控件<input>等等。后序遍历时一旦遇到这些标签，就直接将其删除。

很多网页都有与用户进行交互的机制，通常是一些文本输入区域，如<textarea>标签。根据观察结果(2)，包含这些标签的块一般不会再包含主体文本信息，从而可以将其删除。这里的块指的是直接包含这些交互标签的最内部的块。

引入黑名单的确可以过滤掉更多的噪声，但使用过度就会误判，对于一些页面设计紊乱的网页，尤其会误删。所以在实际使用中，可以根据需要改变相应的配置。

3.2.5 引入链接长度比例的原因

如今，大多数网页为了提高信息的完整性和对用户的友好性，会在文本中加入额外的链接，指向对于该词汇的解释信息或相关内容网页，而那些链接本身也是文本的一部分。典型的网站有维基百科、百度百科等等。在这些网页的主体文本中充斥着链接，我们采用计算比例的方法，对于这些网页也是适用的，因为在主体文本中链接文本所占的比重仍然是较小的，而对于链接列表而言，链接文本的比重是很大的。

经过多次测试和权衡，我们认为阈值 τ 一般设定在 0.3 至 0.4 为宜。

3.2.6 算法复杂度

对于很复杂的嵌套结构网页，本算法的最差时间复杂度是 $O(N^2)$ ， N 是 DOM 树中的节点数。如果结构较简单，则典型的运行时间为 $O(N)$ 。在大量测试中，算法都能很快地运行出较满意的结果。

本算法较适用于那些有大块文本的网页，大多数网页外链接和广告等噪声都能被去除。

3.3 本章小结

本章详细介绍了本系统所采用的基于链接长度比例的网络信息提取算法的原理和方法，以及所要达到的目的。

第四章 倒排索引和词频统计

当我们从因特网上得到了关于影视节目的网页，再经过网页信息提取工具的过滤，就得到了相对较好的影视节目的描述信息。接下来的工作就是处理这些文档。本章主要介绍了一些目前常用的处理方法。我们借鉴和使用了这些方法的基本思想和设计原理，并融入本系统中来处理和分析文档。

4.1 文档集

首先，我们建立文档集。将某一个影视节目名字输入搜索引擎之后得到了许多相关链接，每个链接指向了一个网页，每个网页中可以提取出一篇主体文档，而这些文档都是描述同一个影视节目的。所以我们将关于同一个节目的文档放入同一个文档集中，并以节目的名称命名。

我们并不把关于同一个节目的文档合并，因为从因特网上抓取的信息可能存在重复，如果简单合并文件，就意味着在同一篇文档中会有不必要的重复和冗余，这对于我们后续的分析工作是不利的。所以我们将它们组成文档集，仍然以单独的文档保存。

4.2 中文分词

在本系统中，我们针对的是中文影视节目，描述信息文本也自然是中文的。为了进行分析工作，首先要对文本预处理。

为了进行分析，我们要将每篇文档转换成一个个词条的列表，这个过程通常称为词条化^[17]。然后进行语言学预处理，产生归一化的词条来作为词项。这一步主要用于西方语言，汉语和许多其它西方语言不同，它没有词形变化，不需要词干还原、词形归并^[17]，故我们省去此步。

中文文本分析的主要工作是分词。本系统采用著名的中科院计算所的 ICTCLAS 工具进行中文分词。

分词结束之后，每篇文档就可以看成是由一个一个词项组成的，这就是词袋模型^[17]。虽然在我们的分析中不考虑词项在文档中出现的位置顺序，但直观上词袋模型仍能从一定程度上反映文档的特征。

4.3 倒排索引

在中文分词之后，我们建立倒排索引^{[17][18]}。倒排索引的基本思想见图 4-1。

图 4-1 中间称为词项词典。每个词项都有一个记录出现该词项的所有文档的列表，该表中的每个元素记录的是词项在某文档中的出现信息，如出现频率。

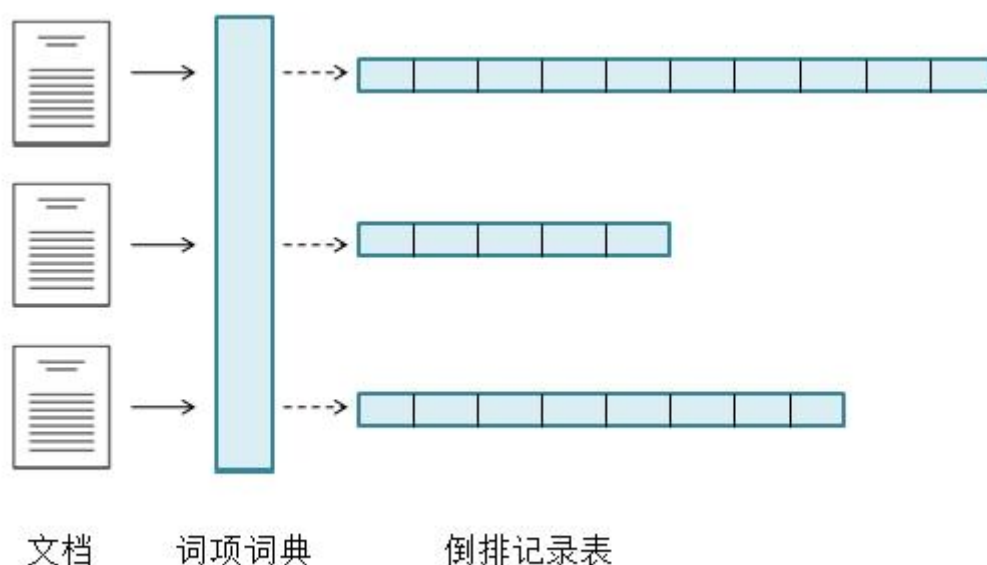


图 4-1 倒排索引

使用倒排索引的原因首先是为了节省空间^[17]。如果建立和存储一个词项-文档矩阵，空间开销将大得惊人。不难发现，这个庞大的矩阵具有高度的稀疏性，即大部分元素是 0，所以只记录非零位置更可行。倒排索引就是这么做的。

其次，倒排索引将文档集中出现某一词项的所有文档都记录在了一起，方便我们进行数据统计。

4.4 词频统计

在已经得到的词袋模型的基础上，我们要进行词项频率统计，并记录进倒排索引。

对于每个词项，首先记录它在文档中出现的次数，称为词项频率，记为 $tf^{[19]}$ 。

考虑到文档中所有词项的重要性并不一样，我们需要引入其他计算方式。比如说信息检索领域的停用词，像“的”、“地”这样的对于文档相似度计算没有或很少有贡献的词项，不需要对它们进行索引。

这里我们不引入停用词，因为停用词表的大小对于最终的相似度计算有一定影响。停用词表过大，虽然减小了索引的大小，但可能会丢失一些重要信息。停用词表过小，效果又不明显。在信息检索系统不断发展的历程中，有从大停用词表到小停用词表，最后不用停用词表的趋势^[17]。

出现某词项的所有文档的数目称为文档频率，记为 $df^{[19]}$ 。我们给文档频率较高的词项赋予较低的权重。 df 本身往往很大，所以通常需要将它映射到一个较小的取值范围中去。令所有文档的数目为 N ，词项的逆文档频率 idf 定义如下^[19]：

$$idf = \log \frac{N}{df} \quad (4-1)$$

对于每篇文档中的每个词项，将其 tf 和 idf 组合在一起形成最终的权重 $tf-idf^{[19]}$ 。

$$tf-idf = tf \times idf \quad (4-2)$$

tf-idf 按照如下的方式对文档 d 中的词项 t 赋予权重^[17]:

- (1) 当 t 只在少数几篇文档中多次出现时, 权重取值最大。因为此时能够对这些文档提供最强的区分力。
- (2) 当 t 在一篇文档中出现次数很少, 或者在很多文档中出现, 权重取值次之。因为此时对最后的相关度计算作用不大。
- (3) 如果 t 在所有文档中都出现, 那么权重取值最小。这就类似于引入了停用词表。

这样, 就可以把文档看成是一个向量, 其中的每一个分量都对应词典中的一个词项, 分量值采用 tf-idf。当某词项在文档中没有出现时, 其对应的分量值为 0。这种向量形式对于分类十分重要。可以想见, 文档向量是非常稀疏的, 在实际实现中, 如果空间要求很高, 可以改用链表结构, 只存储非零值。

除了原始的 tf-idf 计算权重的方法外, 还有很多其他方法。我们考虑到即使一个词项在文档中出现了 10 次, 它所携带信息的重要性也不可能是只出现 1 次的词项的 10 倍。一个常用的修改方法是采用原始词项频率的对数函数^[17]。此时, 计算方法如下:

$$wf = \begin{cases} 1 + \log tf, & tf > 0 \\ 0, & \text{其他} \end{cases} \quad (4-3)$$

其他的方法还有: 基于最大值的 tf 归一化^[17]等等。

4.5 向量空间

一系列文档在同一向量空间中的表示称为向量空间模型^[17]。一篇文档所对应的向量中的每个分量代表了词项在该文档中的相对重要性, 比如 tf-idf。在同一空间中, 我们就能够计算两篇文档对应向量之间的余弦相似度:

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| |\vec{V}(d_2)|} \quad (4-4)$$

公式(4-4)中除以分母的效果是将向量长度进行归一化。

图 4-2 是余弦相似度示意图。

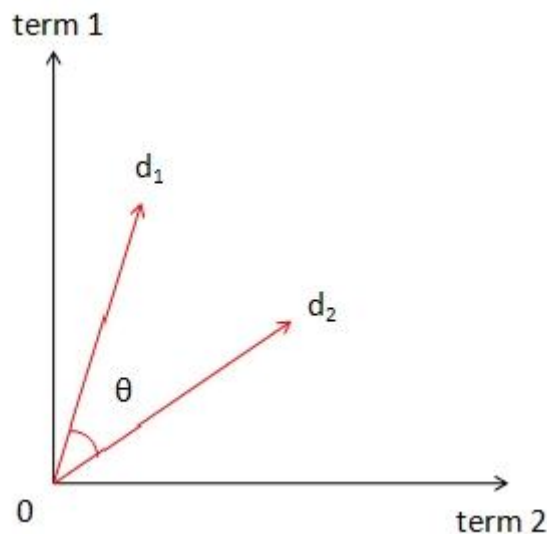


图 4-2 余弦相似度示意图

如 4.1 节所述, 关于同一个影视节目有多篇描述文档, 每一篇文档都有一个向量, 为了表征影视节目的特征, 我们将关于它的文档向量合并起来, 同时进行归一化操作。这样我们对于每一个节目就都得到了一个描述它特征的向量, 接着就可以用向量空间的方法进行讨论了。今后本文中不再区分文档向量空间和影视节目向量空间。

4.6 本章小结

本章主要描述了本系统在获取文本后的处理工作, 涉及了中文分词、倒排索引、词频统计等, 并将这些内容付诸实现为后续处理做准备。

第五章 潜在语义索引

将文档和节目都表示成向量空间有许多优点，如：可以使用余弦相似度进行计算，可以推广到聚类、分类领域。但是，向量空间表示法没有能力处理自然语言中两个经典问题：一义多词和一词多义问题。一义多词是指不同的词具有相同的含义，而一词多义指的是某个词具有多个含义^[17]。

考虑建立同义词词典，但是人工构建同义词词典代价很大。主流的有两种取代方法，一种是使用词项共现信息，另一种是采用浅层语法分析器来分析文本得到词汇间的语法关系或语法依存性^[17]。一般来说，采用词项共现信息更具鲁棒性，但是采用语法关系可能会更精确。本系统中使用了词项共现信息。

在原来的向量空间中，仅仅关注了单个词项的出现频率，但词项的共现也是重要的信息。这基于一个事实：文档中两个或者更多的词项的出现不是偶然的事件。潜在语义索引(Latent Semantic Indexing, LSI)就是一种探查内在语义联系的方法^{[17][23][24]}。潜在语义索引就是把同现词条映射到同一维空间上，而非同现词项则被映射到不同的空间上。同现的词项被认为是语义相关的。

潜在语义空间与我们原来的空间相比，维数要小得多，因为很多词项被投影到同一维上，因此，潜在语义索引的方法其实是一种降维方法。维数消减过程就是将高维空间中的对象映射到较低维的空间上的过程。从空间降维角度讲，潜在语义索引方法具备了某种最优特性，即它把原对象投影到方差最大的轴上，即数据在我们选取的映射轴上的投影具备最大的可分性。从这个意义上讲，它非常近似于主成分分析(Principal Component Analysis, PCA)，但 PCA 的方法仅仅适用于方阵，而潜在语义索引的操作对象可以是任何矩阵^[24]。

我们可以把之前得到的向量空间转换为词项-文档矩阵，即由 M 个词项和 N 篇文档组成的一个 $M \times N$ 的权重矩阵 C ，矩阵的每行代表一个词项，每列代表一篇文档。接着就可以应用矩阵理论了。

5.1 矩阵奇异值分解

我们首先叙述一个定理：

令 r 是 $M \times N$ 矩阵 C 的秩，那么 C 存在如下形式的奇异值分解(Singular Value Decomposition, SVD)^[17]：

$$C = USV^T \quad (5-1)$$

$$\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \cdot \begin{bmatrix} * & & & & \\ & * & & & \\ & & * & & \\ & & & * & \\ & & & & * \end{bmatrix} \cdot \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$$

$$C = U \quad S \quad V^T$$

图 5-1 矩阵奇异值分解

其中， U 的列之间相互正交， V 的列之间也相互正交， S 是对角矩阵，对角线上的元素

都是 C 的奇异值(singular value)^[23]。

$$CC^T = USV^T VS^T U^T = USS^T U^T \quad (5-2)$$

公式(5-2)左边是一个实对称方阵，其每行每列都对应 M 个词项中的一个。矩阵中的第 i 行第 j 列的元素实际上是第 i 个词项与第 j 个词项基于文档共现次数的一个重合度计算指标^[24]。

$$C^T C = VS^T U^T USV^T = VS^T SV^T \quad (5-3)$$

公式(5-3)左边是一个实对称矩阵，其每行每列都对应 N 篇文档中的一个。矩阵中的第 i 行第 j 列的元素代表了第 i 篇文档与第 j 篇文档的相似度^[24]。

5.2 矩阵低秩逼近

SVD 可以用于解决矩阵低秩逼近问题^[17]。需要进行如下 3 步操作：

- (1) 给定 C ，按照公式(5-1)构造 SVD 分解： $C = USV^T$ 。
- (2) 把 S 对角线上 $r-k$ 个最小奇异值置为 0，从而得到 S_k 。
- (3) 计算 $C_k = US_k V^T$ 作为 C 的逼近。

由于 S_k 至多包含 k 个非零元素，所以 C_k 的秩不高于 k 。我们知道，小特征值对于矩阵的影响也小，因此，将这些小特征值替换成 0 不会对最后的乘积有实质性影响。Eckart 和 Young 证明了上述过程产生了一个秩为 k 的矩阵，它的 F 范数误差最小^[17]。

利用新的低秩矩阵来重新计算公式(5-3)就将词项共现的性质同时计算进去了。

Dumais 的实验表明对于适合的 k 值，LSI 能部分解决一义多词的问题^[17]。

5.3 LSI 的应用

矩阵 $S_k V^T$ 是一个 $k \times N$ 的矩阵， k 是使用 LSI 之后 S 的秩， N 是文档总数。矩阵中每一列都是对应文档在降维空间中的新坐标。由此，将该矩阵拆开可以还原成向量空间，即矩阵中每一列就是对应的向量。

SVD 在信息检索领域的应用最初由 Deerwester 提出来的^[23]。LSI 把原文档向量在一个新的降维空间中表示出来，新的降维空间实际上是原有空间的各维的线性组合，可以看成是一种软聚类的方法。除了揭示内在语义联系，LSI 还体现出了紧凑的优点，一般可以降到 150 维左右，不再是稀疏的^{[17][24]}。

本系统采用了 LSI 来降维，并缓解了一义多词的问题。

5.4 本章小结

本章重点介绍了 LSI 的原理和在信息检索领域的应用。本系统引入了 LSI 处理方式。

第六章 文本分类

得到了文本之间的相似度之后,我们需要将文本分类。分类是指将给定对象归入一个或者多个类别的过程。

分类方法主要有^{[17][21]}:

- (1) 人工分类。最古老的方法,但是一旦规模化则开销很大。
- (2) 人工编写规则。具有很好的扩展性,但是创建和长时间维护需要很高的人力成本。
- (3) 基于机器学习的方法。是目前的主流方法,研究的热门方向。

6.1 基于机器学习的方法

基于机器学习的方法主要分为两类^[17]:

- (1) 监督学习分类。
- (2) 无监督学习分类。

6.1.1 监督学习分类

监督学习分类主要有 Rocchio 方法, kNN 方法^[21], 专家网络 (Expert Network) 方法^[20] 等等。这类方法都需要事先有一个足够大的已经标注了的训练集,训练完成之后,就能对其他新的文档进行分类了。

6.1.2 无监督学习分类

无监督学习意味着不存在对文档进行类别标注的人类专家。聚类是无监督学习的最典型的形式^[17]。在聚类中,数据的分布和结构决定了最后的类别归属。

聚类算法的一个关键输入是距离计算方法。一般采用欧氏距离。

扁平聚类和层次聚类是两种典型的聚类方法^{[17][22]}。扁平聚类算法会给出一系列扁平结构的簇,它们之间没有任何显式的结构来表明彼此的关联性。而层次聚类算法会产生层次性的聚类结果。

聚类还可以分为硬聚类和软聚类^[17]。硬聚类计算的是一个硬分配过程,即每篇文档仅仅属于一个簇。而软聚类中,一篇文档的分配结果是在所有簇上的一个分布。在软分配结果中,一篇文档可能对多个簇都具有隶属度。

扁平聚类具有概念简单、速度快的优点,但是同时又有许多缺点,如:需要预先定义簇的数目,聚类结果具有不确定性。与之不同的是,层次聚类会输出一个具有层次结构的簇集合,因此能够比扁平聚类输出的无结构簇集合提供更丰富的信息。层次聚类不需要事先指定簇的数目,并且大部分层次聚类算法都是确定性算法。当然,层次聚类在获得这些好处的同时,付出的代价就是效率降低。最普遍的层次聚类算法的时间复杂度至少是文档数目的平方级,而扁平聚类 K-均值算法和 EM 算法的时间复杂度都是线性的^[17]。

层次聚类可以是自顶向下或自底向上的一个过程。自底向上的算法一开始将每篇文档都看成是一个簇,然后不断地对簇进行两两合并,直到所有文档都聚成一个类为止。自底向上的聚类算法也因此被称为凝聚式层次聚类 (Hierarchical Agglomerative Clustering, HAC)^[17]。而自顶向下的聚类算法则首先将所有文档看成一个簇,然后不断地利用某种方法对

簇进行分裂直到每篇文档都成为一个簇为止。在信息检索的领域，HAC 算法的应用比自顶向下的算法更普遍。

HAC 的聚类结果一般可以采用图 6-1 所示的树状图来表示^[17]。其中，每篇文档一开始都被看成一个单独的簇，聚类过程中的每次合并都表示成一条水平线，而 y 轴对应的是合并时两个簇的相似度，这个相似度也被称为合并后的簇的结合相似度。

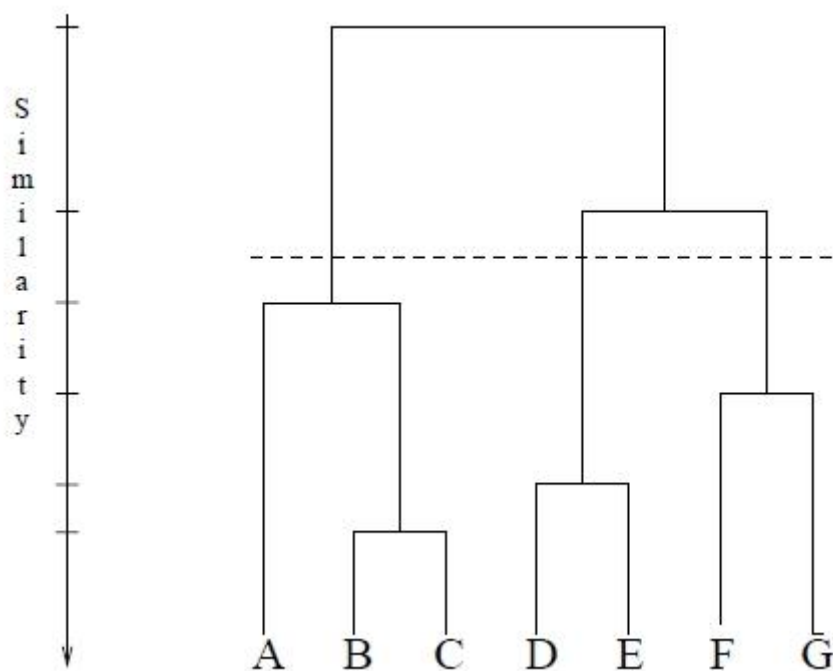


图 6-1 单连接聚类得到的树状图^[22]

通过从底层节点一直上升到顶层节点，树状图能够记录整个合并历史，并能够重构整个合并过程。

HAC 的一个基本假设是，HAC 聚类中的合并操作是单调的，即，如果 HAC 中的合并结果的结合相似度依次是 $s_1, s_2, \dots, s_k$ ，那么有 $s_1 \geq s_2 \geq \dots \geq s_k$ ，而非单调的层次聚类中则至少包含一个颠倒现象，如质心聚类^[17]。

层次聚类不需要事先指定簇的数目。但是在应用中，我们希望能得到像扁平聚类那样的互不相交的多个簇。事实上，我们只需要在层次结构的某一点上进行截断，停止聚类，就能够得到多个簇。截断点的确定方法有很多种^[17]：

- (1) 在某个事先给定的相似度水平上进行截断。
- (2) 当两个连续的聚类结果的结合相似度之差最大时进行截断。
- (3) 事先指定结果簇的数目 K ，在产生 K 个簇时截断。

6.1.3 算法简介

简单的朴素 HAC 算法伪代码如图 6-2 所示。

```

SimpleHAC( $d_1, \dots, d_N$ )
1  for n from 1 to N
2    do for i from 1 to N
3      do  $C[n][i] := \text{sim}(d_n, d_i)$ 
4       $I[n] := 1$ 
5   $A := \phi$ ; (collects clustering as a sequence of merges)
6  for k from 1 to N-1
7    do  $\langle i, m \rangle := \arg \max_{\langle i, m \rangle: i \neq m \wedge I[i]=1 \wedge I[m]=1} C[i][m]$ 
8    A. APPEND( $\langle i, m \rangle$ ) (store merge)
9    for j from 1 to N
10     do  $C[i][j] := \text{sim}(i, m, j)$ 
11      $C[j][i] := \text{sim}(i, m, j)$ 
12      $I[m] := 0$  (deactivate cluster)
13 return A

```

图 6-2 SimpleHAC 伪代码^[17]

该算法中，首先计算一个 $N \times N$ 的相似度矩阵 C ，然后算法执行 $N-1$ 步，每一步都将最近的两个簇进行合并。每次迭代中，合并最相近的两个簇 i 和 m ，合并后得到的簇仍然用 i 来标识，此时簇 i 在 C 中对应的行和列要进行更新。每次聚类的结果不断添加到 A 中。数组 I 表示仍然可以被合并的簇(此时对应的值为 1)。 $\text{sim}(i, m, j)$ 计算的是簇 i 及 m 的合并簇与簇 j 的相似度。

采用不同的相似度计算方法可以将 HAC 分为多个类别。常用的有单连接和全连接聚类算法、组平均和质心聚类算法^{[17][21]}。

在单连接聚类中，两个簇之间的相似度定义为两个最相似的成员之间的相似度。这种单连接的合并准则是局部的，即它仅仅关注两个簇互相邻近的区域，而不考虑簇中更远的区域和簇的整体结构。

在全连接聚类中，两个簇之间的相似度定义为两个最不相似的成员之间的相似度，这也相当于选择两个簇进行聚类，使得合并结果具有最短直径。全连接聚类准则是非局部的，聚类结果中的整体结构信息会影响合并的结果。这种聚类实际上相当于优先考虑具有较短直径的紧凑簇，而不是具有长直径的松散簇，当然这种做法可能会对离群点较为敏感，比如某个远离中心的文档会显著增加候选簇的直径从而完全改变最后的聚类结果。

单连接和全连接聚类方法将簇质量的计算过程简化成两个文档的单一相似度计算，其中在单连接方法中计算的是两篇最相似的文档之间的相似度，而在全连接方法中计算的是两篇最不相似的文档之间的相似度。

与前两种算法不同的是，组平均凝聚式聚类通过计算所有文档之间的相似度来对簇的质量进行计算。而质心聚类通过两个簇的质心相似度来定义两个簇之间的相似度。组平均凝聚式聚类和质心聚类的区别在于，组平均聚类在计算平均相似度的时候考虑了所有文档之间的相似度，而质心聚类中仅仅考虑来自同簇的文档之间的相似度。

6.1.4 affinity propagation 算法

近年来，仍有许多学者在对聚类算法进行深入研究。Brendan J. Frey 和 Delbert Dueck 提出了 affinity propagation(AP)算法^[25]。该算法同时将所有数据点作为簇中心或种子的候选者。该算法是把每个数据点作为网络中的一个节点，通过递归地在网络中传递数值信息，直

到簇中心和对应的簇中的元素都被决定。在该算法中，数值消息是通过一些简单的公式计算出来的。算法定义了一个能量函数，而那些计算公式正是基于寻找该能量函数的极小点。在整个算法的执行过程中，数值消息的大小反映了一个数据点和另一个数据点之间的亲和度。当节点之间互相传递的数值消息的前后改变量低于一个预设的阈值，或者节点成为种子的递归结果不再改变时，算法就可以结束了。此时，我们不仅找到了簇中心，还同时得到了每个节点所属的聚类。

6.1.5 本系统的选择

经过学习和比较多种文本分类方法之后，在本系统中我们实现了单连接层次聚类和 AP 算法，偏向于采用单连接聚类，因为该算法产生的层次性结构在推荐系统中有很大的用途，我们在第七章将会看到。

在推荐系统中，正确有效地对所涉及的商品进行合理分类，能够为推荐的准确度提供保障。推荐在同一分类中的产品，往往都会符合用户的心意。

6.2 本章小结

本章详细介绍了文本分类，并细述了其中的方法。本系统采用了 HAC 算法进行文本分类。

第七章 推荐算法

在将所有影视节目转化为向量，并完成相应分类后，我们就能够对用户进行基于观看历史的推荐了。

7.1 用户的个性向量空间

本系统采用基于内容的推荐方法。

我们通过从网络上获取的影视节目描述信息建立了能表征节目特征的向量空间，这就类似于得到了一系列分类标签。我们通过聚类算法将影视节目分类，得到了节目之间的相互联系。

我们希望通过用户的历史观看记录或者他对于节目的反馈信息来判断用户的偏好，从而确定用户最有可能喜爱哪一个聚类中的电视节目，并将该聚类中的节目推荐出去。

为了记录用户的观看历史，我们需要设计一种能够动态变化的统计方式。很自然地，我们想到了为每个用户建立一个个性向量。

如果使用了LSI，则每个影视节目的向量都投影到了低维空间，我们可以将用户的个性向量也设置成相同的维数。由于我们使用了余弦相似度，所以在高维空间中向量的方向极为重要，它表示了向量所代表的影视节目或者用户个性的特征。

7.2 衰减公式

在向量空间中，某个向量 α 加上另一个向量 β 就意味着 α 向量的方向将向 β 向量的方向偏转。我们利用这一点，当用户观看了某个影视节目很长时间或者他对其反馈良好，我们就能判断该节目符合该用户的喜好，于是将该影视节目的向量叠加到用户的个性向量中。由此，用户的个性向量就向该影视节目的向量方向偏转，也就是说，经过长时间的积累，用户的个性向量就能够表征用户的偏好了。

我们知道，人的喜好是可能随着时间而改变的，所以，对于用户的个性向量空间要能随着时间慢慢衰减。我们可以设定一个衰减系数 $0 < d < 1$ ，按照公式(7-1)在每次叠加时都进行适当的衰减。

$$\overrightarrow{V_{\text{new}}} = d\overrightarrow{V_{\text{old}}} + (1 - d)\overrightarrow{V_{\text{program}}} \quad (7-1)$$

公式(7-1)中 $\overrightarrow{V_{\text{old}}}$ 是用户过去的个性向量， $\overrightarrow{V_{\text{program}}}$ 是该用户偏爱的影视节目的向量， $\overrightarrow{V_{\text{new}}}$ 是最终用户新的个性向量。通过衰减系数 d 的控制，我们可以调节用户的个性向量对于喜好变化的敏感度。

可以想象，用户的个性向量在高位空间中随着时间和他喜好的变化而不断旋转。所以，个性向量能够记录用户的观看历史，同时对用户偏好的变化也能够及时反应。

7.3 推荐方式

在得到用户的个性向量之后，我们就可以进行推荐了。

在聚类时，我们同时得到了一个树状图，它记录了整个合并历史。如图(6-1)所示，沿竖直方向从下到上，簇的结合相似度逐渐降低。为了快速定位用户最为偏爱的影视节目类型，我们要充分利用这个树状图。容易证明，在聚类过程中构造的这棵树除了叶节点之外所有节点都有且仅有两个子节点。事实上，凝聚式聚类总是对簇进行两两合并，所以某个节点若存在子节点，则必然有且仅有两个。

对于某一个特定的用户的个性向量，我们从上至下沿着树状图寻找最适合用户的影视节目。在任一内部节点，分别计算出用户的个性向量和其两个子节点的余弦相似度，取其大者，沿着该子节点继续向下寻找。

伪代码如图 7-1 所示。

```
SimpleNavigate(V, Node)
1  if Node has children
2  then sim1=sim(V, Node.Child1)
3      sim2=sim(V, Node.Child2)
4      if sim1>sim2
5          then SimpleNavigate(V, Node.Child1)
6      else SimpleNavigate(V, Node.Child2)
7  else return Node
```

图 7-1 SimpleNavigate 函数

图 7-1 是最简单的寻找最相近节点的伪代码。其中， V 是用户的个性向量， $Node$ 是树状图中的节点。初始调用 $SimpleNavigate(V, Tree.root)$ 。算法复杂度是 $O(M\log(N))$ ，其中 M 是向量的维数， N 是影视节目个数。以这种方法找到的是一个最适合该用户的影视节目。

如前所述，基于内容的推荐算法有一个缺点是过分细化用户的特征，为了减轻该缺陷带来的不良影响，我们可以稍稍改进 SimpleNavigate 算法，使之能够分叉，得到 Navigate 算法，如图 7-2。主要区别在第 4、5 行，如果求出的相对于两个子节点的相似度之差的绝对值小于某一个阈值 τ ，则两边都要遍历。这样就既扩大了可推荐的范围，又确保了推荐的准确度。

```
Navigate(V, Node,  $\tau$ )
1  if Node has children
2  then sim1=sim(V, Node.Child1)
3      sim2=sim(V, Node.Child2)
4      if  $|sim1-sim2| < \tau$ 
5          then Navigate(V, Node.Child1,  $\tau$ ) and Navigate(V, Node.Child2,  $\tau$ )
6      else if sim1>sim2
7          then Navigate(V, Node.Child1,  $\tau$ )
8      else Navigate(V, Node.Child2,  $\tau$ )
9  else return Node
```

图 7-2 Navigate 函数

7.4 用户行为挖掘

由于影视节目的特殊性，反馈信息的获取有一定难度，一般我们需要从用户的换台信息中得到用户对影视节目喜爱与否的判断。

最简单的方式是统计用户在某一个频道逗留的时间占该频道正在播放的节目的总时间的长度比例。注意时间跨度可能是多个节目。为了实现这个方式，只需要有一份详细的节目时间表，我们就能全部统计出来。

另一个问题是，电视节目中经常会插播广告，此时，有些用户会在频道间来回切换。为了应对这种情形，我们可以存储一个缓存，存储若干个用户换台前的频道。由于在一般情形下广告不会太长，如果用户在一小段时间内切换回来，我们就在原基础上继续累计时间。

对于用户在频道之间快速随意地浏览，我们可以设定一个阈值，比如 30 秒，小于这个阈值，则不将此节目列入用户的观看历史。对于某些用户只是开着电视机，并没有关注节目，此时节目的声音也许只是一种背景音乐，针对于这种情况，容易发现，用户不会拿着遥控器换台，即会在同一个频道逗留很久，同样，我们可以设定一个阈值，比如 3 小时，如果大于这个阈值，则不将这些节目记录入用户的观看历史。

7.5 时间矩阵

对于影视节目的推荐相对于网络上各种商品的推荐最大的不同在于影视节目的时间性。也就是说，影视节目只在某一个或几个时间段播放，而用户也大都喜欢有自己喜欢的收看时间段。注意，此处我们不考虑数字电视的回看功能，目前这个功能是收费的，也就是说并不是所有用户都能享受这样的自由。

如果将时间因素引入，我们计算影视节目之间的相似度方式就需要有所改变。对于两个播放时间段完全不同的影视节目，它们之间的内容相似度应该适当乘以一个衰减系数，而后得到最终相似度。反之，如果两个影视节目播放时间相近，它们之间的相似度就应该适当提升。

假设我们将一天 24 小时分为若干段，然后将原来的影视节目的向量空间扩展成含有时间的矩阵空间，即 $M \times N$ 矩阵空间，其中 M 是时间段数， N 是原向量维数。一般来说，某档节目只在某几个时间段播放，因此，在该影视节目的矩阵中，只有在它播放时间段的那几行非零，而在其他地方全为零。图 7-3 是一个例子。

$$\begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_N \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ a_1 & a_2 & a_3 & \cdots & a_N \\ 0 & 0 & 0 & \cdots & 0 \\ a_1 & a_2 & a_3 & \cdots & a_N \end{bmatrix}$$

图 7-3 时间矩阵

若把一天 24 小时分为 6 段，则矩阵为 $6 \times N$ 。图 7-3 中所示节目仅在第 1、4、6 个时间段播放。

我们如下定义时间矩阵之间的相似度：

设有两个影视节目 P_1, P_2 ，它们的矩阵分别为 A_1, A_2 ，计算 A_1 乘以 A_2 的转置 $R = A_1 \cdot A_2^T$ 。易知， R 中元素 (i, j) 的值代表 A_1 的第 i 行向量和 A_2 的第 j 行向量之间的相似度，即节目 P_1 在时间段 i 与节目 P_2 在时间段 j 之间的相似度。如果 P_1 在时间段 i 没有播放，或者 P_2 在时

间段 j 没有播放，则 R 中 (i, j) 的值为零；否则 R 中 (i, j) 的值非零。

根据矩阵乘法的运算法则，可以看到，在结果矩阵 R 的主对角线上的元素代表了 P_1 、 P_2 在某个相同时间段上的相似度。如果在主对角线上有元素非零，则说明它们的播放时间有重叠。类似可知，在结果矩阵 R 中，离主对角线较远的元素表明两个节目在相距较远的两个时间段上的相似度。又由时间段的特殊性，晚上 23 点和凌晨 1 点也应判断为是较邻近的。所以我们针对于结果矩阵 R 定义了公式(7-2)所示的衰减函数，充分考虑了时间的循环性。公式(7-2)中 M 是时间段数。图 7-4 是公式(7-2)的函数图像。

$$f(i, j) = \frac{H-h}{\frac{M}{2}} \left| |i - j| - \frac{M}{2} \right| + h \quad (7-2)$$

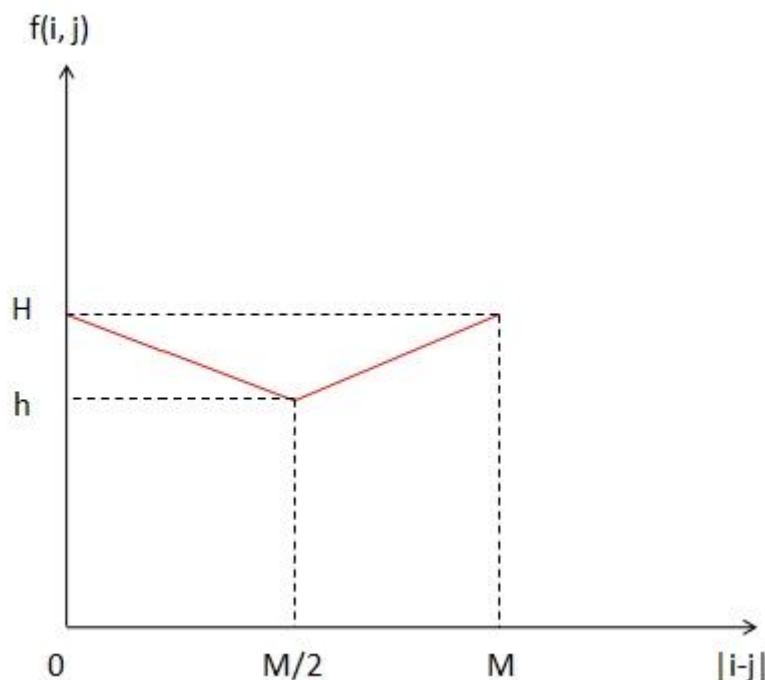


图 7-4 衰减函数的函数图像

现在我们定义矩阵之间的相似度为：

$$\text{matrix_sim}(A_1, A_2) = \max_{i,j=1}^n f(i, j) \cdot R(i, j) \quad (7-3)$$

公式(7-3)中， f 函数就是公式(7-2)， $R(i, j)$ 表示 R 中元素 (i, j) 的值。

图 7-5 是矩阵元素分布与时间的关系。

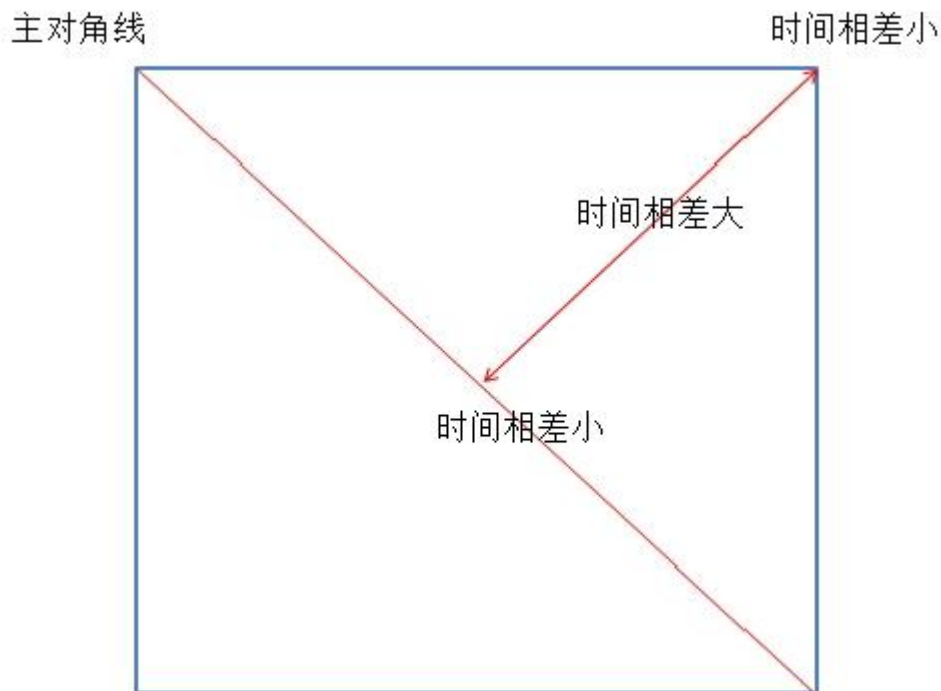


图 7-5 矩阵元素分布与时间的关系

这样定义的矩阵之间的相似度充分考虑了时间段的因素，时间因素的影响大小可以通过公式(7-2)进行调节。

矩阵乘法的时间复杂度是 $O(MN^2)$ ， M 是时间段数， N 是原向量维数。我们发现，矩阵是非常稀疏的，我们只要为每个影视节目维护一个时间数组 T ，当两个矩阵相乘的时候，只计算那些二者时间数组对应元素都非零的值。如 $T_1[2]=1$ ， $T_2[4]=1$ ，则在结果矩阵中只有元素(2, 4)的值非零，只需计算它即可。

类似地，我们同时将用户的个性向量空间也扩展成矩阵空间，即 $M \times N$ 矩阵空间，其中 M 是时间段数， N 是原向量维数。如果用户喜爱某一个影视节目，则将该影视节目的矩阵叠加到用户的个性矩阵上。由于影视节目的矩阵仅在某几行非零，叠加上去时，只会影响用户个性矩阵的某几行。这样我们就同时记录了用户的收看习惯(何时看电视)。

在推荐时，我们也按照时间矩阵的方式计算相似度，寻找内容相似、时间相近的影视节目进行推荐。

7.6 本章小结

本章系统论述了我们核心的推荐算法，引入了用户的个性向量和时间矩阵的概念。至此，我们完成了整个系统的构建。

第八章 系统测试

本系统的性能主要决定于网络信息抽取的准确度、聚类的纯度、推荐的精确度。

8.1 网络信息抽取的测试

基于链接长度比例的网络信息提取工具对于那些以整块普通文本为主题内容的网页效果最佳。

图 8-1 是未经处理的网页——新浪影音娱乐。图 8-2 是经过处理之后的页面。可以看到，广告都被滤掉了，而主体内容留下来了。



图 8-1 未经处理的网页——新浪影音娱乐

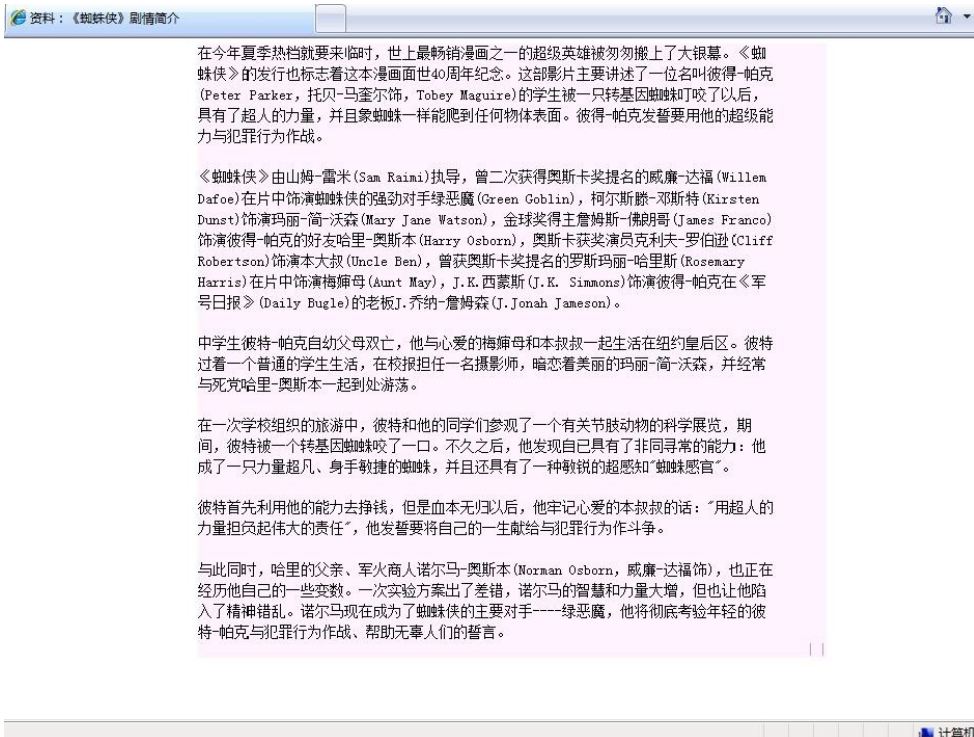


图 8-2 经过处理后的网页——新浪影音娱乐

对于维基百科这类普通文本中混有链接文本的网页效果也很好。

图 8-3 是未经处理的网页——维基百科。图 8-4 是经过处理之后的页面。可以看到，包含链接文本的主体内容都留下来了。

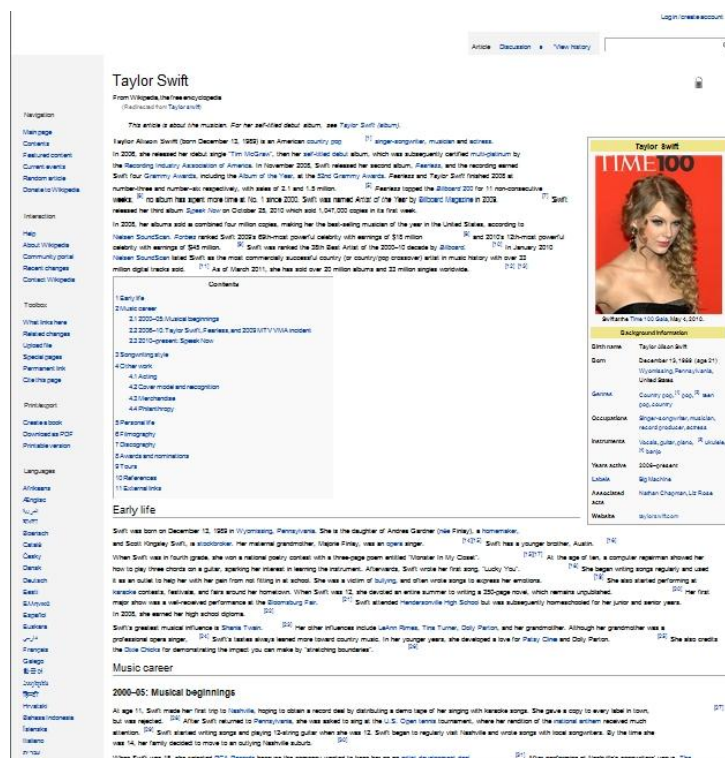


图 8-3 未经处理的网页——维基百科

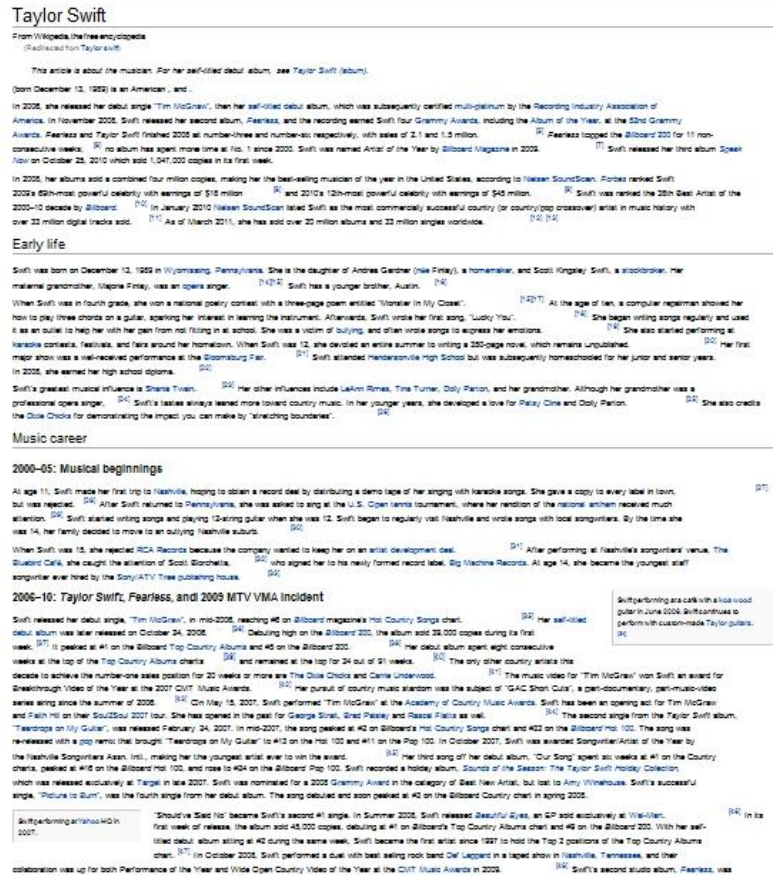


图 8-4 经过处理后的网页——维基百科

但是，对于一些链接列表存在误判的现象。比如一个表格中列举了导演和演员，但它们都是链接，指向关于该导演或演员的另外的页面。由于在该表格中链接数量过多，被提取工具判为噪声并删除。图 8-5 和图 8-6 就是一个例子，处理完之后，视频信息中的文本被成功留下，而导演演员表被删除。有一个弥补的方法，可以预先设定几个字符串，在过滤时一旦遇到不删除即可。



图 8-5 未经处理的网页——pptv 视频网站

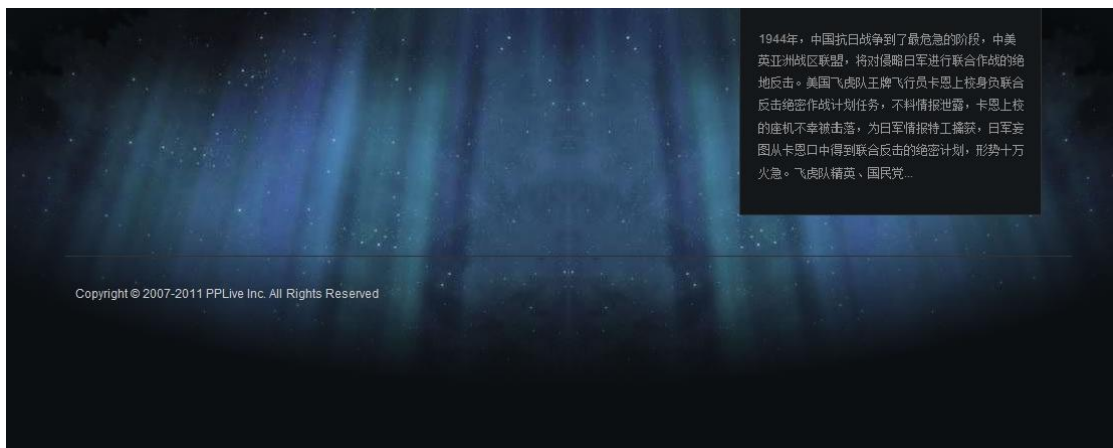


图 8-6 经过处理后的网页——pptv 视频网站

8.2 聚类的评价

聚类质量的内部准则是将簇内高相似度及簇间低相似度的目标进行形式化后得到的一个函数，但是，内部准则的高分并不意味着应用中的效果就一定好^[17]。另一种方法是根据应用的需求来直接评价，但是，这种做法很昂贵，需要大量用户调查。

代替用户判断的一种做法是，利用已有的分类文档集合作为评价基准^[17]。但是，我们

难以找到与本系统的分类标准相符合的已分类集合。因为本系统是按照影视节目的具体内容来进行分类的，而主流视频网站大都以影视节目的类别来分，分得较粗，如：剧情片、古装片等等。而本系统的分类较细，比如剧情片中还可以划分为很多小类，这样就难以准确评估。

于是，我们采取计算聚类纯度的方式。计算纯度时，每个簇被分配给该簇当中出现数目最多的文档所在的类别，然后通过正确分配的文档数除以文档集中的文档总数来得到纯度^[17]。纯度范围为[0, 1]。

当簇的数目很大时，很容易得到高纯度的结果。所以我们规定，簇之间的类别不得重复。

8.2.1 聚类测试一

首先我们选取了一些类型明显的影视节目，聚类结果如下：

S1={X 特工，亮剑，大刀向鬼子们的头上砍去，我的兄弟叫顺溜，敌营十八年，旗袍，潜伏，红岩，风声，黎明之前}

S2={中国式离婚，中国式结婚，你是我兄弟，双面胶，奋斗，媳妇的美好时代，我的团长我的团，老马家的幸福往事，蜗居，金婚，金婚风雨情}

S3={乾隆王朝，倚天屠龙记，天下粮仓，射雕英雄传，康熙微服私访记，康熙王朝，朱元璋，神雕侠侣，美人心计，走西口，还珠格格，铁齿铜牙纪晓岚，雍正王朝}

S4={功夫熊猫，将爱情进行到底，泰坦尼克号，绿灯侠，蜘蛛侠，蝙蝠侠，超人归来}

S5={海神号，空中监狱，阿凡达}

S1 可看成是战争剧，S2 可看成家庭情感剧，S3 可看成古装剧，S4 可看成大侠剧，S5 可看成惊险片。五个簇之间没有重复和包含的关系。

$$\text{计算纯度为 } \text{purity} = \frac{1}{44} (10 + 10 + 12 + 4 + 3) = 0.886$$

8.2.2 聚类测试二

我们再随机选取一组影视节目(2011 年 5 月 9 日至 5 月 15 日中央电视台一至十套节目中的部分电视剧、电影)，聚类结果如下：

S1={上海风云，京华烟云，伤痕我心，决不饶恕，凤凰起舞，刀锋，又见阿郎，女人如花，幸福密码，幸福来敲门，情系西西里，暗香，比爱更多（第一部），比爱更多（第二部），那些迷人的往事，香格里拉，龙凤呈祥}

S2={中华英雄，无敌英雄，马戏小子}

S3={九月杀，佛魔之剑，关东棋王，围剿，战争角落，未来警察，谍海风云，驯马手}

S4={乡恋，乱战，冰刀双人舞，功夫爹，喜神，女狙击手，惊天动地，戒烟不戒酒，文化战车，日照好人，死亡迷局，真相，血性山谷，踏界，错还，震撼世界的七日}

S5={亮剑，英雄郑成功，血战落魄桥}

S6={功夫拳霸三之火云盗，女杀手，孤胆义侠，尖峰时刻，幸福在 2010，海盗的宝藏，火线干探之永远的爱，灵魂战车（字幕），的士飚花（字幕），绝命速递，绝命速递 2，闯入者}

S7={嘻游记，塞上情仇，武当，武当龙影，藏客，跑吧马晓翠，辣嫂，隋朝来客，黑林鼓声}

S8={天下无双，新啼笑因缘，红粉金刚}

S9={天脉传奇，天龙八部，家家有本难念的经，至爱，连城诀}

S10={家有儿女, 少林四小龙, 蓝宝石}
S11={庄稼院里的年青人, 星火, 神厨小福贵}
S12={新仇旧恨, 碧血宝刀, 铁血金魂}
S13={缉毒英雄, 落花有声, 黑色走廊}

计算纯度为 0.807。

8.3 推荐部分的评价

由于条件限制, 无法搜集到电视用户最真实的第一手资料, 所以我们进行了小型的模拟测试。

准确率(precision)和召回率(recall)是评价推荐系统的常用标准^[4]:

$$\text{precision} = \frac{\text{recommended} \cap \text{interested}}{\text{recommended}} \quad (8-1)$$

$$\text{recall} = \frac{\text{recommended} \cap \text{interested}}{\text{interested}} \quad (8-2)$$

下面我们就用这两个标准进行评价。

8.3.1 推荐测试一

我们使用聚类测试一中的影视节目数据。

首先我们找到了 10 位测试者, 其中 3 位老年, 3 位中年, 3 位青年, 1 位少年。

我们将节目单分为两个部分 P1、P2, 每个部分都有不同的节目类型, 即都包含战争剧、家庭情感剧、古装剧、大侠剧和惊险片。

测试者首先从 P1 中选出他们喜欢的影视节目, 作为观看历史, 放入推荐系统中进行记录分析, 接着, 推荐系统推荐出 P2 中的节目, 再由测试者进行评价。

本系统测试结果为:

precision = 95.33% recall = 98.18%

8.3.2 推荐测试二

我们仍然使用聚类测试一中的影视节目数据。本测试中, 我们进行比较测试。

我们根据 item-based collaborative filtering recommendation algorithms 实现了协同推荐系统^[26]。将该协同推荐系统的测试结果与本系统在测试一中的结果进行比较。

item-based 推荐算法通过所有用户的行为来度量节目之间的相似度, 从而进行推荐^[26]。

在本测试中, 我们请 10 位测试者为喜欢和不喜欢影视节目打分(每位测试者不必为所有节目都打分), 通过这些打分为 item-based 推荐算法将会建立节目之间的相关度。

接着, 我们沿用测试一中的方式, 将节目单分为两个部分 P1、P2, 请另外 10 位测试者从 P1 中选出他们喜欢的影视节目并进行评分, 作为他们的观看历史, item-based 推荐算法会推荐出 P2 中的影视节目。

item-based 推荐算法测试结果为:

precision = 90.91% recall = 86.96%

8.3.3 数据分析

从测试一中的数据可以看到, 在聚类效果很好的情况下, 本系统的准确率和召回率都较高。

在测试二中，item-based 推荐算法是基于所有用户的行为的，它通过用户的行为来进行节目的相似度计算。由于用户的喜好广泛，算法可能将不同种类的影视节目归为一类，所以在测试二中 item-based 推荐算法的准确率下降。同时，如果有些影视节目没有用户看过或者评分，那么 item-based 推荐算法将无法在该节目和其他节目之间建立关联，所以在测试二中 item-based 推荐算法的召回率下降。

在作为参照的 item-based 推荐算法中，节目之间的相似度是建立在大量的所有用户观看记录或者评分之中的，所以能够推荐用户过去从未涉足的领域，但是对于一些品味独特的用户，个性化不足会导致准确率下降。对于新上线的影视节目，由于观看记录不足，无法建立关联，导致召回率下降。同时，该算法对于用户喜好变化的敏感度较低，不能紧紧跟随。

在测试中，我们发现，本系统针对于用户的个性化较强，能够紧跟用户的喜好变化。而且，本系统会对影视节目进行预处理，即使该节目从未被收看，我们仍能将其推荐给合适的用户。在本系统中，如果聚类的效果较好，则能保证准确率和召回率，但是，存在对用户喜好过分细化的缺点，即无法推荐用户过去从未涉足的领域。在推荐过程中引入的分叉(图 7-2)能够部分解决该问题。

综上所述，本系统能够很好地应用于影视节目的推荐，向用户提供个性化较强的推荐。

8.4 系统缺点

本系统从网络上抽取信息，信息的可靠度依赖搜索引擎的准确度。也就是说，如果搜索引擎返回的结果不正确，将会影响整个推荐效果。

本系统设计的网络信息抽取算法是基于链接文本相对于普通文本的长度比例的，这样做的确可以过滤掉很多广告之类的噪声，但存在假阳性和假阴性的错误判定。即，如果一段文本中有大量链接，比例高于阈值，会被删除，但其实是网页的主体内容；而对于一些版权信息、提示信息，由于是普通文本形式，会被误判为主体内容。

在测试中，两种误判形式都存在，但是由于文本总量较多，噪声所占的比例较小，而且我们引入了 tf-idf 能从一定程度上缓解噪声的影响。因为，类似于版权信息的段落描述方式极为相似，如果出现较为普遍，则 tf-idf 能将其权重降低。

正如 8.3 节中所述，在推荐部分，我们主要针对每个用户的偏爱建立了个性向量，存在过分细化的问题，没有考虑协同推荐。

8.5 本章小结

本章主要对系统进行了测试和评估，总体来说，效果不错。

第九章 结论

本论文详细论述了一个基于用户行为的影视节目分类和推荐系统的构建过程及其原理。

我们设计的基于链接长度比例的网络信息抽取算法,首先从网络上抽取与影视节目相关的文本信息,弥补了传统推荐系统信息不完整的缺陷。在对文本进行分词和 LSI 自然语言处理后,为每个影视节目构建一个能描述其特征的向量,构成向量空间。接着对影视节目进行聚类分析,合理分类,便于推荐相似的节目。

为了能提供个性化服务,我们也建立了用户的个性向量,来表征用户的偏好。通过寻找与用户的个性向量最相似的影视节目来进行推荐。同时,为了加入电视节目的时间敏感性度量,设计了时间矩阵,这样不仅能够推荐相似的节目,还能够适应用户的收看习惯。

我们对本推荐系统进行了测试评价,最终实现的系统能够进行具有一定精度的推荐。

在今后的工作中争取提高网络信息抽取算法的准确率,并在推荐部分引入协同推荐,同时增加节目的属性,如导演、演员、主持人等等。

参考文献

- [1] Gediminas Adomavicius, Alexander Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 17, NO. 6, JUNE 2005.
- [2] Srinivas Gutta, Kaushal Kurapati, KP Lee, Jacquelyn Martino, John Milanski, J. David Schaffer, John Zimmerman. TV Content Recommender System. American Association for Artificial Intelligence.2000.
- [3] James Davidson, Benjamin Liebal, Junning Liu, Palash Nandy, Taylor Van Vleet. The YouTube Video Recommendation System. ACM 978-1-60558-906-0/10/09.
- [4] SeungGwan Lee, Daeho Lee, and Sungwon Lee. Personalized DTV Program Recommendation System under a Cloud Computing Environment. IEEE Transactions on Consumer Electronics. 2010, Vol. 56, No. 2.
- [5] Zhiwen Yu, Xingshe Zhou, Yanbin Hao, and Jianhua Gu. TV Program Recommendation for Multiple Viewers Based on User Profile Merging. User Model User-Adap Inter (2006). 16: 63–82.
- [6] Claire Laudy, Jean-Gabriel Ganascia. Information Fusion in a TV program Recommendation System. IEEE Xplore. 2008.
- [7] Kazuki Ikawa, Tomohiro Fukuhara, Hideki Fujii, Hideaki Takeda. Evaluation of a TV Programs Recommendation using the EPG and Viewer's Log Data. www-kasm.nii.ac.jp/papers/takeda/10/ikawa10euroitv.pdf.
- [8] Suhit Gupta, Gail Kaiser, David Neistadt, Peter Grimm. DOM-based Content Extraction of HTML Documents. 12th International World Wide Web Conference, May 2003.
- [9] Suhit Gupta, Gail E. Kaiser, Peter Grimm, Michael F. Chiang, Justin Starren. Automating Content Extraction of HTML Documents. the World Wide Web Journal, January 2004.
- [10] A.F.R.Rahman, H.Alam, and R.Hartono. Content Extraction from HTML Documents. In 1st Int. Workshop on Web Document Analysis (WDA2001), 2001.
- [11] Suhit Gupta, Gail Kaiser, and Salvatore Stolfo. Extracting Context to Improve Accuracy for HTML Content Extraction. WWW 2005, May 10--14, 2005, Chiba, Japan.
- [12] Dayne Freitag. Information Extraction from HTML: Application of a General Machine Learning Approach. AAAI-98 Proceedings.
- [13] Alberto H.F.Laender, Berthier A.Ribeiro-Neto, Altigran S.da Silva, Juliana S. Teixeira. A Brief Survey of Web Data Extraction Toolst. SIGMOD Record, Vol. 31, No. 2, June 2002.
- [14] Arvind Arasu, Hector Garcia-Molina. Extracting Structured Data from Web Pages. SIGMOD 2003, June 9-12, 2003.
- [15] Bing Liu, Robert Grossman, Yanhong Zhai. Mining Data Records in Web Pages. SIGKDD '03, August 24-27, 2003.
- [16] Yanhong Zhai, Bing Liu. Web Data Extraction Based on Partial Tree Alignment. ACM 1-59593-046-9/05/0005.
- [17] Christopher D.Manning, Prabhakar Raghavan, Hinrich Schutze. Introduction to information retrieval. 北京:人民邮电出版社. 2010:1.

- [18] David A. Grossman, Ophir Frieder. Information retrieval: algorithms and heuristics second edition. 北京: 人民邮电出版社. 2010.
- [19] Alistair Moffat, Justin Zobel, Self-Indexing Inverted Files for Fast Text Retrieval. ACM Transactions on Information Systems, Vol. 14, No. 4, October 1996, Pages 349–379.
- [20] Yiming Yang. Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval. In Proc. SIGIR, pp.13-22. ACM Press. [291].
- [21] Pavel Berkhin. A Survey of Clustering Data Mining Techniques. Grouping Multidimensional Data: Recent Advances in Clustering, pp. 25-71. Springer. [343].
- [22] A.K. JAIN, M.N. MURTY, P.J. FLYNN. Data Clustering: A Review. ACM Computing Surveys, Vol. 31, No. 3, September 1999.
- [23] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman. Indexing by Latent Semantic Analysis. JASIS 41(6):391-407. [383].
- [24] Christopher D. Manning, Hinrich Schütze, Foundations of Statistical Natural Language Processing. 北京: 电子工业出版社. 2005.
- [25] Brendan J. Frey, Delbert Dueck. Clustering by Passing Messages Between Data Points. Science 16 February 2007: 315 (5814), 972-976.
- [26] Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl. Item-Based Collaborative Filtering Recommendation Algorithms. ACM 1-58113-348-0/01/0005. 2001.

谢辞

感谢我的导师朱其立副教授在研究期间给予的帮助和指导,感谢 ADAPT 实验室中的同学们的关心和支持。

在我们的学习和实验过程中,朱老师向我们推荐了很多优秀的参考资料,并与我们一起讨论、思考,为我们打开了思路、拓展了视野,老师严谨的教学态度使我们受益良多。

A PERSONALIZED TV PROGRAM RECOMMENDATION SYSTEM BASED ON VIEWERS' PREFERENCES AND TEXT CATEGORIZATION

The plethora of information has made it an exhausting task for consumers to target the data that fits their preferences. And recommendation systems have won the approval of many consumers due to the precision and personalized services. Therefore, it has become a promising research field.

The main functions of recommendation systems are tracking consumers' preferences and recommending products that may fit consumers' interest. Nowadays, recommendation systems have had far-reaching influence on every aspect of commerce. The main applications include book recommendation on Amazon.com, videos on youku.com, products on taobao.com. It's said that recommendation systems are accelerating the development of e-business and people are now quite dependent on them.

Meanwhile, the application of recommendation system on TV programs is also attractive. With the rapid development of modern digital communication, digital TV has become one of the main information sources in every household. The revolution of technology has enabled set-top boxes to receive and store large number of programs, which also makes it difficult for consumers to select appropriate programs swiftly and precisely. The employment of recommendation system on set-top boxes can bring new life to the whole industry.

Although there have already been lots of breakthroughs, researchers find it still necessary to improve the strategy and algorithm to achieve higher precision and wider cover.

In general, recommendation systems can be classified into three categories:

- (1) Content-based recommendation;
- (2) Collaborative recommendation;
- (3) Hybrid approaches.

Content-based recommendation stores consumers' history and recommend products similar to what the consumers have bought before. The characteristic of content-based recommendation is simplicity and efficiency. However, the shortcoming is over-specifying consumers' preferences. And to those new users, content-based recommendation cannot do the job precisely since there is not enough history.

Collaborative recommendation tries to recommend to one consumer according to the history of other consumers who enjoy similar tastes and preferences. Collaborative recommendation can recommend products that have never appeared in the history of consumers, but may cater to consumers' needs. However, it suffers from one problem: some consumers may be too different to

be classified into any group.

Hybrid approaches combine content-based methods and collaborative methods together, taking both advantages and making up disadvantages.

TV program recommendation is quite similar to other recommendation systems as far as algorithms are concerned. But there are three disparities:

- (1) The description about programs is hard to acquire;
- (2) Consumers' feedback is relatively little or difficult to analyze;
- (3) TV programs are time-sensitive.

Due to these disparities, recommendation systems for TV programs have attracted much attention.

Most recommendation systems pay lots of emphasis on how to analyze consumers' preferences, which is more than correct. Yet little work is applied to preprocessing of TV programs. Most of the theses abroad about recommendation systems assume there is already enough information about TV programs, which is not the same case in China. In order to solve this problem, we have developed a web information extraction tool to acquire enough description.

Our system combines web extraction and text categorization with traditional recommendation systems. The information acquired from the Internet offers integrity and sufficiency while text categorization brings flexibility.

In this paper, we mainly describe the design of a personalized TV program recommendation system based on viewers' preferences and text categorization. We have designed a web extraction tool analyzing the link length ratio, in order to solve the difficulty of getting TV programs' description. With the help of this tool, we can extract large amount of document data from the web. We then construct inverted index of all the documents and a vector space. We also introduce Latent Semantic Indexing to lessen the influence of synonymy. Then we categorize the programs using the method of text classification. Meanwhile, the system maintains a dynamic personalized vector for each consumer, based on which recommendation is made. At the end of this paper, we take time-sensitiveness into account and introduce time-matrix to capture more habits of consumers.

The overall structure of our system is composed of eight parts: web information extraction, Chinese segmentation, inverted index, Latent Semantic Index, vector space, clustering, personalized vector space, and time-matrix.

In web information extraction, there have been many methods:

- (1) Manual extraction;
- (2) Semi-automatic extraction;
- (3) Automatic extraction.

In our system, web pages come from search engine results, so they are of a wide range. As a result, no explicit rules can be found. Automatic extraction is the most appropriate choice.

Automatic extraction methods include:

- (1) Visual cues;
- (2) Tree alignment.

Visual cues method requires lots of training materials and tree alignment method is designed

for discovering data record. Thus, for the totally random web pages in our case, they do not offer effective approaches.

We propose an algorithm based on calculations of link text length ratio. We define several certain tags in HTML files as block tags. The HTML file is first parsed through a HTML parser which will correct mistakes and construct a Document Object Model (DOM) tree. Tree structure offers us great flexibility to traverse and analyze.

The whole algorithm is based on three observations:

- (1) Most noises are links. For example, ads, videos are in most cases links.
- (2) Similar contents tend to be in the same block or near blocks.
- (3) In the main content, the ratio of link text length to plain text is relatively small. In the contrary, the ratio in the noises tends to be large.

Therefore, when we navigate the DOM tree post-orderly, every time we detect a block tag, we calculate the ratio of link text length to plain text. If the ratio is greater than a certain threshold, we may tell it is noise and thus remove this node and all its children from the DOM-tree. Otherwise, the node is kept.

In this way, we can extract main content from web pages. This algorithm works better when there is an explicit main content block in web page.

After we get enough documents from the Internet, we go to process them.

In our system, we mainly focus on Chinese programs. Therefore the documents are all written in Chinese. To process Chinese, the first step is segmentation. We employ ICTCLAS, one of the most famous natural language tools in the world. For Chinese language, stemming and lemmatization are not as important as segmentation.

The documents are then turned into terms. We can then employ bag of words model where only term frequency accounts, not the order terms appear.

In order to save space and calculate term frequency conveniently, we introduce inverted index, which is composed of a dictionary and posting lists. We use term frequency and inverse document frequency to analyze the document precisely. And then vector space can be easily built.

There exist two typical problems in vector space model: synonymy and polysemy. We then introduce Latent Semantic Index (LSI) to reduce the influence of these problems. LSI is based on matrix singular value decomposition and low-rank approximation. Then we convert LSI back into a new vector space whose dimension is low and also involve co-occurrence phenomena.

Since each program has a corresponding vector, we can classify them into different categories.

There are mainly two different kinds of text categorizations:

- (1) Supervised classification;
- (2) Unsupervised classification.

Supervised classification requires training sets while unsupervised classification does not. Clustering is a typical unsupervised classification method.

In our system, we use Hierarchical Agglomerative Clustering (HAC) to classify programs and derive a dendrogram as a by-product which we may use in the recommendation part.

In the recommendation part, we try to tell consumers' preferences out of their history. Since we already get each program's vector, we can construct a vector for each consumer as well. When a consumer likes one program, we add the program's vector onto the consumer's personalized vector. In the long run, the personalized vector can represent the consumer's preference to some extent. And when recommending programs, we just search for the programs whose vectors are quite similar to the personalized vector.

People's preferences may change over time. That's why a decay function is employed in our approach.

The greatest disparity between TV program recommendation system and other recommendation system is its time-sensitiveness. In other words, a program may only be aired in a certain period of time of the day, and consumers may have their preferred watching time.

Therefore, we expand the original vector space into a time-matrix space where the time factor is involved. And we also define the similarity between two time-matrices where time similarity is calculated in it. And the similarity function takes matrix structure into consideration. So when we recommend programs, programs similar to consumers' tastes and watching habits will rank high.

In the end of this paper, we evaluate the whole system and display several results. Overall, our system works well.