创新实践 2

期末总结

姓　　名：**

学　　号：**

选课编号：**

指导教师：***

杭州电子科技大学

2018 年 1 月

新闻推荐系统的学习与实现

** [1)]

[1)](杭州电子科技大学 计算机科学与技术专业, 杭州 中国 310018)

摘 要 本文总结了本学期创新实践课程中学习并实现一个新闻推荐系统的过程。

关键词 新闻推荐系统;推荐算法;中文分词;数据挖掘;协同过滤;基于内容推荐

Learning News Recommendation System

** [1)]

1)(Hangzhou Dianzi University,310018, China)

Abstract This paper summarizes the process of learning News Recommendation system in Innovation and Practice class this semester.

Key words News Recommendation ; Recommendation Algorithm ; Chinese Word Segmentation；Data Mining ; Collaborative Filtering ; Content-based Recommendations

1 学期目标

1.1 学习途径

本学期《创新实践》课程的学习目标主要包括以下几个方面:①了解基本的网络协议的基础知识，实现对网页新闻的采集抓取，并储存到本地数据库;②了解中文分词算法,对抓取新闻的标题(或摘要等)进行分词;③基于分词结果,了解关键词筛选算法,筛选出关键词;④了解常用的新闻推荐算法的基本原理。

1.2 预期目标

尝试实现一个集数据采集、关键词提取、展示、用户分析、智能推荐等功能的 NBA 新闻智能推荐系统。

2 新闻推荐系统概述

2.1 新闻推荐系统简介

互联网的出现和普及给用户带来了大量的信息，满足了用户在信息时代对信息的需求，但随着网络的迅速发展而带来的网上信息量的大幅增长，使得用户在面对大量信息时无法从中获得对自己真正有用的那部分信息，对信息的使用效率反而降低了，这就是所谓的“信息超载”问题。

为了解决“信息超载”问题，互联网电商平台、内容提供商们一直以来都在不断改进智能推荐算法，希望能根据用户的信息需求、兴趣等，将用户感兴趣的信息、产品推荐给用户，从而提升用户满意度，实现利益最大化。

新闻推荐就是智能推荐领域中非常具有代表性的一个领域。如今，用户对新闻推荐的要求越来越高，除了各大网站推出的头条外，不同用户渴望读到的新闻千差万别，比如球迷群体对体育赛事的更为关注，股民群体对财经新闻更为关注，女性群体对娱乐八卦更为关注，科技发烧友们对科技数码、前沿新闻更感

兴趣……如何实现实现一个好的新闻推荐算法，使得它能智能分析用户群体，针对不同用户的不同需求推送不同的新闻，对提高用户黏性、提高内容点击率都有非常大的意义。因此，一个性能好、投放准的新闻推荐系统对于各大新闻网站都非常重要。

2.2 新闻推荐系统的难点

为实现一个好的新闻推荐系统，以下几个方面是难点、要点：

(1) 获取用户兴趣偏好

获取用户的兴趣偏好是推荐的前提条件，如何对不同的用户精准分类、画像是这一阶段要关注的难点和重点。在传统的 PC 时代，获取用户的偏好仅仅能对用户的注册信息、点击事件进行搜集和分析，而在如今的移动互联网时代，APP 能获取和搜集的用户信息量远远大于 PC 时代。不仅如此，许多 APP 之间也能互相共享搜集的用户资料，例如在搜索引擎搜索“龙井茶”，再次打开电商 APP，或许能看到电商平台为你推荐与“龙井茶”相关的商品。可以说，如今在获取用户兴趣偏好方面，比以前有了更多的渠道。

(2) 分析用户兴趣偏好

如何根据搜集到的用户兴趣偏好对用户进行画像分析，是本阶段的难点和重点。这时就需要不同的推荐算法起作用，常见的推荐算法有以下几类：基于内容推荐、协同过滤推荐、基于关联规则推荐、组合推荐等。

(3) 隐私和安全问题

推荐系统一般通过记录并分析移动用户的行为、位置等提供更准确的推荐，然而，出于隐私与安全的考虑，移动用户不愿意提供完整和准确的信息。推荐系统记录的信息也存在被泄露的风险。推荐系统通过记录移动用户的地理位置信息分析移动用户的移动轨迹，这里涉及到个人隐私问题。隐私保护和安全问题制约着移动推荐系统的发展。

(4) 推荐系统的评价

如何直观评价推荐系统的好坏是另一个重点、难点。以新闻推荐系统为例，不可能要求用户每次看完一篇新闻后进行反馈，新闻网站往往只能借助于 APP 的下载量、使用率等数据来间接评价推荐系统的好坏。

3 中文分词

3.1 分词概述

中文分词指的是将一个汉字序列切分成一个一个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。

在英文中，单词之间是以空格作为自然分界符的，而中文只是以字、句和段等通过明显的分界符来简单划界，唯独词没有一个形式上的分界符，因此中文分词比之英文分词要复杂、困难得多。

3.2 分词难点

在中文分词过程中，主要有一下两大难题：

(1) 歧义识别

歧义是指同样的一句话，可能有两种或者更多的切分方法。主要的歧义有两种：交集型歧义和组合型歧义。

①**交集型歧义**(Overlapped ambiguities)：A、X、B 分别为汉字串，如果其组成的汉字串 AXB 满足 AX 和 XB 同时为词，则汉字串 AXB 为交集型歧义字段。

例如：“研究生命的起源”可以切分为“研究/生命/的/起源”,也可以切分为“研究生/命/的/起源”。其中，“研究生命”为交集歧义字段。

②**组合型歧义**(Combinatorial ambiguities)：汉字串 AB 满足 A、B、AB 同时为词，则该汉字串为组合型歧义字段。 例如：“他从马上下来”可以切分为“他/从/马/上/,下来”,也可以切分为“他/从/马上/下

来”。其中，“马上”为组合型歧义字段。

目前，歧义切分的研究已经比较成熟，通过统计和规则相结合的方法，歧义字段的正确切分已经达到了较高的水平。

(2) 新词识别

未登录词，也就是那些在分词词典中没有收录，但又确实能称为词的那些词。最典型的是人名，人可以很容易理解。句子“王军虎去广州了”中，“王军虎”是个词，因为是一个人的名字，但要是让计算机去识别就困难了。如果把“王军虎”做为一个词收录到字典中去，全世界有那么多名字，而且每时每刻都有新增的人名，收录这些人名本身就是一项既不划算又巨大的工程。除了人名以外，还有机构名、地名、产品名、商标名、简称、省略语等都是很难处理的问题，而且这些又正好是人们经常使用的词，因此分词系统中的新词识别十分重要。新词识别准确率已经成为评价一个分词系统好坏的重要标志之一。

3.3 分词算法分类

目前的分词算法主要可分为三大类：**基于字符串匹配的分词算法**、**基于理解的分词算法**和**基于统计的分词算法**。

3.3.1 基于字符串匹配的分词算法

这种方法又叫做**机械分词方法**。它是按照一定的策略将待分析的汉字串与一个“词典”中的词条进行配，若在词典中找到某个字符串，则匹配成功，识别出一个词。

按照扫描方向的不同，串匹配分词方法可以分为正向匹配和逆向匹配；按照不同长度优先匹配的情况，可以分为最长匹配和最短匹配；按照是否与词性标注过程相结合，又可以分为单纯分词方法和分词与标注相结合的一体化方法。

3.3.2 基于理解的分词算法

这种分词方法是通过让计算机模拟人对句子的理解，达到识别词的效果。其基本思想就是在分词的同时进行句法、语义分析，利用句法信息和语义信息来处理歧义现象。这种分词方法需要使用大量的语言知识和信息。

但是由于汉语语言的复杂性，难以将各种语言信息组织成机器可直接读取的形式，因此目前基于理解的分词系统还处在试验阶段。

3.3.3 基于统计的分词算法

从形式上看，词是稳定的字的组合，因此在上下文中，相邻的字同时出现的次数越多，就越有可能构成一个词。因此字与字相邻共现的频率或概率能够较好的反映成词的可信度。可以对语料中相邻出现的各个字的组合的频度进行统计，计算它们的互现信息。

互现信息体现了汉字之间结合关系的紧密程度。当紧密程度高于某一个阈值时，便可认为此字组可能构成了一个词。这种方法只需对语料中的字组频度进行统计，不需要切分词典，因而又叫做**无词典分词法**或**统计取词方法**。

但这种方法也有一定的局限性，会经常抽出一些共现频度高、但并不是词的常用字组，例如“这一”、“之一”、“有的”、“我的”、“许多的”等，并且对常用词的识别精度差，时空开销大。实际应用的统计分词系统都要使用一部基本的分词词典进行串匹配分词，即同时使用基于字符串匹配的分词算法和基于统计的分词算法，这样既发挥字符串匹配速度快、效率高的特点，又发挥了统计分词结合上下文识别生词、自动消除歧义的优点。

3.4 常用分词算法的具体实现

3.4.1 基于字符串匹配的分词算法实现

以正向最大匹配算法为例：

算法思想：选取固定长 Maxlen 个汉字的符号串作为待切分字符串，把待切分字符串与词典中的单词条目相匹配，如果不能匹配，就去掉一个汉字继续匹配，直到在词典中找到相应的单词为止。注意，在以正向最大匹配算法中匹配方向是从左向右，减字方向是从右向左。

算法步骤：

(1)初始化字符串并设置最大符号串长度

S1 为待分析字符串，初始值为用户输入的句子

S2 为分词结果字符串，初始值为空

W 为候选子串，初始值为空

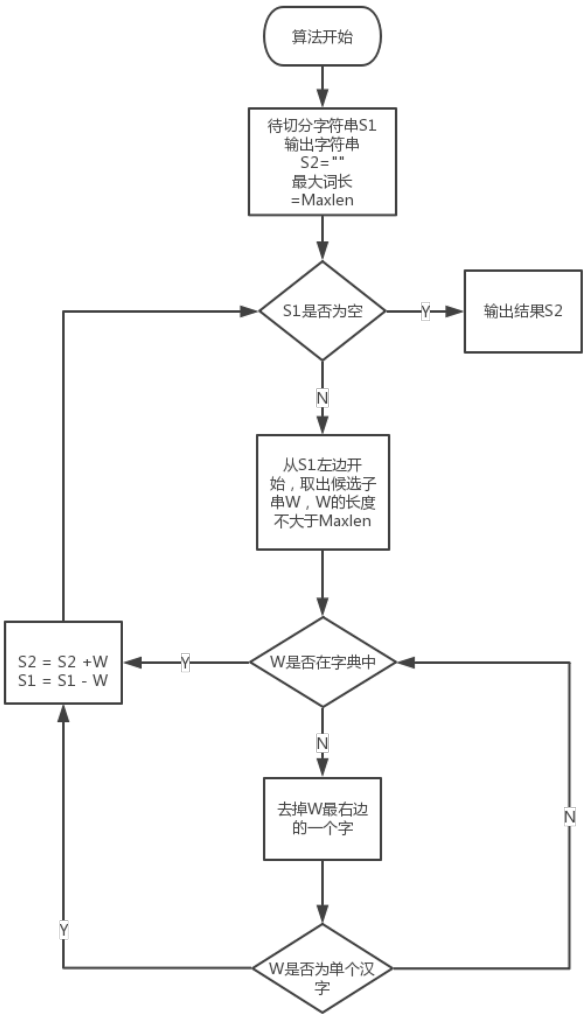
(2)若 S1 不为空，从 S1 左边取出候选子串 W

若 S1 为空，输出 S2 作为分词结果

(3)查词典，若 W 在词表中，将 W 加入到 S2 中，并将 W 从 S1 中去掉，转(2)

若 W 不在词表中，将 W 的最右边一个字去掉，转(3)

算法流程图：



3.4.1 基于统计的分词算法实现

基于统计的分词算法主要有：**N 元文法模型**（N-gram）、**隐马尔科夫模型**(Hidden Markov Model, HMM)两种。这里以 N 元文法模型为例详述基于统计的分词算法的基本实现。

算法思想：第 n 个词的出现只与前面 n-1 个词相关，而与其他任何词都不相关，整句的概率就是各个词出现概率的乘积。

对于一个字符串 T，假设由词序列 W1,W2,W3,…Wn 组成，则 T 的出现概率如下：

$$p(\mathrm{T}) = p(W1W2W3 \dots \mathrm{Wn}) = p(W1)p(W2|W1)p(W3|W1W2) \dots p(Wn|W1W2 \dots W_{n\text{-}1})$$

引入马尔科夫假设,假设一个词的出现仅仅依赖于它前面出现的有限的一个或者几个词。如果一个词的出现仅依赖于它前面出现的一个词，那么我们就称之为 Bi-gram，同样地，如果一个词的出现仅仅依赖于前面出现的两个词，那么我们就称之为 Tri-gram。

以 Bi-gram 为例，则 T 的概率公式可改写成：

$$p(T) \approx p(W1)p(W2|W1)p(W3|W2) \dots p(Wn|Wn_{n\text{-}1})$$

Bi-gram 和 Tri-gram 是实际中使用最多的两种模型，而且效果很不错。高于四元的用的很少，因为训练它需要更庞大的语料，而且数据稀疏严重，时间复杂度高，精度却提高的不多。

4 推荐算法

4.1 预备知识

4.1.1 余弦相似度算法

算法思想：用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小的度量。余弦值越接近 1，就表明夹角越接近 0 度，也就是两个向量越相似。

向量的余弦相似度计算公式如下：

$$similarity=\cos(\theta)=\frac{A\cdot B}{\|A\|\|B\|}=\frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

4.1.2 Jaccard 相似度算法

算法思想：两个集合 A 和 B 的交集元素在 A，B 的并集中所占的比例，称为两个集合的 Jaccard 相似系数，用符号 J(A,B)表示。

Jaccard 相似系数计算公式如下：

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|}=\frac{|A\cap B|}{|A|+|B|-|A\cap B|}$$

4.1.3 k 近邻算法(KNN)

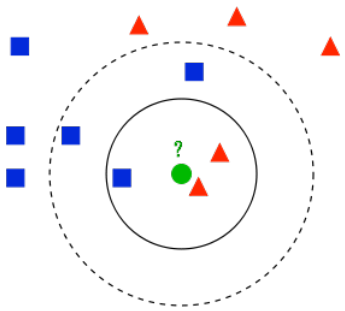
算法思想：K 近邻算法是一种简单的分类算法。一个样本与数据集中的 k 个样本最相似，如果这 k 个样本中的大多数属于某一个类别，则该样本也属于这个类别。

算法步骤：

(1) 给定测试对象，计算它与训练集中的每个对象的距离

(2) 圈定距离最近的 k 个训练对象，作为测试对象的近邻

(3) 根据这 k 个近邻归属的主要类别，来对测试对象分类



例如上图中，先要对标有问号的绿色点进行分类，已知的类别有蓝色正方形和红色三角形两种，选取 K 个距离最近的点，然后统计这 K 个点，在这 K 个点中频数最多的那一类就作为分类结果。

如果 K=3，绿色圆点的最近的 3 个邻居是 2 个红色小三角形和 1 个蓝色小正方形，基于统计的方法，判定绿色的这个待分类点属于红色的三角形一类。

如果 K=5，绿色圆点的最近的 5 个邻居是 2 个红色三角形和 3 个蓝色的正方形，基于统计的方法，判定绿色的这个待分类点属于蓝色的正方形一类。

4.1.4 TF-IDF 算法

TF-IDF 是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

TF 即 Term Frequency，词频。表示某一个给定的词语在该文件中出现的次数。

IDF 即 Inverse Document Frequency，逆向文件频率。是一个词语普遍重要性的度量。某一特定词语的 IDF，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取对数得到。

算法思想：如果某个词或短语在一篇文章中出现的频率 TF 高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

TF 的计算公式如下：

$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

其中，$n_{i,j}$ 是该词t_i 在文件d_j中的出现次数，而分母则是在文件d_j中所有字词的出现次数之和。

IDF 的计算公式如下：

$$\mathrm{idf_i} = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

其中，|D|是语料库中的文件总数，分母为包含词语t_i的文件数目。如果该词语不在语料库中，会出现除数为零的情况，一般用 $1+|\{j : t_i \in d_j\}|$ 作为分母。

最后，TF-IDF 计算公式如下：

$$\mathrm{tfidf_{i,j}} = \mathrm{tf_{i,j}} \times \mathrm{idf_i}$$

4.1.5 关联规则

关联规则可以描述成：项集 → 项集。

①项集 X 出现的**事务次数**（亦称为 support count）定义为：

$$\sigma(X)=|ti|X\subseteq ti, ti\in T|$$

其中，ti 表示某个事务，T 表示事务的集合。

②**支持度**（support）定义为：

$$s(X \longrightarrow Y) = \sigma(X \cup Y) \ / \ |T|$$

支持度刻画了项集 X∪Y 的出现频度。例如，比如某超市 2016 年有 100w 笔销售，顾客购买可乐又购买薯片有 20w 笔，顾客购买可乐又购买面包有 10w 笔，那可乐和薯片的关联规则的支持度是 20%，可乐和面包的支持度是 10%。

③**置信度**（confidence）定义为：

$$s(X \longrightarrow Y) = \sigma(X \cup Y) / \sigma(X)$$

置信度可理解为条件概率 p(Y|X)，度量在已知事务中包含了 X 时包含 Y 的概率。

例如，某超市 2016 年可乐购买次数 40w 笔，购买可乐又购买了薯片是 30w 笔，顾客购买可乐又购买面包有 10w 笔，则购买可乐又会购买薯片的置信度是 75%，购买可乐又购买面包的置信度是 25%。

④**频繁项集**是支持度大于**最小支持度阈值**的项集。

4.1.6 Apriori 算法

Apriori 表示“先验的，推测的”的意思。Apriori 算法第一个关联规则算法，用于做快速的关联规则分析，从而挖掘出**频繁项集**。

算法思想：利用逐层搜索的迭代方法找出数据库中项集的关系形成规则。

算法步骤：

(1) 设定最小支持度和最小置信度。

(2) 首先产生出候选的项的集合，即候选项集，若候选项集的支持度大于或等于最小支持度，则该候选项集为频繁项集

(3) 首先从数据库读入所有的事务，每个项都被看作候选 1-项集，得出各项的支持度，再使用频繁 1- 项集集合来产生候选 2-项集集合，因为先验原理保证所有非频繁的 1-项集的超集都是非频繁的

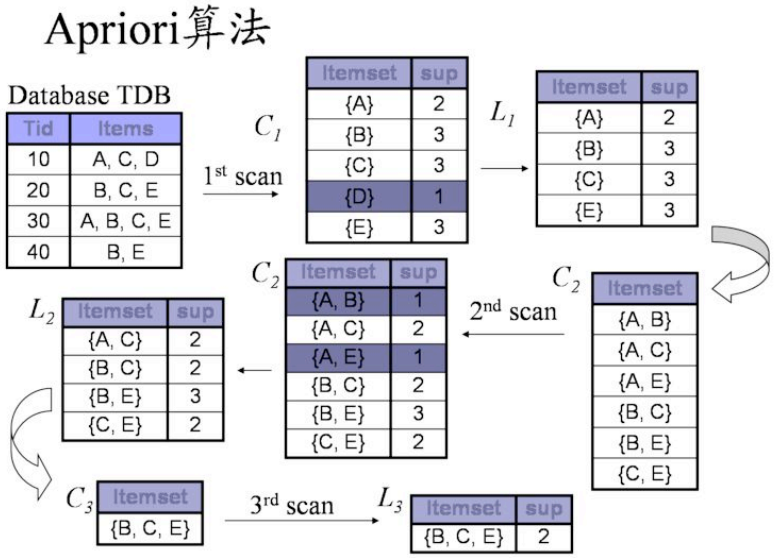
(4) 再扫描数据库，得出候选 2-项集集合，再找出频繁 2-项集，并利用这些频繁 2-项集集合来产生候选 3-项集。

(5) 重复扫描数据库，与最小支持度比较，产生更高层次的频繁项集，再从该集合里产生下一级候选项集,直到不再产生新的候选项集为止。

在此算法中要不断地重复两个步骤：连接和剪枝。具体内容如下：

(1) **连接**。为找 Fk，通过 Fk-1 与自己连接产生候选 k-项集。该候选项集的集合记做 Lk。设 F1 和 F2 是 Fk-1 中的项集。执行连接 Fk-1∞ Fk-1，其中 Fk-1 的元素 F1 和 F2 是可以连接的

(2) **剪枝**。Lk 的成员不一定都是频繁的,所有的频繁 k-项集都包含在 Lk 中。扫描数据库,确定 Lk 中每个候选集计数,并利用 Fk-1 剪掉 Lk 中的非频繁项,从而确定 Fk。

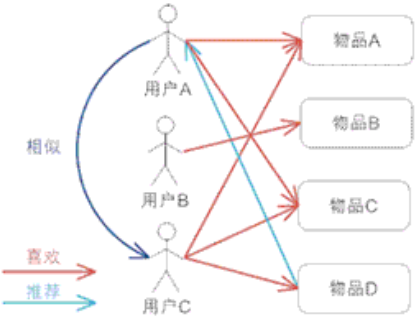


4.2 协同过滤算法

算法思想：找到与该用户偏好相似的其他用户，将他们感兴趣的内容推荐给该用户。协同过滤算法的基本思想有点类似“物以类聚，人以群分”，打个通俗的比方就是“如果你要了解一个人，可以从他最亲近的几个朋友去推测他是什么样的人”。

如下图所示，为用户 A 推荐相似偏好用户 C 喜欢的物品 D，就是一种协同过滤算法。

用户/物品	物品A	物品B	物品C	物品D
用户A	√		√	推荐
用户B		√		
用户C	√		√	√



算法过程：

(1) 对各个用户的偏好进行采集，抽象每个用户的偏好为某个数据结构,比如一个 Python 字典。

(2) 使用 KNN 算法寻找“相邻”用户，相似度的计算可以使用余弦相似度或 Jaccard 相似度，也可以采用其他相似度计算方法。

(3) 根据(2)的结果来计算“推荐度”，产生推荐列表。

协同过滤算法的优点：

①共享其他人的经验，避免了内容分析的不完全和不精确，并且能够基于一些复杂的，难以表述的概念（如信息质量、个人品味）进行推荐；

②基于内容的过滤推荐很多都是用户本来就熟悉的内容，而协同过滤可以发现用户潜在的但自己尚未发现的兴趣偏好；

协同过滤算法的问题：

稀疏性问题是协同过滤算法的主要问题。在大部分协同过滤算法的实现中，用户历史偏好是用稀疏矩阵进行存储的，而稀疏矩阵上的计算有些明显的问题，包括可能少部分人的错误偏好会对推荐的准确度有很大的影响等等。

4.3 基于内容的推荐算法

算法思想：把那些在内容上与该用户以往感兴趣的信息项目相似的项目再推荐给这个用户。基于内容的推荐算法类似于“江山易改，本性难移”。

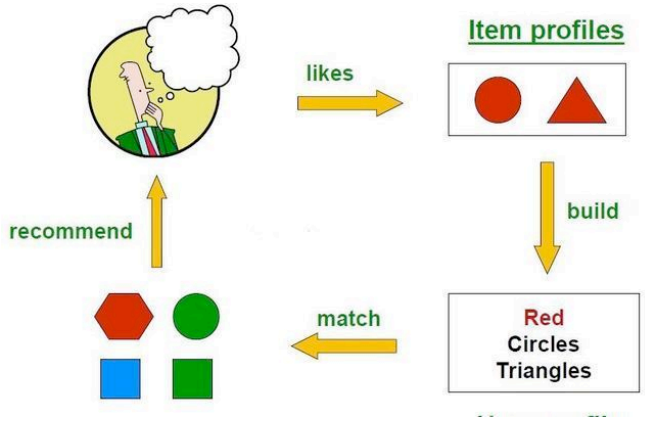
算法过程（以新闻推荐系统为例）：

(1)项目表示(Item Representation)。比如在新闻推荐系统中，将数据库中每一篇文章看作一个项目，使用分词技术提取新闻标题的关键词，再通过 TF-IDF 算法计算出文章中词的权重，利用这种方法，我们就可以把文章向量化。

(2)特征学习(Profile Learning)。利用一个用户过去喜欢(以及不喜欢)的项目的特征数据，来学习出此用户的喜好特征(profile)。比如在新闻推荐系统中，将用户过往浏览过的新闻通过(1)中方法进行向

量化处理，然后求平均值，代表该用户普遍喜欢的“内容”

(3)生成推荐(Recommendation Generation)。比如在新闻推荐系统中，我们利用相似性算法（包括余弦相似度、Jaccard 相似度）等计算要推荐的文字和用户特征 profile 距离最近的文章，即为推荐给用户的最佳选择。



基于内容的推荐算法的优点：

①只需要一个用户的历史数据，而不需要其他用户的数据。

②当数据库收录了一个新项目，比如一篇新闻，这篇新闻可以立即得到推送，不必像协同过滤算法那样要等到“相似”的人也浏览了才推荐。

基于内容的推荐算法的问题：

①使用 TF-IDF 算法进行文本向量化，不可能保留下文本的所有特征。

②无法像协同过滤算法那样挖掘用户的潜在兴趣。

4.4 基于关联规则的推荐算法

算法思想：通过不同项目之间的之间的支持度、置信度量化项目之间的“关联”，从而实现推荐。

算法步骤（以新闻推荐系统为例）：

(1) 数据清理，过滤一些特别冷门的新闻。

(2) 计算两两新闻之间的支持度、置信度，根据最低支持度、最低置信度，把低于阈值的规则扔掉

(3) 对新闻 A 进行推荐。找出新闻 A 的所有规则，按照置信度降序排序，Top-N 即为和新闻 A 最相关的前 N 篇新闻。

基于关联规则的算法的优点：

①推荐效果较好，当用户浏览了某频繁集合中的若干新闻后，浏览该频繁集合中其他新闻的可能性更高

基于关联规则的算法的问题：

①计算量大，对于数据库中的每两个项目间都要进行计算，如果新闻数据库庞大，这个计算量不可忽视

②存在热门项目容易被过度推荐的问题

4.5 其他推荐算法

4.5.1 基于图结构的推荐算法

用户一项目矩阵可建模为一个二分图（Bipartite Graph），其中节点分别表示用户和项目，边表示用户对项目的评价。基于图结构的推荐算法通过分析二部图结构给出合理的推荐。

4.5.2 混合推荐算法

混合推荐是为解决协同过滤、基于内容和基于图结构推荐算法各自问题而提出的，达到“相互取长补短”的推荐效果。例如，基于内容方法可以解决协同过滤中“新项目”问题，而协同过滤可降低基于内容算法面临的“过拟合”问题。混合推荐可以独立运用协同过滤、基于内容和基于图结构的推荐算法，将两者或多者产生的推荐结果进行融合，再将融合后的结果推荐给用户。

5 NBA 新闻推荐系统的具体实现

5.1 选题的原因和背景

本学期初我设想的是做一个像 Readhub (https://readhub.me) 一样的新闻网站。Readhub 作为一个科技新闻咨询站和传统的新闻网站不同，传统的新闻网站要么是自己撰写文章发布，要么是盲目转载别的门户网站的新闻，而 Readhub 选择了另一种策略：广泛从其他科技新闻网站搜集新闻，采用语义分析的方式合并那些话题相同或者内容相似的新闻，合并完成后再以最简洁的文字归纳推送给用户。简单的说，Readhub 实现了对新闻内容聚合、筛选和排序。

(Readhub 新闻首页)

然而，要想完成一个功能类似 Readhub 的网站，主要难点在于对新闻的语义分析，即如何判定两篇文章讲述的是同一件事情。传统的方法可以用上面提到的一些相似度算法对向量化之后的新闻文章进行比对和排序，但一个新闻数据库中的新闻数目不是一个小数目，计算量非常大且效率也很低。

后来更改目标为实现一个最简单的新闻推荐系统。简单在两个方面，一是数据源少，新闻只集中于一种新闻，这样就能减少在实现推荐时的计算量，数据库中的任意两篇文字之间的关联度也大；二是推荐算法简单，由于项目选择用 web 实现，使用浏览器的 Cookies 或者 Session 来记录用户点击过的关键词，通过这些历史关键词实现推荐。

之所以选择 NBA 新闻作为内容源有一下几个考虑：

(1) NBA 新闻的篇幅往往不长，非常便于分析和计算。而其他类别的新闻经常会有一些长篇大论。

(2) NBA 新闻的标题往往有非常明显的标志。比如球员名字，球队名字，这些关键词只要设置一部用户词典就可以非常准确的切分出来；

(3) 作为体育赛事，NBA 新闻往往有非常集中的热点事件爆发。比如“xx 球队大胜”，“xx 球员受伤”，“xx 球员退役”之类的新闻热点，而其他的新闻类别比如“财经”、“国际新闻”、“军事”等等，很难像 NBA 新闻那样有周期性的热点新闻出现，这给推荐相似新闻增加了难度。

(4) NBA 新闻文章的主要内容非常集中。无非就是球员、球队、比赛结果等等和赛事有关的事件，这些新闻报道之间的关联度高；而其他类别的新闻关注点非常大，同一类的两篇新闻之间的关联度有时非常低。比如今天国际新闻类别的两篇新闻《国防部发言人:中方坚持实现半岛无核化维护和平》和《埃及前总统穆尔西因侮辱法庭被加判 3 年有期徒刑》，虽然我们人一眼就可以看出来这两篇新闻都可以分为政治新闻，但是对于计算机来说，这两篇新闻之间的关联度非常低，因此，实现推荐也就非常困难。

5.2 新闻采集功能的实现

5.2.1 采集

本项目采用 Python 的第三方库 requests 和 BeautifulSoup 实现。

Requests 是用 Python 语言编写，采用 Apache2 Licensed 开源协议的 HTTP 库。Requests 使用的是 urllib3，因此继承了它的所有特性。Requests 支持 HTTP 连接保持和连接池，支持使用 cookie 保持会话，支持文件上传，支持自动确定响应内容的编码，支持国际化的 URL 和 POST 数据自动编码。现代、国际化、人性化。

Beautiful Soup 提供一些简单的、python 式的函数用来处理导航、搜索、修改分析树等功能。它是一个工具箱，通过解析文档为用户提供需要抓取的数据，因为简单，所以不需要多少代码就可以写出一个完整的应用程序。且 Beautiful Soup 能自动将输入文档转换为 Unicode 编码，输出文档转换为 utf-8 编码，大大方便了数据的采集。

使用 Beautiful Soup 对使用 requests 方法后得到的对象进行解析，结合对被采集页面 HTML 文件的分析，定位到所需的相关字段所在的 class 与 div 等标签，即可方便地采集目标网站的新闻数据。

本项目一共采集了 3 个 NBA 新闻相关网站的新闻数据：

(1) NBA 中国官方新闻站(http://china.nba.com/news/)

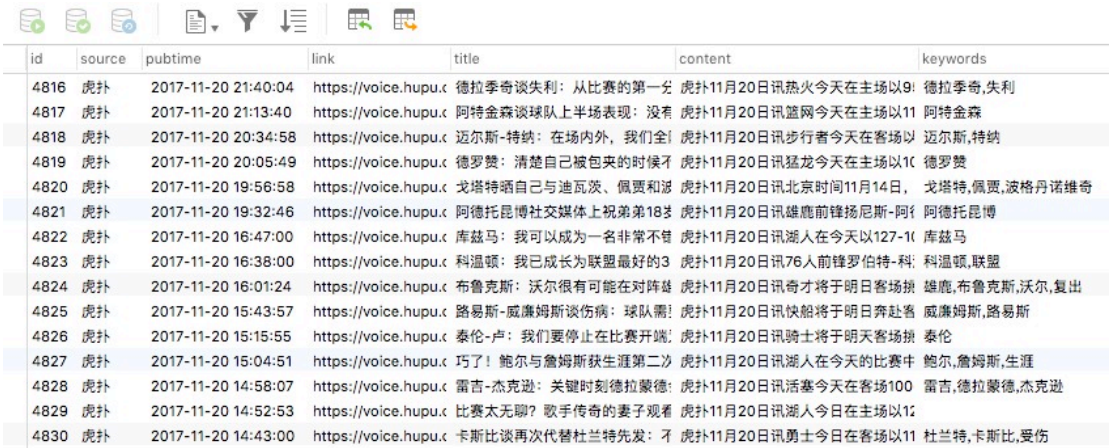
(2) 虎扑 NBA 新闻(https://voice.hupu.com/nba/1)

(3) 搜狐体育 NBA 新闻(http://sports.sohu.com/nba_a.shtml)

5.2.1 存储

本项目采集到的新闻数据保存在 MySQL 数据库中。

数据表的组织结构如下：



id	source	pubtime	link	title	content	keywords
4816	虎扑	2017-11-20 21:40:04	https://voice.hupu.c	德拉季奇谈失利：从比赛的第一分	虎扑11月20日讯热火今天在主场以9	德拉季奇,失利
4817	虎扑	2017-11-20 21:13:40	https://voice.hupu.c	阿特金森谈球队上半场表现：没有	虎扑11月20日讯篮网今天在主场以11	阿特金森
4818	虎扑	2017-11-20 20:34:58	https://voice.hupu.c	迈尔斯-特纳：在场内外，我们全	虎扑11月20日讯步行者今天在客场以	迈尔斯,特纳
4819	虎扑	2017-11-20 20:05:49	https://voice.hupu.c	德罗赞：清楚自己被包夹的时候才	虎扑11月20日讯猛龙今天在主场以10	德罗赞
4820	虎扑	2017-11-20 19:56:58	https://voice.hupu.c	戈塔特晒自己与迪瓦茨、佩贾和波	虎扑11月20日讯北京时间11月14日，	戈塔特,佩贾,波格丹诺维奇
4821	虎扑	2017-11-20 19:32:46	https://voice.hupu.c	阿德托昆博社交媒体上祝弟弟18岁	虎扑11月20日讯雄鹿前锋扬尼斯-阿德	阿德托昆博
4822	虎扑	2017-11-20 16:47:00	https://voice.hupu.c	库兹马：我可以成为一名非常不错	虎扑11月20日讯湖人在今天以127-10	库兹马
4823	虎扑	2017-11-20 16:38:00	https://voice.hupu.c	科温顿：我已成长为联盟最好的3	虎扑11月20日讯76人前锋罗伯特-科	科温顿,联盟
4824	虎扑	2017-11-20 16:01:24	https://voice.hupu.c	布鲁克斯：沃尔很有可能在对阵雄	虎扑11月20日讯奇才将于明日客场挑	雄鹿,布鲁克斯,沃尔,复出
4825	虎扑	2017-11-20 15:43:57	https://voice.hupu.c	路易斯-威廉姆斯谈伤病：球队需	虎扑11月20日讯快船将于明日奔赴客	威廉姆斯,路易斯
4826	虎扑	2017-11-20 15:15:55	https://voice.hupu.c	泰伦-卢：我们要停止在比赛开端	虎扑11月20日讯骑士将于明天客场挑	泰伦
4827	虎扑	2017-11-20 15:04:51	https://voice.hupu.c	巧了！鲍尔与詹姆斯获生涯第二次	虎扑11月20日讯湖人在今天的比赛中	鲍尔,詹姆斯,生涯
4828	虎扑	2017-11-20 14:58:07	https://voice.hupu.c	雷吉-杰克逊：关键时刻德拉蒙德	虎扑11月20日讯活塞今天在客场100	雷吉,德拉蒙德,杰克逊
4829	虎扑	2017-11-20 14:52:53	https://voice.hupu.c	比赛太无聊？歌手传奇的妻子观看	虎扑11月20日讯湖人今日在主场以12	
4830	虎扑	2017-11-20 14:43:00	https://voice.hupu.c	卡斯比谈再次代替杜兰特先发：不	虎扑11月20日讯勇士今日在客场以11	杜兰特,卡斯比,受伤

5.3 标题分词功能的实现

本项目采用 Python 的第三方分词库 Jieba 实现对新闻标题的关键词提取。

Jieba 分词库的特点如下：

①基于前缀词典实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图 (DAG)

②采用了动态规划查找最大概率路径，找出基于词频的最大切分组合

③对于未登录词，采用了基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法

④支持精准模式、全模式、搜索引擎模式三种各有特点的分词模式；

⑤支持导入自定义词典；

⑥离线分词，分词速度快。

由于 NBA 新闻的主要关键词集中于队名、球员名，而 Jieba 作为一个离线分词库对未登陆词的识别不如在线分词库来得准确。因此需要配合用户词典使用，本项目中另外采集了三部用户词典，分别为 coachdict.txt(教练员姓名词典)，playerdict.txt(球员姓名词典)，teamdict.txt(球队名词典)、eventdict.txt(事件词典，主要保存一些 NBA 等赛事专用的名词)

结果表示，使用 Jieba 分词库配合用户词典的分词效果还是非常理想的，对于大部分新闻的标题都能够准确提取出关键词，且速度相当之快。但有一些新闻可能出现无关键词的现象，这加大了未来推荐实现的难度。

5.4 Web实现

本项目使用 Django 作为 web 框架，配合 Bootstrap 框架对网页进行了一些基本的设计和美化。

通过 cookie 保存用户点击过的关键词，用于推荐。

（首页效果图）

（内容页效果图）

5.5 目前还存在的问题

① 首页的新闻展示的顺序没能按照时间顺序从近到远排序。因为每次抓取新闻都是从前往后抓，抓到数据库中已有的新闻为止，所以数据库中保存的新闻并不是按照顺序排列的。后续可以考虑在后端查询数据库时对“发布时间”字段进行排序。

② 由于期末时间较紧张，推荐部分仅仅是在学习了一些常见的推荐算法后做到了访问关键词的记录在 cookie 中，目前的系统尚不能实现根据历史访问的关键词智能排序，这个也有待于后续的完善。

③ 目前的采集程序不能采集原有新闻中的图片，仅仅能抓取文字，且抓取来的新闻正文会合成一段，不能保留原来的段落结构，后续考虑改进。

6 总结

本学期做的这个新闻推荐系统如果仅仅从功能实现上来讲还是做的比较粗糙，不够完善。但整体涉及的方面还是挺多的，包括算法、网页抓取、数据库、Python、Web 等等。经过了一个学期的学习，一方面是对现有的推荐算法和原理进行了较为系统的学习；另一方面也着实提高了我的动手能力，尤其是后面完成 Web 部分的时候，一开始我比较畏难，觉得从来没有学过网页制作相关课程，做起来难度会有点大。但后来在老师的指导和其他同学的指点下，我发现做一个简单的 Web 实现还是比较容易的，尤其是现在非常多的前端框架，可以让我们非常轻松地写出比较漂亮的页面。

总得来说，经过一个学期对新闻推荐系统的学习，自己还是非常有收获的。尤其是在算法方面，涉及到了相当多的自然语言处理、数据挖掘方面的经典算法，对我的提升很大。

致　谢　感谢邬惠峰老师的指导

参 考 文 献

[1] 吴登能. 面向移动互联网的个性化新闻推荐算法研究[D]. 杭州师范大学, 2013.

[2] 李乐田, 吴林, 高永存. 关于新闻推荐算法的研究[J]. 中国传媒大学学报: 自然科学版, 2016, 23(1): 40-44.

[3] 推荐系统-百度百科(https://baike.baidu.com/item/%E6%8E%A8%E8%8D%90%E7%B3%BB%E7%BB%9F)

[4] 杨博, 赵鹏飞. 推荐算法综述[J]. 山西大学学报: 自然科学版, 2011 (3): 337-350.

[5] 协同过滤-维基百科(https://zh.wikipedia.org/wiki/协同过滤)

[6] 协同过滤算法从原理到实现(http://blog.csdn.net/dark_scope/article/details/17228643)

[7] 常用的推荐算法解析 (http://blog.csdn.net/u014605728/article/details/51274814)

[8] TF-IDF与余弦相似性的应用-阮一峰的网络日志(http://www.ruanyifeng.com/blog/2013/03/tf-idf.html)

[9] K最近邻算法（KNN）(http://blog.csdn.net/saltriver/article/details/52502253)

[10] KNN算法 - HackerVirus - 博客园(https://www.cnblogs.com/Leo_wl/p/5602481.html)

[11] 基于内容的推荐算法-CSDN(http://blog.csdn.net/dq_dm/article/details/39851867)

[12] TF-IDF及其算法-CSDN(http://blog.csdn.net/sangyongjia/article/details/52440063)

[13] 周由, 戴牡红. 语义分析与 TF-IDF 方法相结合的新闻推荐技术[J]. 计算机科学, 2013 (S2): 267-269.

[14] 韩家炜, 坎伯. 数据挖掘: 概念与技术[M]. 机械工业出版社, 2012.

[15] 赵洪英, 蔡乐才, 李先杰. 关联规则挖掘的 Apriori 算法综述[J]. 四川理工学院学报 (自然科学版), 2011, 24(1): 66-70.

[16] 项亮. 推荐系统实战[J]. 2012.

[17] 中文分词算法总结(http://blog.csdn.net/yelbosh/article/details/45896051)

[18] 基于统计的词网格分词(http://blog.csdn.net/lengyuhong/article/details/6021461)

[19] 结巴中文分词(https://github.com/fxsjy/jieba)