西南林業大學
SOUTHWEST FORESTRY UNIVERSITY

本科毕业论文（设计）
（2020 届）

题 目 面向社交网络数据的主题跟踪算法与系统实现

学 院 大数据与智能工程学院 **专 业** 计算机科学与技术

学生姓名 葛宇航 **学 号** 20160952004

指导教师 张雁(教授)

李培培（教授/合肥工业大学）

评阅人

2020 年 4 月 20 日

原创性声明

本人郑重声明，所呈交的学位论文是本人在指导教师指导下进行的研究工作及取得的研究成果，论文成果归西南林业大学所有。尽我所知，除了论文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得西南林业大学或其他教育机构的学位或证书而使用过的材料。与我共同工作的同志对本研究所作的贡献均已在论文中作了明确的说明。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

作者签名：_____ **日期**：2020 年 4 月 20 日

面向社交网络数据的主题跟踪算法与系统实现

葛宇航

（西南林业大学 大数据与智能工程学院，云南昆明 650224）

摘　要：社交网络平台不断产生大量的文本数据，这些文本都是实时、大量、连续、有序的，并别通常长度较短，我们称之为短文本数据流。短文本数据流长度较短，缺乏充分的上下文信息，文本特征高维稀疏，数据产生速度快、数量大，并且会随时间产生潜在的概念漂移问题。因此，如何高效地从短文本数据流中发现有用信息（主题）成为研究的核心问题。Twitter作为英文语言环境最流行的社交平台，它为数据科学研究者、互联网公司等提供了丰富的短文本数据，利用这些海量的短文本数据，可以分析和洞察网络舆情，帮助互联网公司优化产品用户体验(如优化搜索引擎结果)。本文面向Twitter文本数据，设计一套针对短文本数据流的数据挖掘算法，对短文本数据流进行挖掘，从中提取有用的信息。通过Twitter官方提供的数据下载API获取到实验数据，用于本地模拟数据流。考虑到短文本的稀疏性问题，利用外部语料库扩展的方式，增强语义信息。借助BTM主题模型从文本数据中提取主题分布作为特征，并基于Scikit-Learn机器学习框架进行SVM和集成算法的搭建，并考虑到数据流随时间推移产生的概念漂移现象。最后，利用Django框架，采用B/S设计模式搭建一套可视化的数据挖掘平台，对集成分类算法进行了整合，同时提供可交互的用户界面，方便用户快速了解数据挖掘的基本流程，展示实验结果。

关键词：短文本数据流分类；主题模型；概念漂移；数据可视化；

Topic tracking algorithm and system implementation for social network data

Yuhang Ge

College of Big Data and Intelligence Engineering
Southwest Forestry University
Kunming 650224, Yunnan, China

Abstract: Social network platforms continue to generate large amounts of text data, which are real-time, large, continuous, and Sequential, and usually not short in length, we call it short text data flow. Short text data stream is short in length and lacks charging Divided context information, text features are high-dimensional and sparse, data is generated quickly, the number is large, and potentials are generated over time In the concept of drifting issues. Therefore, how to efficiently find useful information (topics) from short text data streams has become a research The core issue of research. As the most popular social platform for English language environment, Twitter is a data science researcher, Networked companies, have provided a wealth of short text data. Using these massive short text data, you can analyze and gain insight into the web Network public opinion to help Internet companies optimize product user experience (such as optimizing search engine results). This article is for Twitter Text data, design a set of data mining algorithms for short text data streams, mining short text data streams, from To extract useful information. Obtain experimental data through the official data download API provided by Twitter for local use Simulate data flow. Taking into account the sparseness of short text, the use of external corpus expansion methods to enhance semantic information. Using the BTM topic model to extract topic distribution from text data as features, and based on Scikit-Learn machine learning framework builds SVM and ensemble algorithms, and takes into account the concept drift phenomenon that the data stream generates over time. Finally, using the Django framework, the B/S design mode is used to build a visual data mining platform to integrate The

classification algorithm is integrated, and an interactive user interface is provided to facilitate users to quickly understand the basics of data mining.

Key words: Short Text Data Stream; Topic Model; Concept Drift; Data Visualization;

目录

插图

表格

1 绪论

随着移动网络时代的到来，各种社交网络APP也风靡全球，文本数据不断地产生。这些数据大多是短文本，并且数据规模大，数据信息冗余，我们通常无法直接从中获得有价值的信息。由此，借助计算机，使用数据挖掘技术对这些短文本数据进行数据分析显得尤为重要。

1.1 研究背景及意义

因为短文本的研究价值和其自身的特点，使得国内外研究者们越发重视这面的研究，关于短文本的研究不断涌现，包括针对短文本的分类[1, 2]和聚类[3, 4]、主题发现[5, 6]、特征选择[7-9]等数据挖掘算法。为了分析短文本数据中有价值的隐含知识，关于短文本分类的研究价值十分凸显。相比长文本,短文本自身长度较短,缺乏语义信息.导致其有严重稀疏性。由于整个语言字典中单词数量非常大，以英文来说，可能就能达到20万个左右，而如Twitter这样的社交网络平台单个文档中单词数最大不能超过140个，这就意味着表示每条短文本的向量中可能仅有几十维甚至几维有值，其余几万维都为零，从而造成短文本数据在表示形式上的严重稀疏性。且随着互联网的高速发展,用短短几万维的向量肯定是不足以表示海量的短文本数据,可能需要几十万维甚至几千万维的向量来表示一条短文本,因此短文本的高维问题也是处理短文本数据亟需解决的问题之一。由于短文本数据具有文本长度短，特征高维稀疏等问题，导致传统的文本分类方法很难有效处理，例如SVM[10]，随机森林[11]，贝叶斯网络[12]等分类方法在处理长文本上具有很好的效果，但对于短文本本身所具有的特性就很难适应.针对以上问题,国内外研究学者提出的各种解决方案.通过借助Web搜索引擎,从外部语料库获取文本扩展本地短文本,从而缓解文本稀疏

性问题[13]. 用熵来优化改进决策树,发现数据的隐藏规则[14] 。随着机器学习与数据挖掘技术的发展,文本分类算法层出不穷,计算力不断提高。针对目前研究中所遇到的如特征高维稀疏等问题也会逐渐得到解决。但面临即将到来的5G时代,数据量又会有一个甚至几个数量级的提高，短文本相应也会有新的问题出现等待解决。

1.2 本文主要研究内容

1.2.1 主要研究内容

本文面向社交网络的短文本数据流，通过数据挖掘技术，结合机器学习方法，对带有类标签的数据进行分析，并从中提取有用的信息。由于社交网络平台提取到的数据往往长度较短，没有足够的语义信息。本文采用的数据来源于Twitter，一条Twitter数据通常限制在140个单词以内。对短文本的向量化表示时，容易得到的是高纬稀疏的矩阵。这是语言模型自身带来的问题，例如，在使用文本分类任务中的最常见的文本表示方法词袋模型（Bag-of-Words），也就是最简单的一元的语言模型，通常会采集整个Twitter语料库中出现的单词组成的集合作为词汇表，由于使用的词汇是十分丰富的，达到几万甚至到十几万的量级，而一句话中出现的单词却很少，这样得到的短文本向量可能仅仅只有几维或者几十维，其余出现的都是零，从而造成了文本表示上的高维稀疏性。

正是由于这种高维稀疏性，使得传统的文本分类方法在处理时很难做到高效。例如，SVM，KNN，Bayes等方法在处理长文本问题具有很好的效果，但很难适用到短文本当中。

不仅如此，由于数据是流的形式，数据是持续无限的，并且产生的速度很快，隐含在数据中的信息就会随着时间产生迁移，带来严重的概念漂移现象，这种现象会导致模型的准确率下降甚至是失效。由此，数据高维稀疏和概念漂移是急需处理的问题。

主题模型是一种概率模型，不像传统的空间向量和语言模型那样，只是单纯地考虑文档在词典空间上的维度，而是引入了主题空间，从而实现了文档在主题空间上的表示。每个主题都是在词典空间上的概率分布，通过引入主题这个概念，就能很方便地文档进行低维表示，这便相应地可以缓解短文本数据出现的第一个问题–高维稀疏。另外，主题

模型还能够抽取文档中的隐含信息，即主题。使得特征表示更加精准。基于主题模型的特征表示方法，本文提出了一种带概念漂移检测的集成学习算法。

搭建一个基于Django框架的Web平台，深度展示了数据挖掘过程中的一般步骤，包括信息收集、数据集成、数据清理、数据变换、数据挖掘过程、算法应用以及数据可视化等。其中信息收集使用Twitter官方提供的“根据关键字搜索”API，从Twitter中获取到从2012年11月到2013年1月的数据，一共5个类别。考虑到计算量过大，实验中我们只使用了部分数据。

1.2.2 组织框架

本文内容一共分为5个章节，各章节的结构和主要内容如下:

第一章绪论，主要介绍了该研究课题的背景、意义、研究内容以及目的，最后简要给出了文章的组织结构。

第二章相关工作概述，介绍了短文本分类问题以及短文本数据流分类用到的相关技术和方法，概念漂移相关概念、定义，以及面临的挑战。

第三章本章是本文的核心章节，针对短文本高维稀疏和概念漂移现象，提出了带概念漂移的集成分类方法，介绍了该算法的设计思路，以及算法构建的流程。

第四章本章节主要介绍平台搭建的相关细节。

第五章总结和展望，对本文提出的方法和出现的问题作出总结与分析，并考虑今后继续研究的方向。

1.3 本章小结

随着互联网技术和通信技术的发展，人们渴望向外界分享信息，社交网络呈现一种爆炸式的增长，随之而来的数据也不断增加。这些数据大多是短文本数据，对这些短文本数据进行数据分析有十分重要的价值。在本章节中，首先简要了短文本数据流分类的研究背景和意义，介绍了解决该问题中可能出现的挑战，最后给出了全文组织结构。

2　相关技术及理论

本章将概述短文本分类问题的研究现状，介绍短文本分类概念，有监督的分类方法，文本扩展技术，基于主题模型的短文本分类方法和基于深度学习的短文本分类方法以及有监督的短文本数据流分类，并介绍本课题采用的技术和方法，最后给出了概念漂移相关的概述和简要解决办法。

2.1　引言

互联网上每时每刻都涌现出各种各样如推文、微博、评论、新闻标题等数据，通常他们都是由几个或几十个词组成，这种文本被称作是短文本。而这些数据从宏观上看，是持续无限的，呈现出一种流的形式，称之为短文本数据流。

短文本数据流的主要问题在于，实际应用的数据通常长度较短，上下文信息不足，数据高维稀疏，传统机器学习算法很难直接适用；此外，社交网络上的文本通常都是口语化的表达，导致其数据有较多的噪音；并且，数据都是以流的形式产生的，随着时间的推移，数据所隐含的信息可能会发生改变，这就带来了概念漂移的问题，该问题的出现，会影响模型的预测精度。针对这些问题，国内外科研人员提出了各种解决办法。Bollegala[15]等借助Web搜索引擎，通过查询两个词条同时出现的频率以及文本片段估计他们语义相似度，并由此进行语义扩展，缓解稀疏性问题。Sun[16]和Ramage[17]等人研究了数据去噪问题。Abel等提取了数据中的散列标签并与之前的数据进行关联，丰富了文本的语义。Tang等人提出了一种端对端的学习方法来扩展短文本，其借助注意力机制，每次保留有价值的相关文档，通过GRU模型整合文档，多次迭代扩展增加了文档的语义信息。

本章将介绍国内外学者在短文本数据流分类作出的成果，分析短文本分类中的难题，

并介绍解决这些问题需要的理论和技术基础。

2.2 文本分类的一般过程

2.2.1 概述

文本分类(Text classification)[18]，是为未标记的文本分配一组预定义类别的标签。利用文本分类可以完成很多事情，例如，将新闻以某个主题自动化归类，自动化进行语言对话，对某段文本进行情感分析等。通常，借助机器学习的手段，提取文本特征，利用分类器进行分类。例如下面这个句子：

```
"The Computer Science is a quite uesful subject."
```

将其作为模型的输入，模型通过分析其内容，即可自动化赋予一个标签”Computer Science Userful”：

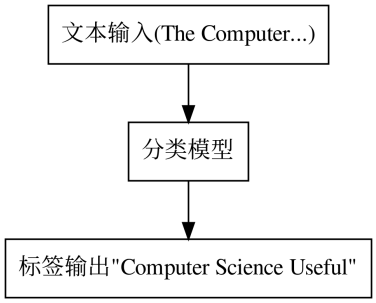

图 2.1 文本分类

2.2.2 流程

文本分类的一般化流程如图2.2所示，首先获取到原始数据，通常可使用API或爬虫。获取到原始数据后就要通常要先进行分词，本文采用英文文本，分词较为简单，直接取空格分隔即可。分词后，须对每个词进行数据清洗和预处理，取出数字、标点等特殊符号，去掉停用词、网址、超链接等。获取到“干净”的数据后，使用如TF-IDF、BOW、主题模型等方式对文本进行特征表示，最后使用提取到的特征，进行分类算法的应用。

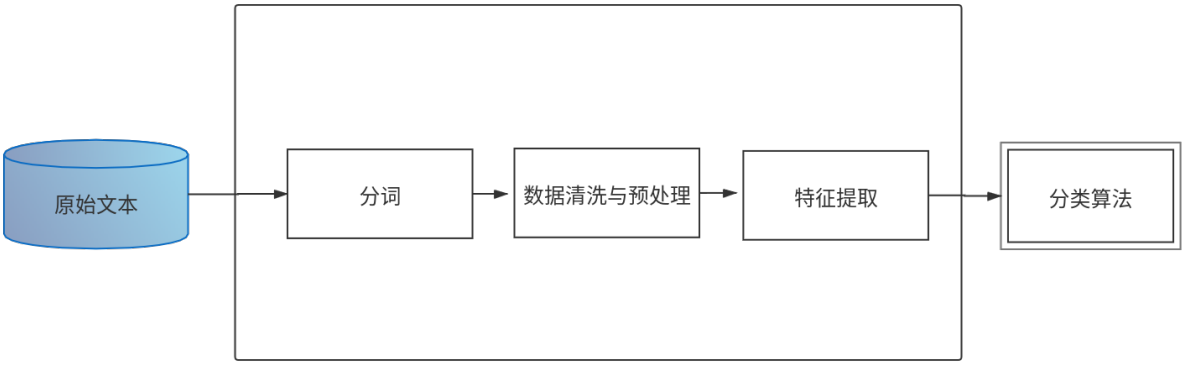

图 2.2 文本分类的一般流程

2.3 有监督的短文本分类方法

2.3.1 短文本预处理

文本预处理是信息检索和文本数据挖掘任务中的重要环节和关键性步骤。一般包括，文档分割、大小写转换分词、词干提取、词形还原、去除停用词、规范化、去噪等。

文档分割的步骤通常需要根据文本数据的形式来判断是否需要进行，如果数据集中的每篇文档没有过大且较为独立，则不需要进行。反之，如果所有的或者大部分数据保存在一个文档中，即需要进行文档切割，将其分成若干小份，方便后续处理。当然，若多篇文档处于同一个文件中，会有一些特殊的标记区分数据块，在编程时也很方便去处理。

如果文本是英文字符，则需要将其统一为小写字母，可以降低字典重复率。

分词，就是将一个句子分成一个个词汇的过程。目的是使得词变成最基础的“数据结构”，方便后续处理。分词在某种意义上就可以看成是对问题的形式化，将“无结构”的文本数据，整理成“结构化数据”，方便进行特征表示，分词是所有文本处理类问题形式化的第一步。

词干提取（steming），是将词语中的变形减少到其“根”的形式的过程。例如将“connect”、“connectedy”以及“connection”都转化成“connnect”。词干提取能够一定程度上缓解稀疏性问题，在搜索引擎中也常常会发现这个技术，比如你在Google搜索“deeplearning course”和“deep learning course”，通常你都可以获得正确的结果，就是因

为搜索引擎做了这样的处理。词形还原(lemmatization)，目标是删除变形并找到“根”形式。与词干提取区别在于，词形还原寻找正确的词缀，而词干提取只是在相同的位置做切断。下面是词形还原和词形提取的对比图:

表 2.1 词干提取和词形还原对比

Original word	stemmed word	lemmatized word
`trouble`	troubl	trouble
`troubling`	troubl	trouble
`troubled`	troubl	trouble
`troubles`	troubl	trouble
`troublesome`	troubl	trouble

停用词是一种语言中最常见的、无实际意义的词。例如，英文中的be动词、介词、连词等。去停用词的目的是删除文本中低信息量词对主题词的影响，使得算法对文本内容有更强的辨别性。例如:

```
"The Computer Science is a quite uesful subject."
```

去停用词后,

```
"Computer Science useful subject"
```

规范化, 例如“$100”可以表示成“one hundred dollars”,“2morrow”转化成“tomorrow”。去噪，用于删除文本非正常文本中的字符，如符号和数字等特殊字符，他们会对文本分析产生干扰。例如hashtag中开头的“#”，提醒关注的“@”，超文本链接“https”,转发推文的“RT”等都是需要被处理的。

2.3.2 词向量特征表示

类似于图像像素点，对于计算机来说，文本字符实际上是无法直接被理解的。在解决文本分类问题时，需先将文本转化成结构化的向量信息，才能继续应用算法进行处理。

最简单的特征表示方法就是词袋模型（Bag-of-Words）[19]，它将所有的语料中出现的单词看成一个词汇表，词汇表的大小即为向量空间的大小，文本中所有文档的维度都和

该词汇表的维度相同，通过统计每个词出现的次数，当作该词在向量中的权重。通俗的讲，就是将一篇文档看做是词的袋子，里面装着一个个不同的词。显然，文档被转换为一个个词以后，词和词之间的上下文关系也就丢失了。

TF-IDF(Term Frequency–Inverse Document Frequency)[20]，频率-逆文档频率，旨在反映单词对集合或语料库中的文档的重要程度。

TF（Term Frequency），表示某个单词出现的次数，即词频。其表达式如2.1所示:

$$TF(t_i) = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2.1}$$

其中，t_i表示文本中的第i个单词,$n_{i,j}$表示该单词在第j个中文档中出现的次数，而分母是整个语料中出现的所有词的出现次数之和。

IDF（Inverse Document Frequency, 逆文档频率), 是一个词语普遍重要性的度量, IDF的值越大，说明该单词重要性越高，更具有代表性。某一单词的IDF值可以由总文本数除以包含该单词的文本数，再对其得到的商求对数，其表达式如2.2所示:

$$IDF(t_i) = lg\frac{|D|}{|\{j : t_i \in d_j\}|} \tag{2.2}$$

其中，|D|为语料库中的文件总数，$|\{j : t_i \in d_j\}|$包含词语t_i的文件数目，如果词语不在数据中，就导致分母为零，因此一般情况下使用$1 + |\{j : t_i \in d_j\}|$。最后，得到TF-IDF:

$$TF - IDF_{i,j} = TF_{i,j} \times IDF_i \tag{2.3}$$

TF-IDF本质上基于词袋模型。

2.3.3 主题模型

主题模型是一种基于概率的非监督聚类统计模型，它通过挖掘文档的隐含语义结构（Latent Semanmitc Structure），对文档进行分析。不同于传统的语言模型，只考虑文档在空间上的维度，而引入主题的概念，实现文档在主题空间上的表示。主题模型在文本挖掘问题中的应用十分广泛，其中最著名的就是David Blei提出的LDA（Latent Dirichlet

Allocation）主题模型[21]。

LDA是一种生成模型，假设整个语料库一共包含K个主题，每个主题z都被表示成一个词典V上的一元语言模型ϑ_z，即词典上的一个多项式分布。我们进一步假设每个文档对于这个主题有一个文档特定的多项式Φ_d。那么，文档的生成过程如图2.3:

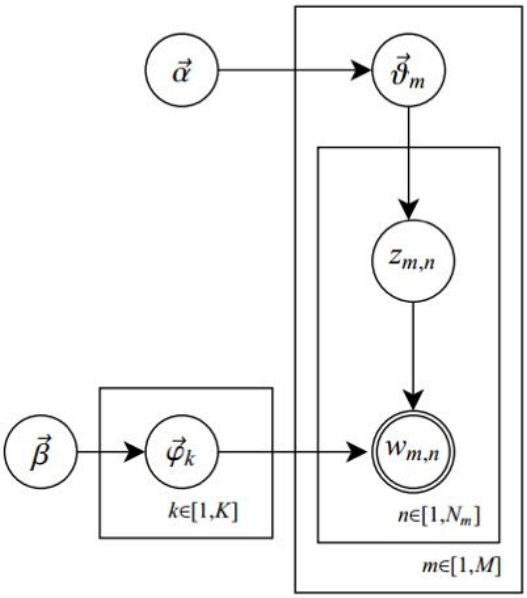

图 2.3 LDA主题模型

其中，N_m表示文档中词语的数量，M表示文档的总数量，K表示主题数，ϑ表示文档与主题之间的多项式分布，φ表示主题和主题词之间的多项式分布，α和β分别是ϑ和φ的Dirichlet先验参数，$Z_{m,n}$表示词语的主题分布，$w_{m,n}$表示生成的词。生成文档的步骤如下:

- 从参数为α的Dirichlet分布中采样生成主题的多项式分布φ,
- 从主题的多项式分布φ中采样生成第j个主题词$z_{i,j}$
- 从参数为β的Dirichlet分布中采样生成主题$z_{i,j}$对应词语的多项式分布φ_k
- 从词语的多项式分布φ_k中采样生成最终词语$w_{i,j}$

2.4 有监督的短文本数据流分类方法

2.4.1 数据流定义

数据流是由随着时间推移到达的大量的、连续的数据项组成的序列[22]。若令t表示某一时刻，则数据流可形式化地表示为$D = \{d_1, d_2, ..., d_{t-1}, d_t, d_{t+1}, ...\}$其中，$d_t = \{x_t, y_t\}$，$x_t$为第i个属性值，$y_t$表示该属性值对应的类标签。

数据流具备动态实时、持续到达、易变、高维稀疏等特点。正是由于这些特点，使得数据流无法将传统的机器学习算法如决策树、支持向量机、贝叶斯等算法直接应用，这给数据流分析带来了挑战。

2.4.2 支持向量机

支持向量机（Support Vector machine），是一种有监督（Supervised Learning）[23]的二元分类器。首次由Vapnik等人1964年提出[24]，而后又经过多次优化与提高，使其可以应用于多元分类任务[25]。其基本模型定义为特征空间上的间隔最大的线性分类器，即在样本空间中找到一个超平面，使得它可以最大分隔两个类。

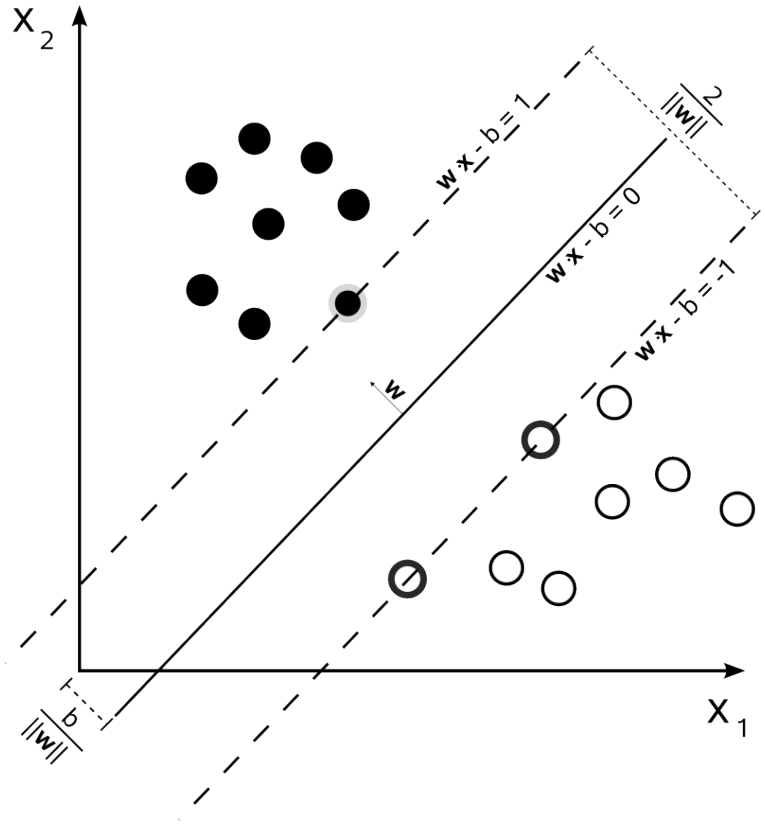

图 2.4 支持向量机

如图2.4所示, 其中$wx-b=1$和$wx-b=-1$表示两条支持向量, 中间的$wx-b=0$即为找到的超平面。对于这样一个二维数据集来说，最优超平面的寻找较为简单，但实际使用中的数据集往往远超二维，对于这样的高维数据来说，不一定都是线性可分的，因此想要找到最佳超平面的就需要借助核函数，将样本空间映射到更高维的空间，使得数据变得线性可分，以方便更好的进行样本分离。通常，核函数的选择也会直接影响到模型的分类性能，故如何选择更好的核函数也很重要，下面是几种SVM常用的核函数:

表 2.2 SVM核函数对比

核函数	计算公式	描述
线性核函数	$K(x_i, x_j) = x_i^T x_j$	其中，K为半正定核矩阵，适用于线性可分的数据，速度较快
多项式核函数	$K(x_i, x_j) = (x_i^T x_j)^d$	其中，$d \geq 1$
高斯核函数	$K(x_i, x_j) = exp(-\frac{\|\|x_i - x_j\|\|^2}{2\sigma^2})$	其中，$\sigma \geq 0$为高斯核的带宽
拉普拉斯核函数	$K(x_i, x_j) = exp(-\frac{\|\|x_i - x_j\|\|}{\sigma})$	$\sigma \geq 0$
Sigmoid 核函数	$K(x_i, x_j) = tanh(\beta x_i^T x_j + \theta)$	其中，$tanh$为双曲正切函数，$\beta > 0$，$\theta > 0$

2.4.3 集成方法

集成方法也叫集成学习（Ensemble Learning)，不同于决策树、支持向量机等分类器的是，它不是一个新的分类算法，而是一种机器学习范式。集成方法的思想是，通过训练多个分类模型，将所有的模型综合起来得到更好的预测结果。大量的实践证明它能够带来更高的准确率和更强的鲁棒性。

在快速数据流中，分类模型通常不能稳健地建立。因此，采用集成分类方法，通过组合不同的分类器可以使模型更加健壮。另一方面，数据以流的形式出现，在机器的内存中直接容纳大量流数据是不切实际的，而且通常是不可行的。因此在这种情况下，可以借助集成模型，连续更新并通过使用最近一批数据进行再训练来增量地训练预测模型。bagging和boosting是两种最常见的集成方法[26]。

Street等[27]将集成应用到文本数据流分类中，思想是以块为单位顺序读取训练数据，在当前块学习分类器,并在下一个块上进行评估,缺点是没有考虑到概念漂移。wettschereck等[28]设计了处理概念漂移的集成方法，它可以有效地发现数据流的迁移，该方法中，数据流被分割成块，每个块上都有多个分类器，并且最终的分类权值通过每个块上函数计算。kotler,zhang等[29, 30]也都使用这种模型加权或模型选择方法，来确保概念漂移数据流分类获得更好的精度。

2.4.4 概念漂移

概念漂移（Concept Drift）[31]是指在非平稳的环境中，数据分布会随着时间而变化，改变数据流上下文中的隐含信息，从而产生概念漂移现象。数据流分类的目标是训练分类器，建立一个类标签和特征之间的函数关系。而概念漂移是数据流分类中最早出现的，也是比较棘手的问题之一。

一个典型的示例是用户对新闻信息流兴趣的变化，虽然新闻文档的分发通常保持不变，但该用户感兴趣的新闻文档的条件分布却发生了变化。

最通用的解决思路就是通过历史数据分析，抽象出数据的模式随时间变化的规律，其中可能包括若干趋势和周期的混杂信号。但多数情况下，随时间的变化有很强的随机性，这很难做到。

在过去的一段时间里，与概念漂移有关的学习的研究越来越多，并且已经开发了许多漂移感知的自适应学习算法。自适应学习是指在运行过程中在线更新预测模型以对概念漂移做出反应。Tsymbal等人在2004年发表的概念漂移综述文章[32]对该问题给出了相对较全的定义，并附出了当时相关工作；kuncheva[33, 34]将集成学习技术应用于概念漂移检测上。maloof等提出了归纳规则学习算法[35]；shirakawa等基于语义扩展的方法[36]以及phan等基于主题模型的方法[37]下面将主要将描述自适应学习算法的特点。

自适应学习算法可以看作是先进的增量学习算法，能够随着时间的推移适应生成数据的过程。假设数据满足独立同分布，首先建立静态的基模型作为评估基准，默认随时间推移自变量和因变量的映射关系一致，在通过这个模型检测是否存在概念漂移。在训练之前，根据时序给样本不同的权重，时间越新的样本权重可以给大一点，越老的数据可以减少权重。然后定期的加入新数据更新模型。在加入新数据的时候，也可以进一步筛选出最适合的样本进行重新训练，得到更加适合的模型。

2.5 平台开发相关技术

本文处实现相关算法外，还构建了一个可视化的数据挖掘平台。本小节将介绍搭建数据挖掘平台需要用到的Web相关技术，其中包括Django框架、MongoDB数据库和Echart

可视化图标的相关介绍。

2.5.1 Django框架

Django是一个高级的Python网络框架，通过自动化或简化Web开发的常见任务，快速开发安全和可维护的网站。Django让代码编写者只需要专注于编写应用程序，网站开发中麻烦的部分已经被封装完成，无需重新开发。Django开发的应用有如下几个优点

- 具有很好的完备性，几乎提供Web开发所需要的所有模块；
- 可以构建几乎任何类型的后端，处理各种格式（HTML、JSON、XML）的数据；
- 重视安全问题，例如常见的SQL注入、CSRF攻击等都有很好的防护；
- 采用类MVT的设计原则，易于扩展性、可维护性高；
- 灵活性高，采用Python编写，天生具备跨平台的能力；

Django的核心是一组高效协同工作的库，涵盖了Web开发的各个方面，其中包括：ORM对象关系映射器，该库知道数据库的结构，代码的结构，并且无需重复的手写SQL语句就能弥合它们之间的鸿沟。一组HTTP库，它们知道如何解析传入的Web请求，并返回标准格式给用户。URL路由库，可让您准确定义所需的URL并将它们映射到代码的适当位置。一个视图系统,用于处理请求。Template模板系统，使用它编写混合模板语言的HTML代码，接受后端传来的数据，在网页中显示表单并处理用户提交的数据。图2.5是Django的整体框架图。

2.5.2 MongoDB

MongoDB是一种NoSQL数据库，NoSQL被称作“Not Only SQL”，在数据爆炸的今天使用十分广泛。几年前，应用程序通常只拥有数千个用户到上万个用户，而现在流行的APP如“新浪微博”、“Wechat”的用户都数以亿计，并且每年365天，每天7*24小时处于连接状态。传统的关系型数据库在处理少量数据时它们具有良好的性能。但处理当下信息大爆炸时代互联网，多媒体和社交网络的海量数据，使用传统的关系数据库效率低下。为了克服这个问题，引入了“NO SQL”一词。NoSQL术语由Carlo Strozzi于1998年创造，指非关系数据库。MongoDB就是最流行的NoSQL数据库之一。

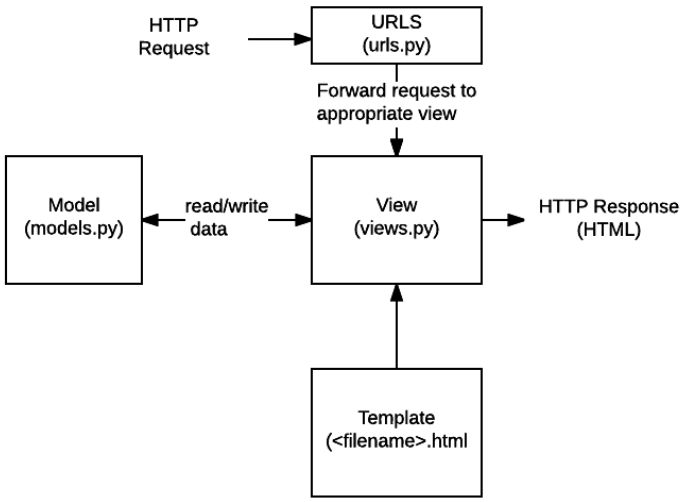

图 2.5 Django MVT设计框架图[38]

MongoDB采用C++语言编写的，一个开源的分布式文件存储数据库系统。MongoDB将所有的数据都是为文档，文档数据结构表示方式采用Key-Value的形式，类似于JSON。

MongoDB主要特点如下:

- 可实现传统数据库能实现的所有操作;
- 采用面向文档的设计模式，操作更加简单;
- 由于存储特点，可应用于分布式计算;
- 读取效率相比传统数据库更高;
- 丰富的查询表达式。可轻易查询文档中内嵌的对象及数组;
- 支持多种编程语言，安装便捷;

图2.6是MongoDB的架构:

2.5.3 Echart

ECharts（Enterprise Charts）是国产的应用于浏览器的可视化工具箱。使用简单的Javascript语法，就可以完成对数据的操作，同时画出美观的图表。

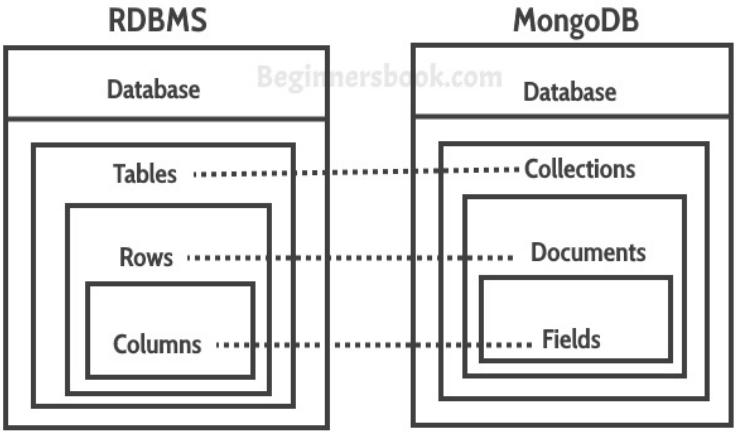

图 2.6 RDBMS vs MongoDB[39]

2.6 本章小结

本章节详细介绍了本文研究问题相关的技术和理论，其中包括大数据分析的基本流程，讨论了文本分类常用的技术手段。并对有监督的文本分类方法和有监督的文本数据流分类方法分别做了理论介绍，大致包括文本预处理、语言模型、词特征表示、主题模型、数据流定义、支持向量机、集成分类方法、概念漂移现象等。通过阅读本章节，读者可以对本文研究的背景知识有了大致的了解。

3 基于概念漂移检测的短文本数据流分类算法设计

本文提出一种基于概念漂移检测的短文本数据流分类算法，用于对短文本数据流进行隐含主题的跟踪。该方法采用集成学习, 将时序的文本数据流进行分块, 针对每个块训练一个基础SVM分类器，并根新旧据块之间的语义距离判断是否发生概念漂移，结合分配给每个分类器的权重，动态地更新模型池中的基分类器，使得模型池中始终保持恒定数量的分类器。对于新到来的未知标签的数据块，采用该集成模型进行预测和分析，这就是本文提出的主题跟踪的算法基础。

3.1 框架设计

假定数据流由N个数据块组成，命名为$D = \{D_1, D_2, ..., D_N\}$，每个数据块包含一些短文本，命名为$D_i = \{d_1, d_2, ..., d_n\}$,其中n表示第i个数据块的短文本数目。通常，每个文档都能被表示成一个向量空间，命名为$d_j = \{(v_x, v_y)|v_x \in R^M, v_y \in Y\}$，其中$R^M$表示一个文档空间，Y表示该文档的标签，M表示维度。

为解决短文本数据流的稀疏性问题，本文使用Wikipedia作为外部语料库对短文本进行语义扩展，缓解该问题，扩展后的数据块表示为$D_i^{'} = \{d_j^{'}\}_{j=1}^{|D_i|}$。然后使用BTM主题模型将扩展后的数据块特征表示K维的主题分布，记作$D_i^{''} = \{d_j^{''}\}_{j=1}^{|D_i|}$。

最终，本文的目标是训练一个动态的集成模型$f : E_{\sum D_i} \rightarrow Y$，将特征向量映射到标签上，以适应未知的短文本数据流并且发现短文本数据流中存在的概念漂移现象。

为了高效地处理短文本数据流分类问题，该方法的框架是由H个基分类器组合，构

建一个集成分类模型$E = \{f_1, f_2, \ldots\ldots\ f_h\}$。当新的数据块$d_j$来的时候，集成分类模型通过公式**??**赋予$d_j$一个类标签$y^*$。

$$y^* = argmax_{y \in Y} P(y|d, E) \tag{3.1}$$

其中后验概率$P(y|d, E)$是通过K个基分类模型进行加权平均所得，如公式3.2

$$P(y|d, E) = \sum_{i=1}^{K} w_i P(y|d, f_i) \tag{3.2}$$

图3.1是整个算法的流程：

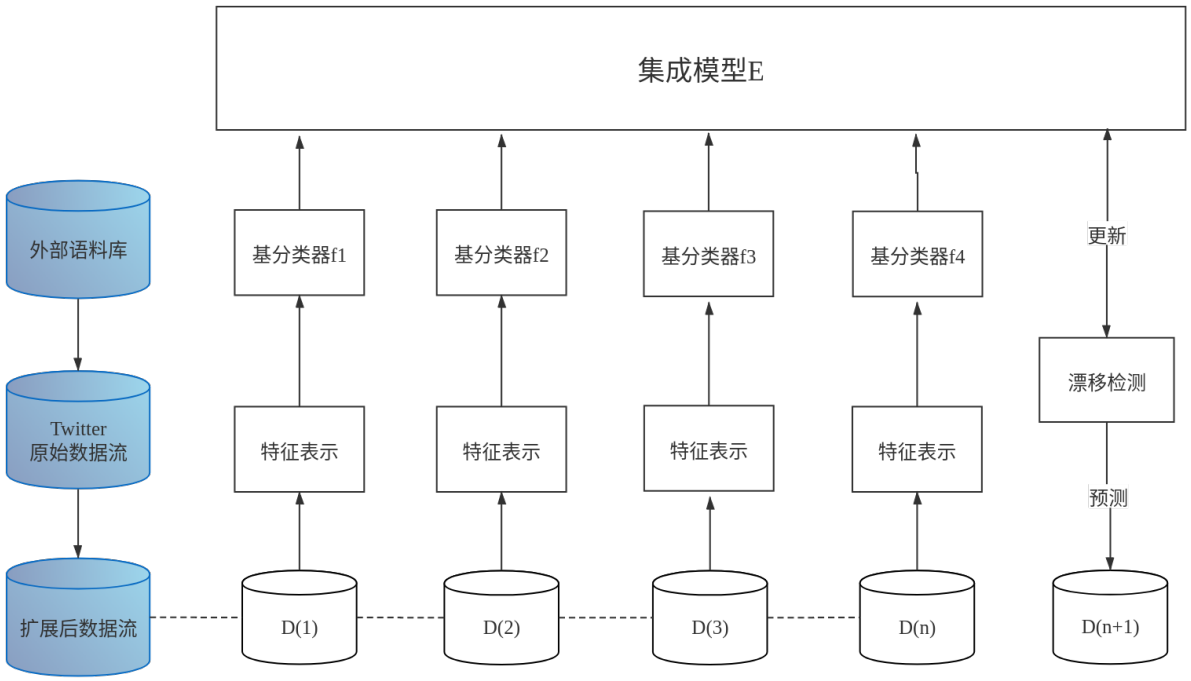

图 3.1　短文本数据流分类流程图

3.2　数据采集与预处理

本课题的实验数据来自Twitter，使用Twitter官方提供的“关键字搜索”API进行自动化采集。一共收集了从2011年11月份到2013年1月份的5个类别的共约181万条数据。图3.1是每个类别的数量，图3.2是数据实例。

表 3.1 每个类别的数量

标签	数量
arsenal	515017
blackfriday	137767
smartphone	401361
chelsea	647879
obama	112155

表 3.2 类别实例

标签	实例
arsenal	why arsenal fans always proud of their history, are they living in past?
blackfriday	BeachBody's having Black Friday deals too! Check out these great deals! http://t.co/NphfZ5Xu
smartphone	@MsEmeraldEyez ok got you posted up here: http://t.co/jzPIpstF feel special ur the 1st one :)
chelsea	@chelseaaaakay god fucking damnit Chelsea you don't get it.
obama	I love these auto-add programs, b/c I just got an e-mail saying that "Barack Obama is now following you on Twitter!" How cool is that?

为了有效挖掘数据隐藏的主题信息，需要进行数据预处理，操作步骤如下：

1. 首先将文本所有的字母变成小写，并借助正则表达式去掉文本中的email、以@开头的词、换行符、网址链接、标点符号等特殊字符；
2. 加载停用词列表，删除文本中的停用词，并对每个单词进行词干提取和词形转换；
3. 将文本字符串分词，生成词汇列表；

图3.5和图3.4是数据预处理前后对比：

3.3 短文本扩展

由于短文本数据缺乏足够的语义信息，本节借助短文本扩展技术，对缺乏短文本的数据进行语义扩展，缓解特征高维稀疏的问题。短文本扩展需要用到外部语料库，而语料

表 3.3 处理前

文本	标签
@MsEmeraldEyez ok got you posted up here: http://t.co/jzPIpstF feel special ur the 1st one :)	smartphone
Useful Smartphone Apps http://t.co/2WDnIIgg	smartphone
New post: Consumers face many more tablet choices this holiday season http://t.co/s76GLU5R	smartphone
Iphone 5 is the fastest smartphone, beat galaxy s3 and others! http://t.co/gScgoW95	smartphone
Yeah..bibinyagan ko ang bgong tablet pc... :-)	smartphone

表 3.4 处理后

文本	标签
msemeraldeyez ok got post http t co jzpipstf feel special ur 1st one	smartphone
use smartphone app http t co 2wdniigg	smartphone
new post consum face mani tablet choic holiday season http	smartphone
iphon fastest smartphon beat galaxi s3 http t co gscgow95	smartphone
yeah bibinyagan ko ang bgong tablet pc	smartphone

库的质量将对实验结果有着较大的影响。本文采用的语料库来自Wikipedia。Wikipedia是最权威的在线百科，噪音数据较少，并且内容丰富、有多种类型的短文本。借助Wikipedia官方提供的工具JwikiDocs，通过在短文本数据流中提取到的关键词进行检索，可以很方便地获取到数据。

同样，需要对这些获取到的数据进行相关数据预处理和清洗。然后，就可以开始进行短文本扩展了。首先使用LDA（Latent Dirichlet Allocation）主题模型，提取语料库中的主题信息，将得到的模型记作M_{lda}。然后，将模型M_{lda}应用到每个数据块中进行主题推断（Topic Infer）[37]，获得这些数据块中短文本的主题分布。最后再根据该主题分布将语料库中具有相同分布的词添加到短文本中，就实现了对文本的扩展。

3.4 特征表示

完成对短文本的扩展后，借助另一种主题模型BTM（Biterm Topic Model），将扩展后的短文本表示为主题分布。LDA（Latent Dirichlet Allocation）主题模型依靠的是词袋模型的假设，在对短文本进行表示时，容易得到高维稀疏的矩阵，影响算法的运算效率和精度。相比之下，BTM主题模型采用词对共现的方式进行建模，就在一定程度上避免了这个问题，因此，BTM主题模型更加适用于短文本。图3.2是BTM模型的生成过程。

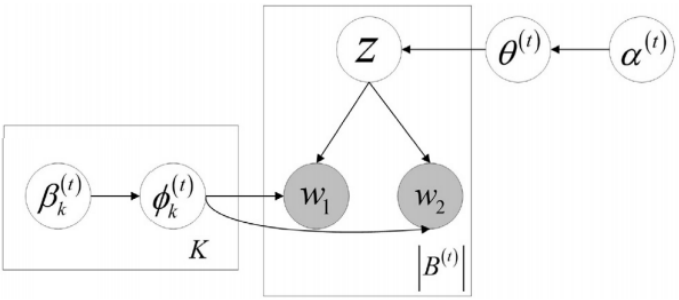

图 3.2 t^{th}时间片内的BTM主题模型

其中，K表示主题数，B^t表示由$\{w_1, w_2\}$组成的词对集合，称作一个biterm。$\alpha^{(t)}$和$\beta_k^{(t)}$分别是$\varphi^{(t)}$和$\theta^{(t)}$的Dirichlet分布的先验参数，$\theta^{(t)}$是K维的文档-主题多项式分布，$\varphi^{(t)}$是W维的主题-词分布；$Z_{m,n}$表示词语的主题分布，w_1和w_2是在该分布下取样生成的词；下面是BTM模型的生成步骤:

- Draw a topic distribution in the t^{th} time slice $\theta^{(t)}$ $Dirichlet(\alpha^{(t)})$;
- For each topic k from K topics:

  (a) Draw a word distribution of topic in the t^{th} time slice $\varphi^{(t)}$ $Dirichlet(\beta^{(t)})$
- For each biterm $b_j = w_{j,1}, w_{j,2} \in B^{(t)}$:

  (a) Draw a topic z_j $Multinomial(\theta^{(t)})$;

  (b) Draw words $w_{j,1}, w_{j,2}$ $Multinomial(\varphi_{z_j}^{(t)})$;

由于短文本是以流的形式传入模型的，本文将流数据按时间顺序进行分块，并设置好块大小。每当数据块来临的时候，即使用BTM算法对块数据进行特征表示，得到块的特征值。本文实验数据总文本数为10000，共5个类别，采用的块大小为1000，BTM主题

模型主题数为5，BTM迭代次数100次，借助python第三方库“biterm”，对BTM算法进行了调用，从而得到每个块的特征值。借助BTM模型，对扩展后的数据块$D_i^{'}$进行主题推断，将起表示为一组主题分布，即得到$D_i^{''} = \{d_j^{''}\}_{j=1}^{|D_i|}$，其中$d_j^{''} = \{z_{j,k}\}_{k=1}^{K}$,，$z_{j,k}$表示$j^{th}$短文本中$k^{th}$主题值。

表 3.5 BTM主题词提取结果

Topic 0 | Top words = black friday chelsea mobile obama arsen
Topic 1 | Top words = obama presid bush czar blackberri smartphone
Topic 2 | Top words = chelsea arsen phone intern free obama
Topic 3 | Top words = win day smartphon chelsea tablet liverpool
Topic 4 | Top words = tangan jam murah hdmi ic 20

3.5 概念漂移检测

在短文本数据流分类中，主题随着时间的推移发生会变化从而产生概念漂移。概念漂移会严重影响到模型的预测精度，因此解决该问题对本模型十分重要。本文通过主题的分布来检测检测概念漂移。

每当新的数据块来的时候，我们通过计算该数据块中的每个短文本和当前数据块的语义距离，再计算均值来判断新数据块是否发生了概念漂移，公式3.3如下：

$$dist(D_{i+1}^{''}, D_i^{''}) = 1/|D_{i+1}| \sum_{j=1}^{|D_{i+1}|} dist(d_j^{''}, D_i^{''}) \tag{3.3}$$

其中，$dist(d_j^{''}, D_i^{''})$为短文本和当前数据块之间的距离，首先根据类分布，将当前数据块分割成大小为C的簇，记作$\{I_c\}_{c=1}^{C}$，其中$I_c = \{d_i^{''}\}_{I=1}^{|I_c|}$，$|I_c|$表示$c^{th}$类簇中的短文本数，再计算短文本和所有类簇之间的语义距离，选择语义距离最小的值表示该短文本与数据块之间的语义距离，如公式3.4：

$$dist(d_j^{''}, D_i^{''}) = mindist(d_j^{''}, I_c) \tag{3.4}$$

其中，$dist(d_j'', I_c) = 1/|I_c| \sum_i^{|I_c|} dist(d_j'', d_l''), d_I'' \in I_c$。图3.3是本算法的流程图。

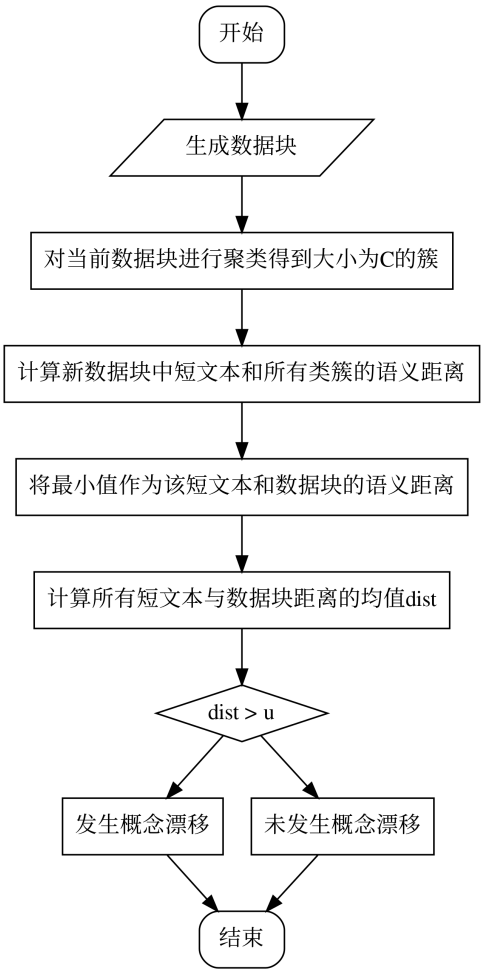

图 3.3 概念漂移检测流程图

若某个类簇中的短文本数量很少，在计算两个短文本之间的语义距离$dist(d_j'', d_l'')$时便会产生误差，由此对该项增加权重，如公式3.5:

$$dist(d_j'', d_l'') = 1 - |I_c|/|D_i''|cos(d_j'', d_l'') \tag{3.5}$$

其中，$cos(d_j'', d_l'')$为文本的余弦距离。

最后，根据阈值u去判断是否发生概念漂移。如果$dist(D_{i+1}'', D_i'') \in (u, 1]$，则认为数据块$D_{(i+1)}''$发生了概念漂移。

3.6 集成模型的构建与更新

本章是算法的核心，构建并更新集成模型，用于预测未知标签的短文本数据。选择H个上述的数据块用于构建SVM基分类器。当新的块D_e到来时，首先对它进行文本扩展和特征表示，然后使用BTM主题模型将其表示为一组主题D_e''。构建集成模型之前，要计算短文本d_j''相对基分类器的f_h的权值$w_{h,j}$:

$$W_{h,j} = (1 - dist(d_j'', D_h'')) * (1 - dist(D_e'', D_h'')) \tag{3.6}$$

其中，$1 - dist(d_j'', D_h'')$ 表示数据块 D_e'' 和短文本 d_j'' 数据块 D_h'' 的语义相似度，$1 - dist(D_e'', D_h'')$ 表示新数据块 D_e'' 和当前数据块 D_h'' 之间的语义相似度，用于减少概念漂移对准确率的影响。

下面是更新集成模型E的步骤，首先计算新数据块D_e''和集成模型中每个旧数据块之间的语义距离，并将新数据块构建一个分类器f。如果数据块D_e''相对于E中的每个分类器都发生了概念漂移，并且E中基分类器数量未满（即小于H），则将f添加到集成模型E中，如果E中基分类器数量已满，则替换E中最老的基分类器。否则，将分类器f替换E中与其语义距离最小的基分类器。

Algorithm 1 集成模型更新与构建

输入: 未到达数据块D，LDA主题模型M_{LDA}，BTM主题模型M_{BTM}

输出: 数据流S，集成模型E，概念漂移检测阈值μ

for D_e in D **do**
  利用M_{LDA}进行语义扩展，将D_e扩展为$D_e^{'}$
  利用M_{BTM}进行特征表示，将$D_e^{'}$表示为$D_e^{''}$
  for $D_h^{''}$ in S **do**
    for $d_j^{''}$ in $D_e^{''}$ **do**
      计算短文本$d_j^{''}$和旧数据块$D_h^{''}$的语义距离(公式3.4)
    end for
    计算新数据块$D_e^{''}$和旧数据块$D_h^{''}$的语义距离（公式3.3）
  end for
  for $d_j^{''}$ in $D_e^{''}$ **do**
    计算基分类器权重（公式3.6）
    使用集成模型预测新数据块中的短文本$d_j^{''}$ （公式3.1）
  end for
  计算$D_e^{''}$和S中每个块的语义距离，并根据阈值u判断是否发生了概念漂移
  使用数据块$D_e^{''}$训练新的基分类器f，并根据概念漂移的结果更新集成模型
end for

3.7 本章小结

本章主要介绍了本文核心算法的技术细节，提出一种带漂移检测的短文本分类算法，给出了算法的数学推导以及运作流程。首先说明了数据集的来源，给出了短文本数据流分类的问题定义。通过Wikipedia将短文本数据进行扩展并用BTM模型表示成主题，将数据流分块形成多个基分类器，构建集成模型，并进行了概念漂移的检测，得到一个可用的集成分类模型，用于新数据的预测。

4　算法整合与数据挖掘平台实现

本章将在上一章节提出的短文本数据流集成分类算法的基础上，设计一个用户友好、逻辑清晰、运行高效的面向社交网络数据的Web数据挖掘平台。该平台基于Django网络应用程序开发框架，整合数据挖掘算法，提供可视化的前端界面。在该平台的设计中，使用了较为流行的MVC开发模式，使得前后端分离，这样组织的代码结构清晰，易于后期维护和功能拓展，提高代码运行效率。

4.1　面向社交网络的数据挖掘平台设计

在对“面向社交网络的数据挖掘平台设计”的软件进行设计时，需按照软件工程的开发流程完成。软件设计包括对软件进行需求分析、功能设计、架构设计、API设计和服务器部署设计。软件设计是开发中的最重要一环，也是优秀开发者必须进行的工作。“面向社交网络的数据挖掘平台设计”解决的是“怎样做”的问题。开发的软件系统满足可用性和稳定性要求，还需对后续扩展维护提供便利。

4.1.1　系统设计目标

本系统核心算法使用SVM作为基分类器，将数据流进行分块，并考虑到对概念漂移的检测，构建高效可用的集成学习模型。本文事先使用Twitter API抓取推文数据，对算法进行测试，测试结果表明该算法具有良好的拟合能力，能够对文本主题实时分类和追踪。接着，在此算法的基础之上，本文构建了一个的可视化的数据挖掘平台，将文本挖掘的一般步骤UI化，使其能更好与用户进行交互，达到功能可复用、可交互的目的。本

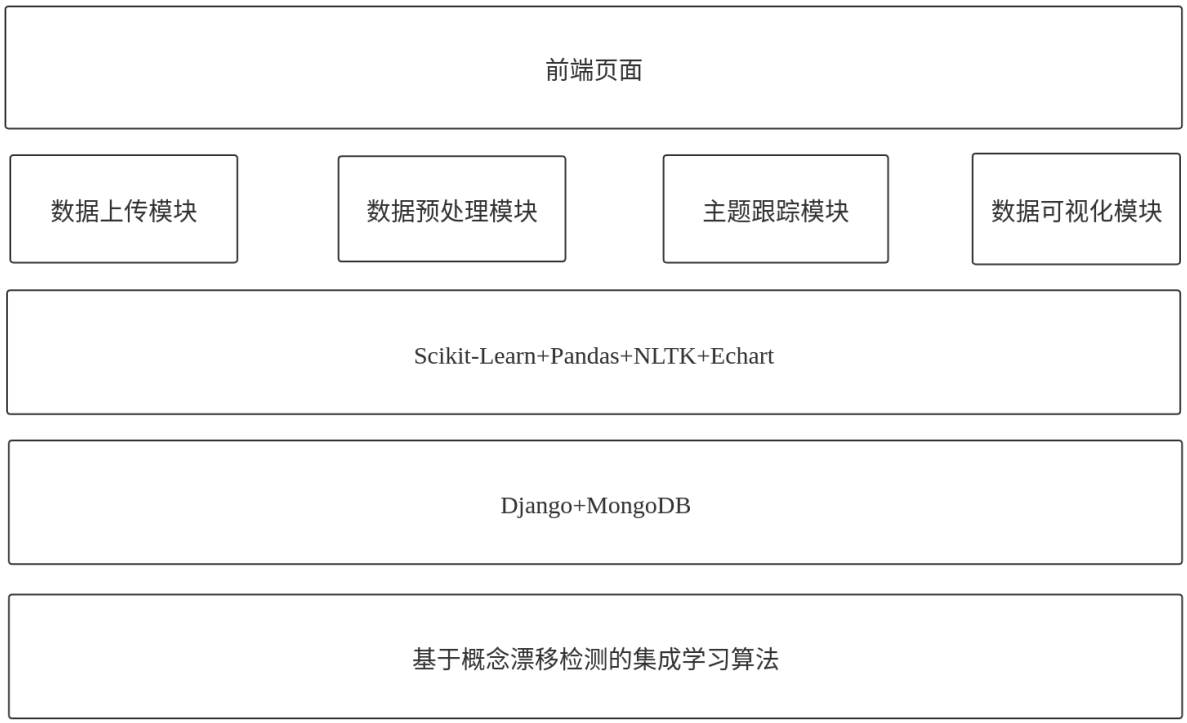

图 4.1　数据挖掘可视化平台架构图

系统的技术栈是Python+Scikit-Learn+Django+MongoDB，它集成了数据上传、数据预处理与清洗、数据建模、数据可视化等常用的数据挖掘技术，使得用户无需了解算法和数据挖掘技术内部的运行机制，即可完成对数据的分析工作。同时，该系统考虑到扩展性问题，当新的数据挖掘算法被提出时，也可以很方便地进行整合。

4.1.2　系统架构设计

根据本系统的设计目标，将其划分4个模块，分别是数据上传、数据预处理、主题跟踪和数据可视化。图4.1是本系统的整体架构图：

本文采用MongoDB作为数据库，由于着重于算法研究，本文数据库较为简单，只有数据内容、数据标签和时间等项。

4.1.3 系统功能模块

系统的功能模块主要包含四个部分，分别是：数据上传模块、数据预处理模块、主题跟踪模块和数据可视化模块。各个模块的主要功能如下：

数据上传： 用于将本地的数据文件导入到MongoDB中；

数据预处理： 负责对上传后的文档进行去停用词、词干提取、词形转换、去噪等操作；

主题跟踪： 本平台的核心模块，是集成学习的算法整合；

数据可视化： 对数据和主题跟踪结果进行了图表展示；

4.2 系统实现

4.2.1 软硬件环境

本次实验的所有模型框架均在python3.6上运行，为保证代码运行稳定、高效，基于Scikit-Learn机器学习框架进行二次开发，它的特点是可以快速实现研究人员的想法而不拘泥于模型细节。平台运行环境分本地环境和服务器端环境。本地用于算法的实现与测试，服务器端用于部署整合算法的Django服务。

表 4.1 软硬件环境

环境	硬件信息
本地环境	操作系统：Debian 10 Testing版
	CPU：Intel Core i5 5300U 2.3GHz
	RAM：8.00GB
	显卡：Intel(R)HD Graphics 630(1024MB)
	硬盘：NVMe SAMSUNG MZVLW128(128GB)+1T机械硬盘
服务器环境(阿里云计算平台)	操作系统：Debian 9 Stable版
	CPU：1.0GHz
	RAM：2.00GB
	带宽：2M

4.2.2 数据上传模块

数据上传模块主要用于上传本地文件到数据库中，用户通过在前端页面点击“文件路径”按钮，即触发文件上传控件，用户通过弹出的文件目录浏览功能访问本地文件，选中后既可获得文件路径，并会自动填充文件名到“数据集标签”中。

本地文件格式为.csv，文件名即为类标签，文件的每一行代表一条推文数据，每一列为该推文的属性，其中包括发布人TwitterID、是否转推、发布时间、推文内容等。由于本文并不需要所有属性，数据上传时就需先对需要的文本进行提取，保留发布人时间、正文内容以及类标签。图4.8是数据上传模块参数设置。

图 4.2 数据上传参数设置

设置好其他参数分割符、编码方式、文本开始位置等后，点击提交保存，前端Javascript就会发起ajax请求，django接受到请求后，url机制会匹配到对于视图模块，调用相应的方法，对数据进行保存。

```
urlpatterns = [
    re_path(r'^$', upload),
    re_path(r'uploadfile', upload_file)
]
```

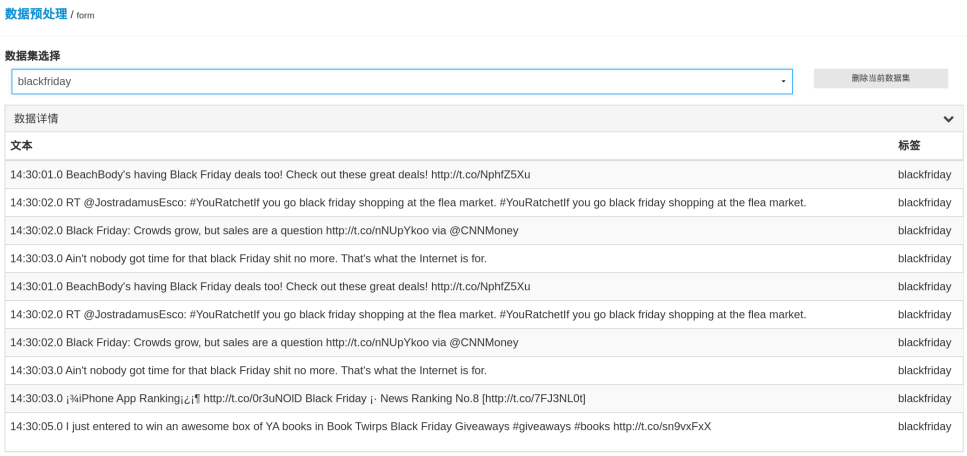

图 4.3　上传数据展示

图4.3是上传后的数据集blackfriday。

4.2.3　数据预处理模块

数据预处理模块用于负责对上传后的文档进行去停用词、词干提取、词形转换、去噪等操作。用户在前端页面选好需要处理的一类文档，选择停用词词库，询问是否进行词干提取、词形转换、大小写转换，自定义正则表达式等。图4.4为数据预处理模块的参数设置。

图 4.4　数据预处理参数设置

用户点击“提交保存”后，前端Javascript发起ajax POST请求，url路由匹配到相应的

视图并进行函数调用，对选中的文本数据集进行数据预处理。通过ORM条件查询获得表单对象，遍历该对象中的文本，借助的预处理函数完成数据预处理的一系列操作。下面是视图views.py中的接受处理请求的部分代码：

```
@csrf_exempt
def startProcess(request):
    try:
        if request.method == 'POST':
            #接受前端Post请求
            label = request.POST['label']
            ... ...
            # ORM查询符合条件的数据集
            dataset = DataModel.objects.filter(label=label)
            #进行文本预处理
            if handleProcess(dataset, stopwords, regex):
                status = "text preprocessing success!"
            else:
                status = "sorry, text preprocessing fail."
            response = {"status": status}
            return HttpResponse(json.dumps(response), content_type='application/
            ↪  json')
    except Exception as e:
        print(e)
```

4.2.4 主题跟踪模块

该模块是本系统的核心模块，整合第三章提出的集成学习算法，采用用户上传的数据模拟数据流，进行主题跟踪。用户上传数据后，调用预处理模块对文档进行文本预处理，借助外部语料库对文本进行语料扩展，并完成特征表示，得到每个文档的的文档-主题矩阵，即可放入集成模型进行训练，该模块能够实时监控训练过程，展示集成模型的日志输出情况，同时可视化地显示BTM提取到的主题分布。

用户需预设数据块大小、数据块数量以及概念漂移的阈值，同时用户可以调整BTM主题模型的参数，选择合适的主题值和迭代次数，调整SVM的参数，采用核函数类型、核函数系数Gamma以及惩罚系数C。图4.5是主题跟踪模块的参数设置。

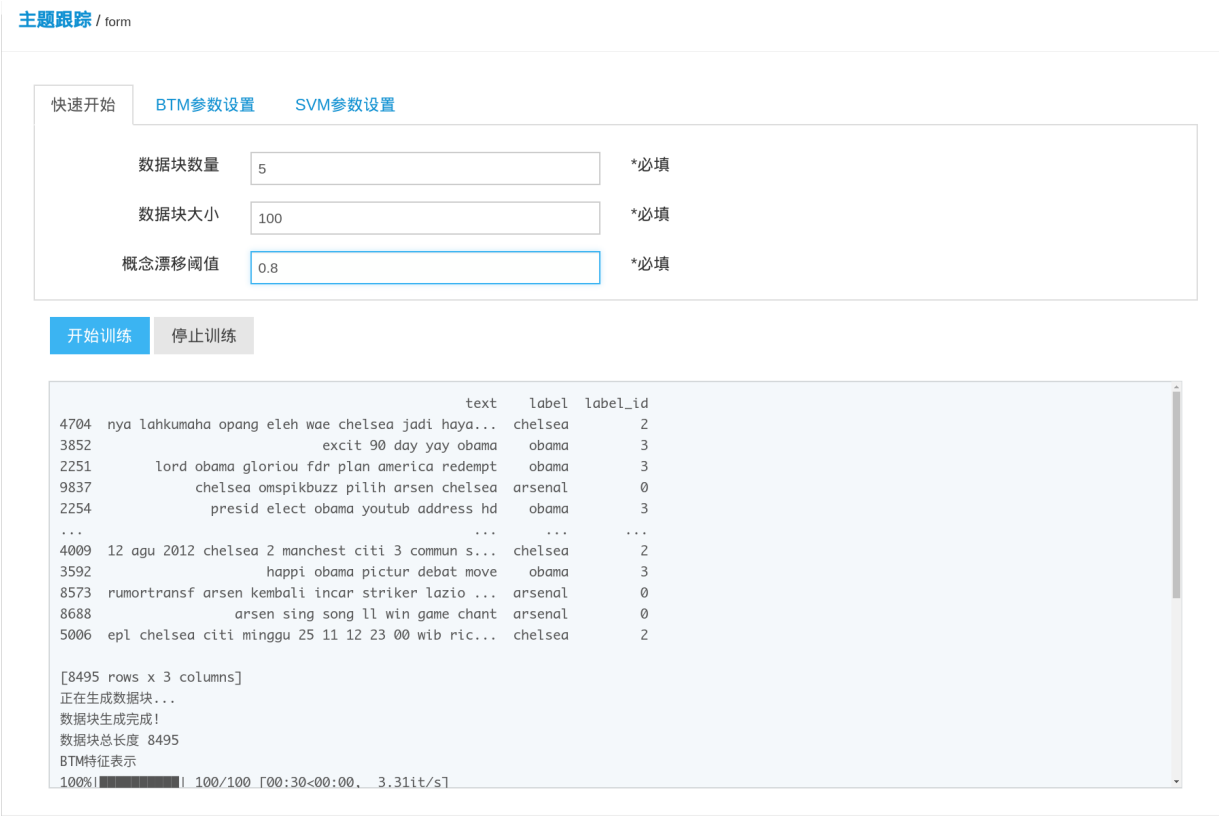

图 4.5 主题跟踪模块

用户设置好参数后，点击“开始训练”，前端即发起请求，后端URL将请求进行分发，调用topicProcess()函数进行主题跟踪，该函数会将数据库中的数据读出，放入一个Pandas的dataframe对象中并打乱，用于模拟数据流，接着数据流被分块传入集成模型中进行训练，动态更新集成模型。前端通过Websocket建立长连接，实时将训练日志输出到前端页面中，并展示主题的追踪结果。

该模块的URL路由：

```
urlpatterns = [
    re_path('^$', topic),
    re_path('topicProcess', topicProcess)
]
```

主题跟踪模块视图函数:

```
@csrf_exempt
def topicProcess(request):
    try:
        if request.method == 'POST':
            #获取前端传入参数
            H = int(request.POST['H'])
            blocksize = int(request.POST['blocksize'])
            u = float(request.POST['u'])
            ... ...
            #获取数据流
            for label in labels_:
                query = DataModel.objects.filter(label=label)
                for row in query:
                    texts.append(row.text)
                    labels.append(row.label)
            ... ...
            data.dropna(axis=0, how='any', inplace=True)
            data = shuffle(data)
            #初始化集成模型
            E = Ensemble(H = H, blocksize = blocksize, u = u,
            base = "svm", K = K, btm_iterations = iter,
            svm_gamma = "auto", svm_C = C, svm_kernel = "linear")
            #集成模型训练
            E.fit(data.text, data.label_id)
            ... ...
    except Exception as e:
        print(e)
```

4.2.5　数据可视化模块

该模块主要用于图表的展示，使用Echart模块进行图表绘制，其中包括对源数据中的文本进行词频统计，分别统计数据集中每个类簇的文本总数，生成条形图和饼状图。统计每个文档的长度，使用折线图表示各长度文档出现的频率，可以很清楚的发现，长度在40左右的文档最多，这符合社交网络中文本的特征。使用主题模型分析文本数据，通过设置主题数，对文本进行主题划分，得到主题分布图。

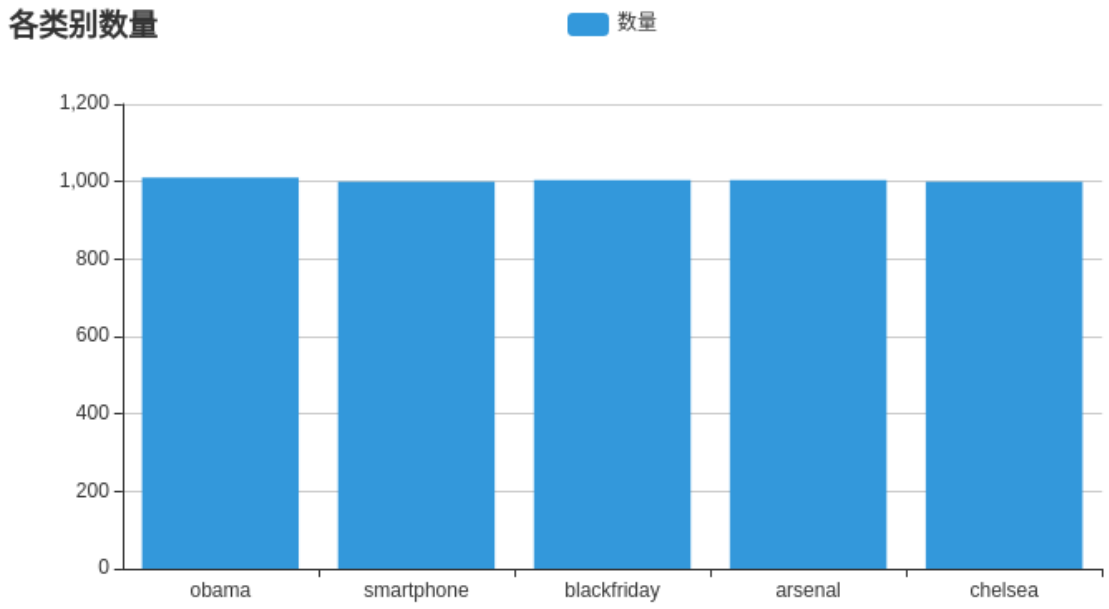

图 4.6　各类别数量图

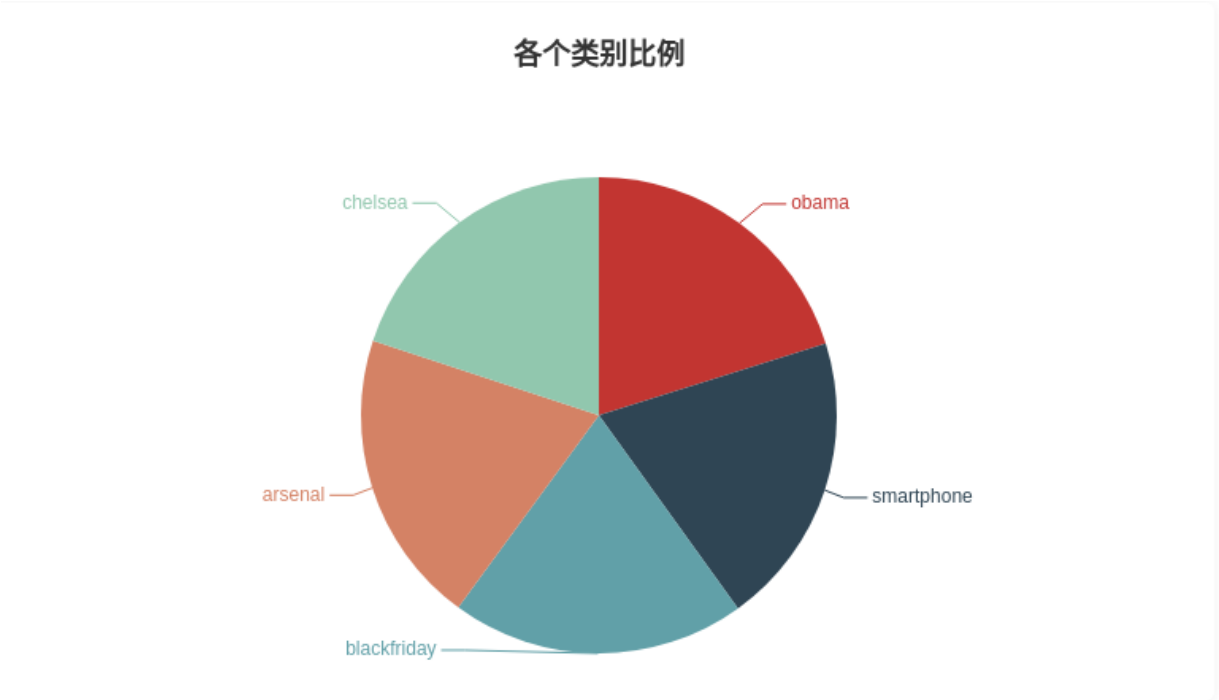

图 4.7 各类别比例图

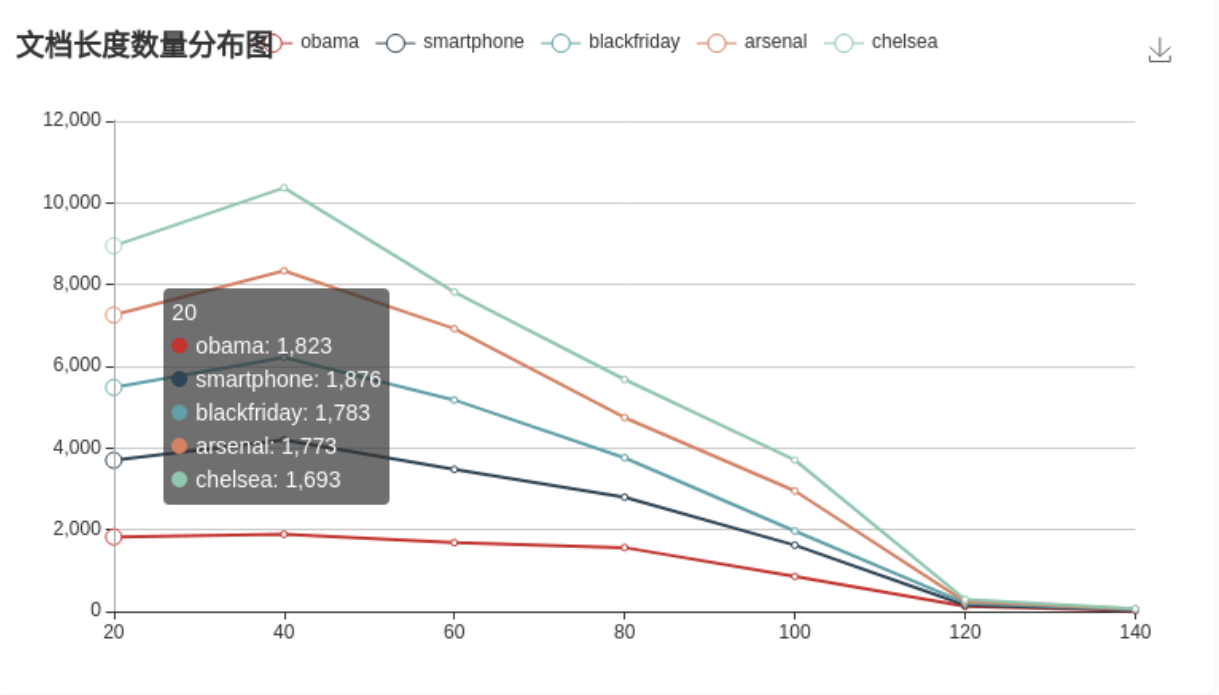

图 4.8 文档长度数量分布图

4.3 本章小结

本章主要介绍了Twitter数据挖掘可视化平台的搭建,使用的技术包括但不限于Django开发框架、Scikit-learn、pandas、numpy、echart、mongodb等。以Django框架为Web平台主体，使用Scikit-learn进行算法实现，完成了整个平台的技术整合。实现的功能模块包括：数据上传、数据预处理、数据可视化、主题跟踪等。将数据挖掘中需要使用到的如数据预处理与数据清洗、特征提取、分类算法应用等过程UI化，大大降低了使用者的门槛。同时，也使研究人员能及时、快速发现研究中的问题，具有较高的应用价值。

5　总结与展望

5.1　工作总结

当下，随着信息与通信技术的发展，各种网络平台不断壮大，信息随时产生，海量短文本数据汇入信息的洋流。这些数据蕴含丰富的信息，有很高的研究价值，因此高效挖掘这些数据的内涵变得愈发重要。短文本分类工作一直是数据挖掘领域研究的热点，它有着不用一般文本数据的特点，即语义缺失、高维稀疏等。并且，短文本数据流还会随时间迁移产生概念漂移现象。这些特点使得传统算法的对其应用的效果较差。因此，本文借助了文本扩展的方法，缓解短文本的语义信息缺乏的问题，并提出一种基于块的集成分类模型，考虑对概念漂移的检测，对短文本数据流分类进行了以下的研究：

1.本文第一章介绍了短文本数据流分类的研究背景及意义，给出短文本数据流分类面临的问题和挑战，介绍国内外研究人员对短文本分类问题研究的现状，对已存在的问题提出的解决方式。接着阐述了本文主要研究的内容，即设计一个集成分类算法和可视化的数据挖掘平台。给出论文的组织框架，分别简要介绍了每个章节的主要内容。

2.第二章为相关技术和理论的介绍，给出了文本挖掘的一般步骤，包括“数据清洗与预处理”、“分词”、“特征提取”、“特征表示”、“算法应用”等。并分别就短文本分类和短文本数据流分类介绍相关的技术。

3.第三章为本文核心，介绍了基于概念漂移检测的短文本数据流分类算法。该算法大致思想是：通过文本扩展，缓解语义信息缺失，借助时间序列对数据流进行块分割，对每个块训练基分类器构建集成模型，并通过计算块和块语义距离判断是否发生了概念漂移。文本扩展使用的Wikipedia作为外部语料，借助LDA主题模型进行主题分析，通过主

题相似性进行文本扩展。

4.第四章介绍了构建数据挖掘平台的细节，并对前面提到了基于概念漂移检测的短文本数据流分类算法进行整合。该平台基于Django搭建，采用MVT的开发模式，前后端逻辑分离，使用Scikit-learn、pandas、numpy、nltk等强大的数据挖掘与机器学习库，完成算法设计。

5.2 工作展望

进入大数据时代的今天，互联网中越来越多的如“Tweets”、“微博”、“新闻标题”等短文本数据，短文本分类问题将一直将是研究的焦点。短文本数据流带来语义信息不足、特征高维稀疏等问题，本文提出借助外部语料库扩展的方式，缓解稀疏性，实验表明具有良好的效果。同时，本文提出的数据挖掘平台当前较为简单，经过进一步研究认为，未来还有一下几个方面的工作值得进行:

- 本文通过外部语料库扩展的方式缓解文本特征高维稀疏的问题，虽然方法可行，但是扩展过程较为耗时，考虑是否可以使用更好的方式对文本空间进行扩展。
- 对于短文本数据流的分类问题，本实验采用的数据集较小，取得的效果无法证明实际应用大数据的普适性，接下来需要继续研究如何将算法应用到真实的大数据集当中。
- 数据挖掘平台的优化，文本仅应用了一种基于集成学习的分类方法到数据挖掘平台当中，在后续研究中，考虑添加更多的数据挖掘算法到平台中。同时，平台的数据需使用手动上传的方式进行模拟，未来考虑整合API或者爬虫的方式，对数据进行实时分析和预测。

总之，短文本数据流分类作为数据挖掘研究的重要方向，可以研究的内容还很多，在接下来的学习中，如何将算法真正应用到实际中，提供优雅的解决方案，始终是研究者的重要任务。

参考文献

[1] GE S, YE Y, DU X, et al. Short Text Classification: A Survey. Journal of Multimedia, 2014, 9(5): 635-643.

[2] SRIRAM B, FUHRY D, DEMIR E, et al. Short text classification in twitter to improve information filtering. in: Proceeding of International Acm Sigir Conference on Research & Development in Information Retrieval. 2010.

[3] RAKIB M R H, ZEH N, JANKOWSKA M, et al. Enhancement of Short Text Clustering by Iterative Classification., 2020.

[4] 杨波,杨文忠,殷亚博, 等.基于词向量和增量聚类的短文本聚类算法. 计算机工程与设计, 2019(10).

[5] HU X, WANG H, LI P. Online Biterm Topic Model based Short Text Stream Classification using Short Text Expansion and Concept Drifting Detection. Pattern Recognition Letters, 2018.

[6] LEE J Y, DERNONCOURT F. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks., 2016.

[7] XUEGANG H U, ZHOU P, PEIPEI L I, et al. A survey on online feature selection with streaming features. Frontiers of Computer Science, 2018, 12(3).

[8] 李太白.短文本分类中特征选择算法的研究. 重庆师范大学.

[9] XIE J, HOU Y, WANG Y, et al. Chinese text classification based on attention mechanism and feature-enhanced fusion neural network. Computing, 2020, 102(3): 683-700.

[10] VAPNIK V. An overview of statistical learning theory. IEEE Transactions on Neural Networks, 1999, 10(5): 988-999.

[11] SVETNIK V, LIAW A, TONG C, et al. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. Journal of Chemical Information and Computer Sciences, 2003, 43(6): 1947-1958.

[12] FRIEDMAN N, GEIGER D, GOLDSZMIDT M. Bayesian Network Classifiers. Machine Learning, 1997, 29(2): 131-163.

[13] YANG Y, LIU X. A re-examination of text categorization methods., 1999: 42-49.

[14] ALI M M, QASEEM M S, RAJAMANI L, et al. Improved decision tree induction: Prioritized Height Balanced tree with entropy to find hidden rules. in: Acm International Proc ” the Second International Conference on Computational Science. 2012.

[15] MENG W, LANFEN L, JING W, et al. Improving short text classification using public search engin es. in: International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making. 2013: 157-166.

[16] SUN A. Short text classification using very few words. in: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. 2012: 1145-1146.

[17] RAMAGE D, DUMAIS S, LIEBLING D. Characterizing microblogs with topic models. in: Fourth international AAAI conference on weblogs and social media. 2010.

[18] AGGARWAL C C, ZHAI C X. A Survey of Text Classification Algorithms., 2012.

[19] ZHANG Y, JIN R, ZHOU Z H. Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics, 2010, 1(1-4): 43-52.

[20] RAMOS J, et al. Using tf-idf to determine word relevance in document queries. in: Proceedings of the first instructional conference on machine learning:vol. 242. 2003: 133-142.

[21] BLEI D M, NG A Y, JORDAN M I. Latent dirichlet allocation. Journal of machine Learning research, 2003, 3(Jan): 993-1022.

[22] 陈火旺.数据流挖掘分类技术综述. 计算机研究与发展, 044(011): 1809-1815.

[23] RUSSELL S, NORVIG P. Artificial intelligence: a modern approach., 2002.

[24] VAPNIK V N, CHERVONENKIS A. A note on one class of perceptrons. Automation & Remote Control, 1964, 25(1).

[25] HSU C W, LIN C J. A Comparison of Methods for Multiclass Support Vector Machines. IEEE Transactions on Neural Networks, 2002, 13(2): 415-425.

[26] OZA N C. Online bagging and boosting. in: 2005 IEEE international conference on systems, man and cybernetics:vol. 3. 2005: 2340-2345.

[27] STREET W N, KIM Y. A streaming ensemble algorithm (SEA) for large-scale classification. in: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. 2001: 377-382.

[28] WETTSCHERECK D, AHA D. Mining concept-drifting data streams with ensemble classifiers. in: Proc. of KDD:vol. 3. 2003.

[29] KOTLER J, MALOOF M. Dynamic weighted majority: A new ensemble method for tracking concept drift. in: IEEE International Conference on Data Mining. 2003: 123-130.

[30] ZHANG P, ZHU X, GUO L. Mining data streams with labeled and unlabeled training examples. in: 2009 Ninth IEEE International Conference on Data Mining. 2009: 627-636.

[31] WIDMER G, KUBAT M. Learning in the presence of concept drift and hidden contexts. Machine learning, 1996, 23(1): 69-101.

[32] TSYMBAL A. The problem of concept drift: definitions and related work. Computer Science Department, Trinity College Dublin, 2004, 106(2): 58.

[33] KUNCHEVA L I. Classifier ensembles for changing environments. in: International Workshop on Multiple Classifier Systems. 2004: 1-15.

[34] KUNCHEVA L I. Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. in: 2nd Workshop SUEMA:vol. 2008. 2008: 5-10.

[35] MALOOF M A. The AQ methods for concept drift. in: Advances in Machine Learning I. Springer, 2010: 23-47.

[36] SHIRAKAWA M, NAKAYAMA K, HARA T, et al. Wikipedia-based semantic similarity measurements for noisy short texts using extended Naive Bayes. IEEE Transactions on Emerging Topics in Computing, 2015, 3(2): 205-219.

[37] PHAN X H, NGUYEN C T, LE D T, et al. A hidden topic-based framework toward building applications with short web documents. IEEE Transactions on Knowledge and Data Engineering, 2010, 23(7): 961-976.

[38] Mozilla. Django introduction. [EB/OL]. `https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction` Nov 30, 2019.

[39] Beginnersbook. RDBMS vs MongoDB. [EB/OL]. `https://beginnersbook.com/2017/09/mapping-relational-databases-to-mongodb/`.

指导教师简介

致　谢

完成本篇论文，意味着我的本科学习生涯就快要结束了。借此机会感谢一路上帮助、关心过我的老师、同学和我的家人们。大数据与智能工程学院是我心中的霍格沃茨学院，四年里，每每我更加深刻地理解计算机时，它总会让我感到犹如魔法般绚丽。正是那些理性的光辉，一次次令我着迷，让我对这门学科的兴趣被一次次激活。

首先，由衷感谢我的毕业论文指导老师张雁教授以及合肥工业大学李培培教授，张老师学识渊博、为人正直而且平易近人，深得学生们的爱戴与尊敬。在我完成我的毕业论文的过程中，从开题到中期报告再到论文撰写，张老师都给予了我很大的帮助，即便疫情期间没能去到学校，张老师仍通过网络不断监督、指导着我们的学习，我们作为学生被张老师认真负责、严谨治学的态度深深打动，时刻提醒自己今后一定要学会这种优秀的品质，做一个对社会、对国家有价值的科研人。李老师是我未来研究生生涯的指导老师，她给了我本论文的命题和研究方向，在前期学习、编码以及论文撰写过程中提出了许多宝贵的建议，也让我在短短几个月时间里对未来研究内容有了大致的了解，为研究生生涯提前做好准备。

其次，感谢四年本科生涯帮助过我的其他老师们以及同学们。王晓林教授是我来到大数据与智能工程学院遇到的第一位老师，我始终记得那个昆明独有的沁人心脾的早晨，我走进教室，终于开始了我计算机学习生涯的第一堂课，王老师轻松幽默却又深刻的授业方式让我终身难忘，他的出现点燃我对这个学科的热情，让我接下来能够一步一个脚印完成本科阶段的学习。当然，大数据与智能工程学院的其他各位老师也都是我学习路上的明灯，与他们的交流过程中，我学习到的不仅仅是知识，还有如何思考问题的方式，这对我来说无疑是巨大的收获。计算机科学与技术专业的每位同学，都有值得学习的地

方，他们当中不乏有十分聪明又努力的，也有喜欢不断追求新事物、新挑战的，他们对我的影响足以改变我的人生。

同时，感谢我的家人，因为只有他们作为我强大的后盾，让我在生活上没有后顾之忧，我才能专心下来学习。在我遇到困难时，也是他们第一个站出来，给我支持和力量，他们是我不断前行的人生路中最大的动力。

最后，感谢美国计算机教授高德纳（Donald Ervin Knuth）编写的功能强大的排版软件TEX。感谢美国计算机科学家莱斯利·兰波特（Leslie Lamport）教授为TEX开发的简单易用的LATEX宏包。感谢王老师编写的优秀的LATEX模板。让在撰写毕业论文时有着非同寻常的体验，基本无需考虑繁杂的排版与格式问题，真正专注到书写论文本身，才能更好完成这篇论文。