# 摘 要

微博立场分析是自然语言领域中一个新兴的研究热点，本质上属于短文本分类问题。

本文研究了小数据集上的短文本立场分析的监督学习方法。考虑到使用不同分类器集成能够克服分类器各自的缺点，提升分类效果，本文首先设计了两个集成算法框架，分别使用相同和不同的分类器进行集成。

随后本文研究了 BOW 模型下文本语义的表示，提出了微博文本的层次主题模型假设，并使用了潜在语义索引(LSI，又称 LSA)，隐含狄利克雷分布(LDA)，情感词词典特征(SL)进行支持向量机分类实验。之后本文提出了一种基于文本相似度计算的分组投票模型(GVM)，并基于层次主题模型假设分析了 LDA 相比于 LSA 表现较差的原因，推断了使用 GVM 时 LDA 同样将表现较差；本文在 GVM 上的实验结果证实了该推断的正确性，也证明了层次主题模型假设的合理性。同时，实验结果也证明了使用 LSA 特征的 GVM 方法（LSA-GVM）有着参数较少，训练较为简单，运行效率较高，分类较为准确等优点。

最后，基于前面进行实验的各个基分类器的实验结果，本文在同质集成框架中采取了 LSA-GVM 作为基分类器，并提出了一种针对 LSA-GVM 的快速集成方法（称为 Fast Ensemble LSA-GVM）。实验证明，Fast Ensemble LSA-GVM 进一步解决了 LSA-GVM 的参数选择问题，使得参数选择更为简单，平均分类效果进一步提升。本文在非同质集成框架中，选择了 LSA-SVM，LDA-SVM，LSA-GVM 作为基分类器，并在实验中证实了集成学习确实在能够带来分类效果的提升。

本文的最主要贡献是提出了 LSA-GVM 方法和 Fast Ensemble LSA-GVM 方法。其中本文使用 Fast Ensemble LSA-GVM 方法在 NLPCC 2016 Shared task 4 发布的评测数据集上的实验结果与参评队伍比较，仅次于第一名的队伍。但因为第一名的队伍对评测数据集中每个“目标”（“target”）设计了不同的分类模型，所以其方法在实际应用中有着很大的局限性，而本文的 Fast Ensemble LSA-GVM 则是一种可以应用于不同“目标”的简单高效的分类模型，有着广泛的应用场景和很高的应用价值。

**关键词**：立场分析；集成学习；微博；短文本分类

# Abstract

Chinese Microblog Stance Analysis is a new research hotspot in the field of natural language processing, which belongs to the classification of short text.

In this paper, we study the supervised learning method of short text stance analysis on small data set.Considering the use of different classifiers for integration can be help for overcoming the shortcomings of the classifier and improving the classification effect, this paper first designed two ensemble algorithm frameworks, using the same and different classifier for integration respectively.

Then, this paper studies the representation of the text semantics under the BOW model, and puts forward the hypothesis of the hierarchical theme model of the microblogging text, and uses the Latent Semantic Index (LSI, also known as LSA), the Latent Dirichlet distribution (LDA), the Sentiment Lexicon(SL) Feature for classification experiments,which use support vector machine as classification method. Then, this paper proposes a group voting model (GVM) based on the text similarity calculation. Based on the hierarchical thematic model, the reason why LDA performed poorly than LSA while using SVM as classification method is analysed, and it is concluded that the LDA will perform poorly when the LDA is compared with the LSA under the GVM. In this paper, the experimental results of GVM confirmed the correctness of the inference, and proved the rationality of the hypothesis of the hierarchical topic model. Further more, the experimental results also prove that the LSA-GVM method has the advantages of having less parameters, only simply training needed, higher operation efficiency and more accurate in classification tasks.

Finally, based on the experimental results of each base classifier, the LSA-GVM is adopted as the base classifier in the homogeneous ensemble framework, and a fast ensemble method for LSA-GVM is proposed (called Fast Ensemble LSA-GVM). Experiments show that Fast Ensemble LSA-GVM further solves the

problem of parameter selection of LSA-GVM. And it makes the selection of parameters more simply and the average classification effect is further improved. In this paper, LSA-SVM, LDA-SVM and LSA-GVM are selected as base classifiers in the non-homogeneous ensemble framework, and it is proved that the ensemble learning can improve the classification effect by the experiment.

The main contribution of this paper is to propose the LSA-GVM method and the Fast Ensemble LSA-GVM method.In this paper, the experimental result on the published evaluation data set of NLPCC 2016 Shared task 4, which using the Fast Ensemble LSA-GVM as classification method, can be rank second place compared with the participating teams. However, because the first team uses different classification models for each "target" in the evaluation data set, the method has great limitations in practical application, and the Fast Ensemble LSA-GVM is a simple and efficient classification model which can be applied to different "target", which has a wide range of application scenarios and high application value.

**Keywords**: Stance Analyse； Ensemble Learning； Weibo； Short text classification

# 目 录

# 第1章　绪 论

## 1.1 研究背景

随着移动互联网的飞速发展以及移动电子设备的发展与普及，人们的生活越来越离不开网络；人们逐渐习惯并乐于通过网络开展社交，发表自己的观点，社交媒体在人们的生活中变得愈加重要。如今，大量的网民喜欢在微博，推特等平台上对日常的热点事件发表自己的观点，使得网络中聚集了大量的人们对于热点事件的意见。因此，如何提取和分析人们的这些观点变成了商业团体和学术机构都密切关注的研究焦点。

提取和分析文本中观点的问题，大致可以分为以下几类：情感分析，意见挖掘和立场分析，它们都可以看做是文本挖掘的子任务。其中情感分析着重于分析说话者的情绪，是高兴还是愤怒，是积极还是消极；意见挖掘着重于分析说话者对于某事或某物的意见，通常要识别出其话语中的“关键概念”，并分析说话者对于这个“关键概念”的看法；立场分析则是分析说话者对于给定“目标”或者说是“主题”的立场，包括 FAVOR（支持），AGAINST（反对）和 NONE（中立或不明）。以用户对于 2017 年发布的“小米 6”手机的评价为例，情感分析在分析用户的评价时，分析的是用户表达了一种积极还是消极的情感；而意见挖掘是分析用户所表达的对于手机各方面的看法，要先识别用户评价中提到的手机的属性，然后分析用户对于这些属性的看法，比如用户认为续航如何，拍照功能如何，价格如何等；立场分析则是分析用户对于“小米 6”这款产品的立场，更关心用户是不是支持这款产品。

从上面的例子可以看出，意见挖掘更像是情感分析的细化和升级，而立场分析要总结出说话者的综合观点，并且有着明显的“目标/主题”倾向性，例如如果说话者比较了“小米 6”和“华为 p10”两款手机产品并进行了评价，他对“小米 6”和“华为 p10”的立场可能大不相同，因此立场分析必须要考虑所面向的“目标/主题”（下文也称为 target）。这个“target”往往是人为给定的，而意见挖掘则往往需要自动地识别一些实体和属性。

立场分析区别于情感分析和意见挖掘，在许多场合下有着重要而独到的应用。有时，我们更关心或只关心作者对于“target”的态度，这个“target”

可以是一件产品，一个社会现象，一条政府出台的新政策，一个传统习俗，等等。以 2016 年美国大选为例，大选的进行过程中民众在网络上发表了许多自己的观点，对于美国大选，我们更关心的是人们是不是支持希拉里，而不是希拉里邮件门事件曝出后民众是高兴还是愤怒，支持或不支持她的人其情感都有可能是愤怒的。

微博作为我国最为流行的一种通过关注机制分享简短实时信息的广播式社交网络平台，拥有海量的用户和海量的微博数据，内容涵盖广泛。随着微博用户量的逐渐增大，微博立场分析应运而生，并且很快变成了新兴的热点研究课题。

## 1.2 研究意义

由于微博数据的广泛性和海量性，微博立场分析将有助于公共情感监测，社会管理，商业决策等多方面的工作，有着广泛而重要的现实意义，也有着很大的商业价值。

微博立场分析属于立场分析工作，本质上是短文本分类问题，其研究工作对于不同类型的短文本分类的研究有着借鉴和指导意义。其应用的相关理论和技术在信息检索，文本摘要，文本蕴含等方面同样有着广泛应用[1]。

因为微博的海量和广泛性，使得其很有研究价值，成为了备受关注的研究热点，因此研究微博立场分析除了其本身的应用价值之外，它对微博其他相关研究的促进作用也有着重要的价值。此外，由于微博文本的简短，非正式，口语化，多网络用语，多拼写错误等原因，为立场分析带来了新的机遇和挑战，因而有着较大的研究意义。

## 1.3 研究现状

立场分析早期起源于对辩论或者在线对话的研究，托马斯等人提出了一种利用基于说话者身份的约束和语句之间的直接文本引用的国会辩论立场检测的新框架[2]。Walker 等首先在立场分类中应用 Max-Cut，从建模对话关系中可以看出，性能提升很大[3]。Murakami 和 Raymond 在提出了一种方法，使用了辩论中用户评论的本地信息[4]。Sridhar 等设计了一种新的集成分类方法来检测立场，使用了结构化特征和语法特征，并且捕获了帖子和用户之间的潜在关系

[5]。

早期多数关于立场分析的文献都是针对辩论和在线对话的，这些问题都有足够的上下文来暗示作者的语义，也可能有作者的个人资料等额外信息可以辅助推断。而在微博立场分析任务中，要分析的短文本既没有上下文，我们也无法利用作者的其他信息来进行推断。

随着微博，Twitter 等社交平台和社交方式的发展，针对微博这类短文本的立场分析逐渐受到人们关注。Liu L 等使用监督学习和半监督学习的框架，设计了不同的非均衡数据集处理策略和分类器来完成立场分析任务，例如支持向量机，朴素贝叶斯和随机森林[6]。Qingying Sun 等探索了词典特征，形态学特征，语义特征和语法特征四种特征在立场分析中的应用[7]。Jiaming Xu 等提出了一个集成学习的框架，使用多种特征和分类方法，以及 top-k 和 leave out-k 特征选择策略来处理立场分析问题[8]。Prashanth Vijayaghavan 等使用了字符级和单词级的卷积神经网络来实现立场检测[9]。Nan Yu[10]等使用 Bi-LSTM 来进行立场分析，但其测试结果并不理想，效果明显低于传统的机器学习方法，可能是由于训练集数据量较小以及分词错误对 word embedding 层影响较大导致。Ruifeng Xu 等在[1]中，对 NLPCC 2016 shared task 4，即微博立场检测任务做了总结，其中提到取得测评第一名的队伍针对每个“target”设计了不同的分类模型，这应该是使得系统总体性能提升的重要原因。

## 1.4 问题陈述

### 1.4.1 立场的定义

参考 Ruifeng Xu 等在[1]中的描述，本文将立场分为了三类：FAVOR，AGAINST，NONE。其中 FAVOR 表示作者支持给定的主题（直接地或通过支持或反对其他人来间接地表达），AGAINST 表示作者反对给定的主题（直接地或通过支持或反对其他人来间接地表达），NONE 表示不是以上两种，可能是作者表达了一种中立的立场，或者是文本中没有任何线索可以表明作者的立场。

FAVOR 和 AGAINST 的意义很明确，但 NONE 包括的含义可以有两种理解方法：一种是狭义的 NONE，认为微博文本一定是和给定的主题相关的，表达了一种中立的立场或者是看不出立场；另一种是广义的 NONE，即除了上述两种情

况外还可能是文本根本和给定的主题无关，作者没有在谈论给定的主题，例如文本为“有关部门在下决定的时候能不能考虑周全一点，全面禁摩，禁电动车，快递的电动三轮也禁止上路，直到现在半夜了，还在路上查，导致快递员不敢开车上路，快件全部滞留，堆积如山。”，主题为“春节放鞭炮”。

这两种不同的理解可能使得立场分析的实现方法上有很大不同。如果按照广义的理解，在立场分析方法中就必须考虑到“主题”，应该考虑文本是不是和主题无关的，而在实际数据中很多和主题相关的文本并没有显示地提到主题，且在缺乏上下文语境的情况下，许多含蓄的表达将有多义问题，看起来和某几个“主题”有关都可以，甚至不同的人会有不同的理解。因此广义的理解方式将会比狭义的理解方式处理起来更为复杂，甚至不好评判。

在实际应用场景中，我们获取要分析的数据时不会故意混淆进无关数据，而是会力争保证文本和主题都是相关的。在默认主题和数据相关的情况下，对于少量和主题无关的实际文本数据（例如微博，百度贴吧回复中刷经验的现象等），我们基于一些语义分析的方法，也能较大概率地把它们分类到 NONE 类别，不会使得分类效果下降太多。因此，本文综合考虑了实用性和实现复杂性的问题，默认数据中的文本与主题是相关的。

### 1.4.2 评测数据集简介

本文使用了 NLPCC 2016 shared task 4 提供的微博立场分析数据集作为评测数据集。该数据集包含五个主题：#春节放鞭炮，#IphoneSE，#俄罗斯在叙利亚的反恐行动，#开放二胎以及#深圳禁摩限电。其中每个主题有约 600 条标注的训练数据，600 条未标注的训练数据，以及 200 条标注的测试数据。

数据的格式如下：

<ID><Tab><Target><Tab><Text><Tab><Stance>

其中 ID 是数据的编号，从 1 开始，Target 指主题，Text 是用户所发微博的全部内容，包括了纯文本，表情，地址信息，原始来源等。

该数据集的详细统计信息请见表 5-1。

### 1.4.3 分类结果的评价方法

对于每个 target，使用 FAVOR 和 AGAINST 两个类别的 F 值的宏观平均值

来评价实验结果。如式(1-1)所示：

$$F_{AVG} = \frac{F_{FAVOR} + F_{AGAINST}}{2} \tag{1-1}$$

其中$F_{FAVOR}$和$F_{AGAINST}$的定义如式(1-2)和式(1-3)所示：

$$F_{FAVOR} = \frac{2 \times P_{FAVOR} \times R_{FAVOR}}{P_{FAVOR} + R_{FAVOR}} \tag{1-2}$$

$$F_{AGAINST} = \frac{2 \times P_{AGAINST} \times R_{AGAINST}}{P_{AGAINST} + R_{AGAINST}} \tag{1-3}$$

P 和 R 分别代表准确率和召回率。

本文使用所有 target 的$F_{AVG}$的平均值来作为整体评价指标。

## 1.5 研究内容

本文研究的内容主要包括：1.集成学习方法的理论与应用；2.文本的表示及文本的语义特征；3.主题模型在短文本分类中的应用；4.支持向量机在文本分类中的应用 5.短文本稀疏性问题的解决方法；6.基于相似度度量的监督学习分类方法。

## 1.6 本文的组织结构

第 1 章主要概述了本文的研究背景、研究意义以及研究现状；并对本文要研究的问题进行了定义，划清了问题边界；最后阐述了本文的研究内容。

第 2 章介绍了集成学习理论以及本文所采用的集成算法框架。

第 3 章阐述了本文的通用数据预处理工作。

第 4 章介绍了本文所采用的特征，包括特征的基本原理，特点，以及这些特征在本文中的应用。

第 5 章阐述了特征集在支持向量机上的实验结果，并作出了结果分析和方法评价。

第 6 章提出了一种基于相似度计算的分组投票模型(Group Voting Model，简称 GVM)，并使用 LSA 和 LDA 特征进行了实验，给出了实验结果并作出了分析和评价。

第 7 章介绍了本文在第 2 章中提出的两个集成算法框架的内部细节，并

使用评测数据集进行了实验，与使用相同评测数据集的工作进行了比较，分析了实验结果，对本文的集成算法框架做出了评价。

第 8 章对于本文的工作作出了总结和展望。

# 第2章　集成学习框架

集成学习是基于群策群力的基本思想而提出的。它是通过集成多个基学习器来共同决策的机器学习技术,通过构造有差异的基学习器,得到的集成学习器可以有效地提高学习效果[11]。考虑到单一的特征和分类方法难以很好地对微博文本语义进行建模,本文采用了集成学习的算法框架以期提高分类的整体效果，减少单一模型的语义鸿沟所带来的影响。

## 2.1 集成学习理论介绍

集成学习之所以能提升学习效果，与基学习器的“多样性”密不可分。

我们可以先用一个简单的例子来初步理解一下集成学习,假设我们想预测一支股票在一段时间内的涨幅会不会超过 6%，有三个团队可以帮助你做预测工作，根据历史经验，每个团队的预测准确率均为 60%，如果三支团队的预测结果是相互独立的,那么三支团队平权投票的预测准确率为：$C_3^3 0.6^3 + C_3^1(0.6^2 \times 0.4) = 0.648$，有了明显的提升。当然这个例子的假设有些极端，显然实际情况下，三个团队的预测不会是相互独立的，因为三个团队可能在预测方法，数据源方面相同或相似，从而导致他们的预测结果有着较大相关性。从这个简单的例子我们可以推断，如果多个学习器的结果能够有着较弱的相关性，那么集成学习就可以为我们带来效果的提升,这就是为什么集成学习要求基学习器具有“多样性”，或者也可以说是“差异性”,“弱相关性”。

对于二分类问题，当准确率低于 50%时，容易证明，集成的效果反而会下降。由此可见，集成学习要想取得较好的效果，各个基学习器需要一定的准确性（或者说是学习效果）和一定的差异性。

在集成学习中，平权投票是一个简单易行但并不是十分有效的策略。我们将上面的例子稍稍更改一下：团队 1 的预测准确率有 80%，团队 2 的预测准确率为 70%,团队 3 的预测准确为 60%,如果三支团队的预测结果是相互独立的，那么三支团队平权投票的预测准确率为：$0.8 \times 0.7 \times 0.6 + 0.2 \times 0.7 \times 0.6 + 0.8 \times 0.3 \times 0.6 + 0.8 \times 0.7 \times 0.4 = 0.788$，可以看到，集成之后的准确率比最好团队的准确率要低了一些。我们需要将投票权重更改一下，使得各个团队的预测准确率和其投票权重是正相关的，这样的加权投票的策略就可以改变上面的情况，

使得集成之后效果确实提升了。不过，平权投票策略集成之后的准确率仅仅略低于最高团队的准确率（考虑到非独立性因素，实际效果可能会再低一点，但不会低于准确率的平均值，参见下面的 error-ambiguity decomposition 介绍），因此，这种策略可以用在一些参数难以学习（如学习出一个好的参数的复杂度过大）且参数难以凭经验推断的模型之中，我们可以采用这样一个折中的方法：通过在合理区间内选取多个参数进行平权投票，以期取得较接近最好参数组的实验结果，解决参数选取困难的问题。本文在第七章就采用了这种策略，详见 7.1。

关于集成学习差异性和准确性的一些严密分析论证可以参考[12-13]。在[12]中，Krogh 和 Vedelsby 提出了 error-ambiguity decomposition(误差-分歧分解)：

$$E = \bar{E} - \bar{A} \tag{2-1}$$

$$\bar{E} = \sum_{i=1}^{T} w_i \times E_i \tag{2-2}$$

$$\bar{A} = \sum_{i=1}^{T} w_i \times A_i \tag{2-3}$$

其中 E 表示集成后模型的泛化误差。$\bar{E}$表示个体学习器范化误差的加权平均值。$\bar{A}$表示基学习器的加权分歧项,即基学习器差异性度量的加权平均。这个式子表明基学习器准确性越高,多样性越大,则集成效果越好;同时也证明了集成后的泛化误差始终小于基学习器泛化误差的加权平均值,为集成的可行性和有效性奠定了理论基础。

要想集成后的泛化误差小于基学习器中泛化误差的最小值,即集成后学习器的效果优于所有单一基学习器的效果，需要在提高$\bar{A}$上下功夫。

## 2.2 集成学习的常用方法

2.1 中已经说明，要想提高集成学习的效果，必须努力提高$\bar{A}$，即基学习器的差异性。学习模型的差异性来源主要有 4 种：研究总体不同，模型假设不同，建模技术不同，初始参数不同。针对这四种差异性来源，我们可以提出四种针对性的增强基学习器之间差异性的思路：从原始训练集中构造多个差异性

的训练集;使用不同的特征来提取原始数据的属性;使用不同的学习技术建模;使用不同的模型参数。

按照基学习器是否相同可以将集成学习分为同质集成和非同质集成。同质集成即各个基学习器的建模方法相同，只是输入的训练数据不同；非同质集成即将利用了不同建模手段的基学习器集成起来。

同质集成对于原始训练数据的处理手段主要有套袋（Bagging）和提升（Boosting）。套袋技术指在总体样本中通过多次有放回的采样生成多个数据集,然后用它们分别来进行训练得到多个学习器,最终取其预测结果的(加权）平均值作为输出，这有助于减少集成模型的方差错误。提升是一种迭代算法，它能根据上一次分类的预测情况来调整观测值的权重,改变样本的分布而非重新采样。如果一个观测值的分类被预测错误，那么该算法将会增加这个观测值的权重，否则反之。通常情况下，提升能有效减少集成模型的偏误同时降低过拟合的风险（按照统计学的观点，一般化模型的误差为模型方差，模型偏误的平方，噪声三者之和)。

## 2.3 本文的集成学习算法框架

为了探索集成学习在立场分析任务中的应用,本文设计了两个集成算法框架。第一个框架是同质集成框架；结合第 5,6 章的实验结果，本文选择了 LSA-GVM 作为基分类器，采取了平权投票的策略，以期解决单一基分类器的参数选择较难的问题，同时提高分类的平均效果。第二个框架是非同质集成框架；结合第 5,6 章的实验结果，本文选择了 LDA-SVM，LSA-SVM，LSA-GVM 作为基分类器，采用了加权投票的的策略，以期得到比各个基分类器更好的分类效果。

本文将在第七章详细介绍两个集成算法框架是如何对基分类器进行集成的，并给出实验结果，对本文的两个集成算法做出评价。

## 2.4 本章小结

本章首先介绍了集成原理的基本思想以及本文采用集成算法框架的原因。之后在 2.1 节介绍了集成学习的基础理论，包括集成学习为何能够有效，集成学习对基分类器的准确率和差异性的要求,平权投票策略的优劣及合适的应用场景，Krogh 和 Vedelsby 提出的误差-分歧分解理论。在 2.2 节介绍了集成学

习的常用方法，包括从四个角度增加模型差异性的理论方法以及两种具体的通过原始训练集构造多个多样性训练集的算法：套袋和提升（Bagging and Boosting）。最后在 2.3 节中介绍了本文的集成学习算法框架。

# 第3章　通用数据预处理工作

本文所处理的数据为微博原始数据，包括用户所发微博的全部文本，表情，@标签，地址信息，链接等内容。原始数据中有许多冗余信息，会为立场分析引入噪声，因此需要先进行数据预处理。本章描述了本文中的通用数据预处理工作，对于提取部分特征时要做的一些专门的数据预处理工作，将在第 4 章讲述。

## 3.1 数据预处理的内容

本章的数据预处理工作是为了去除原始数据中几乎不含作者立场信息的冗余部分。基于这个目的，本文对原始文本采取了：去除 HashTag；去除 url；去除分享标识；去除@标识；全角转半角；字母转小写；将文中的多个连续空格转换为一个空格；去除特殊字符的预处理方法；并在此基础上进行分词得到最终的预处理结果。

## 3.2 数据预处理的流程

1. 使用正则表达式去除 Hashtag；
2. 基于 1 的结果，使用正则表达式去除 urls；
3. 基于 2 的结果，使用正则表达式去除分享标识；
4. 基于 3 的结果，使用正则表达式去除@标识；
5. 基于 4 的结果，将全角字符转为半角字符；
6. 基于 5 的结果，将字母全部转换为小写；
7. 基于 6 的结果，将多个连续空格转换为单个空格；
8. 基于 7 的结果，去除文中的标点，罗马数字，特殊字符等，保留中英文和数字；
9. 基于 8 的结果，使用 thulac[14]对原始语料进行分词。分词输出为单词流，词语之间使用空格间隔。分词得到的结果即为最终的预处理结果。

## 3.3 本章小结

本章首先阐述了为何要对原始数据进行预处理，以及什么是本文的“通用

数据预处理”。之后在 3.1 节介绍了数据预处理所包含的内容。最后在 3.2 节介绍了本文的通用数据预处理流程。

# 第4章　特征工程

文本的语义表示是本文工作中的一个重要问题。我们要用一种便于处理的数据结构来表示离散化的文本，同时为文本和这种数据结构之间建立一个合理的映射，以保证分类的效果。这种数据结构和映射关系构成了完整的特征。本文采用向量作为文本语义特征的数据结构。本文假定，每个文档包含多种离散化且有限的语义，每种语义都有一定的强度，可以用一个实数值来表示。两个文档在同一语义上强度（数值）越接近，表示它们的该语义越接近。设文档集合 D 的语义集合为 S（假设集合是有序的），那么 D 中的第 j 个文档$D_j$（下文也称$w_j$）的语义就可以表示为一个|S|维向量$V_j$。其中，$V_j$的第 i 个元素记作$V_j[i]$，$V_j[i]$的值即为$S_i$在$D_j$中的强度值。由语义向量构成的|S|维向量空间就是语义空间。

语义空间的维度|S|通常是我们指定的，它对于分类的表现有一定的影响，但不是至关重要的，语义空间的生成过程（即文本到语义向量的映射）才是影响分类表现的最重要因素。我们认为特征是很难完全拟合真实语义的，语义空间的生成过程越符合自然语言的特点，特征就越会接近文本的真实语义，在使用相同分类方法的情况下，文本分类效果就会越好。特征所表达的语义和真实语义之间的差距我们称之为语义鸿沟，语义鸿沟越大，文本的分类效果的上界越低。当语义空间的每个维度不具有完全的可解释性时，我们称之为隐语义空间。

我们将选取合适的特征的过程，称为特征工程。为了保证良好的分类效果，特征工程至关重要，我们需要将原始文本映射到一个保留立场语义信息并且含有较少噪声的语义空间，同时考虑尽量减小语义鸿沟。同时为了构造用于集成学习的差异性的基分类器，我们还要考虑选取差异性较大的特征模型。综合考虑上述因素，本文选择了基于词袋模型（Bag of Words）的 TF-IDF，LSA，LDA，以及 SL 作为特征，用来表征原始文本。

## 4.1 向量空间模型与 TF-IDF

### 4.1.1 向量空间模型（VSM）

向量空间模型（Vector Space Model）是信息检索中最常用的检索方法，可以用于计算文本之间的相似度。向量空间模型将文本表示成一个维数为词汇表长度（即文档中的不同词语的总数）的向量。假设文档 D 的单词列表为 L，文档向量为 V，V[i]表示向量中第 i 个元素，L[i]表示单词列表中的第 i 个单词。那么 V[i]代表着 D 中的 L[i]的映射值。V[i]的取值可以有多种方案，假设 V[i]为 D 中 L[i]出现的频次，这时 V[i]的取值只与 L[i]有关，与其他单词无关。我们认为文本向量中的每一维代表一种隐语义，那么上述的文档向量 V 将是严格“一词一义”的，这并不符合自然语言的“一词多义，一义多词”的特点。因此，向量空间模型作为一种最为简单的模型，其缺陷也是尤为明显的，其语义鸿沟很大程度上限制了应用效果的上界。当然，在实际应用中，V[i]的取值并不会同上例那样，通常会取单词的 TF-IDF 值，TF-IDF 使得一个单词能尽量与文本在语义上相关，但本质上仍没有解决一词一义的问题。

### 4.1.2 TF-IDF

TF-IDF 的主要思想是：词语在不同的文章中出现的频率反差越大，则其类别区分能力越强。TF-IDF 值就是评价词语是否具有良好类别区分能力的指标，TF-IDF 值越大则词语的类别区分能力越高。设 D 为文档集合，$n_{i,j}$表示词语 i 在文档 j 中出现的次数，则词语 i 在文档 j 中的 TF-IDF 值的计算方法如式(4-1)所示：

$$TFIDF_{i,j} = TF_{i,j} \times IDF_{i,j} \tag{4-1}$$

其中$TF_{i,j}$和$IDF_{i,j}$的计算方法如式(4-2)和式(4-3)所示：

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{4-2}$$

$$IDF_{i,j} = \log \frac{|D|}{\left|\left\{ j : L_i \in D_j \right\}\right|} \tag{4-3}$$

在 TF-IDF 中，TF 表示词频(Term Frequency)，TF 值越大，则词语在文章

中出现的频率越高，IDF 表示逆向文件频率(Inverse Document Frequency)，IDF 值越大，则包含词语的文章越少，即词语越少在其他文章中出现。TD-IDF 方法经常会和余弦相似度(cosine similarity)一同使用于向量空间模型中，可以应用于文本分类，文本聚类，文本相似度计算等问题。

### 4.1.3 基于 TF-IDF 的向量空间模型（VSM）在本文中的应用

本文使用基于 TF-IDF 的向量空间模型作为文本的一个基础特征。该特征不直接被用于分类，而是由它来生成 LSA 特征。假设文档集合为 D，词语集合为 L，那么文档$D_j$可以表示为一个向量$V_j$，其中 $V_j[i] = TFIDF_{i,j}$，$V_j$即为文档$D_j$的特征。矩阵$M_{m\times n}$为词语-文档矩阵(Term-document Matrix)，其中 n=$|D|$，m=$|L|$，矩阵的第 j 行$M_j = V_j$。

在 M 矩阵的基础之上，本文分别使用 SVD 分解对向量空间模型进行降维，将文本映射到新的语义空间，得到 LSA 特征。

## 4.2 潜在语义分析（LSA）

### 4.2.1 潜在语义分析（LSA）理论简介

LSA(latent semantic analysis)潜在语义分析，也被称为 LSI(latent semantic index)，是 Scott Deerwester, Susan T. Dumais 等人在 1990 年提出来的一种新的索引和检索方法[15]。该方法和传统的向量空间模型一样使用向量来表示词和文档，但是 LSA 将词和文档映射到了一个低维的潜在语义空间。向量空间模型可以看做是一个最为原始的语义空间，其“一义一词”的基本假设导致特征模型有着巨大的语义鸿沟，而 LSA 通过 SVD 分解解决了文本的“一义多词”，构造出了词语之间的语义潜在相关性（即它能够把文档集合中具有隐含语义联系的词语联系起来）。又因为这个映射是基于共现表（即词语-文档矩阵）的奇异值分解，其基本假设“在同样的语境中使用的词语一般具有相似的含义”较为符合自然语言的特点，所以构造出的语义潜在相关性是合理的。

奇异值分解是对矩阵进行分解的一种方法，一个$\mathrm{m}\times\mathrm{n}$维的词语-文档矩阵 X，可以分解为$\mathrm{U}\times\Sigma\times V^T$。其中 U 为$\mathrm{m}\times\mathrm{r}$维正交矩阵，U 中的每一列称为左奇

异向量；S 为$\mathrm{r}\times\mathrm{r}$维对角矩阵，对角线上的每个值称为奇异值，r 为矩阵 X 的秩；V 为$\mathrm{n}\times\mathrm{r}$维正交矩阵,V 中的每一列称为右奇异向量。

在对词语-文档矩阵 X 做 SVD 分解之后，如果我们只取$\Sigma$中 K 个最大的奇异值，以及 U 和 V 中对应的 K 个奇异向量，可以得到三个新的矩阵：$\Sigma_1$，$U_1$和$V_1$。其中，对角矩阵$\Sigma_1$由我们所选取的 K 个奇异值构成，$U_1$由对应的 K 个左奇异向量构成，$V_1$由对应的 K 个右奇异向量构成。$X_1 = U_1 \times \Sigma_1 \times V_1^T$形成了一个新的 t*d 维矩阵，$X_1$即为去噪后的词语-文档矩阵，$X_1$和 X 在 F 范数上是相似的。LSA 的本质是将高维的语义空间映射到低维语义空间，同时通过选取 K 个奇异值（K<r）去噪。当原始的高维语义空间的噪声较少时，就要考虑选取较大的 K，否则会在去掉少量噪声的同时丢失大量有用信息。我们称 X 中的文档向量为原始向量，$X_1$中的文档向量为原始修正向量，他们都是处于原始语义空间之中的，原始修正向量一定程度上体现了 LSA 所捕获的潜在语义关系。

### 4.2.2 潜在语义分析（LSA）在本文中的应用

本文使用向量空间模型作为原始语义空间。SVD 分解所得到的矩阵$U_1$是$\mathrm{m}\times\mathrm{k}$维的，可以看做词语-隐语义分布；$V_1$是$\mathrm{n}\times\mathrm{k}$维的，可以看做文档-隐语义分布。设$V_1^T$的第 i 列为向量$\hat{d}_{1i}$，$X_1$的第 i 列为向量$d_{1i}$，则可以得到$\hat{d}_{1i}$和$d_{1i}$的关系如式（4-4）所示：

$$\hat{d}_{1i} = \Sigma_1^{-1} \times U_1^T \times d_{1i} \tag{4-4}$$

其中$d_{1i}$为$X_1$中第 i 个文档的在原始语义空间中的原始修正向量，$\hat{d}_{1i}$为该文档在新语义空间内的向量。

本文使用训练集构建 LSA 输入语料，对于任意文本 q（训练集或测试集中），使用$\hat{q} = U_1^T \times \mathrm{q}$作为其特征向量，并使用 SVM 来对特征进行分类。实验内容将在第五章介绍。

本文使用训练集构建 LSA 输入语料，对于测试集中的新文本在原始语义空间中的向量 q，使用式（4-5）将 q 映射到新的语义空间：

$$\hat{q} = \Sigma_1^{-1} \times U_1^T \times q \tag{4-5}$$

并在 GVM 中使用$\hat{q}$和$\hat{d}_{1i}$的余弦相似度作为测试文本和训练文本中第 i 个文本的相似度。相关实验内容将在第六章介绍。

## 4.3 隐含狄利克雷分布（LDA）

### 4.3.1 主题模型与生成模型

在机器学习和自然语言处理等领域中，主题模型（Topic Model）是用来在文档集合中发现抽象主题的一种统计模型。主题模型认为，一篇文章通常包含多个主题，每个主题所占比例各不相同，每个词语在不同的主题下出现的频率也各不相同。主题模型试图使用统计学理论来对文档-主题分布和主题-词语分布进行建模，进而推断文章的主题。

主题模型将文本映射到主题空间，主题空间与语义空间的本质是一样的。对于语义空间中的一种语义，我们同样可以认为是一种主题。因此像潜在语义分析(LSA)这样的方法我们可以认为是广义的主题模型。潜在语义分析同样像主题模型一样构建了词语-主题分布及主题-文档分布，在 4.2 节中提到的矩阵 U 可以看做词语-主题分布，矩阵 V 可以看做主题-文档分布。

本文认为，文章的主题是有层次的。一篇文章往往会有一个抽象级别最高的主题，然后围绕这个总的主题，有一些子主题；子主题之下还可以有子主题；词语是围绕最基本（最具体）的一个层级的子主题来写的。我们将这样的主题模型称之为层次主题模型，这个模型是较为符合自然语言特点的模型，本文并没有来构建文本的层次主题模型，我们仅假设层次主题模型为自然语言的真实模型，利用它来做理论分析。文章的主题层次与文章的长短有着密切联系。

本文通过把文本映射到一个合理的隐语义空间来获得其隐语义的向量表示。隐语义可以近似映射到层次主题模型中某一层次的主题，即每一种隐语义都可以看做是一种主题，本文对“主题”与“语义”不做区分。本文使用语义向量作为特征进行文本分类，影响某种特征分类的效果有两种重要影响因素：一是特征是否能够很好地拟合层次主题模型中某一层级的主题；二是该层次的主题与立场关系是否明确。我们在第 5,6 章可以看到不同的特征使用两种分类方法取得了不同的实验结果，本文在第六章将会对 LSA 和 LDA 两个主题模型做出相应分析，结合层次主题模型假设探讨其表现不同的原因。

本文其他小节所介绍的特征，其“主题”（或者说是“语义”）有着较差的可解释性，本节所介绍的潜在狄里克雷分配（Latent Dirichlet Allocation）

是 Blei 提出的一种生成模型[16]，其提取出的主题和建模过程有着较好的可解释性。在[16]中，Blei 介绍了三种除 LDA 以外的生成一篇文档的模型，本文接下来会先介绍下这三种模型，再着重讨论 LDA 模型。

### 4.3.2 一元语法模型(UM)

第一种是一元语法模型(Unigram Model)，它假设文本中的词服从多项式（Multinomial）分布。图 4-1 描述了生成 M 份每份包含 N 个词语的文档的过程[16]：

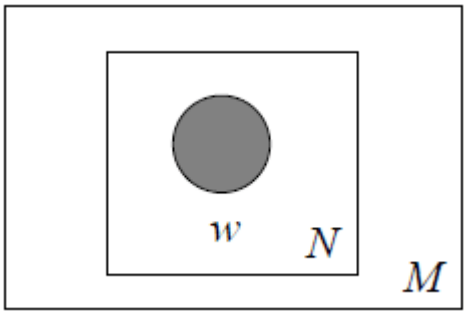


图 4-1 Unigram Model 的文档生成过程

对于每份文档中的每个词，每次从词的分布 p(w)中选取一个词，伪码如下：

```
for document_id in range(1,M):
    for word_id in range(1,N):
        choose a word w～p(w)
```

假设一个文档$\mathrm{d}=(w_1,w_2,\ldots,w_n)$,其中$w_n$表示文档中的第 n 个词语，那么生成文档 d 的概率如式（4-6）所示：

$$P(d)=\prod_{n=1}^{N}P(w_n) \tag{4-6}$$

### 4.3.3 混合一元语法模型（MU）

Unigram 模型的方法生成的文本没有考虑不同的主题，所有生成的词都是从同一个词语分布里随机抽取的，过于简单，混合一元语法模型（Mixture of

Unigrams）对其进行了改进。图 4-2 描述了如何生成 M 份由 N 个词语构成的文档[16]：

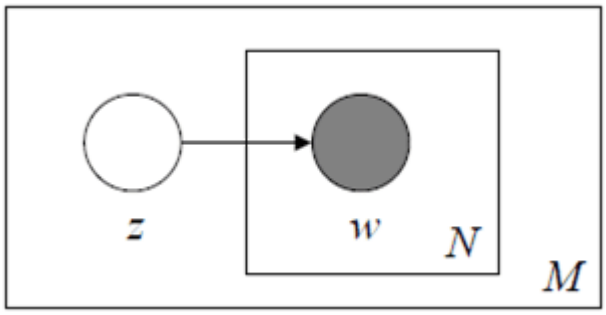


图 4-2 Mixture of Unigram 的文档生成过程

该模型使用如下方法生成 1 个由 N 个词语构成的文档：

1. 从主题分布 P(z)中选取一个主题 z
2. 每次从主题-词语分布 P(w|z)中随机选取一个词语，重复 N 次
3. 重复上述过程 M 次，即可生成 M 份文档。其中，P(z)表示主题的概率分布， P(w|z)表示给定主题时词语的分布。

因此，对于一个给定的文档$\mathrm{d}=(w_1,w_2,\ldots,w_n)$，其生成概率如式（4-7）所示：

$$P(d)=\sum_z P(z)\prod_{n=1}^{N}P(w_n|z) \tag{4-7}$$

### 4.3.4 概率潜在语义索引（pLSA）

在 Mixture of Unigrams 模型中，一篇文档只有一种主题，而概率潜在语义索引（Probabilistic latent semantic indexing）认为一篇文章由多个主题构成。使用 pLSA 生成 M 份每份 N 个词语的文档的过程如图 4-3 所示[16]：

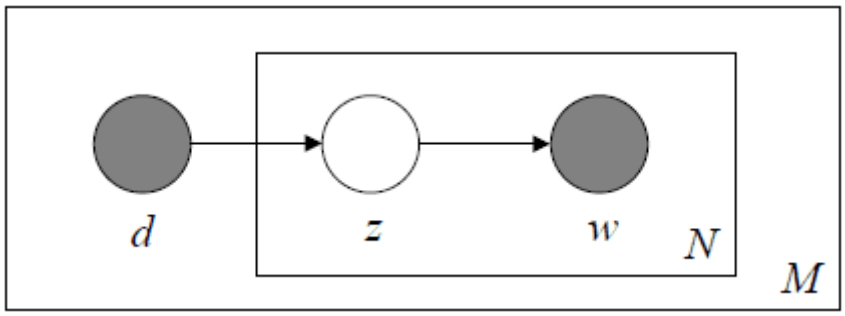


图 4-3 pLSA 的文档生成过程

pLSA 生成一篇文档的过程为：

1. 从概率分布 P(d)中选择一篇文档$d_i$，然后重复 2-3 N 次

2. 选定文档$d_i$后，从主题分布$P(z|d_i)$中选择一个隐含主题$z_k$

3. 选定$z_k$后，从词分布$P(w|z_k)$中选择一个词$w_j$

上述过程重复 M 次即可生成 M 份文档。

因此，对于一个给定的文档$d = (w_1, w_2, \ldots, w_n)$，第 i 个词为$w_i$的概率如式（4-8）所示，生成文档 d 的概率如式（4-9）所示：

$$P(d, w_i) = P(d)\sum_z P(w_i|z)P(z|d) \tag{4-8}$$

$$P(d, w_1, w_2, \ldots, w_n) = P(d)\prod_n \sum_z P(w_i|z)P(z|d) \tag{4-9}$$

### 4.3.5 隐含狄利克雷分布（LDA）的文档生成方法

LDA 模型是在 pLSA 模型的基础之上加上了贝叶斯框架，其文档集合 D 中每一篇文章 d 的生成方式如下（D 中共有 M 篇文章）：

1.按照先验概率$P(d_i)$选择文档$d_i$的长度 N，$P(d_i)$为以 ξ 为参数的泊松分布

2.从 Dirichlet 分布$\vec{\alpha}$中取样生成文档$d_i$的主题分布$\theta_i$（主题分布$\theta_i$是由参数为$\vec{\alpha}$的 Dirichlet 分布生成的），然后重复 3-5 N 次

3.从主题的多项式分布$\theta_i$中取样生成文档$d_i$的第 j 个词的主题$z_{i,j}$

4.从 Dirichlet 分布$\vec{\beta}$中取样生成主题$z_{i,j}$对应的词语分布$\Phi_{z_{i,j}}$（词语分布$\Phi_{z_{i,j}}$是由参数为$\vec{\beta}$的 Dirichlet 分布生成的）

5.从词语的多项式分布$\Phi_{z_{i,j}}$中采样最终生成词语$\omega_{i,j}$

这里模型假设了文档集合 D 中文档的长度是服从泊松分布的。这个假设不

是至关重要的，可以根据需要使用更实际的文档分布长度。

### 4.3.6 隐含狄利克雷分布（LDA）的参数估计

在 4.3.5 小节中我们介绍了 LDA 模型的文档生成方式，现在回到我们使用 LDA 模型的目的上来：我们希望能够把文档映射到主题空间。从 4.3.5 中我们可以看出，生成的每个文档有其特有的主题分布，这些分布就是我们所需要的。我们将每个文档的主题分布合并在一起，就得到了整个文档集合的文档-主题分布矩阵。

LDA 的参数估计主要就是为了得到主题-词语分布 $\Phi(k \times n)$，文档-主题分布 $\Theta(m \times k)$。在[16]中，Blei 提出了使用变分-EM 算法来进行参数估计，后来人们发现一种对 LDA 进行参数估计的更好方法:Gibbs 采样。Gibbs 采样基于马尔可夫链蒙特卡尔理论，可以帮助我们获取多个随机变量的联合概率分布的近似观察样本。

在 LDA 模型中，文档集合 D 中的每个文档 w 均为已知变量；$\vec{\alpha}$和$\vec{\beta}$是根据经验给定的先验参数；z， θ 和$\boldsymbol{\Phi}$为未知隐含变量，需要我们根据观察到的变量来估计。根据 4.3.5 节的介绍，我们可以写出生成一篇文档$w_i$时的联合概率分布，如式（4-10）所示：

$$P\left(\vec{w}_l,\vec{z}_l,\vec{\theta}_l,\Phi_l \mid \alpha,\beta\right)=P\left(\theta_i \mid \alpha\right)\prod_{j=1}^{N}P\left(Z_{i,j} \mid \theta_i\right)P\left(\Phi_{Z_{i,j}} \mid \vec{\beta}\right)P\left(w_{i,j} \mid \Phi_{Z_{i,j}}\right) \quad (4\text{-}10)$$

因为$\vec{\alpha}$ 产生主题分布$\overrightarrow{\theta_l}$，主题分布$\overrightarrow{\theta_l}$确定具体主题，$\vec{\beta}$产生词分布$\boldsymbol{\Phi}$、词分布$\boldsymbol{\Phi}$确定具体词，即在$\overrightarrow{z_l}$确定时$\vec{\alpha}$和$\overrightarrow{w_l}$是独立的，所以式（4-10）等价于式（4-11）所表达的联合概率分布$\mathrm{P}(\overrightarrow{w_l},\overrightarrow{z_l})$：

$$P\left(\vec{w}_l,\vec{z}_l \mid \vec{\alpha},\vec{\beta}\right)=P\left(\vec{w}_l \mid \vec{z}_l,\vec{\beta}\right)P\left(\vec{z}_l \mid \vec{\alpha}\right) \quad (4\text{-}11)$$

其中，对$\mathrm{P}\left(\overrightarrow{w_l}|\overrightarrow{z_l},\vec{\beta}\right)$有：

$$P\left(\vec{w}_l \mid \vec{z}_l,\vec{\beta}\right)=\int P\left(\vec{w}_l \mid \vec{z}_l,\Phi\right)P\left(\Phi \mid \vec{\beta}\right)d\Phi=\prod_{z=1}^{K}\frac{\Delta\left(\vec{n}_z+\beta\right)}{\Delta\left(\vec{\beta}\right)} \quad (4\text{-}12)$$

其中$\Delta\left(\vec{\beta}\right)$和$\overrightarrow{n_z}$的定义如式（4-13）和式（4-14）所示，$\Delta$为 Dirichlet 分布的归一化系数公式：

$$\Delta\left(\vec{\beta}\right)=\int\prod_{k=1}^{V}P_k^{\beta_k-1} \tag{4-13}$$

$$\vec{n}_z=\left\{n_z^{(t)}\right\}_{t=1}^{V} \tag{4-14}$$

其中，$n_z^{(t)}$是词 t 在主题 k 中出现的次数。

对$P(\vec{z_l}|\vec{\alpha})$有：

$$P\left(\vec{z}_l\mid\vec{\alpha}\right)=\prod_{m=1}^{M}\frac{\Delta\left(\vec{n}_m+\vec{\alpha}\right)}{\Delta\left(\vec{\alpha}\right)},\vec{n}_m=\left\{n_m^{(k)}\right\}_{k=1}^{K} \tag{4-15}$$

上式中$n_m^{(k)}$为主题 k 在文章 m 中出现的次数。这样我们便得到了$P(\vec{w_l},\vec{z_l}|\vec{\alpha},\vec{\beta})$的联合概率分布的结果：

$$P\left(\vec{w}_z,\vec{z}_l\mid\vec{\alpha},\vec{\beta}\right)=\prod_{z=1}^{K}\frac{\Delta\left(\vec{n}_z+\vec{\beta}\right)}{\Delta\left(\vec{\beta}\right)}\prod_{m=1}^{M}\frac{\Delta\left(\vec{n}_m+\vec{\alpha}\right)}{\Delta\left(\vec{\alpha}\right)} \tag{4-16}$$

有了联合概率分布，便可以通过贝叶斯理论计算在给定观测变量$\vec{w_l}$下隐变量,$\vec{z_l}$的条件分布$P(\vec{z_l}|\vec{w_l},\vec{\alpha},\vec{\beta})$：

$$P\left(\vec{z}_l\mid\vec{w}_l,\vec{\alpha},\vec{\beta}\right)=P\left(\vec{w}_l,\vec{z}_l\mid\vec{\alpha},\vec{\beta}\right)/P\left(\vec{w}_l\mid\vec{\alpha},\vec{\beta}\right) \tag{4-17}$$

由于分母要对 K 的 N 次方个项求和因此直接求不可行，即概率分布$P(\vec{w_l}|\vec{\alpha},\vec{\beta})$的样本难以生成，因此可以采用 Gibbs 抽样的方法完成对$P(\vec{w_l}|\vec{\alpha},\vec{\beta})$的抽样，然后便可以利用抽样结果通过简单的似然估计求得 Φ 和 Θ。关于 Gibbs 采样的深入理论以及使用 Gibbs 采样来计算 LDA 的详细细节，可以参考 Heinrich G 在[17]中的工作。

### 4.3.7 隐含狄利克雷分布（LDA）在本文中的数据预处理及应用

本文使用 jGibbLDA（一个 LDA 的 java 开源工具包）来为训练文件和测试文件建模。

在建模之前要先对第 3 章得到的结果做去除停止词的预处理，将预处理得到的结果作为 LDA 的输入。将每个 target 的训练集语料（包括 FAVOR,AGAINST,NONE，UNKNOWN 四个类别）整理到一个文件中，作为 jGibbLDA 的训练输入，模型训练完后生成的文件包括一个“.theta”文件和一个“.phi”文件。“.theta”文件为训练语料的文档-主题分布，“.phi”文件为训练语料的主题-单词分布。

我们利用训练 LDA 得到的模型文件，分别对测试集和训练集进行了推断，每次推断同样会得到一个“.theta”文件和一个“.phi”文件，它们分别存储了被推断的文档的文档-主题分布和主题-单词分布。每一条微博文本在“.theta”文件中都可以找到其对应的主题分布向量，我们使用这个向量作为微博文本的特征，分别使用 SVM 和 GVM 进行了分类实验，相关内容将在第 5,6 章详细介绍。

## 4.4 情感词词典特征（SL）

本文对情感分析的基本方法——情感词词典特征（Sentiment Lexicon Feature,又称 SL）在立场分析中的应用做了一定的尝试和探索。本文使用 Hownet 的情感词数据集以及清华大学自然语言处理与社会人文计算实验室李军等人制作的中文褒贬义词典作为词典集[18]。我们使用这两个词典来计算文本的情感特征。设$P_1$和$N_1$分别为利用 Hownet 词典计算出的正面情感特征和负面情感特征，$P_2$和$N_2$分别为利用李军等制作的中文褒贬义词典计算出的正面情感特征和负面情感特征；对于一条微博文本 w，$P_{w1}$为文本 w 中出现的 Hownet 正面情感词的数量，$N_{w1}$为文本 w 中出现的 Hownet 负面情感词的数量，$P_{w2}$为文本 w 中出现的中文褒贬义词典中正面情感词的数量，$N_{w2}$为文本 w 中出现的中文褒贬义词典中负面情感词的数量的数量。我们使用式（4-18），式（4-19），式（4-20），式（4-21）分别来计算$P_{w1}$，$N_{w1}$，$P_{w2}$，$N_{w2}$：

$$P_1 = \frac{P_{w1}+1}{P_{w1}+N_{w1}^{\lambda}+2} \tag{4-18}$$

$$N_1 = \frac{N_{w1}^{\lambda}+1}{P_{w1}+N_{w1}^{\lambda}+2} \tag{4-19}$$

$$P_2 = \frac{P_{w2}+1}{P_{w2}+P_{w2}^{\lambda}+2} \tag{4-20}$$

$$N_2 = \frac{N_{w2}+1}{N_{w2}+N_{w2}^{\lambda}+2} \tag{4-21}$$

其中，λ用来控制负面情感词的影响。我们使用$P_{w1}$，$N_{w1}$，$P_{w2}$，$N_{w2}$作为特征进行了 SVM 分类实验，参数设置和实验结果将在第五章介绍。

## 4.5 本章小结

本章首先介绍了文本的语义表示，语义空间，特征工程的概念。介绍了什么是特征的语义鸿沟以及在特征工程中减小语义鸿沟的重要性。阐述了本文特征的选择原则以及选用了哪些特征。接下来分别介绍了 VSM，TF-IDF，LSA，LDA 和 SL 特征，包括特征的基本思想和基本原理以及在本文中是如何应用的等内容。

特别地，本文在 4.3 节介绍 LDA 时，提出了本文的层次主题模型假设，并基于这个假设提出了在映射过程中影响特征分类效果的两个重要因素。这个假设我们在第六章还会提到，基于这个假设我们分析了 LDA 和 LSA 的效果差异。

# 第5章　SVM 分类

支持向量机(Support Vector Machine,SVM)是由 AT&TBell 实验室的 C Corte 和 V Vapnik 提出的一种监督学习方法[19]，广泛应用于分类，回归分析等任务。SVM 通过使用非线性映射算法将低维空间的线性不可分样本转化为高维特征空间的线性可分样本，从而使得采用线性算法对样本的非线性特征进行线性分析成为可能。本文使用 libsvm[20]来对第四章所提出的特征进行分类。在 libsvm 中，提供了 5 种任务类型，其中 c-svc 和 nu-svc 都可以用来完成多分类任务，本文使用 c-svc 进行实验。

为了更好地证明本文方法的有效性，同时也因为第七章的方法需要额外的数据集来进行参数选择，本章将评测数据集的测试集和训练集整合在一起，按照 75%训练集，25%测试集的比例对数据集进行重新划分，生成了 2 组验证集，分别称为验证数据集 1 和验证数据集 2。本章将对这 2 组验证数据集以及评测数据集分别进行实验，并分析实验结果。

## 5.1 评测数据集及验证数据集的统计信息

评测数据集的统计信息如表 5-1 所示：

表 5-1 评测数据集的统计信息

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 |
|---|---|---|---|---|---|
| 训练集 FAVOR 数据条数 | 245 | 250 | 260 | 250 | 160 |
| 训练集 AGAINST 数据条数 | 209 | 250 | 200 | 250 | 301 |
| 训练集 NONE 数据条数 | 146 | 100 | 140 | 100 | 126 |
| 测试集 FAVOR 数据条数 | 75 | 94 | 99 | 88 | 63 |
| 测试集 AGAINST 数据条数 | 104 | 86 | 95 | 94 | 110 |
| 测试集 NONE 数据条数 | 21 | 20 | 6 | 18 | 27 |
| 训练集数据总量 | 600 | 600 | 600 | 600 | 587 |
| 测试集数据总量 | 200 | 200 | 200 | 200 | 200 |
| 测试集词汇量 | 1945 | 1688 | 2463 | 2709 | 1272 |
| 训练集词汇量 | 3517 | 4068 | 5456 | 5570 | 4490 |
| 数据集总词汇量 | 4231 | 4709 | 6352 | 6534 | 4886 |

验证数据集 1 的统计信息如表 5-2 所示：

表 5-2 验证数据集 1 的统计信息

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 |
|---|---|---|---|---|---|
| 训练集 FAVOR 数据条数 | 240 | 258 | 269 | 253 | 167 |
| 训练集 AGAINST 数据条数 | 234 | 252 | 221 | 258 | 308 |
| 训练集 NONE 数据条数 | 125 | 90 | 109 | 88 | 114 |
| 测试集 FAVOR 数据条数 | 80 | 86 | 90 | 85 | 56 |
| 测试集 AGAINST 数据条数 | 79 | 84 | 74 | 86 | 103 |
| 测试集 NONE 数据条数 | 42 | 30 | 37 | 30 | 39 |
| 训练集数据总量 | 599 | 600 | 599 | 599 | 589 |
| 测试集数据总量 | 201 | 200 | 201 | 201 | 198 |
| 测试集词汇量 | 1934 | 1943 | 2655 | 2779 | 2136 |
| 训练集词汇量 | 3524 | 3954 | 5378 | 5545 | 4135 |
| 数据集总词汇量 | 4231 | 4709 | 6352 | 6534 | 4886 |

验证数据集 2 的统计信息如表 5-3 所示：

表 5-3 验证数据集 2 的统计信息

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 |
|---|---|---|---|---|---|
| 训练集 FAVOR 数据条数 | 240 | 258 | 270 | 254 | 168 |
| 训练集 AGAINST 数据条数 | 235 | 252 | 222 | 258 | 309 |
| 训练集 NONE 数据条数 | 126 | 90 | 110 | 89 | 115 |
| 测试集 FAVOR 数据条数 | 80 | 86 | 89 | 84 | 55 |
| 测试集 AGAINST 数据条数 | 78 | 84 | 73 | 86 | 102 |
| 测试集 NONE 数据条数 | 41 | 30 | 36 | 29 | 38 |
| 训练集数据总量 | 601 | 600 | 602 | 601 | 592 |
| 测试集数据总量 | 199 | 200 | 198 | 199 | 195 |
| 测试集词汇量 | 1814 | 2047 | 2634 | 2807 | 2184 |
| 训练集词汇量 | 3635 | 3845 | 5402 | 5452 | 4092 |
| 数据集总词汇量 | 4231 | 4709 | 6352 | 6534 | 4886 |

## 5.2 使用 SVM 对 LSA 特征进行分类

### 5.2.1 参数设置

本节使用 4.2 节中的方法为每个 target 进行 LSA 建模，并生成相应训练数据和测试数据的特征文件，每一条微博文本对应一个特征。根据经验，本节使用训练文档数量（即训练数据中包含的微博数量）作为 LSA 的维数（即选取的奇异值的个数）。

### 5.2.2 在评测数据集上的实验

首先，本文使用了默认参数的 4 种核函数进行了实验，实验结果如表 5-4 所示（表中 0-Favor 表示使用第一种核函数时 Favor 类别的 F 值。0,1,2,3 分别对应了 linear，polynomial，radial basis function，sigmoid 四种核函数。本文所有表格中空白的地方表示计算对应类别的准确率时发生了除 0 错误，即没有样本被预测成该类别。本文实验结果均使用去尾法进行舍入，保留四位有效数字。）：

表 5-4 评测数据集上使用四种默认参数核函数的 LSA-SVM 实验结果

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| 0-Favor | 0.4927 | 0.5856 | 0.7425 | 0.7608 | 0.6277 | 0.6419 |
| 0-AGAINST | 0.5578 | 0.5795 | 0.7471 | 0.7956 | 0.7557 | 0.6872 |
| 0-Average | 0.5253 | 0.5825 | 0.7448 | 0.7782 | 0.6917 | 0.6645 |
| 1-Favor | 0.5454 | 0.6394 | 0.6622 | | | |
| 1-AGAINST | | | | 0.6394 | 0.7096 | |
| 1-Average | | | | | | |
| 2-Favor | 0.5421 | 0.0980 | 0.7000 | 0.7191 | | |
| 2-AGAINST | | 0.5971 | 0.3157 | 0.7941 | 0.7096 | |
| 2-Average | | 0.3475 | 0.5078 | 0.7566 | | |
| 3-Favor | 0.5454 | 0.0606 | 0.6780 | 0.7187 | | |
| 3-AGAINST | | 0.5978 | 0.1372 | 0.7789 | 0.7096 | |
| 3-Average | | 0.3292 | 0.4076 | 0.7488 | | |

表 5-5 展示了使用线性核函数时，使用不同的惩罚因子 C 时的实验结果

（表格中的数值为 $F_{AVG}$ 值）：

表 5-5 评测数据集上使用线性核函数的 LSA-SVM 实验结果

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.001 | | 0.3292 | | 0.7327 | | |
| C=0.01 | 0.3215 | 0.4420 | 0.6964 | 0.7741 | 0.3725 | 0.5213 |
| C=0.05 | 0.5264 | 0.4982 | 0.7869 | 0.7914 | 0.7035 | 0.6613 |
| C=0.1 | 0.5545 | 0.5629 | 0.7774 | 0.7914 | 0.7462 | 0.6865 |
| C=0.2 | 0.5557 | 0.5925 | 0.7625 | 0.7924 | 0.7264 | 0.6859 |
| C=0.3 | 0.5542 | 0.6002 | 0.7535 | 0.7760 | 0.7163 | 0.6800 |
| C=0.4 | 0.5360 | 0.5775 | 0.7424 | 0.7760 | 0.7110 | 0.6686 |
| C=0.5 | 0.5390 | 0.5789 | 0.7458 | 0.7760 | 0.7110 | 0.6701 |
| C=1 | 0.5253 | 0.5825 | 0.7448 | 0.7782 | 0.6917 | 0.6645 |
| C=1.5 | 0.5253 | 0.5713 | 0.7448 | 0.7461 | 0.6798 | 0.6535 |
| C=3 | 0.5253 | 0.5647 | 0.7448 | 0.7406 | 0.6610 | 0.6473 |
| C=10 | 0.5253 | 0.5641 | 0.7448 | 0.7406 | 0.6610 | 0.6472 |
| C=10000 | 0.5253 | 0.5641 | 0.7448 | 0.7406 | 0.6610 | 0.6472 |

### 5.2.3 在验证数据集 1 上的实验

首先，本文使用了默认参数的 4 种核函数进行了实验，实验结果如表 5-6 所示：

表 5-6 验证数据集 1 上使用四种默认参数核函数的 LSA-SVM 实验结果

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| 0-Favor | 0.5767 | 0.5176 | 0.7487 | 0.7302 | 0.6355 | 0.6417 |
| 0-AGAINST | 0.5576 | 0.4918 | 0.8079 | 0.7425 | 0.7768 | 0.6753 |
| 0-Average | 0.5671 | 0.5047 | 0.7783 | 0.7363 | 0.7061 | 0.6585 |
| 1-Favor | 0.5694 | 0.6014 | 0.6186 | | | |
| 1-AGAINST | | | | 0.5993 | 0.6844 | |
| 1-Average | | | | | | |
| 2-Favor | 0.5683 | 0.5871 | 0.6769 | 0.7111 | | |
| 2-AGAINST | 0.1124 | 0.5207 | 0.5192 | 0.7396 | 0.6844 | 0.5152 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2-Average | 0.3403 | 0.5539 | 0.5981 | 0.7253 | | |
| 3-Favor | 0.5694 | 0.6014 | 0.6290 | 0.7174 | | |
| 3-AGAINST | | | 0.1707 | 0.7340 | 0.6844 | |
| 3-Average | | | 0.3999 | 0.7257 | | |

表 5-7 展示了使用线性核函数时，使用不同的惩罚因子 C 时的实验结果（表格中的数值为 $F_{AVG}$ 值）：

表 5-7 验证数据集 1 上使用线性核函数的 LSA-SVM 实验结果

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.001 | | | | 0.6880 | | |
| C=0.01 | 0.4657 | 0.5472 | 0.7300 | 0.7530 | 0.4871 | 0.5966 |
| C=0.05 | 0.5858 | 0.5614 | 0.7787 | 0.7574 | 0.7077 | 0.6782 |
| C=0.1 | 0.5823 | 0.5501 | 0.7859 | 0.7343 | 0.7121 | 0.6729 |
| C=0.2 | 0.5904 | 0.5211 | 0.7833 | 0.7337 | 0.6930 | 0.6643 |
| C=0.3 | 0.5818 | 0.5349 | 0.7833 | 0.7307 | 0.6985 | 0.6659 |
| C=0.4 | 0.5714 | 0.5495 | 0.7837 | 0.7307 | 0.7164 | 0.6704 |
| C=0.5 | 0.5687 | 0.5468 | 0.7885 | 0.7363 | 0.7085 | 0.6698 |
| C=1 | 0.5671 | 0.5047 | 0.7783 | 0.7363 | 0.7061 | 0.6585 |
| C=1.5 | 0.5811 | 0.4934 | 0.7783 | 0.7363 | 0.7061 | 0.6591 |
| C=3 | 0.5870 | 0.4875 | 0.7783 | 0.7363 | 0.7061 | 0.6591 |
| C=10 | 0.5749 | 0.4875 | 0.7783 | 0.7363 | 0.7155 | 0.6585 |
| C=10000 | 0.5320 | 0.4875 | 0.7783 | 0.7363 | 0.7155 | 0.6499 |

### 5.2.4 在验证数据集 2 上的实验

首先，本文使用了默认参数的 4 种核函数进行了实验，实验结果如表 5-8 所示：

表 5-8 验证数据集 2 上使用四种默认参数核函数的 LSA-SVM 实验结果

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| 0-Favor | 0.6207 | 0.5340 | 0.7097 | 0.7598 | 0.6538 | 0.6556 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0-AGAINST | 0.5752 | 0.4908 | 0.7261 | 0.7955 | 0.8072 | 0.6789 |
| 0-Average | 0.5979 | 0.5124 | 0.7179 | 0.7776 | 0.7305 | 0.6673 |
| 1-Favor | 0.5735 | 0.6014 | 0.6202 | | | |
| 1-AGAINST | | | | 0.6035 | 0.6869 | |
| 1-Average | | | | | | |
| 2-Favor | 0.5948 | 0.4196 | 0.6693 | 0.7802 | | |
| 2-AGAINST | 0.1364 | 0.5815 | 0.4554 | 0.7701 | 0.6869 | 0.5260 |
| 2-Average | 0.3656 | 0.5005 | 0.5624 | 0.7751 | | |
| 3-Favor | 0.5735 | 0.6014 | 0.6449 | 0.7802 | | |
| 3-AGAINST | | | 0.2381 | 0.7701 | 0.6869 | |
| 3-Average | | | 0.4415 | 0.7751 | | |

表 5-9 展示了使用线性核函数时，使用不同的惩罚因子 C 时的实验结果（表格中的数值为 $F_{AVG}$ 值）：

表 5-9 验证数据集 2 上使用线性核函数的 LSA-SVM 实验结果

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.001 | | | | 0.7520 | | |
| C=0.01 | 0.4768 | 0.4995 | 0.6831 | 0.7913 | 0.4498 | 0.5801 |
| C=0.05 | 0.5971 | 0.5216 | 0.7515 | 0.7947 | 0.6824 | 0.6695 |
| C=0.1 | 0.5753 | 0.5299 | 0.7530 | 0.8048 | 0.6753 | 0.6677 |
| C=0.2 | 0.6152 | 0.5296 | 0.7357 | 0.7950 | 0.7153 | 0.6781 |
| C=0.3 | 0.6174 | 0.5273 | 0.7299 | 0.7892 | 0.7364 | 0.6801 |
| C=0.4 | 0.6052 | 0.5116 | 0.7183 | 0.7871 | 0.7337 | 0.6712 |
| C=0.5 | 0.6026 | 0.5116 | 0.7067 | 0.7871 | 0.7310 | 0.6678 |
| C=1 | 0.5979 | 0.5124 | 0.7179 | 0.7776 | 0.7305 | 0.6673 |
| C=1.5 | 0.5997 | 0.5064 | 0.7179 | 0.7776 | 0.7250 | 0.6653 |
| C=3 | 0.6056 | 0.5134 | 0.7179 | 0.7663 | 0.7282 | 0.6663 |
| C=10 | 0.6056 | 0.5134 | 0.7179 | 0.7663 | 0.7349 | 0.6676 |
| C=10000 | 0.6056 | 0.5134 | 0.7179 | 0.7663 | 0.7349 | 0.6676 |

### 5.2.5 实验结果分析及方法评价

在三个数据集上的 F 值均值随参数 C 的变化如图 5-1 所示：

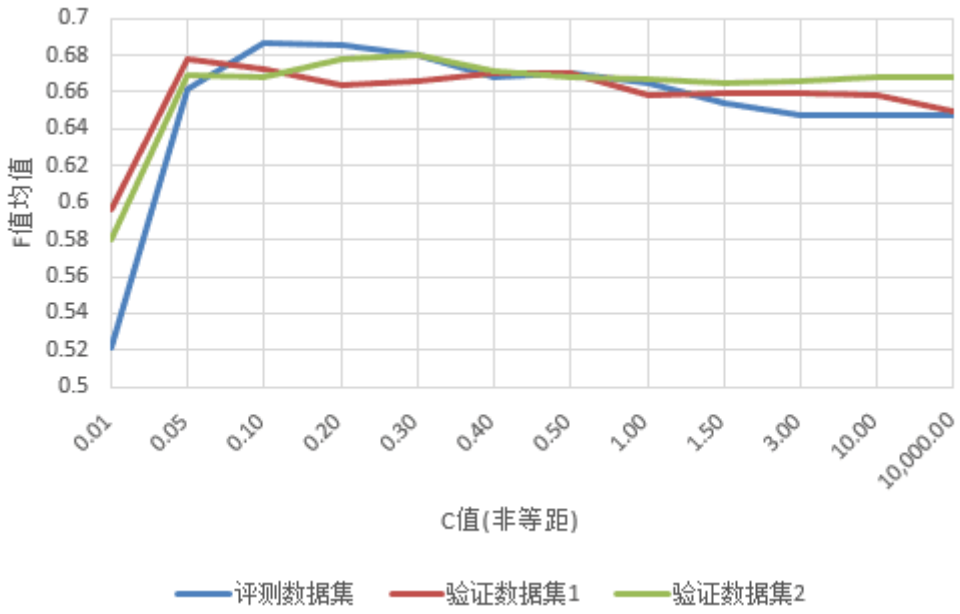


图 5-1 使用线性核函数的 LSA-SVM 实验结果随参数变化趋势图

我们可以看到，在各个数据集上使得 $F_{AVG}$ 取得最大值的参数 C 各不相同，我们可以找到一段较优参数区间 D[0.05,0.5]，使得 LSA-SVM 有总体相对不错的表现。

在评测数据集，验证数据集 1 和验证数据集 2 上，D 区间内 $F_{AVG}$ 的极差大致分别为 0.24,0.2,0.13，极差较大。这就意味着我们在参数调优后，还有可能在新的测试集上取得与最好结果（平均）差距达 1.9%的实验结果。我们希望这个差距能够更小，分类器的泛化能力更强。

总体而言，LSA-SVM 有着较为不错的效果，在 5,6 章提到的方法中，其效果仅次于 LSA-GVM 方法，但是 LSA-SVM 运行效率很低，当训练集数据量达到 600 时，在 PC 上对词语-文档矩阵进行 SVD 分解的时间就已经远远不能满足用户实时分析的需求，因此其应用有着一定的局限性。

## 5.3 使用 SVM 对 LDA 特征进行分类

### 5.3.1 参数设置

本文使用 java 开源工具 jGibbLDA 来进行 LDA 建模与推断，令 $\beta$ =0.01，$\alpha$ =50/k。其中 k 为主题数量。我们分别取 k=60,100,200 进行实验。

本文使用线性核作为 SVM 的核函数，惩罚因子 C 分别取 0.1,0.5,1,10 进行实验。

### 5.3.2 在评测数据集上的实验

k=200 时，实验结果如表 5-10 所示：

表 5-10 评测数据集上的 LDA-SVM 实验结果（k=200）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | | 0.3949 | 0.4279 | 0.7077 | | |
| C=0.5 | 0.3597 | 0.4909 | 0.5804 | 0.7280 | 0.4282 | 0.5174 |
| C=1 | 0.4488 | 0.5131 | 0.6755 | 0.7062 | 0.5418 | 0.5770 |
| C=10 | 0.5045 | 0.4645 | 0.5953 | 0.6050 | 0.5911 | 0.5520 |

k=100 时，实验结果如表 5-11 所示：

表 5-11 评测数据集上的 LDA-SVM 实验结果（k=100）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | | 0.4733 | 0.4850 | 0.7440 | | |
| C=0.5 | 0.3677 | 0.5060 | 0.7011 | 0.7519 | 0.4881 | 0.5629 |
| C=1 | 0.4048 | 0.4842 | 0.7450 | 0.7626 | 0.5870 | 0.5967 |
| C=10 | 0.3932 | 0.4614 | 0.6959 | 0.7361 | 0.6644 | 0.5902 |

k=60 时，实验结果如表 5-12 所示：

表 5-12 评测数据集上的 LDA-SVM 实验结果（k=60）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | | 0.3320 | 0.5116 | 0.7590 | 0.3854 | |
| C=0.5 | 0.4481 | 0.4225 | 0.6242 | 0.7461 | 0.5497 | 0.5581 |

| | | | | | | |
|---|---|---|---|---|---|---|
| C=1 | 0.5311 | 0.4293 | 0.6437 | 0.7456 | 0.5937 | 0.5886 |
| C=10 | 0.5745 | 0.4792 | 0.6624 | 0.7331 | 0.6230 | 0.6144 |

### 5.3.3 在验证数据集 1 上的实验

k=200 时，实验结果如表 5-13 所示

表 5-13 验证数据集 1 上的 LDA-SVM 实验结果（k=200）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | 0.2982 | | 0.4527 | 0.6764 | 0.3934 | |
| C=0.5 | 0.4557 | 0.5019 | 0.6715 | 0.6663 | 0.4679 | 0.5526 |
| C=1 | 0.4802 | 0.4917 | 0.6640 | 0.6357 | 0.5844 | 0.5712 |
| C=10 | 0.4432 | 0.4020 | 0.5763 | 0.6313 | 0.5574 | 0.5220 |

k=100 时，实验结果如表 5-14 所示

表 5-14 验证数据集 1 上的 LDA-SVM 实验结果（k=100）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | 0.3159 | | 0.5018 | 0.7212 | 0.3965 | |
| C=0.5 | 0.4775 | 0.5215 | 0.5952 | 0.7368 | 0.5865 | 0.5835 |
| C=1 | 0.5101 | 0.5292 | 0.6059 | 0.7175 | 0.6208 | 0.5967 |
| C=10 | 0.5028 | 0.5787 | 0.5785 | 0.6727 | 0.6621 | 0.5989 |

k=60 时，实验结果如表 5-15 所示

表 5-15 验证数据集 1 上的 LDA-SVM 实验结果（k=60）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | 0.3466 | 0.3102 | 0.5628 | 0.7195 | 0.4456 | 0.4769 |

| | | | | | | |
|---|---|---|---|---|---|---|
| C=0.5 | 0.4850 | 0.4983 | 0.6300 | 0.7588 | 0.5102 | 0.5764 |
| C=1 | 0.5174 | 0.5603 | 0.6561 | 0.7521 | 0.6019 | 0.6175 |
| C=10 | 0.4620 | 0.5072 | 0.6738 | 0.7451 | 0.6356 | 0.6047 |

### 5.3.4 在验证数据集 2 上的实验

k=200 时，实验结果如表 5-16 所示

表 5-16 验证数据集 2 上的 LDA-SVM 实验结果（k=200）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | | | 0.4930 | 0.6998 | 0.3633 | |
| C=0.5 | 0.4715 | 0.4550 | 0.6261 | 0.7153 | 0.4771 | 0.5490 |
| C=1 | 0.4783 | 0.4636 | 0.6212 | 0.6954 | 0.5976 | 0.5712 |
| C=10 | 0.4584 | 0.4436 | 0.6230 | 0.6901 | 0.5876 | 0.5605 |

k=100 时，实验结果如表 5-17 所示

表 5-17 验证数据集 2 上的 LDA-SVM 实验结果（k=100）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | 0.3518 | | 0.4907 | 0.6842 | 0.3633 | |
| C=0.5 | 0.4657 | 0.5023 | 0.6169 | 0.7480 | 0.4971 | 0.5660 |
| C=1 | 0.4406 | 0.5079 | 0.6403 | 0.7283 | 0.5245 | 0.5683 |
| C=10 | 0.4515 | 0.4887 | 0.6295 | 0.7159 | 0.5947 | 0.5760 |

k=60 时，实验结果如表 5-18 所示

表 5-18 验证数据集 2 上的 LDA-SVM 实验结果（k=60）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| C=0.1 | 0.5189 | 0.3689 | 0.5451 | 0.7210 | 0.4324 | 0.5172 |
| C=0.5 | 0.5271 | 0.4891 | 0.7175 | 0.7373 | 0.5502 | 0.6042 |
| C=1 | 0.5452 | 0.4887 | 0.6955 | 0.7499 | 0.5951 | 0.6148 |
| C=10 | 0.5160 | 0.4642 | 0.6949 | 0.7380 | 0.6455 | 0.6117 |

### 5.3.5 实验结果分析及方法评价

LDA-SVM 方法在评测数据集上的实验结果可以由图 5-2 直观地展示：

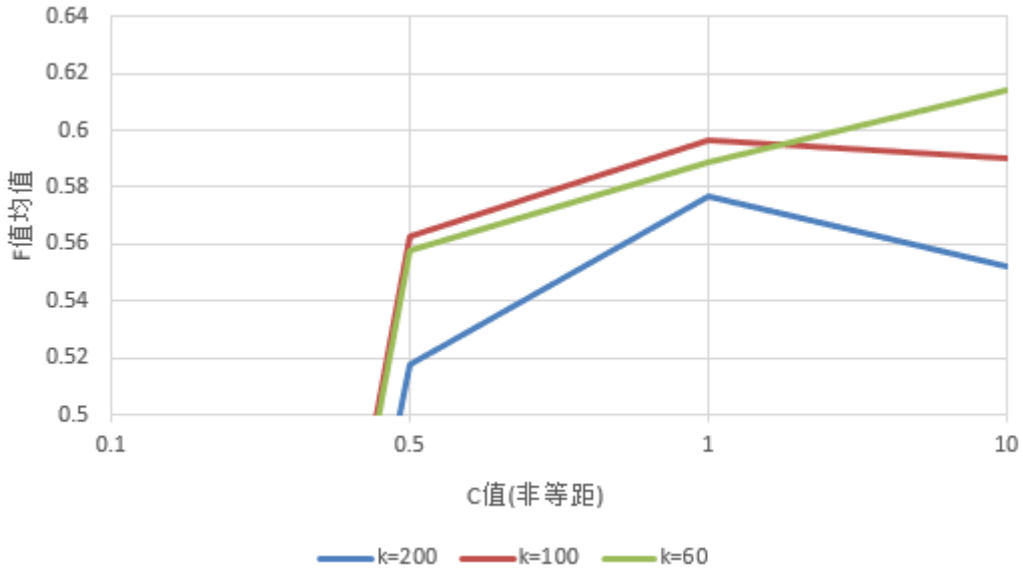


图 5-2 LDA-SVM 实验结果随参数变化趋势图（评测数据集）

LDA-SVM 方法在验证数据集 1 上的实验结果可以由图 5-3 直观地展示：

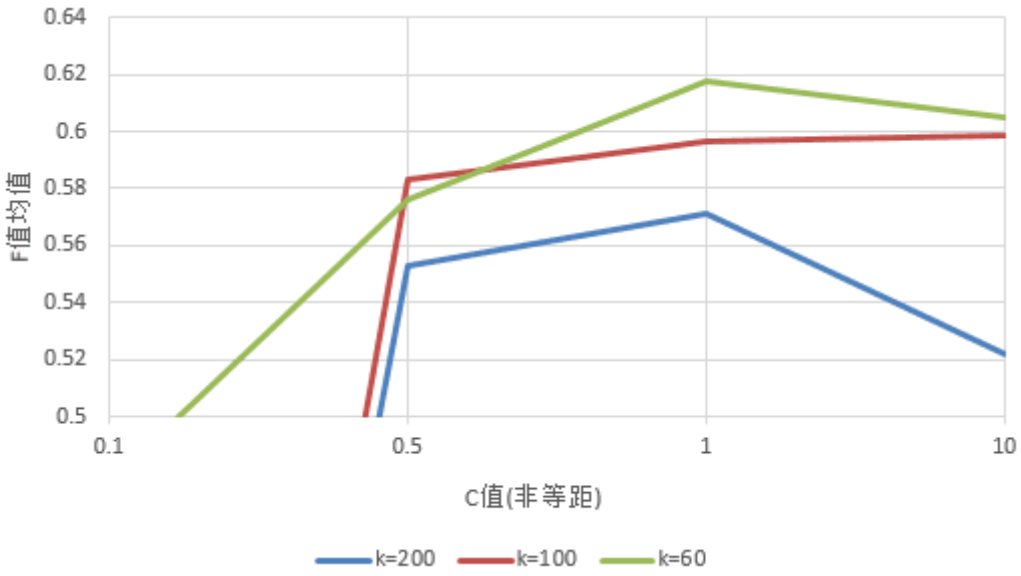


图 5-3 LDA-SVM 实验结果随参数变化趋势图（验证数据集 1）

LDA-SVM 方法在验证数据集 2 上的实验结果可以由图 5-4 直观地展示：

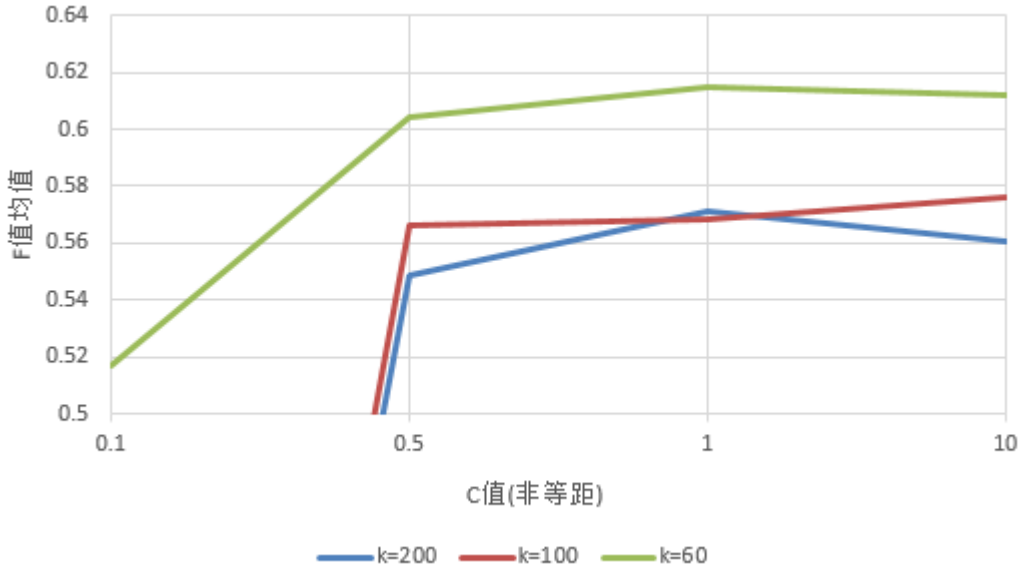


图 5-4 LDA-SVM 实验结果随参数变化趋势图（验证数据集 2）

C 值从左到右分别为 0.1,0.5,1,10。可以看到，主题数 k=60 时取得了最好的结果。从三个数据集上看，C=1 时的结果总体较好。

本文中，LDA 特征有着较差的表现，同时基于 LDA 特征的分类方法有着过

多的参数，难于调参。对于本节的 LDA-SVM 方法，与本章的 LSA-SVM 方法相比差距较大，但优于本章的基于 SL 特征的 SVM 分类方法。对于 LDA 表现较差的原因，本文在第六章给出了合理的猜测，分析和论证。

## 5.4 使用 SVM 对 SL 特征进行分类

### 5.4.1 参数设置

在本文中，SL 特征的参数 λ 分别取 0.01，0.1，1，10；使用线性核作为 svm 的核函数，其中惩罚因子 C 分别取 0.1,0.5,1,10。

### 5.4.2 在评测数据集上的实验

当 λ=0.01 时，实验结果如表 5-19 所示：

表 5-19 评测数据集上的 SL-SVM 实验结果（λ=0.01）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | | 0.6457 | 0.4482 | | | |
| C=0.5 | 0.4110 | 0.6457 | | 0.4482 | | |
| C=1 | 0.5582 | 0.6457 | | 0.4482 | | |
| C=10 | 0.5582 | 0.6466 | | 0.4449 | | |

当 λ=0.1 时，实验结果如表 5-20 所示：

表 5-20 评测数据集上的 SL-SVM 实验结果（λ=0.1）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | | 0.6457 | | 0.4482 | | |
| C=0.5 | 0.4110 | 0.6457 | | 0.4482 | | |
| C=1 | 0.4110 | 0.6457 | | 0.4482 | | |
| C=10 | 0.5582 | 0.6411 | | 0.4449 | | |

当 λ=1 时，实验结果如表 5-21 所示：

表 5-21 评测数据集上的 SL-SVM 实验结果（λ=1）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | | 0.6227 | | 0.4276 | | |
| C=0.5 | 0.4273 | 0.6227 | | 0.4276 | | |
| C=1 | 0.5726 | 0.6227 | | 0.4276 | | |
| C=10 | 0.5726 | 0.6227 | | 0.4383 | | |

当 λ=10 时，实验结果如表 5-22 所示：

表 5-22 评测数据集上的 SL-SVM 实验结果（λ=10）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | 0.3850 | 0.4165 | | 0.4244 | | |
| C=0.5 | 0.4470 | 0.4165 | 0.5700 | 0.4244 | | |
| C=1 | 0.4470 | 0.4165 | 0.5328 | 0.4244 | | |
| C=10 | 0.4470 | 0.4165 | 0.5328 | 0.4244 | | |

### 5.4.3 在验证数据集 1 上的实验

当 λ=0.01 时，实验结果如表 5-23 所示：

表 5-23 验证数据集 1 上的 SL-SVM 实验结果（λ=0.01）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | | | | | | |
| C=0.5 | 0.5324 | 0.4635 | | | | |
| C=1 | 0.5324 | 0.5404 | | 0.4817 | | |
| C=10 | 0.5324 | 0.5404 | | 0.5001 | | |

当 λ=0.1 时，实验结果如表 5-24 所示：

表 5-24 验证数据集 1 上的 SL-SVM 实验结果（λ=0.1）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | | | | | | |
| C=0.5 | 0.5497 | 0.4635 | | 0.3045 | | |
| C=1 | 0.5324 | 0.5405 | | 0.4466 | | |
| C=10 | 0.5553 | 0.5404 | | 0.5001 | | |

当 λ=1 时，实验结果如表 5-25 所示：

表 5-25 验证数据集 1 上的 SL-SVM 实验结果（λ=1）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | 0.3086 | | | | | |
| C=0.5 | 0.5500 | 0.4489 | | 0.4342 | | |
| C=1 | 0.5500 | 0.5278 | | 0.4391 | | |
| C=10 | 0.5500 | 0.5278 | | 0.4342 | | |

当 λ=10 时，实验结果如表 5-26 所示：

表 5-26 验证数据集 1 上的 SL-SVM 实验结果（λ=10）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | 0.4688 | | 0.4915 | | | |
| C=0.5 | 0.4688 | | 0.4829 | 0.4186 | | |
| C=1 | 0.5500 | 0.3510 | 0.4829 | 0.4186 | | |
| C=10 | 0.5500 | 0.4603 | 0.4829 | 0.4186 | | |

### 5.4.4 在验证数据集 2 上的实验

当 λ=0.01 时，实验结果如表 5-27 所示：

表 5-27 验证数据集 2 上的 SL-SVM 实验结果（λ=0.01）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | | | | | | |
| C=0.5 | 0.5177 | 0.5192 | | 0.4797 | | |
| C=1 | 0.5177 | 0.5192 | | 0.4386 | | |
| C=10 | 0.5177 | 0.5192 | | 0.4771 | | |

当 λ=0.1 时，实验结果如表 5-28 所示：

表 5-28 验证数据集 2 上的 SL-SVM 实验结果（λ=0.1）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | | | | | | |
| C=0.5 | 0.5299 | 0.4714 | | 0.4797 | | |
| C=1 | 0.5299 | 0.5192 | | 0.4386 | | |
| C=10 | 0.5177 | 0.4714 | | 0.4771 | | |

当 λ=1 时，实验结果如表 5-29 所示：

表 5-29 验证数据集 2 上的 SL-SVM 实验结果（λ=1）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | 0.3022 | | | | | |
| C=0.5 | 0.5481 | 0.3630 | | 0.4562 | | |
| C=1 | 0.5481 | 0.4955 | | 0.4562 | | |
| C=10 | 0.5481 | 0.5173 | 0.3588 | 0.4585 | | |

当 λ=10 时，实验结果如表 5-30 所示：

表 5-30 验证数据集 2 上的 SL-SVM 实验结果（λ=10）

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| C=0.1 | 0.4635 | | 0.4805 | | | |
| C=0.5 | 0.4635 | 0.4330 | 0.4676 | 0.4382 | | |
| C=1 | 0.5481 | 0.4330 | 0.4676 | 0.4382 | | |
| C=10 | 0.5481 | 0.4330 | 0.4676 | 0.4382 | | |

### 5.4.5 实验结果分析及方法评价

本节以情感词词典为基础，生成了文本的四个情感特征，并使用 SVM 进行分类。可以看到，在每个数据集上，在 C 的几个不同取值下，都没有能够得到完整的 5 个 target 的 $F_{AVG}$ 值；本节的分类方法是本章中表现最差的方法。这从一定程度上说明了情感分析方法不太适用于立场分析问题，也进一步证明了情感分析任务和立场分析任务有着较大区别。

## 5.5 本章小结

本章介绍了本文对于 LSA-SVM，LDA-SVM，SL-SVM 三种方法的实验工作。本文按照按照 75%训练集，25%测试集的比例重新划分了评测数据集，生成了用于对照和集成模型参数选择的验证数据集 1 和验证数据集 2。在 5.1 节，介绍了这三个数据集的详细统计信息，包括每个 target 的数据量，训练集和测试集词汇量，总词汇量，每个类别的数据数量等信息。在 5.2,5.3,5.4 节中，本文分别给出了 LSA-SVM，LDA-SVM，SL-SVM 的实验结果，并做出了结果分析和方法评价。

# 第6章　基于文本相似度的分组投票分类模型

本文提出了一种基于相似度计算的监督学习分类方法，称为基于文本相似度的分组投票模型（Similarity Based Group Voting Model，简称 GVM）。在本文的基分类器中，LSA-GVM 取得了最好的效果。本章将详细讲述算法的基本思想和基本原理，介绍算法如何实现，并给出本章方法在评测数据集和验证数据集上的实验结果。此外，本章还会对第 4,5 章提出的一些问题进行回答和总结。

## 6.1 微博文本的层次主题模型及生成模型

在第五章中，我们已经讲述过层次主题模型的思想。本文认为，本文所使用的特征是近似地将文本映射到了某一层主题空间，实际情况可能是特征中包含了多个主题层次中的一部分信息。特征中包含哪个层次的信息最多，我们就称将特征将文本近似映射到了哪一层主题空间。文本的立场也对应微博文本的某一层次主题。

本文认为一条微博文本是这样生成的：首先作者确定了文本所谈的话题（即本文所说的 target），确定话题之后可能会想到某些事情，然后由想到的事情生成了立场；也有可能作者在确定话题之后就有了自己的立场（FAVOR，AIAGINST 或 NONE）。不同的立场决定了作者可能会谈及不同的事情，也可能会直接发表自己的见解，提出赞同或反对意见，或者只是客观地评价一下。所以，由立场可能生成新一层级的主题，然后由主题继续生成新的主题或词语；也有可能直接由立场生成词语。这样文本的生成过程就对应了一个自上而下的树状层次主题模型。本文把层次主题模型中除了根节点及 target，立场之外的层次称为隐含层次。根节点位于第 0 层，target 位于第 1 层。定义第 0 层比第 1 层高。

从上面的生成过程我们可以看出，隐含层次既有可能存在于立场所在的主题层次之上，也有可能存在于主题层次所在的之下。在生成立场层次之前，我们在没有任何对于微博作者的先验知识的情况下，我们是无法推断作者立场的；换句话说，我们没有办法使用比立场层次高的隐含层次来推断立场。举个简单的例子，如果我们只知道作者要谈哪个 target，那我们无法推断作者对这个

target 的立场是什么，只能认为接下来作者会等概率地选择自己的立场。

基于上面的理论，我们可以容易地看出，如果特征 A 将文本近似映射到了高于立场的主题层次，那么使用相同的分类器分类效果将会明显低于将文本近似映射到低于立场主题层次的特征 B。对于这种情况，我们称特征 B 的效果高于特征 A。由于我们很难根据数据进行层次模型建模，也无法判断特征到底是把文本近似映射到了哪个具体层次以及该层次和立场层次关系如何，但是我们可以根据一些实验结果来推测特征 β 的效果是否高于特征 α 。考虑到分类器和特征组合时产生的一些微妙影响，本文建议，当不同特征使用同一分类器的结果有显著差异（特征 β 比特征 α 结果更好），或者多个分类器下特征 β 的实验结果都高于特征 α ，我们才认为特征 β 的效果高于特征 α 。

这个推论可以帮助我们进行特征选择，当我们使用多种特征和多种分类方法来构建一种使用单一特征和分类方法的高效分类器的时候，如果实验证实特征 β 的效果高于特征 α ，那我们可以丢弃特征 α ，减少组合验证所花费的时间。在第 5 章我们可以看到，使用线性核的 svm 分类器时，LDA 的效果明显低于 LSA 的效果，因此，当我们使用新的分类方法构建新的单一特征的分类器时，在 LDA 和 LSA 之间，就应该选择 LSA 作为特征。在本章第 7 节我们可以看到 LDA-GVM 的实验结果同样明显低于 LSA-GVM 的结果，证明了本节推论的正确性。

## 6.2 集成学习与投票分类器

投票分类器是集成学习的常见手段，本章所介绍的分类模型与集成学习中的投票分类器有着共同之处。本文将集成学习的思想引入了 GVM 是期望利用集成学习的优点来提高 GVM 的效果。

本章的 GVM 方法将训练数据分为多个组，每个组看做一个分类器，每个分类器将给出一个分类结果，然后根据某种策略将每个组的结果整合起来，得出统一的分类结果。

## 6.3 文本稀疏性与语义稀疏性问题

微博文本属于短文本，短文本的稀疏性问题为文本的表示或者说是特征工程带来了一定的难度。短文本的稀疏性会造成所选取的特征具有语义稀疏性。语义稀疏性是指文本在语义空间内的语义向量是稀疏的，这种稀疏性会为分类

带来不好的影响。

本章的 GVM 为了解决语义的稀疏性问题，将训练数据划分为若干组，将同组之内的数据连接成为一条或若干条数据。因为同组之内的数据的类别是一致的，所以连接后得到的新数据和原数据是相似的，这样我们就可以用新数据取代原数据。对一个类别进行分组时，本文采取了按顺序划分的策略。这样划分的数据在立场上有着相似语义，但在比立场低的主题层次上，可能并不十分相似，这一特性有助于减小新数据的语义稀疏性。

我们将一组内的包含的数据数量称为聚合量（也称组内数量）。聚合量增大，就意味着分组数量的减少。在 6.2 我们提到了本文希望通过引入集成学习的思想来提高分类效果，因此我们需要权衡“基分类器”的数量（即组数）和聚合量。

## 6.4 基于相似度计算的分组投票模型

在 6.2 和 6.3 介绍了分组和聚合的概念。接下来要讨论的就是“基分类器”的构造。前面我们已经说明了，同组内的数据将被聚合成若干条数据，我们使用这些数据给出一个分类结果。对于某个固定的聚合量，组内数据应该如何连接，使用什么样的分类方法又是我们需要权衡的问题。因为要进行基分类器的集成，所以我们希望连接策略简单，且分类方法复杂度低，以保证整体 GVM 的复杂度较低；同时我们还要考虑保证基分类器的效果。

一个最简单的连接策略就是组内数据全部连接成为一条。当组内数据为同一类别时，最容易想到的分类器是单分类器。单分类器会给出测试数据属于这个类别的概率或者判断数据属于或不属于这个类别。我们使用相似度来近似地代替概率，因为相似度和概率应该是成正比的，它保证了基分类器的准确率和合理性。

当每个基分类器给出了测试数据属于某个类别的概率的时候，我们就可以对结果进行集成了。我们考虑将每个类别的基分类器产生的“概率值”累加起来，最终哪个类别数值最大，即判定为哪个类别，这种情况下，“概率值”即为投票的权重。

如果训练数据集是非均衡的，那么全部基分类器参与投票很有可能出现“后来居上”的现象，即与测试数据最为相似的几组数据都是和测试数据同类

别的，但是由于该类别数据较少，导致投票值总和小于其他类别。对于这种非均衡训练数据集导致的分类错误，我们选择取相似度排名最高的 k 个基分类器投票来解决。

虽然连接操作一定程度上缓解了语义稀疏性，但并没有从根本上解决。我们不能只选择相似度最高的一组来进行投票，因为语义稀疏性以及一些其他原因有一定可能会导致相似度最高的一组数据和测试数据不是同一类别的。假设测试数据属于 FAVOR 类别，语义稀疏性会导致这条测试数据与一部分类别为 FAVOR 的训练数据较为相似，并且与一部分 AGAINST 类别的数据较为相似。由于语义的稀疏性，各个基分类器按照给出的相似度从高到低排列，其类别分布可能是：[FAVOR-0.7,FAVOR-0.6,AGAINST-0.58, AGAINST-0.57, AGAINST-0.55, AGAINST-0.5, AGAINST-0.5, FAVOR-0.49, FAVOR-0.48, FAVOR-0.47,NONE-0.2,NONE-0.15,NONE-0.1]。这种情况下我们选取 k=3,4,10,11 时结果都是正确的，而选择 k=5,6,7 时结果都是错误的。

综上所述，当固定分组，k 从 1 逐渐增大时，由于集成学习的优点和语义稀疏性的缺点，准确率往往先逐渐增大，在增大的过程中会有局部波动，在 k 达到一定值时由于非均衡数据集等原因开始有较为明显的下降，在这个点之后随着 k 增大而准确率将总体呈较为明显的下降趋势。

因此，对于选好 k 值对于分类效果十分重要，然而实验表明取不同分组数时最好的 k 值并没有明显的规律，只能在一定区域内选择一个较好 k 值。较好 k 值的平均结果往往比最好 k 值低 2 至 3 个百分点。于是，我们希望能有一种方法能够让我们便捷地选取 k 值（比如不管如何分组都取某个固定值等），同时使得分类的平均效果能够接近或达到原来最好 k 值的效果。本文解决了这个难题，相关工作将在 6.5 节详细介绍。

## 6.5 使用 VWR（Voting weight refine）函数改善投票效果

通过前几节的介绍，可以了解到，GVM 中有两个需要确定的参数，一是分组数 gn，投票数量 k。为了进一步降低模型的参数调优难度，我们希望有一种方法可以使得 k 便于选取，其中一种方案就是令 k 等于基分类器的数量（即组数）。在 6.4 提到的方法中，由于非均衡数据集等原因的影响，取 k=gn 并不会得到很好的效果，因此我们考虑通过降低低相似度分组的投票权重并提高高相

似度分组的投票权重来解决这个问题。出于这个考虑，本文提出了一个类 Sigmoid 函数来改变投票权重，我们称之为 VWR 函数，如式（6-1）所示：

$$VWR(w)=\frac{1}{1+e^{6-10\times w}} \tag{6-1}$$

其中 w 为原始投票权重，VWR(w)为新的投票权重。

至此，本文已经介绍完 GVM 的全部细节。下面我们将给出使用 VWR 函数和不使用 VWR 函数时的部分实验结果。

以 8 条数据为 1 组，在评测数据集上的实验结果随投票数量的变化如图 6-1 所示（其中蓝色曲线代表没有使用 VWR 函数，红色曲线代表使用了 VWR 函数，纵坐标为 5 个 target 的$F_{AVG}$的平均值，横坐标为投票数量 k）：

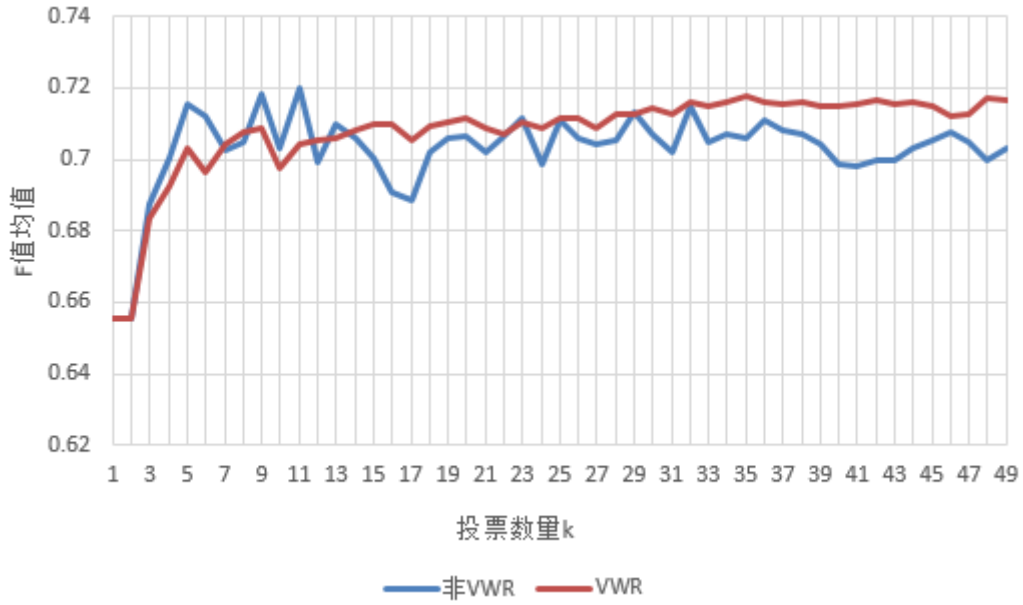


图 6-1 非 VWR 方法与 VWR 方法比较（组内数量 n=8）

以 10 条数据为 1 组，在评测数据集上的结果随投票数量的变化如图 6-2 所示：

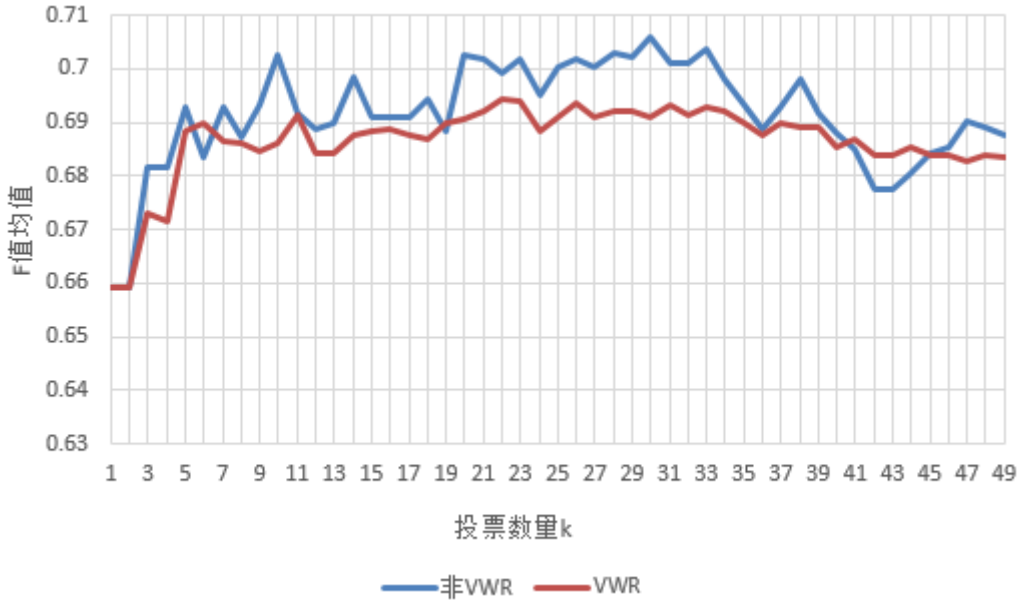


图 6-2 非 VWR 方法与 VWR 方法比较（组内数量 n=10）

以 15 条数据为 1 组，在评测数据集上的结果随投票数量的变化如图 6-3 所示：

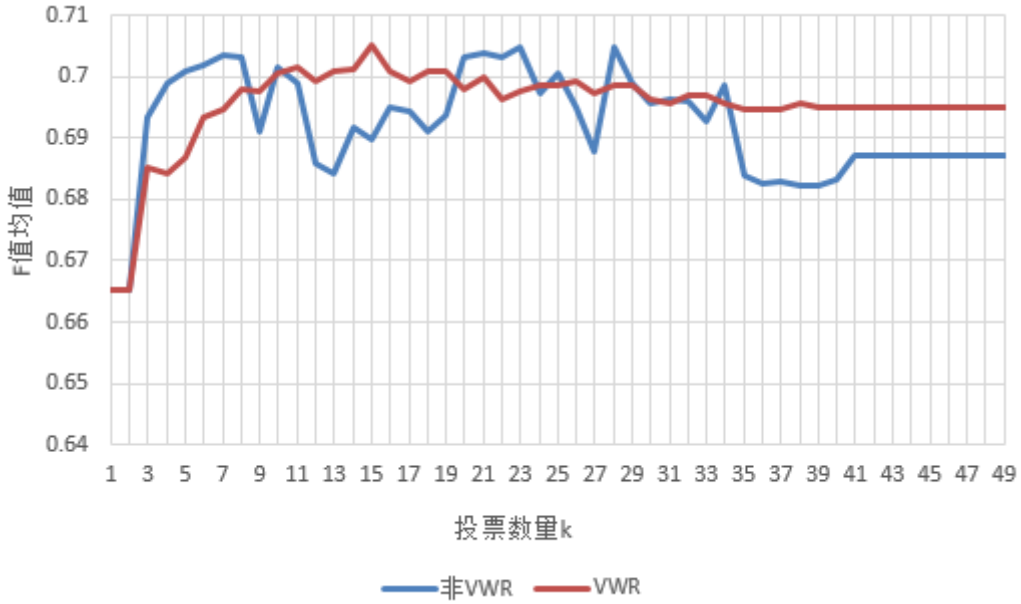


图 6-3 非 VWR 方法与 VWR 方法比较（组内数量 n=15）

以 20 条数据为 1 组，在评测数据集上的结果随投票数量的变化如图 6-4 所示：

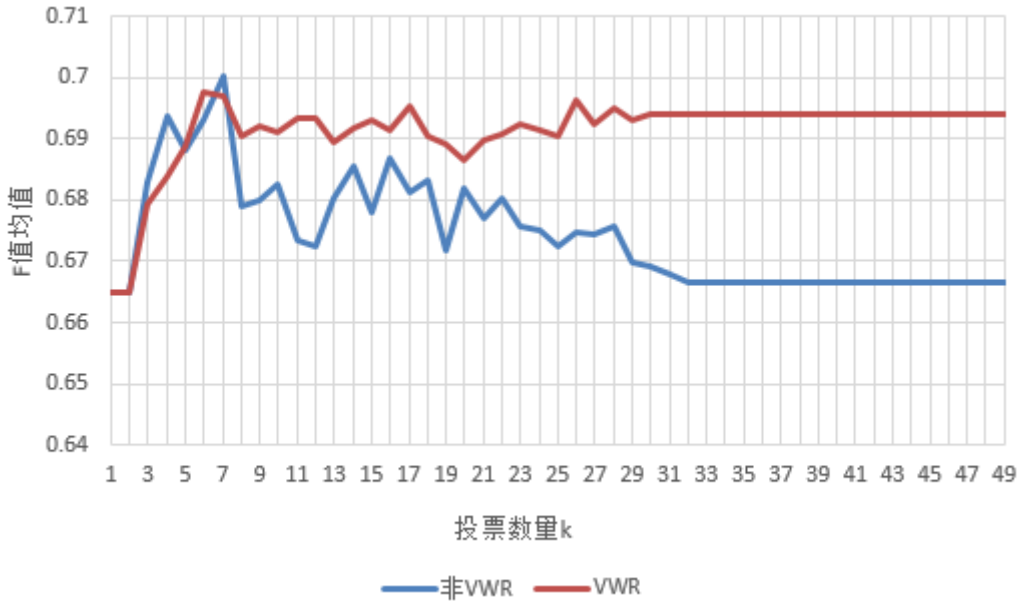


图 6-4 非 VWR 方法与 VWR 方法比较（组内数量 n=20）

以 30 条数据为 1 组，在评测数据集上的结果随投票数量的变化如图 6-5 所示：

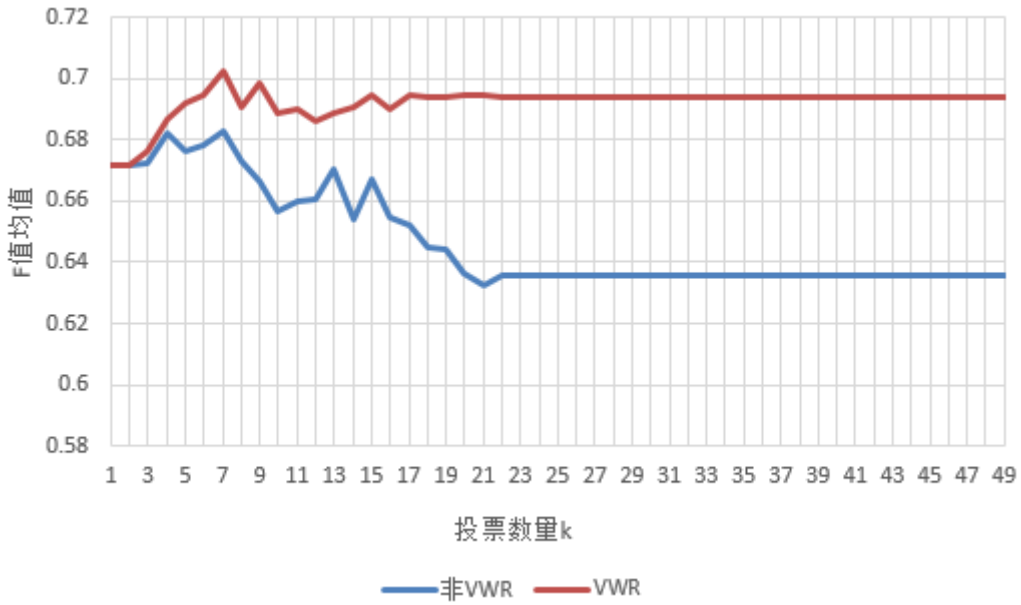


图 6-5 非 VWR 方法与 VWR 方法比较（组内数量 n=30）

通过上面的图例可以看到，当不使用 VWR 时，在不同分组下 F 值均值取得峰值时对应的 k 没有规律，并且 F 值均值在峰值附近波动很大，因此我们难以找到一个合适的策略来选择不同分组下的 k 值，使得 GVM 模型在不同分组下取得足够好的平均效果。

VWR 函数极大地改善了 F 值均值随着投票数量的增加而下降的问题，解决了局部 F 值均值随 k 变化波动过大的问题，在绝大多数情况下提高了 GVM 模型的整体效果（也存在反例，见图 6-2）。

因此，VWR 函数使得我们在不同分组下选取大部分 k 值时均可获得较好的效果。甚至在部分分组下，VWR 函数使得 GVM 模型的效果在取不同 k 值时都有了全面的提升（例如图 6-5 表示的情况）。于是我们便可以使用简单的策略便捷地选取 k 值，在 6.6 的实验中，本文选取了 k=20，k=50，k=gn 三组取值在不同分组下进行了实验。

## 6.6 基于 LSA 的 GVM 实验及方法评价

前面已经介绍过 LSA 及 GVM 的细节，本节将先给出 LSA-GVM 分类器的完整算法流程，随后给出 LSA-GVM 在评测数据集，验证数据集 1 和验证数据集 2 上的实验结果。

LSA-GVM 算法流程如下：

1. 对原始微博数据进行通用数据预处理。
2. 假设训练集中 FAVOR，AGAINST，NONE 三个类别的数据数量分别为 tf,ta,tn。选定组内数量 n，对训练集的 FAVOR，AGAINST，NONE 三个类别的数据分别进行划分，从上到下每 n 个为 1 组，未划分的数据不足 n 个时，直接归为归为一组，划分完毕。这样就得到了$\lceil tf/n \rceil + \lceil ta/n \rceil + \lceil tn/n \rceil$个分组。
3. 将每个分组内的微博文本连接成为一条数据，这样就得到了$\lceil tf/n \rceil + \lceil ta/n \rceil + \lceil tn/n \rceil$条新训练数据。
4. 将得到的$\lceil tf/n \rceil + \lceil ta/n \rceil + \lceil tn/n \rceil$个新训练数据作为 LSA 的建模输入。计算基于 TFIDF 的词语-文档矩阵，并对矩阵进行 SVD 分解。
5. 使用式（4-4）来计算每条数据在语义空间内的坐标，或者说是语义向量，选取的奇异值个数为 gn=$\lceil tf/n \rceil + \lceil ta/n \rceil + \lceil tn/n \rceil$。
6. 对每条测试数据，计算它的语义向量和每条新训练数据的语义向量的余弦相似度 w，并计算出投票权重 VWR(w)。
7. 使用如下方法判断每条测试数据所属类别：每条新训练数据使用它对应的权重进行投票，票数最高的类别即为 LSA-GVM 方法预测的测试数据所属类别。

分别取 k=20，k=50，k=gn；LSA-GVM 在评测数据集上的结果随组内数据数量的变化如图 6-6 所示(横坐标为组内数量 n，n 从 7 开始，纵坐标为 5 个 target 的$F_{AVG}$的平均值，下文中称为 F 值均值或$F_{AVG}$均值)：

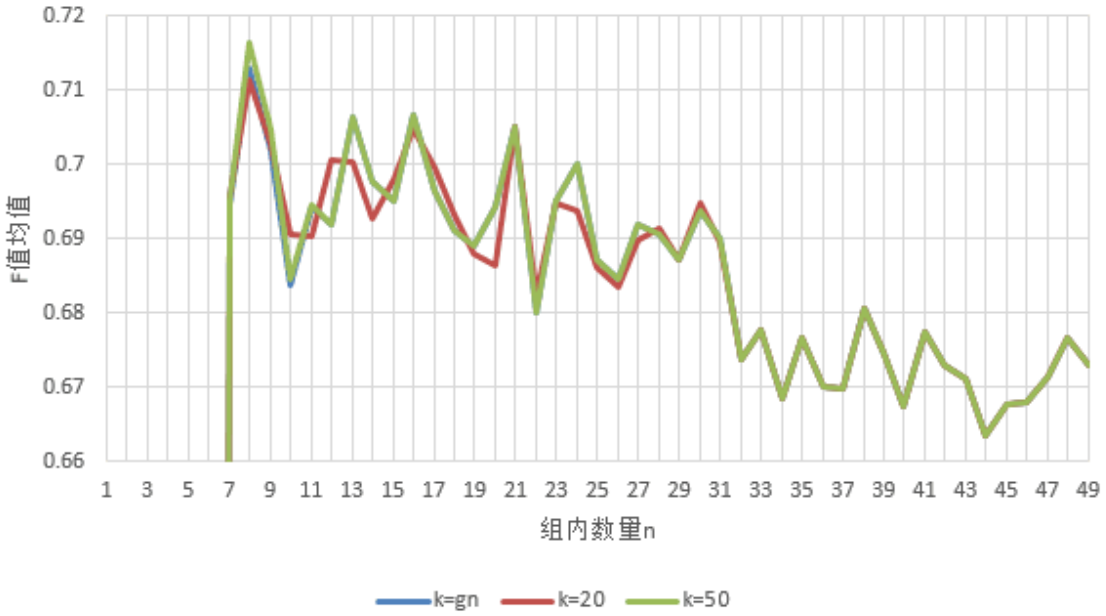


图 6-6 评测数据集上 3 种投票数量下 F 值均值随组内数量 n 的变化

分别取 k=20，k=50，k=gn；LSA-GVM 在验证数据集 1 上的结果随组内数据数量的变化如图 6-7 所示：

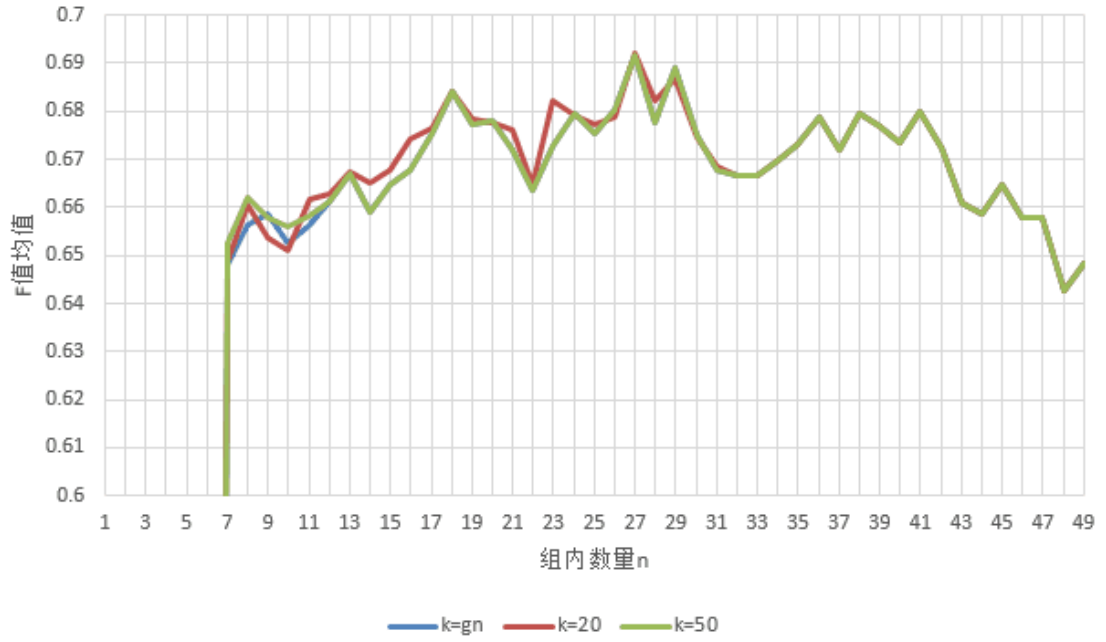


图 6-7 验证数据集 1 上 3 种投票数量下 F 值均值随组内数量 n 的变化

分别取 k=20，k=50，k=gn；LSA-GVM 在验证数据集 2 上的结果随组内数据数量的变化如图 6-8 所示：

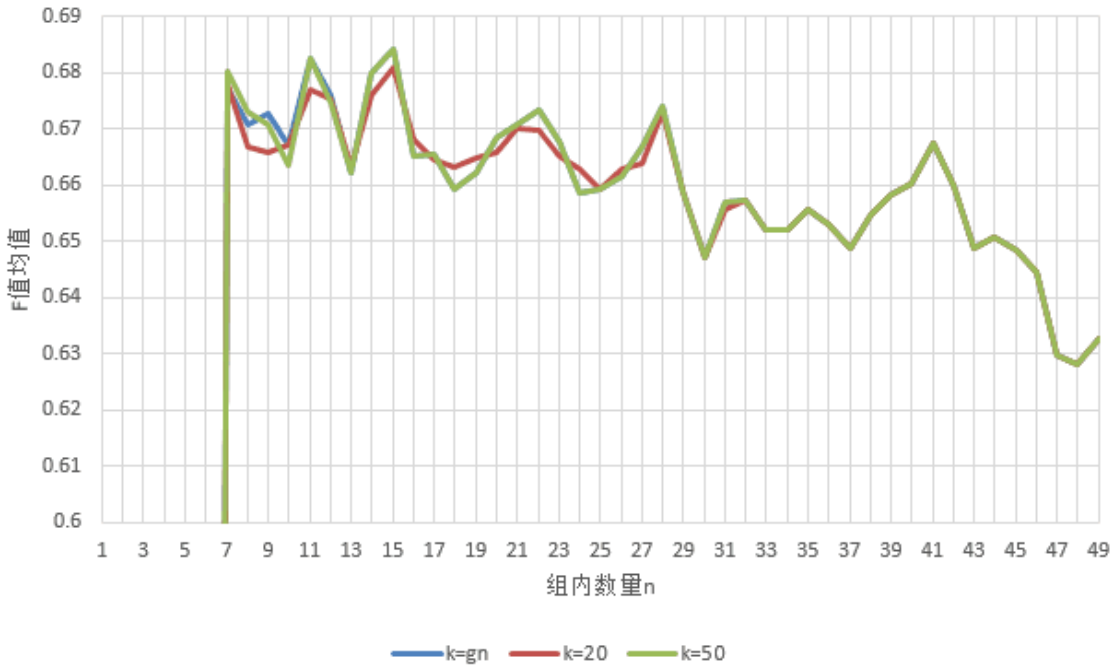


图 6-8 验证数据集 2 上 3 种投票数量下 F 值均值随组内数量 n 的变化

LSA-GVM 在不同数据上表现不同的原因主要与测试集中未登录词的数量和比例有关，未登录词越多，比例越高，测试结果就会越差。通过表 5-1，表 5-2，表 5-3 我们可以算出评测数据集，验证数据集 1 和验证数据集 2 中的未登录词比例分别为 35.83%，36.48%，37.32%；平均每个 target 中未登录词数量分别为 722.2,835.2,857.2。我们对比图 6-6，图 6-7，图 6-8 可以验证，在评测数据集，验证数据集 1 和验证数据集 2 上，最高的$F_{AVG}$均值分别约为 0.716，0.691，0.684；平均的$F_{AVG}$均值也表现为评测数据集上最高，验证数据集 1 次之，验证数据集 2 最低，符合上述规律。

我们还可以看到，在三个数据集上，LSA-GVM 的组内数量 n 的较优参数取值区间为[7,30]。

事实上，LSA-GVM 的$F_{AVG}$均值曲线应该是是呈先升后降的总体趋势的，由于 GVM 的连接特性，决定了当组内数量 n 过小时 LSA-GVM 的$F_{AVG}$均值较低(此外 n 较小时词语-文档矩阵较大，受 SVD 分解效率的限制，LSA-GVM 的运行效

率也会变低)。由于 GVM 的集成特性，当组内数量 n 过大时，基分类器的数量过小，会对 LSA-GVM 的效果有很大的不良影响。因此，GVM 的连接和集成特性综合决定了其$F_{AVG}$均值曲线先升后降的总体趋势。限于运行效率等原因，本文没有给出 n<7 时的实验结果，因此我们无法从图 6-6 和 6-8 中看出这种趋势，但这一点我们可以从图 6-7 中得到验证。

从图中我们还可以观察到，在不同的数据集上，使得$F_{AVG}$均值达到峰值的组内数量 n 同样没有太大规律，并且在峰值附近同样有较大波动，因此，我们根据数据集 A 调整的参数很可能并不适用于数据集 B。关于 GVM 如何选取组内数量 n，本文建议使用多个数据集进行验证来选取一个较优参数区间并在这个较优区间内选取参数。同时根据 GVM 的连接和集成特性以及实验经验，本文建议组内数量 n 不要小于 5，组内数量 n 不要大于 1.5 倍的组数 gn。在第 7 章，本文将多个取不同组内数量 n 的 LSA-GVM 集成起来，以期提高 LSA-GVM 的平均效果，相关内容详见 7.1。

本文所提出的 LSA-GVM 方法包含三个参数：LSA 中所取奇异值的个数 q，GVM 中组内数量 n 以及投票数 k。根据经验，奇异值个数 q 固定为组数 gn。因此，我们在使用 LSA-GVM 方法时只需选择 n 和 k 两个参数，由于 VWR 函数的贡献，我们可以在 10 到 gn 范围内较为随意地选取 k 值，而对于参数 n，本文在本节中也已经给出了合理的选择策略。

相比于 LDA 特征和 SVM 方法，LSA-GVM 的参数都是离散化的数值，且参数个数很少，当我们对模型参数进行调优时，LSA-GVM 完全可以遍历参数空间，使得我们对各个参数的影响能有更为直观的认识。由于 LSA-GVM 将组内数据连接成为了一条数据，使得它和 LSA-SVM 相比，运行效率提高了很多。此外，第 5,6 章的实验证明，LSA-GVM 方法是本文效果最好的基分类器。综上所述，本文所提出的 LSA-GVM 是一个参数较少，参数选取较为简单，运行效率较高，分类结果较为准确的模型。

## 6.7 基于 LDA 的 GVM 实验及方法评价

我们将 LSA-GVM 算法中基于 LSA 的相似度计算改为基于 LDA 的相似度计算，即可得到 LDA-GVM 算法。

本节取 5.3 中 topic 数量为 200 的一组参数设置，分别令 GVM 中的组内

数量 n=8,10,15,20,30，投票数量 k=50，在评测数据集,验证数据集 1 和验证数据集 2 上进行了实验。实验结果分别如表 6-1，表 6-2，表 6-3 所示：

表 6-1　LDA-GVM 在评测数据集上的实验结果

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| n=8 | 0.3351 | | | 0.3719 | 0.0757 | |
| n=10 | 0.2017 | | | 0.3735 | 0.0750 | |
| n=15 | 0.1538 | | | 0.3541 | 0.0820 | |
| n=20 | | | | 0.3147 | 0.0244 | |
| n=30 | 0.0682 | | | 0.2958 | | |

表 6-2 LDA-GVM 在验证数据集 1 上的实验结果

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| n=8 | 0.4624 | 0.1559 | 0.1394 | 0.4747 | 0.3834 | 0.32316 |
| n=10 | 0.3796 | 0.2814 | 0.1566 | 0.4536 | 0.4679 | 0.34782 |
| n=15 | 0.1549 | 0.1724 | | 0.3798 | 0.2444 | |
| n=20 | 0.1647 | 0.1825 | | 0.3052 | 0.1621 | |
| n=30 | 0.0482 | 0.1517 | | 0.2725 | 0.1663 | |

表 6-3 LDA-GVM 在验证数据集 2 上的实验结果

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| n=8 | 0.4674 | | | 0.4462 | 0.2875 | |
| n=10 | 0.4359 | 0.1832 | | 0.5242 | 0.4178 | |
| n=15 | 0.2436 | 0.1527 | | 0.3907 | 0.2656 | |
| n=20 | 0.1351 | | | 0.3516 | 0.1778 | |
| n=30 | 0.0470 | 0.1699 | | 0.3441 | 0.1902 | |

可以看到 LDA-GVM 的效果很差，其原因可能有多种：LDA 调参困难，难以选出合适的参数，且主题数量的选择对结果有较大影响，本文没有选取出合适的参数；LDA 将文本近似映射到了高于立场主题的主题层次，特征本身不适合用于推断立场，并且可能因此导致了文本的连接引入的噪声多于有用信息；停

止词词典包含的词汇过于广泛导致一些关键词缺失等。

事实上在很多实践中证明了 LDA 确实不太适合进行短文本分类[21]。LDA 比较适用于提取一些粒度较粗的隐含主题（也就是说适用于提取层次主题模型中层次较高的主题），这样就导致了 LDA 更不太适合微博短文本的立场分析。关于 LDA 在短文本分类中的应用，可以参考张志飞等在[22]中的工作以及方东昊在[23]中的工作。这两篇文章都是对文本进行大类（如教育，经济，军事）划分，与本文的立场分析任务不同，提取粗粒度的隐含主题即可较好地解决这些问题。

一些 LDA 在短文本处理的扩展模型可以参考 Lacoste-Julien S 关于 DiscLDA 的工作[24]以及 Zhu J 关于 MedLDA 的工作[25]。

## 6.8 本章小结

本章首先在 6.1 节中详细讲述了本文对于微博的层次主题模型假设，以及在这个假设下微博文本是如何生成的，并基于这个假设给出了一种特征选择策略同时做出了 LDA 特征在 GVM 方法中表现同样不佳的推断，本章 6.7 节的实验结果证明了这个推断的正确性。接下来本章的 6.2 和 6.3 节分别讨论了 GVM 的两个基本策略：投票和连接，包括策略提出的原因和具体实现方法等。在 6.4 节，本文详细描述了一个基本的分组投票模型，并阐述了本文的分组投票模型是基于相似度计算的。随后在 6.5 节中，本文提出了投票权重改善函数（VWR 函数）解决了投票数量 k 难以选择的问题，使得 GVM 模型得到了很大的提升和改善。在 6.6 节中，本文总结了前面对于 GVM 的介绍，给出了完整的使用 LSA 特征进行分类的 GVM 算法（即 LSA-GVM 算法）流程，并给出了 LSA-GVM 算法在评测数据集，验证数据集 1，验证数据集 2 上的实验结果，分析了实验结果并作出了方法评价。最后，在 6.7 节，本文给出了 LDA-GVM 的实验结果，并对实验结果进行了分析。

# 第7章　模型集成

本章将介绍本文的第二章所提出的两个集成算法框架的细节，给出它们在评测数据集上的实验结果，并与使用相同评测数据集的工作进行比较，对结果做出分析和评价。

## 7.1 同质集成框架:LSA-GVM 的集成（Ensemble LSA-GVM）

在第六章，我们提到了为 LSA-GVM 的组内数量 n 选取一个合适的取值是 LSA-GVM 的重点和难点。虽然我们给出了一个合理的参数选择策略，但是 LSA-GVM 分类模型仍有提升的空间。因此，我们将 LSA-GVM 方法集成起来，以期进一步提高模型的平均效果。我们将集成后的分类方法称为 Ensemble LSA-GVM 方法。为了降低 Ensemble LSA-GVM 方法的复杂度，本文提出了一种 LSA-SVM 的快速集成方法，称为 Fast Ensemble LSA-GVM 方法。

### 7.1.1 LSA-GVM 的集成策略与参数选择

考虑到 2.1 中提到的平权投票的优点,本文选择使用平权投票对多个 LSA-GVM 进行集成。Ensemble LSA-SVM 模型的算法流程如下：

1. 根据标注数据按照 75%训练集，25%测试集比例生成 t 个验证数据集
2. 对于这 t 个验证数据集进行 LSA-GVM 实验
3. 根据实验结果选取最优参数区间，并且令该区间在$[5,1.5\times n1]$范围内，n1 为根据 6.6 中的参数选择策略确定的最大最内数量 n1。设训练集数据量为 N，由$\mathrm{n1}=1.5\times gn1$且$\mathrm{N}=n1\times gn1$，解得$gn1=\sqrt[2]{N/1.5}$，$n1=1.5\times\sqrt[2]{N/1.5}$。最优区间可以按如下方式选取：求每个验证数据集上 top m 的$F_{AVG}$均值对应的组内数量 n，取最小和最大的 n 值作为区间的左右边界，并和$[5,1.5\times n1]$求并集，得到最终的最优参数区间。
4. 使用训练数据集训练 LSA-GVM 模型，取最优参数区间中的每个整数作为 LSA-GVM 的参数，对测试集进行分类，每条测试数据得到 rn 个分类结果，rn 为最优参数区间内整数的个数。
5. 对每条测试数据，使用 rn 个分类结果进行平权投票，得票最高的分类即为 Ensemble LSA-GVM 方法的分类结果。

模型中最优参数区间的选择策略并不唯一，本文仅给出了一个简单的选择策略。此外，验证数据集的生成方法也不唯一，可以采取随机生成，等距选取等方法，不同的生成方法对 Ensemble LSA-SVM 模型的影响没有统一的倾向性，且影响不大。

上述算法中需要选择 t 和 m 两个参数，且选择最优参数区间的过程很耗时。我们可以考虑直接使用$[5,1.5\times n1]$的子区间作为最优参数区间，当我们选择使用$[5+(1.5\times n1-5)\times p/2,(1-p/2)\times 1.5\times n1+2.5\times p]$作为最优参数区间时，就得到了一个快速的 LSA-GVM 的集成算法，我们称之为 Fast Ensemble LSA-GVM 算法，其中$0\leq p<1$，p 表示对原区间的缩减比例。

### 7.1.2 Fast Ensemble LSA-GVM 方法在评测数据集上的实验

本文选取了 4 组不同的 p 和 k（投票数量）进行了实验，实验结果如表 7-1 所示：

表 7-1 Fast Ensemble LSA-GVM 方法在评测数据集上的实验结果

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| p=0.2,k=50 | 0.5670 | 0.5567 | 0.8019 | 0.7801 | 0.7729 | 0.6958 |
| p=0.2,k=20 | 0.5655 | 0.5533 | 0.8005 | 0.7801 | 0.7746 | 0.6948 |
| p=0.4,k=50 | 0.5664 | 0.5569 | 0.7995 | 0.7798 | 0.7735 | 0.6952 |
| p=0.4,k=20 | 0.5656 | 0.5527 | 0.7991 | 0.7802 | 0.7760 | 0.6947 |

### 7.1.3 NLPCC 2016 Shared task 4 参与团队在评测数据集上的实验结果

NLPCC 2016 Shared task 4 参与团队在评测数据集上的实验结果如表 7-2 所示[1]：

表 7-2 NLPCC 2016 Shared task 4 参与团队在评测数据集上的实验结果

| Team ID | OVERALL | | | Target-1 | Target-2 | Target-3 | Target-4 | Target-5 |
|---|---|---|---|---|---|---|---|---|
| | $F_{FAVOR}$ | $F_{AGAINST}$ | $F_{AVG}$ | $F_{AVG}$ | $F_{AVG}$ | $F_{AVG}$ | $F_{AVG}$ | $F_{AVG}$ |
| RUC_MMC | 0.6969 | 0.7243 | 0.7106 | 0.7730 | 0.5780 | 0.5814 | 0.8036 | 0.7652 |
| TopTeam | 0.6601 | 0.7186 | 0.6894 | 0.7749 | 0.5764 | 0.5232 | 0.7661 | 0.7949 |
| SDS | 0.6758 | 0.6965 | 0.6861 | 0.7784 | 0.5852 | 0.5332 | 0.7948 | 0.6883 |
| CBrain | 0.6618 | 0.7094 | 0.6856 | 0.7604 | 0.5528 | 0.4787 | 0.8135 | 0.7835 |
| nlp_polyu | 0.6476 | 0.6870 | 0.6673 | 0.7352 | 0.5312 | 0.5584 | 0.7708 | 0.7090 |
| Scau_SDCM* | 0.6304 | 0.7027 | 0.6666 | 0.7033 | 0.5493 | 0.5780 | 0.7639 | 0.7138 |
| NEUDM | 0.6268 | 0.6858 | 0.6563 | 0.7173 | 0.5485 | 0.5240 | 0.7497 | 0.7052 |
| Printf | 0.6183 | 0.6702 | 0.6443 | 0.7048 | 0.5769 | 0.5547 | 0.7150 | 0.6417 |
| CQUT_AC996 | 0.5897 | 0.6557 | 0.6227 | 0.7015 | 0.4646 | 0.5280 | 0.7661 | 0.5879 |
| March* | 0.5858 | 0.6244 | 0.6051 | 0.6950 | 0.5466 | 0.4906 | 0.6442 | 0.6169 |
| BIT_NLP_FC* | 0.5573 | 0.5833 | 0.5703 | 0.7444 | 0.3460 | 0.3769 | 0.5888 | 0.4195 |
| HLJUNLP | 0.4584 | 0.6729 | 0.5656 | 0.5281 | 0.4494 | 0.5126 | 0.7553 | 0.4355 |
| CIST-BUPT | 0.4660 | 0.6136 | 0.5398 | 0.4754 | 0.4579 | 0.5003 | 0.6867 | 0.5048 |
| Lib1010 | 0.4636 | 0.4944 | 0.4790 | 0.4551 | 0.4420 | 0.4934 | 0.4946 | 0.5045 |
| USCGreenTree* | 0.3609 | 0.5904 | 0.4756 | 0.4799 | 0.4052 | 0.4586 | 0.5288 | 0.3871 |
| SCHOOL | 0.3329 | 0.4662 | 0.3995 | 0.3422 | 0.4222 | 0.3903 | 0.4613 | 0.3676 |

其中 Target-1 指“春节放鞭炮”，Target-2 指“IphonSE”，Target-3 指“俄罗斯在叙利亚的反恐行动”，Target-4 指“开放二胎”，Target-5 指“深圳禁摩限电”。

### 7.1.4 对本文同质集成框架的评价

通过表 7-1 和表 7-2 可以看到，本文所提出的 Fast Ensemble LSA-GVM 方法在评测数据集上取得了很好的实验结果，比 NLPCC 2016 shared task 4 参与团队中第二名的平均$F_{AVG}$高约 0.5%个百分点。此外，Fast Ensemble LSA-GVM 很好地解决了参数不易选取的问题，使得不同参数下的结果更为稳定，并且提高了模型的平均效果。当然，本节的集成学习框架也有着一定的缺点：各个基分类器仅模型参数不同，具有较小的差异性，因而集成时较难取得效果提升，集成效果较为依赖基学习器的效果。

在[1]中，Xu R 等提到了取得 NLPCC 2016 shared task 4 评测第一名的队伍(RUC_MMC)使用了不同的方法对于 5 个 target 分别建立了不同的分类模

型，且这对该队伍取得较高的成绩有着极其重要的作用。因此，RUC_MMC 提出的方法有着很大的局限性，在实际应用中，我们不大可能去为每个 target 分别进行建模，而本文提出的 Fast Ensemble LSA-GVM 方法是一种高效，稳定，快速的算法，较为适合小数据集上的短文本立场分析任务，可以应用在真实的监督学习立场分析任务中，取得较好的效果。

## 7.2 非同质集成框架:LDA-SVM,LSA-SVM, LSA-GVM 的集成

### 7.2.1 LDA-SVM,LSA-SVM,LSA-GVM 的集成策略

本文集成算法框架的集成策略如下：

1. 使用标注数据按照 75%训练集，25%测试集比例生成 t 个验证数据集
2. 分别使用 LDA-SVM，LSA-SVM，LSA-GVM 对每个验证数据集进行实验，选取出令验证数据集测试结果较好的一组参数 Para，并得到与之对应的每种方法在每个 target 上的$F_{AVG}$值。本文用$F_{AVG}$值来构建权重矩阵 W。设$F_i$["LDA − SVM"]["开放二胎"]表示在第 i 个验证数据集上，参数组为 Para 的条件下，LDA-SVM 方法对于“开放二胎”这个 target 取得的$F_{AVG}$值，则 W 的计算方法如式 7-1 所示：

$$W = \frac{1}{t} \times \sum F_i \tag{7-1}$$

3. 令基分类器的参数组为 Para，使用训练数据集分别训练 LDA-SVM，LSA-SVM，LSA-GVM，并使用训练得到的分类模型对测试数据集进行分类，每条数据得到三个分类结果。
4. 根据每种方法在对应 target 上的权重和分类结果进行加权投票，得到最终的分类结果。

### 7.2.2 本文的非同质集成方法在评测数据集上的实验

本文使用验证数据集 1 和验证数据集 2 来选择参数，对评测数据集进行了实验。本文由第 5,6 章的实验结果得到：LDA-SVM 模型中，令主题数为 60，α=5/6，β=0.01，SVM 使用了线性核函数，其参数 C=1；LSA-SVM 模型中，SVM 使用了线性核函数，其参数 C=0.05；LSA-GVM 中令投票数 k=20，组内数量 n=15。

集成后的效果与各基分类器的实验结果如表 7-3 所示：

表 7-3 本文的非同质集成框架及各基分类器在评测数据集上的实验结果

| | IphoneSE | 俄罗斯在叙利亚的反恐行动 | 开放二胎 | 春节放鞭炮 | 深圳禁摩限电 | Total-AVG |
|---|---|---|---|---|---|---|
| LDA-SVM | 0.5311 | 0.4293 | 0.6437 | 0.7456 | 0.5937 | 0.58868 |
| LSA-SVM | 0.5264 | 0.4982 | 0.7869 | 0.7914 | 0.7035 | 0.6613 |
| LSA-GVM | 0.5698 | 0.5479 | 0.8123 | 0.7879 | 0.7709 | 0.6978 |
| Ensemble | 0.5158 | 0.5062 | 0.8145 | 0.8084 | 0.7102 | 0.67102 |

### 7.2.3 对本文非同质集成框架的评价

本文提出的非同质集成框架与本文的同质集成框架相比实验结果稍差，其在评测数据集上的实验结果与 NLPCC 2016 shared task 4 参与队伍相比可排在第五名。

从表 7-3 中可以看出，在“开放二胎”和“春节放鞭炮”两个 target 上，本文的非同质集成框架确实带来了效果的提升，证明了集成学习理论和方法的正确性，对于集成学习在文本分类中的应用有着借鉴意义。而在其他三个 target 上集成之后的效果没有提升，其原因可能是多方面的：基分类器的差异性较小；各个基分类器的表现差异过大；投票方式有待改善；基分类器数量过少等等。

## 7.3 本章小结

本章详细介绍了本文在第二章所提出的两个集成学习框架，包括集成框架的集成策略，实验结果等内容。

在 7.1 节中，讲述了本文同质集成框架的集成策略，并给出了一个简化的 Fast Ensemble LSA-GVM 方法，实验证明，本文所提出的 Fast Ensemble LSA-GVM 方法是一种简单，稳定，高效的立场分析方法，其在评测数据集上的实验结果与 NLPCC 2016 shared task 4 参与队伍相比可排在第二名，且该方法与第一名的方法相比，有着更高的应用价值。

在 7.2 节中，讲述了本文非同质集成框架的集成策略，其在评测数据集上的实验结果与 NLPCC 2016 shared task 4 参与队伍相比可排在第五名。此外，实验结果证明了集成学习确实可以带来效果上的提升。在 7.2 节的最后，本文

分析了该框架在某些 target 上没有使得效果提升的原因，这也为其优化提供了方向。

# 第8章　总结与展望

本文探索了集成学习在微博立场分析中的应用，提出了两个用于立场分析的集成学习框架。为了设计具有差异性的基分类器，本文研究了 BOW 模型下小数据集上的监督学习立场分析方法，包括文本的语义表示，语义特征的分类方法等。

本文在第四章介绍了本文采用的文本语义特征；在第五章介绍了 LSA，LDA，SL 特征在 SVM 上的实验结果；在第六章，本文提出了一种基于相似度计算的分组投票模型（简称 GVM），并使用 LSA 和 LDA 分别进行了实验，实验证明，LSA-GVM 是一种较为简单高效的分类模型。

在第七章，为了解决 LSA-GVM 中的一些参数选取问题，本文使用 LSA-GVM 作为本文同质集成框架的基分类器，并提出了 Fast Ensemble LSA-GVM 方法，实验证明 Fast Ensemble LSA-GVM 方法是本文表现最好的方法，其在评测数据集上的实验结果与 NLPCC 2016 shared task 4 参与团队相比可排在第二名。本文在第七章还用 LSA-SVM，LDA-SVM，LSA-GVM 作为本文非同质集成框架的基分类器，给出了集成算法，其实验结果比 Fast Ensemble LSA-GVM 方法较差，但实验证实了集成学习确实可以在基分类器具有足够差异性和准确性的条件下提升分类效果。本文在第七章也给出了本文非同质集成框架的不足之处，这也是其的改进方向。

由于本文的非同质集成框架有着较大的提升空间，因此未来的工作可以对它进行优化。可以考虑在框架中使用 Fast Ensemble LSA-GVM 方法取代 LSA-GVM 方法；使用更多的基分类器；优化投票策略等等。还可以考虑使用一些针对短文本的主题模型，如 Yan X 等提出的 BTM(Biterm Topic Model)[26]。此外，因为本文仅探索了基于 BOW 的语义特征在立场分析中的应用，所以未来的工作可以考虑使用 n-gram 特征，embedding 等方法，这些非 BOW 的特征同样可以使用本文的 GVM 方法进行分类。GVM 也可以就如下方向进行改进：连接时根据各类别数据数量决定连接长度，避免非均衡数据集问题；改进概率（相似度）计算方法。

# 致谢

在做毕业设计的这几个月里，我遇到了很多困难，也有许多收获，有实验不顺利时的挫败和痛苦，也有最终成功的喜悦。在这里我要感谢所有在这一路上帮助过我的人，没有他们，我在过去几个月的工作会更加困难。

首先，我要感谢我的导师李舟军老师，赵刚老师在毕业设计期间对我耐心细致的指导。感谢李舟军老师在毕设选题时对我的细心指导，在毕业设计过程中对我的精心点拨，以及在论文修改时对我的严格要求。感谢赵刚老师在毕业设计过程中在工程细节，论文格式，毕设流程和要求等方面的悉心指导。

感谢北京理工大学计算机学院的牛振东老师，很有幸在毕业设计选题期间认识了牛老师。牛老师对教育的热情，对学术的一丝不苟深深感染了我，对我的学术态度和作风以及我对教育的认识和理解等方面有着重要的影响。这也是使得我秉承严谨求实，精益求精的态度完成高质量毕业设计的重要原因之一。

感谢哈尔滨工业大学的徐睿峰老师在评测数据集等方面对我的帮助。

感谢清华大学的刘知远老师关于 THULAC 分词工具的一些帮助。

感谢闫昭师兄在实验设计，论文修改等方面对我的帮助。

感谢同学张家培，杨旋，邵凯阳在文档编辑等方面对我的帮助。

最后，感谢在毕业设计期间家人和朋友对我无私的关怀，在未来的日子里，我将不负家人和朋友的期望，努力在自己选择的道路上走得更远。

# 参考文献

[1] Xu R, Zhou Y, Wu D, et al. Overview of NLPCC Shared Task 4: Stance Detection in Chinese Microblogs[M]// Natural Language Understanding and Intelligent Applications. Springer International Publishing, 2016.
[2] Thomas M, Pang B, Lee L. Get out the vote: determining support or opposition from congressional floor-debate transcripts[C]// Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2006:327-335.
[3] Walker M A, Anand P, Abbott R, et al. Stance classification using dialogic properties of persuasion[C]// Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013:592-596.
[4] Murakami A, Raymond R. Support or oppose?: classifying positions in online debates from reply activities and opinion expressions[C]// International Conference on Computational Linguistics: Posters. Association for Computational Linguistics, 2010:869-875.
[5] Sridhar D, Getoor L, Walker M. Collective stance classification of posts in online debate forums[J]. ACL 2014, 2014, 109.
[6] Liu L, Feng S, Wang D, et al. An Empirical Study on Chinese Microblog Stance Detection Using Supervised and Semi-supervised Machine Learning Methods[M]// Natural Language Understanding and Intelligent Applications. Springer International Publishing, 2016.
[7] Sun Q, Wang Z, Zhu Q, et al. Exploring Various Linguistic Features for Stance Detection[J]. 2016.
[8] Xu J, Zheng S, Shi J, et al. Ensemble of Feature Sets and Classification Methods for Stance Detection[C]// International Conference on Computer Processing of Oriental Languages. Springer International Publishing, 2016:679-688.
[9] Vijayaraghavan P, Sysoev I, Vosoughi S, et al. DeepStance at SemEval-2016 Task 6: Detecting Stance in Tweets Using Character and Word-Level CNNs[J]. 2016:413-419.
[10] Yu N, Pan D, Zhang M, et al. Stance Detection in Chinese MicroBlogs with Neural Networks[M]// Natural Language Understanding and Intelligent Applications. Springer International Publishing, 2016.
[11] 李勇, 刘战东, 张海军. 不平衡数据的集成分类算法综述[J]. 计算机应用研究, 2014, 31(5):1287-1291.
[12] Krogh A, Vedelsby J. Neural Network Ensembles, Cross Validation, and Active Learning[J]. Advances in Neural Information Processing Systems, 1994, 7(10):231--238.

[13] Touretzky E D S, Sollich P, Krogh A. Learning with ensembles: How over-fitting can be useful[J]. Advances in Neural Information Processing Systems, 1997, 8:190-196.
[14] Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, Zhiyuan Liu. THULAC: An Efficient Lexical Analyzer for Chinese[EB/OL]. http://thulac.thunlp.org/, 2016.
[15] Deerwester S. Indexing by latent semantic analysis[J]. Journal of the Association for Information Science and Technology, 1990, 41(6):391-407.
[16] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[M]. JMLR.org, 2003.
[17] Heinrich G. Parameter Estimation for Text Analysis[J]. Technical Report, 2008.
[18] Jun L I, Sun M. Experimental Study on Sentiment Classification of Chinese Review using Machine Learning Techniques[C]// International Conference on Natural Language Processing and Knowledge Engineering. IEEE, 2007:393-400.
[19] Cortes C, Vapnik V. Support-Vector Networks[M]. Kluwer Academic Publishers, 1995.
[20] Chang C C, Lin C J. LIBSVM: A library for support vector machines[M]. ACM, 2011.
[21] 王海林, 张雅君. 基于 LDA 的长短文本分类比较[J]. 数字技术与应用, 2016(10):230-230.
[22] 张志飞, 苗夺谦, 高灿. 基于 LDA 主题模型的短文本分类方法[J]. 计算机应用, 2013, 33(6):1587-1590.
[23] 方东昊. 基于 LDA 的微博短文本分类技术的研究与实现[D]. 东北大学, 2011.
[24] Lacoste-Julien S, Sha F, Jordan M I. DiscLDA: Discriminative Learning for Dimensionality Reduction and Classification.[J]. Proceedings of NIPS Neural Information Processing Systems (2008, 2008:897-904.
[25] Zhu J. MedLDA: Max-Margin Supervised Topic Models[J]. Journal of Machine Learning Research, 2009, 13(4):2237-2278.
[26] Yan X, Guo J, Lan Y, et al. A biterm topic model for short texts[J]. 2013:1445-1456.