

申请上海交通大学硕士学位论文

基于数据流形的主动学习和对抗深度学习算法研究

论文作者 郭文博

学 号 1140329044

导 师 杨煜普 教授

专 业 控制科学与工程

答辩日期 2017 年 02 月 16 日



Submitted in total fulfillment of the requirements for the degree of Master  
in Control Science and Engineering

# ACTIVE LEARNING AND ADVERSARIAL DEEP LEARNING ALGORITHMS BASED ON DATA MANIFOLD STRUCTURE

WENBO GUO

Advisor

Prof. YUPU YANG

SCHOOL OF ELECTRONIC INFORMATION AND ELECTRICAL ENGINEERING

SHANGHAI JIAO TONG UNIVERSITY

SHANGHAI, P.R.CHINA

Feb. 16th, 2017



## 上海交通大学 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名： 郭文博

日 期： 2017 年 2 月 13 日



## 上海交通大学 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

保 密 ☐，在 \_\_\_\_\_ 年解密后适用本授权书。  
不保密 ☒。

(请在以上方框内打 ✓)

学位论文作者签名： 郭文博

指导教师签名： 杨

日 期： 2017 年 2 月 14 日

日 期： 2017 年 2 月 14 日





# 基于数据流形的主动学习和对抗深度学习算法研究

## 摘要

主动学习作为一个框架，它需要与有监督学习方法结合才能被应用于一个实际问题，最常用且性能最好的有监督学习方法就是支持向量机和深度神经网络。因此本文的研究内容是支持向量机以及深度神经网络与主动学习的结合。主动学习和支持向量机的结合可以降低支持向量机训练时所需的标签样本数量，降低算法的成本。主动学习和深度神经网络的结合可以提升深度神经网络的算法安全性以及对对抗样本的抵抗性。

但是这两种结合存在着以下问题：对于主动学习支持向量机来说，主要的问题是算法对初始状态的敏感度高，易陷入局部最优解，以及对数据集的信息挖掘不够，造成信息浪费；对于主动学习深度神经网络，主要的问题是现有的算法无法保证深度学习算法对对抗样本的抵抗性，攻击者依然可以找到有效的对抗样本攻击目标模型。

针对以上问题，本文从数据的流形结构出发，主要做了以下研究：

- (1) 针对传统的主动学习支持向量机对无标签样本数据利用不充分的问题，设计了基于谱聚类的主动学习支持向量机算法，该算法通过利用无标签样本中的信息，提高主动学习支持向量机的性能。在文本分类标准数据集上的实验结果表明，所提出的算法取得了比传统主动学习算法更好的效果。
- (2) 针对传统的主动学习支持向量机对初值和噪声敏感的问题，利用数据集的低秩特性，设计了基于低秩子空间聚类的主动学习支持向量机方法。该算法利用低秩子空间聚类挖掘数据的低秩子空间结构。根据聚类的结果，选择两个簇中间稀疏区域的样本作为初始的标签样本，利用这些初始标签样本继续训练主动学习模型。在标准数据集上的实验说明了所提出的基于低秩子空间聚类的主动学习算法在分类性能和对初始状态的鲁棒性上优于传统的主动学习算法。
- (3) 针对传统的主动学习支持向量机对有标签样本数据利用不充分的问题，设计了基于低秩转换的主动学习算法。该算法引入低秩转换来挖掘数据集中有标签样本的信息，经过低秩转换后的数据可以到达类内距离最小化和类间距离最大化。在每次迭代中，数据先经过低秩转换被映射到特征空间，之后在特征空间更新分类模型。随着迭代的进行，越来越多的标签样本可以使低秩转换得到更精确的子空间信息，从而使分类模型有更好的效果。在多个标准数据集上的实验表明，本章提出的算法在分类性能和收敛速度上优于传统的主动学习算法以及被动学习支持

向量机方法。

- (4) 针对传统的主动学习深度神经网络,即对抗深度学习依然可以生成针对模型的对抗样本的问题,设计了一个全新的对抗深度学习框架。对抗主动学习是深度神经网络在主动学习框架中的实现,它的目的是提高深度学习对对抗样本的抵抗性。基于现有的对抗深度学习方法,结合数据的流形特性,设计了一种随机特征丢弃(random feature nullification)算法。本算法在模型中引入随机变量,从而阻止了针对模型的对抗样本的生成,提高了深度神经网络对对抗样本的抵抗性。所提出的算法比现有的对抗深度学习算法在 MNIST 和 CIFAR-10 数据集上有更好的抵抗性。本算法还被应用到恶意软件分类中,并取得了很好的分类效果和鲁棒性。

**关键词：**数据流形结构    主动学习支持向量机    对抗深度学习    随机特征丢弃

# ACTIVE LEARNING AND ADVERSARIAL DEEP LEARNING ALGORITHMS BASED ON DATA MANIFOLD STRUCTURE

## ABSTRACT

Active learning is a kind of framework. It has to be combined with some certain supervision learning method. In this thesis, we mainly focus on support vector machines (SVM) and deep neural networks (DNN). Integrating SVM with active learning can reduce the need of labeled data and decrease the training cost. Feeding deep neural networks into the framework of active learning aims to improve the resistance of deep neural network to adversarial samples.

However, these integrations encounter certain problems. To be more specific, as to active learning SVM, the main problem are that this algorithm is sensitive to original situation and still waste information of data structure. As to active learning DNN, state-of-the-art method can not provide enough security guarantee. Attackers can still craft model specific adversarial samples to attack the target model.

To solving the aforementioned problems, from the prospective of data manifold structure, we did the following research in this thesis:

- (1) In order to make full use of the information containing in unlabeled data, an improved active learning SVM is proposed. Before the active learning process, spectral clustering algorithm is applied to divide the dataset into two categories, and instances located at the boundary of two categories are labelled to train the initial classifier. In order to reduce the calculation cost, an incremental method is added to the proposed algorithm. The algorithm is applied to several text classification problems, with the results of being more effective and more accurate than the traditional active learning algorithm.
- (2) An novel active learning SVM with low-rank representation (LLR) subspace clustering is proposed to solve the sensitive initial states problem. At the initialization step of proposed active learning algorithm, a representation of data is obtained by applying LLR to the whole dataset, which get rid of the error and noise in dataset. Then subspace clustering is applied to the dataset based on an affinity grape built from the representation

matrix. Data points lie in the sparse region of two clusters are selected to be the initial support vector. After the initialization, active learning SVM is conducted to classify dataset into two classes. Experiment results of several standard binary classification datasets indicate that the proposed method has a higher accuracy than other state-of-the-art methods and eliminate the influence of different initial state.

- (3) An active learning SVM based on low-rank transformation (LRT) for binary classification is proposed to take best advantage of the data distribution characteristic and information contained in labeled data. In each iteration, before updating the classifier model, we use the labeled data samples chosen by select engine to derive a transformation matrix. Then, we project the whole dataset to a union of subspaces by learnt matrix, where data samples pertaining to different classes are lying in nearly orthogonal subspaces of the original data space. After that, SVM are adopted to renew the classifier. The transformation conducts in the same kernel space as SVM. As the iteration goes on, more data are labeled, which makes it easier to explore the intrinsic structure of dataset and get a better classification performance. Experiments on several standard datasets indicate the proposed algorithm achieve a better performance than other classification algorithms.
- (4) DNN been proven to be quite effective in many applications such as image recognition and using software to process security or traffic camera footage, for example to measure traffic flows or spot suspicious activities. Despite the superior performance of DNN in these applications, it has recently been shown that a DNN is susceptible to a particular type of attack that exploits a fundamental flaw in its design. Specifically, an attacker can craft a particular synthetic example, referred to as an adversarial sample, causing the DNN to produce an output behavior chosen by attackers, such as misclassification. Addressing this flaw is critical if a DNN is to be used in critical applications such as those in cybersecurity. Previous work provided various defense mechanisms by feeding DNN into active learning framework. However, after a thorough analysis of the fundamental flaw in the DNN, we discover that the effectiveness of such methods is limited. As such, we propose a new adversary resistant technique that obstructs attackers from constructing impactful adversarial samples by randomly nullifying features within samples. Using MNIST and CIFAR-10 datasets, we evaluate our proposed technique and empirically show our technique significantly boosts the robustness of DNN against adversarial samples while maintaining high accuracy in classification. The results of applying proposed method to malware classification also show better resistance and classification

performance than state-of-the-art method.

**KEY WORDS:** Manifold Structure, Active Learning Support Vector Machines, Adversarial Deep Learning, Random Feature Nullification



# 目 录

<b>第一章 绪论</b>	<b>1</b>
1.1 研究背景及意义	1
1.2 主动学习的框架和研究现状	2
1.2.1 主动学习框架	2
1.2.2 主动学习研究现状	5
1.3 深度神经网络基本结构	7
1.3.1 深度神经网络	7
1.3.2 深度神经网络基本框架	9
1.4 数据的流形假设和流形结构	13
1.5 本文主要研究内容	14
<b>第二章 基于谱聚类的主动学习支持向量机</b>	<b>17</b>
2.1 引言	17
2.2 主动学习支持向量机	18
2.2.1 支持向量机	18
2.2.2 选择引擎	19
2.3 谱聚类算法	20
2.4 基于谱聚类的主动学习支持向量机	21
2.5 算法有效性验证	21
2.6 本章小结	24
<b>第三章 基于低秩子空间聚类的主动学习支持向量机</b>	<b>25</b>
3.1 引言	25
3.2 低秩子空间聚类算法	26
3.2.1 低秩表示	26
3.2.2 低秩子空间聚类	27
3.3 基于低秩子空间聚类的主动学习支持向量机	28
3.3.1 所提出的主动学习算法	28
3.3.2 算法复杂度分析	30

3.4	算法有效性验证	30
3.4.1	ala 数据集的结果	31
3.4.2	diabetes 数据集的结果	31
3.4.3	german.numer 数据集的结果	32
3.4.4	ionosphere 数据集的结果	32
3.4.5	初值对不同算法的影响	33
3.5	本章小结	34
<b>第四章</b>	<b>基于低秩转换的主动学习支持向量机</b>	<b>35</b>
4.1	引言	35
4.2	低秩转换算法	36
4.3	基于低秩转换的主动学习支持向量机	39
4.3.1	算法详述	39
4.3.2	算法的时间复杂度分析	41
4.4	算法有效性验证	41
4.4.1	DNA 数据集的结果	41
4.4.2	w5a 数据集的结果	44
4.4.3	letter 数据集的结果	45
4.4.4	其他标准数据集的结果	46
4.5	本章小结	48
<b>第五章</b>	<b>主动学习深度神经网络-对抗深度学习</b>	<b>51</b>
5.1	引言	51
5.2	对抗深度学习	53
5.2.1	对抗样本	53
5.2.2	主动学习深度神经网络：主动选择对抗样本	54
5.3	基于 RFN 的对抗深度学习	55
5.3.1	模型描述	56
5.3.2	对抗样本抵抗性分析	57
5.3.3	基于流形结构的算法分类有效性分析	59
5.3.4	不同算法的对比	59
5.4	算法有效性验证	60
5.4.1	实验设置和初始化	60
5.4.2	实验结果	62



5.5	基于 RFN 的对抗深度学习在恶意软件分类中的应用 . . . . .	64
5.5.1	恶意软件分类 . . . . .	64
5.5.2	基于 RFN 的对抗深度学习在恶意软件分类的效果 . . . . .	66
5.6	本章小结 . . . . .	68
<b>第六章</b>	<b>总结与展望</b>	<b>69</b>
6.1	研究工作总结 . . . . .	69
6.2	未来研究工作 . . . . .	69
6.2.1	模型超参数选择 . . . . .	69
6.2.2	流形学习算法时间复杂度 . . . . .	70
	<b>参考文献</b>	<b>71</b>
	<b>致 谢</b>	<b>81</b>
	<b>攻读学位期间发表的学术论文</b>	<b>83</b>
	<b>攻读学位期间申请的专利</b>	<b>85</b>



## 第一章 绪论

### 1.1 研究背景及意义

机器学习是一个统计和计算机完美结合产生的交叉学科，和大多数传统学科相比，机器学习还是一个新生儿。但是随着计算机软件和硬件技术的快速发展，机器学习在这样的时代背景下成为了最热的学科之一。机器学习已经被成功应用到很多领域。随着 GPU 技术的发展，以及大量的数据可以被轻易得到，神经网络再一次走入人们的视野，并一跃成为了效果最好的机器学习方法之一。

虽然机器学习功能强大，但是没有免费的午餐<sup>[1]</sup> 理论告诉我们，机器学习并不是万能的，至少在样本过少和特征不足的情况下，机器学习就达不到很好的效果。虽然在现如今的大数据时代，这两种情况很少发生，但是现有的机器学习方法确实还是有这样那样的不足，值得研究者进一步的研究。对机器学习算法的提升可以从很多方面，比如提高算法的效率或者分类算法的分类性能。

机器学习<sup>[2]</sup> 就是计算机从数据样本中获取有用的数学模型的过程。这些新的模型可以用于分类、回归等，对于最常见的分类问题，主要的方法分为有监督学习<sup>[3]</sup>，半监督学习<sup>[4-6]</sup> 和主动学习<sup>[7-9]</sup>。

在以上三种方法中，主动学习所需要的标签样本最少，算法的消耗也相应较小，在大量的无标签样本可以获得，但是标记样本的成本依然很高的今天，本文把研究的重点放在主动学习上面。对主动学习思想，框架和研究现状的介绍将在后续章节中展开。主动学习的最大特点就是在标签样本不足的情况下，它可以主动选择样本，用更少的有标签样本量，达到尽可能好的效果。虽然主动学习的设计目的就是减少标签成本，机器学习算法中最大的成本消耗，

主动学习是一个框架，它需要和特定的有监督学习方法相结合，才能应用到实际的应用场景中去，解决实际问题。本文关注的模型是支持向量机<sup>[10]</sup> 和深度神经网络。在深度学习出现之前，支持向量机是最好的单一模型分类器，但是深度学习的问世，取代了支持向量机的地位。2014 年研究者也第一次把主动学习的思想应用到了深度学习中，对抗深度学习应运而生，虽然不是典型的主动学习框架，但是对抗深度学习是主动学习思想在深度学习的实现。

随着深度学习的不断发展和应用，它强大的性能得到了各个领域的认可。井喷式的发展同时也引起了人们的质疑。深度学习是否能保证安全，特别是在一些关键问题还没有完整的数学理论的前提下，这个问题就更加突出。由于对深度学习的安全存在质疑，

深度学习的应用也受到了限制,迄今为止,深度学习在很多对安全要求很好的领域还很少有应用,比如计算机安全和金融行业。

大量的实践表明,主动学习在和不同的有监督算法相结合时还是存在各种各样的问题,主动学习支持向量机的问题是对数据集信息挖掘不够深入,造成信息浪费,以及主动学习支持向量机通常随机初始化,造成对初始状态的敏感等问题。传统的对抗深度学习学习方法,主要是对抗性训练并不能阻止攻击者针对特定模型生成对抗样本,也就是传统的对抗深度学习不能保证深度学习的安全。

本文从一个全新的视角,审视主动学习,发现传统的主动学习很少挖掘数据的分布特性。大多数的主动学习方法都是把训练数据拿来,最多经过预处理就直接用于训练模型,对数据集本身,没有挖掘的过程,这就造成了信息的浪费。实际上,数据集本身的一些信息,如分布特性是可以被获取并且为后续工作所用的,而且也有各种各样的算法是为了挖掘这些信息而设计的。本文就着眼于通过挖掘这些信息,来提升主动学习的性能,希望通过对数据的挖掘,得到有用信息,利用这些信息解决主动学习存在的问题。

## 1.2 主动学习的框架和研究现状

### 1.2.1 主动学习框架

#### 1.2.1.1 主动学习基本思想

本节从有监督学习开始讨论主动学习和其它两种学习算法在思想上的不同。

从控制论的角度来看,有监督学习是一个开环的过程。学习算法的输入是有标签的训练数据集,输出是分类模型。在有监督学习中,机器的目标是从人提供的有标签数据集中归纳出一个高精度的分类模型。实践表明,机器最终学到的模型通常只取决于小部分的关键数据。训练数据集中的其他数据都是冗余的,对分类模型没有任何贡献,它们所包含的有用信息已经被那小部分的关键数据完全覆盖。这些冗余数据不仅增加了模型训练过程的计算量开销,还造成了数据采集、标注过程中大量人力物力的浪费。训练集中之所以包含大量的冗余数据,是因为传统模式下,数据的采集过程是完全随机进行的,具有盲目性。

实际应用中,采集无标签数据通常比较容易,但是获取样本的类别标签要困难得多。人力物力成本主要集中于对数据进行标注的过程。例如,在基于内容的图像检索<sup>[11, 12]</sup>应用中,检索系统需要从图像数据库中挑选出用户所感兴趣的那一小部分图像。数据库中的所有图像都是现成的,而样本标签非常缺乏。

很明显,对无标签样本进行标注所需的人力成本成为制约有监督学习的首要瓶颈。为了降低机器学习算法对有标签样本的依赖,减少数据标注过程的人力开销,研究者们

提出了半监督学习。

半监督学习方法的基本思想是同时从有标签数据和无标签数据中进行学习,充分利用大量无标签数据中所包含的有用信息,以提高有标签样本不充分时机器学习的效果。通常,半监督学习方法所使用的无标签样本在数量上远远超过有标签样本。从控制论的视角来看,半监督学习仍然是一个开环的过程,它与有监督学习的主要区别在于系统的输入不仅包含有标签数据,还包括大量的无标签数据。半监督学习着眼于有标签样本已经给定的情形。实际应用中,有标签样本通常还是随机选择的,既可能遗漏关键样本,也可能包含冗余性。

为了更有针对性地选择有标签样本,最大限度地降低对样本进行标注所需要消耗的人力成本,研究者们提出一系列方法,让机器主动地挑选最有价值的数据样本用于学习,即主动学习。主动学习系统包含两个关键组成部分:学习引擎和选择引擎。学习引擎通常就是传统的有监督学习算法,负责从已获得标注的有标签数据集中学习得到需要的模型。选择引擎负责从无标签样本池中挑选学习用的数据。

在每次迭代过程中,选择引擎首先根据当前已学得的模型,从无标签样本池中挑选出它认为最有价值的样本,交给人工进行标注。获得标注后,该样本被加入有标签样本集。学习引擎再使用增广的有标签样本集更新学得的模型。之后,选择引擎再根据新的模型挑选新的最有价值的样本。这样循环迭代,直到满足特定的停止条件。可以看出,选择引擎根据当前模型选择样本并请求相应的类别标签。从控制论的角度,主动学习是闭环过程,当前模型的结果反馈会输入端,影响输入的样本,同时新输入又会更新模型。

### 1.2.1.2 主动学习算法框架

主动学习作为机器学习的重要分支,发展至今也产生了各种各样的框架,主要的框架有以下两种:基于样本池的主动学习和基于数据流的主动学习。在各种主动学习框架中,应用最广泛的是基于样本池的主动学习,这里的“样本池”指的是一个大的无标签样本集。本节主要介绍基于样本池的主动学习。

基于样本池的主动学习的结构图在图 1-1 中给出。如图所示,主动学习的各个部分组成了一个闭环系统,值得注意的是,虽然主动学习是主动选择样本,但是在学习开始时,有标签样本集不能为空,需要包含少量样本用于学习得到初始模型,作为主动学习迭代的起点。主动学习的关键在于如何选择学习样本,如果样本是随机选择的,主动学习就退化成了被动学习。负责选择样本的是选择引擎,选择引擎根据某种指标来定量地衡量无标签样本的“价值”,而且衡量无标签样本的“价值”还必须以当前已学得的模型为依据。这就是主动学习的关键。

图中除了选择引擎,另一个重要组成部分是学习算法,这个学习算法是典型的有监

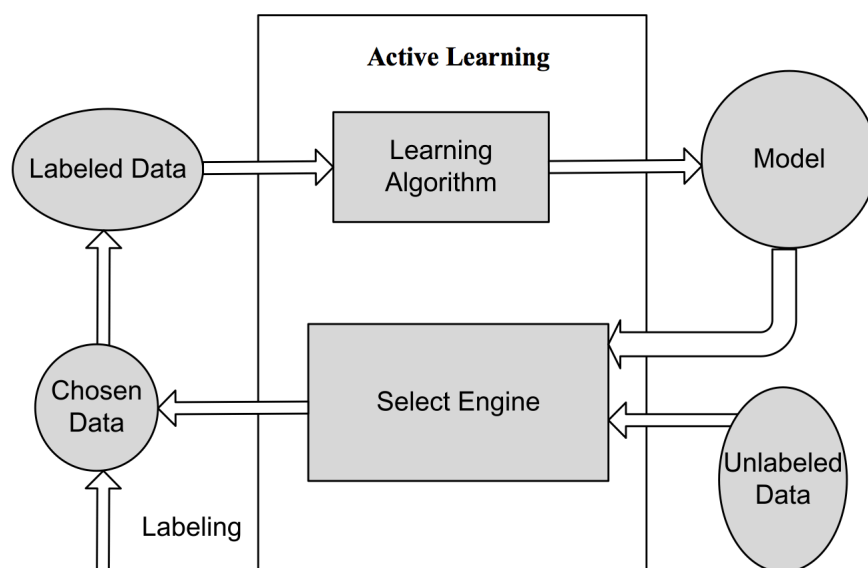


图 1-1 基于样本池的主动学习

督学习算法，选择不同的算法，就会得到不同的模型。本文中，我们主要研究基于两种算法：支持向量机和深度神经网络的主动学习。在之后章节中会给出基于不同的有监督学习方法的主动学习算法的具体步骤，这里就不再展开说明。

从图 1-1 可以看出，主动学习需要反复地进行模型重新训练。如果选择引擎每次只选择一个样本，所需要的迭代步骤会很多，学习引擎需要频繁地进行重新学习。为了减少这种计算量开销，学习引擎可以采用增量式学习算法<sup>[13]</sup>，每次增加新的有标签样本后，学习引擎只需要根据新样本对已学得模型进行更新，而不必使用整个有标签样本集重新学习。

另一种方法是让选择引擎每次选取多个样本，以减少所需要的迭代次数，从而减少模型重复训练的次数。这种模式被称为批量式主动学习<sup>[14]</sup>。需要注意的是，选择引擎所选取的多个样本之间会存在信息交叠，选择样本时需要综合考虑样本的价值和样本之间的差异。

传统的主动学习是贪婪算法，因为选择引擎总是选择对当前模型来说信息量最大的样本。贪婪算法所能取得的效果往往严重依赖于初始点。如果初始模型较差，主动学习有可能会陷入较差的局部最优解，即使经过很多次迭代，仍然无法得到高精度的模型，甚至出现主动学习不如被动学习的情况。因此，如何确定主动学习的初始有标签样本集是一个值得研究的问题。

## 1.2.2 主动学习研究现状

由于主动学习和被动学习的主要差别是主动学习的选择引擎,因此主动学习领域的主要研究内容也是围绕选择引擎展开的,选择引擎的性能直接影响了主动学习算法的效果。因此,主动学习研究现状这一节的内容主要是围绕现阶段几种比较常用的样本选择策略展开的。

### 1.2.2.1 不确定性采样

主动学习中最常用也最简单的样本选取策略是不确定性采样<sup>[15, 16]</sup>。选择引擎总是从无标签样本池中挑选类不确定性最大的样本。例如,对判别模型(如支持向量机<sup>[9]</sup>等)而言,不确定性最大的点就是离当前模型的分类面最近的点。

样本的不确定性越大,要把它分到正确的类别需要的额外信息越多,人在给出该样本的类别标签时,得到的有用信息也就越多。常用的衡量样本不确定性的标准有以下三种:

- 最小置信度:  $x^* = \underset{x \in U}{\operatorname{argmin}} P_f(\hat{y}|x)$ , 其中  $\hat{y} = \underset{y \in Y}{\operatorname{argmin}} P_f(y|x)$ ,  $f$  为当前模型;
- 边界:  $x^* = \underset{x \in U}{\operatorname{argmin}} (P_f(\hat{y}_1|x) - P_f(\hat{y}_2|x))$ , 其中  $\hat{y}_1, \hat{y}_2$  是当前模型下, 输入样本  $x$  置信度最高的两个类别;
- 熵:  $x^* = \underset{x \in U}{\operatorname{argmin}} \sum_{y \in Y} P_f(y|x) \log P_f(y|x)$ , 本标准是根据样本在当前模型下的熵来选择熵最大的样本, 值得注意的是样本熵的衡量标准是:  $-\sum_{y \in Y} P_f(y|x) \log P_f(y|x)$ 。

这些公式中  $U$  表示的是无标签样本池。

不确定性采样在大量的实际应用中都能取得非常好的效果,极大地降低学习所需要的有标签样本数量。本文中使用的选择引擎就是不确定性采样。

### 1.2.2.2 基于委员会的选择

基于不确定性采样的主动学习使用的是单模型学习引擎。实际应用中,使用多个模型的集成方法<sup>[17]</sup>通常能取得比单个模型更高的精度。基于委员会的样本选择方法<sup>[18]</sup>正是基于多个模型。

委员会指的是学习引擎所使用的多个模型,它们都是从现有的有标签数据集中学习得到的。对于整个委员会而言,信息量最大的样本就是在委员会内部引起争议最大的样本。为了定量地衡量无标签样本  $x$  引起争议的程度,多种指标被提出来,其中应用最广的是投票熵,熵越大说明该样本的争议越大,信息含量越大。

委员会产生分歧的前提是各个成员间的存在明显差异。但是所有模型都是从相同的有标签数据集中学习得到的，为了使它们之间存在明显的差异，常用的方法有：

- 委员会中不同成员选择不同的训练算法；
- 委员会中不同成员选择不同的超参数；
- 对数据集做不同的特征提取处理。

### 1.2.2.3 期望模型变化和期望误差下降

本节介绍两种基于决策理论的样本选择方法：期望模型变化和期望误差下降。其中，期望模型变化的基本思想为：如果一个样本能够对模型造成的变化越大，那么它所包含的信息量也就越大。因此，主动学习的选择引擎应该选择可能对模型带来最大变化的样本。

基于这一思想，文献<sup>[19]</sup>中介绍了一种适用于概率模型的基于期望梯度大小的主动学习算法。从理论上讲，该方法适用于所有采用梯度下降法进行训练的机器学习算法。设优化问题的损失函数为  $L$ ，学习引擎的模型参数是  $\theta$ ，损失函数对优化参数的梯度为  $\nabla_{\theta} L$ 。

假设在新一轮主动学习迭代过程中，新的无标签样本被选中并获得标注。由于前一轮迭代结束时梯度近似为零，新的梯度值可以有效衡量新引入的样本对模型造成的变化的程度。事实上，选择引擎在挑选无标签样本时并不知道样本对应的类别标签，因此，梯度大小应该针对所有可能的标签值取期望。由此得到的样本选择策略为：

$$\hat{x} = \underset{y}{\operatorname{argmax}} \sum P_{\theta}(y|x) \|\nabla_{\theta} L\|_2 \quad (1-1)$$

公式 1-1 中的后验概率采用自举法<sup>[20]</sup>进行估算。最大化期望模型变化有着严格的理论基础，但是由于计算量较大，期望模型变化并没有得到广泛的应用。

期望误差下降也是一种基于决策理论的样本选择方法，它直接以模型的泛化误差为目标。其基本思想是：对于所有的无标签样本，估算出将选到的样本加入有标签样本池之后从中学得的新模型的泛化误差，然后选择使新模型泛化误差最小的样本用于标注。通常，无标签样本集都很大，能够较好地反映数据的分布特征，也就被用作验证集来估算模型的泛化误差。常用的估计模型泛化误差的方法有 0-1 损失函数和对数损失函数，基于这两种损失函数，期望误差下降的样本选择标准有以下两个：

- $x^* = \underset{x \in U}{\operatorname{argmin}} \sum_{y \in Y} P_f(y|x) \sum_{x \in U} (1 - P_{f+(x,y)}(\hat{y}|x))$
- $x^* = \underset{x \in U}{\operatorname{argmin}} \sum_{y \in Y} P_f(y|x) (-\sum_{x \in U} \sum_{y_j \in Y} (P_{f+(x,y_j)}(y_j|x) \log P_{f+(x,y_j)}(y_j|x)))$



上述公式中  $f^{+(x,y)}$  表示把数据  $x$  加上标签后训练出来的模型。期望误差下降的方法和基于互信息的样本选择策略<sup>[21]</sup> 相似，都是估计加入该样本后模型的泛化误差。

与其他样本选取策略相比，期望误差下降法直接以减小模型的泛化误差为目标，通常能用更少的有标签样本获得分类精度满足要求的模型。但是期望误差下降法所需要的计算量也是常用主动学习算法中最大的，因此实际应用并不广泛。

期望误差下降法不仅仅是单一的方法，还是一种有效的框架，可以很容易地推广到以优化模型的其他指标（如 F1 测度、ROC 曲线下面积等）作为选取无标签样本的标准。

#### 1.2.2.4 有标签样本充足时主动学习的作用

虽然主动学习是为有标签样本不充分的应用场景设计的，但是当有标签样本十分充足时，主动学习也有用武之地。在这种情况下使用主动学习的主要优势有：

- 加快模型的训练速度，虽然有标签样本充足，但是有标签样本包含的信息量不尽相同，主动学习可以选择信息量最大的样本来快速的训练模型，而被动学习只能随机选择样本；
- 提高样本不均匀时的模型精度，当样本不均匀时，至少有一个类别的有标签数据是不够的，这时，主动学习也可以发挥它的优势，即在有标签样本不充足的时候还能达到较好的训练效果；
- 降低噪声干扰，主动学习可以自主选择信息量较大的样本来训练模型，这从一定程度上滤掉了一定的噪声数据。这个效果可以使主动学习在训练模型时不受到噪声的干扰。

### 1.3 深度神经网络基本结构

#### 1.3.1 深度神经网络

随着互联网技术的发展，人们获得数据的途径越来越多，可以被获取的数据呈现指数级的增长，这对机器学习领域是机遇也是挑战。首先，海量的数据确实给机器学习算法提供了更丰富的资源来训练出更精准的模型，更好的完成机器学习任务。但是传统的机器学习方法，特别是特征提取方法，算法的时间复杂度较高，很难有效的处理这么多的数据，同时这些算法设计简单，很容易在训练数据过多的时候出现过拟合的现象，纵然有海量数据，但是可以被真正利用的数据却只占很小的一部分。

但是深度神经网络<sup>[22]</sup> 的出现，从很大程度上解决了这个问题。通过增加网络层数，深度神经网络可以处理大量的数据而不出现过拟合现象，同时由于其灵活复杂的结构，深度神经网络不需要进行特征提取过程。也就是说原始的数据可以被直接用于训练神经

网络模型，这就大大减少了训练算法的复杂度。同时还打破了机器学习算法对特征提取方法的依赖。神经网络已经被证明可以模拟任意非线性函数，所以从理论上来说，只要有充足的数据，神经网络可以用于完成各种机器学习任务。

在实际应用中，深度神经网络在诸如图像识别<sup>[23]</sup>，特征提取<sup>[24]</sup>，文本分类<sup>[25]</sup>等领域取得了颠覆性的结果。同时神经网络也已经被用于一些以前的机器学习方法没有涉及的领域，比如是计算机安全。计算机安全是一个对专家知识，模型的领域，所以是用不到机器学习方法的。但是随着数据的指数增加，单纯依靠人力来解决问题越来越不现实，因此机器学习方法，尤其是深度学习方法也开始被用于这个领域。已经有研究把深度学习应用到了诸如恶意软件分类<sup>[26]</sup>，二进制代码分析<sup>[27]</sup>等领域，并且取得了惊人的效果，这也促使了深度神经网络在这个领域更深层次的应用。

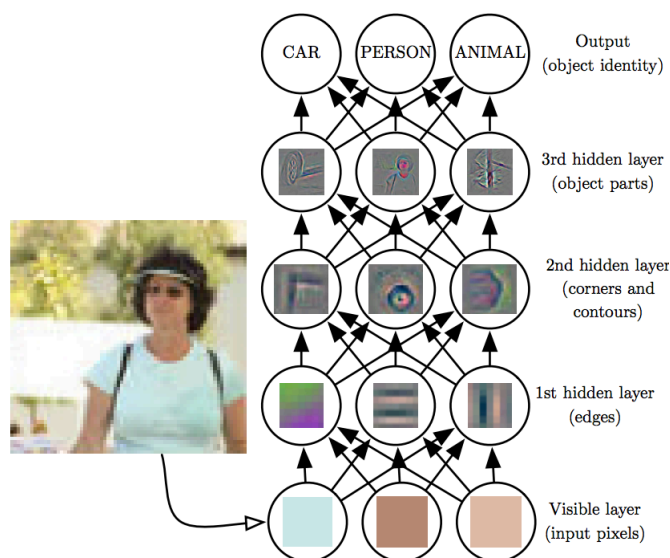


图 1-2 深度学习示意图

图 1-2<sup>[28]</sup> 是深度学习的示意图，本图以图像作为例子解释深度学习的原理，图中深度神经网络的网络结构是多层感知机。从图中可以看出，该结构把复杂映射分解为一系列嵌套的简单映射。一个完整的神经网络包括输入层，隐含层和输出层。所谓隐含层是网络中用于提取特征的不可见的层，从图中可以看出，随着层数的增加，深度神经网络提取出的特征也越来越复杂和抽象，依据最后提取出的高度抽象的特征，在输出层进行判断并输出决策结果。和普通的机器学习方法最大的区别是神经网络可以提取更加抽象的特征，而且这个过程不需要借助其他算法，具体的网络结构在后续章节介绍。

虽然神经网络是现在最流行和有效的机器学习方法，但是它并不是近期提出的算法。最早的深度模型被用来识别裁剪得很合适且非常小的图像中的单个对象<sup>[29]</sup>。最早

网络只能识别两种对象，远远无法满足实际应用的需求。神经网络迅速崛起的标志是卷积网络在图像领域的成功，它在 ImageNet<sup>[30]</sup> 数据集上的效果远远超过了传统的方法，比如支持向量机。同时，深度学习也对自然语言处理产生了巨大影响，特别是语音识别<sup>[31]</sup> 和机器翻译<sup>[32]</sup> 等，google 的基于神经网络的机器翻译模型可以说是颠覆了整个机器翻译领域，取得的效果也远远超过了传统的方法，比如隐式马尔可夫链<sup>[33]</sup>，条件随机场<sup>[34]</sup>。

随着神经网络在各个领域取得的成功，它可以完成的任务也日益复杂。比如文献<sup>[35]</sup> 中提出的神经图灵机，可以说是人工智能领域的又一大突破。图灵机是可以模拟任意程序的，所以如果可以用神经网络实现一个图灵机，那么相信不久的将来，由机器自动编写程序，实现复杂的任务将成为现实。

总之，依赖于计算机基础技术的发展，比如 GPU 技术的发展，数据库技术的革新，深度学习也会越来越多的被应用于各种领域中。同时，随着人们对深度学习数学基础的探索，深度学习自身也会不断发展和完善，相信深度学习刮起的旋风必将持续很长一段时间。

本节中我们介绍了深度学习基本思想和发展历程，深度学习最常见的结构有多层感知机<sup>[24]</sup>，卷积神经网络<sup>[36]</sup> 和递归神经网络<sup>[37]</sup>，其中卷积神经网络和多层感知机属于前馈网络，而递归神经网络涉及了神经元自身的连接，不是典型的前馈网络结构。递归神经网络主要用于处理时序数据，比如语音数据。由于本文不涉及这类数据，因此我们并没有用到递归神经网络这种结构，在本文中也不做介绍。下一章节，我们主要介绍两种前馈网络结构：多层感知机和卷积神经网络。

### 1.3.2 深度神经网络基本框架

#### 1.3.2.1 多层感知机

多层感知机又叫前馈神经网络，是典型的深度学习模型。它的目标是近似某个函数  $f^*$ 。例如，对于分类器， $y = f^*(\mathbf{x})$  将输入  $\mathbf{x}$  映射到一个类别  $y$ 。多层感知机定义了一个映射  $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$ ，并且学习参数  $\boldsymbol{\theta}$  的值，使它能够得到最佳的函数近似。因为信息流的传递是从输入到输出单向传递的，没有同层，所以它被称为前馈网络。

网络中每一层由神经元组成，这也是这种结构被称为神经网络的主要原因。每一个神经元都是一个单独的向量到向量的函数。每个神经元都有一定的刺激区域和抑制区域。神经元刺激和抑制的功能是靠激活函数实现的，激活函数一般是非线性函数，在特定的范围内呈现线性特性，在其它的区域内呈现非线性特性，多数是饱和特性，常见的激活函数有 Sigmoid 函数，Tanh 函数，ReLU 函数等<sup>[38]</sup>。

神经元和神经元之间通过网络中的边连接，每一条边都有特定的权重  $w$ ，和偏置  $b$ ，

这些参数组成了神经网络的参数集，神经网络的训练过程就是找出最优的参数取值的过程。神经网络除了有输入层和隐含层之外，还有一层输出层，输出层的作用是把提取出的特征映射到特定的输出空间中，比如对于分类问题来说，输出层的作用就是输出样本的类别标签，常见的输出层是 softmax 分类器<sup>[38]</sup>。

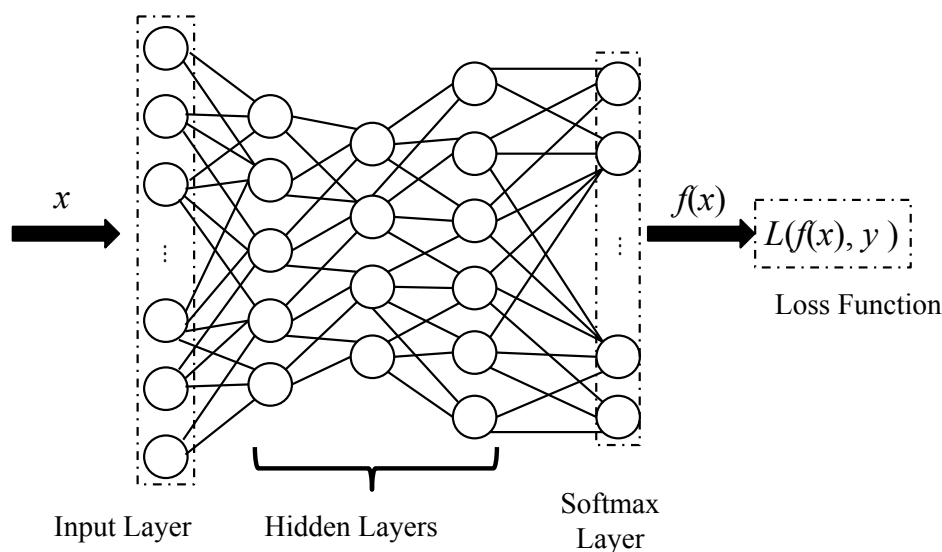


图 1-3 多层感知机示意图

图1-3是一个简单的多层感知机结构，如图所示，该网络有一层输入层，三层隐含层和一层输出层，输入样本  $x$  经过神经网络处理之后输出的  $f(x)$ ，和样本的标签  $y$  比较构成损失函数（目标函数） $L$ ，损失函数会在后续内容介绍。值得注意的是，作为前馈网络，多层感知机的每一层内没有相互连接，但是层与层之间是全联通的。所谓全联通就是上一次的一个节点和下一层的每一个节点都有连接。

在多层感知机这种链式结构中，主要的结构考虑是选择网络的深度和每一层的宽度。有些情况下，即使只有一个隐含层的网络也足够适应训练集。更深层的网络通常能够对每一层使用更少的单元数和更少的参数，并且有更好的泛化性能，但是通常也更为难以优化。对于一个具体的任务，理想的网络结构必须通过一些特定的超参数选择手段来确定。

在介绍完了结构之后，我们介绍一下神经网络的训练过程。在训练神经网络之前，我们首先要明确自己的目标，也就是要有一个目标函数，即损失函数。对于大多数的分类问题，神经网络都是作为判别模型使用，也就是说神经网络的输出是一个条件概率分布  $p(y|x; \theta)$ 。对于每一个输入样本  $x$ ，输出的是在当前输入和模型参数条件下的预测结

---

**Algorithm 1-1: Compute derivative of each parameter by BP algorithm<sup>[39]</sup>.**

---

**Input:** data sample  $(x, y)$ , number of layers in neural network  $n$ ;

**Output:**  $\nabla_{W^{(l)}} L(W, b; x, y)$ ,  $\nabla_{b^{(l)}} L(W, b; x, y)$ , where  $l=1,2,\dots,n$ .

---

**Step 1:** Compute the activation value of each layer:  $z^{(2)}, z^{(3)}, \dots, z^{(n)}$ ;

**Step 2:** As to output layer, residual error can be computed by:

$$\delta^{(n)} = \frac{\partial L}{\partial z^{(n)}} = \frac{\partial L}{\partial g(z^{(n)})} g'(z^{(n)});$$

**Step 3:** As to other layers, the residual error of layer  $l$  can be computed by:

$$\delta^{(l)} = ((W^{(l)})^T \delta^{(l+1)}) \cdot g'(z^{(l)});$$

**Step 4:** Compute the derivative:  $\nabla_{W^{(l)}} L(W, b; x, y) = \delta^{(l+1)} g(z^{(l)})^T$

$$\nabla_{b^{(l)}} L(W, b; x, y) = \delta^{(l+1)}.$$


---

表 1-1 利用 BP 算法计算参数梯度

果  $f(x)$ ，它和样本标签  $y$  组成损失函数，损失函数由损失项和正则项组成，常见的损失项有 0-1 损失函数，对数损失函数等<sup>[38]</sup>，常见的正则项有  $l_2$  和 lasso 等<sup>[38]</sup>。

基本上所有的神经网络训练方法都是基于梯度的方法，所以我们首先要得到损失函数对每一个参数的梯度。求的这些梯度的方法是 BP 算法<sup>[39]</sup>。BP 算法，又称为反向传递算法，基础是微积分中的链式法则。算法过程是通过把损失函数对最后一层输出层的梯度作为残差，沿着神经网络的反方向，把残差传递到每条边。同时根据链式法则求出损失函数对每一个参数的梯度值。基于 BP 算法的这个基本思想，我们可以求的损失函数对每一个参数的梯度，下面我们直接在表1-1中给出 BP 算法在多层感知机上的实现。

一个神经网络首先给出了前向传播，它将参数映射到与单个训练样例（输入，目标） $(x, y)$  相关联的有监督损失函数  $L(f(x), y)$ ，其中  $f(x)$  是当  $x$  提供输入时神经网络的输出。表中公式中的  $g(\cdot)$  代表激活函数， $z^{(l)} = W^{(l-1)}g(z^{(l-1)}) + b^{(l-1)}$  代表输入和参数的线性组合， $\cdot$  表示向量乘积运算符。

得到各个参数的梯度之后，我们可以通过权重更新方法来更新参数，其中比较常用的方法是批量梯度下降法，具体更新权重过程在表1-2中给出：

以上就是利用批量梯度下降法更新权重的过程，其实随着研究的深入，还有一些收敛速度更快的优化方法被提出，比如 Adam<sup>[40]</sup>，RMSprop<sup>[41]</sup> 等，不过它们都是基于梯度的方法。

在具备足够的数据和适当的计算资源的前提下，多层感知机的表现非常好。基于多层感知机派生出来的算法，如各种自编码器在机器学习的很多关键领域都得到了应用，例如特征提取，在这种情况下，多层感知机的应用也从有监督学习扩展到了无监督学习。

### 1.3.2.2 卷积神经网络

卷积神经网络是多层感知机的变体，是一种专门用来处理图像数据的神经网络。卷积是一种特殊的线性运算。卷积神经网络就是用卷积代替一般矩阵运算的网络结构。

卷积操作<sup>[42]</sup>的数学公式如下：

$$S(i, j) = (X * K)(i, j) = \sum_m \sum_n X(m, n) K(i - m, j - n). \quad (1-2)$$

其中  $K$  是卷积操作中的核函数，又被称为卷积核， $*$  代表卷积操作， $X$  是数据，根据输入数据维度的不同，卷积核也可以取多个维度。

卷积神经网络的另一个重要操作是池化（pooling）<sup>[38]</sup>，池化函数使用某一位置的相邻输出的总体统计特征来代替网络在该位置的输出。例如，maxpooling<sup>[38]</sup> 函数给出相邻矩形区域内的最大值。其他常用的 pooling 函数包括相邻矩形区域内的平均值、 $L^2$  范数以及依靠据中心像素距离的加权平均函数。

卷积神经网络和普通的多层感知机最大的区别就是池化和卷积操作，因此卷积神经网络的隐含层分为卷积层，池化层（下采样层）和全联通层。其中卷积层指的是输出数据是通过输入数据和特定的卷积核进行卷积操作得到，下采样层就是池化函数，做常用的池化函数是 maxpooling，全联通层就是典型的单层感知机，它们组合形成了卷积神经网络的结构。下图是一个最常用的卷积神经网络结构：

图中的卷积神经网络是 AlexNet<sup>[43]</sup>，AlexNet 共有 9 层，其中隐含层有七层，七层隐含层中前两层是卷积池化层，也就是进行完卷积操作要在本层的最后进行池化操作，第三，四层是单纯的卷积层，没有池化，第五层是卷积池化层，最后两层是全联通层。卷

---

Algorithm 1-2: Update each parameter<sup>[38]</sup>.

---

Input:  $W^{(l)}, b^{(l)}$ , number of iteration  $m$ ;

Output:  $W^{(l)}, b^{(l)}$ .

---

Step 1: let  $\Delta W^{(l)} = 0, \Delta b^{(l)} = 0$ ;

Step 2:  $t = 1$ ;

Step 3: Compute  $\nabla_{W^{(l)}} L(W, b; x, y)$  and  $\nabla_{b^{(l)}} L(W, b; x, y)$  according to algorithm 1-1;

Step 4:  $\Delta W^{(l)} = \Delta W^{(l)} + \nabla_{W^{(l)}} L(W, b; x, y)$ ,  $\Delta b^{(l)} = \Delta b^{(l)} + \nabla_{b^{(l)}} L(W, b; x, y)$ ;

Step 5:  $W^{(l)} = W^{(l)} - \alpha[\frac{1}{m}\Delta W^{(l)} + \lambda W^{(l)}]$ ;

Step 6:  $b^{(l)} = b^{(l)} - \alpha[\frac{1}{m}\Delta b^{(l)}]$ ;

Step 7:  $i = i + 1$  and return to step 3, until  $i = m$ .

---

表 1-2 利用批量梯度下降法更新参数

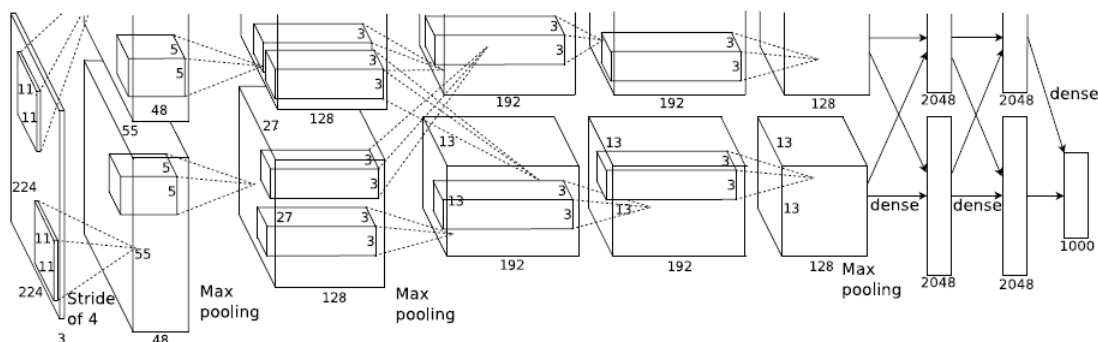


图 1-4 卷积神经网络示意图

积神经网络的输出层根据任务不同而不同，对于分类问题，和多层感知机相同，它一般也是 **softmax** 回归。根据不同的任务有不同的目标函数，网络中的参数主要是卷积核中的参数和偏置项，这些参数的梯度也是通过 **BP** 算法求得的。值得强调的是，卷积和池化有特定的残差传递方法，和标准的全联通层的残差传递稍有区别，但是整体的思路还是和标准 **BP** 相同，利用链式法则求到目标函数对不同参数的梯度。得到参数的梯度之后，可以利用批量梯度下降法等优化方法更新参数。

卷积神经网络是基于 BP 算法训练的有效的深度网络之一。相较于多层感知机，卷积神经网络有更少的超参数，因此计算效率更高。使用它们可以更快的进行交叉验证选择更好的超参数，训练机构更复杂的网络也相对容易。卷积神经网络可以处理具有清楚的网格结构拓扑的数据。这种方法在二维图像拓扑上是最成功的，因此卷积神经网络在图像领域取得了巨大的成功，其在 ImageNet 上面的效果没有其他方法可以企及。卷积神经网络已经成为图像相关的所有领域，诸如人脸识别<sup>[44]</sup>、图像分割<sup>[45]</sup>等的标准方法。

## 1.4 数据的流形假设和流形结构

由于本文的、研究内容都是通过挖掘数据集的分布信息来提高主动学习的性能，其中挖掘数据集的特性的基础是数据的多流形结构，所以本节我们介绍数据的流形假设和多流形结果的含义。

在大量的数据可以被轻易获取的情况下, 如何挖掘数据中的信息成为了一大难题。流形学习<sup>[46]</sup>是挖掘数据结构特性一种重要方法, 它的基础是流形假设。所谓流形假设就是假设数据集采样于一个欧氏空间, 数据集中每一个样本点的附近区域都属于一个相同的欧式子空间。

最早的流形学习算法研究的情况是所有的数据都属于同一个子空间,但是随着研究

的深入, 研究者们越来越多的转向对现实世界中更加复杂的结构进行研究。而现实中的大多数数据都具有多个混合子空间的结构。数据的多流形结构就是指具有多个流形(多个混合子空间)的数据结构。对于此类数据, 准确的判断每个点所属的子空间是问题的关键。因此准确挖掘数据的多流形结构是问题的关键。现有的算法中, 挖掘子空间结构的方法包括稀疏表示<sup>[47]</sup>和低秩表示<sup>[48]</sup>:

- 稀疏表示方法的假设是不同的子空间相互独立。在这种情况下, 每一个数据点只能由属于相同子空间的数据表示。也就是说, 如果用数据集中的其它数据表示一个数据, 该系数矩阵中的大部分位置应该为零, 只有在和它属于同一子空间的数据点对应的位置才有值, 因此这个矩阵是稀疏的, 这就是数据的稀疏性。基于数据稀疏性得到的数据表示就是稀疏表示。
- 低秩子空间表示<sup>[48]</sup>的目标是得到一个秩尽可能低的数据表示矩阵。子空间相互独立的情况下, 数据的表示矩阵应该具有低秩特性。因为数据集中的每一个数据都只能写成和它属于同一子空间的数据的线性组合。低秩的数据表示通过限制表示矩阵的核范数来实现。矩阵核范数是矩阵秩的最优凸近似。

## 1.5 本文主要研究内容

本文的主要研究对象为主动学习支持向量机, 即以支持向量机作为学习引擎的主动学习方法, 以及主动学习深度神经网络, 即以深度学习作为学习引擎的主动学习方法。

传统的主动学习方法在挑选无标签样本时根据当前已学得模型来估计样本的信息量, 而忽略了数据集整体的分布特征, 存在一些明显的不足:

- (1) 主动学习支持向量机忽略冗余样本, 即无标签样本中的信息。
- (2) 主动学习支持向量机对初始状态和噪声数据敏感: 主动学习本质上属于贪婪算法。如果初始化不当, 主动学习算法很容易陷入差的局部最优解。通常, 主动学习在初始化阶段随机选择少量的有标签样本来训练初始模型。因此, 传统主动学习方法受初始有标签样本集影响很大, 容易陷入局部最优。
- (3) 传统主动学习支持向量机对信息利用不够充分: 主动学习对数据集的利用仅限于从有标签数据集中归纳出分类模型。相较于对数据利用率较高的半监督学习, 传统的主动学习方法不仅没有利用无标签数据中的信息, 甚至对有标签数据集的利用也不够充分。
- (4) 传统的对抗深度学习方法对对抗样本的抵抗性有限, 它无法从根本上阻挡对抗针对模型的样本生成。

为了克服这些不足, 本文主要研究了以下内容:

本文第二章研究如何利用数据集中的无标签样本的信息。我们设计了基于谱聚类的



主动学习算法来提高无标签样本的利用率，在主动学习开始之前，谱聚类算法被用于则整个数据集把数据集聚为两簇，之后我们选择在两簇之间的稀疏区域，根据不确定性采样的样本选择策略，选择出信息量最大的无标签样本，作为初始的标签样本，即初始的支持向量。这些样本被用于训练初始的支持向量机模型。我们把基于谱聚类的主动学习方法应用于文本分类领域，取得了比传统主动学习支持向量机更好的效果。

本文的第三章，我们专注于解决主动学习支持向量机对初值和噪声敏感的问题。在本章中，我们依据数据的流形假设，通过挖掘数据的低秩子空间结构，得到一个描述数据之间相关性的邻接矩阵，之后根据邻接矩阵的信息，通过子空间聚类的方法，把原始的数据集聚到两个不同的簇中。根据聚类的结果，选择两个簇中间区域的样本，作为初始的支持向量，训练初始的支持向量机模型。之后按照主动学习的算法训练后续的模式。在标准数据集上的实验说明了所提出的基于低秩子空间聚类的主动学习算法比传统的主动学习支持向量机有更好的鲁棒性。

本文第四章专注于挖掘数据集中有标签样本的信息，并把挖掘到的信息和主动学习的过程相结合，来提高主动学习的性能。算法的大致过程是在每次迭代开始的时候，在更新模型之前，先利用所有的有标签样本，得到一个低秩转换。这个低秩转换的目的是找到数据隐含的多流形结构，使属于同一类别的数据矩阵的秩越小，属于不同类别的数据矩阵的秩越大，这样就保证了同一类别样本之间的相似度更大，不同样本之间的相似度更小。在学习到这个转换矩阵之后，利用这个矩阵把所有数据映射到相应的特征空间中，在特征空间中完成本次迭代的剩余操作，即更新支持向量机模型。按照此方法进行迭代，直至学习结束。我们把这种基于低秩转换的主动学习方法应用到近 20 个标准数据集，并和传统的主动学习算法以及被动学习支持向量机方法进行对比，实验结果显示本章提出的算法在分类性能和收敛速度上优于其它算法。

本文第五章针对传统对抗深度学习抵抗性不足的问题，提出了一种 **random feature nullification** 算法可以在模型中引入随机变量，从而阻挡残差的向回传递。这种方法可以阻止针对模型的对抗样本的产生，从而增强算法的鲁棒性。在 **MNIST** 数据集上的实验表明，所提出的算法比对抗性训练有更好的抵抗性和分类效果。同时所提出的方法可以和对抗性学习相结合，达到更好的效果。最后，我们把本章提出的方法应用到恶意软件分类中，并取得了很好的分类效果和鲁棒性。

本文的第六章对全文进行了总结，在此基础上提出了未来的研究内容。



## 第二章 基于谱聚类的主动学习支持向量机

### 2.1 引言

在大数据的背景下，主动学习<sup>[49]</sup>以其有效性高的特点已经成为最流行的机器学习方法之一，尤其是处理一些标签数据难以获取的问题。主动学习算法的一大优点是通过标记信息量最大的数据点来减小学习的成本，也就是用最少的标签数据学习到性能最好的分类器。主动学习的另一个优点是通过选择信息量最大的样本，学习过程的迭代次数会减少，因此节省了训练时间。为了更优的选择样本，主动学习引入了一个选择引擎，有了这个选择引擎，主动学习会有更好的普适性。

在多数的基于样本池的主动学习任务中，数据集都是由一个很小的有标签数据集  $L$  和一个更大的无标签数据集  $U$  组成。通过这一小部分有标签数据，我们可以学到一个初始化模型。之后，为了提高模型的性能，在每一次迭代过程中，主动学习都会选择一些最有信息量的样本点并标记它们。这些被标记的样本点被用于更新模型。这个不断标记和更新的过程在达到一定的停止标准之后停止。通过这种学习方法，那些不重要的，冗余的数据点将不会被用于模型更新，这样就减小了标签成本和计算成本。

主动学习是一个迭代过程<sup>[50]</sup>。每一次新的样本被标记，分类模型都会相应的更新，直到达到一定的停止标准。这个过程需要大量的运算。而且很有可能在结束了前几次迭代之后，模型已经基本稳定，不再会有大幅度的变动。同时，值得注意的是新的标签数据是基于现有的模型选择的。所以，很有可能随着迭代的进行，主动学习模型选出的样本不再有足够的信息量，这就增加了标签成本。例如，如果现有的分类超平面和最优的分类平面相距很远，那么基于现有模型选出的样本就没有什么价值，不停的标记这样的样本会增加标签成本，这种无谓的成本增加是由于学习过程中忽略了数据的分布信息。

本文使用了增量式学习方法。当一个样本点被标记，分类模型只基于新标记的样本信息来进行更新，而不是基于之前所有的标签样本更新模型。主动学习的另一问题是传统的主动学习算法对初始状态的灵敏度高，不同的初始点对学习的效果影响很大，造成了学习模型的鲁棒性不强，这个问题极大的限制了主动学习的应用。为了解决这个问题，同时为了减少学习过程的标签成本以及更大程度上利用数据的分布特性，本文提出了基于谱聚类的主动学习算法。在主动学习开始之前，本文先利用谱聚类算法把数据集聚为两簇，之后那些位于两簇边界处的样本被选为初始的标签样本来训练初始的主动学习模型。在学习过程中，随着模型的更新，那些离分类超平面最近的样本点被选为新的标签数据，这个过程直到达到停止标准时停止。这个算法的有效性已经在一些文本分类

的标准数据集上得到验证。

本章的后续内容安排如下，在子章节 2.2 介绍本章和后续两章算法的基础：主动学习支持向量机，接着在 2.3 中，引入谱聚类算法，在之后的章节 2.4 中提出本章的算法，并且给出算法的具体过程。之后 2.5，我们通过实验证明算法的有效性。最后对本章进行总结。

## 2.2 主动学习支持向量机

主动学习指的是那些自动选择要标记的数据点的一类学习算法，因此所有被动的有监督学习算法都可以被用于训练分类模型。本文选择支持向量机作为分类模型<sup>[51, 52]</sup>，支持向量机只有一个最优的分类超平面，因此多用于二分类问题。支持向量机学到一个线性分类器，这个分类器不仅可以用于对数据分类，同时可以通过数据点离分类超平面的距离来判断样本点所包含的信息量，这使得支持向量机非常适用于主动学习。在主动学习支持向量机的每次迭代中，离超平面最近的样本点被选为新的标签样本并用于在下次迭代中更新分类模型。

### 2.2.1 支持向量机

支持向量机的目的是学习一个最优的分类超平面。假设一个用于二分类的支持向量机模型有  $N$  个输入样本点  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ ,  $x_i \in \mathbb{R}^n$  且  $y_i \in \pm 1$ ，那么支持向量机的分类方程如下：

$$f(x) = \text{sign}(w^T \cdot x + b) \quad (2-1)$$

其中  $w$  和  $b$  决定了分类超平面，这个最优的超平面是通过最大化以下的分类边界得到的：

$$\max_{w, b} \{ \min \{ \|x - x_i\| : x \in \mathbb{R}^n, w \cdot x + b = 0 \} \} \quad (2-2)$$

在数据不是线性可分的情况下，我们引入软间隔最大化和一个参数  $C > 0$ 。于是通过求解下面的优化方程，我们可以得到一个线性支持向量机分类器：

$$\min_{w, b, \xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\zeta} \xi_i \quad (2-3)$$

$$s.t. y_i(w \cdot \Phi(x) + b) \geq 1 - \xi_i, \forall i \quad (2-4)$$

$$\xi_i \geq 0, \forall i \quad (2-5)$$

其中  $\Phi(\cdot)$  是一个从输入空间到特征空间的映射， $C$  是惩罚系数，控制着对误分样本的惩罚程度， $\xi_i$  是松弛系数。

这个问题是一个凸多项式问题，可以通过求解它的 Wolfe 对偶方程<sup>[53]</sup> 来得到最优解，其对偶方程的形式如下：

$$\max_a \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \Phi(x_i) \Phi(x_j) \quad (2-6)$$

$$s.t. 0 \leq \alpha_i \leq C \quad (2-7)$$

$$\forall i, \sum_{i=1}^N \alpha_i y_i = 0 \quad (2-8)$$

其中  $i$  是拉普拉斯乘子。

在实际应用中，大多数的问题面对的是非线性问题，例如非线性分类问题指的是需要非线性模型完成分类任务的分类问题。为了解决这类问题，本文引入核函数<sup>[54]</sup>，其中最常用的核函数是 RBF 核：

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \quad (2-9)$$

其中  $x_i$  和  $x_j$  是两个不同的样本点， $\gamma$  是核函数的参数。

引入核函数之后，2-6中的  $\Phi(\cdot)$  被2-9所取代，于是非线性的分类模型可以通过求解新的优化问题得到，求得的分类模型如下：

$$f(x) = \text{sign}(\sum_{i=1}^N \alpha_i^* y_i K(x_i, x) + b) \quad (2-10)$$

其中  $\alpha^*$  是最优解，因此模型中的参数可由以下方程得出：

$$w = \sum_{i=1}^N \alpha_i^* y_i x_i \quad (2-11)$$

$$b = y_i - \sum_{i=1}^N \alpha_i^* y_i K(x_i, x) \quad (2-12)$$

由于  $w$  和  $b$  是由  $\alpha_i^* > 0$  的数据点决定的，因此  $\alpha_i^* > 0$  的样本点  $(x_i, y_i)$  就是所谓的支持向量。

## 2.2.2 选择引擎

主动选择需要标记的样本是主动学习的最大特点和核心内容。迄今为止，关于如何选择样本，已经有很多研究，也产生了各种各样的方法，比如基于委员会的决策方法<sup>[55]</sup>，期望模型变化<sup>[56]</sup> 和期望误差下降<sup>[18]</sup>。然而最流行的方法是不确定性采样<sup>[57-59]</sup>。理论上可以通过引入版本空间<sup>[9]</sup> 的概念来解释不确定性采样的有效性。

对于不确定性采样，在每次迭代过程中，选择引擎总是会基于当前模型选择最不确定的样本，比如，对于概率模型中，后验概率接近 0.5 的数据点会被选出来：

$$x = \operatorname{argmin} |P_f(y|x) - 0.5| \quad (2-13)$$

对于判别模型，那些具有最大不确定性的数据点指的是离现有分类超平面最近的样本点。具体到支持向量机，选择引擎基于以下公式来选择样本：

$$x = \operatorname{argmin}_{x \in U} \left| \sum_{i=1}^N \alpha_i y_i K(x_i, x) + b \right| \quad (2-14)$$

其中  $U$  为无标签样本池，根据本节的介绍，这里我们在表 2-1 给出基于样本池的主动学习。

---

Algorithm 2-1: Pool-based active learning with SVM<sup>[60]</sup>.

---

Input: Unlabeled dataset  $U^{(0)}$ , set of initial labeled samples  $L^{(0)}$ ,  
number of data samples selected each iteration  $k$ ;

Output: Classification model  $f^{(t)}$ .

---

Step 1: Let  $t = 0$ , learn  $f^{(0)}$  from  $L^{(0)}$  by SVM;

Step 2:  $t = t + 1$ ;

Step 3: Select  $k$  most uncertain data samples from  $U^{(t)}$  according to  
equation 2-14 and label them;

Step 4: Update  $L$  and  $U$ ;

Step 5: Learn  $f^{(t)}$  from  $L^{(t)}$ ;

Step 6: Repeat step 2 - 5, until stopping criterion is met;

Step 7: Return  $f^{(t)}$ .

---

表 2-1 主动学习支持向量机算法

## 2.3 谱聚类算法

谱聚类算法<sup>[61]</sup>通过样本点间的相似度将数据分为两簇。该算法先把数据点转换为一个图，这个图可以由  $G = (V, E)$  表示，其中  $V$  表示数据样本， $E$  表示样本点间的相似度，因此解决这个问题就转化为通过最大化簇内部的  $E$  值，最小化簇之间的  $E$  值，把一个图  $G$  分为两个簇。本章中谱聚类算法如下表所示：

---

**Algorithm 2-2: Spectral clustering algorithm<sup>[61]</sup>.**


---

**Input:** Unlabeled dataset  $X = x_i$ ;

**Output:** Label vector  $Y$ , where  $y_i \in \pm 1$ .

---

**Step 1:** Build the graph  $G = (V, E)$  and calculate adjacent matrix  $W$ ,

where  $W_{ij} = K_{ij}(x_i, x_j)$ ;

**Step 2:** Calculate the degree of each point  $d_{ij} = \sum_j W_{ij}$ ;

**Step 3:** Calculate diagonal matrix  $D$ ,  $D_{ij} = 0$ , when  $i \neq j$ ,  $D_{ij} = d_{ij}$ , when  $i = j$ ;

**Step 4:** Calculate the Laplace matrix of  $G$ :  $L_{sym} = I - D^{-1/2}WD^{-1/2}$ ;

**Step 5:** Apply eigenvalue decomposition to  $L_{sym}$  and

the second less feature value of  $L_{sym}$  can be obtained, written as  $v_2$ ;

**Step 6:**  $y = \text{sign}(v_2)$ , and return  $y$ .

---

表 2-2 谱聚类算法

## 2.4 基于谱聚类的主动学习支持向量机

在分别介绍完了主动学习支持向量机和谱聚类算法之后，这一小节介绍本章的核心内容。假设有一个二分类问题，为了挖掘数据的分布信息，减少标签样本的成本，以及增加分类准确性，我们先对数据进行谱聚类。在主动学习进行之前，谱聚类算法被用于整个数据集，之后那些位于所聚成的两簇边界处的样本点被选为最初的支持向量。在主动学习的过程中，每次更新模型之后，距离分类超平面最近的样本点被选为新的支持向量，并对其进行标记，之后根据最新标记的样本更新分类模型，直到学习结束，具体的算法表 2-3：

## 2.5 算法有效性验证

本章中，为了检验所提出算法的有效性，我们把这种算法应用到文本分类问题，尤其是两个公开数据集：news20 和 w2a 中。这些数据集都来自于 LIBSVM<sup>[62]</sup>。通过比较所提出算法和传统主动学习支持向量机算法，本文提出的算法的有效性可以得到验证。在这里传统主动学习支持向量机算法指的是标准的基于样本池的主动学习支持向量机，即表 2-1 中介绍的算法。news20 数据集有 20000 个样本点，每一个样本点代表一个用英文写的新闻，一共有 20 个不同的类别。w2a 数据集有 3470 个样本的训练集和含有 46279 个样本的测试集。本章的所有实验程序都采用 C++ 编写。

对 news20 数据集，我们使用线性支持向量机，因为本数据集中的数据是线性可分的。而对于 w2a 数据集，我们使用带有高斯核的非线性支持向量机，在本章中，高斯核

---

**Algorithm 2-3: Active learning with spectral clustering**


---

**Input:** Unlabeled dataset  $U$ , the number of initial labeled data instance  $k$ ;

**Output:** An SVM classifier.

---

**Step 1:** Divide  $U$  into two categories with the use of spectral clustering algorithm,  
where  $e_i$  is the difference of weights between  $x_i$  and each category center;

**Step 2:** Choose  $k$  data points which have the least value of  $e$  for labeling,  
save them as dataset  $L$ ;

**Step 3:** Learn a classification model  $f^{(0)}$  from the labeled datasets  $L$ ;

**Step 4:** Set  $t = 1$ ;

**Step 5:** Actively select unlabeled instance  $x^*$  which is located nearest to  
the hyper plane, delete  $x^*$  from  $U$  and let:  $L = L \cup x^*$ ;

**Step 6:** Update classifier according to  $x^*$ ;

**Step 7:**  $t = t + 1$

**Step 8:** Repeat step 5 - step 8, until the stopping criterion is met;

**Step 9:** Return final classification model.

---

表 2-3 基于谱聚类的主动学习支持向量机算法

的超参数被选为  $\gamma = 0.01$ ，而支持向量机的超参数被设为  $C = 100$ 。我们用传统的主动学习支持向量机作为对比，这两种算法被分别应用与两个数据集。每组实验都有 100 次重复实验，每次选择 5 个样本点作为初始的支持向量（标签样本）。对于传统的主动学习支持向量机算法，每次随机选择初始的支持向量。

news20 数据集的实验结果分别在图 2-1、2-2 和 2-3 中展示。在每幅图中，蓝线代表传统的主动学习算法，红线代表本章提出的算法。每幅图都给出了从第 21 次迭代到第 51 次迭代的结果，从中可以看出两种方法在效果上的差异。

从图 2-1 中可以看出，在第 11 和 12 类新闻的分类任务中，所提出的算法的准确率一种比传统的主动学习算法高，特点是最后一次迭代之后，本章的算法比传统算法的误分率低 33%。在图 2-2 所显示的分类任务中，本章所提的算法在开始时有较低的误分率，但是随着迭代的进行，传统算法逐渐逼近所提出的算法，最终两者达到了相似的误分率。然而对于图 2-3 的分类任务，虽然一开始时传统算法表现了较好的效果，但是最终还是被本章提出算法超越。

对于 w2a 数据集，我们用传统的主动学习支持向量机进行了 100 次实验，从中选择了误分率相对较高，误分率相对较低和平均误分率三组结果作为和所提出算法的比较，这些结果都在图 2-4 中给出，每个都给出了从第 1 到 51 次迭代的误分率。从图 2-4



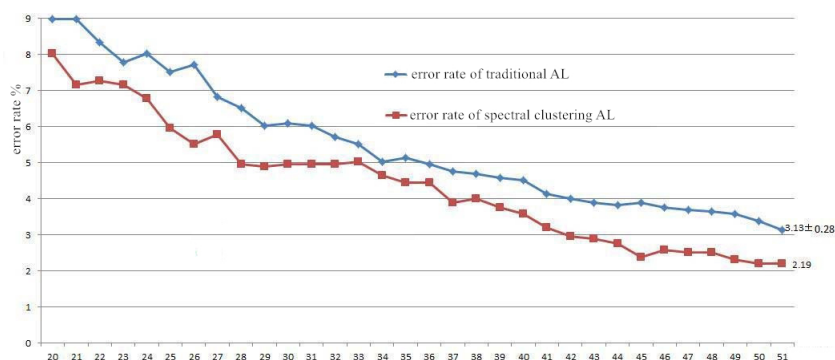


图 2-1 两种方法在 news20 中第 11 和 12 类新闻的误分率

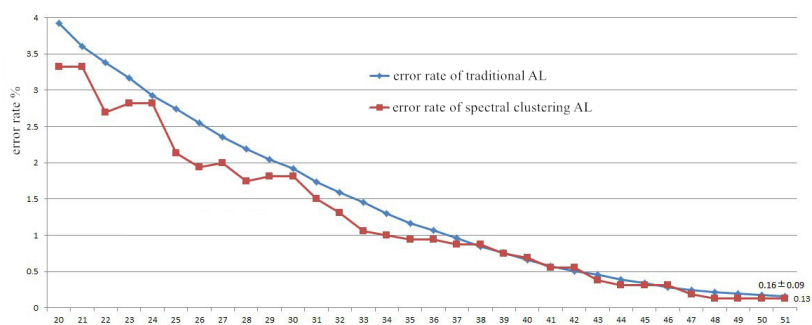


图 2-2 两种方法在 news20 中第 7 和 15 类新闻的误分率

中我们可以得到无论传统主动学习的哪一组结果都比本章所提的算法有更高的误分率。从图 2-4 我们还可以得到传统主动学习的结果波动很大,这说明了传统主动学习算法受初始状态的影响很大,有一定的随机性。这正是基于谱聚类的主动学习方法避免的地方。通过谱聚类算法对数据进行处理后,我们可以选出更有信息量的样本作为出事的标签样本,从而降低了算法对初始状态的灵敏度。从所有的实验结果中,我们还可以发现 news20 的误分率总是低于 w2a 的误分率,这是因为 news20 的数据线性可分,分类难度相对较低。

从以上的实验结果,我们可以看出本章所提出的算法在文本分类的标准数据集上较传统的主动学习算法体现出了更高的分类精度和更快的收敛速度。这是因为在设计算法时,本章提出的算法考虑到了数据集中无标签数据包含的信息,并且把这些信息加以利用。而传统的主动学习算法在算法开始时并没有对数据集的分布信息加以利用,随着算法的进行,一步一步的对这些信息进行挖掘,相较于本章所提出的直接利用数据集分布信息的方法,这种逐步挖掘信息的方法效率较低。同时本章所提出的算法从不同的角度利用这些信息,先从整体上把握数据集的分布趋势,再一步步选择对分类超平面最有用

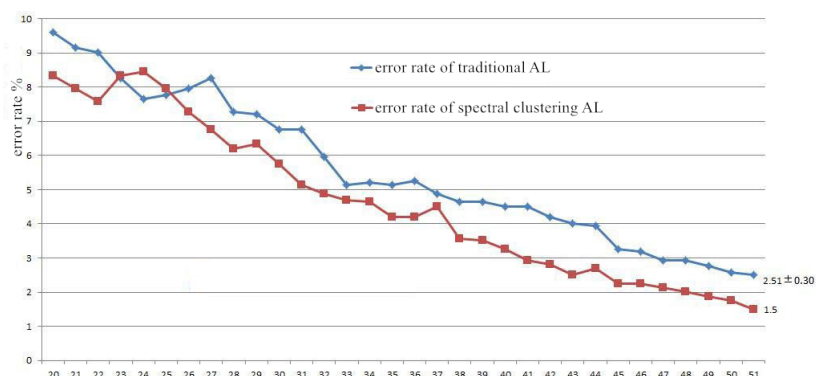


图 2-3 两种方法在 news20 中第 19 和 20 类新闻的误分率

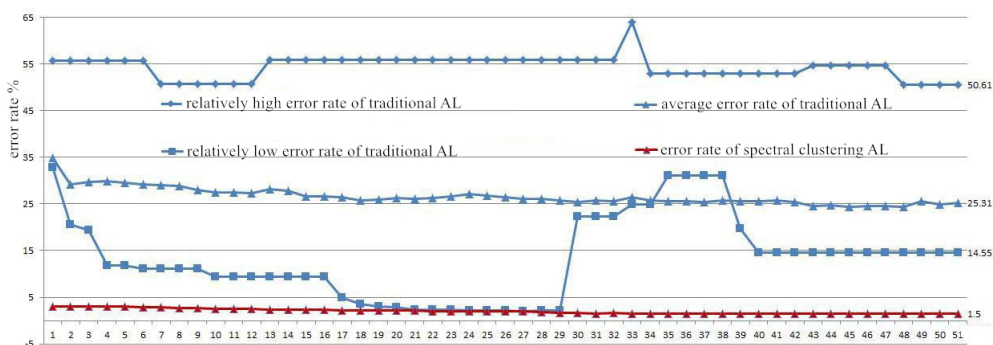


图 2-4 两种方法在 w2a 的误分率

的样本加以利用，这样可以更充分的挖掘挖掘整个数据集，因此可以达到更好的效果。

## 2.6 本章小结

实验结果显示本章所提出的算法超越了传统的主动学习支持向量机算法，因为本算法的结果准确率更高，并且更稳定。这说明了适当的利用数据集的分布信息可以提高主动学习算法的性能。

事实上，所提出的算法可以作为一个算法设计框架。谱聚类算法只是一种用于提取数据集特征的一种方法。当数据集变得更加复杂时，更有效的算法可以被用于提取数据集的分布信息，比如 Greedy Gradient Max-cut (GGMC)<sup>[63]</sup>, Sparse Subspace Clustering (SSC)<sup>[47]</sup>, Spectral multi-manifold clustering (SMMC)<sup>[64]</sup>。

## 第三章 基于低秩子空间聚类的主动学习支持向量机

### 3.1 引言

主动学习针对的应用场景是那些数据集缺少标签的分类或者回归任务。在机器学习的过程中,尤其是有监督学习的学习过程,一个重要的步骤就是获得数据样本的标签信息。通过应用主动学习的方法,可以降低机器学习过程所需要的标签数量从而减小标签成本,即使是对那些有充分标签数据的机器学习任务,引入主动学习也可以使模型得到更好的效果并且加快训练模型的收敛速度。然而大多数的主动学习方法都对初始状态很敏感,这就意味如果没有选择合适的初始状态,主动学习会陷入局部最优解,而训练不出全局最优的模型<sup>[9]</sup>。造成这一结果的主要原因就是传统的主动学习方法忽视了数据集的分布信息,但事实上,大多数数据集中都蕴含着大量有用信息可以被挖掘和利用。

对于大多数机器学习任务,比如人脸识别和运动分割,数据都属于不同的子空间。比如,主成分分析<sup>[65]</sup>和矩阵补全<sup>[66]</sup>都是基于数据属于一个子空间,即数据的单流形假设而设计出来的。然而,实际应用中,数据分布情况往往更加复杂,因此单一流形的假设并不适用,所以假设数据符合多流形分布更为合理。同时,对于一个给定的数据集,它不含有任何误差和噪声的情况是很难出现的。

低秩表示<sup>[48]</sup>是一种全新的数据表示方法,这种方法目的是揭露数据内部的子空间信息,特别是对有噪声的数据。这种方法已经被证实是一种有效的特征提取方法<sup>[67]</sup>,并且被应用到了很多领域中。同时学习到的低秩表示也可以和子空间聚类<sup>[68]</sup>结合来把数据聚到不同簇中。

在这一章中,我们针对分类问题,首先使用低秩子空间聚类方法来挖掘数据的多流形分布特征提取,并在进行过特征提取的数据基础上进行主动学习,完成分类任务。在初始化阶段,我们首先使用无监督的低秩子空间聚类方法把数据集分为两簇,之后根据聚类结果选择初始的支持向量。被选择的样本被用于学习初始的分类模型,之后随着迭代的进行,更多的数据被标记,同时模型不停的进行更新,直到满足主动学习停止标准。本章依然选择支持向量机作为分类器,把所提出的算法应用于二分类问题。在一些标准数据集上的实验结果表示所提出的算法在分类性能和鲁棒性上优于现有的算法。

本章的后续内容安排如下,在子章节 3.2 介绍低秩子空间聚类算法,接着在 3.3 中,我们介绍本章提出的算法,也就是基于低秩子空间聚类的主动学习支持向量机。之后 3.4,我们通过实验证明算法的有效性。最后对本章进行总结。

## 3.2 低秩子空间聚类算法

### 3.2.1 低秩表示

基于流形假设<sup>[69]</sup>，大多数的机器学习数据内部都有子空间结构，这种子空间结构对分类任务十分的重要。为了更大程度上的利用这个重要的信息，本章我们引入一个数据恢复方法来挖掘这种信息。这种方法就是低秩表示，低秩表示的设计目的是用尽可能低秩的数据表示方法来把原始数据表示为给定的字典矩阵的线性组合。这里我们先介绍低秩表示算法。

假设有一个数据集  $X = [x_1, x_2, x_3, \dots, x_n]$ ，其中每一个数据点  $x_i$  的维度是  $d$ ，每一个数据都可以表示成一个字典矩阵的线性组合：

$$X = AZ \quad (3-1)$$

其中  $A$  是字典矩阵， $Z$  是系数矩阵， $Z$  的每一列  $z_i$  是每一个数据点的表示。这样以来， $Z$  就可以表示数据的子空间信息。为了得到尽可能低秩的表示，算法的优化方程被设计为如下形式：

$$\min_Z \text{rank}(Z) \quad (3-2)$$

$$s.t. X = AZ \quad (3-3)$$

然而  $\text{rank}(Z)$  是非凸的，所以在解这个优化方程时，核范数可以被用来作为秩的近似，因此优化方程转换为：

$$\min_Z \|Z\|_* \quad (3-4)$$

$$s.t. X = AZ \quad (3-5)$$

其中  $\|Z\|_*$  是  $Z$  的核范数<sup>[70]</sup>，核范数的含义是矩阵特征值的和。

实际应用中很多数据都含有噪声和误差，因此低秩表示算法在设计时采用了如下的方法处理数据中的这些噪声：

$$\min_{J,E} \|J\|_* + \lambda \|E\|_{2,1} \quad (3-6)$$

$$s.t. J = Z, X = AZ + E \quad (3-7)$$

$$\|E\|_{2,1} = \sum_{j=1}^n \sqrt{\sum_{i=1}^d E_{ij}^2} \quad (3-8)$$

我们使用增广拉格朗日乘子<sup>[71]</sup>的方法来解这个优化方程，首先方程 3-6 的增广拉格朗日形式如下：

$$L(Z, J, E, Y_1, Y_2, \mu) = \min_{Z, J, E, Y_1, Y_2, \mu} \|J\|_* + \lambda \|E\|_{2,1} + \text{tr}[Y_1^t(X - XZ - E)] + \text{tr}[Y_2^t(Z - J)] + \frac{\mu}{2} (\|X - XZ - E\|_F^2 + \|Z - J\|_F^2) \quad (3-9)$$

本章通过非精确增广拉格朗日乘子方法求解方程 3-9，具体的求解过程如表 3-1 所示

---

**Algorithm 3-1: Inexact ALM<sup>[72]</sup>.**

---

Input: matrix  $X$ , parameter  $\lambda$ ;

Output: Low-rank representation  $Z$ , error matrix  $E$ .

---

Initialize: Set  $Z = J = 0$ ,  $E = 0$ ,  $Y_1 = Y_2 = 0$ ,  $\mu = 10^{-6}$ ,  $\max \mu = 10^9$ ,

$\eta = 1.1$ , threshold  $\varepsilon = 10^{-8}$

Step 1: Fix other variable and update  $J$ :  $\min_J \frac{1}{\mu} \|J\|_* + \frac{1}{2} \|J - (Z + Y_2/\mu)\|_F^2$ ;

Step 2: Update  $Z$ :  $Z = (I + X^t X)^{-1} (X^t X - X^t E + J + (X^t Y_1 - Y_2)/\mu)$ ;

Step 3: Update  $E$ :  $E = \min_E \frac{\lambda}{\mu} \|E\|_{2,1} + \frac{1}{2} \|E - (X - XZ + Y_1/\mu)\|_F^2$ ;

Step 4: Update parameter  $Y_1$ :  $Y_1 = Y_1 + \mu(X - XZ - E)$ ;

Step 5: Update parameter  $Y_2$ :  $Y_2 = Y_2 + \mu(Z - J)$ ;

Step 6: Update  $\mu$ :  $\mu = \min(\eta\mu, \max \mu)$ ;

Step 7: if  $\|X - XZ - E\|_\infty > \varepsilon$  and

$\|Z - J\|_\infty > \varepsilon$ , return to step 1.

---

表 3-1 低秩表示算法

### 3.2.2 低秩子空间聚类

本章第一节提到过低秩表示可以和子空间聚类结合来把原始数据聚到不同的簇中，这样可以提高子空间聚类算法的性能，因为经过对数据低秩特征的挖掘，数据的线性可分性将增强，这时再使用子空间聚类可以更好的把数据正确的聚类。同时低秩表示后的数据中的噪声和误差点经过了一层过滤，对过滤过的数据进行聚类也可以提高聚类算法的性能。本节就引入基于低秩表示的聚类算法。

由于矩阵  $Z$  的列向量空间可以反应数据间的相关性，因此基于得到的低秩转换矩阵，我们可以构建一个样本关系图，图中的每一个顶点表示一个样本点，每一条边表示样本间的相关性。表示这个图连接关系的邻接矩阵可以由如下方法构造：

$$Z = U\Sigma V^T \quad (3-10)$$

$$\tilde{U} = U\Sigma^{\frac{1}{2}} \quad (3-11)$$

$$W_{ij} = ([\tilde{U}(\tilde{U})^T]_{ij})^2 \quad (3-12)$$

其中  $U, \Sigma, V$  是  $Z$  的瘦奇异值分解,  $W$  是  $X$  的邻接矩阵。 $\tilde{U}(\tilde{U})^T$  被用来表达  $Z$  的列向量空间，平方的目的是避免负数值，得到邻接矩阵后，子空间聚类<sup>[73]</sup>的方法被用来将数据聚为两类，具体的算法过程如表 3-2 所示。

---

Algorithm 3-2: Subspace spectral clustering based on LLR<sup>[48]</sup>.

---

Input: matrix  $X$ , number of clusters  $k$ .

---

Step 1: Apply inexact ALM to  $X$  and get optimal solution  $Z$ ;

Step 2: Calculate affinity matrix  $W$  according to equation 3-10, 3-11, 3-12;

Step 3: Divide  $X$  into  $k$  clusters by subspace clustering algorithm.

---

表 3-2 基于低秩表示的自空间聚类

图3-1给出了一个由表 3-2 中方法聚类的一个例子，如图所示，虽然有误差和噪声的存在，属于不同类别的不同数据被该算法很好的聚为两簇，这个例子可以说明该算法是一种有效的无监督聚类算法，并且对离群点有很强的鲁棒性。

### 3.3 基于低秩子空间聚类的主动学习支持向量机

#### 3.3.1 所提出的主动学习算法

上一章中，我们已经介绍过主动学习支持向量机的基本原理，本章我们依然使用和上一章一样的主动学习模型和选择引擎：不确定采样。同时处于实际应用多为非线性数据的角度考虑，本章也采用基于核函数的支持向量机。但是值得一提的是本章采用批量式主动学习方法<sup>[74]</sup>。选择批量式主动学习方法的原因是实际应用中的数据往往含有不同程度的噪声，如果我们每次只选择一个样本来更新模型，会有一定的风险。如果被标记的是噪声数据，更新模型将不会提升模型的性能，有可能是模型陷入局部最优解，造成性能下降的同时也带来了不必要的资源浪费。

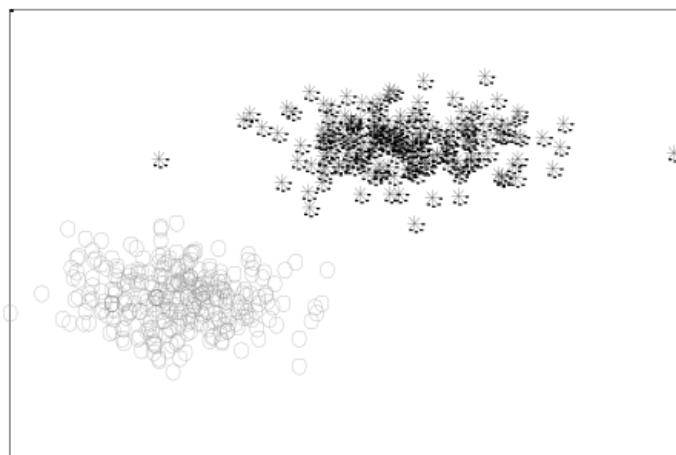


图 3-1 低秩子空间聚类效果示意

因此，为了减小这种情况发生的可能性，我们可以在每次更新模型时多选择几个样本，标记并根据这些样本更新模型，这就是批量式主动学习方法。同时采用批量式主动学习方法可以减少迭代次数，从而减少计算复杂度。因此对有噪声和误差的数据集采用更适合使用批量式主动学习。

正如本章引言中提到的，现有的大多数主动学习算法都对初始状态非常敏感，这就意味着如果没有选择合适的初始标签样本，主动学习过程很有可能得不到全局最优解。然而，对于却没有一个有效的方法解决这种问题。这也限制了主动学习的应用和推广。同时由于主动学习是半监督算法，具体来说只有被标记样本的信息才会被模型利用。但是主动学习过程中用到的样本总是数据集的一小部分，这就造成了大多数的样本中包含的信息被主动学习忽视，对学习过程没有帮助。

然而，大多数无标签样本中包含的信息是非常有价值的，至少它们可以被用于指导初始标签样本的选择，从而帮助解决主动学习对初始状态敏感的问题。本章为了利用这些无标签样本中的信息并且为主动学习设计一个初始状态选择方法，提出了一种全新的主动学习算法。本节中，我们就详细介绍所提出的算法。

在主动学习开始之前，低秩子空间聚类被用于整个数据集，从而把数据集分为不同的簇。之后，根据聚类结果，最有信息量的样本被选为初始的支持向量。样本的选择标准如下：

$$x = \underset{x \in X}{\operatorname{argmin}} |d_{x1} - d_{x2}| \quad (3-13)$$

其中， $d_{xk}$  表示样本点  $x$  和  $k$ -th 聚类中心的距离。该公式的含义是在所有的样本中选择离两个聚类中心的距离比较接近的样本，也就是位于两个聚类簇中间稀疏区域的样本作

为初始的支持向量。因为我们较难确定它们具体属于哪一个样本簇，所以它们包含的不确定性最多，信息量也最大。

同主动学习过程的样本选择一样，公式 3-13 也是基于不确定性采样选择样本，因为这些样本点位于不同簇的分界处，它们所包含的不确定性最强，因此它们最有可能影响支持向量机模型的结果。选择这些样本作为初始的支持向量可以降低选到噪声的概率，从而提高模型的性能。具体的算法流程在表 3-3 列出。

---

**Algorithm 3-3: Active learning based on low-rank representation and subspace clustering**

---

**Input:** Dataset  $X$ , the number of initial labeled data instance  $k$ , batch number  $m$ ;

**Output:** An SVM classifier.

---

**Initialize:** Divide  $X$  into two categories with algorithm 3-2,

select  $n$  data samples according to equation 3-13 and label them;

**Active learning process:**

**Step 1:**  $t = 0$ , and learn a classification model  $f^{(t)}$ , according to current labelled data;

**Step 2:** Set  $t = t + 1$ ;

**Step 3:** Actively select  $m$  data samples according to uncertainty sampling and label them;

**Step 4:** Learning new classification model from all labeled data;

**Step 5:**  $t = t + 1$

**Step 6:** Repeat step 3 - step 5, until the stopping criterion is met.

---

表 3-3 基于低秩子空间聚类的主动学习支持向量机算法

### 3.3.2 算法复杂度分析

本文所提出算法的复杂度由支持向量机和低秩表示的复杂度分别决定。支持向量机的时间复杂度在  $O(n^2)$  和  $O(n^3)$  之间，低秩表示的时间复杂度由奇异值分解决定，如果采用并行雅可比方法<sup>[75]</sup>，奇异值分解的时间复杂度会降到比  $O(n^2)$  还要小，此时所提出算法的时间复杂度就由支持向量机决定，即在  $O(n^2)$  和  $O(n^3)$  之间。

## 3.4 算法有效性验证

本文采用不同的标准数据集来测试所提出算法的有效性。所有的数据集都来自 LIB-SVM。本章的所有实验程序都采用 C++ 编写。以下两个现有的方法被用于和所提出算法进行对比：



- 基于主动学习的支持向量机：这种方法随机选择初始的支持向量，之后根据不确定性采样选择样本，标记并更新模型，这是最常用的主动学习方法；
- 随机采样：随机选择样本标记并更新模型，这等价于被动的支持向量机方法。

### 3.4.1 a1a 数据集的结果

首先，我们采用在 a1a 数据集上进行对比实验，这是一个二分类问题。这个数据集反映了成年人的各种信息和收入的关系。因此标签的信息是这个人是不是年收入超过 5 万美元，数据集的特征就是他或者她的相关信息，比如：年龄和受教育程度。这个数据集有 1065 个训练样本和 30956 个测试样本。由于原始的数据集是有标签的，这里我们在训练之前先隐去训练集中的标签。支持向量机的超参数由交叉验证<sup>[76]</sup>选取，这组实验中选择超参数值如下： $C = 100, \gamma = 0.01, n = 10, m = 5$ 。每次实验都重复进行 20 次，它们的误分率的平均值在图 3-2 中表示。

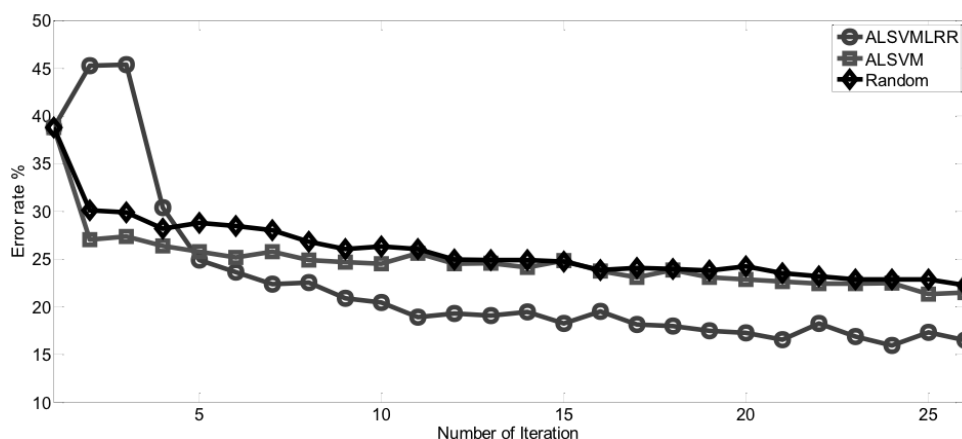


图 3-2 a1a 数据集不同方法的误分率

图 3-2 中 ALSVMLRR 表示本文所提出的方法，ALSVM 指的是传统的主动学习支持向量机，Random 指的是随机采样支持向量机方法。从图中可以看出，从第四次迭代开始所提出的算法就表现出来更低的误分率，直到学习结束。而且最后的误分率比其他方法低了接近 50%。

### 3.4.2 diabetes 数据集的结果

diabetes 储存了患者是否得癌症的信息。标签表示这个人是不是癌症患者。这个数据集有 768 个样本。支持向量机的超参数如下： $C = 10, \gamma = 0.1, n = 3, m = 3$ 。每次实验都重复进行 20 次，结果在图 3-3 中表示。

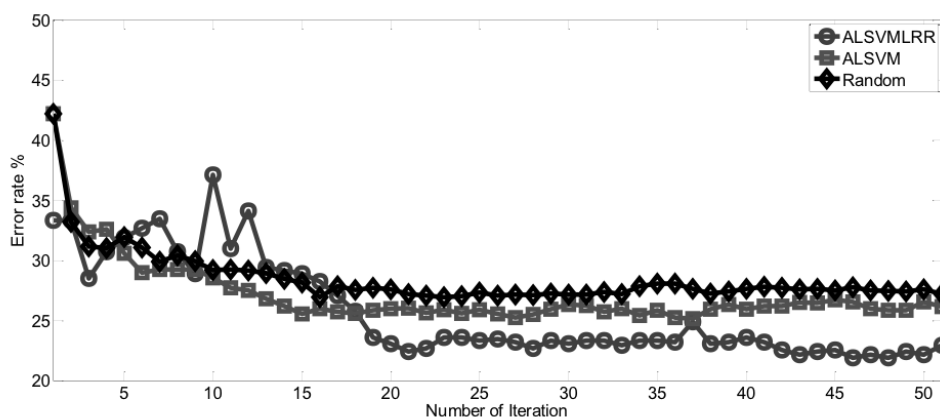


图 3-3 diabetes 数据集不同方法的误分率

从图 3-3 中可以得到，所提出的算法在开始时有较低的误分率，虽然在之后的 10 次迭代中结果有所波动，从低 18 次迭代开始它依然比其他的方法显示出了更好的性能，直到学习结束。这组实验结果比第一组实验更清楚的说明了在进行主动学习之前对数据集进行低秩子空间聚类可以使模型在一开始就有更低的误分率，在这种情况下，ALSVMLRR 自然会学到更好性能的模型，实验结果也证实了相对于其他方法，所提出的算法确实得到了更低的误分率。

### 3.4.3 german.numer 数据集的结果

这组式样在 `german.numer` 数据集上进行。这个数据集同样是一个二分类问题，它有 1000 个样本点，每个样本点有 24 维特征。这组实验同样进行 20 次重复式样，超参数值如下： $C = 1000, \gamma = 0.001, n = 3, m = 3$ 。结果在图 3-4 中表示。

图 3-4 同样可以验证所提出的算法是对初始状态鲁棒的。ALSVNMLRR 在一开始就取得了比其他方法更低的误分率，虽然之后的 7 次迭代有一些波动，最终这种方法还是在分类精度上打败了其他方法。从图中还可以看出，其他两种方法的初始误分率是 100%，这说明了如果初始支持向量选择的不对会导致学到的模型效果很差，甚至毫无效果。这种情况无疑会影响算法最终的效果。正如图中所示这两种方法的精度都低于 70%。

### 3.4.4 ionosphere 数据集的结果

`ionosphere` 数据集也是一个二分类问题。这个数据集记录了由 16 个高频触角在离子扬声器中产生的自由电子。这组实验中选择超参数值如下： $C = 1, \gamma = 0.1, n = 4, m =$

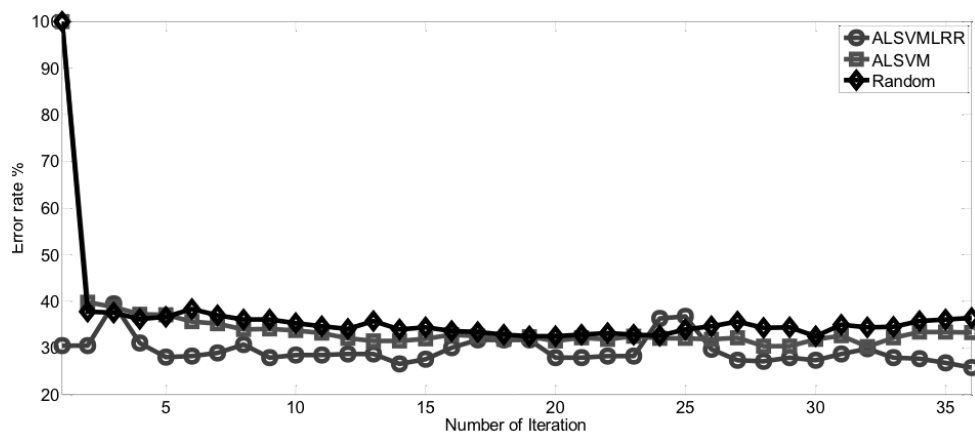


图 3-4 german.numer 数据集不同方法的误分率

4。

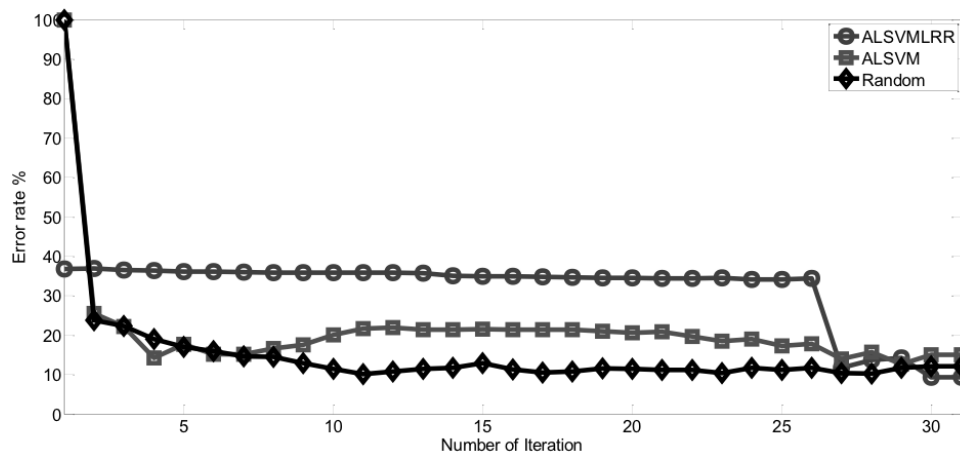


图 3-5 ionosphere 数据集不同方法的误分率

图 3-5 中是这组实验的结果。和上一组实验结果相似，所提出的算法在一开始就取得了最好的性能，并且在学习结束后也可其他方法保持着同级别的误分率。

### 3.4.5 初值对不同算法的影响

表 3-4 中记录了三种方法在四个不同数据集上最终结果的标准差。正如之前介绍的，每组实验进行 20 次重复实验。

这些统计结果说明了不同初始状态对两种传统方法的影响，不管是哪种数据集，这两种方法的标准差都在一定范围内波动。其中最大的标准差可以达到 12%。但是

Datasets	ALSVMLRR	ALSVM	Random
a1a	0	0.0468	0.0187
diabetes	0	0.0678	0.0400
german.numer	0	0.1199	0.1003
ionosphere	0	0.0255	0.0715

表 3-4 三种算法在四个数据集上结果的标准差

ALSVMLRR 的标准差为 0，这是因为每次实验，地址子空间聚类都会选择最有信息量的数据点作为初始的支持向量，所以每次训练的结果都相同。从表 3-4 可以得出，本章所提出的算法对初始状态的鲁棒性强于传统的主动学习算法。同时也可以看出尽管主动学习的过程中会主动选择信息量更大的样本点进行标记，但是对初始支持向量，传统的主动学习依然体现了被动学习支持向量机近乎相同的敏感性，这一现象侧面说明了设计合适的初始支持向量选择标准的重要性。

### 3.5 本章小结

上一节给出的实验结果证明了所提出算法可以达到高于现有方法的分类精度和更快的收敛速度。而且，所提出的算法解决了主动学习对初始状态敏感的问题<sup>[19]</sup>。低秩子空间聚类算法可以选择最有用的数据来训练初始模型。和其他方法相比，这个分类器可以得到更高的精度，因为大多是现有的方法都是随机选择初始样本点。基于更好的分类器选出的样本也会含有更多的信息量，从而进一步提高分类器的性能，这种精度的累计可以提升最终分类模型的性能。

然而本章中提出的方法只利用了数据集的无标签数据所包含的信息，并且这种对数据分布特性的挖掘只在初始化阶段。随着迭代的进行，更多的数据会被标记，因此更多的信息将被挖掘出来，如果可以适当利用这些信息，也可以进一步提升主动学习支持向量机的性能。这就使如何同时利用初始数据集中的信息和主动学习过程中发掘的信息成为一个重要的研究课题。

## 第四章 基于低秩转换的主动学习支持向量机

### 4.1 引言

机器学习的两大分支分别是有监督学习和无监督学习<sup>[77]</sup>。对于有监督学习来说，一个最核心的领域就是分类问题。对于分类问题来说，在给定训练数据集和每个训练样本的标签的之后，这个分类任务的目的是学习出从训练数据到样本的一个映射，利用学到的映射推测剩余测试样本的标签。比如垃圾邮件分类，图像识别都是分类问题的范畴。对于有监督学习和无监督学习，并没有一个明确的定义区别，但是无监督学习的一大特点是，在面临这类问题时，我们并没有明确的信息来指引我们的学习方向<sup>[78]</sup>，比如如何建立模型或者估计模型中的参数。

虽然有监督学习和无监督学习有不同点，但是它们都需要数据来推导出所需要的模型。通常情况下最耗时间和资源的过程之一是数据收集过程，尤其是当资源不充足时<sup>[9]</sup>。因此如何利用现有数据就变得十分重要，比如，在分类任务中，我们可以通过现有模型的信息选择合适的样本来继续学习过程，这种学习方式就是主动学习。

主动学习引入了一个选择引擎，在每次迭代过程中，选择引擎都会选择合适的样本作为标签样本。这使得主动学习比被动学习算法更有针对性。基于样本池的主动学习是主动学习中最主要的分支，并且已经被应用到分类问题中<sup>[79]</sup>，分类模型会选择样本池中的无标签样本，并询问它们的标签。在处理分类问题时，通常选择支持向量机作为主动学习的分类模型<sup>[49]</sup>。

然而传统的主动学习方法都忽略来数据集的数据分布信息，这种忽略数据分布信息的情况不仅会造成信息的浪费，而且有可能会引起主动学习陷入局部最优解的情况。<sup>[80]</sup>而且这些算法不仅忽视了数据集整体的分布信息，而且还没有对数据有标签样本的信息充分利用。虽然主动学习对比被动学习已经减少标签数据的成本，并且保证较高的准确度，但是造成这种信息的浪费还是对主动学习的性能和发展造成了很大影响，因此我们可以通过有效的应用数据集的分布特性来进一步提高主动学习的性能。

在本章中，我们提出了一种全新的主动学习方法，这种算法主要针对二分类问题。算法的设计思路是在每次迭代过程中，在标记了所选择的样本之后，在更新模型之前，我们先根据所得都的标签样本学习一个低秩转换。之后在更新下一次模型之前先把学习到的低秩转换应用于整个数据集，把整个数据集映射到一个更加线性可分的特征空间。之后在映射后的空间进行下一步迭代操作。随着迭代的进行，更多的数据样本会被标签，学到的低秩转换也可以更好的表达数据集的分布特性。引入低秩转换是为了把原有

的数据集映射到一个更加线性可分的空间去,在映射后的空间里,数据被分到不同的线性子空间里,在这个空间里进行之后的主动学习过程会减少主动学习的学习难度,这就是本算法设计的核心。为了验证算法的有效性,我们选择里其他的机器学习进行比较,并且在多个标准数据集上做了对比实验,实验结果说明了所提出算法的有效性。

本章的剩余内容安排如下,子章节 4.2 介绍了低秩转换算法,子章节 4.3 详细说明了所提出算法的过程,子章节 4.4 给出了在多个标准数据集上的实验结果。最后我们对本章内容进行了总结,并且讨论了未来可能的发展方向。

## 4.2 低秩转换算法

正如之前讨论过的,主动学习的主要目的是在尽可能保持模型效果的前提下减少标签成本。因此,如何通过更好的挖掘数据分布的特征是对达到这个目的有大帮助作用。然而,传统的主动学习方法不仅忽略了无标签样本池中样本的分布特性,也没有充分利用标签样本中的信息。

在本节中,为了挖掘这些有用信息,我们首先专注于有标签样本。对于传统的主动学习方法,有标签样本仅仅用于更新模型。在本章提出的算法中,我们先通过现有的有标签样本学习一个低秩转化  $\Psi$ 。得到这个转换  $\Psi$  之后,我们可以利用这个学习到的转换把数据集映射到一个特征空间,在这个特征空间中,这些数据根本不同的标签被分配到不同的子空间中,因此不同类别样本间的距离变得更大。在一个线性可分性更强的特征空间中,我们在进行后续的模式学习,会得到更好的模型性能。

低秩转换<sup>[81]</sup>的目的是找到高维数据的低维子空间。假设输入数据空间  $X$  的维度是  $d$ ,低秩转换可以用一个矩阵  $P(d \times d)$  表示,之后对于所有的标签数据,这个转换可以由以下的公式得到:

$$L_{\Psi} = (Px_i, y_i) \quad (4-1)$$

假设  $\{S_c\}_{c=1}^C$  是输入数据  $X = \{x_i\}_{i=1}^N$  的  $C$  个子空间,其中  $x_i \in \mathbb{R}^n$  而且  $X_c = \{x|x \in X, x \in S_c\}$ ,因此  $X = [X_1, X_2, \dots, X_C]$ 。  $X_c$  是  $X$  的低秩子空间,这些低秩子空间对分类和聚类非常的重要。如果可以得到这些子空间的信息,后续的分类或者聚类工作就会变得容易许多。但是对于绝大多数高维数据来说,这种自空间结构不是那么的清晰,所以我们需要一个合理的转换方法,把原始的数据映射到一个线性可分性更强的特征空间。

假设映射后的数据可以被写为如下形式:

$$PX = [PX_1, PX_2, \dots, PX_C] \quad (4-2)$$

为了找打这些空间,映射  $PX$  需要具备以下两种性质:

- 映射之后的每一类数据的距离尽可能的小，相关性尽可能大，这就意味着  $\text{rank}(PX_1), \text{rank}(PX_2), \dots, \text{rank}(PX_C)$  要尽可能的小；
- 映射之后的不同类数据之间的相关性要尽可能小，这就意味着  $\text{rank}(PX)$  要尽可能大。

因为上述两个特征，我们可以设计如下的目标函数来找到一个合适的转换矩阵  $P$ ：

$$\min_P \sum_{c=1}^C \text{rank}(PX_c) - \text{rank}(PX) \quad (4-3)$$

$$s.t. \|P\|_2 = 1 \quad (4-4)$$

其中，限制条件是为了避免  $P = 0$  的情况出现。

假设有两个有相同维度的矩阵  $A$  和  $B$ ， $[A, B]$  表示  $A$  和  $B$  连接在一起组成的矩阵，为了求解上述优化问题，我们先给出如下的定理<sup>[47]</sup>：

$$\text{rank}(A, B) \leq \text{rank}(A) + \text{rank}(B) \quad (4-5)$$

其中，等号只有在  $A$  和  $B$  不相交的时候才会出现，基于不等式 4-5，我们可以得到

$$\text{rank}(PX) \leq \text{rank}(PX_1) + \text{rank}(PX_1, PX_2, \dots, PX_C) \leq \sum_{c=1}^C \text{rank}(PX_c) \quad (4-6)$$

当矩阵相互独立时等号成立，因此只有矩阵相互独立时，公式 4-3 达到最小值， $P = 0$ 。然而，独立并不意味着不同子空间之间的距离最大。比如，有两条只在原点相交的线段，这就意味着两个线段相互独立。然而，它们之间的距离只有在夹角达到  $\pi/2$  时才达到最大。

因此仅仅独立无法满足距离最大的目标，我们希望的是不同的子空间尽量正交。同时由于矩阵的秩是非凸的，我们常常用矩阵的核范数作为秩的凸近似<sup>[82, 83]</sup>。于是，优化的目标方程可以转换为如下形式：

$$\min_P \sum_{c=1}^C \|PX_c\|_* - \|PX\|_* \quad (4-7)$$

$$s.t. \|P\|_2 = 1 \quad (4-8)$$

其中矩阵  $A$  的核范数等价于  $A$  所有奇异值之和。这里的限制同样是为了避免无意义的  $P = 0$  的情况出现。新构造的目标函数在矩阵的列向量都正交的情况下可以达到最小值。

然而，我们不难发现，目标函数 4-7 并不是一个凸函数<sup>[84]</sup>，因为该目标函数是由两个凸的项相减得到的。但是这种特殊形式可以通过基于次梯度的凹凸过程求解方法求解。<sup>[81]</sup>

凹凸过程的基本思想是用目标函数中的凹项的一阶泰勒展开代替该项<sup>[84, 85]</sup>，从而求解到结果。首先目标方程  $J(P)$  可以写为一个凸项和一个非凸项的和：

$$J(P) = J_{vex}(P) + J_{cav}(P) \quad (4-9)$$

$$J_{vex}(P) = \sum_{c=1}^C \|PX_c\|_* \quad (4-10)$$

$$J_{cav}(P) = -\|PX\|_* \quad (4-11)$$

于是根据上面介绍的凹凸过程的思想，目标函数 4-7 的子问题可以写为如下形式：

$$J_{sub}(P) = P^{(t+1)} = \min_p \sum_{c=1}^C \|PX_c\|_* - \partial\|P^{(t)}X\|_* X^T P^T \quad (4-12)$$

当  $t = 0$  是，让  $P(0) = I$  作为迭代的初始条件。但是因为核范数是不可微分的，为了求解 4-12，我们引入了矩阵核范数的次梯度。根据<sup>[86]</sup>，我们在表 4-1 给出求解核范数次梯度的方法。

---

Algorithm 4-1: Sub-gradient of matrix nuclear norm<sup>[86]</sup>.

---

Input: matrix  $A$ , threshold value  $\delta$ ;

Output:  $\partial\|A\|_*$ .

---

Step 1: Calculate Singular Value Decomposition (SVD) of  $A$ ,  $A = U\Sigma V^T$  and

$\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p)$ , where  $\sigma_1 > \sigma_2 > \dots > \sigma_p$  are singular value of  $A$ ;

Step 2: Let the number of singular values which are less than  $\delta$  is  $s$ , and  $k = \min(m, n)$ ;

Step 3: Let  $U = [U_1, U_2]$  and  $V = [V_1, V_2]$ , where the number of columns of  $U_1, V_1$  is  $s$ ;

Step 4: If  $s > 0$ , randomly create a matrix  $B$   $(m - k + s) \times (m - k + s)$ ,

then let  $B = \frac{B}{\|B\|}$  if  $s = 0$ , then  $B = 0$ ;

Step 5: Compute sub-gradient  $\partial\|A\|_* = U_1 V_1^T + U_2 B V_2^T$ ;

Step 6: Return  $\partial\|A\|_*$

---

表 4-1 求解矩阵核范数次梯度的算法

基于求解次梯度的算法， $J_{sub}(P)$  的次梯度可以由如下公式得到：

$$\sum_{c=1}^C \partial\|PX_c\|_* X_c^T - \partial\|P^{(t)}X\|_* X^T \quad (4-13)$$



---

Algorithm 4-2: Sub-gradient method for Concave -Convex subproblem<sup>[81]</sup>.

---

Input:  $P^{(t)}$ , step length  $\eta$  convergence precision  $\varepsilon$ ;

Output:  $P^{(t+1)}$ .

---

Step 1: Calculate  $P_0 = P^{(t)}$ ;

Step 2: Compute sub-gradient according to equation 4-13;

Step 3: Let  $P_1 = P_0 - \eta \partial J_{sub}$ ;

Step 4: If  $|J_{sub}(P_1) - J_{sub}(P_0)| > \varepsilon_1$ , then let  $P_0 = P_1$ , and return to step 2;

Step 5: Let  $P_1 = \frac{P_1}{\|P_1\|}$ ;

Step 6: Let  $P^{(t+1)} = P_1$  and return  $p^{(t+1)}$

---

表 4-2 通过次梯度求解凹凸子问题的方法

---

Algorithm 4-3: Computing  $P$  by Concave-Convex Procedure<sup>[81]</sup>.

---

Input:  $X = [X_1, X_2, \dots, X_C]$ , number of iteration  $a$ ;

Output:  $P$ .

---

Step 1:  $t = 0, P^{(t)} = I$ ;

Step 2: Compute  $P^{(t+1)}$  according to algorithm 4-2;

Step 3: If  $t < a$ , let  $t = t + 1$ , return to step 2;

Step 4: Let  $P = P^{(a)}$ , and return  $P$ .

---

表 4-3 计算转换矩阵  $P$  的算法

给定步长  $\eta$ , 计算凹凸子问题最优解的算法可以由表 4-2 得到:

基于以上的分析和给出的两个算法, 我们可以得到求解转换矩阵  $P$  的算法, 具体的算法流程如表 4-3 中所示。

在实际应用中, 根据实践经验, 求解转换矩阵算法中的参数  $a$  选择 1 – 3 时, 优化方程 4-7 趋向于稳定。

## 4.3 基于低秩转换的主动学习支持向量机

### 4.3.1 算法详述

正如引言中分析的那样, 合理的利用数据集中数据的分布信息可以提高主动学习的性能, 进一步降低标签成本。本章中我们为了更有效的利用有标签样本中的信息, 避免造成信息浪费, 提出了一种全新的基于低秩转换的主动学习方法。在主动学习的每一次迭代过程中, 除了更新分类模型, 标签数据还被用来学习一个转移矩阵  $\mathbf{P}$ 。在下次迭

代开始之前，我们先利用学到的转移矩阵把数据映射到不同的子空间。但是由于本文使用的低秩转化是线性方法，所以在进行主动学习的时候，我们还是和上两章一样，先把数据映射到核空间，之后更新分类模型。所提出算法的具体步骤在表 4-4 中给出。

---

**Algorithm 4-4: Active learning based on low-rank transformation**

---

**Input:** Unlabeled dataset  $X$ , set of initial labeled samples' index  $I_L^{(0)}$ ,

number of data samples selected each iteration  $k$ ;

**Output:** Classification model  $f^{(t)}$ , and transformation matrix  $P^{(t)}$ .

---

Step 1: Let  $t = 0$ ,  $P^{(0)} = I$ , and create  $I_U^{(0)}$  to store of index unlabeled data samples;

Step 2: Create labeled data samples set  $L^{(t)}$ , and learn  $f^{(t)}$  from  $L^{(t)}$  by SVM;

Step 3: Select  $k$  most uncertain data samples from  $U^{(t)}$ , according to uncertainty sampling;

Step 4: Update  $I_U^{(t)}$  and  $I_L^{(t)}$ ;

Step 5: Create tow matrix of data samples having different labels:  $X_-^{(t)}$  and  $X_+^{(t)}$ ;

Step 6: Let  $X_L = [X_+, X_-]$  and compute  $P$  according to algorithm 4-3;

Step 7:  $X^{(t+1)} = P^{(t+1)}X^{(t)}$ ;

Step 8:  $t = t + 1$ ;

Step 9: Repeat step 2-8, until stopping criterion is met;

Step 10: Return  $f^{(t)}$  and  $P^{(t)}$ .

---

表 4-4 基于低秩转换的主动学习支持向量机算法

传统的主动学习和表 4-4 中的算法的主要区别是在步骤 5-7。在第五步中， $X_+^{(t)}$  那些标签是 +1 的数据组成的数据矩阵， $X_-^{(t)}$  标签是 -1 的数据组成的数据矩阵。此外， $X_i^{(t)}$  表示  $X^{(0)}$  的第  $i$  列。

随着迭代的进行，不停的有样本被标记， $L$  中元素的个数也在不断的增加。在信息更多的情况下，转移矩阵  $P$  可以更好的表达数据集  $X$  的内在结构，这就意味着，随着  $P$  的更新，被映射后数据不同类别间的距离变的更大，同类别数据间的距离变的更小。因此，在此基础上学习到的分类模型的性能会更好。

本章所提出的算法和传统算法的另一个不同点是学习结束后我们不仅得到了分类模型  $f$ ，还得到了一个转移矩阵  $P$ 。利用这个转移矩阵，我们可以把新来的测试数据归到它属于的类别中，可以借以对模型的分类结果进行检验。

阈值  $\delta$ ，步长  $\eta$  和收敛精度  $\varepsilon$  是超参数，需要事先给定。这些超参数主要影响学习到的转移矩阵，对分类模型并没有直接的影响，在本章中，超参数的取值如下： $\delta = 0.01$ ， $\eta = 0.02$  and  $\varepsilon = 0.1$ 。

### 4.3.2 算法的时间复杂度分析

基于低秩转换的主动学习算法的复杂度主要由低秩转换和主动学习支持向量机决定。

对于，支持向量机，我们采用相对快速的方法 **LASVM**<sup>[87]</sup>。这种方法的时间复杂度和支持向量的个数  $s$  成正比。假设我们进行  $K$  次迭代，这时 **LASVM** 的时间复杂度和  $nsK$  成正比，其中  $n$  是数据集中的样本数。在支持向量机的最后一步解优化时，我们使用 **SMO**<sup>[88]</sup>。因此近似的时间复杂度在  $O(n^2)$  和  $O(n^3)$  之间。

对于低秩转换，时间复杂度主要取决于奇异值分解。对于一个矩阵  $A (m \times n)$  奇异值分解的时间复杂度是  $O(mn^2)(n \leq m)$ 。基于奇异值分解的复杂度， $\partial J_{sub}$  的复杂度是  $O(nd^2)$ ，其中  $X (d \times n)$ 。这是一个  $O(n^3)$  级别的复杂度。由于最后求解转换矩阵时要计算有限次的凹凸子问题，因此低秩转换的复杂度是  $O(n^3)$ 。

由于所提出算法每次迭代要计算一次  $P$  和一次  $f$ ，基于以上分析，所提出算法的时间复杂度是  $O(n^3)$ ，其中  $n$  是数据集中样本个数。

## 4.4 算法有效性验证

为了评估所提出算法的性能，我们在不同的标准数据集上进行了对比实验，所欲的数据集都是来自 **LIBSVM**<sup>[62]</sup>。同时为了衡量算法的有效性，我们用其他的常用的机器学习方法作为对比。给定批量次数  $k$  的前提下这些用来对比的算法如下：

- 主动学习支持向量机：这个方法选择  $k$  个不确定性最强的样本作为新的支持向量，用选择的样本更新模型，这是传统的主动学习方法，也是第二章中介绍的方法，这种方法方法被证实一些标准数据集上是有效的，是一种常用的主动学习方法；
- 被动学习支持向量机：在每次迭代过程中，随机选择  $k$  各样本，标记并更新模型，这是标准的支持向量机算法；
- 基于主成分分析的主动学习支持向量机：由于本章提出的算法基于流形假设<sup>[89]</sup>，因此我们选择另一种基于流形假设的主动学习方法作为对比。这种方法用主成分分析进行特征提取，之后在特征空间中进行后面的主动学习过程<sup>[90-92]</sup>。

### 4.4.1 DNA 数据集的结果

首先，我们在 **DNA** 数据集上测试我们的算法。**DNA** 数据集<sup>[62]</sup> 有 2000 各样本点，这些样本点被分为三个不同类别。由于本章提出的算法是针对二分类问题的，所以我们在这个数据集上分别进行三种实验，每组实验将一个类别的 **DNA** 和其他类别分开。比

如，在一组实验中，我们把其中一个类别的标签设为  $+1$ ，其他类别的标签设为  $-1$ ，这样我们就把问题转为了二分类问题。这种策略被广泛应用于以支持向量机作为算法的对分类问题。<sup>[93]</sup>

由于原始的数据集是有标签的，我们把整个数据集分为训练集和测试集，同时隐去训练集的标签，这是因为主动学习不需要所有数据样本的标签信息，所有在学习过程中，只有被选择的样本才会被加标签。支持向量机的超参数  $C$ ， $\gamma$  由网格搜索和交叉验证<sup>[76]</sup> 确定。本实验中超参数设置如下： $C = 100$ ， $\gamma = 1$ 。

在每次实验中，我们选择 10 个样本点作为初始的支持向量，同时在每次迭代中，无标签样本池中有 5 个数据点被选为新的标签样本。每组实验，我们都进行 100 次重复实验，然后给出这些实验的平均误分率和这些误分率的统计结果。

#### 4.4.1.1 DNA 1 vs Others

图 4-1 中给出的是从 DNA 数据集中分出第一类样本的实验结果。这就意味这当进行这组实验时，我们重新对数据样本加标签，属于第一类的数据被标记为  $+1$ ，第二和第三类的数据被标记为  $-1$ 。在图中，Proposed 表示本文所提出的算法，ALSVM 代表传统的主动学习算法，ALPCA 代表基于 PCA 的主动学习算法，Passive 代表标准的支持向量机。从图中，我们可以看到所提出的算法从第九次迭代开始就超过了其他的算法。我们还可以从图中得到，在迭代结束之后，本文所提出的算法在准确率和收敛速度都超过了其它算法。

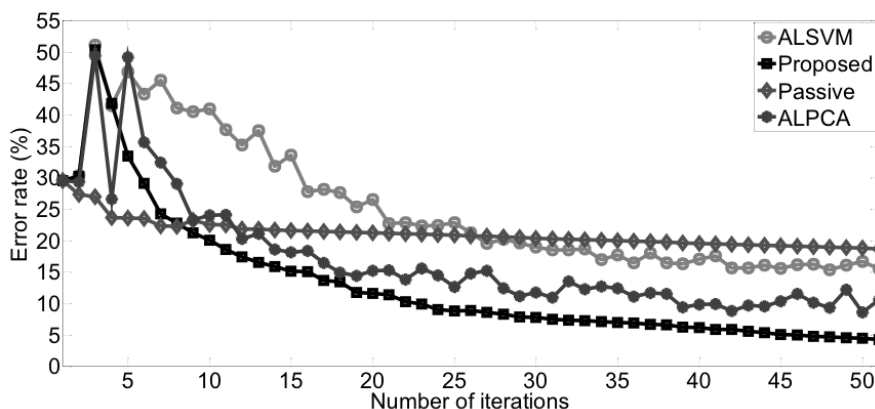


图 4-1 DNA 1 vs 其它的平均误分率

同时我们还计算了每次实验结束后得到的误分率的统计指标，这些指标是在 100 次实验的基础上得到的，其中 STDEV 代表最终结果的标准差。

	ALSVM	Proposed	Passive	ALPCA
MEAN	0.1545	<b>0.0426</b>	0.1867	0.1058
STDEV	0.0632	0.0061	<b>0.0049</b>	0.0465
MAX	0.3260	<b>0.0605</b>	0.1975	0.1680
MIN	0.0585	<b>0.0209</b>	0.1775	0.0405

表 4-5 DNA 1 vs others 在 100 次实验的误分率的统计指标

从表 4-5 中, 我们可以得到, 所提出的算法在四个指标中的三个都优于其它方法, 这证明了本算法不仅有很好的性能, 而且在这个问题上比较鲁棒。较低的标准差证明了所提出的算法不受初始状态的影响。

#### 4.4.1.2 DNA 2 vs Others

从 DNA 数据集中分出第二类的结果如图 4-2 所示。从图中可以得到, 相比于其他方法, 所提出的算法有最低的平均误分率和最快的收敛速度。表 4-6 是本组实验结果的统计指标, 我们可以从表中看出, 基于低秩转换的主动学习方法在各项指标上都是最好的, 也进一步说明了算法的鲁棒性。

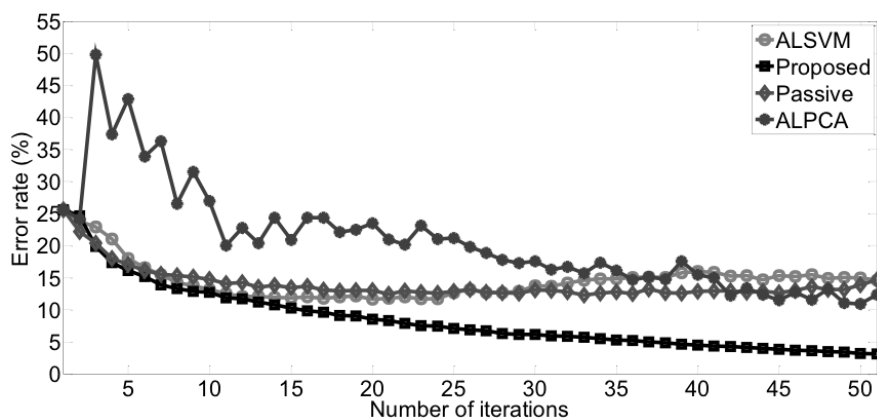


图 4-2 DNA 2 vs others 的平均误分率

#### 4.4.1.3 DNA 3 vs Others

图 2-4 给出了 DNA 3 vs others 的实验结果。在这种情况下, 所提出的算法在第二次迭代就达到了最好的效果, 虽然最后被 ALPCA 打败, 另一种基于流形学习的主动学习算法。本组实验说明了基于流形假设的主动学习算法性能的优越性。

	ALSVM	Proposed	Passive	ALPCA
MEAN	0.1462	<b>0.0318</b>	0.1473	0.1237
STDEV	0.0911	<b>0.0052</b>	0.0789	0.0472
MAX	0.4500	<b>0.4650</b>	0.3915	0.1835
MIN	0.0495	<b>0.0245</b>	0.0780	0.0575

表 4-6 DNA 2 vs others 在 100 实验中结果的统计指标

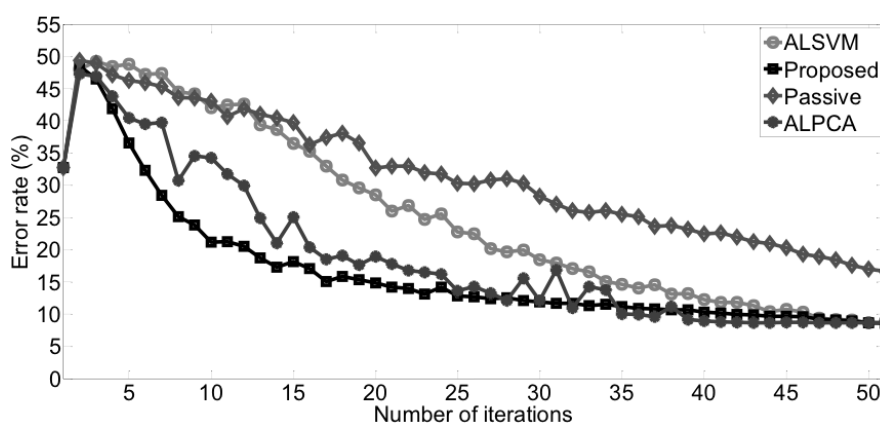


图 4-3 DNA 3 vs others 的平均误分率

表 4-7 中是本组实验的统计结果，从表中可以得到，虽然所提出的方法有最低的均值，但是两种方法的差距很小，这意味着它们在鲁棒性上基本有相同的性能。

#### 4.4.2 w5a 数据集的结果

W5a 是一个文本分类数据集，每个数据样本的标签表示一个网页是不是属于一个特定的类别。这个数据集包含 9888 个样本，每个样本有 300 非 0 即 1 的特征<sup>[94]</sup>。我们每个类别选择 5 个样本作为初始化的标签样本，在每次迭代中，都有 5 个样本被选为新的标签样本，用于之后的模型更新。本组实验的蚕食被设定为： $C = 100$ ,  $\gamma = 1$ 。我们同样进行 100 此重复实验。

这个数据集的结果在图 4-4 和表 4-8 中给出。从结果中可以得到，所提出的算法从第二次迭代开始就显示出了最好的分类特性，同时它还有最小的标准差，这说明了在这个数据集上，基于低秩转换的主动学习比其它的方法有更好的鲁棒性。

	ALSVM	Proposed	Passive	ALPCA
MEAN	0.0871	<b>0.0862</b>	0.1663	0.0865
STDEV	0.0428	<b>0.0096</b>	0.0658	0.0097
MAX	0.4525	0.1235	0.3485	<b>0.0910</b>
MIN	0.0605	0.0665	0.0815	<b>0.0580</b>

表 4-7 DNA 3 vs others 100 次实验的统计指标

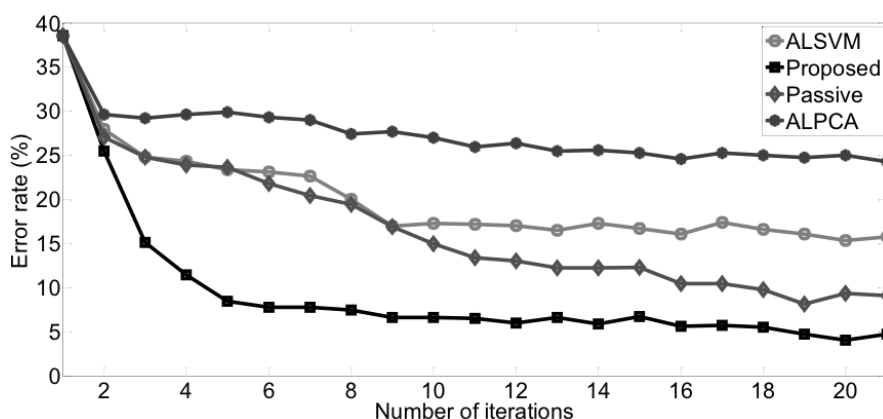


图 4-4 w5a 数据集的平均误分率

### 4.4.3 letter 数据集的结果

我们也在 letter 数据集上进行了实验。letter 数据集是一个手写字符识别数据集，它有 20000 个样本，分别是英文的 26 个字母，每一个数据点有 16 个特征<sup>[62]</sup>。在这个实验中，我们同样是选择了两个二分任务来测试本章提出的算法。它们分别是 ‘D’ vs ‘O’ 以及 ‘M’ vs ‘N’，这两组的字母都是比较难以识别的。

每组实验依然有 100 次重复实验，每此实验迭代 50 次，在这个数据集上的实验参数设置如下： $C = 100$ ,  $\gamma = 1$ 。初始化过程，我们依然选择 10 个初始标签样本，每个类别 5 个样本，在每次迭代过程中，选择五个样本作为新的标签样本。

#### 4.4.3.1 letter D vs O

letter D vs O 的实验结果如图 4-5 和表 4-9 所示，对于这个实验任务来说，所提出的方法虽然还是达到来最好的效果，但是优势没有像之前的数据集那么明显，不过无论是分类性能还是统计结果，三种主动学习方法都击败了被动学习方法，不仅说明了所提出算法的有效性，而且也体现来主动学习方法的优势。

	ALSVM	Proposed	Passive	ALPCA
MEAN	0.1575	<b>0.0474</b>	0.0913	0.2426
STDEV	0.1632	<b>0.0425</b>	0.1545	0.1230
MAX	0.7685	<b>0.1413</b>	0.9127	0.5319
MIN	<b>0.0187</b>	0.0195	0.0254	0.0311

表 4-8 w5a 数据集 100 次结果的统计指标

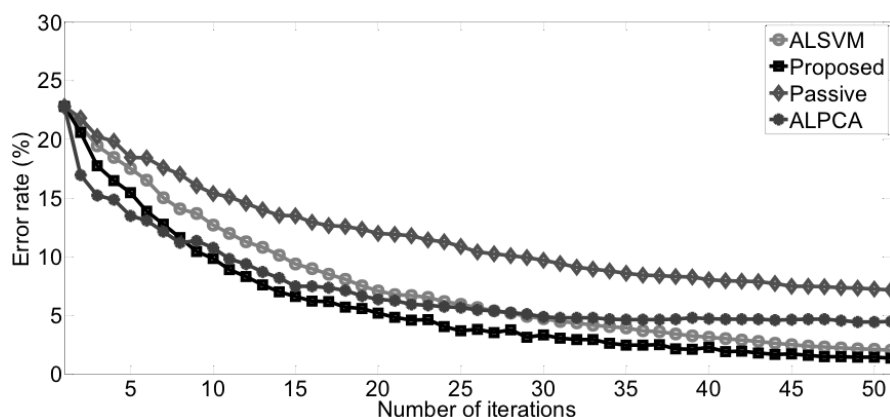


图 4-5 letter D vs O 的平均误分率

#### 4.4.3.2 letter M vs N

本组实验的任务是区分 letter 数据集中的 M 和 N，这组实验的结果在图 4-6 和表 4-10 中给出。在这组实验中，ALPCA 体现出了最好的性能，我们提出的算法比其它两种算法的误分率更低。这种现象说明了基于流形假设的算法在本数据集上的有效性。也说明了挖掘数据分布特性的重要性。

#### 4.4.4 其他标准数据集的结果

除了在以上的数据集，我们还在其它的标准数据集上进行了实验<sup>[62, 95]</sup>。这些实验的结果在表 4-11 中给出，这些数据集都是二分问题。在表中给出了每组实验的超参数，包括支持向量机的超参数，每次迭代的选择的样本数和迭代次数。同样，对这些实验，除了本文提出的算法，我们也都用了三种方法进行了对比实验，并给出了误分率和实验结果的标准差。

在表 4-11 中，每个单元有两个数值，上面一个是实验结束后的平均误分率，另一个是多次实验结果的标准差。从此表中，我们可以得到，所提出的算法在一共 18 组实验



	ALSVM	Proposed	Passive	ALPCA
MEAN	0.0205	<b>0.0137</b>	0.0716	0.0446
STDEV	0.0105	<b>0.0079</b>	0.0269	0.0181
MAX	0.0520	<b>0.0410</b>	0.1860	0.0734
MIN	0.0060	<b>0.0043</b>	0.0282	0.0188

表 4-9 letter D vs O 上 100 次实验结果的统计指标

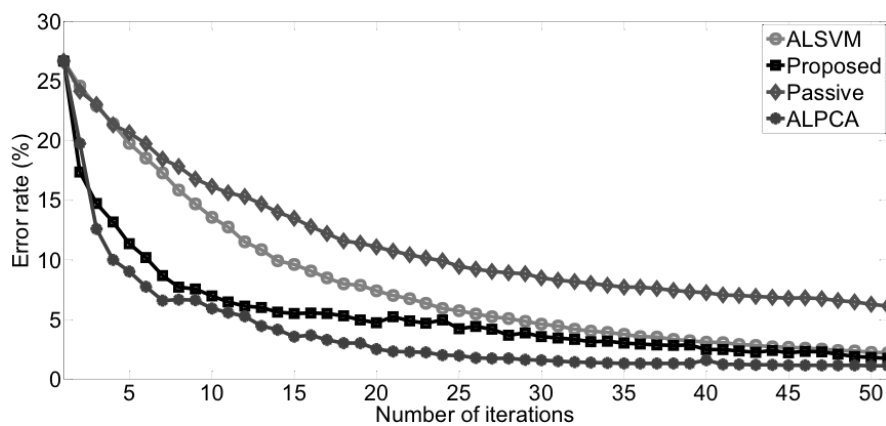


图 4-6 letter M vs N 的平均误分率

中的 13 组都有最好的分类结果，在 18 组实验中的 12 组有最低的标准差。这些结果说明了基于低秩转换的主动学习算法不仅有很好的性能，而且有很好的泛化能力以及对初始状态的鲁棒性，从表中，我们依然可以得出结论，主动学习的算法性能比被动学习更优越，因为在所有数据集上主动学习的误分率都低于被动学习。

	ALSVM	Proposed	Passive	ALPCA
MEAN	0.0219	0.0172	0.0611	<b>0.0111</b>
STDEV	0.0285	0.0079	0.0177	<b>0.0028</b>
MAX	0.1576	0.0586	0.1155	<b>0.0165</b>
MIN	<b>0.0041</b>	0.0058	0.0330	0.0074

表 4-10 letter M vs N 上 100 次实验的统计指标

## 4.5 本章小结

通过对上面给出的不同数据集结果的分析,我们可以得出所提出的算法比传统的主动学习算法效果更好,不仅如此,由于其对初始状态的敏感,不当的初始支持向量会导致主动学习学到一个性能很差的分类器<sup>[9, 80]</sup>。然而,每组结果标准差的比较结果说明了本章所提出算法并受初值选择问题的影响小。

通过引入低秩转换,我们可以充分的利用数据集中有标签样本的信息。通过把原始数据集映射到一个特征空间,可以使数据的分布结构变的更清晰,这样也更有利于分类算法学习到一个性能更优的模型。

但是值得注意的是,所提出算法的时间复杂度是  $O(n^3)$ , 由于算法涉及了大量的矩阵运算,算法的时间复杂度较高。如果遇到数据量比较大的情况,算法的数据储存成本也会相应提高。所以,之后的研究可以关注于如何提高算法的计算和储存效率。

此外,本章中,我们这研究了如何利用数据集中有标签样本的信息,但是无标签样本中也包含有用的信息。除了本文的前两章,像<sup>[89, 96, 97]</sup>等文献也分别讨论了如何利用数据集中无标签样本的信息。因此,以后的研究也可以挖掘如何同时利用有标签样本和无标签样本中的信息。

Dataset	C	r	k	Iterations	ALSVM	Proposed	Passive	ALPCA
ala	100	0.1	5	50	0.1514	<b>0.1439</b>	0.1822	0.1608
					0.0094	<b>0.0087</b>	0.0118	0.0259
australia	100	0.001	5	20	0.1629	<b>0.1428</b>	0.1757	0.1857
					0.0619	<b>0.0098</b>	0.0542	0.0817
breast-cancer	10	1	3	50	0.0233	<b>0.0204</b>	0.0360	0.0361
					0.0072	<b>0.0069</b>	0.0081	0.0079
diabets	10	0.1	3	40	0.2564	<b>0.2365</b>	0.2665	0.3671
					<b>0.0272</b>	0.0345	0.0406	0.0439
fourclass	1000	1	3	15	0.2357	<b>0.2046</b>	0.2391	0.2357
					0.0678	<b>0.0623</b>	0.0820	0.0678
german	1000	0.001	3	40	<b>0.2855</b>	0.3041	0.3541	0.4530
					<b>0.0445</b>	0.0978	0.0771	0.1025
heart	100	0.001	3	45	0.2430	<b>0.1735</b>	0.2944	0.2833
					0.1269	<b>0.0870</b>	0.1276	0.1255
ionosphere	1	0.1	4	50	0.1185	0.1818	<b>0.1092</b>	0.1439
					0.0826	0.0693	<b>0.0305</b>	0.0480
liver-disorders	100	0.1	3	45	0.4291	<b>0.3191</b>	0.4257	0.4439
					0.0779	<b>0.0482</b>	0.0724	0.0563
letter B vs P	100	1	5	50	<b>0.0037</b>	0.0206	0.0112	0.0288
					0.0091	0.0312	0.0044	<b>0.0038</b>
satimage 1 vs others	4	4	5	50	0.0246	<b>0.0069</b>	0.0549	0.0409
					0.0374	<b>0.0028</b>	0.0159	0.0209
satimage 6 vs others	4	4	5	50	<b>0.0451</b>	0.0654	0.0819	0.1360
					0.0142	<b>0.0113</b>	0.0495	0.0897
svmguide1	2	2	5	50	0.5501	0.3617	0.3297	<b>0.0640</b>
					0.1458	0.1839	0.0623	<b>0.0085</b>
svmguide3	128	0.125	5	50	0.4428	<b>0.1853</b>	0.3119	0.2963
					0.2068	<b>0.0167</b>	0.1964	0.1592
splice	10	0.01	5	44	0.1826	<b>0.1592</b>	0.2462	0.1608
					0.0489	<b>0.0179</b>	0.0548	0.0225
usps 1 vs 7	10	0.01	10	15	0.0021	<b>0.0018</b>	0.0084	0.0021
					0.0009	<b>0.0005</b>	0.0021	0.0006
usps 3 vs 8	10	0.01	10	15	0.0022	<b>0.0013</b>	0.0115	0.0023
					0.0011	0.0005	0.0035	<b>0.0003</b>
w2a	100	0.01	5	20	0.3310	<b>0.0566</b>	0.1967	0.4842
					0.2409	<b>0.0574</b>	0.2422	0.1567

表 4-11 其它数据集上的结果



## 第五章 主动学习深度神经网络-对抗深度学习

### 5.1 引言

深度神经网络以其强大的性能已经在很多应用领域都远远的超过了传统的机器学习算法<sup>[27, 98-102]</sup>。这些领域包括计算机视觉, 自然语言处理, 生物医学等。随着深度神经网络越来越被人认可, 它也被应用到一些关键的领域, 比如金融中的风险控制, 安保系统中的人脸识别。这些领域不仅要求所使用的算法有很好的性能, 也要求算法有一定的鲁棒性, 假如所使用的算法会被轻易的攻击, 那么它也无法满足这些领域的需求, 甚至造成很严重的后果。但是, 最近的文献<sup>[103]</sup> 却揭露了深度神经网络的一个本质漏洞。攻击者可以利用这个漏洞, 设计特定的对抗样本来攻击神经网络。

给定一个深度神经网络模型, 并且已知模型的参数, 一个攻击者可以通过网络结构找到模型的弱点和那些被模型认为是重要的特征。获悉这种信息之后, 这个攻击者可以通过对原始样本做很小的更改来构建一种特定的对抗样本。而深度神经网络对这种对抗样本的效果很差, 比如分类问题, 神经网络的对抗样本的分类不仅会产生误分类, 而且是以很高的置信度相信这个样本属于错误的类别。同时, 这种对抗样本可以在同一数据集或者同一数据集的子集训练出的多个模型上产生攻击效果。对抗样本的存在对神经网络在安全相关领域的应用前景蒙上了一次阴霾。假如一个用神经网络进行特征提取的风险控制系统, 我们可以通过产生对抗样本使神经网络提取错误的信息, 从而降低后续操作的准确性, 导致整个系统输出错误的结果。同时我们还可以通过对抗样本骗过人脸识别系统, 从而使基于人脸识别的安保系统失效。

近期的研究对这种现象的存在原因, 提出了一些猜测, 其中<sup>[104]</sup> 提出深度学习存在这种问题的原因是深度学习的线性本质。虽然深度学习的激活函数大多数是非线性函数, 但是当深度神经网络被训练的非常好时, 它一般会体现出很强的线性特性。神经网络的中参数会使网络每一层的输出在激活函数的线性部分, 这样才会使不同的样本有不同的输出。这种线性本质使神经网络存在特定的盲区, 攻击者就是利用这些盲区来攻击神经网络算法。基于这种解释, 一些研究<sup>[105-107]</sup> 提出了减小神经网络盲区的方法来减小对抗样本对神经网络的影响。具体的算法是在训练时, 不仅使用原始的训练集, 同时在每次迭代中, 根据当前的模型生成对应的对抗样本。

这种主动选择的样本的训练策略和主动学习的思路一致。回顾本文之前的内容, 主动学习是一种学习思路和框架, 只要是学习算法在学习过程中有主动选择样本的步骤, 我们都可以把这种学习算法纳入主动学习的范畴。本文之前几章研究的主动学习支持向

量机是在支持向量机模型的训练过程中,主动选择信息量最大的样本,基于的选择引擎是不确定性采样。本章提到的对抗深度学习是主动学习思想在深度神经网络中的实现,和之前的支持向量机模型相比,本章的深度学习算法选择了不同的选择引擎,本章的算法不再选择不确定性最强的样本,而是选择对当前模型威胁最大的样本,即当前模型下的对抗样本。把这种选择引擎选出的样本重新当作训练样本,训练模型,可以有效的减小模型的盲区大小,增强模型的非线性程度,从而减轻对抗样本对模型的攻击效果。

但是这些研究中的方法虽然可以从一定程度上提高模型对对抗样本的分类性能,但是它们并不是最理想的解决方法,因为深度学习模型的盲区总是无穷大的,虽然现有的方法可以有效的减小模型的盲区,但是攻击者还是可以在改进后的模型上选择新的对抗样本,即改进后模型的盲区。这并没有从本质上解决这个问题,因此这些研究中提出的算法还是存在一定的风险。在本章后面的小节,我们也会分析这些方法存在的问题,通过实验证明它们还是会被攻击。

因此,本章的研究目的是如何从本质上解决这个问题,本章研究的出发点是设计一种深度学习算法,使攻击者无法基于本算法训练出来的模型生成对应的对抗样本。具体的算法是在深度神经网络结构之前加上一层输入特征选择层(**Random Feature Nullification**)。这个输入特征选择层的操作是在每个样本输入到神经网络之前先按照一个特定的比例随机扔掉一些样本,之后基于处理后的数据训练一个深度神经网络模型。这个输入特征选择层使整个神经网络引入了不确定性。这种不确定性会使攻击者在攻击模型使无法生成针对模型的对抗样本,因为模型的一部分信息是不可知的,这就大大减少了对抗样本对模型的攻击性。

本章的后续章节会证明为什么无法基于本算法训练的模型生成对抗样本,同时也会从实验上证明本算法的有效性。因为这种设计是把一个输入特征选择层和后续的深度神经网络栈式的组合到一起,因此它对神经网络的结构不产生影响,这就意味着我们依然可以把本算法中放入主动学习的框架,从而结合本算法和基于主动学习的深度神经网络的优势。

本章的后续内容安排如下,在子章节 5.2 介绍对抗深度学习的基本概念和现有的对抗深度学习算法,接着在 5.3 中,我们介绍本章提出的算法,并且证明无法基于本算法生成相应的对抗样本。之后 5.4,我们通过实验证明算法的有效性,在 5.5 中把算法应用到一个计算机安全的重要领域:恶意软件检测中。最后对本章进行总结。

## 5.2 对抗深度学习

### 5.2.1 对抗样本

虽然一个训练的很好的神经网络可以学习到训练样本所包含的分布特性，从而在面临新的样本时，准确的判断出该样本所属的类别，但是一个深度神经网络有本质的弱点，就是通过对原始的样本做微小的修改就可以使神经网络产生误分，这种修改的幅度往往是人眼无法察觉的<sup>[103]</sup>。这种样本被称作对抗样本。对抗样本存在的原因是深度神经网络的输入空间是无界的<sup>[104]</sup>。

因此，神经网络总是存在特定的盲区，盲区中的样本是神经网络无法识别的，即使有些样本在人眼看来是属于特定类别的。同时，一个训练的很好的神经网络往往表现的非常线性，这也使神经网络模型的盲区更大。基于神经网络的这种特性，攻击者可以通过训练好的神经网络模型的信息，生成特定的对抗样本。之后用这些对抗样本攻击神经网络，可以使神经网络失效。更准确的说，有文献<sup>[104]</sup>显示，攻击者可以通过标准的优化过程找到攻击性最强的对抗样本。在对分类问题中，这种对抗样本可以使神经网络产生误分，把这个样本随机的分到除了正确的分类之外的其它类别，或者是特定的类别。

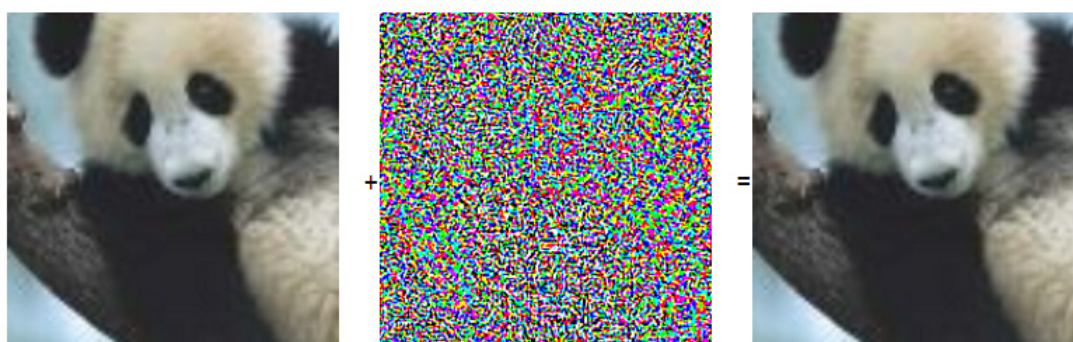


图 5-1 熊猫的对抗样本示例

图 5-1<sup>[103]</sup> 给出了一个对抗样本的例子，原始的图片是一只熊猫。在对其进行特定的修改后，虽然该样本还是一只熊猫（人眼可以辨别出来是一只熊猫），神经网络确以很强的置信度把样本分到了猿猴的类别。这是对对抗样本的一个典型例子，说明了对抗样本的特性和强大的攻击性。

此外，还有文献<sup>[103]</sup>表明，有相同目标函数，用相同数据集或者数据集子集训练而成的深度神经网络模型，比如识别相同图像集的模型。它们都模仿了相近的非线性函数。因此，从其中一个模型生成的对抗样本，可以对其它的模型产生攻击性，虽然这些模型的超参数和参数都不相同。因此，给定一个深度学习模型，我们可以生成大量的跨

模型对抗样本，这个特性使神经网络算法防御对抗样本的攻击变的更加困难。

对抗样本是通过计算神经网络的目标函数对输入的导数得到的，这个导数值代表了高维数据空间中的一个方向。在这个方向上，一个微小的变化就会导致神经网络产生完全不同的结果。这个特定方向代表了使一个神经网络失效的最有效的方向。发现这个方法是把目标函数对每一层的导数信息从最后一次一直传到输入层，之后在输入层的导数值被加到原始样本上，生成一个对抗样本。

更具体的来说，假设神经网络的目标函数被定义为  $\mathcal{L}(\theta, X, Y)$ ，其中  $\theta$ ， $X$  和  $Y$  表示模型的参数，输入和标签。一般来讲，对抗样本可以通过在原始样本上加一个很小的扰动  $\delta X$  生成。在文献<sup>[104]</sup>中提出了一种直观的有效的生成对抗样本的方法，该方法叫做 **fast gradient sign**，具体的生成方式如下：

$$\begin{aligned}\delta X &= \phi \cdot \text{sign}(\mathcal{J}_{\mathcal{L}}(X)) \\ \mathcal{J}_{\mathcal{L}}(X) &= \nabla_x L(\theta, x, y)\end{aligned}\tag{5-1}$$

其中  $\delta X$  通过计算目标函数对输入  $X$  的偏导的符号函数并在其上乘一个小的系数  $\phi$  得到的。 $\mathcal{J}_{\mathcal{L}}(X)$  表示目标函数  $\mathcal{L}(\cdot)$  对输入  $X$  的导数。 $\phi$  控制对原始样本更改的幅度。对抗样本生成的公式如下：

$$\hat{X} = X + \delta X\tag{5-2}$$

对抗干扰代表着在原始样本上所加的干扰的方向。这个向量代表的是目标函数  $\mathcal{L}(\cdot)$  对输入  $X$  最敏感的方向。然而，需要注意的是  $\delta X$  必须要比较小，否则在原始样本上加上相应的干扰  $\delta X$  会使样本发生过大的变化，使生成的对抗样本可以被人眼发现，这样的对抗样本会失去它的意义。

### 5.2.2 主动学习深度神经网络：主动选择对抗样本

为了减小对抗样本对深度神经网络的影响，提高模型的鲁棒性。近期有很多研究提出了一些相对有效的方法，其中最有效且被广泛应用的是对抗性训练 (**Adversarial training**)。如引言中介绍的，对抗性训练是在训练模型时基于当前的模型，选择对应的对抗样本，并且把生成的对抗样本作为训练样本，和原始样本一起训练模型。这一设计方法利用了主动学习的思路，可以说是主动学习在深度学习的实现。

对抗学习是通过生成对抗样本对数据集进行扩充<sup>[103]</sup>。更准确的说，这种数据扩充方法不仅可以被认为是主动学习在深度学习中的应用，而且可以被当作一种正则化的方法。它相当于是于在原始的目标函数上加了一个正则项，这个正则项的目的是惩罚目标函数对输入样本的导数，这个被惩罚的导数代表了目标函数在输入空间最敏感的方向，也



就是在这个方向上对输入样本进行修改会对模型的性能产生最大的影响。比如，最近的文献<sup>[104]</sup>，提出了一种全新的目标函数，如下：

$$\hat{L}(\theta, x, y) = \alpha L(\theta, x, y) + (1 - \alpha) L(\theta, x + \text{sign}(\mathcal{J}_{\mathcal{L}}(X))) \quad (5-3)$$

从公式中可以看出，这个新的优化目标方程直接把原始的目标函数和一个正则化项拼到一起，其中正则化项正是之前介绍的对抗样本。通过这种方法，神经网络的最强的对抗样本被当作了训练集的一部分，这样训练出来的神经网络模型就可以处理最难分辨的样本。当然有效的算法不止这一个，不过其它的算法基本是在这个算法基础上的改进，所以基本的思想都是一致的，都是通过主动的增加训练集来提高模型的性能。比如研究工作<sup>[108]</sup>中只使用了对抗样本来训练模型，而不是对抗样本和原始样本共同训练模型，还有其它的算法<sup>[105, 109]</sup>对这种基本算法进行了不同程度的改进。

在文献<sup>[110]</sup>中提出了一种统一的框架，它定义了如下的目标函数：

$$L_{DG}(t, d, \theta) = \lambda_0 L_0(t, d, \theta) + \lambda_1 R_1(J_{L_1}(t, d, \theta)) + \dots + \lambda_m R_m(J_{L_m}(t, d, \theta)) \quad (5-4)$$

其中  $L_1, L_2, \dots, L_m$  是代价方程， $R_1, R_2, \dots, R_m$  是正则化项， $J_{L_i}$  表示代价方程  $L_i$  对输入数据的偏导数值， $d$  表示了输入数据。这个目标函数是对上述方法的高度概括，其中如果代价函数的个数取 1，正则项取为无穷范数，该目标函数就等价于方程 5-3。

还有其它的算法都可以套入该目标函数提出的框架。如前所述，文献<sup>[108]</sup>只采用了对抗样本没有使用原始样本，带入到这个框架，相当于是取  $r = 1$ ， $\lambda_0$  和  $\lambda_1$  取适当的值，从而消去目标函数中的第一项。文献<sup>[106, 107]</sup>提出了一种改进的优化函数，这种优化函数通过限制对抗干扰的模长，并保证最小的代价值，在求解时，为了使问题可解，并且保证一定的速度，它们忽略了二次导数项，只计算一次导数，这就相当于是和<sup>[108]</sup>相同的求解过程，同样可以套入统一框架中。

<sup>[105, 109]</sup>中提出的方法，惩罚了目标函数对输入的梯度的斐波那契范数，但是它们的做法是一层一层的求导数，做法和自编码器的一种 **contractive auto-encoder** 方法一样，这种一层一层的方法还是在对梯度加正则项，因为框架 5-4 也是对梯度加特定的正则项，所以这些方法也可以套入统一框架中。目标函数 5-4 的具体解法可以参考<sup>[110]</sup>，这里不在详细介绍。

### 5.3 基于 RFN 的对抗深度学习

图 5-2 说明了一个使用本文设计的随机去掉固定比例特征 (**random feature nullification**) 的方法的深度神经网络模型。不同与标准的深度神经网络模型，它在输入样本和第一个隐含层之间引入了额外的一层。这一个外加的隐含层具有不确定性，它的作用是

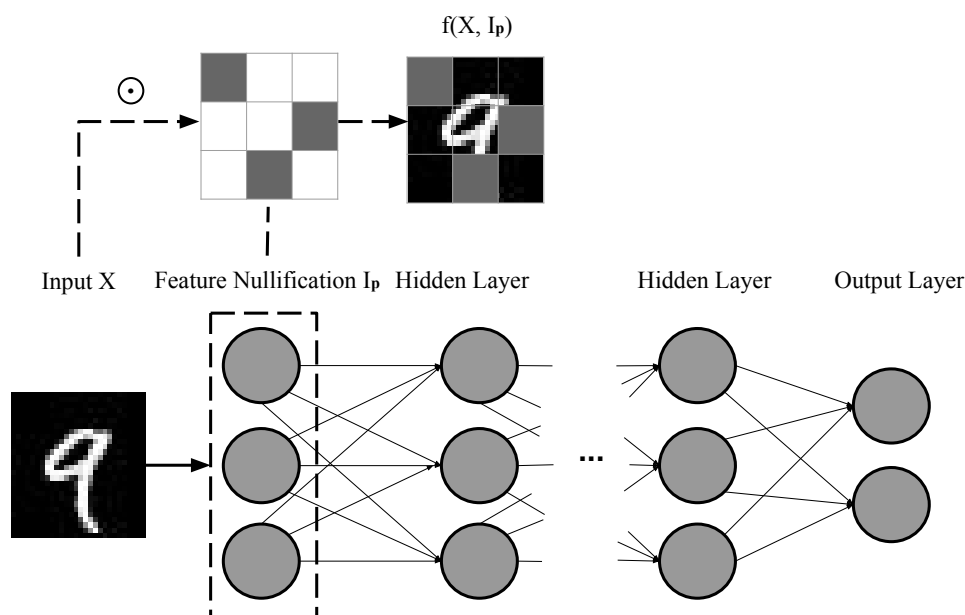


图 5-2 基于 random feature nullification 的深度神经网络模型

在标准深度神经网络中引入随机性。由于在训练和测试时都会进行 RFN 的操作，因此在训练和测试时，模型都引入了随机性。这就是本设计的核心。值得注意的是为了保持训练样本和测试样本同分布，在训练和测试时要保证随机丢掉的样本的比例是一定的，比如在训练时，每一个样本都扔掉 20% 的特征，那么在测试时也要丢掉相同比例的特征。同时值得注意的是，我们只在输入时丢掉样本。举例如下，假设有一个图像识别任务，当一个原始输入图像通过我们设计的这一层时，图像中一定比例的像素点会被随机丢掉，之后生成的特定的图像会被用于训练后续的神经网络。

决定输入样本特征丢弃的比例的参数是  $p$ ，对于每一个模型都有唯一的固定的丢弃参数。在对整个算法系统有一个大概的了解之后，我们再详细的介绍如何训练一个基于 RFN 的深度神经网络。之后我们解释为什么所设计的算法是可以抵抗对抗样本的，最后我们分析我们的方法和其它方法的不同之处。

### 5.3.1 模型描述

给定一个如下的输入样本  $X \in \mathbb{R}^{N \times M}$ ，random feature nullification 代表的操作是对输入样本  $X$  进行元素级别的更改，所依照的标准是  $I_p$ 。这里， $I_p$  是一个面具矩阵，它和  $X$  有相同的维度 (i.e.,  $I_p \in \mathbb{R}^{N \times M}$ )。  $I_p$  中的每一个元素是独立同分布的，每一个元素是一个随机变量。这些随机变量服从 Bernoulli 分布  $B(1, p)$ ，值得注意的是每个元素的  $p$  值相同。更具体的，在本章中， $I_p$  中的每一个元素都以概率  $p$  取到 0，以概率  $1 - p$

取到 1。

从图 5-2 中，我们可以看出，**random feature nullification** 可以被看成这样的深度神经网络的一个隐含层，只是在该隐含层中的操作和标准的操作不同。因此，我们可以得到如下的目标函数：

$$\min_{\theta} \mathcal{L}(\theta, f(X, I_p), Y). \quad (5-5)$$

其中  $Y$  和  $\theta$  分别是输入  $X$  的标签和对应的参数，这些参数需要在训练过程中确定。方程  $f(X, I_p)$  代表了一个随机丢掉输入数据特征的操作，这个函数的形式是： $f(X, I_p) = X \odot I_p$ 。其中， $\odot$  表示 *Hadamard-Product*。

图 5-2 中的神经网络可以通过随机梯度下降方法训练，这和标准神经网络的训练方法相同。唯一的不同是，对于每个训练样本，我们都基于模型  $p$  的随机的选一个  $I_p$ 。在训练过程的前向和后向过程中保持这个  $I_p$  值，直到本个样本训练结束，下一个样本到来。通过这种方法，我们可以计算目标函数  $\mathcal{L}(\theta, f(X, I_p), Y)$  对参数集  $\theta$  的梯度，并且根据计算出的梯度更新  $\theta$ 。

这样的设计可以保证模型中目标函数的残差传递到 **RFN** 的前一层，也就是含有参数的隐含层的最开始一层。这就保证了所有的参数可以得到更新，但是无法计算出目标函数对输入的偏导值，因为随机变量是不可导的，这就是本设计可以抵抗对抗样本的关键。我们在下一节中对这个关键点进行具体分析。

### 5.3.2 对抗样本抵抗性分析

本节给出本章设计算法可以抵御对抗样本的具体原因。为了生成一个对抗样本，一个攻击者必须计算一个对抗干扰，并且把这个干扰加到原始的样本上。正如在方程 5-1 中给出的那样，这个对抗干扰是通过计算深度神经网络的目标函数对输入的梯度得到的。基于这一点，攻击者可以计算代价函数  $\mathcal{L}(\theta, f(\tilde{X}, I_p), Y)$  对输入样本  $\tilde{X}$  的偏导数。对于本章提出的方法，基于链式法则，这个过程可以用如下的公式表示：

$$\begin{aligned} \mathcal{J}_{\mathcal{L}}(\tilde{X}) &= \frac{\partial \mathcal{L}(\theta, f(\tilde{X}, I_p), Y)}{\partial \tilde{X}} \\ &= \mathcal{J}_{\mathcal{L}}(f) \cdot \frac{\partial f(\tilde{X}, I_p)}{\partial \tilde{X}}. \end{aligned} \quad (5-6)$$

其中  $\mathcal{J}_{\mathcal{L}}(f) = \frac{\partial \mathcal{L}(\theta, f(\tilde{X}, I_p), Y)}{\partial f(\tilde{X}, I_p)}$ 。值得注意的是  $\tilde{X}$  表示的是原始样本，同时可以通过把  $\phi \cdot \text{sign}(\mathcal{J}_{\mathcal{L}}(\tilde{X}))$  加到  $\tilde{X}$  上面得到一个相应的对抗样本。

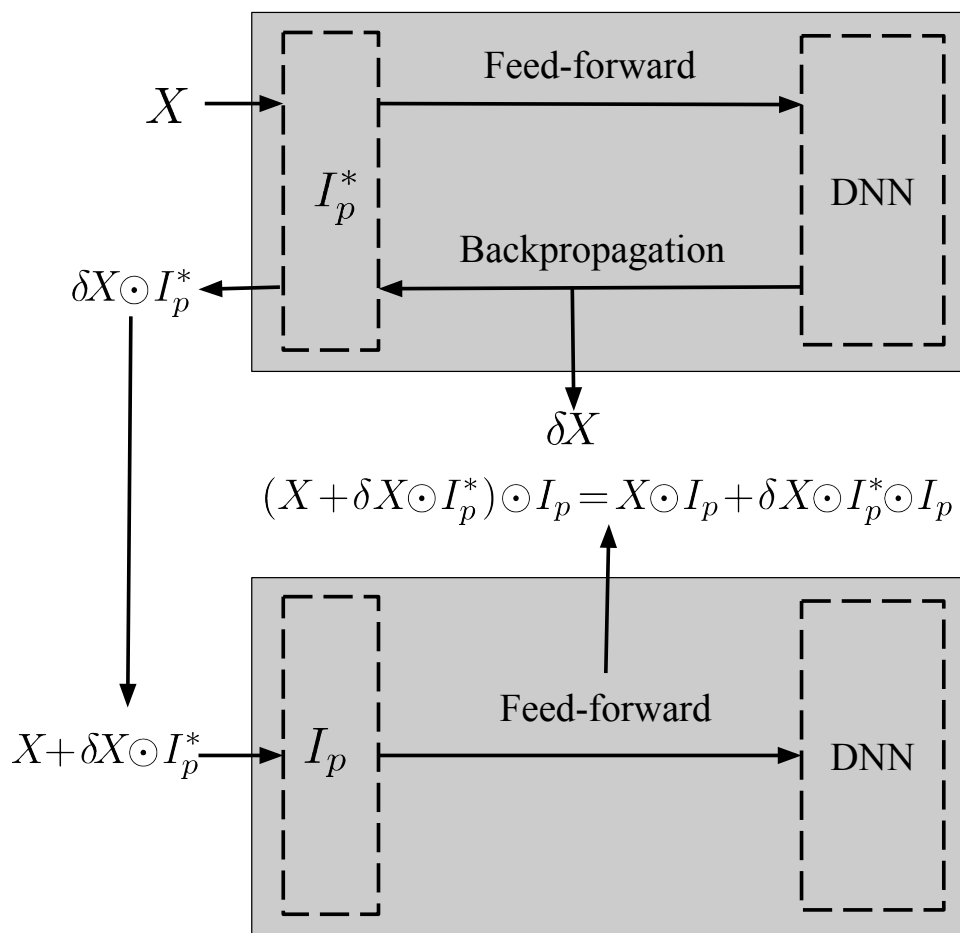


图 5-3 基于 random feature nullification 的神经网络中对抗样本的生成和测试示例

在公式 5-6 中,  $\mathcal{J}_{\mathcal{L}}(f)$  可以被轻易计算出。但是,  $\frac{\partial f(\tilde{X}, I_p)}{\partial \tilde{X}}$  中有随机变量  $I_p$ , 这个随机变量使这一项的梯度值无法计算。从另一个角度可以得到, 随机变量  $I_p$  阻止了模型残差的传递, 同时也阻止了攻击者针对这个模型生成相应的对抗样本。由于这个随机变量的存在, 基于这种算法生成对抗样本的唯一放大就是在随机变量的分布中随机选择一个固定的  $I_p$ , 并且用这个固定的  $I_p$  代替原本的随机变量。但是即使一个攻击者可以通过这种方法生成一个替代的对抗样本, 它不是这个模型的最好近似, 具体的检测方法在图 5-3 中给出。

假设一个攻击者生成了一个样本, 其中  $I_p^*$  代表了对随机变量  $I_p$  的替代, 值得注意的是  $I_p^*$  的  $p$  和  $I_p$  的  $p$  要保持一致, 这是为了保证训练样本和测试样本有相同的分布, 使这个深度学习过程有意义。基于这个替代值, 攻击者可以生成原始样本的对抗干扰  $\delta X \odot I_p^*$ 。如图 5-3 中表示的那样, 我们把基于 random feature nullification 的深度学习神经网络当作一个黑箱。  $\delta X$  表示对这个黑箱最强的攻击方向, 也就是模型在输入空间最敏感

的方向。于是  $\delta X \odot I_p^*$  就是攻击者用来攻击模型的对抗干扰，

在把这个样本输入到模型之后，模型会先进行 **random feature nullification** 操作，之后在把操作后的样本输入到后续的隐含层中，这时的样本可以被表示为：

$$(X + \delta X \odot I_p^*) \odot I_p = (X \odot I_p) + \delta \odot I_p^* \odot I_p. \quad (5-7)$$

其中,  $X \odot I_p$  对原始样本进行过丢弃操作的样本,  $\delta \odot I_p^* \odot I_p$  代表在这个样本上加的对抗干扰。正如之前提到的,  $\delta X$  是对这个深度神经网络攻击性最强的干扰方向, 但是经过乘以  $I_p^* \odot I_p$  这样一个矩阵的操作, 这个干扰方向的攻击性就会有所降低, 因为干扰中的一部分信息被丢掉了, 这样一来生成的对抗样本的有效性会大大降低。这就从理论上证明了不仅攻击者不能针对所提出的算法生成特定的对抗样本, 而且即使生成近似的对抗样本也不能保证攻击性。

### 5.3.3 基于流形结构的算法分类有效性分析

本节从流形学习的角度分析所提出算法的有效性, 如图 5-4 所示, 根据流形假设, 机器学习的目的是学习图中的数据流形。图中的  $x$  是原始的样本点,  $\tilde{x}$  是丢弃过特征的样本点。

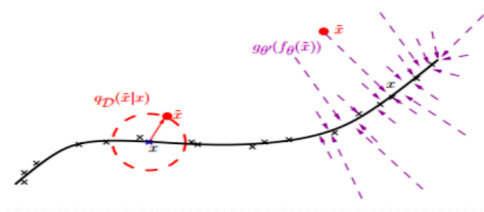


图 5-4 数据流形示意图

从图 5-4 可以看出, 所要学习的数据流形由原始样本  $x$  组成, 但是这不意味着这个流形只能从原始数据得到, 我们可以通过学习  $g_{\theta'} f_{\theta}(x)$  也就是图中的虚线, 得到从  $\tilde{x}$  到数据流形的映射。同时这样学习到的模型分类效果可能会更好, 因为它覆盖的样本空间更大, 对更多的测试样本有效。

### 5.3.4 不同算法的对比

本节中, 我们把提出的算法和现有的一些算法进行对比。在本章中我们提出了一种新的提高深度学习鲁棒性的算法。该算法通过在神经网络中引入随机性来提高其鲁棒性。但是, 正如前文所述, 还有其它的防御方法。这里我们分析为什么这些方法不能有效的抵挡对抗样本, 尤其是针对它们自身生成对抗样本。

基于一些近期的文献<sup>[110–112]</sup>，绝大多数的抵御算法都是通过扩充训练集的方法提高模型的鲁棒性，可以被表示如下：

$$\min_{\theta} \mathcal{G}(\theta, X, Y) = \mathcal{L}(\theta, X, Y) + \gamma \cdot \mathcal{R}(\theta, X, Y), \quad (5-8)$$

其中  $\mathcal{L}(\theta, X, Y)$  标准神经网络的目标函数， $\mathcal{R}(\theta, X, Y)$  是当前模型的对抗样本，也可以看作一个正则项，其中  $\gamma$  控制着对原始样本的更改程度。

基于这一类模型，攻击者还是可以通过如下的方法轻易计算目标函数对输入样本的偏导值：

$$\begin{aligned} \mathcal{J}_{\mathcal{G}}(X) &= \frac{\partial \mathcal{G}(\theta, X, Y)}{\partial X} \\ &= \frac{\partial (\mathcal{L}(\theta, X, Y) + \gamma \mathcal{R}(\theta, X, Y))}{\partial X}. \end{aligned} \quad (5-9)$$

在原始目标函数中引入一个正则项来惩罚最效的方向，虽然可以抵御一定的攻击，但是我们依然可以通过在原始样本上加一个干扰  $\phi \cdot \text{sign}(\mathcal{J}_{\mathcal{G}}(X))$  来攻击对应的模型。同时，增加  $\phi$  可以增强对抗样本的攻击性，这是因为一个特定的正则项对模型的惩罚力度有限。一个近期的研究中<sup>[104]</sup> 解释到，这种对抗性训练无法从根本上解决问题，因为深度学习模型的输入空间是无界的，对抗性训练无法涵盖所有的盲区。我们的 **random feature nullification** 方法可以被看作是从根本上阻挡了对抗样本的生成，可以理解为，我们把模型的盲区藏了起来，虽然盲区依然存在，但是攻击者是很难准确的找到模型对应的盲区。

## 5.4 算法有效性验证

在把算法应用到恶意软件分类之前，我们先在标准数据集上验证我们算法的有效性。我们把本章提到的算法和最流行的对抗学习算法：对抗性训练和 **dropout** 方法进行对比。同时我们还把本章方法和对抗性训练结合，组成一个新的主动学习深度神经网络方法，并且验证了算法的有效性。

### 5.4.1 实验设置和初始化

对于上面提到的方法，本文准备从以下两个标准衡量算法的有效性：模型的抵抗性和分类精度。如果在相同的数据集上一个算法表现了更好的鲁棒性，同时还保证了一定的分类精度，我们就认为这种算法优于其它方法。

为了验证算法的有效性，我们在公开数据集 **MNIST** 和 **CIFAR-10** 上进行实验，**MNIST** 有 60000 个训练样本和 10000 个测试样本。为了测试分类精度，我们使用 10000

测试样本测试不同算法训练得到的模型。对于模型抵抗性，我们要针对不同模型生成不同的对抗样本，但是所有的对抗样本都是基于原始的 10000 个原始测试样本生成的。CIFR-10 数据集有 50000 个训练样本和 10000 个测试样本，对抗样本从 10000 个测试样本生成，所生成的对抗样本如图 5-5 所示。本章所有的实验都是用基于 Theano 库<sup>[113]</sup>，用 python 编写的。

我们假设攻击者可以得到模型的所有信息，这样就可以生成最有攻击性的对抗样本，所以本文给出的结果是每种方法对对抗样本抵抗性的下限值。

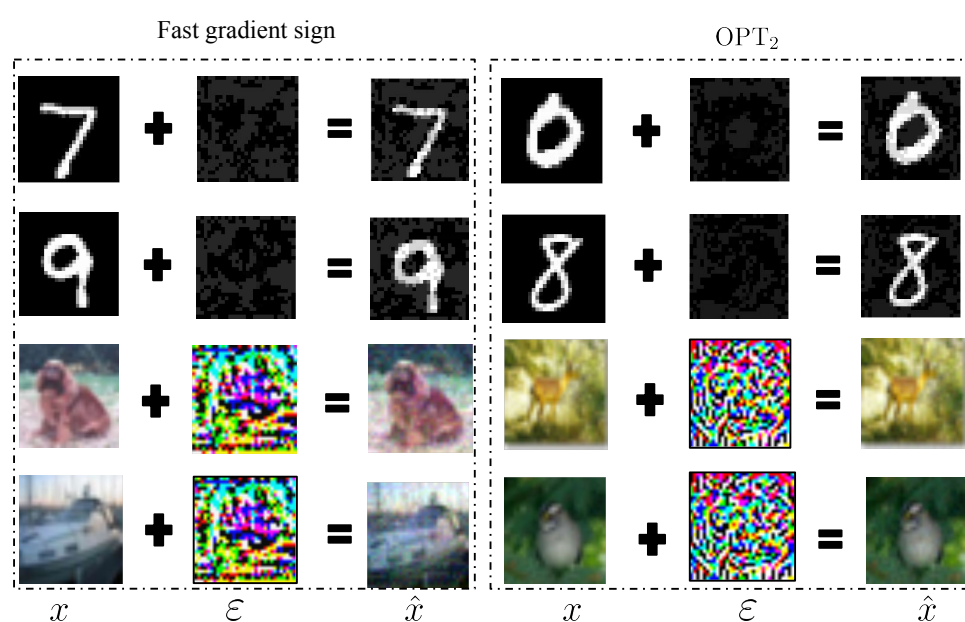


图 5-5 基于 random feature nullification 的深度学习模型

为了选择最好的特征丢弃比例，我们首先在两个数据集上对不同的比例进行了实验，这些比例从 10% 到 70%，每次增加 10%。同样，在训练和测试时要保持相同的比例。我们给出了这些设置下的分类精度和模型抵抗性。

从表 5-1 中，我们可以得到特征丢弃比例和模型分类性能以及抵抗性的关系：

- 随着丢弃比例的增加，模型分类性能有所下降；
- 模型的抵抗性随着丢弃比例的上升有很大程度的提高。

正如表 5-1 中显示的，当 nullification rate 为 50% 时，达到了最好的模型抵抗性。对于分类精度，我们的方法在不同的比例上达到了近似相同的误分率，因此，在后续的实验，我们选择 50% 作为我们的 feature nullification rate。

在给出实验之前，我们先给出本实验中不同神经网络的超参数，具体数值如表 5-2，5-3，5-4。

Expectation of nullification rates	MNIST		CIFAR-10	
	Accuracy (%)	Resistance $\phi = 0.15$ (%)	Accuracy (%)	Resistance $\phi = 0.15$ (%)
10%	98.17	70.39	80.01	55.87
20%	97.81	73.55	77.62	59.55
30%	97.61	78.31	75.95	61.63
40%	97.23	81.49	74.49	65.59
50%	96.27	83.68	74.02	67.85

表 5-1 不同特征丢弃比例训练出的模型效果对比

Learning Technologies	Hyper Parameters						
	DNN Structure	Activation	Optimizer	Learning Rate	Dropout Rate	Batch Size	Epoch
Standard DNN	784-500-300-100-10	relu	SGD	0.1	$\times$	100	25
Dropout	784-500-300-100-10	relu	SGD	0.1	0.5	100	25
Adv. Training	784-100-100-100-10	relu	SGD	0.01	0.5	100	70
RFN	784-100-100-100-10	relu	SGD	0.1	0.25	100	25
RFN & Adv. Training	784-100-100-100-10	relu	SGD	0.01	0.25	100	70

表 5-2 MNIST 模型的超参数：网络结构指的是该网络的隐含层个数和每一层的隐含节点个数，每一个模型的最后一层都是 softmax 分类器，标准 DNN 没有任何正则项。

## 5.4.2 实验结果

我们进行了五组不同的实验，每组实验所使用的算法都时之前提过的算法，实验结果在表 5-5 中给出。其中 dropout 方法的 dropout 比例是 25%。我们给出了每种方法训练出来的模型在不同  $\phi$  值下的分类精度和模型抵抗性。正如之前章节介绍的，系数  $\phi$  代表了对原始样本的修改程度。表 5-6 同时给出了一些在不同干扰程度下的对抗样本例子。值得注意的是在本章中模型的抵抗性就是指训练该模型的算法的鲁棒性。

当在原始样本上添加一个微小的干扰时，表 5-5 首先说明了标准神经网络模型在面对对抗样本时有很高的误分率。这说明了标准神经网络对对抗样本的抵抗能力较差。对与 Hinton's dropout 和 adversarial training，我们分别用相应的对抗样本测试模型，结果显示这两种方法可以在一定程度上提高模型的鲁棒性。原因是两种方法从不同角度增加了模型的非线性特性，减小了模型的盲区，增加了模型的鲁棒性。表 5-5 中的结果同时说明了 adversarial training 比 Hinton's dropout 在模型抵抗性上有更好的效果。

在表 5-5 中，我们同时发现 adversarial training 和 Hinton's dropout 都这是增加了生成对抗样本的难度。但是随着  $\phi$  的增加，对应的对抗样本相对于原始样本有更大的变化，同时这两种方法的鲁棒性也相应的下降了很多。基于之前的文献<sup>[104]</sup>，这是因为深度学习模型的输入空间是没有边界的，通过加正则项的方法不可能惩罚模型所有的盲区。

和之前的方法相比，我们的方法 random feature nullification 在模型的抵抗性实现了



Learning Technology	Hyper parameters					
	Activation	Optimizer	Learning rate	Dropout rate	Batch Size	Epoch
Standard DNN	Relu	Adam	0.001	$\times$	128	50
Dropout	Relu	Adam	0.001	0.5	128	50
Adv. Training	Relu	SGD	0.01	0.5	128	50
RFN	Relu	Adam	0.001	0.5	128	50
RFN & Adv. Training	Relu	SGD	0.01	0.5	128	50

表 5-3 CIFAR-10 模型的超参数，这里我们使用 CNN 替代 DNN。

Layer type	Learning Technology				
	Standard DNN	Dropout	Adv. Training	RFN	RFN & Adv. Training
Convolutional	64 filter( $3 \times 3$ )	64 filter( $3 \times 3$ )	64 filter( $3 \times 3$ )	64 filter( $3 \times 3$ )	64 filter( $3 \times 3$ )
Convolutional	64 filter( $3 \times 3$ )	64 filter( $3 \times 3$ )	64 filter( $3 \times 3$ )	64 filter( $3 \times 3$ )	64 filter( $3 \times 3$ )
Max pooling	$2 \times 2$	$2 \times 2$	$2 \times 2$	$2 \times 2$	$2 \times 2$
Convolutional	72 filter( $3 \times 3$ )	72 filter( $3 \times 3$ )	128 filter( $3 \times 3$ )	128 filter( $3 \times 3$ )	128 filter( $3 \times 3$ )
Convolutional	72 filter( $3 \times 3$ )	72 filter( $3 \times 3$ )	128 filter( $3 \times 3$ )	128 filter( $3 \times 3$ )	128 filter( $3 \times 3$ )
Max pooling	$2 \times 2$	$2 \times 2$	$2 \times 2$	$2 \times 2$	$2 \times 2$
Fully Connect	512 units	512 units	256 units	256 units	256 units
Fully Connect	256 units	256 units	256 units	256 units	256 units
Softmax	10 units	10 units	10 units	10 units	10 units

表 5-4 CIFAR-10 模型的网络结构。模型的超参数在表 5-3 中给出。

很大程度的提高。正如图 5-5 中所示，本章提出的方法在面对对抗样本时有较低的误分率。此外，我们还发现这个模型的抵抗性在干扰不断增加的情况下还是保持在了一个较好的水平，即使对原始样本的修改已经很大的情况下 (i.e.,  $\phi = 0.35$ )。一个解释是我们的算法在进行 **random feature nullification** 操作时，扔掉了对分类结果影响较大的特征点。

回顾之前的内容，我们的 **random feature nullification** 方法在原始模型中引入了一定的随机性，这可以被看作是一种数据预处理方法。同时，该方法不影响深度神经网络模型的结构。因此该方法可以和其它的对抗深度学习算法相结合。我们希望这样的结合可以带来更好的鲁棒性，为了验证这一点，我们把 **random feature nullification** 和 **adversarial training** 相结合。

表 5-5 中给出了这种结合之后算法的分类精度和模型抵抗性。和原来的 **adversarial training** 方法相比，我们的方法在面对对抗样本时有更低的误分率。同时，我们还发现这种结合之后的方法的抵抗性也超过了单一的 **random feature nullification** 方法。这是因为 **random feature nullification** 和 **adversarial training** 从两个不同的方面提高模型的鲁棒性，并且互不影响，因此结合在一起可以到达更好的性能。

最后需要指出的是，在表 5-5 中同时给出了不管是我们提出的 **random feature nullification** 方法还是和该方法和 **adversarial training** 结合的方法，比其它三种方法都体现出了更好的分类特性。这不仅是由于对模型加了正则项之后会提升模型的性能，还因为我们在训练和测试时保持了相同的样本丢弃比例，保证了训练样本和测试样本同分布。

Learning Technology	MNIST				CIFAR-10			
	Accuracy (%)	Resistance (%)			Accuracy (%)	Resistance (%)		
		$\phi = 0.15$	$\phi = 0.25$	$\phi = 0.35$		$\phi = 0.15$	$\phi = 0.25$	$\phi = 0.35$
Standard	97.93	8.19	0.56	0.01	73.59	19.48	13.51	10.68
Dropout	98.43	19.51	3.86	0.96	81.07	17.43	16.59	16.40
Adv Training	96.46	67.68	28.37	7.62	80.62	33.97	19.76	13.73
RFN	96.27	83.69	71.44	60.69	74.02	67.85	51.89	41.29
Adv Training & RFN	96.11	91.28	84.92	78.18	74.12	71.03	55.49	49.84

表 5-5 不同算法在两个数据集上的实验结果。
















Learning Technology	$\phi = 0.15$	$\phi = 0.25$	$\phi = 0.35$
Standard			
Hinton			
Adv Training			
RFN			
Adv Training & RFN			

表 5-6 不同  $\phi$  下的对抗样本示例。

## 5.5 基于 RFN 的对抗深度学习在恶意软件分类中的应用

### 5.5.1 恶意软件分类

#### 5.5.1.1 恶意软件分类数据集

本节中，我们把所提出的算法应用到计算机安全领域的一个应用中，即恶意软件分类。这是一个对算法鲁棒性要求非常高的应用，因为一旦发生误分，特别是恶意软件的误分，就会造成十分严重的后果。之前的分类方法基本是基于模型和专家知识的，近些年来，随着机器学习方法的发展，越来越多的研究人员或者公司开始采用机器学习的方法进行检测和分类，特别是深度学习方法，以其强大的能力也受到了恶意软件分类和检测领域的重视。但是深度学习又存在对抗样本的问题，这就限制了它在安全关键领域的应用。

本节就是在这样的背景下，选择了一个安全领域的核心问题。通过在这类问题上测试本文提出的方法，来进一步验证算法的有效性和实用性。本文使用的数据集是 window 系统审计日志的集合，所谓 window 系统审计日志就是系统中操作的收集和记录，每个

日志对应一个软件。它们是通过标准的，嵌入系统中的工具收集的。数据的来源有两类：一个企业网络的用户和运行了特定良性和恶意软件的虚拟沙盒。

这个恶意软件分类数据集中每个样本的原始维度非常的高，无法直接用于构建神经网络，所以在把数据输入到神经网络之前，我们先根据文献<sup>[114]</sup>中的特征选择方法，把数据集的原始特征降到 10000 维。其中每一个特征的含义是日志中的每一个事件序列是否发生，时间序列的含义是一组按照一定时间先后关系发生的不同事件的组合，在一个时间序列中，最多有三个事件。因此每个特征的数值非 0 即 1，0 表示在特定日志中，这个特征代表的事件序列不发生，1 则有相反的含义。本数据集是二分类问题，数据集中的样本要么是良性软件，要么是恶意软件。我们把整个数据集分为有 26078 个样本的训练集和含有 6000 个样本的测试集，在训练集中有 14399 个良性样本，有 11679 个恶意样本，在测试集中有 3000 个良性样本和 3000 个恶意样本。我们的任务是准确的区分出测试集中的良性样本和恶意样本，并测试训练的模型对对抗样本的抵抗性。

### 5.5.1.2 基于恶意软件生成对抗样本

在进行实验之前，我们先介绍如何在这个数据集上生成对抗样本。由于本数据集特征的数值是离散的，所以我们在生成对抗样本时对梯度加上  $l_0$  范数的正则项。比如，在文献<sup>[115]</sup>中，作者用  $l_0$  来衡量对抗样本相对于原始样本的变化程度，这个研究所使用的数据集是 Android 系统上的恶意软件数据集。此外，正如<sup>[115]</sup>中讨论的那样，恶意软件分类数据集相对于图像数据有更强的限制，即特征间的相关系数更大，相互之间的约束更强。在我们的数据集中，特征的含义是对应的软件是不是读，写或者删除了系统的文件。同时，恶意软件的目的是在对系统造成破坏的同时，不被系统的防火墙发现。因此，大规模的修改样本中的特征是不现实的，因为如果修改太多的特征，会导致修改后的样本被系统防火墙检测到，甚至会使恶意软件的功能失效。

为了避免这种情况，我们把每个恶意软件样本可以修改的特征数限制在一个比较小的范围。在本章的实验中，我们把可以修改的特征数设置为 10。同时，值得注意的是，我们生成对抗样本的目的是骗过一个深度神经网络模型，所以我们不需要把良性样本修改成恶意样本，因此，在本实验中我们只对测试集中的恶意样本进行修改。在表 5-7 中，我们给出了一些在生成对抗样本过程中被修改的特征的例子。

从表 5-7 可以看出，这些操作只是增加了原来的恶意软件对某种动态链接库的调用，这说明我们选择的生成对抗样本的方法是合理的。表 5-8 给出了本实验神经网络的超参数。

## Examples of Changed Features

WINDOWS\_FILE:Execute:[system]\slc.dll,  
 WINDOWS\_FILE:Execute:[system]\cryptsp.dll  
 WINDOWS\_FILE:Execute:[system]\wersvc.dll,  
 WINDOWS\_FILE:Execute:[system]\faultrep.dll  
 WINDOWS\_FILE:Execute:[system]\imm32.dll,  
 WINDOWS\_FILE:Execute:[system]\wer.dll  
 WINDOWS\_FILE:Execute:[system]\ntmarta.dll,  
 WINDOWS\_FILE:Execute:[system]\apphelp.dll  
 WINDOWS\_FILE:Execute:[fonts]\times.ttf,  
 WINDOWS\_FILE:Execute:[fonts]\times.ttf  
 WINDOWS\_FILE:Execute:[system]\faultrep.dll,  
 WINDOWS\_FILE:Execute:[system]\imm32.dll

表 5-7 针对恶意软件样本生成对抗样本时篡改的样本示例

Learning Technology	Hyper Parameters					
	DNN Structure	Activation	Optimizer	Learning Rate	Dropout Rate	Batch Size
Standard DNN	5000-1000-100-2	relu	Adam	0.001	×	500
Dropout	5000-1000-100-2	relu	Adam	0.001	0.5	500
Adv. Training	5000-1000-100-2	relu	SGD	0.01	0.5	500
RFN	5000-1000-100-2	relu	Adam	0.001	0.5	500
RFN & Adv. Training	5000-1000-100-2	relu	SGD	0.01	0.5	500

表 5-8 恶意软件模型的超参数。

## 5.5.2 基于 RFN 的对抗深度学习在恶意软件分类的效果

在介绍完了本节所使用的数据集，需要解决的问题以及如何基于本数据集生成对抗样本之后，我们用不同的方法完成这个恶意软件分类任务。由于在上一节中已经验证了本章提出的方法比 Hinton' dropout 在鲁棒性上有更好的表现，而且 Hinton' dropout 并不是为了抵抗对抗样本而设计的，所以在这一节我们不把这种方法作为对比方法中的一种。因此，本节我们还是使用标准的神经网络，adversarial training，random feature nullification 以及 random feature nullification 和 adversarial training 相结合这几种方法来完成这项分类任务。

本组实验的结果在表 5-9 中给出。从表中可以看出所有的方法都达到了高于 92% 的分类精度，这些结果都高于或者和文献<sup>[114]</sup>中使用 logistics regression 的效果，这说明了深度神经网络在处理大数据量，高维度数据时的优越性。这是因为神经网络不需要进

Learning Technology	Classification rate (%)	
	Accuracy	Resistance
Standard	93.99	30.00
Adversarial training	92.68	26.07
Random feature nullification	93.08	62.30
Random feature nullification & adversarial training	93.67	67.96

表 5-9 不同算法在恶意软件分类的效果

行特征提取，同时多层隐含层的结构也保证神经网络可以更深层的挖掘特征信息，达到更好的效果。

同时，值得注意的是，使用了 random feature nullification 技术的方法可以达到比标准神经网络更好的分类效果，这说明了在这个数据集上，random feature nullification 比 adversarial training 有更好的正则效果。

在模型的抵抗性，即算法的鲁棒性方面，adversarial training 对模型的抵抗性没有提升，这和之前的结果并不矛盾。实验结果显示，adversarial training 对基于标准深度神经网络训练出的模型生成的对抗样本有很好的分类性能，分类精度可以达到 92.78%。但是，本章的之前章节提到了，在本章的实验中，我们测试的是模型对攻击性最强的对抗样本的抵抗性，所以当面对针对这个模型生成的对抗样本时，adversarial training 的效果就变得很差，甚至比标准神经网络的结果还差，这也说明了 adversarial training 并不保证对所有的对抗样本都有效，它对模型抵抗性的提高是有限的。

相反的是，random feature nullification 和 Random feature nullification & adversarial training 都大幅度提高了模型对对抗样本的抵抗性。这种提高的贡献来自于 random feature nullification 的优势，因为和 adversarial training 结合的方法还没有单独的 RFN 模型抵抗性强。这是因为基于 RFN 的模型中引入了随机变量，阻断了残差的传递，使攻击者无法计算模型目标函数对输入的偏导数，从而无法得到针对模型自身的对抗样本，只能近似的生成一些对抗样本。这些对抗样本的攻击性本身就很差。而且，在测试时，我们的方法会随机的丢到一些特征，这也会减轻对抗样本的攻击性，因为被攻击者修改过的特征点很可能在模型的第一层被丢掉了。

## 5.6 本章小结

在这一章中为，我们提出了一种新的对抗深度学习算法。这种算法可以被用于构建一个对对抗样本更鲁棒的神经网络。我们的设计是基于对深度学习本质缺陷和现有方法的全面分析的基础上提出的。使用本章提出的方法，我们证明了基于我们的方法是无法生成有攻击性的对抗样本，这说明了我们提出的方法可以抵御针对模型自身的攻击。

作为对比，我们说明了现有的对抗性训练方法不足以抵御对抗样本。通过在标准数据集上的实验，我们说明了所提出的算法大幅降低了深度学习模型在面对对抗样本时的误分率，而且，我们的方法不会降低深度学习模型对正常测试样本的分类性能。之后，我们把所提出的方法应用到了计算机安全中的一个关键领域：恶意软件分类中，结果表明了我们算法的优越性。

本章提出的算法可以和对抗性训练算法相结合，形成一种主动学习深度神经网络方法，实验结果也表明了结合后的算法在面对正常样本和对抗样本时都达到了最好的效果。

## 第六章 总结与展望

### 6.1 研究工作总结

本文的研究工作集中在通过挖掘数据集中的有用信息提高不同主动学习算法的性能。

本文的第一个研究对象主要是主动学习支持向量机。传统的主动学习支持向量机方法只考虑样本对当前模型的价值，并没有挖掘数据集整体的分布特征，特别是在很多数据集中都存在的数据多流形结构，这就造成了信息浪费。本文提出了三种改进的主动学习方法，这些方法通过适当的算法挖掘数据包含的信息对主动学习支持向量机进行改进。本文的三种方法分别从不同角度提高主动学习支持向量机的性能。第二章通过挖掘无标签样本的信息，提升主动学习支持向量机的分类性能。第三章的方法是优化初始支持向量的选择，从而降低主动学习算法对初始状态和噪声的敏感性。第四章提出的算法旨在通过在每次迭代中利用标签样本的信息，挖掘数据集的数据分布特性，提高主动学习支持向量机的分类精度和收敛速度。

本文的另一研究对象是主动学习在深度学习中的实现：对抗深度学习。对抗深度学习的基本算法是对抗性训练，对抗性训练通过在训练过程中主动选择当前模型下的对抗样本，并且把这些对抗样本作为训练集训练模型，从而提升模型对对抗样本的抵抗性。本文第五章基于现有的对抗深度学习方法的不足和数据流形特性，提出了全新的对抗深度学习方法。在验证了算法的有效性之后，我们把所提出的算法应用到恶意软件分类中，不仅提高了深度学习模型对对抗样本的抵抗性，同时提高了模型的分类性能。

### 6.2 未来研究工作

#### 6.2.1 模型超参数选择

本文中主要涉及的三种算法：主动学习支持向量机，流形学习和深度学习算法都面临一个相同的问题就是超参数的选择。特别是深度学习，随着模型的隐含层层数的增加，需要确定的超参数也相应的增加，而且不同的超参数对模型的性能有很大的影响。因此，如何选择合适的超参数在模型训练中十分重要。

机器学习算法可以看成将数据集映射到函数的一个泛函。通常，函数由一系列参数确定。学习算法就是通过特定的优化方法寻找到合适的参数。但是算法包含一些特殊的参数，在优化开始之前，这些参数会被确定。这样的参数就被称作算法的“超参数”。超

参数是不能通过优化算法求出来的，需要人为给定。现有的确定超参数的方法主要是基于经验，在一个范围内进行网格搜索，或者通过交叉验证的方法确定超参数。

但是当算法中超参数很多时，这种方法的效率不仅会大大降低，同时由于超参数的模型的作用往往是耦合在一起的，所以对超参数的选择带来了很大的麻烦。目前，不管是学术界还是工业界都没有一个很好的超参数选择方法，更别说是参数选择标准。现在效果比较好的方法，除了上面提到的之外还有基于贝叶斯的参数选择方法。不过这些方法都有各自的局限性，所以，研究出一种对更好的超参数选择方法，具有十分重要的实际意义。

### 6.2.2 流形学习算法时间复杂度

流形学习算法是基于流形假设的一类算法，这类算法往往旨在挖掘数据分布的拓扑结构。这类算法对挖掘数据的特征有十分重要的作用，已经被应用到了很多领域中。但是这类算法在在线学习这个领域的应用并不是很多，主要的原因是算法的时间复杂度比较高。

流形学习方法往往涉及到复杂的矩阵运算，特别是奇异值分解等算法复杂度很高的运算，因此使这类算法的时间复杂度也相对较高。本文第三，第四章中的算法就受到奇异值分解的限制，无法在更快的时间内完成。文中也对现有的计算方法的复杂度进行了介绍，即使是时间复杂度最低的奇异值分解方法的复杂度也比较高。因此，如何提高流形学习算法的时间效率变的十分重要。当然，在试图降低流形学习时间复杂度之前，要先研究如何加快奇异值分解等基本矩阵运算速度，这些工作的重要性和意义不仅仅是对流形学习而言，对整个机器学习领域都起着关键的作用。



## 参考文献

- [1] 谭松波. 高性能文本分类算法研究[D]. 中国科学院研究生院 (计算技术研究所), 2006.
- [2] CARBONELL J G, MICHALSKI R S, MITCHELL T M. An overview of machine learning[G]//Machine learning. [S.l.]: Springer, 1983: 3–23.
- [3] KOTSIANTIS S B, ZAHARAKIS I, PINTELAS P. Supervised machine learning: A review of classification techniques. 2007.
- [4] ZHU X. Semi-supervised learning literature survey[J]. 2005.
- [5] ZHU X, GOLDBERG A B. Introduction to semi-supervised learning[J]. Synthesis lectures on artificial intelligence and machine learning, 2009, 3(1): 1–130.
- [6] CHAPELLE O, SCHOLKOPF B, ZIEN A. Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006)[Book reviews][J]. IEEE Transactions on Neural Networks, 2009, 20(3): 542–542.
- [7] KROGH A, VEDELSBY J, et al. Neural network ensembles, cross validation, and active learning[J]. Advances in neural information processing systems, 1995, 7: 231–238.
- [8] PRINCE M. Does active learning work? A review of the research[J]. Journal of engineering education, 2004, 93(3): 223–231.
- [9] TONG S, KOLLER D. Support vector machine active learning with applications to text classification[J]. Journal of machine learning research, 2001, 2(Nov): 45–66.
- [10] SUYKENS J A, VANDEWALLE J. Least squares support vector machine classifiers[J]. Neural processing letters, 1999, 9(3): 293–300.
- [11] 刘忠伟, 章毓晋. 十种基于颜色特征图像检索算法的比较和分析[J]. 信号处理, 2000, 16(1): 79–84.
- [12] 王涛, 胡事民, 孙家广. 基于颜色-空间特征的图像检索[J]. 软件学报, 2002, 13(10).
- [13] SYED N A, HUAN S, KAH L, et al. Incremental learning with support vector machines[J]. 1999.

- [14] HOI S C, JIN R, ZHU J, et al. Batch mode active learning and its application to medical image classification[C]//Proceedings of the 23rd international conference on Machine learning. ACM. [S.l.]: [s.n.], 2006: 417–424.
- [15] LEWIS D, GALE W. Training text classifiers by uncertainty sampling[J]. 1994.
- [16] JUSZCZAK P, DUIN R P. Uncertainty sampling methods for one-class classifiers[C]//Proceedings of the ICML. Vol. 3. [S.l.]: [s.n.], 2003.
- [17] DIETTERICH T G. Ensemble methods in machine learning[C]//International workshop on multiple classifier systems. Springer. [S.l.]: [s.n.], 2000: 1–15.
- [18] ROY N, MCCALLUM A. Toward optimal active learning through monte carlo estimation of error reduction[J]. ICML, Williamstown, 2001: 441–448.
- [19] FU Y, ZHU X, LI B. A survey on instance selection for active learning[J]. Knowledge and information systems, 2013, 35(2): 249–283.
- [20] EFRON B. Bootstrap methods: another look at the jackknife[G]//Breakthroughs in Statistics. [S.l.]: Springer, 1992: 569–593.
- [21] GUO Y, GREINER R. Optimistic Active-Learning Using Mutual Information.[C]//IJCAI. Vol. 7. [S.l.]: [s.n.], 2007: 823–829.
- [22] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553): 436–444.
- [23] SUN Y, CHEN Y, WANG X, et al. Deep learning face representation by joint identification-verification[C]//Advances in Neural Information Processing Systems. [S.l.]: [s.n.], 2014: 1988–1996.
- [24] VINCENT P, LAROCHELLE H, LAJOIE I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion[J]. Journal of Machine Learning Research, 2010, 11(Dec): 3371–3408.
- [25] GLOROT X, BORDES A, BENGIO Y. Domain adaptation for large-scale sentiment classification: A deep learning approach[C]//Proceedings of the 28th International Conference on Machine Learning (ICML-11). [S.l.]: [s.n.], 2011: 513–520.
- [26] WANG Q, GUO W, ZHANG K, et al. Learning Adversary-Resistant Deep Neural Networks[J]. ArXiv preprint arXiv:1612.01401, 2016.

- [27] SHIN E C R, SONG D, MOAZZEZI R. Recognizing functions in binaries with neural networks[C]//24th USENIX Security Symposium (USENIX Security 15). [S.l.]: [s.n.], 2015: 611–626.
- [28] ZEILER M D, FERGUS R. Visualizing and understanding convolutional networks[C]//European Conference on Computer Vision. Springer. [S.l.]: [s.n.], 2014: 818–833.
- [29] WILLIAMS D, HINTON G. Learning representations by back-propagating errors[J]. Nature, 1986, 323: 533–536.
- [30] DENG J, DONG W, SOCHER R, et al. Imagenet: A large-scale hierarchical image database[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE. [S.l.]: [s.n.], 2009: 248–255.
- [31] HINTON G, DENG L, YU D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. IEEE Signal Processing Magazine, 2012, 29(6): 82–97.
- [32] CHO K, VAN MERRIËNBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. ArXiv preprint arXiv:1406.1078, 2014.
- [33] RABINER L, JUANG B. An introduction to hidden Markov models[J]. Ieee assp magazine, 1986, 3(1): 4–16.
- [34] FINKEL J R, KLEEMAN A, MANNING C D. Efficient, Feature-based, Conditional Random Field Parsing.[C]//ACL. Vol. 46. [S.l.]: [s.n.], 2008: 959–967.
- [35] GRAVES A, WAYNE G, DANIHELKA I. Neural turing machines[J]. ArXiv preprint arXiv:1410.5401, 2014.
- [36] LAWRENCE S, GILES C L, TSOI A C, et al. Face recognition: A convolutional neural-network approach[J]. IEEE transactions on neural networks, 1997, 8(1): 98–113.
- [37] MIKOLOV T, KARAFIÁT M, BURGET L, et al. Recurrent neural network based language model.[C]//Interspeech. Vol. 2. [S.l.]: [s.n.], 2010: 3.
- [38] GOODFELLOW I, BENGIO Y, COURVILLE A. Deep Learning[M]. [S.l.]: MIT Press, 2016.
- [39] JIN W, LI Z J, WEI L S, et al. The improvements of BP neural network learning algorithm[C]//Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on. Vol. 3. IEEE. [S.l.]: [s.n.], 2000: 1647–1649.

- [40] KINGMA D, BA J. Adam: A method for stochastic optimization[J]. ArXiv preprint arXiv:1412.6980, 2014.
- [41] BENGIO Y, CA M. RMSProp and equilibrated adaptive learning rates for non-convex optimization[J].
- [42] JOHANNESSON R, ZIGANGIROV K S. Fundamentals of convolutional coding[M]. Vol. 15. [S.l.]: John Wiley & Sons, 2015.
- [43] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. [S.l.]: [s.n.], 2012: 1097–1105.
- [44] HADDADNIA J, FAEZ K, MOALLEM P. Neural network based face recognition with moment invariants[C]//Image Processing, 2001. Proceedings. 2001 International Conference on. Vol. 1. IEEE. [S.l.]: [s.n.], 2001: 1018–1021.
- [45] CHEN L.-C, PAPANDREOU G, KOKKINOS I, et al. Semantic image segmentation with deep convolutional nets and fully connected crfs[J]. ArXiv preprint arXiv:1412.7062, 2014.
- [46] MA Y, FU Y. Manifold learning theory and applications[M]. [S.l.]: CRC press, 2011.
- [47] ELHAMIFAR E, VIDAL R. Sparse subspace clustering: Algorithm, theory, and applications[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35(11): 2765–2781.
- [48] LIU G, LIN Z, YAN S, et al. Robust recovery of subspace structures by low-rank representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(1): 171–184.
- [49] SCHOHN G, COHN D. Less is more: Active learning with support vector machines[C]//ICML. Citeseer. [S.l.]: [s.n.], 2000: 839–846.
- [50] TONG S. Active learning: Theory and Applications. 2001.
- [51] SMOLA A J, SCHÖLKOPF B. A tutorial on support vector regression. 2004.
- [52] BURGESS C J. A tutorial on support vector machines for pattern recognition[J]. Data mining and knowledge discovery, 1998, 2(2): 121–167.
- [53] WOLFE P. A duality theorem for non-linear programming[J]. Quarterly of applied mathematics, 1961: 239–244.

- [54] BAUDAT G, ANOUAR F. Kernel-based methods and function approximation[C]//Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on. Vol. 2. IEEE. [S.l.]: [s.n.], 2001: 1244–1249.
- [55] SHAO H, TONG B, SUZUKI E. Query by committee in a heterogeneous environment[C]//International Conference on Advanced Data Mining and Applications. Springer. [S.l.]: [s.n.], 2012: 186–198.
- [56] SETTLES B, CRAVEN M, RAY S. Multiple-instance active learning[C]//Advances in neural information processing systems. [S.l.]: [s.n.], 2008: 1289–1296.
- [57] LEWIS D D, GALE W A. A sequential algorithm for training text classifiers[C]//Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. Springer-Verlag New York, Inc. [S.l.]: [s.n.], 1994: 3–12.
- [58] JANSSEN H. Monte-Carlo based uncertainty analysis: Sampling efficiency and sampling convergence[J]. Reliability Engineering & System Safety, 2013, 109: 123–132.
- [59] PRUDENCIO R B, SOARES C, LUDERMIR T B. Uncertainty sampling methods for selecting datasets in active meta-learning[C]//Neural Networks (IJCNN), The 2011 International Joint Conference on. IEEE. [S.l.]: [s.n.], 2011: 1082–1089.
- [60] GUO W, ZHONG C, YANG Y. Spectral Clustering based Active Learning with Applications to Text Classification[C]//MATEC Web of Conferences. Vol. 56. EDP Sciences. [S.l.]: [s.n.], 2016.
- [61] ZHOU D, BOUSQUET O, LAL T N, et al. Learning with local and global consistency[C]//Advances in Neural Information Processing Systems 16. [S.l.]: MIT Press, 2004: 321–328.
- [62] CHANG C.-C, LIN C.-J. LIBSVM: a library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2(3): 27.
- [63] WANG J, JEBARA T, CHANG S.-F. Semi-supervised learning using greedy max-cut[J]. Journal of Machine Learning Research, 2013, 14(Mar): 771–800.
- [64] WANG Y, JIANG Y, WU Y, et al. Spectral clustering on multiple manifolds[J]. IEEE Transactions on Neural Networks, 2011, 22(7): 1149–1161.
- [65] SHLENS J. A tutorial on principal component analysis[J]. ArXiv preprint arXiv:1404.1100, 2014.

- [66] MA Y, DERKSEN H, HONG W, et al. Segmentation of multivariate mixed data via lossy data coding and compression[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007(9): 1546–1562.
- [67] LIU G, YAN S. Latent low-rank representation for subspace segmentation and feature extraction[C]//2011 International Conference on Computer Vision. IEEE. [S.l.]: [s.n.], 2011: 1615–1622.
- [68] PARSONS L, HAQUE E, LIU H. Subspace clustering for high dimensional data: a review[J]. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 90–105.
- [69] LIN T, ZHA H. Riemannian manifold learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(5): 796–809.
- [70] FAZEL M. Matrix rank minimization with applications. PhD thesis. PhD thesis, Stanford University, 2002.
- [71] 濮定国, 金中. 新的拉格朗日乘子方法[J]. 同濟大學學報 (自然科學版), 2010, 38(9): 1387–1391.
- [72] LIN Z, CHEN M, MA Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices[J]. ArXiv preprint arXiv:1009.5055, 2010.
- [73] GUAN-ZHONG C X.-Y D, LI-BIN Y. Survey on Spectral Clustering Algorithms [J][J]. Computer Science, 2008, 7(005).
- [74] GUO Y, SCHUURMANS D. Discriminative batch mode active learning[C]//Advances in neural information processing systems. [S.l.]: [s.n.], 2008: 593–600.
- [75] 胡家赣. 双参数并行 Jacobi 型方法及其收敛性[J]. 计算数学, 1992, 14(1): 70–78.
- [76] HSU C.-W, LIN C.-J. A comparison of methods for multiclass support vector machines[J]. IEEE transactions on Neural Networks, 2002, 13(2): 415–425.
- [77] JORDAN M, MITCHELL T. Machine learning: Trends, perspectives, and prospects[J]. Science, 2015, 349(6245): 255–260.
- [78] HASTIE T, TIBSHIRANI R, FRIEDMAN J. Unsupervised learning[G]//The elements of statistical learning. [S.l.]: Springer, 2009: 485–585.
- [79] MCCALLUMZY A K, NIGAMY K. Employing EM and pool-based active learning for text classification[C]//Proc. International Conference on Machine Learning (ICML). Citeseer. [S.l.]: [s.n.], 1998: 359–367.

- [80] DASGUPTA S, HSU D. Hierarchical sampling for active learning[C]//Proceedings of the 25th international conference on Machine learning. ACM. [S.l.]: [s.n.], 2008: 208–215.
- [81] QIU Q, SAPIRO G. Learning transformations for clustering and classification[J]. Journal of Machine Learning Research, 2015, 16: 187–225.
- [82] RECHT B, FAZEL M, PARRILO P A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization[J]. SIAM review, 2010, 52(3): 471–501.
- [83] CANDÈS E J, LI X, MA Y, et al. Robust principal component analysis?[J]. Journal of the ACM (JACM), 2011, 58(3): 11.
- [84] SRIPERUMBUDUR B K, LANCKRIET G R. A proof of convergence of the concave-convex procedure using zangwill's theory[J]. Neural computation, 2012, 24(6): 1391–1407.
- [85] YUILLE A L, RANGARAJAN A. The concave-convex procedure[J]. Neural computation, 2003, 15(4): 915–936.
- [86] WATSON G A. Characterization of the subdifferential of some matrix norms[J]. Linear algebra and its applications, 1992, 170: 33–45.
- [87] BORDES A, ERTEKIN S, WESTON J, et al. Fast kernel classifiers with online and active learning[J]. Journal of Machine Learning Research, 2005, 6(Sep): 1579–1619.
- [88] CAO L J, KEERTHI S S, ONG C J, et al. Parallel sequential minimal optimization for the training of support vector machines[J]. IEEE Transactions on Neural Networks, 2006, 17(4): 1039–1049.
- [89] ZHANG Z, WANG J, ZHA H. Adaptive manifold learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(2): 253–265.
- [90] FU C, GONG L, YANG Y. An improved active learning method based on feature selection[J]. 2015.
- [91] SANDERSON C. Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments[J]. 2010.
- [92] CURTIN R R, CLINE J R, SLAGLE N P, et al. MLPACK: A scalable C++ machine learning library[J]. Journal of Machine Learning Research, 2013, 14(Mar): 801–805.

- [93] LEE C.-P, LIN C.-J. A study on L2-loss (squared hinge-loss) multiclass SVM[J]. *Neural computation*, 2013, 25(5): 1302–1323.
- [94] PLATT J, et al. Sequential minimal optimization: A fast algorithm for training support vector machines[J]. 1998.
- [95] ASUNCION A, NEWMAN D. UCI machine learning repository. 2007.
- [96] FU C.-J, YANG Y.-P. A batch-mode active learning SVM method based on semi-supervised clustering[J]. *Intelligent Data Analysis*, 2015, 19(2): 345–358.
- [97] STIKIC M, VAN LAERHOVEN K, SCHIELE B. Exploring semi-supervised and active learning for activity recognition[C]//2008 12th IEEE International Symposium on Wearable Computers. IEEE. [S.l.]: [s.n.], 2008: 81–88.
- [98] LOREGGIA A, MALITSKY Y, SAMULOWITZ H, et al. Deep Learning for Algorithm Portfolios[C]//AAAI Conference on Artificial Intelligence. [S.l.]: [s.n.], 2015.
- [99] YANN M L.-J, TANG Y. Learning Deep Convolutional Neural Networks for X-Ray Protein Crystallization Image Analysis[C]//AAAI Conference on Artificial Intelligence. [S.l.]: [s.n.], 2016.
- [100] BAR Y, DIAMANT I, WOLF L, et al. Deep learning with non-medical training used for chest pathology identification[C]//SPIE Medical Imaging. International Society for Optics, Photonics. [S.l.]: [s.n.], 2015: 94140V–94140V.
- [101] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. *Nature*, 2016, 529(7587): 484–489.
- [102] FARABET C, COUPRIE C, NAJMAN L, et al. Scene parsing with multiscale feature learning, purity trees, and optimal covers[J]. *ArXiv preprint arXiv:1202.2160*, 2012.
- [103] SZEGEDY C, ZAREMBA W, SUTSKEVER I, et al. Intriguing properties of neural networks[C]//International Conference on Learning Representations. [S.l.]: [s.n.], 2014.
- [104] GOODFELLOW I J, SHLENS J, SZEGEDY C. Explaining and harnessing adversarial examples[J]. *ArXiv preprint arXiv:1412.6572*, 2014.
- [105] GU S, RIGAZIO L. Towards deep neural network architectures robust to adversarial examples[J]. *ArXiv preprint arXiv:1412.5068*, 2014.
- [106] HUANG R, XU B, SCHUURMANS D, et al. Learning with a strong adversary[J]. *CoRR*, abs/1511.03034, 2015.



- [107] MIYATO T, MAEDA S.-I, KOYAMA M, et al. Distributional smoothing with virtual adversarial training[J]. Stat, 2015, 1050: 25.
- [108] NOKLAND A. Improving Back-Propagation by Adding an Adversarial Gradient[J]. ArXiv:1510.04189 [cs], 2015arXiv: 1510.04189.
- [109] RIFAI S, VINCENT P, MULLER X, et al. Contractive auto-encoders: Explicit invariance during feature extraction[C]//Proceedings of the 28th international conference on machine learning (ICML-11). [S.l.]: [s.n.], 2011: 833–840.
- [110] ORORBIA II A G, GILES C L, KIFER D. Unifying Adversarial Training Algorithms with Flexible Deep Data Gradient Regularization[J]. ArXiv:1601.07213 [cs], 2016arXiv: 1601.07213.
- [111] KANG G, LI J, TAO D. Shakeout: A New Regularized Deep Neural Network Training Scheme[C]//AAAI Conference on Artificial Intelligence. [S.l.]: [s.n.], 2016.
- [112] WAGER S, WANG S, LIANG P S. Dropout Training as Adaptive Regularization[G]//Advances in Neural Information Processing Systems 26. Ed. by BURGESS C J C, BOTTOU L, WELLING M, et al. [S.l.]: Curran Associates, Inc., 2013: 351–359.
- [113] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions[J/OL]. ArXiv e-prints, 2016, abs/1605.02688. <http://arxiv.org/abs/1605.02688>.
- [114] BERLIN K, SLATER D, SAXE J. Malicious behavior detection using windows audit logs[C]//Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security. ACM. [S.l.]: [s.n.], 2015: 35–44.
- [115] GROSSE K, PAPERNOT N, MANOHARAN P, et al. Adversarial Perturbations Against Deep Neural Networks for Malware Classification[J]. ArXiv preprint arXiv:1606.04435, 2016.



## 致 谢

回首研究生的这段奋斗岁月，我的学习和科研能力以及其他方面的综合素质得到很大的提高。在这期间，许多人给予我指导和帮助，让我受益匪浅，特别是能在病中完成学位论文更离不开导师，同学对我的指导和照顾。

首先，我要感谢尊敬的导师杨煜普教授。在我研究生学习期间，他在科研工作上给予我深入的指导，指点我找到自己的科研方向，传授我科研方法。杨老师高度概括的逻辑思维方式和务实的作风使我受用终生。同时，杨老师还给予我无限的支持和鼓励，在我对未来的职业规划非常迷茫的困难时期鼓励支持我坚持自己的科研梦想，在我申请博士期间也给予了很大的支持和帮助。在此，我要表达我无限的敬意和真心的感谢。

其次，我要感谢宾夕法尼亚州立大学的 Xinyu Xing 教授，C. Lee Giles 教授和 Lynn Lin 教授。在我访问宾州州立大学期间对我学术上的指导。当我遇到科研问题，自己无法解决的时候，他们总能为我提供一些行之有效的方案和建议。还要感谢他们对我申请博士的帮助，特别感谢 Xinyu Xing 教授对我生活的帮助，让我第一次在美国长时间的生活变得不那么困难。同时，我也要感谢宾州州立实验室的师兄师姐，在科研和生活上给予我的帮助和支持。

我还要向实验室的傅春江师兄表示特别感谢，傅春江师兄在我科研初期对我的帮助，使我可以更快的完成从学习到研究转换，找到自己的科研方向。感谢李楠师兄在科研方法上的帮助，感谢姜腾师兄和张泽瀚师兄在深度学习方面对我的指导，感谢李双宏师兄在分布式控制方面的指导，是你们帮助我在科研道路上披荆斩棘。感谢实验室的其它师兄、师姐、师弟、师妹。想起大家一起讨论，一起努力，一起度过的时光，我十分怀念。在我两年半的科研生活中，这珍贵的情谊是我非常宝贵的回忆。

感谢宋新建，孔祥瑞和尹莹莹在我生病期间对我的照顾，你们对我的帮助是我按时完成学位论文的保证。感谢平时一起健身、游泳、打球的朋友们，你们的督促让我保持了良好的体魄。感谢 B1403292 班的所有同学和班主任崔平老师，可以和你们成为一个集体共同度过宝贵的研究生时光是我的荣幸。

我要向父母表达感谢。本科毕业时，毫不犹豫的支持我读研究生的选择和从事自己喜欢的科研工作。没有父母无私的爱与奉献，无限的支持与鼓励，我不可能顺利完成硕士的学习，以及即将开始的博士深造。所以，我要感谢父母的爱与关心。

最后，我要再一次向所有给予我指导与帮助的老师，关心支持我的亲友们表达感谢。



## 攻读学位期间发表的学术论文

- [1] WENBO GUO, CHUN ZHONG, YUPU YANG. Spectral Clustering based Active Learning with Applications to Text Classification[C]. MATEC Web of Conferences. Vol. 56. EDP Sciences, 2016. (EI)
- [2] WENBO GUO, LIANG GONG, YUPU YANG. Low-rank data representation and subspace clustering for active learning[C]. International Journal of Machine Learning and Computing (Accepted) (EI)
- [3] SHUANGHONG LI, WENBO GUO, YUPU YANG. Distributed Fault Diagnosis of Plantwide Process for Fuel Cell Power System[J]. Journal of Computational and Theoretical Nanoscience. (Accepted) (SCI)



## 攻读学位期间申请的专利

- [1] 宋新建, 郭文博, 詹承俊, 张泽瀚, 杨煜普。“基于 FPGA 的磁通门微小信号检测系统及方法”, 专利号公开号: CN105572606A