硕士学位论文

基于码表理论的恶意代码检测技术研究

RESEARCH OF MALICIOUS CODE DETECTION TECHNOLOGY BASED ON CODE BOOK THEORY

朱玉龙

哈尔滨工业大学

2011 年 6 月

国内图书分类号：TP393.08　　学校代码：10213
国际图书分类号：621.3　　密级：公开

工学硕士学位论文

基于码表理论的恶意代码检测技术研究

硕士研究生：朱玉龙
导　　师：翟健宏 副教授
申请学位级别：工学硕士
学科、专业：计算机科学与技术
所在单位：计算机科学与技术学院
答辩日期：2011 年 6 月
授予学位单位：哈尔滨工业大学

Classified Index： TP393.08
U.D.C.： 621.3

Dissertation for the Master Degree in Engineering

RESEARCH OF MALICIOUS CODE DETECTION TECHNOLOGY BASED ON CODE BOOK THEORY

Candidate：	Zhu Yulong
Supervisor：	Associate Prof. Zhai Jianhong
Academic Degree Applied for：	Master of Engineering
Specialty：	Computer Science and Technology
Affiliation：	School of Computer Science and Technology
Date of Defence：	June， 2011
University：	Harbin Institute of Technology

摘　要

信息时代的来临，使网络来到我们每个人的身边。而病毒、蠕虫、木马等恶意代码，也随着网络经济的蓬勃而迅猛发展。网络安全和人身安全对网民而言一样重要。恶意代码检测系统使用的检测技术，主要有特征码扫描和行为特征检测等，商业中主要应用基于特征比对的静态判定方法。然而目前大多数方法提取的特征都包含大量冗余特征。

机器学习的方法是近年来恶意代码检测的一个方向，也取得了较好的成果。码表理论目前广泛应用于图像压缩领域，比如智能视频监控系统中稀疏编码算法的应用，这类似于机器学习中模式分类问题，也是目前研究的热点。行人的检测问题被看作是一个二分类问题，分类的最终目的是区分出行人或非行人。这与恶意代检测有很多类似之处。

码表理论的核心就是生成码书，采用机器学习的方式直接对样本的原始信息进行相应处理，使处理的结果具备一定规律和有序的结构，且能完全表达出原始信息的结构本质。码书的设计实际上就是解决如何选择有代表性的矢量作为码字来尽可能精确表示整个矢量空间。图像空间中为了实现压缩的目的，要对相似或相近的点进行代表选取，而要从海量的指令特征集中选取那些最具代表性的特征码也要进行类似的代表选取。

本文将码书应用于恶意代码检测领域，使恶意代码样本经过码书处理后含有尽可能少的特征向量。稀疏化的特征向量能完全表达出原有样本的性质，去除掉部分冗余特征。根据恶意代码的结构特征进行分析与分类，实验中取得了良好的效果。可以对已知的恶意代码进行检测，对未知的恶意代码也有一定的检测能力。

关键词：恶意代码；静态检测；码书

Abstract

With the advent of information age，network can be seen everywhere around us. Malicious codes such as virus， worms and trojans are spread more frequent on the network.The network is as important as personal safety. The main ways for detecting the malicious code are signatures detection and behavior detection．The static method is more useful in commercial.But the current methods have extra features.

In recent years ，machine learning is a hot spot in malicious code detection，and achieved good results.Code book theory are more used in image compression field. Pedestrian detection has become one o f t he hottest topics in the domain o f computer vision. It can be co nsidered as a two classification problem. We used sparse coding to learn a slightly hig her-level， more succinct feature represent ation from the unlabeled data that randomly dow nloaded from the Internet. Then we applied this representation to the target classification problem by transfer learning .To distingguish between acts and non acts.This is as same as the malicious code detection，put things into two categories. The core of the code table theory is code book generation, is to reduce the original information processing，it is a machine learing approach.The processing results have certain rules and ordered structure， and it can fully express the essence of the original information structure. Code book design is actually to address how to select a representative vector as a codeword to represent the entire vector space as accurately as possible. Image space in order to achieve the purpose of compression， in the process ，select the effective features， and the feature set of instructions from a large number of the most representative of those selected have a similar signature representative selection.

This article will apply the theory of code book in malicious code detection. the malicious code samples processed through the code book feature vector that contains as little as possible， sparse feature vectors fully express the nature of the original sample to get rid of some redundant characteristics. According to the structural features of malicious code analysis and classification， the experiment has achieved good results. Can detect known malicious code， on unknown malicious code detection has some ability.

Keywords：malicious code, static detection, code book

目录

第1章 绪论

1.1 课题研究的背景和意义

1.1.1 信息安全现状

当前网络安全已经成为国际性问题，它不仅影响到社会领域，而且也影响到军事领域。2011 年 5 月 27 日搜狐新闻报道：“我国国防部证实中国已组建网络部队，强化网络安全防护，已成为军队军事训练的重要内容之一”。

随着互联网产业爆炸式的发展，现在的人们或主动或被动的不可避免的进入到了网络时代。信息时代的来临，使计算机深入到各行各业之中，当今时代人们的工作、生活和娱乐正紧密的和计算机结合在了一起。互联网的已经彻底改变了全世界的社会结构和经济结构，改变了许许多多人的工作、生活方式和习惯。经济的高速发展和生活的快节奏使人们通过因特网传递、共享资源和下载信息成为人们工作生活的必要方式和途径。然而，人们在网络的这些活动并非是风平浪静，无处不在的各种病毒、木马、蠕虫后门正充斥在人们的活动之中，可以说我们所使用的网络现状是百毒缠身、防不胜防。

CNNIC（中国互联网中心）第 26 次中国互联网络发展状况调查统计报告[1]的统计结果：“截至 2010 年 12 月年底，中国网民规模达到 4.57 亿人，比 2009 年增加 7330 万人，互联网普及率快速上升至 34.3%”。

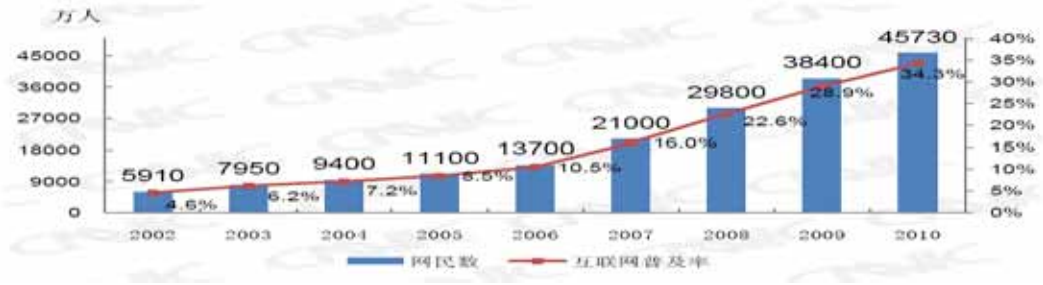

图 1-1 国内网民数量变化图

CNCERT（国家互联网应急中心）5 月 6 日，公布了 2010 年一季度网络安全情况[2]，一季度国家互联网应急中心共收到 6225 个事件报告，其中垃圾邮件、网页挂马、网页仿冒事件的报告数量尤为突出。对此，国家互联网应急中心提醒信息系统安全漏洞仍是互联网企业和普通用户面临的主要安全威胁，2010 年一季度国家信息安全漏洞共享平台上新增漏洞条目 539 条。

2010 年安天实验室关于互联网信息安全威胁报告称：“2010 年捕获恶意代码总量与 2009 年相比，增长了 30%，为 939.7721 万个；这其中又以黑客工具最少，

占总比率的 1%，木马数量最多，占总比例的 67%，而这其中网络游戏盗号木马排在第一位，已达到了 170 万个”。

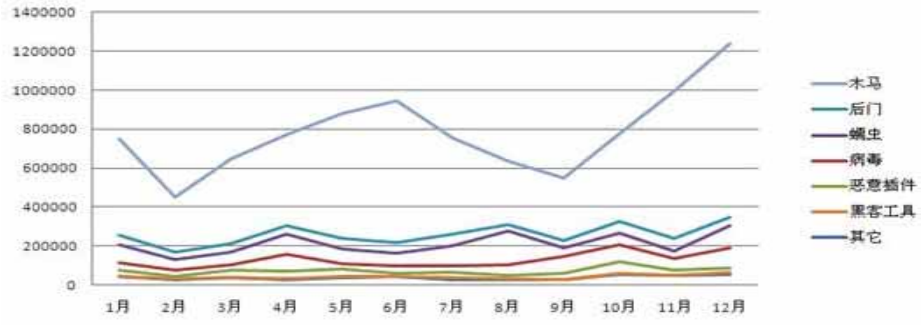

图 1-2 按类别月统计恶意代码变化图

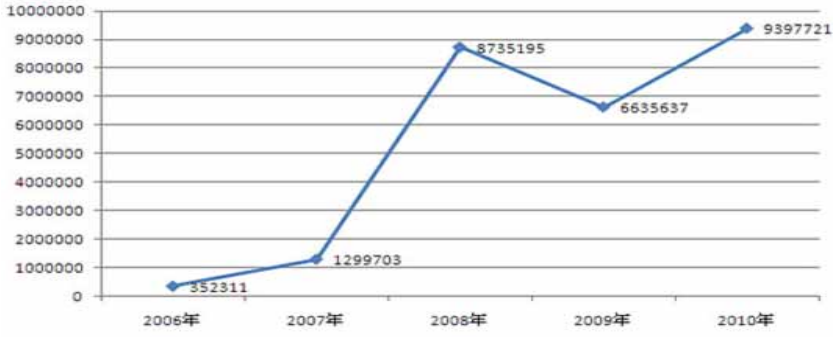

图 1-3 近年捕获恶意代码变化图

从报告中我们还可以看出随着网络规模的扩张和网民数量的飞速增长，现代社会对网络的依存度越来越高，特别是在当今效益至上的这个年代，随着各种网络应用娱乐的快速发展，互联网的经济效益呈现爆炸性的增长。互联网在使人们的工作生活在得到丰富和便利的同时，也将他们各自的各种隐私和利益在网络上呈现出来出来，网络安全问题也凸显成为全民急需应对的问题。恶意代码的使命也从当初的炫耀变成了现在纯粹为了经济利益，并形成了庞大的产业链。2010 年发现的通过挂马网站传播木马，其主要目的是以盗网游戏、银号账号等。另外还有为了达成同样目的的木马下载器、“点击器”等木马大量出现。

1.1.2 恶意代码定义

电脑病毒概念的起源很早，病毒程序的蓝图第一次是在电脑的先驱者John Von Neumann的论文《复杂自动装置的理论及组识的进行》中被提及并被泛化描述。1975 年美国科普作家John Brunner写了他的著名的“Shock Wave Rider”（《震荡波骑士》）一书，书中首次畅想了人类未来生活的信息社会中的各种冲突，计算机如何被作为正义和邪恶两方斗争的工具的故事。1983 年 11 月 3 日，正在南加州大学读书的学生Fred Cohen，写了一个在UNIX系统下会引起系统死机的程序，让人们初步认识了电脑病毒天然的破坏性概念。1984 年在他的论文[3]给出定义：“计算机

病毒是一种程序，它可以通过修改其他程序使之含有自身可能演化的副本，而感染其他程序”，并说明了计算机病毒已经真真切切的存在了，让计算机病毒在世界范围内正式进入人们的视线。早期的恶意代码的主要指代计算机病毒。

1990 年，Howard L.Johnson 针对恶意代码给出 Malicious Code 这个称呼。90 年代末，随着计算机网络技术的发展恶意代码定义的内延和外涵逐渐丰满，当时 Grimes 定义恶意代码为三个方面：“一是媒介可以是存储介质和网络，二是主要的操作在两台计算机系统之间进行，三是它是能够对计算机系统进行入侵和破坏的程序或代码，且重点是并未取得授权许可”。J.Bergeron 等人定义恶意代码为：“它本质是代码的有机组合，其工作的主要目的是破坏系统的保密性和完整性，影响数据的传输和控制流的正确工作方式”。Christodorescu and Jha 用具有达到非法目的人为程序描述恶意代码。McGraw and Morrisett 定义恶意代码为：“它是一种程序代码，目的是恶意破坏或改变系统功能，途径是对软件系统中的那些增加、修改或删除操作进行修改”。微软定义恶意代码为：“当运行时，实现攻击者故意破坏目的的软件”。

1994 年 2 月 18 日颁布的《中华人民共和国计算机信息系统安全保条例》指出：“计算机病毒，是指编制或者插入的破坏计算机功能或者毁坏数据，影响计算机使用，并能自我复制的一组计算机指令或者程序代码”。

恶意代码指那些破坏计算机可靠性、机密性、安全性和数据完整性的代码，一般把恶意程序分成下面几类：木马、蠕虫、计算机病毒、攻击脚本、ActiveX 控件等。

1.1.3 恶意代码的危害性

我们国家的网络现状可以说是“底子薄、网民多、分布广”，目前是网络攻击的受害者，恶意代码的检测和铲除是保障我国网络安全的关键和难点。信息技术的高速发展，推动了信息战、网络战等的新军事技术的应用，而恶意代码是这其中的主要手段之一。

恶意代码的破坏活动[4]，是信息社会的一种新的流氓活动，一方面企业的生产经营、个人用户的私密信息都象被一双隐形的眼睛随时盯着。从更高的层面来说，国家的安全也正面临不可预知的严重威胁。

第一个电脑病毒 C-BRAIN 在 1987 年诞生。巴基斯坦兄弟， Basit 和 Amjad 编写了第一个无论从目的还是工作方式来看，都具有电脑病毒特质的代码段，起因是为了防止别人盗拷他们所写的软件，这是真正被业界公认的病毒的开始。

Morris蠕虫病毒 1988 年在全球泛滥，网络上瘫痪的计算机不计其数，造成了灾难性的后果[5]。

1998 年爆发的 “CIH”病毒，轻易的大面积攻克了网络上超过数十万台计算

机[6]，其编写者仅仅是一个大学生。

Happy 99、Melissa病毒在 1999 年引领了新的破坏潮流，它不但使邮件服务器停止所有服务，而且还造成了互联网络的几近停顿[7]。

近年来让计算机信息界最为震惊的病毒之一——“爱虫”病毒，2000 年 5 月爆发让人始料未及，最后生成的与其相关的变种病毒超过了 50 多个，将 4000 多万台计算机感染[8]的时间也只用了区区一年而已。

在公安部 2001 年主办的我国首次计算机病毒疫情网上调查工作报告中称“曾经感染过计算机病毒的用户高达 73％多，其中占 59％多的用户感染三次以上”，可以看出我国的网络安全现状的严重性。

“红色代码”2001 年 8 月诞生，它主要利用了微软Web服务器中Index服务的安全漏洞。它目标范围的机器会很快被感染，并不间断的肆意传播蠕虫病毒，互联网在相当长一段时间内进入了“红色代码”时代[9]。

SLammer 蠕虫出现在 2003 年，其破坏速度如此之快，仅十分钟就使网络中九成的脆弱主机遭到侵犯。“冲击波"同年 8 月也登上了历史的舞台，并卷走全球电脑用户 20 多亿美金。

“振荡波”仅 2004 年，就使 100 多万台计算机被感染，经济损失以百万美元计。

2005 年出现的“灰鸽子”木马号称是当年中国十大恶意软件之一，其破坏力可见一斑。

2006 年—2009 年间，感染型的蠕虫病毒“熊猫烧香”、利用漏洞从网络入侵的“扫荡波”病毒、以机器狗病毒为代表的利用内核 rootkitrf 技术的多种病毒在我国肆意泛滥，经济损失巨大。

2009 年 12 月—2010 年 1 月，包括 Symantec、Mcafee、TrendMacro 大内的国际安全厂商报道，多家大型国际企业的工作人员遭受到针对客户端程序的 0- Day 漏洞攻击，进而被植入了木马。

2010 年 1 月 12 日晨 7 时起至当日 12 时左右，百度域名遭劫持。其域名 www.baidu.com 遭到了恶意篡改（注册于美国域名注册商处），全球多处用户均无法正常访问百度网站。

恶意代码也正随着网络经济的繁荣产生出巨大的经济“效益“，甚至出现了以此为生的群体，导致恶意代码的数量呈现指数级的增长，各种变形及新技术不断涌现。最近在网络上随处可见的大批盗取网络银行账号密码、游戏账号密码的木马程序是其中的当今代表。恶意代码问题也成为信息安全问题中，刻不容缓的、迫在眉睫的问题。本文的目的就是对恶意代码检测理论进行研究以寻找出可行的检测方法。

1.1.4 恶意代码将会长期存在

就象是俗语中所说的“道高一尺，魔高一丈”那样，总是在新的恶意代码出现之后恶意代码检测技术才会姗姗来迟。从网络状况看恶意代码长期存在的原因有两个方面，首先很多个人用户信息系统缺乏或者根本没有必要的防护系统，最近免费的杀毒软件的兴起使个人电脑的安全状况有了一定的改善，但是因为疏虞防范、戒备心不强再加上很多人的好奇心，恶意代码还是经常被人们无意的执行了；其次是较为准确的分出恶意代码和正常代码还是很难的，Chone 和 Adelmna 曾经提出一个著名的论断：“恶意代码通用检测方法的不可判定性。”目前的情况就是出现新的恶意代码然后经过一段时间、造成一定破坏之后查杀，接着再出现新的，再破坏，再查杀，周而复始。

恶意代码能够快速大范围传播的另一个主要原因是信息共享技术和使用的普及。信息共享在方便人们的工作和生活同时，也引起了信息的广泛、快速流动，而这也是恶意代码入侵的主要途径之一。无论是在 Internet 上的下载，还是从光盘、U 盘等复制的软件，甚至接收的电子邮件都有可能是恶意代码的藏身之处。

恶意代码的一个主要工作向方向就是利用系统的各种漏洞，而系统的脆弱性不仅是无可避免的，而且会随着时间的推移不断暴露出来，针对这些脆弱性的新的恶意代码不断出现，补丁越打越多。AT&T 试验室的 S.Bellovin 在他的一份安全报告中说，计算机网络安全问题中的大约 50%是由软件工程中产生的安全缺陷带来的，而这其中有很来归根结底源于操作系统的脆弱性。就 2010 年而言，微软官方公布的仅浏览器一个方面的高危漏洞就出现很多次：1 月 14 日，微软官方发布安全通报与 WINDOWS 自带浏览器 IE6、IE7、IE8 相关的代号为“Aurora”的漏洞，CVE 编号为 CVE-2010-0249；3 月 9 日，微软官方再次通报与上述浏览器相关的 CVE 编号为 CVE-2010-0806 的漏洞； 11 月 3 日，微软官又一次通报了还是与上述浏览器相关的 CVE 编号为 CVE-2010-3962 的漏洞。

近年来恶意代码随着与各种各样的经济利益关联日益加深，其潜伏性和隐藏性日益增强，破坏的目标、目的及要达成的后果都更加明确。而恶意代码大多数编写者的目的也从最初的技术炫耀变成为获得经济或者政治利益。互联网的开放性给人们带来了便利的同时，也加快了恶意代码的传播，推动了恶意代码编写技术的发展，一条看不见、摸不着的黑客产业链正在形成，中国的木马产业链一年的收入已达到上百亿元，数字惊人。恶意代码的检测与反检测注定是一场持久战。

1.2 国内外研究现状

国外针对恶意代码检测的研究工作开始的很早。最早的是James在 1980 年发

表的一篇具有里程碑意义的论文——《计算机安全威胁监测》。恶意代码检测的概念也第一次进入了人们的视线，从而打开了更多人通向恶意代码检测的研究之门。1987 年Dorothy E.Denning提出了检测的抽象模型[10]，该模型首次将入侵检测做为一种计算机系统安全防御措施提出。最近 20 年，无论国外还是国内在恶意代码检测系统研发方面都有了长足的进展，在智能化和分布式两个方向取得了不错的成果。2001 年，Matthew G.Schultz和Eleazar Eskin二人把数据挖掘理论应用于恶意代码检测方法[11]之中。Mihai Christodorescu在恶意代码静态分析方面[12]，J.Bergeron在基于行为特征的代码分析方面[13]，Tony Abou-Assaleh在模式匹配方面[14]做了大量的研究工作。

以IDS和杀毒软件为代表的传统检测系统都是使用特征码检测技术[15]为基础开发出的恶意代码检测系统，来查杀恶意代码。它使用可以从特定恶意代码中提取出来的，同时不大可能出现在正常的程序中的特征字节序列，进行相关判别。这种方法只处理程序的字节码，不关心它的行为，不用运行恶意代码。这种检测方法工作的前提大多是我们已知且获得这种恶意代码，并利用一定的技术手段分析出能有效代表这种恶意代码的特征码，同时在病毒库内将其定义。下面要做的就是在未知的代码片段中进行比对查找，一旦某种具有这种特征码的恶意代码被检测发现，则其将会被程序锁定。特征码检测技术作为大多数恶意代码检测的主要方法，一直都没有太大的改进。基于特征码查杀技术主要有两个问题：一是对特征匹配的精确性要求高；二是只能检测已知恶意代码，而不能检测未知恶意代码。

为了应对对未知病毒的检测，启发式分析检测方法被提了出来。它是一种利用某种规则和模式达到对未知恶意代码检测的方法。目前主要有静态、基于代码仿真、基于神经网络三种启发检测方法。静态启发检测是对传统特征码扫描的一种补充；基于代码仿真的启发式检测实现了用一个虚拟机来仿真 CPU 和内存管理系统，进而模拟出代码执行过程，从而可以监视恶意代码的行为。避免了因需要在真实环境中执行恶意代码，使操作系统和用户数据受到威胁。基于神经网络的启发式分析方法成功地将神经网络应用扩展到引导型病毒和 Win32 病毒的启发式检测中。

基于行为特征[16]的恶意代码检测是目前的研究热点之一。它的原理就是分析出恶意代码的代码行为和功能特征，并将它定义为区分恶意代码的依据。行为分析是一种动态分析技术，也是目前被较多应于计算机安全方面一个技术。如抗间谍软件、广告软件以及僵尸网络等等。在Windows系统中，恶意代码的行为特征从两个方面来定义：一是有异于正常程序的调用参数的调用序列；二是有异于正常程序的API集合，所有这些系统的调用时序序列的调用参数集合构成了程序的行为

特征。目前比较典型的基于行为的恶意代码判定方法是基于系统调用序列异常模式和参数的检测方法，但是这种方法在成功率高的同时，也带来了高的误判率。

近年来恶意代码检测技术的还有一个应用较为广泛的方向，它主要是利用对网络中主机的网络行为进行分析，以达到对未知恶意代码检测。今天的互联网无处不在，物联网在很多国家都被提上了议事日程。穿行于网络环境中的恶意代码便具有了高度的、无处不在的主动性和分布性。基于网络行为分析的恶意代码检测系统通过分析网络主机和主机之间互连的数据，根据不同的恶意代码中包含的攻击方法差异，设计不同的网络行为检测规则。虽然恶意代码的种类不胜枚举，但是形形色色的恶意代码所要达到的目的就那么几种，其主要攻击方法也很容易被定义。网络行为进行分析主要表现在两个方面：一是对网络数据进行行为分析，网络数据行为分析是指识别出日常网络通讯流量中异常通讯方式的能力。简单的说就是网络分析人员尝试从简单地阻止过量的网络通讯设置中，识别查找出网络中可能存在的异常行为；二是将网络中的数据进行分门别类。拒绝服务攻击是让人们记忆很深的一种攻击方式，即攻击者想办法让目标机器停止提供服务，是黑客常用的攻击手段之。拒绝服务攻击无论是对互联网服务提供商还是对大型网络基础设施，甚至对一个地区、国家都会产生重大安全威胁。而对主机的网络行为进行分析，是解决这种问题的最有效方法之一。

1.3 本文的主要研究内容

本文从我国的现状开始，针对当前网络的发展，恶意代码的破坏和影响，简述了国内外目前的恶意代码检测技术的研究，并对恶意代码检测技术研究的意义进行了详细论述。

另外还从恶意代码的定义讲起，介绍了常见的相关理论和检测技术，从码书理论的介绍到基于码表理论的 LBG 算法和稀疏编码算法进行的详细研究。本文考虑针对有效的指令序列进行特征码查找，所以对反汇编算法进行了介绍，并在 UBUNTU 环境下进行了指令截取实验。

此外还对应用于恶意代码检测中的分类方法进行了部分的总结和归纳，并在此基础上结合了机器学习在恶意代码的检测技术中的应用。并分别根据距离分类器和贝叶斯分类原理，分别设计出应用码书理论，结合基于距离的的分类和基于结构相似度的理论设计出一种较好的分类器，它们都能对恶意代码和正常代码作出较为准确的判断，各自取得了不同的效果，证明了码书理论应用于恶意代码检测的可行性。主要包括以下几个方面：

（1）首先介绍了我国网络安全的现状和恶意代码检测技术研究在我国的迫切性，概述了恶意代码的现状、定义以及发展方向；研究恶意代码检测系统的现

状、主要检测技术以及发展方向，并对部分恶意代码检测系统的主要工作原理进行了阐述。

（2）讨论了不同阶段恶意代码的定义，恶意代码的发展历程。还有我国关于恶意代码的定义和相关措施。

对目前在图像处理中广泛应用的基于码书理论的 LBG 算法和目前较为前沿的稀疏编码算法进行了理论和算法方面的研究；并结合机器学习和当前的恶意代码检测联系起来，设计出一种可行的基于码书理论的恶意代码检测方向。

（3）利用反汇编技术分析和处理代码样本，对目前存在的特征提取截取方法和分类算法进行了介绍和对比，分别进行了距离分类和相似度分类的对比。

（4）实现码表理论在恶意代码检测中的应用，并取得了较好的效果。对应分类器的各个模块，一是设计出基于稀疏编码算法的恶意代码检测算法，并对恶意代码和正常代码的指令序列进行了有效的提取和分类；二是设计出基于相似度的 LBG 算法和稀疏编码算法结合的整体检测算法，这种算法对类型相同可者相近的代码检测效果较好。

（5）对两种设计方法分别设计了实验方案，并根据比对结果进行了测试分析，总结规律，给出了可以改进的方案。

1.4 本文的组织结构

本文共分为四章，各章的组织结构如下：

第一章，介绍了我们国家的网络现状，概述了近些年主要恶意代码在全球范围内的影响和破坏。最后给出国内外的研究现状，以及在此基础上描述了此项研究的意义。

第二章，简要介绍了恶意代码的一些常识，又结合机器学习对恶意代码的检测技术研究做了说明。其次介绍了码书理论的相关技术，最后讨论了当前码表理论在图像领域的应用，对 LBG 算法和稀疏编码算法进行详细研究。

第三章，对反汇编理论进行了简单介绍，并对目前常用的特征提取截取方法和分类算法作了简述。利用反汇编工具将代码进行预处理，并在此基础上进行特征段截取。本文最后实验应用基于反汇编截取定长特征段，对距离分类等相关分类进行了对比。

第四章，恶意代码检测系统的设计与实现。主要是以码书理论为基础，稀疏编码算法和相似度算法在恶意代码检测系统中的设计实现，实验中分别选取不同码书大小和不同种类的恶意代码，取得了不错的效果。

第2章 机器学习与码表理论

为了瓦解恶意代码的威胁，人们除了研究其实现机理之外，还进一步研究恶意代码的分析方法与检测技术。现阶段常用分析技术和检测方法常见的主要有：静态检测技术、启发式扫描技术[17]、沙箱技术[18]、校验和技术[19]、机器学习技术等等。另外还有针对恶意代码的变形技术，利用复杂结构进行检测的方法[20]，如基于语义[21]等。机器学习中的多种方法近年来在恶意代码检测方面的应用研究很多，取得很多成果。码表理论作为图像处理中应用较为成功的应用，对模式分类的问题具有较好的效果，和恶意代码检测的两分类性具有相通性。

2.1 机器学习

T. Mitchell在《机器学习》一书中定义[22]：如果一个计算机程序要完成某类任务T，其完成任务的性能可以用P衡量，该程序根据经验E改进P，则称该程序针对任务T以性能P衡量从经验E中学习。他把学习分为三种类型：有监督（有指导）的学习——从其输入/输出的实例中学习一个函数；无监督（无指导）的学习——在未提供明确的输出值情况下，学习输入的模式；强化学习——从强化物中学习，而不是根据指导进行学习。

机器学习起源于研究人类的学习行为，研究人类认识客观世界，获得各种技能知识的基本方法（如一般化、特殊化、类比、归纳等等），并最终用机器模拟人类的学习行为，以使机器可以象人类一样具备从实例中学习的能力，能够从已知的实例中总结出规律，并能从中预测出可能的事实或现象，能够利用已经获取的技能知识，不断的改变原有的知识结构以完善和改进自身的性能。这其中目前的研究现状主要有类人思考、类人行为、理性思考、理性行为四种。类人思考或类人行为：直接模拟 /追随人；理性思考或理性行为：间接模拟 / 概括人 —更普遍。

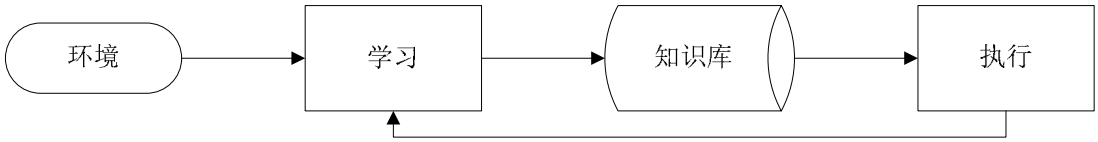

图 2-1 机器学习模型执行

学习能力的好坏表示了判断智能行为的其中一个非常重要特征。现在对学习能力的描述大都从三个方面来讲：一是强调学习的外部行为效果，认为学习是通过某种规则使系统能做出适应性的变化，通过学习之后系统能够更为有效的完成下一次同样或者类似的任务；二是强调学习的内部过程，它用构造和修改表示所

经历的事务；三是从知识工程的实用性角度进行强调，也就是说学习的主要目的是知识的获得。目前来说大家普遍认同的比较简单的学习模型如图 2-1 所示。

环境将外部信息源传递至系统，学习单元根据接收到的信息更新知识库，提高系统有效执行任务的效能；执行单元利用知识库中已知的知识表示执行任务，并返回结果；最后把结果反馈回到学习单元，作为其如何进一步学习的依据。学习单元的目的是改善执行单元动作，执行部分是执行整个学习系统的核心[23]。机器学习模型之外，知识表示是一个核心问题，知识表示就是研究如何用最恰当的形式来组织所需的各种知识，就是把知识映射成一种形式化语言的过程。

2.1.1 码表理论的分类性

利用机器学习检测未知恶意代码已经有了很多尝试，1994 年 Kephart 等人提出用人工神经网络方法检测引导区病毒，2000 年 Arnold 提出用同样的方法检测未知恶意代码，都取得了一定的效果。

机器学习中的很多算法都曾经应用于恶意代码检测，主要是利用它的分类作用，机器学习在文本自动分类[24]方面得到了广泛的应用，这就是上面所说的有监督学习。它先从大量的已知性质的恶意代码和正常代码中学习到正确有效的类别特征，生成能够识别出恶意代码和正常代码的标尺。未知代码用已经经过训练的分类器进行判断，从而决定将其定义为恶意代码还是正常代码。利用这种学习方法能较好的检测出未知恶意代码，也是目前恶意代码理论研究中的重点和热点之一。

比如现在的热点研究，码表理论算法之一的稀疏编码算法，在行人的检测中的应用。稀疏编码算法被应用于模式分类问题，行人的检测问题被看作是一个二分类问题。分类的最终目的区分出行人或非行人[25]，或者从大量含有行人的训练样本中，提取特征向量，再用分类器直接进行分类[26]。

2.2 码表理论介绍

2.2.1 码表理论简介

码表（也称作 Code book）理论的核心就是采用机器学习的方式直接对样本的原始信息进行相应处理，对处理的结果具备一定规律和有序的结构，且能完全表达出原始信息的结构本质。Code book 理论核心思想是重建或重构原样本的信息，而且要求性质基本保持，简要表述形式如下：

$$X = D\alpha \tag{2-1}$$

式中，X 代表样本的原始信息；

D 指代码表，也用来指代对样本 X 信息重建过程中所使用的一组基；

α 具体用来表示信息重构过程中需要的新特征。

码表的生成算法也就是码书生成的过程，码表传统的生成算法有 K_means、Mean_shift（聚类）、稀疏编码算法和 LBG 算法。码表理论是目前广泛应用于图像处理中的一种有效方法，并已在图像压缩传输中取得较好的效果，关于它的研究也是目前图像处理中的热点之一。用于图像压缩传输中，主要分为两种：无损压缩和有损压缩。

（1）无损压缩是一种压缩率较小的方法。无损压缩是依据字符出现的概率来构造其相应的码字，是一种变长编码。常见的有哈夫曼编码等。它的一般步骤是：将信息源中各块按出现的概率从小到大排序，最小的两个概率合并成新的概率，与其它剩余的概率再一起重新排序，重复上述过程，直至最后两个概率之和为 1，然后从上到下对各块进行编码，传输时传递索引至解码端进行解压缩即可。它是一种无损压缩，不会造成信息的损失，可以较好的回复到原始图像效果。但是其压缩比率很小。

（2）有损压缩的压缩比率比较高，应用性较好。有损压缩是目前图像处理中的主要研究热点之一。其实现主要体现在两个方面：用相对较小的分割块实现高的压缩比以和快速解压缩。目前关于码书在有损压缩图像处理中的应用有很多成果，而这其中最典型是就是 LBG 算法，而稀疏编码则是近年来是研究热点，是一种自适应图像统计方法。

在图像处理中数据能够压缩到的比率和恢复的效果是检验方法可不可行的主要指标。而这里面码书质量的好坏则起到至关重要的作用。码表理论的关键是设计一个高质量的码书，而对于码书生的高质量研究重点则在主要关乎以下两点：一是提高码书质量，就是使在训练集下生成的码书宽被惩罚的平均失真尽量最小；二是算法的效率要好，就是说要耗费可能少的时间去生成等质量的码书。

码表算法被常见应用于文本检测、图像处理、信号处理等方向，而这些方向的处理方法类似于二进制流的技术，相应的二进制流检测是在恶意代码检测中最常见的。这也说明码表理论之于恶意代码检测，与之前的检测方法有相似性，是具备一定的理论基础的。

本文主要对在图像处理中现在应用比较广泛的 LBG 算法和目前的热点稀疏编码算法进行分析，同时结合机器学习在恶意代码检测的应用，期望寻找出一种能生成较好质量的码书并能较好适用于恶意代码检测中的算法模型。

2.3 LBG 相关研究

LBG 算法本质上相当于 Lord-Max 方法的多维推广，是由 Linde、Buzo、Gray 三人在 1980 年提出，算法主要是通过训练矢量集和一定的迭代算法来逼近最优的

再生码本。

2.3.1 基本原理

LBG算法[27]属于矢量量化算法的一种，它采用有损数字图象压缩方式，应用广泛，在众多矢量量化算法中，是一个获得广泛应用的代表。使用LBG算法设计一个可行的数据压缩算法，首先要被考虑的是要保证图像可以被恢复到可以忍受的质量范围[28]。当然另一方面也要达到能尽量减少存储位率，以使大部分的图像冗余信息被过滤掉，达到能够在传输中节省时间这一主要目的。所以矢量量化在这里需要做到以下两点：第一实现较高压缩比，一般是用已被分割成相对较小的块来达到这一目的，也就是实现了分解的目标。它是把一个要被处理的整幅图像割裂成类似$n*n$量化单元模式，同时根据量化单元的相邻相似性对每个量化单元内的像素进行有规律的重新排列，这样就可以生成一个有规律的输入矢量集，并将这些矢量集作为整个原始图像的代表保存起来；第二是传输或者应用完成后的解压缩要能够实现快速化，传输过程中针对每个分解矢量都是先寻找到可以代表其本身的“最近码字”，并将所有的“最近码字”按某种规则用“码字索引”表示出来，这样传输量就大大减少，实现了压缩的目的。最后在对应的接收端，只需进行查找索引这样的解压缩操作，就可以得到容忍范围内的重构图像。

在输入矢量的量化阶段要实现较低的位率，处理方法是用同一可忍受范围内的类似矢量，用相同的“最近码字”来表示。对于这个处理过程中需要先定义一个在容忍范围内的失真测度，常见的有欧式距离（euclidian distance）、标准化欧氏距离(normalized euclidian distance)、马哈朗诺比斯距离(mahalanobis distance)等等。其中不需要计算属性方差与协方差的欧氏距离速度较快，效果较弱；标准化欧氏距离进行了属性方差的计算，提高了分类的效果，时间稍慢，被应用较多；马哈朗诺比斯距离分类准确率最高，但是其时间需求很高，适合用于数据量较少时。一般情况，在搜索出并生成所有“最近码字”集合，即定义成由所有具有代表性的码字组成码书集合：$C = \{c_1, c_2, \cdots, c_m\}$。而在生成码字过程中要进行可容忍的限制，就是对于每一个分割得到的输入矢量x，在图像块集合中用相同维数的、最近距离的“最近码字”代替，其中距离u限制为$u(x,c_j) = min\{u_2(x,c_i) \mid i \neq j\}$，其中$u_2(x,c_i) = \sum_{t=1}^{w}(x_t - c_{it})^2$（$i = 1,\cdots,m$，$w$ 为线性空间的维数），这里的码字c_i就是那些可以替代某些范围x的“最近码字”。

2.3.2 算法简述

LBG算法在向量量化过程中，首先定义一个w维空间R^w。该空间的一个样本子集被某种分割方法割裂成m个被称作Voronoi区域的R_i 区域（i= 1，2，…，m）。对

于任一个被分裂到区域R_i内的分割块都用一个w维矢量c_i表示，这就相当于由所有该区域内的样本选举出能真正代表样本中心的“最近码字”，它能使均方量误差化最小，所有上通过上述方式在所有Voronoi区域均生成一个码字（code word）。由所有上述码字组成一个集合$C=\{c_1, c_2, \cdots, c_m\}$，这个集合C被称为码书。图 2-2 表示当w=2 时Voronoi区域的简单表示。

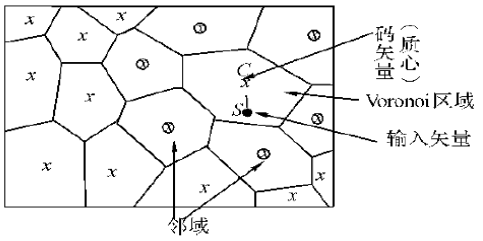

图 2-2 划分 w=2 时的码字区域

上面生成码书的过程中，需要尤为注意的是象如何控制量化失真使其值尽可能的小，至少要在可控范围内。通常对任一样本集$X=\{x_1, x_2, \cdots, x_m\}$，其失真值的计算公式为：$D=\sum_{i=1}^{m}\sum_{x_{kj}\in p_i}\|x_k - c_i\|_2$。LBG算法大多都是先定义一个初始码书$C_0$(有时也可以直接当做最终码书)，对由某一划分给出的初始码书$C0=\{c^1_0, c^2_0, \cdots, c^m_0\}$，根据最邻近的原则计算出量化失真$D_0$；若是能够寻找到一种较为先进的码书划分方法能够更新C_0为C_1，并使得$C_1=\{c^1_1, c^2_1, \cdots, c^m_1\}$，此时再根据最邻近的原则计算出新量化失真$D_1$，如此重复上述过程，对码书进行不断的优化，直至前后两次C_0和C_1的下降比率$(D_L - D_{L+1})/D_L$小于最开始设定的可以被容忍的门限值 ε，我们认为此时的划分是完成了且是最优化的。

2.3.3 初始码书的选择

能不能够找到一个较好的生成初始码书的划分直接影响到LBG算法的效果，初始码书如何分布，每一个码字的代表性如何，都使LBG算法的收敛速度快或者慢，以及局部最优解产生的可能性发生变化。在LBG算法的研究中初始码书的选择[29]占有举足轻重的地位，好的初始码书可以直接作为最终码书使用，关于初始码书的选择方法很多，常见的方法有以下四种。

（1）随机编码法是最简单的一种方法。针对任一训练集，随机编码法就是随机从训练样本集中抽取 N（N 为初始定义的码字个数值）个特征矢量，并把由这些特征矢量组成的集合定义为初始码书。这种选择方法优点是计算量低，对运算数据量很大的样本有优势。而且这种方法从原理上来说比较公平，有能产生最优码书的可能。因为根据香农信源编码理论，每一个特征矢量被选中的机会都是1/N，符合其理论基础。但同时也是因为机会均等的存在，也可能有一些非典型的特征矢量被抽取选中，有可能形成过疏或过细空间，还有可能产生个别不可利用

的的码字，最终结果是大大减弱了码书的性能。

（2）成对最近邻法[30]是一种性能较为优越的初始化码书的技术。这种码书的算法比随机编码法复杂一些，它是在初始码书的生成初级阶段，要求每一个特征矢量都占有一个单独的集合空间。算法的下一步就是每次都对任意两个单独的集合空间进行距离计算，做一个从头至尾的比较，从所有的独立空间集合中查找出其中最近的一对集合空间，找到后把这两个集合空间合并成一个较大的集合空间，如此进行不断的循环直至最后，合并出的较大的集合空间数目达到码书的数目要求为止。由成对最近邻法生成的码书不仅可以当作LBG算法的初始码收，还可以当作最终码书直接使用。它和随机编码法最大的不同是它通过循环得到的最终码字不一定是训练矢量集中的原始特征矢量，成对最近邻法的重点是如何查找出最近邻的码字以及如何在最终的每一个集合空间中计算出每一个代表码字。

（3）原始的二分裂法[31]是一种从训练矢量中心不断发散生成码书的一种方法。这种码书算法首先是对整个训练矢量集做求中心点运算，将所求得的中心矢量作为码书的第一个码字c_1。然后对此码字作一个小范围失真的加成D，这样就可以产生两个码字。利用LBG算法把上面产生的码字作为初始码书再次进行发散，把每一个经过优化的码字进行分裂，可以产生二倍的码字，如此通过不断的循环，最后产生码书所要求的码字数目为止。

（4）二进制位分裂法[32]是一种生成快速，能很好显示训练样本集分布性质的初始码书生成算法。这种算法的本质是利用上面提到过的Voronoi区域来对图像空间分割（如以每个特征段包含的单独位像素值的个数为基准）。对每一个图像块，通常的说定义由k个像素点构成（日常生活中常见有是红黄蓝三色），从另外一个方面说就是把每个块看成是k维空间的一个点。关于Voronoi区域的二进制分裂具体表示，我们这里取k=2 详细说明。k=2 时表示每个图像块由 2 个像素单位组成（每个单位用一个 8 位来表示），映射到k维空间就是映射到只有x轴和y轴的平面坐标系中，此时的图像空间被分割成 4 个象限，为了方便标记分别定义为 0、0 象限，0、1 象限，1、0 象限和 1、1 象限。对于任意一个图像块，每一个像素点的从高位到低位决定了图像块应处于的象限位置。如每一个二进制值为（0…，0…）图像块都属于 00 象限，对于所有二进制值为（0…，1…）的图象块，它们都属于 01 象限…。如此类推把所有的图象块都分配到所有的 4 个象限之中。当上面所有图像块都被分配到 4 个象限当中之后，再对每个象限中所属的所有图像块进行子分配，这个时候已经被使用过的最高位被忽略，第二高位被重新定义为最高位，再重新进行上面的分裂过程。如此不断进行上面的重复过程，直至得到生成所定义码书中的码字数目为止，得到最终的码书C。如当分配的过程进行到第三阶段时，就是图像块把第三高们作为最高位时，每一个二进制值为（001…，010…）图像

块都属于 0、0 象限下的 00、01 象限中的 001、010 象限。这里我们生成最终码书后，还要对每个最终象限做一个疏密检测，包含图象块太少的象限与离之最近的象限合并，包含图象块太密的象限则从高到底分离，直至达到标准为止。

2.4 稀疏编码相关研究

起源于哺乳动物视觉系统的稀疏编码是一种神经网络的表示方法（视神经网络研究），同时处于活跃状态的多维数据可以只用一小部分神经元将其表征出来。一般来说稀疏编码属于神经计算科学的研究范畴[33]。

人们在 20 世纪经过大量的生理实验发现：哺乳类生物视觉初级视皮层 V1 区的第四层有 5000 万个细胞，而在视网膜和 LGN 内只有约 100 万个神经节细胞，5000 万个要远远大于 100 万个；通常哺乳类生物感知的外界图像在视觉通路间按层次进行传递和抽取，而 V1 区接收到的经 LGN 外侧状膝体传送过来的由视网膜感光细胞抽取的图像特征中包含的信息量基本没有变化，这也就说明经过 LGN 转发的信息经过 V1 区处理后信息得到了精炼；视觉皮层下的细胞感觉野方向敏感性特别显著，单个神经元只反射感受野中被刺激的部分和某一特定频段的信息（如图像特征中的特定方向的边沿、纹理、线条等）。这表明了在信息编码的级别上，简单细胞层次与复杂细胞层次已经有了很大的不同，V1 区第四层的简单细胞用大容量的信息处理进行抽象特征的表达，它存在超定的性质，它的编码较好的使大的空间维数和较小的空间维数形成了一个有效的映射，具有超完备性。感受野在这里可以被理解成信号编码滤波器，包含局部性、方向性和带通性。稀疏编码理论通过模拟人类神经系统完成对信息的特殊加工机制。在图像特征提取、以及模式识别等方面都得到了较为广泛的应用。稀疏编码原则是对图像特征中的特定方向的边沿、纹理、线条等的处理要象神经系统中每个神经元对这些刺激的表达方式来进行讲述。

稀疏编码在图像处理中，对应于同一刺激下的神经元群，稀疏编码只要处理一小部分被激活的神经元，是一种针对稀疏能量新陈代谢的处理方式。稀疏特性是稀疏编码中神经元响应需要具备的性质，同一稀疏编码处理下的神经元群表现在两个方面：一是在同一信号下不是所有的神经元响应从而激活；二是大段接收信号时间内，大多数的神经元是不活跃的，从单个神经元角度考虑，其活跃的规律会出现波峰和平缓的山脊部分。根据信息论中度量不确定性大小的性质，从信息熵的角度出发，在用同一均值和方差的概率分布考量时，稀疏分布的信息熵较小，而正态分布的信息熵则要大很多。

2.4.1 稀疏编码算法模型

B.A.Olshausen[34]和D.J.Field[35]分别在 1987 年至 2002 年间连续发表了重要论

文，把稀疏编码和V1 区简单细胞感觉野的响应性质有机的结合起来，最后定义了一种“超完备的稀疏编码算法”。他们指出稀疏编码处理自然图像过程中的基函数的性质和感觉野的响应特质具有很强的共性。他们对稀疏编码提出了一个经典的线性叠加稀疏编码模型[36]。这其中最重要的一点就是他们在应用具有简单细胞活跃性质的基函数时，提出了约束稀疏性的方法，对基函数进行优化处理，多维数据经稀疏编码处理后大部分分量处于不能响应的状态，很少一部分编码处于活跃形态，和数学中的超高斯分布相类似，这样最后能够得到不仅稀疏性较好而且性质不变的编码模型。如果用数学的方式来描述的话，可以把稀疏编码理解成用线性分解的方式处理多维数据。形式化的描述就是，如果把输入随机分量定义为$X=(x_1, x_2, \cdots, x_u)^w$（维数为$u$），通过线性分解转换后，得到生成向量$Y=(y_1, y_2, \cdots, y_v)$（维数为$v$）。通常情况下$u \leqslant v$，如果把经过线性处理后得到的矩阵记录为T（维数为$u \times v$），则通用的数学公式可以记为：稀疏变换矩阵$Y=TX$，这里面向量$(y_1, y_2, \cdots, y_v)$被称为独立向量，各自独立；相比较变换矩阵中的每一个行向量，稀疏分量Y具备一般稀疏性（经转换处理后），性质象是小波变化中的小波基。

B.A.Olshausen 和 D.J.Field 还提出了针对如何让目标基函数更好的模拟主视皮层区简单细胞感受野，对应提出了优化的结构，要使基函数能产生很好的稀疏性，同时又具备保持图像质地变化不大。下面是经过一定优化的基函数一种表示方式。

Min F(D，α) = [重建误差]+λ[稀疏性惩罚]

$$= \quad L(X,D) + \lambda[\sum_{j=1}^{n} S(\frac{\alpha_j}{\sigma})] \tag{2-2}$$

$$= \quad \sum_{j=1}^{n} \| x_j - D\alpha_i \|_2^2 + \lambda \sum_{j=1}^{n} S(\frac{\alpha_j}{\sigma})$$

公式中，集合 X 指代输入的初始样本特征；D 指代通常意义上所说的码书。α 指代经过码书 D 相应转换后得到的能代表样本的新特征，是矩阵相乘的一个参量系数。L(X，D)就是重建误差值，通常情况通过计算初始样本特征与重建后的新特征之间的均方差的值来得到，这个值的大小反应了样本经码书 D 处理后能否仍然具备呈现原始样本 X 信息的性质。正参数 λ 能通过来断的调整，来维持第一项和第二项之间合理的比重，也就是不会出现纯粹的稀疏性现象，那样的话重建误差那一项将会变的很大，会超出我们可以容忍的范围，反之也不能出现只考虑重建误差，而这时的稀疏性就没有意义了，也不能表达出原有的只有少数分量处于明显激活状态这个性质了。σ 指代和α 相关的一个归一系数；S(x)被称为稀疏惩罚函数，它通常是非线性的基函数。通过这个函数我们可以对一个样本，计算出

一个经过码书 D 后产生的稀疏性好坏，如稀疏性小时表现为α 中非 0 系数很少，反之稀疏性大时表现为，α 中非 0 系数会有很多。正常情况下非 0 系数太多太少都不好。

2.4.2 稀疏编码算法简介

稀疏编码算法在处理图象中最大的优势就是算法本身会通过机器自主的发现，并定义出图像的相关重要本质的特征。而一般说，对于检测未知恶意代码，至关重要的一点就是从未知的恶意代码中寻找出其固有的本质、能代表它自身的那些特征群族。整个过程中我不一要预先定义特征，稀疏编码算法只是个一个不断的进行迭代发现有用信息的过程，在恶意代码分类过程也就相当于生成一些集合，分别总结出恶意代码特征向量和正常代码集合，这两个集合使代码具备很强的规律性，能对未知代码分析后得到准确的归类。

2.4.3 有监督的稀疏编码标准算法

B.A.Olshausen 和 D.J.Field 提出的稀疏编码算法是一种标准的算法模型，但是它的显著缺点之一是收敛速度慢且很难收敛于最优处的问题，其实它就是机器学习里面的无监督学习问题。如果我们要想使稀疏编码能够得到较为有效的应用，那么如何改进算法，使其更具有效率性，同时要能适应机器学习中的有监督学习的思想。如果应用于恶意代码检测，则相对于有监督学习的稀疏编码算法需要加入一种约束的机制，使得标准稀疏编码算法模型得到改进。一是只要简单监督能够区分包含两种类别的样分集合，二是能够对带有类别的信息进行有效判别。

结合机器学习的相关知识、稀疏编码算法模型理论和恶意代码检测中的判别方法，我们改进无监督学习稀疏编码算法为有监督的学习时，只要加入两种判别约束即可，一是距离约束，二是类别约束。距离约束的主要功能是通过一定的约束条件，使重建后的训练样本在类间距和类内距的作用下分成明显的两类。类别约束就是相当于直接加入具有不同或相同的类别属性的一种判别约束，使训练样本的类别属性得到加强，这和支持向量机的思想具有异典同工之妙。下面是这两种分类作具体约束描述。

（1）距离判别约束，通常的是用某种规则定义样本间的距离计算方式。如使用欧氏距离考量约束。首先我们定义两个样本特征矩阵，用来区分重建后的训练样本，使它能代表基本的两个类别。类别Ⅰ我们用$\alpha_{Ⅰ}=\{\alpha^{1}_{Ⅰ}, \alpha^{2}_{Ⅰ}, \ldots, \alpha^{N}_{Ⅰ}\}$表示第一类特征矩阵，用$\alpha_{Ⅱ}=\{\alpha^{1}_{Ⅱ}, \alpha^{2}_{Ⅱ}, \ldots, \alpha^{M}_{Ⅱ}\}$来表示类别Ⅱ（N、M分别是Ⅰ类和Ⅱ类中所包含样本数量）。比如在欧氏距离计算中用样本间距离、类内距、类间距三种距离完成距离判别约束。类内距主要的作用是对同类样本特征空间中，每个样本与此类样本类别中心的远近程度，它能显示出本类别的紧凑程

度，它就是通过计算求出样本特征向量到它所属类别样本中心欧氏距离。类间距的大小表达了一类样本与另一样本间的距离远近，就是这两者之间的分离程度，它先求出某一类别的样本中心点，然后计算需要考量的那个样本特征向量距离类别中心点的值，最后通过大小进行比较。

样本间距离：

$$D(\alpha_{\mathrm{I}},\alpha_{\mathrm{II}})=\sqrt{\sum_{j=1}^{n}(\alpha_{\mathrm{I}j}-\alpha_{\mathrm{II}j})^2} \tag{2-3}$$

类内距：

$$D_h=D(\alpha_{\mathrm{I}}^{\,i},\tilde{w}_{\mathrm{I}}) \tag{2-4}$$

上式表示第Ⅰ类样本中的第 i 个特征向量到第Ⅰ类样本中心 $\tilde{w}_{\mathrm{II}}$ 的距离。

类间距：

$$D_l=D(\alpha_{\mathrm{I}}^{\,i},\hat{\tilde{w}}_i) \tag{2-5}$$

上式表示第Ⅰ类样本中的第 i 个特征向量到第Ⅱ类样本中心 $\tilde{w}_{\mathrm{II}}$ 的距离。

根据分类的一般原则，在稀疏编码算法中我们要想产生正确有效的分类，就要使两种类别具有明显的界限，就是说一方面要尽可能的使同一类别间的类内距比较小，另一方面要尽可能的使不同类别的类间距比较大。在公式中我们可以定义一个比例系数：

$$\mathrm{Dis}(\alpha)=\log(\frac{D_h}{D_l})^2=\log(\sum_{k=1}^{n}(\alpha_{ik}-\tilde{m}_{ik})^2)-\log(\sum_{k=1}^{n}(\alpha_{ij}-\hat{\tilde{m}}_{ik})^2) \tag{2-6}$$

在 B.A.Olshausen 和 D.J.Field 的模型中加入这个比例系数，可以把目标函数优化成如下形式：

$$\min F(D,\alpha)=[\text{重建误差}]+\lambda_1[\text{稀疏性惩罚}]+\lambda_2\mathrm{Dis}(\alpha) \tag{2-7}$$

优化后的上述目标函数，相当于在 B.A.Olshausen 和 D.J.Field 提出的基本算法中加入了同一类别的监督思想，同一类别之间会变得更加紧凑，与其它类别之间的界限也会变得更加明显。

（2）类别判别约束，就是在训练样本中预先加入类别的标识，使训练之后各类别的特征向量会被查找出来。这相当于在算法模型中加入类别约束，对应于恶意代码检测，是把类别属性标志（比如 y=(-1，+1)）分别添加至两个类别的训练样本中，以使训练具有类别约束性，这里面还要加入判定函数 F*标志位和判定函数的对应关系如下所示：

第一种情况是：F* < 0　　（y = -1）

第二种情况是：F*> 0　　（y = +1）

判定函数F*是线性函数，常用的约束模型形式有下面两种：一是线性函数和系数α 相关，F* (X，α ，θ) = hTα + c(θ是模型参数形式化为 { h∈R^k ， c∈R })；二是双线性函数和系数α 和样本X和都有密切关系，F* (X，α ，θ) = XT hα + c(θ 形式化为{ h ∈R^{n*k} ， c∈R })。上面两种约束模型相比，前者的运算效率较高，但是后者的可以应用的范围则要宽泛一些。

判定函数 F*可以优化模型成下式：

$$\min_{\theta} \sum_{i=1}^{m} C(y_i f(x_i, \alpha_i, \theta)) + \lambda \| \theta \|_2^2 \tag{2-8}$$

式中 C 是一种 log 形式的惩罚函数。

稀疏算法模型被加入类别约束后函数形式优化成下式：

$$\min_{D,\theta,\alpha}(\sum_{i=1}^{m} C(y_i f(x_i, \alpha_i, \theta)) + \lambda_0 \| x_i - D\alpha_i \|_2^2 + \lambda_1 \| \alpha_i \|_1) + \lambda_2 \| \theta \|_2^2 \tag{2-9}$$

于是目标函数就被优化成如下形式：

$$\min_{D,\theta,\alpha} (\sum_{i=1}^{m} C(y_i f(x_i, \alpha_i, \theta)) + \lambda_0 \| x_i - D\alpha_i \|_2^2 + \lambda_1 \| \alpha_i \|_1) \tag{2-10}$$

优化后的稀疏算法模型类似于 SVM 思想，通过加入一定的标记量化区分不同类别的样本信息，这样在实现稀疏算法的既要保证一定的稀疏性，又要有容忍范围内的重建误差，最后才能做到精确的分类。

2.5 本章小结

本章从恶意代码的常识介绍入手，叙述了恶意代码在不同论文中的描述。通过引入机器学习和码表理论知识，对在图像处理中已经得到较为广泛的应用的码表算法作了概述。并针对码表理论在二进制流处理中的应用，机器学习的多种方法在恶意代码检测中取得了很好的成果，利用它们之间的联系，码表理论其引入至恶意代码检测中。

介绍了码表理论和图像处理中的各种应用，分别叙述了有损压缩和无损压缩的在图像处理中的一般过程和应用。这其中的重点是有损压缩，目前应用效果好且被广泛研究的是 LBG 算法和稀疏编码算法。对应 LBG 算法分别从基本原理、算法简述、初始码书选择三个方面进行分析，稀疏编码算法则从模型、算法、有监督的稀疏编码标准算法进行了分析。

机器学习的很多方法都被成功的应用于恶意代码检测和图像处理，码表理论从理论基础上来讲也是能被应用于恶意代码检测的。而且方法中还可以根据码表

理论自身的特点，恶意代码结构有可能只通过机器自身的学习、归纳、分析、总结就可以不用通过人为因素，直接得到代码的标志特征集。而且还可能对未知恶意代码具有一定的识别作用。

第3章 特征提取和分类算法

目前， 在病毒、木马等恶意代码的研究中发现，这些恶意文件通常都是以二进制代码形式存在的。商业上的恶意代码检测就是从已知的恶意代码中查找出其特征码，利用特征码对可疑数据进行分类查杀。而对于恶意代码来说其特征码大多和指令相关较大。

3.1 反汇编

以二进制代码形式存在的恶意代码类似伪装成一个子代码潜伏在被感染后的机器的正常二进制代码中。相当于把正常的二进制代码作为自己的母体，一旦母体二进制代码被执行，则子代码就会很快随之执行，并且会进一步感染其它的二进制代码。最坏的情况是子代码窃取到系统的最高控制权，并实行各种破坏活动。因此利用二进制代码分析恶意代码的程序特征是提取出有效特征查杀恶意代码的有效办法，但是分析恶意代码的二进制程序逻辑是很困难的。后来大量的从恶意代码中抽取有效的二进制特征序列的实验证明，大多数的恶意代码特征段都和指令序列有关，基于指令序列的恶意代码检测研究有很多。对指令序列的处理可以利用反汇编器将二进制代码转化为较容易理解的汇编代码，对二进制反汇编算法进行研究在基于指令序列的恶意代码检测方法中极为重要。反汇编程序一般有两种，静态反汇编如 W32Dasm，动态反汇编的如 Softice。本文是要讨论静态反汇编。

3.1.1 静态反汇编算法

静态反汇编[37]（Linear Sweep）是从根据经过反汇编处理后得到的程序结构样式，从提示信息入手进行分析。过程中将可执行代码作为反汇编对象，对需要处理的代码作扫描遍历分析，通过分析获得的汇编程序，进而归纳出程序功能，不涉及到执行相关代码。目前，常用的静态反汇编算法主要有线性遍历和递归遍历这两种。

线性遍历算法[38]简单的来讲就是对反汇编的全部代码区作简单的有序遍历，起始点从程序第一个指令字节开始，整个过程中对所有的检测到的指令进行逐条反汇编，顺序的把经历的字节识别转化成指令，直到遇到终结点，它的终结点产生在遍历过程中遇到与非法指令时（字节不能和任意存在的操作码相匹配），流程结束。

递归遍历算法[39](Recursive Traversal)是指在遍历的过程中，是按顺序进行还是跳转，主要由遇到的指令类别进行判决。也就是在反汇编的过程中加入已经通

过反汇编得到的控制流指令属性标志，最后把整个过程分成两类，要么顺序遍历，要么转移至另一个新的程序基本模块处。这种遍历算法最大特点是过程中寻找每一个控制流（比如象调用指令、跳转指令、程序返回指令那样的能使程序流程发生变化的指令流）在程序中的位置，把它作为遍历过程中的分叉点。反汇编流进行到这些交叉点处时，它们把这些指令指向的新的地址作为下一个要进行遍历的地址，它会尽可能先行遍历完所有分叉点处那些可能存在的分支。在处理的过程中它相当于对每一个模块进行顺序遍历，每一个起始点定义在模块的入口地址，也就是对目标程序进行反汇编起始处。整个过程就是，从目标程序起始点处开始先是顺序遍历进行反汇编，如果反汇编得到的指令是一个非转移指令，则继续进行顺序反汇编。对下一个反汇编得到的指令再进行判断，如果是转移类指令，则此时需要将目标模块更新为新的转移指令所指向的指的程序基本模块处，把新的目标地址作为下一步进行反汇编开始处，再进行上面的循环，直至遇到非法指令，整个遍历结束。

3.1.2 相关反汇编工具

反汇编工具有很多，如有 IDA Pro、C32Asm、W32DASM、花指令清除器 1.2 等等。IDA Pro 是一款专业的反汇编工具，是由 DataRescue 开发的现已成为了很多 Hacker，Cracker， Reverse engineerer 的必备工具。C32Asm 的特点是集反汇编、16 进制工具 、Hiew 修改功能于一体。静态分析中文软件的利器的优点是速度快。花指令工具可以去除花指令，虽然目前效果不是太好，但目前也是对付静态分析的一个重要手段。

3.1.3 指令提取

本文中在对恶意代码的处理中采用顺序的方式，提取一定长度的代码中的指令集序列。文中使用工具objdump进行相关处理，主要使用了objdump -d指令（例如objdump -d filename.exe），环境在Ubuntu Linux下进行。

主要的过程如下：

首先读取文件，调用system函数执行objdump反汇编shell指令，将结果保存在txt文档下，然后调用system函数执行shell指令运用正则表达式提取十六进制代码，形成一个有序的指令序列。截取指令级特征向量时，把每一个指令当作一个“分词”开始处，对有序的指令序列样本进行逐一扫描，这里把每个指令作为特征向量的首部。把预先定义的固定长度值m做为截取的长度上限，每一个原样本都会被串截成一定规律的定长串的集合，当然最后一个串可能邮现截取长取不够的现象，处理方法可以用简单的加0补足其余位数。

在Ubuntu Linux环境下进行的。

主要步骤：

（1）读取文件名字(readdir( ) 函数)

（2）调用 system 函数执行 objdump 反汇编 shell 指令保存为 txt 文档

反汇编结果如下示例：

CrashReporter.exe：　　file format pei-i386

Disassembly of section .text：

00401000 <.text>：

401000：　55　　push　%ebp

401001：　8b ec　　mov　%esp，%ebp

401003：　6a ff　　push　$0xffffffff

401005：　68 fd 21 45 00　　push　$0x4521fd

…

（3）调用 system 函数执行 shell 指令，运用正则表达式（cat as/filename.txt | grep '^　[0-9]' | awk -F'\t' '{print $2}'）提取十六进制代码。（此时，^和[0-9]之间有两个空格）

结果如下：

55

8b ec

6a ff

68 fd 21 45 00

64 a1 00 00 00 00

50

51

a1 90 ec 46 00

…

3.2 特征提取

特征提取就是在海量的具有可分性质的数据样本中，为了达到使样本分成明显的两类或者两类以上的类别，而在样本的若干分段中选取出那些能代表其根本类别特征的分段。这些特征段选取的好坏直接影响分类器分类的准确概率，也是数据具有可分性的依据。而对应于恶意代码，其必然会包含一些特殊的指令代码，它们在正常代码中不存在或者存在极少，我们称之为恶意代码的特征。恶意代码会包含多个特征，它们是能够正确分清恶意代码还是非恶意代码的标志，我们主要目标就是在恶意代码样本中定义一定的特征段，利用它们可以对恶意还是非恶意代码进行正确标示。

特征提取是恶意代码检测中必要的步骤之一，在恶意代码检测研究中，关于

种特征提取的方法目前经常被应用的有很多。目前，随着恶意代码检测技术不断发展，对于特征提取的研究也在不断深入。如Schultz采用提取ASCII字符串、字节序列等特征代码段[40]进行未知恶意代码的检测；Assaleh采用了一种模型被称为一阶马尔科夫链N-gram特征提取方法；Kolter提出在二进制代码特征的提取中采用变长N-gram特征[41]提取办法。现今N-gram特征提取和变长N-gram特征提取是大多数恶意代码检测中常用的二进制代码检测方法，提取出特征后可以采用如朴素贝叶斯、决策树、K最近邻等方法进行分类，这种方法实现的恶意代码检测系统。

3.2.1 N-gram

N-gram 方法被广泛应用文本分类、信息检索等许多方面。1994 年 Kerphart 最早在恶意代码检测领域使用字节序列 N-gram 提取特征，能自动从数据样本中提取出病毒的有效特征。1997 年 Yang 对 Kerphart 方法中容易产生庞大的字节集合、占用巨大的存储空间的情况进行了改进，他提出了相关特征选择法，只抽取特征集合中相互关联的少部分特征，对字节集合进行了较好的瘦身，这也和本文中的通过稀疏编码算法达到的目的相同。N-gram 提取特征通常的做法是先设计一个滑动窗口，窗口长度为一个可变动的 N，通过窗口的不断滑动从有序的字节序列中收集出部分重叠的子字节序列集合，其滑动幅度每次一个长度单位。比如，有序的字节序列为(05 f2 4b 00 36 81)，窗口长度 N 定义为 3，则通过窗口的不断滑动会产生{(05 f2 4b)、(f2 4b 00)、(4b 00 36)、(00 36 81)}的字节集合。

3.2.2 变长 N-gram

变长 N-gram 先是被应用于文本分类和入侵检测。变长 N-gram 提取特征的核心是使用一定的策略，每次从有序的字节序列中查找那些有实际意义的字节序列串。相对应于 N-gram，它产生的连续字节串长度是变化的，最终产生的字节集合是由长度不一的串组成的。变长 N-gram 的重点是截取有意义的串，这也是难点。通常的做法是先通过一个遍历算法寻找出有序字节序列中所有可能的断点，防止拆开那些有意义连续串，最后把每两个断点之间的连续字节串抽取出来组成集合。

3.2.3 Cohen 算法

Cohen在 2002 年提出了一种专家投票算法[42]，查找存在的断点，进行段落分割。实现过程中定义了两个专家负责投票工作，分别是频率专家和熵专家。其中，频率专家对频率状况进行测量统计，对任一子序列而言，它作为一个段落的可能性随着频率值的增长而增长。如果专家给出的统计值很高，则算法就认定这个子序列是一个完整的子序列，没有可能的断点存在，可以作为一个独立的变长字节串。如果相反则算法认定其中可能包含若干个子序列；熵专家的主要功能是

对每个点进行熵值的计算，每一个点被认定为是断落结尾的可能性和这个点的熵值成正比例增长。如果一个点被认定为断点，则其熵值会较大。这也相当于对于同一个字节串，如果在不同的位置，它后面相邻的字节串大多数情况下都是相同的，则就说明这个字节串可以扩展成更长的串，反之它后面相邻的字节串在所有的情况下都各不相同，则算法就可以认定这所有的位置点都是断点，是段落的结尾的可能性较高。最后综合考虑这两个专家的投票情况，根据累加后得到的分数值的高低来定义位置为断点的可能性大小。将两个被认定为断点之间的字节串提取出来并将其定义特征串。对于每一个位置，滑动窗口在滑动固定长度的情况下，两个投票专家会分别将票投给频率值最大和熵值最大的那个位置，Cohen给每一个位置一个初始值为 0 的分数标记位，位置每得到一次投票其分数就加一。

这里选择通过深度 d=4 的 Trie 数据结构对 Cohen 的专家投票算法进行详细描述。比如有序字节序列 S=(05 2f 4b 05 2f 4b 4b 05)，字节串 S 的深度 d=4 的 Trie 数据结构如图 3-1 所示。

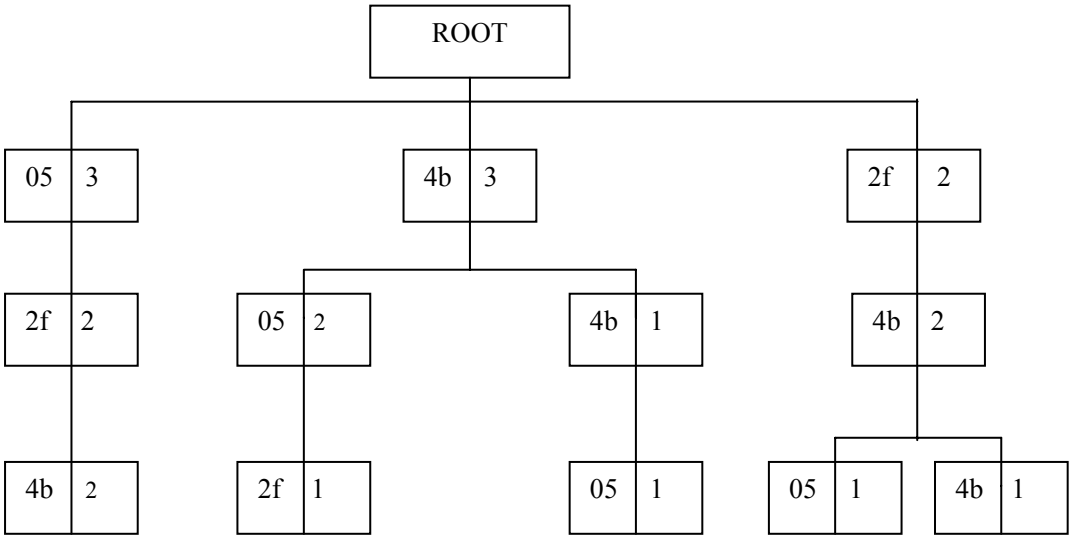

图 3-1 深度 d=4 的 Trie 数据结构

图 3-1 表示固定长度为 3 的滑动窗口，按照从左到右的顺序遍历字节串 S，此时第一个得到的子字节串为(05｜2f｜4b｜)。在 Trie 数据结构中将此字节串表示为左子树，05 为左子树的一个节点，每一个节点旁边的数字表示位置节点在字节串中出现的次数，05 出现了 3 次，(05 2f)出现了 2 次。前一个位置节点的频率通过其后紧跟着的两个位置节点的频率值求和计算得到，比如对 Trie 第一层位置节点(05)频率的计算：F(05)=(F05)+F(2f 4b)，式中 F(2f 4b)代表的是 Trie 第二层且父节点是(2f)的(4b)的频率。Trie 数据结构中第一层的(05)的熵、第二层且 parent 节点是(05)的(2f)的熵分别表示第一个位置点和第二个位置点。分别逐层计算，最后把此窗口中的最大熵的位置计算出来，并做上投票标记，该位置出的相应标记分递增

一个单位。在窗口从左到右的滑动过程中，分别定位出频率和熵的最大位置标记点，并对所有的标记值进行更新。当所有的字节串中的位置点都被遍历且标记值被更新后，可以根据标记值查询出局部极大值的位置，也就是此处的位置可以被定义为字节串的断点处。如上面串 S 经过计算后分割为((05 2f 4b)、(05 2f 4b)、(4b 05))特征集合。

3.3 分类算法

分类算法属于有监督的学习，目前其本身的算法很多。分类算法的好坏直接影响恶意代码检测的准确率和误报率。一般的分类算法的前提是先找出能较好代表各种类别性质的训练样本集合，通过对训练样本的归纳学习，总结出包含于样本且具有各个类别的代表性的特征，并将它们定义为判断为相应类别的规则，使用这种规则可以对其它未知的样本进行一定的定性。如样本$X=\{x_0,x_1,\cdots,x_n\}$，对于其中任一$x_i(i=0,1,\cdots,n)$，都已经知道其性质，并做了类别标记，通过这种监督学习，最后会产生一个可以对未知性质的样本进行类别标记的分类器模型。

3.3.1 决策树

决策树(Decision Tree)是归纳学习算法中最简单也是最成功的算法之一。Decision Tree 的输入是描述事物的属性的集合，而其输出一般情况是一个离散的分类，通常是一个二值分类，或为真或为假。Decision Tree 建树的核心是属性的选择，一个 Decision Tree 包含一个根节点，若干中间结点和许多叶结点。1983 年 Quinlan 提出决策树 ID3 算法，算法的目标就是使用尽可能少的步骤对样本进行分类判断，相同节点的情况下，树的深度要尽可能的少。Quinlan 在算法中提出了考量信息论中熵的信息量的大小概念，来提高 Decision Tree 分类的效率。Decision Tree 的分类方法相当于从根结点开始的一个先序遍历。开始时，在根结点处，算法对被分类对象没有了解，样本的性质处于最大不确定状态。然后随着遍历的逐层进行，样本的属性值被逐一判断，遍历朝着叶子的方向深入，直至到达叶子节点。这相当于不断减少样本的不确定性，不断去选择具体的子树，最终不确性的值降为 0，分类完成。

图 3-2 所示为关于天气是否适合打网球决策树的例子。outlook、humidity、wind 为是否适合打网球的三个属性，具体的属性值则为 sunny、overcast、rain、high， normal、strong、weak。决策树的目标是输出 yes 或者 no 的目标值。通过对已知天气的属性值的逐层属性判断，最后决定其属于那一个子树的 yes 或者 no。

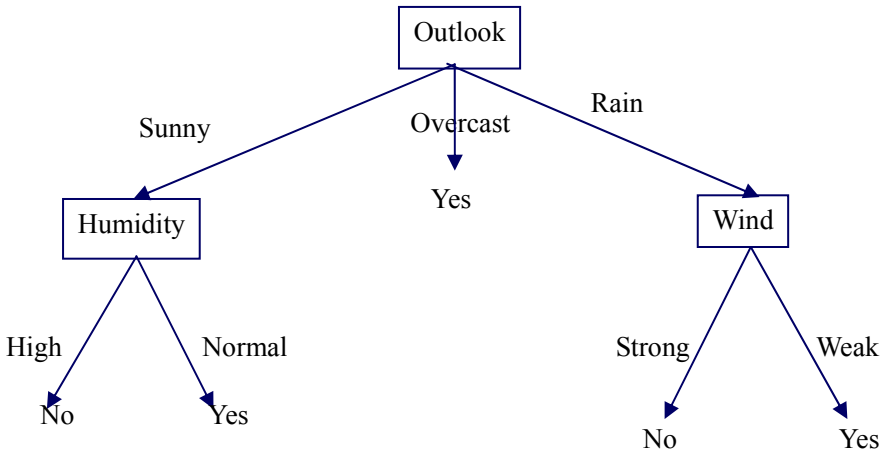

图 3-2 天气是否适合打网球决策树

3.3.2 朴素贝叶斯

贝叶斯学习根据给定数据计算各种假设的可能性，即根据概率为每个假设赋予相应的权值，然后在此基础上进行预测。贝叶斯推理的机理是根据概率判定可能，相当于近似判断，目前在恶意代码检测中应用也较多。贝叶斯分类对于某个假设概率的更迭，主要是通过观察到更多的训练样例，学习得到的。当然在增量学习之前要先求出先验概率，它和之后的学习数据共同决定假设的最终概率。学习过程是一个概率推理过程。对于一个新的未知性质的实例，分类器通过对实例中的多个已知加权假设的概率进行综合考量得到。R.Lo 等提出了朴素贝叶斯检测恶意代码，取得了较好的效果。

朴素贝叶斯算法建立在一般贝叶斯算法的基础上。朴素贝叶斯分类算法和实际存在的情况有差异，它通过假定各属性因素之间完全独立，即不存在任何联系而得到的一种简化贝叶斯算法，一方面它的分类效率较高，别一方面也可能引起大的误差。

假设目标的可能属性集合A={a_1，a_2，…，a_n}，类别的所有集合V={v_1，v_2，…，v_n}。对于A中任意属性ai，属于类v_i的概率为：

$$P(V \mid a1,\ a2,\ \cdots,\ an)=\frac{P(a_1,a_2,...,a_n \mid v_i)\cdot P(V)}{P(a_1,a_2,...,a_n)}$$

$$=\alpha \cdot P(V)\cdot P(v_1,v_2,...,v_n \mid V)$$

$$=\alpha \cdot P(V)\cdot \prod_i^n P(v_1,v_2,...,v_n \mid V) \qquad (3\text{-}1)$$

$$=\alpha \cdot P(V)\cdot \prod_i^n P(A \mid V)$$

上式中，α 是正则因子；

P（d_i）是类d_i的先验概率；

P(d_i | h_1， h_2，…， h_n)是后验概率。

对于先验概率的计算大多采用如下方式计算，如先验概率P(d_i)：

$$P(v_i) = \frac{N_i}{N} \tag{3-2}$$

式中，Ni 验是训练样本中属于的个数；

N 是训练样本集的总数。

在恶意代码检测中，h_i是第i个特征属性的值。d_i表示i个类别，总共有恶意代码和正常代码两类。用朴素贝叶斯推理方法进行分类时，每个类别的先验概率已经被求出，需要判断分出的类别也是固定的，这里是正常和恶意两类，分类器对实例中所有属性类别的后验概率进行计算，从而得到出某个类别的最大概率值，再利用朴素贝叶斯公式求出未知代码的类别分数，根据这个分类器返回的一个类别的预测值判定数据的类别归属。贝叶斯分类器输出函数表示为：

$$v_M = \arg\max_{v_j \in \{yes,no\}} P(v_j) \prod_i P(a_i \mid v_j) \tag{3-3}$$

其中 yes 和 no 分别代表恶意代码和正常代码这两类，公式中的类条件概率 $P(a_i \mid v_j)$ 从训练数据中学习观察得到。

假定各个条件相互独立，可以进行简略的类别分数计算，图 3-3 表示在 m 个类条件的作用下的朴素贝叶斯模型。

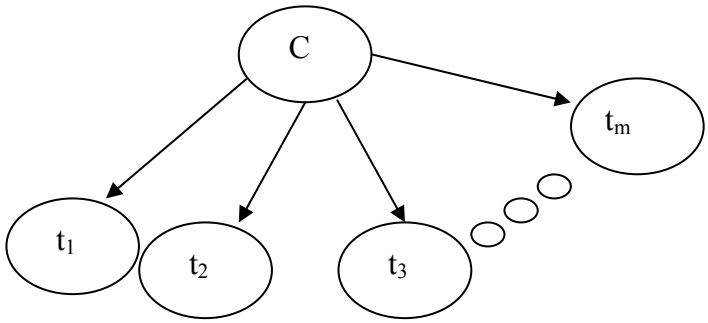

图 3-3 简略朴素贝叶斯模型

根据恶意代码和正常代码的特征属性建立起朴素贝叶斯模型，在假定代码中的各个特征属性相互独立的情况下模型：C指类别某一种属性，t_i(i=1，2，…，m)指某一实例的属性特征值。

3.3.3 最小距离分类器

最小距离分类器[43]是一种基于向量空间模型的简单分类算法。最小距离分类器的关键核心是通过计算出类别中心。比如训练集$S=\{s_1, s_2, \cdots, s_m\}$，通过求平

均计算生成一个中心向量$D_i(i=1，2，\cdots，n)$能够代表该类，n为类别的个数。对于任一个未知类别的数据实例X，分别计算求出其与$D_i(i=1，2，\cdots，n)$之间的距离值d_i^2，根据距离值定义该实例性质和与之距离最类的那个中心所代表的类别相同。共有三个集合表示，样本$X=\{ x_1，x_2，\cdots，x_k ，c\}$，距离$D_i=\{d_{j1}^2，d_{j2}^2，\cdots，d_{jk}^2，c\}$，$C=\{c_1，c_2，\cdots，c_n\}$代表类别集合。

最小距离分类器就是一种设计出某种距离度量方式，通过这种特定的方式来比较距离的远近。关于距离度量方式最常见的可能就是欧式距离（euclidian distance）。因基于特征码的恶意代码检测中，每一个数据实例相当于包含若干个属性的元组，现如今标准化欧式距离（normalized euclidian distance）和马哈郎诺比斯距离（mahalanobis distance）应用在对数元组进行分类的分类器中效果很好。

（1）euclidian distance的应用范围最广泛。如果用L_i^2表示距离，则样本X的欧式距离公式为(3-4)。

$$L_i^2=(X-D_i)^t(X- D_i) \tag{3-4}$$

（2）normalized euclidian distance是应用效果较好的一种方法。方法中同样用L_i^2表示距离，则此时样本X的距离公式改变为(3-5)。

$$L_i^2=(X-D_i)^t \Phi^{-1}_i (X- D_i) \tag{3-5}$$

Φ_i的矩阵形式如下所示，其中σ_{jj}意义是归属第i类数据实例所求出的属性j的方差（j=1，2，…，m），

$$\Phi_i=\begin{bmatrix}\sigma_{11} & & & \\ & \sigma_{22} & & \\ & & \ddots & \\ & & & \sigma_{mm}\end{bmatrix}。$$

（3）mahalanobis distance是一种计算较为复杂的方法。这里也用L_i^2表示距离，相应样本X的距离公式改变为(3-6)。

$$L_i^2=(X-D_i)^t \Psi^{-1}_i (X- D_i) \tag{3-6}$$

Ψ_i的矩阵如下式所示，式中的σ_{jk}（$j=1，2，\cdots，m；k=1，2，\cdots，m;j\neq k$）的意义是第j类数据中属性j类和属性k 类进行的协方差，

$$\Psi_i=\begin{bmatrix}\sigma_{11} & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2m} \\ \vdots & \vdots & & \vdots \\ \sigma_{m1} & \sigma_{m2} & \cdots & \sigma_{mm}\end{bmatrix}。$$

3.3.4 支持向量机

支持向量机（Support Vector Machine，SVM）的理论基础来源于统计学原

理，其泛化力强、结构简单，将结构风险尽可能降到足够小，在包括机器学习和模式识别在内的许多研究和应用领域都取得了明显成果，也正引起更多人的关注。支持向量机的核心是如何构造一个 m-1 维决策超平面，使我们能够把原 n 维空间实例点隔离开来。同时向量机在构造平面过程中遵循尽可能的原则。就是构造最佳分类平面，使得被这个平面分成的各个类别中的最边缘的向量—也就是距最佳分类平面最近的点，与这个平面的距离最大。

Cover定理认为：对于模式分类中较为复杂的问题，将问题映射到高维空间进行线性划分比在低维空间更容易、更准确。Support Vector Machine就是利用这个定理将向量从低维映射到高维，并在最大间隔距离原则制约下，在这个空间里构造向量距离尽可能大的最佳分类平面。而且支持向量机的方法在未知恶意病毒检测[44]中取得了一定的效果。

3.3.5 距离分类实验

用 500 个全 0 行模拟恶意代码进行分类实验。训练样本是随机选取的 2000 行非全 0 行代码，再把 500 个全 0 行代码随机插入到非全 0 行代码中。

实验中类别聚集效果明显。其中有 500 行的代码聚到以 3687 为中心点距离的类别中，这 500 个就是全 0 行代码。结果证明有相同代码结构特征的代码行能聚集到一类之中，对完全相同的代码行，可以准确的分出类别。

3.4 本章小结

本章主要介绍了现阶段的几种常见的反汇编的方法和基本原理，并对静态反汇编在恶意代码中的应用和现状作了分析，最后在 Ubuntu 环境实现了对恶意代码的指令提取，将整个训练样本空间中的向量转化成指令的集合。

对提取出的指令依照原序列，把指令个数作为特征串的集合数，每个指令做为一个起始点，以便于以后在指令集合空间截取定义的特征段。最后又对决策树、贝叶斯分类器、最小距离分类器和支持向量机的分类方法进行了分析和介绍，并对各种分类方法在恶意代码检测中的应用进行了分析和比对，它们也是目前恶意代码检测中常用的方法，也是有效的方法。

最后在实验中进行了距离分类的归类，得到了在类别特征明显的情况下，距离分类是有效的。

第4章 恶意代码检测系统的实现

关于对恶意代码检测的理论研究目前的热点主要都是集中在基于行为特征等动态方面，基于静态的相关研究较少，而另一方面，静态特征码比对检测方法仍然被广大厂商大范围的应用于商用，也说明其实际意义比理论意义具有更大的价值。到现在为止这种方法，可以说是用了至少十几年了，一直没有进行太大的改进。另一方面码表理论是目前很多领域理论的研究热点之一，也取得了较好的成果，它从模仿人工视神经系统开始，从人工智能角度考虑应该有很好的适应性和宽广的范围，将这个理论和恶意代码检测联系起来，希望能给静态检测找到一个新的方向。

4.1 系统主要功能

4.1.1 简要概述

本文设计的恶意代码检测系统是基于码书理论的，主要有由 LBG 算法演变的对训练样本整体的所有数据分段进行压缩处理，以从中找到能代表此类恶意代码或正常代码样本的特征代码段，从而最后根据这些特征代码段对未知性质的实例代码进行类别判定；还有一个主要是基于稀疏编码理论对每一个训练样本进行稀疏化处理，使处理后的特征代码段空间不仅要尽可能的少，稀疏编码中 0 的个数尽可能的多，同时又不损害被稀疏样本的性质，而对未知样本的处理也是先对其进行稀疏化处理，生成若干可以代表样本性质的空间样本集，最后根据这个空间样本集与先前训练样本集产生的特征代码段集合做距离对比，根据最近距离原则定义其性质。

4.1.2 数据集的处理

训练样本主要是从网上下载的各种恶意代码样本，白代码样本是从 Windows XP 系统中随机抽取的可执行代码文件，其数目为 200 个，定义 180 个为训练样本，20 个为测试样本。主要是对三类样本进行了试验，一是变种的恶意代码样本，二是网上下载各类木马样本，三是无关联的恶意代码样本。每组训练样本恶意代码样本为 100 个，正常代码样本也为 100 个，分别抽取 90 个作为训练样本，20 个作为测试样本。

上面的所有的训练和测试样本都是独立且不重复的，每一个训练样本都带有恶意或正常代码的性质标签。

4.1.3 特征提取

对于任意的恶意代码来说，为了达到其目种目的，其程序段必然要有完成自己功能和部分，而那也就是它与正常代码不同的特征部分。特征提取就是从正常代码和恶意代码的众多程序段中提取出能够和对方相互区分的特征段，其实质就是提取出可以将程序定义为恶意，或者是正常的某段可执行的程序机器代码结构。现在大多数的检测都是通过对二进制代码段的有效提取来实现能够代表程序性质的机器代码结构的发现。

由于恶意代码是二进制流文本，如果仅仅是对二进制流文本进行定长或变长载取，不能很好的代表其规律和性质。而且目前关于恶意代码的许多研究表明恶意代码的特征大多数是和指令有关的。现在国内外有很多论文中都是采取一定策略从恶意代码中只对指令字节流进行处理，以指令信息流中提取出有序指令特征。本文中指令特征段的提取主要是通过对恶意代码进行反汇编，从二进制流文本中提取一定的指令长度文本信息、相当于进行“分词”处理，最后将结果保存到预先定义的特征向量集中，以供分类器训练和分类使用。

经过反汇编处理后我们得到如下所示的机器代码结构。

558b ec6a ff68 fd21 4500 64a1 0000 0000 5051 a190 ec46 0033
c550 8d45 f464 a300 0000 0089 4df0 894d f08b 4df0 e872 1100
00c7 45fc 0000 0000 8b4d f083 c120 e897 1100 00c6 45fc 016a
018b 4df0 83c1 38e8 5227 0000 c645 fc02 6a01 8b55 f052 8b4d
f0…

这里的恶意代码的特征段提取，是分类器训练和代码分类的重要前提。相当于分类前的预处理，主要完成了恶意代码的反汇编和文本的指令级特征向量“分词”两个功能。

但是在对程序进行定长字节序列截取时还有一个重要的问题，多长才能更好的代表机器结构，才能更好的提取出能够代表恶意或者正常代码的特征段？本文主要采取基于滑动窗口的调节来实现。如果窗口值定义为 1 时，首次提取“558b”为特征，依次再取“ec6a”、“ff68” …，直至文件未尾为止，最后不足时用 0 补足。窗口值为 2 时，首次提取“558bec6a”、“ff68fd21” …，把“558b”和“ec6a”中间的空格去掉，组成一个新的更长的字节序列。窗口值为 3 时则提取三个连续有序字节合成的新的更长字节。

4.1.4 分类和判别实现

恶意代码检测系统中的关键部分是，完成分类训练和未知性质的判别功能。本文中主要应用码理论具体实现。具体实现分为两个部分，一是用改进的稀疏编码算法进行分类和判别实验；二是结合 LBG 算法和稀疏编码算法提出了基于相似

度算法，进行具体分类和判别实现。

本文的主要目的就是这两种方法的具体实现，下面分别从这两个方面进行码表理论应用在恶意代码检测系统中的实验。

4.2 稀疏编码算法的模型

从根本上来说，基于码书理论的稀疏编码算法设计的恶意代码检测系统，总共被分为三个模块：数据样本预处理模块、分类器训练模块和实例性质分类模块。

第一步数据样本预处理模块。主要做的是先进行反汇编，再进行有序的特征字节流“分词”。分类器训练模块的处理，则主要从个体角度来对每一个样本进行稀疏化处理，以找到能够代表该样本性质的若干特征串集合，然后针对所有的训练样本根据各自的特征串集合进行空间距离计算。分别找到恶意代码样本和非恶意代码样本空间的中心，通过类别限定使本类特征串距离本类样本距离最近，而同时距离另一类别距离尽可能的远。实例性质分类模块实现了新实例样本的的特征向量化和特征项提取，最后对特征向量进行与已经计算出的两个样本中心做距离计算对比，其性质定义为和离其距离最近的样本中心所代表的那个类别。

改进的稀疏编码算法示意图如 4-1 所示：

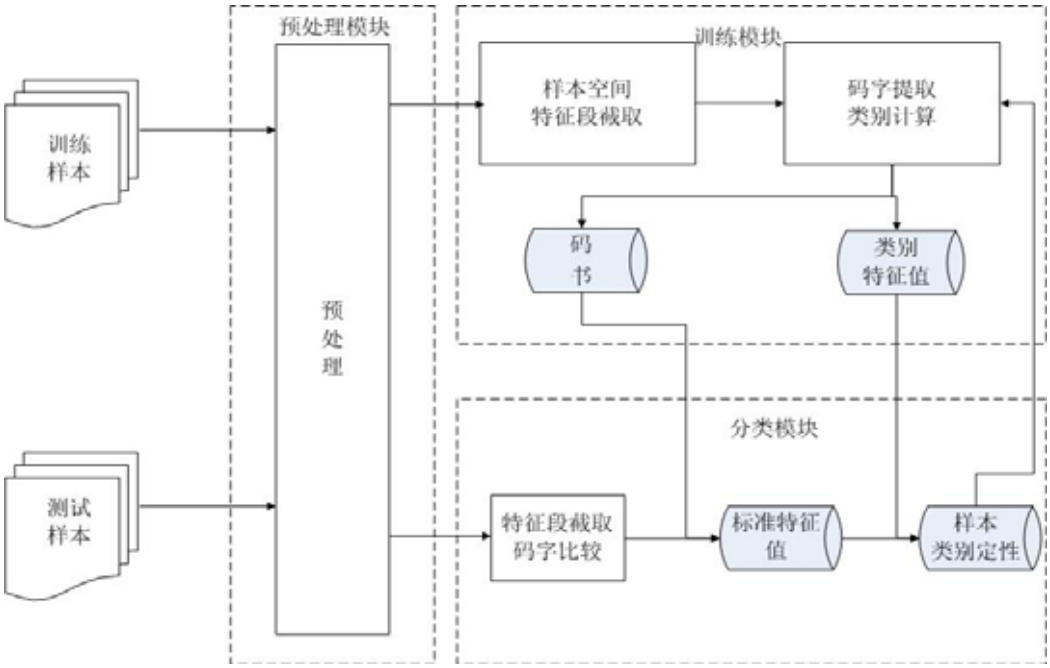

图 4-1 稀疏编码算法检测模型

4.2.1 预处理

相当于 LBG 算法稀疏编码算法中的预处理更注重个体样本的处理。其主要的

区别是生成若干个向量集，就是说每一个训练样本生成一个自己独立的由训练样本本身经过定长截取后的特征段组成的向量集。

4.2.2 分类器训练模块

在稀疏编码算法中，对每一个训练样本进行稀疏化处理。

本模块的算法如下：

初始条件：先给码书定义一个初始的尺寸N，每个训练样本X经预处理后得到的向量集合：$X=\{x_1, x_2, \ldots, x_n\}$；设算法终止条件、停止门限ε，且 $0<\varepsilon<1$；设平均初始失真$D_1=\infty$，迭代次数初始值u=0。

(1)初始化字典D，随机抽取k个x_i，定义为初始码字，如果都是要求的质心则算法中止；

(2)如果非都是质心，先根据对集合中所有的其它向量分别找出离它们最近的初始码字，并且和初始码字一起组成一个初始集合，如果集合能均匀分布，则转到(3)，否则转到(4)；

(3)将样本中最小的集合和与之距离最近的集合合并，并将最大的集合平均分裂成两个集合，直至 N 个集合能均匀分布；

(3)对每一个初始码字样本集合，分别计算一个集合中的元素与其它所有元素的距离（欧几里德距离），根据计算公式：$d_2(x, c_i) = min\{d_2(x, c_j) / j \neq i\}$ $(d_2(x, c_j)$，通过计算求得每个集合的中心点；

(4)此时的所有中心点的集合定义为初始新码书$C_0=\{c_1,c_2,\cdots,c_n\}$；

(5)平均失真和相对失真的计算：平均失真$Dv=1/n\sum_{i=1}^{n}\min\{d(x_i,c_j)\}$，然后再计算相对失真值：$Q_k=(D_{(v-1)}- D_v)/ (D_{v-1})$，将求得的$Q_k$与停止门限ε做比较，如果$Q_k \leqq \varepsilon$则将初始新码书定义为最终码书，否则转(6)；

(6)重新选定一个新的码书尺寸 N*，并转向(1)。

4.2.3 分类模块

在稀疏编码算法中，也先对测试样本进行稀疏处理。会得到和原先已定义的码书大小相同的特征序列集，并把集合按从小到大的顺序排成有序集合。根据类内距定义求出测试样本的中心点。根据类间距定义，用所求得的中心点分别与那些已经知道性质的类别样本中心进行逐一距离比较，从这所有距离当中求出最小距离。测试后是符号猜想：也就是说测试样本的性质将被被定义为，所求测试样本中心点与分类器所有类别样本中心点的距离中最近的那个，离恶意代码的中心点近就将测试样本认定为恶意代码，离正常代码中心点距离近则定义为正正常代码。

在本文中也采用增量学习的方式，如果通过分类器决定未知样本的性质后，则

将此样本的特征向量集返回原分类器中进行重新计算新的加权值，重新进行概率和归类的计算。

4.2.4 实验数据及结果分析

实现根据样本选取的不同性质，总共分为两大部分。分别对任意恶意代码样本、木马（以盗号木马样本为主）样本，这两类样本进行实验。实验中正常性质样本为从 Windows XP 系统中得到的，干净的 200 个样本。其它的恶意代码样本为从网络上下截得到。

实验中从网络上下截任意恶意代码样本、木马（以盗号木马样本为主）样本，这两类样本个数分别为 500 个，其中 450 个作为训练样本，50 个作为测试样本。

在这两类实验当中，每个类别都分为 5 组进行实验，每组 100 个恶意代码样本，90 个作为训练样本，10 个作为测试样本。每组选 100 个正常代码样本。

（1）试验中，把下载的恶意代码样本分成5组，每组100个恶意代码样本，90个作为训练样本，10个作为测试样本，试验结果平均值如表4-1。

4-1 恶意代码样本

序号	码书大小	特征长度/bit	运行时间/s	准确率/%
1	32	16	628	70
2	32	32	610	90
3	32	48	588	89
4	64	16	608	72
5	64	32	582	91
6	64	48	560	87
7	128	16	580	74
8	128	32	566	85
9	128	48	540	81

在表4-1中第5次实验码书大小为64，特征长度为32时准确率最高，其它的情况下大都是在特征长度为32时候准确率很高，在特征长度为16时候准确率最差，试验中平均误报率只为8%。

（2）试验中，把下载的木马样本分成5组。每组100个木马样本，90个作为训练样本，10个作为测试样本，试验结果平均值如表4-2。

在表4-2中同样是在码书大小为64，特征长度为32时准确率最高，在特征长度为16时候准确率最差，试验中平均误报率为7%。

表4-2 木马训练样本

序号	码书大小	特征长度/bit	运行时间/s	准确率/%
1	32	16	620	87
2	32	32	602	89
3	32	48	560	89
4	64	16	600	90
5	64	32	580	93
6	64	48	558	91
7	128	16	602	88
8	128	32	598	86
9	128	48	520	87

从上面两个表中的数据可以看出对于稀疏编码算法下的码表理论算法无论是变种木马还是恶意代码都能达到90%的准确性，误报率不高，变化不大。因为算法主是对每个单一样本进行稀疏化，生成能代表样本特征的那些特征串，不同于整体算法中的整体稀疏化。但同时也因为基本上保持了样本的各自特征对于无类别的恶意代码的检测效果还是有一定作用。

4.3 相似度算法的模型

相似度算法示意图如 4-2 所示，总体来说，基于 LBG 算法计算相似度，设计的恶意代码检测系统。总共被分为三个模块：数据样本预处理模块、分类器训练模块和实例性质分类模块。第一步数据样本预处理模块，主要做法是实现对数据样本的反汇编、样本中代码数据特征向量集中的特征向量“分词”；分类器训练模块的工作主要是实现了特征向量权重、频率的统计、特征向量代表的选取；实例性质分类模块实现了新实例样本的的特征向量化、特征项提取、实例样本类别评分并分类的功能。

4.3.1 预处理

预处理的主要功能就是对无论训练样本还是测试样本都要进行的合理的程序切断处理。

预处理在考虑整体的相似度算法中，主要的过程是，首先将每个样本转化为指令集的形式，把每一个指令当作一个“分词”起始部分。然后再根据指令的原指令序列顺序逐步增加“分词”的长度，每增加一个位数，进行一次长度的比较，如果长度不够则继续增加“分词”的长度，否则将其定义为分词，并存入到

相应集合之中。经指令“分词”处理过后的指令信息流以有序指令特征段的形式保存在一个向量集合中。对训练集合的后续处理，用所有每个实例样本的向量集生成一个新的包含所有被“分词”向量的集合，等待后序模块进行处理。而对于测试集则只保留其生成的向量集，等待后序模块进行处理。

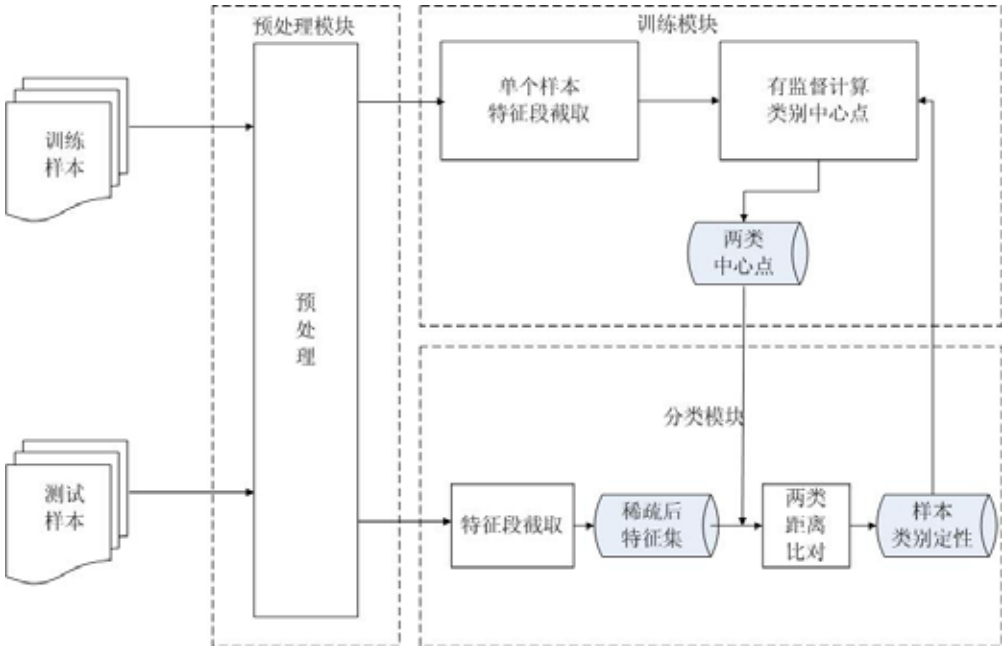

图 4-2 LBG 算法检测模型

4.3.2 分类器训练模块

指令经过预处理后得到了各自的向量集。都存放大量不同的特征向量，如果将每个特征向量都作为样本的特征标志进行类别归类约束的话，也就是说稀疏编码中的非零系数为 0，那无论是计算还是存储都将会非常庞大，而且用大量不同的特征向量对代码类别做分类处理也是不现实的，无法应用在正常的代码检测当中。因此需要通过一定特征选择计算来筛选有代表性的特征向量，实现特征向量的稀疏处理。

在相似度算法中特征码字的选取，是构造一个评价函数，对所有特征向量集的每个特征进行综合评估。本模块采用了基于最小距离失真（欧式距离）来对每段特征进行评估，以决定该段特征选取哪一个作为用来分类的特征项，就是查找特征段中的最佳代表。特征向量经过处理后，向量集就被一个一个从训练样本向量集中选取的特征向量代表所替代，也就是一个一个的码字。分类器会根据已知的类别将特征项代表，也就是码字保存到指定的码书中，本文中分别形成了代表正常代码的码书和代表恶意代码的码书。

所以如选取特征向量代表，是LBG算法的核内容，是决定算法能否收敛和能

否收敛于较为准确的位置。这在图像处理中被称为初始码书的选择，常用的方法有随机编码法、成对最邻近法、原始二分裂法等等。

本文采用了二进制位分裂法完成对特征向量代表的选取。这里将每个特征向量看成是坐标轴上的点的集合，先将坐标轴划分成左右二个部分：0，1。一个点的坐标的最高位决定了它的所处坐标轴的部分。然后再把每个部分分成左右二个部分，最后检查每个码字所包含的特征向量，将可能出现的空的码字与它最近的码字合并。如些循环，直到达到所要求的码书数目 N。这时将特征集分裂成达到码书要求的个数的特征集合，并按规律排列。这当中还要计算其每个码字在整体作用中的比重。该方法是一种统计方法，用于评估一个特征向量代表对特征向量集或一个“分词”库中其它特征向量对代表的聚合程度和作用比重。聚合性就是它到其它特征向量的距离值的和最小，与出现的频率成正比。

4.3.3 分类模块

对于相似度算法而言。先对每个检验样本通过上述方式生成输入矢量集合，针对所有的输入矢量分别与码书对照求出其可能的失真和，并和已知的失真和的范围作比较，如果值在范围内则将认定为和是原样本是同一类别是可容忍的压缩，其性质和原样本相同，否则相反。通过对大量训练样本的“分词”向量进行分析，可以认为恶意代码中的“分词”向量符合一定的概率模型。某些“分词”向量在已知的恶意代码样本中出现的概率比较大，而另外一些“分词”向量在正常代码中出现的概率比较大。因此，对于一个未知性质的代码，对其中的每个“分词”向量可以求出其在恶意代码样本中出现的概率，也就是未知性质的代码根据这个“分词”可以判别为是恶意代码的概率。通过加权求和可以得到一个分数，也就是把未知性质的代码判决为恶意代码得到的分数。同理，也可以得到把代码判决为正常代码的分数。通过求差或求积就可以得到这个代码的最终得分，然后再利用一定的阈值就可以得到判决结果。

在本文中采用增量学习的方式，如果通过分类器决定未知样本的性质后，则将此样本的特征向量集返回原分类器中进行重新计算新的加权值，重新进行概率和归类的计算。

4.3.4 实验数据及结果分析

实现根据样本选取的不同性质，总共分为三大部分。分别对任意恶意代码样本、木马（盗号木马为主）样本，蠕虫病毒样本，这三类样本进行实验。实验中正常性质样本是从 Windows XP 系统中得到的，干净的 200 个样本。其它的恶意代码样本为从网络上下截得到。

实验中从网络上下截任意恶意代码样本、木马（盗号木马为主）样本，蠕虫

病毒样本三类样本个数分别为 500 个，其中 450 个作为训练样本，50 个作为测试样本。在这三类实验当中，每个类别都分为 5 组进行实验，每组 100 个恶意代码样本，90 个作为训练样本，10 个作为测试样本。每组选 100 个正常代码样本。

（1）任意恶意代码样本 500 个，分成 5 组，每组 100 个恶意代码样本，90 个作为训练样本，10 个作为测试样本。试验结果求平均后值如表 4-3。

表4-3 恶意代码样本

序号	码书大小	特征长度/bit	运行时间/s	准确率/%
1	32	16	328	58
2	32	32	286	82
3	32	48	260	78
4	64	16	310	62
5	64	32	274	85
6	64	48	262	81
7	128	16	282	59
8	128	32	264	75
9	128	48	245	70

（2）木马（盗号木马为主）样本 500 个，分成 5 组，每组 100 个木马样本，90 个作为训练样本，10 个作为测试样本。试验结果求平均后值如表 4-4。

表 4-4 木马样本

序号	码书大小	特征长度/bit	运行时间/s	准确率/%
1	32	16	288	64
2	32	32	280	92
3	32	48	278	91
4	64	16	278	65
5	64	32	270	93
6	64	48	264	92
7	128	16	258	66
8	128	32	256	91
9	128	48	248	91

（3）蠕虫病毒样本500个，分成5组，每组100个木马样本，90个作为训练样本，10个作为测试样本。试验结果求平均后值如表4-5。

表4-5 变种样本

序号	码书大小	特征长度/bit	运行时间/s	准确率/%
1	32	16	216	69
2	32	32	204	96
3	32	48	200	97
4	64	16	188	71
5	64	32	182	97
6	64	48	178	95
7	128	16	170	68
8	128	32	164	97
9	128	48	152	96

通过这两种算法得到的结果对比可以看出稀疏编码算法适用范围较广，但是准确性没有使用相似度算法针对类别性明显的恶意代码的查杀准确性高，且算法效率较高。但是算法对恶意代码的检测基本上实现的原来的目的，码表理论应用于恶意代码查杀方面是可行的而且有效的。

现在的很多论文统计中表明，随着产业化、经济化、快速化的趋势，恶意代码的产业要求其不断的快速的更新，技术发展的相对落后则满足不了这种需求。因此新出现的恶意代码有一大部分是在已经出现的恶意代码的基础上，通过一定的方法稍加改变或变种衍生出来的，因为恶意代码变换前后，会呈现出不完全相同的特征码，之后按传统特征码的检测方法效果不是很理想。而在从整体考虑的码书理论的恶意代码检测方法，而本文的方法在实验中则能起到很好的效果。

4.4 本章小结

本章首先将通过反汇编编译处理得到的指令代码，按一定的规律顺序排列成有序的集合形式，再用稀疏编码原理将数据集进行稀疏化处理，分别得到单个的能完全代表样本特征的稀疏特征集合和能表示整个类别样本特征的稀疏数据集。又分别利用分类算法中的距离分类和相似度分类方法把样本分成性质完全不同的两类，一类代表恶意代码，一类代表正常代码。

本文中根据类别分别进行了实验，每个类别进行了五组实验。根据平均实验结果的分析我们发现改进的稀疏编码检测方法的时间复杂度和平均检测效果有了大大的提高。有一定的发现未知恶意代码的能力。

而基于整体的相似度算法的恶意代码检测系统则对变种恶意代码具有很高的检测率，实验中误判率低于 5%，准确率更是达到 96%，但是对规律性不强的杂乱

的恶意代码集合则没有较好的效果。

结 论

把码表理论的算法引入到恶意代码检测领域，是为了改进目前的静态特征码的恶意代码检测技术，寻找到一种新的理论方向，具有更好的检测效果和更快的速度和处理效率。结合 LBG 算法和稀疏编码算法的基于相似度的恶意代码检测方法技术，利用“整体相似度”原理，在对那些具有相似规律的某类恶意代码的查杀效果很好。

本文深入研究了机器学习在恶意代码检测领域的可行性，对比码表理论在图像处理方面的应用。码表理论在视频处理的包括行人分类等方面得到了有效的应用，具有很好的两分类性。特别是对具有代表性的 LBG 算法和目前的研究热点稀疏编码算法，做了更为详细的理论研究和分析，并结合恶意代码检测理论进行了相关理论分析。主要工作有以下几个方面：

（1）本方通过比对机器学习和码表理论算法中的分类常用方法，以及两者在各自应用中的良好的分类特性，介绍了这两者在分类性方面可能具有的相通性，说明了码表理论应用于恶意代码检测中理论上有可行性的。

（2）本文通过码表理论的 LBG 算法和稀疏编码算法，设计出应用于恶意代码检测系统中的稀疏编码算法和相似度算法。

（3）稀疏编码算法恶意代码检测中有强的可操作性，相似度算法虽然准确率较高，但是存在很强的类别局限性。

（4）算法还需要进一步改进，以在不影响检测效果的前提上，进一步改进时间复杂度和空间复杂度。结合随机选取和二分裂法设计基于二者结合的高效率算法，如整体采取二分裂，每个区间采取随机选取的办法。

参考文献

[1] 国家互联网中心(CNNIC)[EB]. 第 26 次中国互联网络发展状况调查统计报告，2011，4.

[2] 国家互联网应急中心(CNCERT)[EB]. 2010 年网络安全信息安全调查报告，2010，12.

[3] Tony About-Assaleh, Nick Cercone, Vlado Keyselj. N-gram-based Detection of New Malicious Code[J]. Computer software and Applications Conference, 2004：2.

[4] 秦志光，张凤荔. 计算机病毒原理与防范[M]. 北京人民邮电出版社，2007：17.

[5] 百度百科[OL]. URL：http://baike.baidu.com/view/1088942.htm.

[6] 百度百科[OL]. URL：http://baike.baidu.com/view/52956.htm.

[7] L.Garber.Melissa.Virus Creats a New Type of Threat of Threat[J]. IEEE Computer, 1999, 32(6)：16-19.

[8] Michal Erbschloe. Computer EconomicsCompany[OL]. URL：http://www.computer economics.com/.

[9] 刘涛，邓璐娟，丁孟宝. 计算机反病毒技术及预防新对策[J]. 计算机技术与发展，2007，17(5)：104-106.

[10] Dorothy E Denning.An intrusion detection model[J]. IEEE Symp on Saecurity and privacy.Oakland, califormia, 1986：118-131.

[11] Mattew G.Schultz, Eleazar Eskin. Erez Zadok.Data Mining Methods for Detection of New Malicious Execatables[J]. IEEE Computer Society, 2001：38-49.

[12] Mihai Christodorescu, Somesh Jha.Static Analysis of Executables to Detect Malicious Patterns[C]. Proc.of the 12th USENIX Security Symp, 2003：169-186.

[13] 王海峰，段友祥，刘仁宁. 基于行为分析的病毒检测引擎的改良研究[J]. 计算机应用，2004，24(2)：109-110.

[14] 王晓洁. 蠕虫病毒特征码自动提取原理与设计[J]. 微计算机信息，2007：3.

[15] 李焕洲，陈婧婧. 基于行为特征库的木马检测模型设计[J]. 四川大学学报，2011：123.

[16] Fred Cohen. On theimplications of Computer Viruses and Methods of Defense[M]. Computers & Security , 1988：167.

[17] 王海峰，段友祥. 基于行为分析的病毒检测引擎的改良研究[J]. 计算机应用. 2004，12:146.

[18] I.Goidberg, D.Wagner, R.Thomas, E.A.Brewer. A secure environment for untrustedhelper applications：Connecting the wihey hacker[C]. Proceedings of the

1996 Usenix Security Symposium. USENIX, 1996：22-25.

[19] T Mitchel. 机器学习[M]. 机械工业出版社，2003：38-59.

[20] JACOB G, DEBAR H, FILIOLE. Behavioral detection of malware：From a survey towards an estabhshed taxonomy[J]. Joumal in Computer Virology, 2008，4(3)：251-266.

[21] 李仕静，梁知音，韦韬，等. 一种基于语义的恶意行为分析方法[J]. 北京大学学报：自然科学版，2008，44(4)：537-542.

[22] Linde, Y.Buzo, A.Gray, R.M. An Algorithm for Vector Quantizer Design[J]. IEEE Transactions on Communications, 1980(28)：84-94.

[23] 张健宏. 人工智能中的机器学习研究及其应用[J]. 江西科技师范学院学报，2004（5）：84-86.

[24] 张雪英. 基于机器学习的文本自动分类研究进展[J]. 情报学报，2006，25(6)：730-739.

[25] 贾慧星，章毓晋. 车辆辅助驾驶系统中基于计算机视觉的行人检测研究综述[J]. 自动化学报，2007，33(1) ：84-90.

[26] 朱文佳. 基于机器学习的行人检测关键技术研究[D]. 上海交通大学， 2008：26.

[27] Gray R M. Vector quantization[J]. IEEE A coust Speech Signal ProcessingMag, 1984 (1)：42.

[28] David S. Date compression, the complete reference[M]. BEIJING：Edition by Publishing House of Electronics Industry, 2003：71-77.

[29] 刘丽娟， 沈绪榜. 图像压缩中一种改进的快速编码方法[J]. 华中科技大学学报，2003，(7)：10-12.

[30] Yukinori S, Takayuk i M. V ector quantization by a self organizing tree with new lyimp lemented pruning algorithm[J]. IEEE International Midwest Sympo sium on Circuits and Systems, 2004：47.

[31] Hugh Q C, L iW eip ing. A fast search algo rithm for vector quantization using a directed graph [J]. IEEE T ransactions on Circuits and Systems for Video Techno logy, 2000, 10 (4)：48-51.

[32] 匡泰，胡众义等. 一种新的用于 LBG 图像压缩初始码书生成算法[J].浙江工业大学学报，1988，36(8)：4.

[33] Michison G. The Organization of sequential Memory：sparse Repres-entations and the Targetiong Problem[J]. 0rganiztion of Neural Networks, VCH VerlagsgeseIlschaft Weinheim, 1988：347-367.

[34] B. A. Olshausen. Sparse coding of time-varying natural images[J]. Vision of Vision, 2002：130.

[35] B. A. Olshausen, D. J. Field. Sparse coding with an overcomplete basis set[J]. A strategy employed by V1 Vision Research, 1997：37.

[36] B. A. Olshausen, Field D J. Emergence of Simple cell Receptive Field Properties by Learning a Sparse Code for Natural images[J]. Nature, 1996, 381：607-609.

[37] 吴金波，蒋烈辉. 反静态反汇编技术研究[J]. 计算机应用，2005， 25(3)：18.

[38] CULLEN LINN, SAUMYA DEBRAY. Ob fuscation of Execu table Code to Improve Resistance to Static Disassem bly[C]. In Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS), 2003：290 -299.

[39] 吴金波，蒋烈辉，赵鹏. 基于控制流的静态反汇编算法研究[J]. 计算机工程与应用，2005，41(30)：21.

[40] Schultz M G, Eskin E, Zadok E, et al. Data Mining Methods for Detection of New Malicious Executables[C]. Proc. of the IEEE Symposium on Security and Privacy, 2001：38-49.

[41] Assaleh T A, Cercone N, Keselj V, et al. Detection of New Malicious Code Using N-grams Signatures[C]. Proc. of the and Annual Conference on Privacy, Security and Trust. Ontario, Canada, 2004：193-196.

[42] Cohen P, Heeringa B, Adams N M. An Unsupervised Algorithm for Segmenting Categorical Time Series into Episodes[C]. Proc. of the ESF Exploratory Work-shop on Pattern Detection and Discovery. London, 2002：49-62.

[43] 张茜，刘磊等. 一种基于最小距离分类器的恶意代码检测方法[J]. 广西师范大学学报：自然科学版，2009，27(3)：183-187.

[44] 王硕，周激流，彭博. 基于 API 序列分析和支持向量机的未知病毒检测[J]. 计算机应用，2007，27(8)：1942-1943.

哈尔滨工业大学学位论文原创性声明及使用授权说明

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于码表理论的恶意代码检测技术研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签名：　　日期：2011 年 6 月 29 日

学位论文使用授权说明

本人完全了解哈尔滨工业大学关于保存、使用学位论文的规定，即：

（1）已获学位的研究生必须按学校规定提交学位论文；（2）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（3）为教学和科研目的，学校可以将学位论文作为资料在图书馆及校园网上提供目录检索与阅览服务；（4）根据相关要求，向国家图书馆报送学位论文。

保密论文在解密后遵守此规定。

本人保证遵守上述规定。

作者签名：　　日期：2011 年 6 月 29 日

导师签名：　　日期：2011 年 6 月 29 日

致 谢

这次毕业设计能够最终完成，得益于给予我多方面无私而热情的帮助的老师、同学和朋友们。值此论文完成之际，谨向所有给予我关心、支持和帮助的人表示衷心的感谢。

首先，感谢实验室主任张宏莉教授。张老师严谨的学术态度、渊博的学识是我做研究的榜样。

感谢翟健宏老师和刘亚维老师，研究方向的确定、项目的开发以及毕业论文的完成，每一步都在与两位老师的谈话中，受到了很大启发。在两位老师的指导与启迪下，我开始懂得应该怎样做正确的研究。在课题上他们悉心的指导，热情的帮助，使我在毕业设计中受益匪浅，为我课题的完成提供了巨大的支持。

感谢实验室的李东老师、张伟哲老师、余翔湛老师、何慧老师，他们机智的学术敏锐性、认真负责的工作热情、学术上创新进取的精神以及平易近人的生活作风都深深的感染了我。

感谢实验室其他老师和同学们，这是一个团结的集体，这是一个和睦的家庭，这是一个互相帮助的中心。它有完备的硬件设施，良好的学习环境，积极向上的学习氛围，都让我难以忘怀。

感谢所有曾经帮助我的人，感谢我的家人。

最后感谢我的母校哈工大，感谢计算机学院。这段生活是我生命中值的铭记的片段。

祝母校的明天会更好。