暨南大学硕士学位论文

题名（中英对照）：

机器学习算法对中国 A 股的适应性比较

作者姓名：　　**谢翔**

指导教师姓名：**姜云卢**
及学位、职称：**副教授**

学科、专业名称：应用统计

学位类型：（专业学位）

论文提交日期：

论文答辩日期：

答辩委员会主席：

论文评阅人：

学位授予单位和日期：

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得暨南大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：　　签字日期：2017 年　月　日

学位论文版权使用授权书

本学位论文作者完全了解暨南大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权暨南大学可以将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、汇编学位论文。
（保密的学位论文在解密后适用本授权书）

学位论文作者签名：　　导师签名：

签字日期：　年　月　日　　签字日期：　年　月　日

学位论文作者毕业后去向：

工作单位：　　电话：

通讯地址：　　邮编：

摘要

在投资领域中，先后出现过诸多派别，有以基本面分析为主的，以巴菲特为特例。有以技术分析为主的，以约翰.墨菲为特例。近年来，随着计算机技术的不断发展，欧美市场早已出现了一股量化投资风潮，并且不断壮大，该投资派别以詹姆斯.西蒙斯为代表。量化投资以其稳定、理性、概率取胜为优势，占领着投资领域的主流风潮。

机器学习是数据挖掘领域的前沿技术，机器学习诸多算法以其严密的逻辑对训练数据进行特征学习，再将学习到的经验特征用于预测。而在投资领域，成功预测未来走势是投资成功的不二保障。

已有诸多学者尝试过将机器学习各大算法运用于量化投资领域，有的取得了不菲的成效，也有投资者因为方式选择不当而导致投资失败。失败的原因主要在于不同的算法对于不同的数据具有不同的适应性。而证券市场的数据是千变万化的，不同走势，不同品种，都将产生具有不同特征的数据。本文将试图对七种机器学习算法在中国 A 股市场的效果作一个比较，结果显示，对于次日涨跌预测，上涨趋势当中，SVM 算法最优，下降趋势当中，随机森林算法最优，盘整走势当中，各大算法表现平平；对于次日高低开预测，上涨趋势当中，决策树算法最优，下降趋势当中，神经网络算法最优，盘整走势当中，逻辑回归算法与 SVM 算法最优；对于不同品种次日涨跌预测，在工业股与农业股中，朴素贝叶斯算法最优，对于服务业股，逻辑回归算法最优，对于高科技产业股，支持向量机算法最优；对于不同品种次日高低开预测，逻辑回归算法几乎都是表现最优，只有在高科技产业股中 SVM 算法表现略好于逻辑回归算法。

关键词：量化投资；机器学习；走势类型；投资品种；适应性。

Abstract

In the field of investment, there are many kinds of investment styles, some people rely on basic analysis, suck as Buffett. Some of them rely on technical analysis, Such as John Murphy. In recent years, with the continuous development of computer technology, the European market has already appeared in a quantitative investment trend, and continues to grow, The faction is represented by James Simmons. Quantitative investment win with the advantages of stable, rational and probability, it occupy the main stream of investment.

Machine learning is the cutting-edge technology in the field of data mining, many machine learning algorithm with its rigorous logic of training data for feature learning, then learn to predict the characteristics of experience. In the area of investment, successfully predicted the future trend is the best guarantee of successful investment.

Many scholars have tried various machine learning algorithm to the field of quantitative investment, some have achieved success, but also failure because of inappropriate investment. The main reason for the failure is that different algorithms have different adaptability to different data. While the stock market data full of myriads changes, different period and different varieties will produce different kinds of data. This paper tries to make a comparison of seven kinds of machine learning algorithms in Chinese A share market,results show that for the price forecast,SVM wins in rising trend.Random forest wins in downward trend,all algorithm perform common in consolidation trend.For the next level of opening forecast,decision tree wins in rising trend,neural network wins in downward trend,logistic regression wins in consolidation trend.For different varieties of the price forecast,Naive Bayesian wins in the industrial and agricultural stocks.Logistic regression wins in service sector,support vector machine wins in high-tech industry stocks.For the next day's prediction of different varieties,the logistic regression algorithm is almost optimal,only in the high-tech industry stock, the SVM algorithm performs better than

the logistic regression algorithm.

Keywords: Quantitative investment; machine learning; trend type; Investment varieties; adaptability.

目录

1 绪论

1.1 研究背景与意义

近年来，量化投资已在欧美市场迅猛发展，一度成为世界投资潮流的主旋律。与基本面分析、技术面分析共同成为投资界的三大方法。

量化投资以其稳定、理性、概率取胜为优势，不但越来越受到机构投资者的关注，个人投资者亦相青睐。

过去的 20 年里，收益最高的基金非量化投资创始人詹姆斯.西蒙斯管理的文艺复兴科技公司旗下的大奖章基金莫属。其客户的平均年收益率居然高达 35%；高盛公司亦设有量化基金，过去 4 年里，其量化基金规模翻了近一倍，超过 1000 亿美元。量化投资在现实中的突出表现，已成为国内外诸多学者的关注点。

有学者曾提出过这样一种假设，假设我们抛出一枚硬币，正反出现的概率各为 0.5，因此无论抛出多少次，正反之间的出现也不会有显著的区别。但如果有办法使得正面出现的概率大于 0.5，即使使得正面出现的概率稳定为 0.51，反面出现的概率稳定为 0.49，理论上只要有足够多抛硬币的次数，随着次数的不断增加，将会使得正面出现的次数显著高于反面的次数。量化投资的原理亦跟此类似，量化投资虽然不能够保证每次投资都成功，但量化投资会根据大量历史数据计算出一个投资成功率，只要投资成功率在大量数据验证下能够稳定高于 0.5，那么我们称该量化投资策略是有效的。有效的策略便能够通过交易次数的增加来取得概率上的胜利。

量化投资虽然有惊人的市场表现，然而要实现起来并且有良好的市场表现却绝非易事。市场上先后出现过诸多量化投资策略，有以基本面为依托的量化选股模型，如多因子选股模型；亦有以技术面为依托的量化择时策略，以 KDJ 及 MACD 等技术指标为判断依据而设计量化投资程序；亦有以统计学方法为判断依据的统计套利策略及配对交易策略；更有学者设计出以江恩理论、易经等玄学作为依托的自动化交易系统；然而用到最多最广泛的，被诸多机构投资者青睐的，非机器学习算法不可。

机器学习算法最早产生于人工智能。人们希望电脑具有如人脑一般学习知识

的能力。并且根据学习到的知识用于指导实践，名之曰让电脑具有知识泛化能力。而在投资市场上，具有先见之明是投资成功的不二保障。虽然市场的表现是千变万化的，但却在某种程度上会呈现一定的规律性。将机器学习算法运用于量化投资领域，有时能够很好地做出正确指导。

但机器学习领域有诸多算法，每种算法都有其局部适应性。将机器学习算法运用于量化投资并非一劳永逸的事情。因此我们有必要对每种算法的适应性作研究，以期在正确的时间正确的品种上选择合适的算法。

本文的研究相对于以往学者所做研究的不同之处以及研究意义主要在于：

以往学者的研究主要集中于某种单一机器学习算法运用于量化投资当中的实际表现，本文并非注重单一机器学习算法在量化投资当中的表现情况如何，本文是对于诸多机器学习算法都做概述，研究各类不同机器学习算法的特征，以及该种特征下所能对应的恰当投资周期与投资品种，比较出各类机器学习算法在量化投资当中的具体表现。以往学者运用单一模型进行量化投资程序的设计，其设计出来的程序会具有局部适应性。这也是人们对于量化投资的看法褒贬不一的原因之一。本文将每一种机器学习算法都编写成量化投资程序，而将重点放在研究算法适应性上。没有任何一种算法是绝对的好，或者绝对的坏，问题的关键在于是否在恰当的条件下选取恰当的方法。以本文的结论作为量化投资的参考，在适当时机选取适当算法，以避免方法选择不当所造成的失误，是本文最希望达到的目的。

1.2 国外研究现状

Irwin[1]（1986）等诸位学者早年的时候就着手建立自动化交易程序，他们编写出自动化交易程序，对期货市场实行自动化交易，结果显示，他们发出的一半以上信号是有效的信号，他们所编写的程序对于期货市场的交易产生了一定程度的变革；Neftci[2]（1991）等学者将 150 日均线作为参考标准，将均线所产生的买卖信号作为输入信号进行自动化交易程序的编写，其主要对道琼斯工业指数进行分析预测，结果显示该种方法能增加交易收益的稳定性；Ritter[3]（1992）等诸多学者首次将行为金融学理论引入量化投资当中，他们的研究结果显示，在过去

一个月内表现欠佳的股票极有可能在接下来具有非常优良的表现，因此他们设计出的量化投资程序思路便是在每个月末对该月股票收益率进行排行，取收益率最后的 10%的股票进行等额的买入并且持仓一个月的时间，如此不断循环，这就是著名的反转策略；Jegadeesh[4]（1993）的量化策略思路与反转策略相反，他们认为在短期内具有较高收益率的股票在更长期限内会有更好的表现，因此他们设计出了动量策略，即选出收益率排行较高的股票进行自动化交易，并且持仓时间要超过反转策略的持仓时间。

以上是在机器学习算法流行于量化投资之前国外学者所进行的研究，随着计算机技术的不断发展，以及数据挖掘技术的不断发展，又出现了一批以机器学习算法做自动化交易的学者；Kimk[5]（2003）等学者率先运用 SVM 进行量化投资程序的编写，并且他们将该结果与神经网络算法所编写的量化交易策略进行对比，实验结果显示出较为稳定的收益率，表明 SVM 算法能够有效运用于量化投资当中；Khan[6]（2008）等学者结合神经网络算法、反向传播神经网络算法以及带有遗传算法的反向传播神经网络算法三者相结合运用于量化投资当中，他们的研究结果显示，带有遗传算法的反向传播神经网络算法能够显著提高量化投资的准确率；Nair[7]（2010）等学者将决策树算法引入到量化投资当中，他们首先利用技术分析方法对于股票价格进行特征提取，接着利用决策树算法对于提取的特征进行特征选择。他们的结果表明，利用决策树算法进行特征选择后的预测结果比单纯根据技术分析提取特征进行交易的效果要显著提高。S Jun[8]（2012）等学者则将技术分析与语音识别领域当中的 DTW 算法相结合进行量化投资程序的编写。他们的思路是历史会重演，因此利用 DTW 算法拟合历史上与当前走势相似的情形，并且根据历史走势来预测接下来的走势。该方法结果显示，无论是何种市场，都能够选择出恰当的历史走势，并且对当下走势进行有效指导。Ticknor J L[9]（2013）等学者将贝叶斯算法与神经网络算法结合，提出贝叶斯正则化神经网络算法，并且编写出相应的量化投资程序，基于该算法的量化投资程序显示，对于任何的周期，任何的品种，即使不进行数据预处理，模型也会有较为优良的表现。Takeuchi[10]（2013）等学者将机器学习算法中的前沿，深度学习算法引入量化投资当中，并且对于传统的动量模型进行改进，他们的结果显示将深度学习算法引入量化投资当中能够显著提升传统动量模型的表现。

1.3 国内研究现状

国内的量化投资领域相较于欧美国家起步较晚，但我国仍然有不少学者投入于这一块的研究，并且有大量的学者将机器学习算法引入到量化投资当中去。吴微[11]（2001）等学者早年曾将 BP 神经网络算法引入到沪深指数的预测当中去，他们认为神经网络算法对于非线性多噪音的数据会具有更好的拟合效果，他们的结果也证明了神经网络算法对于股指的预测具有实用价值。彭丽芳[12]（2006）等学者曾将支持向量机算法引入到时间序列股价预测模型当中去，对于股票收盘价拟合出回归模型进行预测，他们的研究显示，将支持向量机算法引入到时间序列模型当中，比单独用传统的时间序列模型去预测股价会有更加优良的效果。苏治[13]（2013）等学者曾将 SVR 模型运用于量化投资当中，他们利用该算法对股票收益率进行回归分析，建立了经遗传算法改进的支持向量机选股模型，将沪深两市的股票基本面信息及交易数据进行分析，从长期与短期两种周期对沪深两市股票的选股情况以及准确率进行比较，结果显示该方法在长期来看会具有更优的表现。曹正凤[14]（2014）等学者曾经将随机森林算法运用于量化投资当中，将 2012 年 1 月到 2013 年 2 月之间的 300 余只股票，总共包含了 4500 余个样本个数，利用等频算法对于样本数据进行预处理，接着利用随机森林算法实现了高精度的股票分类，该模型能够有效选取出股票当中的优质股。赵志勇[15]（2014）等学者曾将深度学习算法引入到量化投资当中去，并且利用深度学习算法当中的受限玻尔兹曼机用于量化投资程序的设计，他们利用香港证券交易所在 2001 年的数据作为测试数据，结果显示他们的模型能够良好地提取出数据特征。张炜[16]（2015）等学者曾提出一种基于神经网络算法的时间序列模型，并且将遗传算法也引入其中，该算法利用改进遗传算法对于输入变量进行降维处理，该算法的最大好处就是能够显著提高神经网络算法的效率，并且增加模型预测准确率。

从国内外学者对于该模块的研究可以发现，欧美国家的学者早在机器学习算法用于量化投资之前便已开始运用许多其他方法进行量化投资，随后过渡到将机器学习算法引入到量化投资的研究当中。当我国亦兴起量化投资领域时，机器学习算法已经被广泛运用于量化投资领域，因此我国较早从事量化投资研究的学者便同时开始了机器学习算法的研究。

2. 量化投资与机器学习的关系

2.1 量化投资简述

量化投资，简单说来，便是利用统计学、信息学以及数学等相关学科，对投资的对象进行分析，提取交易信息，作出合理投资计划的方法[17]。量化投资成功的关键因素在于能否从纷繁复杂的宏观经济数据、公司财务报表、品种交易数据、政府政策、以及公开的即时市场消息当中，提取出有效的交易信息。

因此，量化投资必定需要借助编程技术，利用计算机方能实现。从狭义的角度来说，量化投资便是指让计算机实现自动化提取市场信息[18]，自动化交易，并且能够实现稳定盈利的过程。

目前，量化投资在全球范围内都得到了充分的认可。美国零售市场当中，量化投资基金占了 16%的市场份额。在机构市场当中，由于机构的科研实力更加先进，量化投资的份额更大。其中，以巴克莱全球投资管理公司、高盛国际资产管理公司以及道富环球投资管理公司，是全球范围内量化投资的佼佼者。

量化投资之所以发展如此迅速，原因有两点，第一，是其具有纪律性、无间隙、稳定性。第二，是全球各大市场正趋向于半强式有效市场。

量化投资的纪律性体现在，其所有的决策都是依靠模型做出的，而非人为[19]。每次都是通过系统运行程序，将市场上的各类信息做即时的做处理。这就避免了人性的种种弱点，面对市场的起伏，人心会随之而起伏，而计算机不会，计算机总能够按照既定的方式运行。或许这种模型不一定每次都正确，但能够被实际运用的模型，都是能够通过大量的历史数据进行检验并且取得良好效果的，在大数据的支持下，能够保证模型在大概率上是正确的。

量化投资的无间隙体现在时间和数量上，任何市场信息都会即时被捕捉，不会延迟太多，能够同一时间接收到所有当下发出的市场信息，这比人为收集信息便捷。并且能够同时捕捉到几乎所有市场上公开出来的信息，只要模型当中有相应的代码。

量化投资的稳定性体现在其以概率取胜上面[20]。犹如在第一章当中所举的抛硬币的模型。

量化投资迅速发展的另外一个重要原因在于世界各大市场都将由最初的无效市场向有效市场的方向发展，而许多大型经济体已经趋近于半强式有效市场。下图为各大市场对应的投资方式有效性：

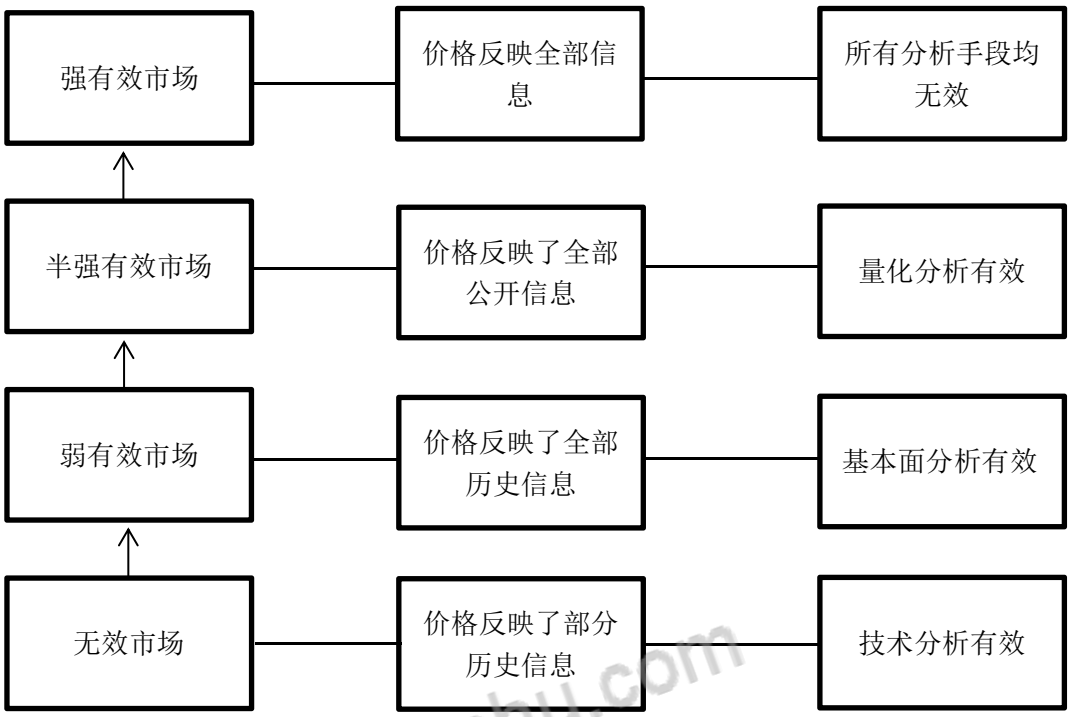


图 2.1 各大市场下的分析方法适应性

如上表所示，在无效市场当中，技术分析有效。中国资本市场最初 10 年便处于这种状态。随着经济的发展，市场进入弱有效市场，此时还能够靠基本面分析获得超额收益，我国 2000 年至 2010 年便是处于这个状态当中。当进入半强有效市场后，也就是 2010 年之后，基本面分析已经无法获得超额收益了。此时，除了靠内幕消息以外，只有依靠量化投资进行数据挖掘，挖掘出潜在的信息，才有可能获得超额收益。

2.2 机器学习简述

机器学习属于是人工智能发展的一个重要分支。随着人工智能的不断发展，人们希望机器也具有自我学习的能力。

机器学习当中有诸多的算法，例如支持向量机算法、神经网络算法、决策树算法、朴素贝叶斯算法、逻辑回归算法、关联算法、K 近邻算法、遗传算法等等

诸多算法。

机器学习算法注重的是归纳而非演绎。一般情况下会区分出训练集和测试集。训练集一般为历史数据，有明确的输入变量与输出变量。根据输入变量与输出变量，各类算法将会拟合出相应的参数，使得训练出来的预测模型的预测误差最小化。之后再将训练集训练出来的模型用于测试集，查看模型的泛化能力。

正因为机器学习具有非常强的归纳能力，因此被大量用于数据挖掘领域。数据挖掘便是从大量不完整、模糊化、有噪音的数据当中提取出隐藏的、有价值的信息。机器学习拟合出来的模型正好能够对新数据进行拟合，能够最大程度利用经验数据，从纷繁复杂的新生数据当中挖掘出有用信息。

2.3 量化投资与机器学习的关联简述

正因为机器学习算法能够根据大量经验数据拟合出具有泛化能力的模型，而量化投资正是需要具有泛化能力的模型进行投资指导。因此，量化投资领域需要大量运用机器学习算法。

目前世界上最成功的量化投资基金——大奖章基金，其发起者与管理人——詹姆斯.西蒙斯，在一次采访当中明确表示：“我们所做的工作，某种程度上来说就是机器学习而已”。

投资领域的信息是瞬息万变的，而且具有非线性性、模糊性、噪音性，然而，投资领域的信息当中又包含着大量潜在的投资信息，这些信息是难以用直观的方式获得的。因此，能否从噪音信息当中提取出有用信息，便是量化投资需要解决的重大问题。若将机器学习算法运用到量化投资当中，是否能够有效挖掘出市场潜在信息，便是最为关键的问题。从以往的表现来看，虽然有不少成功的案例，但从各大成功案例来看，其成功往往是因为该机器学习算法对该时段或者该品种具有良好的适应性。将某时刻成功的模型运用于其他时刻，却往往导致失败。

可见，机器学习算法虽然能够有效挖掘出市场潜在信息，但也具有其局部适应性。机器学习本身的目标是尽量根据经验数据拟合出放之四海而皆准的模型，但目前看来还并没有某种单一的算法能够做到这一点。这也是本文所要重点研究并试图解决的问题。

3. 机器学习算法与量化平台简介

3.1 逻辑回归算法

3.1.1 逻辑回归算法原理

在回归分析之中，因变量 y 有两种可能性：

第一，y 是定量变量。我们可以对 y 做回归。

第二，y 是定性变量，如 y=0 或 1。此时，我们不用普通回归，而可以使用逻辑回归。

逻辑回归的方法主要是运用在研究一些事件发生的概率 p，比如股票涨跌的概率，事情成败的概率。逻辑回归的基本形式为：

$$p(\mathrm{y}=1 \mid x_1,x_1...,x_k)=\frac{\exp(\beta_0+\beta_1x_1+...+\beta_kx_k)}{1+\exp(\beta_0+\beta_1x_1+...+\beta_kx_k)} \quad (3.1)$$

其中，$\beta_0,\beta_1,...,\beta_k$与多元线性中的回归系数类似。

该式子的含义为，当自变量为$x_1,x_2...,x_k$时，因变量 $\mathrm{y}=1$ 的概率。对此式子进行对数变化，可得：

$$\ln\frac{p}{1-p}=\beta_0+\beta_1x_1+...+\beta_kx_k \quad (3.2)$$

此时可见，我们只需要对因变量 p 按照 $\ln(p/(1-p))$的形式进行对数变换，便可以将逻辑回归的问题转化为已知的线性回归问题进行求解。

此时，我们需要做的是估计出模型的参数$\beta_0,\beta_1,...,\beta_k$，此时需要用到极大似然估计。假设我们有 n 个独立的样本，$\{(x_1,y_1),(x_2,y_2)\dots(x_n,y_n)\}$，$y=\{0,1\}$，那么每个观测到的样本$(x_i,y_i)$出现的概率是：

$$p(y_i,x_i)=(p(y_i=1,x_i)^{y_i}(1-p(y_i=1,x_i)^{1-y_i} \quad (3.3)$$

似然函数为：

$$\mathrm{L}(\beta_0,\beta_1,...,\beta_k)=\prod(p(y_i=1,x_i)^{y_i}(1-p(y_i=1,x_i)^{1-y_i} \quad (3.4)$$

在得到估计的参数后，我们可以根据新输入的变量$x_1,x_2...,x_k$来估计$y_i=1$ 的概率。

将其运用到量化投资当中，比如运用到股市预测当中，我们可以选取某只股票昨日开盘价、收盘价、最高价、最低价、成交量、MACD、KDJ、CCI、资金

流入流出等等变量作为自变量，如果第二天股价最后是上升的，则记为 $y = 1$。

求解该似然函数，可求得$\beta_0,\beta_1,...,\beta_k$的值，该求解过程中需要用到梯度下降等算法[21]，由于篇幅有限本文不再赘述。

3.1.2 逻辑回归算法特点

逻辑回归算法发展至今已经变得相当成熟，虽然逻辑回归算法复杂性不如支持向量机、神经网络等算法，但其预测效果却并不绝对落后于其他复杂的算法。逻辑回归算法在诸多检验后主要体现出以下几个特点：

①逻辑回归算法产生较早，先后已有过诸多学者投入进行研究，该算法已经是一种成熟稳定的算法，并且预测普遍较为准确。

②逻辑回归算法所求出来的参数直观并且容易理解，而并非属于黑盒模型。

③该模型被广发用于银行业相关数据的分析与挖掘当中，并且表现出较为优良的准确率。

④该算法由于思想简单，因此程序运行速度非常快。

⑤该算法只适用于 y 较少的分类，对于 y 较多的分类，该算法并不适合。

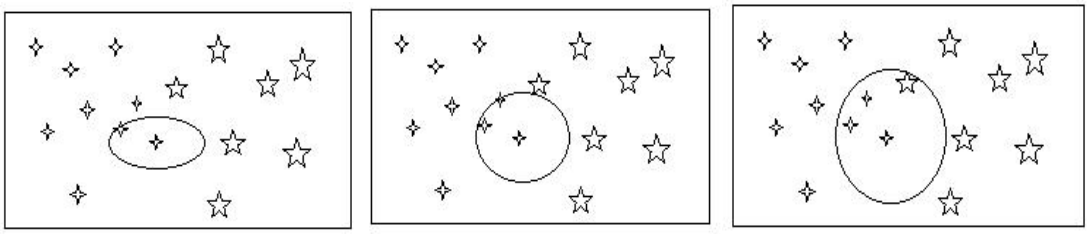
3.2 KNN 算法

3.2.1 KNN 算法的原理

KNN 算法是一种基于实例来进行分类的机器学习算法。有 Cover 与 Hart 在 1968 年提出[22]。属于非参分类方法。

该方法的基本思想是物以类聚。KNN 算法把每个样本看做是 P 维空间里面的一个点，给定一个测试样本，则可以计算出该测试样本与该训练样本中其他点之间的距离。选取与测试样本距离最近的 N 个训练样本点，看这 N 个训练样本的样本属性，哪种属性的最多，则测试样本被归为何种类型。

如图所示：



（1）1-近邻　　（2）2-近邻　　（3）3-近邻

图 3.1 KNN 示意图

N-近邻便是寻找附近 N 个训练样本，并且查看该样本属于何种类型。图中 1-近邻查找到最近的一个训练样本是十字星，则将其归类为十字星。2-近邻查找到最近的两个都是十字星，则将其归类为十字星。3-近邻查找到最近的三个两个为十字星，一个为五角星，仍然将其归类为十字星。

其算法的具体步骤为：

①计算测试样本与每个训练样本的距离 D。

②得到目前的 K 个最近训练样本中的最大距离 MD。

③如果 D<MD，就将该训练样本做为 K-近邻样本。

④重复步骤①②③，直到所有测试样本与所有训练样本的距离算完。

⑤统计 K 个最近邻样本当中每种类别出现次数。

⑥选择出现频率最高的类别作为测试样本的类别。

从算法步骤我们可以看出，KNN 算法对 K 值有较高的依赖性。因此选择 K 值非常重要。如果 K 值太小，则预测的结果稳定性较差，容易产生变动。如果 K 值太大，则会增加误分类的可能性。

一般有两种方法确定 K 的值，一种是用训练数据的个数 n 除以有效参数的个数来得到 K 的值。另外一种是利用通过 CV 的方式选取到合适的 K 值。

3.2.2 KNN 算法的特点

KNN 算法在进行分类时，能够有效解决样本不平衡的问题，因为 KNN 算法只需要取领近的点做参考。另外由于其通过相邻点来分类而不需要依靠某个超平面的特点，因此在类域交叉重叠较多，非线性分类特征较明显的分类当中会有较好的表现。

此方法的最大缺点是计算了非常庞大。由于对于每一个待分类的样本都需要去计算一次其与全部已知样本之间的距离，才可以求得其最近的 K 个最邻近点。对于这个缺陷，有两种方法进行改进：

①根据实际情况，实现对于大样本进行剪辑。去除一部分对分类作用影响不大的样本点。可以有效减少计算了，增加储存空间。但不适合于小样本。

②整理样本，并进行分层处理。使得计算尽可能在测试样本领域的小范围内进行。

总体来看，此算法具有很强的适应性。尤其对于样本容量非常大的自动分类问题。

3.3 朴素贝叶斯算法

3.3.1 朴素贝叶斯算法原理

朴素贝叶斯算法是由 Thomas Bayes 发明。因此用其名字命名。

贝叶斯定理是概率论的一个结果，其跟随机变量条件概率和边缘分布有关。在关于概率的解说当中，贝叶斯定理能够让人们利用新证据修改已有看法。

通常来说，事件 A 在事件 B 发生的条件下发生的概率，与事件 B 在事件 A 发生的条件下发生的概率是不同的。然而，这两者的关系却可以用贝叶斯公式来描述。

假设 X、Y 两个随机变量。他们的联合概率 P(X=x,Y=y)是指 X 取 x 并且 Y 取 y 的概率。条件概率指的是随机变量在另外一个随机变量取值已知情况下取得某一个特定值的概率。比如，P(Y=y ｜ X=x)指在 X 取得 x 时，Y 取 y 的概率。X 与 Y 的联合概率与条件概率会满足如下关系式：

$$P（X,Y）=P（Y ｜ X）\times P（X）=P（X ｜ Y）\times P（Y） \tag{3.5}$$

对式子变形，能够得到下面的公式，我们称之为贝叶斯定理：

$$P（Y ｜ X）=\frac{P（X ｜ Y）P（Y）}{P（X）} \tag{3.6}$$

贝叶斯定理非常有用。它允许我们通过先验概率 P（Y）、条件概率 P（X ｜ Y）和证据 P（X）来表示后验概率。然而在贝叶斯分类器之中，朴素贝

叶斯是最常用的

朴素贝叶斯是一种非常简单的分类算法，正是由于该方法的思想朴素，因此我们称之为朴素贝叶斯[23]。对给出的待分类类项，我们只需要求解出在此项出现的条件下各类别出现的概率，哪个最大，便认为此待分类项是属于哪个类别的。

比如我们见到一男一女一起逛街，我们最大可能是猜测他们之间是一对情侣。虽然他们也可能是兄妹或者是其他关系，但如果没有其他更多的信息提供证明，根据以往数据的统计，100 对逛街的男女，有 90 对是情侣，有 5 对是兄妹，有 5 对是其他关系。一男一女逛街是一种现象，这种现象出现的情况下这一男一女是情侣的统计概率最高，因此我们就将其归类为情侣。这便是朴素贝叶斯的思想。

朴素贝叶斯分类器以其朴素的思想，简便的算法，良好的表现，而受到人们的关注。它属于最优秀的分类器之一。朴素贝叶斯分类器是建立在一个类别条件相互独立的假设基础之上的。给定类结点之后，各属性结点之间是相互独立的。根据朴素贝叶斯类条件独立假设，我们由如下式子：

$$P（X \mid C_i） = \prod_{k=1}^{m} \mathbf{P}(X_k \mid C_i） \tag{3.7}$$

条件概率 $P（X_1 \mid C_i）$，$P（X_2 \mid C_i）$，…，$P（X_n \mid C_i）$ 能够从训练数据集求得。根据此种方法，对于一个未知类别的样本 X，能先分别计算 X 属于每个类别C_i的概率 $P（X_n \mid C_i）P（C_i）$，然后再选择其中概率最大的作为其类别。

朴素贝叶斯分类器的步骤如下：

①假设有 $X = \{a_1,a_2,...a_n\}$ 为一个待分类项，其中每个 a 是 X 的一个特征值。

②有类别的集合 $C = \{y_1,y_2,...,y_n\}$。

③计算 $P（y_1 \mid x）, P（y_2 \mid x）,…, P（y_n \mid x）$。

④如果 $P（y_k \mid x）$=max $\{P（y_1 \mid x）, P（y_2 \mid x）,…, P（y_n \mid x）\}$，则 $x \in y_k$。

问题的关键就在于如何能够计算出第三步当中的各个条件概率。做法如下：

①找到一个已知分类的训练集。

②统计得到各种类别下各特征属性条件概率，即

$P（a_1 | y_1）$，$P（a_2 | y_1）$，…，$P（a_m | y_1）$，

$P（a_1 | y_2）$，$P（a_1 | y_2）$，…，$P（a_1 | y_2）$，

……

$P（a_1 | y_n）$，$P（a_1 | y_n）$，…，$P（a_1 | y_n）$

③如果各个特征属性是条件独立，那么根据贝叶斯公式有如下式子：

$$P（y_i | x）= \frac{P（x | y_i）P（y_i）}{P（x）} \quad （3.8）$$

由于分母对所有类别是常数，因此只需要将分子最大化便可。由于各特征属性是条件独立，因此有：

$$P（x | y_1）P（y_i）= P（a_1 | y_i）P（a_2 | y_i）...P（a_m | y_i）P（y_i）= \prod_{j=1}^{m} P（a_j | y_i） P（y_i） \quad （3.9）$$

因此，朴素贝叶斯分类有三个阶段：

第一阶段：确定训练集特征属性。首先确定有哪些特征属性。运用在量化当中，可以用前一日的开盘价、收盘价、最高价、最低价、成交量、资金流入流出、MACD、KDJ 等数值作为特征属性。输入变量是训练样本的特征属性，输出变量是训练样本的分类，此处的分类可以是次日上涨或是下跌。

第二阶段：训练分类器。此阶段需要生成分类器。主要的工作便是计算出每个类别在训练样本当中出现的频率以及各个特征属性划分对每个类别的条件概率估计，依据上文所述的计算方法计算出相应的数值，并根据数值训练出分类器。

第三阶段：应用阶段。此阶段便是运用训练好的分类器对测试样本作分类，在量化投资当中，运用训练好的模型，输入前一日的开盘价、收盘价、最高价、最低价、成交量、资金流入流出、MACD、KDJ 等数值，便可预测下一次价格是涨或是跌。

3.3.2 朴素贝叶斯算法特点

朴素贝叶斯算法的一个重要前提假设是各特征属性之间是相互独立的。这在现实当中较难实现。在量化投资实际运用当中，开盘价、收盘价、最高价、最低

价、成交量、资金流入流出、MACD、KDJ 等数值之间常常是由相互关联的。因此这可能会降低分类效果。但朴素贝叶斯分类器以其简单明了的分类思想，具有较高的运算效率，因此也常常被用到量化投资当中。朴素贝叶斯分类器通常都具有如下的特点：

①稳健性：朴素贝叶斯算法由于思想简介，计算简单，面对有许多噪音的数据，朴素贝叶斯分类器是非常稳健的。当我们从数据当中估计条件概率的时候，噪音点会被平均。另外当我们选择的特征属性当中如果存在着与分类结果相关性不高的属性时，如果特征属性较多，统计学当中是习惯于用主成分分析的方法去降低维度，或者利用 LASSO 算法将相关性不高的特征属性压缩为 0。这些方法都是进行降维处理。然而朴素贝叶斯算法对存在相关性不高的变量是不需要进行降维处理。如果特征属性与分类结果相关性不高，那么统计出来的 $P（x_i | Y）$ 会趋近于均匀分布，x_i的类条件概率不会对总体后验概率计算产生太多影响。

②特征属性选取越多，分类效果降低的可能性越高。由于朴素贝叶斯算法的前提假设是特征属性之间是不相关的，如果特征属性数量太多，则会增加他们之间存在相关性的可能，从而降低分类效果。因此在用朴素贝叶斯分类器之前，可以使用主成分分析方法先将具有相关性的变量去除。

3.4 支持向量机（SVM）算法

3.4.1 支持向量机（SVM）算法原理

支持向量机（SVM）算法是由 Vapnik 等人于 1995 年提出的具有非常优良表现的机器学习算法[24]。该方法是建立在统计学理论之上的。此方法试图寻找一个对分类有着良好区分能力的支持向量，由此来构造出分类器以最大化区分出各种类别。该方法具有非常强的适应性与分类准确率。并且只需要用到类域边缘的样本来决定最后的分类结果。

支持向量机是属于有监督学习的范畴，即已知训练样本的类别，寻找出训练样本的特征与类别之间的对应关系。以便于将训练集按类别分开，再预测新的测

试集的点所属类别。SVM 以其清晰直观的表现，在文本、文字、图像分类方面有着不菲的表现[25]。运用到量化投资当中，还能够将消息面的文字进行抓取，转化为特征属性之一进行输入，加大了量化投资信息面的范围。

SVM 试图构建出一个超平面，将类别进行最优化的隔离。企图使得两种类别之间的分割达到最大化的程度，如图所示：

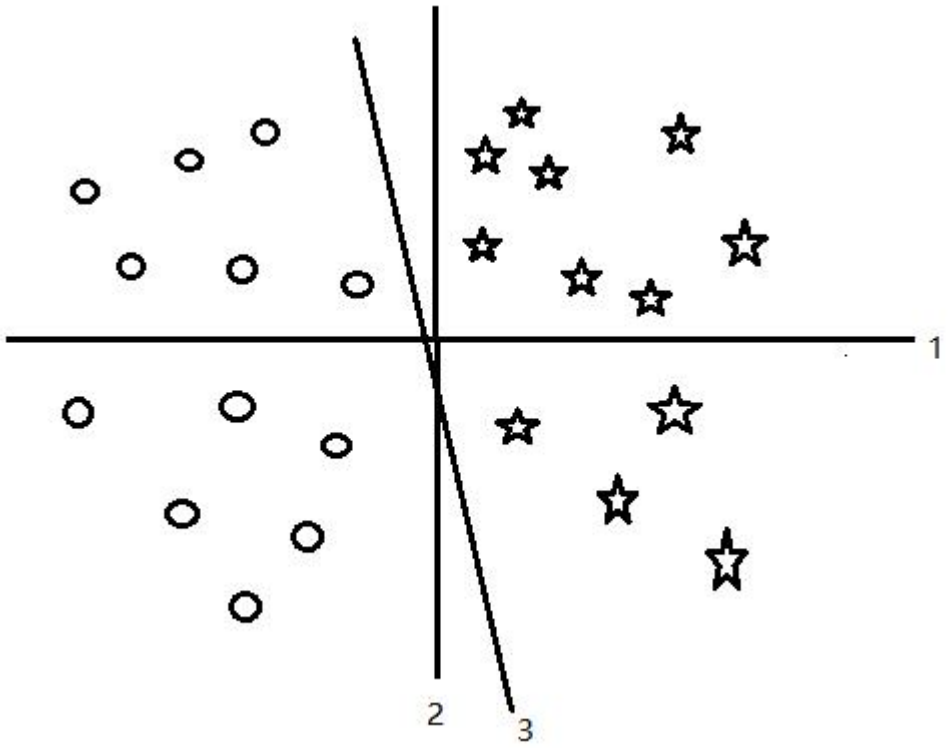


图 3.2 SVM 原理

第一个边缘和第二个边缘都不能将两类样本合理分开，第三个边缘便是支持向量，其能够很好地将两类样本分割开来。

SVM 以一个非常大的超平面分割开两类样本，使得期望泛化误差最小。所谓的泛化误差最小是指，当对测试样本进行分类时，基于学习所得的分类器，使得预测的错误概率被最小化。从直觉上来看，该超平面实现了两个类别之间的边缘最大化。与该超平面平行的，分别穿过训练数据集中一个或多个点的两个平面称为边缘。而 SVM 正是找到最大化这个边缘的超平面。将落于边缘上面的点我们称其为支持向量。

简单说来，便是找到与超平面平行的两个边缘，这两个边缘能很好地分开两类不同的数据，并且分别穿越了两类数据当中的点，而最佳超平面是与两个边缘之间距离最大的平面。如此便实现了分类误差最小化。

支持向量机最初研究的是线性可分问题。首先了解一下线性 SVM 的原理。假设有一个训练集{(x_i,y_i),i = 1,2,.....n}，其中有两个类别，如果x_i属于第一类，则y_i=1；如果x_i属于第二类，则y_i=-1。如果存在超平面：

$$w^T x + b = 0 \tag{3.10}$$

可以将样本正确划分为两类，也就意味着相同类别样本落在超平面同一侧，我们就称该样本集是线性可分的，也便满足：

$$\begin{cases} w^T x_i + b \geq 1，\ y_i = 1 \\ w^T x_i + b \leq -1，\ y_i = -1 \end{cases} \tag{3.11}$$

此外，可以知道平面$w^T x_i + b = 1$ 与$w^T x_i + b = -1$ 便是该类分问题当中的边界超平面。这便转化为了线性规划问题。由线性规划的知识可知$w^T x_i + b = 1$ 到原点距离是丨 b-1 丨/‖**w**‖；$w^T x_i + b = -1$ 到原点距离是丨 b+1 丨/‖**w**‖。因此这两个边界超平面距离便是 2/‖**w**‖。SVM 的目标便是最大化两个平行边缘的距离，也就是最大化 2/‖**w**‖。这就转化为了最小化其倒数，也即：

$$\min: \frac{1}{2}\|\mathbf{w}\| = \frac{1}{2}\sqrt{w^T w} \tag{3.12}$$

（b）式中的参数必然是满足（a）式的，（a）式可以综合为：

$$y_i(w^T x_i + b) \geq 1 \tag{3.13}$$

因此问题转化为求解：

$$\min: \frac{1}{2}\|\mathbf{w}\| = \frac{1}{2}\sqrt{w^T w} \tag{3.14}$$

$$s.t.\ y_i(w^T x_i + b) \geq 1，\ i = 1,2，\ \ldots，\ n$$

至此，问题便转化为一个凸优化问题。此问题能够运用现成的 Quadratic Programming 的优化包进行求解，也可以运用拉格朗日变化进行求解。后者比前者会更加高效一些。拉格朗日变化便是通过给每一个约束条件都加上一个拉格朗日乘值α，便可以将约束条件融合进目标函数当中去。SVM 的拉格朗日表达式为：

$$\mathrm{L}(w,b,a) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum a_i[y_i((w^T x_i + b) - 1] \tag{3.15}$$

其中，a_i>0，i=1,2,…，n，为拉格朗日系数。

再依据拉格朗日对偶理论，得到：

$$\begin{cases} \max: L(\alpha) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n} a_i a_j y_i y_j (x_i^T x_j) \\ s.t. \sum_{i=1}^{n} a_i y_i = 0, \ a_i \geq 0 \end{cases} \quad (3.16)$$

这问题能用二次规划法求解。设所求得到的最优解为：

$$a^* = [a_1^*, a_2^*, \ldots, a_n^*]^T \quad (3.17)$$

得到最优的w^*和b^*是：

$$w^* = \sum_{i=1}^{n} a_i^* x_i y_i \quad (3.18)$$

$$b^* = -\frac{1}{2} w^* (x_r + x_s) \quad (3.19)$$

其中，x_r与x_s是两个类别当中任意的一对支持向量。

最终可得到最优分类函数为：

$$f(x) = \mathrm{sgn}[\sum_{i=1}^{n} a_i^* y_i (x^T x_i) + b^*] \quad (3.20)$$

在输入空间当中，若数据线性不可分，则支持向量机可以通过非线性映射ϕ：$R^n \rightarrow G$将数据投射到另外一个空间 G 当中，在空间 G 当中进行线性运算。此时需要计算点积$\phi(x)^T \phi(x)$表示。

因此 SVM 理论中有三个要点：

①最大化间距。

②选择核函数。

③对偶理论。

对于现行 SVM 问题，我们还能够通过引入松弛变量将问题转化为纯线性规划的问题。该问题解法如下：

典型的 SVM 模型能够被描述为：

$$\begin{cases} \min: \frac{\|w\|^2}{2} + v\sum_{i=1}^{n} \lambda_i \\ y_i(w^T x_i + b) + \lambda_i \geq 1, \ \lambda_i \geq 0, \ i = 1,2, \ldots, n \end{cases} \quad (3.21)$$

有学者已证明该模型与如下模型的解释几乎一致的：

$$\begin{cases} \min：\ v\sum_{i=1}^{n}\lambda_i \\ \\ y_i(w^T x_i + b) + \lambda_i \geq 1，\ \lambda_i \geq 0，\ i = 1,2，\ \ldots，\ n \end{cases} \tag{3.22}$$

此时，二分类的 SVM 问题就转化为了线性规划问题。

3.4.2 支持向量机（SVM）特点

SVM 的特点主要有以下三点：

①SVM 学习问题转化为凸优化问题[26]。因此我们能够用已知的有效的算法来发现目标函数的全局最小值。而其他的分类方法都是运用基于贪心学习的策略来搜索假设空间，这些方法往往只能得到局部最优解。

②SVM 的核心思想是找到最优超平面，因此 SVM 最关键的地方就是如何最大化边际。而在 SVM 当中的支持向量是起着决定作用的点。

③SVM 与其他机器学习算法不同，不设计到概率论与大数定理[27]。其他方法是通过大样本寻找到规律，实现归纳到演绎，而 SVM 巧妙地利用支持向量，高效地解决这个问题。

④SVM 的计算只需要通过少量的支持向量，因此能够避免位数灾难的问题。

⑤SVM 对于超大规模样本的训练难以进行。由于 SVM 是靠二次规划的问题，因此涉及到 n 阶矩阵的运算。（n 为样本量）当 n 非常大的时候，矩阵的运算需要耗费大量的内存和时间。

⑥SVM 主要解决的是二分类问题，而实际问题涉及到多分类问题的时候，需要运用到组合 SVM 等等其他方法才能解决实际问题。

3.5 神经网络算法

3.5.1 神经网络算法原理

神经网络算法的理论最早由 1943 年的数学家 pitts 与心理学家 McCulloch 共同提出，此时的神经网络是用数学的方法模仿神经元工作原理[28]，因此称为神经元数学模型，MP 模型。在随后的 1949 年，Hebb 进一步研究人体神经网络，发现神经网络的运行关键在于突触之间的抑制与激活，神经信号的强度决定着突触之间的激活，从而影响到信息的传递。该思想对神经网络算法奠定了基础。在 50 至 60 年代，神经网络算法开始运用到工学领域[29]，运用于过程控制当中，神经网络算法一时达到高潮。但到了 60 年代后期，神经网络的发展陷入了低谷。由于人工智能创始人 Minsky 出版的《Perceptron》一书当中揭示了神经网络算法的种种局限性[30]。由于其权威性，学术界对于神经网络的研究一时陷入了低潮。再加上当时人工智能的崛起，人们更加投入到计算机的成功之中。当时的人们对于人体神经网络的研究也存在局限性，并且无法在现实中模拟出类似于人体神经网络的模型，因此神经网络一度陷入了低潮。

直到 John Hopfield，提出模仿人脑的神经网络模型时，神经网络才开始重新流行，他提出了“计算能量函数”的概念，其设计的电路系统对计算机模仿人脑神经网络奠定了基础。后来 Rumelhart 与 Mcclelland 提出了并行分布处理 PDP 理论[31]，发展处了多层 BP 神经网络算法，称为至今最为流行的神经网络算法。

近年来，神经网络在模式识别，图像识别，人工智能，机器人科学，量化投资领域等等都有非常广泛的运用。

大脑认知复杂世界依靠的是大脑神经。而大脑神经则表现为庞大神经元之间相互传递信息组成神经网络。神经元与神经元之间的信息传导靠突触传播递质，而神经递质达到一定浓度以后神经元便会被激活。正是这种简单的神经元之间传递信息的原理，使得人脑能够进行复杂的思考，认知复杂事物。因此科学家模仿人脑原理而提出神经网络算法，其算法原理与人脑神经网络极为相似。该原理的基本表示如下图所示：

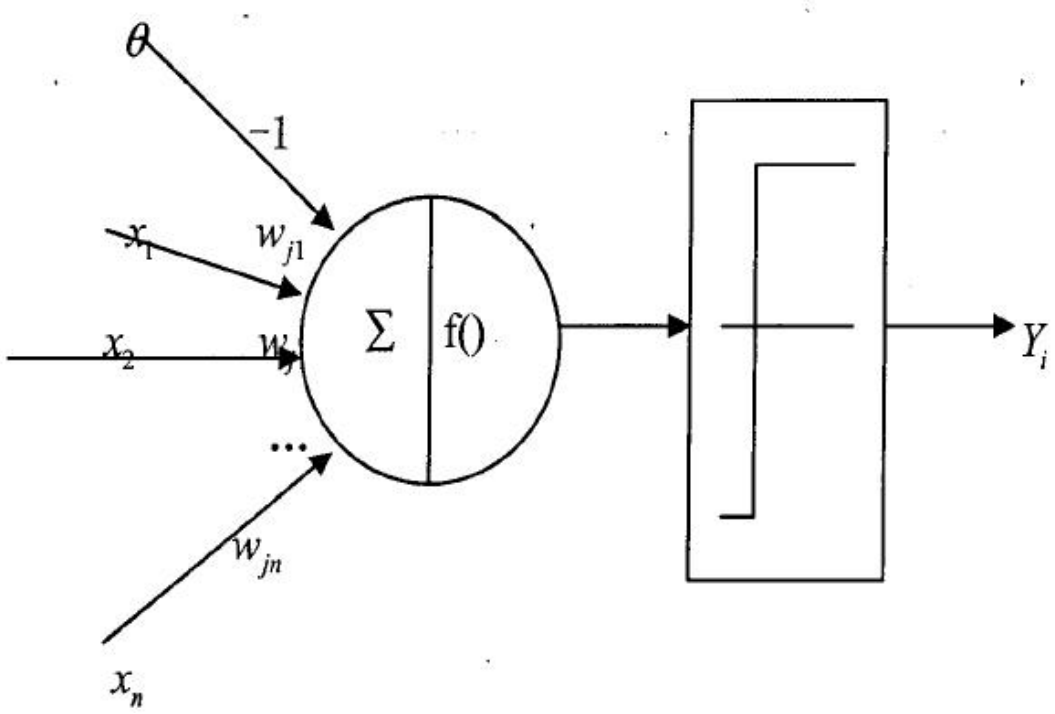


图 3.3 神经网络原理

该原理可表达为：

$$y_j(\mathrm{t}) = f\left(\sum_{i=1}^{n} w_{ji}\, x_i - \theta_j\right) \tag{3.23}$$

其中的x_i为输入，w_{ji}为权重，θ_j为阈值。f 为激活函数。每个输出都是输入经过加权以后的一个激活函数。

激活函数可以是线性的，但更多时候是非线性的，常用的激活函数有如下几种：

①阈值函数：

$$f(x) = \begin{cases} a, & x \geqslant x_o \\ b, & x < x_o \end{cases} \tag{3.24}$$

②S 形函数：

$$f(x) = \frac{1}{1+e^{-ex}},\ f(x) \in (0,1) \tag{3.25}$$

③双曲线正切函数：

$$f(x) = \frac{1-e^{-ax}}{1+e^{-ax}},\ f(x) \in (-1,1) \tag{3.26}$$

神经网络是以一定的准则进行学习之后在转入工作的模式。例如我们需要在输入为 1 的时候输出 A，输入为 2 的时候输出 B。我们要提高输入为 1 和输出为 A 的对应，以及输入为 2 输出为 B 的对应，因此需要减少输入为 1 时输出为 B 以及输入为 2 输出为 A 的出现次数。因此，神经网络的学习准则就是，通过一定的调整，使得下一次出现错误对应的次数减少。

神经网络根据连接的方式不同能分为：

①前向神经网络。前向神经网络便是神经网络只接收前一层的信息到后一层，不会进行反馈。

②反馈神经网络。神经网络之间的传递具有前后反馈性，通过一系列动态过程来达到我们希望达到的状态。

③自组织神经网络。神经元之间通过一定的次序进行排列，彼此之间都是通过相邻的神经元的相互作用来完成工作。

神经网络内部权值的学习规则是与网络内部结构有关的，但总的说来神经网络的学习方式分为有监督学习与无监督学习

有监督学习的情况下神经网络能够根据实际输出与期望输出之间的差值来进行调整学习。

无监督学习的情况下，则没有期望输出，只要提供了输入模式，神经网络便会自行进行学习，按照相似特征把输入的模式进行聚类。

神经网络当中最出名的莫过于 BP 神经网络。BP 神经是误差反向传播算法的简称。本文所用的神经网络模型也是 BP 神经网络模型。

BP 神经网络算法具有多层网络结构，一般由输入层，隐层，输出层组成，如图：

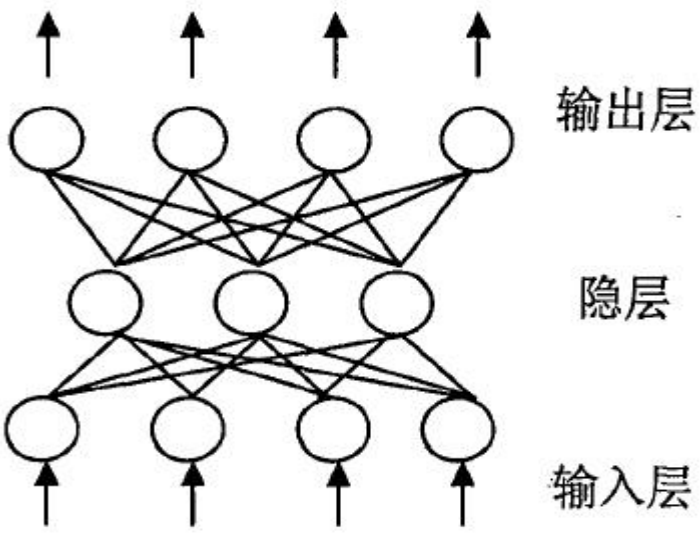


图 3.4 BP 神经网络结构

其传递函数一般是 S 型函数：

$$f(x)=\frac{1}{1+e^{-x}} \tag{3.27}$$

第 P 个样本的误差计算为：

$$\mathrm{Ep}=\frac{\sum_i (t_{pi}-O_{pi})^2}{2} \tag{3.28}$$

其中t_{pi}与O_{pi}分别为实际输出与期望输出。

本文所用到的神经网络输入为开盘价、收盘价、最高价、最低价、成交量、成交金额等等数值，输出为第二天股价上升或者下降。

3.5.2 神经网络算法特点

①具有非线性映射的能力。神经网络通过输入层、隐含层、输出层这种结构，使得非线性拟合能力大大提高，能够很好地拟合各种非线性数据。尤其适合金融数据，这种具有高度非线性、高噪声、多边性的数据。

②具有自学习与自适应的能力。能够通过大量的输入与输出数据不断自己进行参数改进，拟合出具有优良适应性的参数来处理非线性数据。

③具有很强的泛化能力，并且与节点数有关。神经网络的节点数决定了其泛化能力。虽然节点太多，会导致过度拟合，如果节点太少，会导致欠拟合，但总的说来，神经网络的泛化能力非常强。并且能够通过调整节点数去不断增强自身

的泛化能力。

④具有较高容错率。神经网络由于其不同层之间有许多的节点，因此如果某个节点出错，并不会对整体产生太大的影响。

⑤容易陷入局部最小化。神经网络算法常常使用梯度下降算法求得最优解，因此非常容易就会陷入局部最小化的缺陷当中。如果陷入了局部最小化，神经网络的拟合就算失败。并且神经网络的收敛非常依赖于权重，不同的权重会导致完全不同的参数拟合。

⑥神经网络算法的收敛速度非常慢。由于神经网络采用梯度下降算法进行计算，因此其优化函数会非常复杂，会出现“锯齿形现象”，这将导致神经网络算法效率低下。

⑦神经网络结构选择没有标准。神经网络的选择往往没有一定标准，是根据经验来进行选择。如果节点过多容易导致过拟合，节点过少容易导致欠拟合。

3.6 决策树算法

3.6.1 决策树算法原理

决策树也叫做分类树。决策树是运用最广泛的归纳推理算法之一[32]。决策树模型能通过对数据进行不断的划分，使得依赖变量之间的差别达到最大，其最终目的是将数据分到不同枝当中，在依赖变量值之上监理处最优的归类。

决策树的目的为针对类别变量预测或者解释反应的结果。就具体本身来说，此模块分析技术跟非线性估计的作用是相同的。决策树是具有弹性的分类方法，使得其对数据具有非常强的适应性。

决策树属于有监督学习，其最后的分类结果会类似于流程图一般的树形结构。决策树的终端，即“叶子节点”代表分类结果最终类别。分枝代表测试输出，代表了变量的可能值。为了达到最优分类结果的目的，变量值在数据集中进行测试，测试出的每一条路径都代表一个分类规则。

决策树用于处理分类问题时，目标变量是属于类别型变量。目前扩展到了能处理连续型变量的地步，例如 CART 模型[33]。然而不同的决策树算法对数据类

型具有不同的限制于需求。

决策树的构建主要分为三步：

①决策树分割步：决策树是通过递归分割的方法来创建，所谓的递归分割，就是把数据分割成许多细小的部分再进行迭代。决策树的归纳算法步骤如下：

1）将训练样本导入到决策树的树根。

2）将原始数据分割成两组，一组是训练数据，另外一组是测试数据。

3）通过训练样本建立决策树。在每个内部节点根据信息论的法则来评估选择哪个属性作为继续分割的根据，我们称这个步骤为节点分割。

4）再使用测试数据对决策树进行剪枝工作。剪枝使得决策树每个分类只包含一个节点。来提升预测的效率。换句话说就是，经过节点分割以后，判断这些节点是不是枝叶节点，若不是，则以新的内部节点为分枝树根建立新次分枝。

5）将第一到第四步不断递归，直到所有的内部节点都称为叶子节点为止。当决策树完成了分类以后，可以将每个分枝树叶节点萃取出新的知识规则。

一旦发生如下状况，决策树就会停止分割：

1）该数据的每一个数据都已经归类至同一种类别当中。

2）该数据已经没有新属性对于节点进行进一步的划分。

3）该数据已没有任何多余的尚未进行处理的数据。

一般说来，决策树分类的分类正确率会依赖数据的数量。如果决策树是根据庞大的数据进行建立的，那么该决策树会更加符合期望。

决策树的学习过程主要是利用信息论当中的信息增益来进行。寻找出数据集当中有最大信息量的变量，再建立出数据的的一个节点。从而根据这个建立出分枝树。每一个分枝也都重复该过程，直到整个树被构建出来。决策树的每一条路径都代表着一种分类规则，其与其他模型相比较最大的有点就是显得非常直观，有形象的分类。

对于树的每个节点，都通过信息增益来选择测试变量。信息增益是用来度量给定变量区分训练样本能力的。

信息增益可以这么理解。如果一个事件发生的概率是 P，假设这事件发生以后得到的信息量为 $I(p)$，如果 $P=1$，那么 $I(p)=0$。因为该事件是一定会发生的，我们从这件事当中得不到任何信息。相反，如果某事件发生的概率非常小，

那么他所具有的不确定性就越大。所以I（p）是一个减函数。我们可以定义：

$$I(p) = -\log\ （p）\qquad （3.29）$$

给定一个数据集发 Q，假设类别变量 A 其中有 n 个不同的类别（b_1，…，b_i，…，b_n）。可以利用变量 A 将此数据集分成 n 个子集（c_1，…，c_i，…，c_n）。当中的c_i代表 Q 当中包含了数值b_i的样本。在分类过程当中，对于每个具体样本，对应了 n 种可能会发生的概率（p_1，…，p_i，…，p_n），记第 i 种结果信息量为$-\log\ （p_i）$，我们把这个称作是分类信息的熵。熵是测量随机变量不确定性的一种测量标志。能用来测量训练数据集内纯度的标准。熵可以用如下式子表示：

$$I(s_1，\ s_2，\ …，\ s_m) = -\sum_{i=1}^{m} p_i\, log_2(p_i)\qquad （3.30）$$

这里面的p_i表示任意个一个样本属于b_1的概率，对数函数是以 2 为底的。

训练分类数据集能力能用信息增益进行衡量。计算出每个变量的信息增益，有高信息增益的变量便选定为集合 Q 的分割变量，此时生成一个节点，同时也以此变量作为标记，对每个变量值产生分枝。

②决策树剪枝步：决策树在学习过程中常常遇到过度拟合的问题，有时训练样本数量较多，容易产生较多的分枝。这导致模型失去了普遍适用性。因此，除了应该分割外，还应该进行适度地剪枝处理。

当决策树成功生成的时候，由于数据中有离群值，有许多分枝可能对此作出了反应，就容易产生过度拟合的问题。因此剪枝通常要利用统计测量值去减去不可靠的分枝。常常使用的统计测量方法有卡方值以及信息增益等等。这可以加速产生分类结果，也能够大大提高测试数据的分类能力。

剪枝有两种方法，分为先剪枝与后剪枝。先剪枝是靠停止树的生成过程来进行剪枝处理，一旦停止了分类过程，节点就会成为树叶。此树叶很可能会包含最多子集样本当中次数最高的类别。在训练决策树的时候，信息增益与卡方值一类测量值能用来评估分类的质量。若在一个节点划分了样本，将会使得低于预先定义好的阈值的分裂，那么子集进一步划分便停止了。选取合适的阈值是有困难的，如果阈值较高，决策树会过于简单，如果阈值太低，则决策树的简化工作又会做不到到位。而后剪枝是在已经训练完成的树的基础上进行剪枝的。通过删除节点

的方法进行剪枝。在最底下没有被剪掉的节点就成为叶子节点，并且用之前划分次数最多的类别作标记。对于树当中的每个非树叶子节点，通过计算出减去该节点上子树可能产生的期望错误率，之后再利用每个分枝的错误率结合每个分子的权重来进行评估，再计算不减去该节点的期望错误率。如果减去了以后错误率要大于没减去的错误率，则不进行修剪。如果减去了以后错误率要低于没减去的错误率，则进行修剪。同样可以即进行先剪枝，也进行后剪枝。

3.6.2 决策树算法特点

①决策树除了分类还能很好地作其他方面的解释。由于决策树通过形象直观地分枝形式进行展现，因此我们能够很细致地看到决策树内部的工作步骤。并且根据决策树的算法原理，决策树一般会把较为重要的分类变量放在接近根部的位置。这就使得决策树除了能够对数据进行分类，也能够提供更多除了分类以外的其他方面的解释。

②决策树既能够接受分类数据作为输入，也能够接受数值数据作为输入。由于决策树是要寻找能够最大化信息增益的分界线，因此他能够接受数值型数据的输入。但我们用传统的方法比如，如回归，则难以作这种处理。

③决策树不擅长对数值结果做出预测[34]。决策树能够将数据拆分成许多的具有最小方程的均值，可是如果数据非常复杂，训练出的决策树将会非常的庞大，这就不如用传统的方法做回归。

3.6.3 随机森林算法

决策树算法进行衍生便产生了随机森林算法。随机森林算法能够克服决策树的过拟合问题。并且能够更好地克服噪声及异常值的影响。随机森林属于一种集成学习算法。

由于单个决策树具有更大的偶然性，因此通过 Bagging 的集成学习算法，训练处诸多决策树，再进行投票，从而提高分类的精确度。

随机森林的训练步骤如下：

①从训练数据集中有放回地抽取 K 个与训练样本数量相同大小的样本集合，记为$\{T_k，k=1，2...K\}$，利用每一个$T_i，i=1，2...K$，都训练出一课决策树。这 K 个决策树就组成一个随机森林。

②对于每一个决策树的节点进行分裂时，等概率从所有属性当中选取一个属性子集，再从被选中的子集当中选取一个最优属性来对节点进行分裂。

由于每棵决策树之间都是独立同分布的，因此可以采取并行计算的方式进行随机森林的生成。过程可用如下流程图表示。

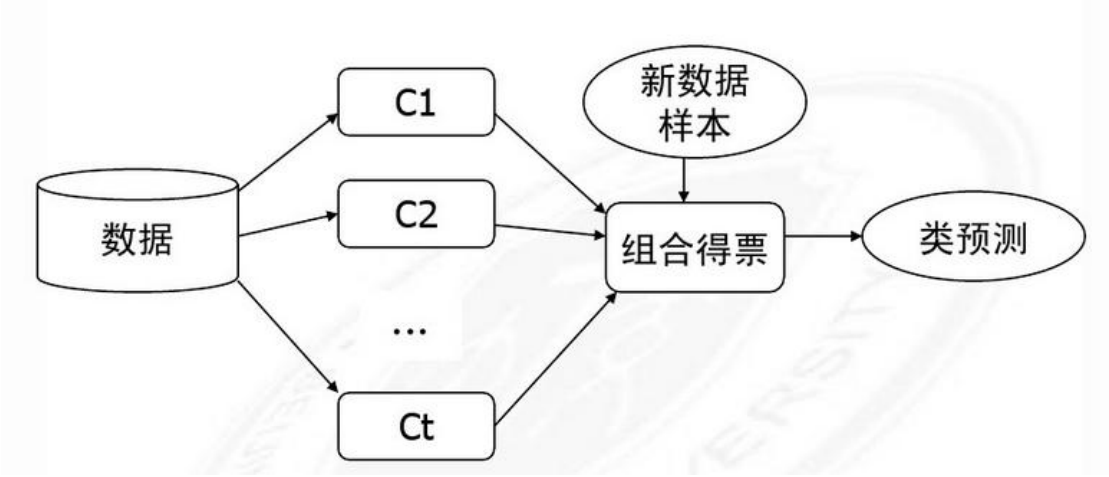


图 3.5 随机森林算法原理

3.7 量化平台简介

目前国内有诸多致力于量化投资的开放平台，比较著名的网络量化平台主要有优矿网、米筐网、金字塔交易平台、掘金量化投资平台。本文所采用的研究平台为优矿网。

优矿网提供了大量优质的金融数据。包括股票、期货，各大上市公司的基本状况都详细搜集起来，只需要会相应的代码便可以轻松获得上市公司的各大数据。以及各大技术分析数据。该量化网站运用 Python 语言。Python 语言因其简洁性，拥有大量的开源包可供编程者使用。Python 语言中的 Numpy 包与 Pandas 包能够高效处理大数据，因此非常适合于金融数据的处理。并且 Python 语言当中拥有 Talib 包能够良好分析技术指标数据，以及 Matplotlib 包具备强大的画图功能。同时具有 Sklearn 包，该包包含大量机器学习算法，只需要加载此包便能方便运用机器学习算法进行量化投资。因此该网站选用 Python 语言作为该网站的技术支持。该网站提供自带的回测功能，只需要将代码写好，就可以自动运用历史数据进行回测，并展示所运用的策略的各种表现。如图：

图 3.6 优矿网策略回测展示图

此图为运用朴素贝叶斯算法进行量化投资的程序运行结果，图中处于上方的蓝色曲线为运用机器学习算法进行量化投资所表现出来的收益率回测情况。处在下方的黑色曲线为基准收益率曲线。从图中可以较为明显地看出一个策略的好坏情况。

评价一个策略好坏主要是从收益性，风险性，与稳定性三个方面进行评价。图中三个方面皆进行了体现：

收益性：运用该策略所能产生的年化收益率为 78.3%，要较基准年化收益率 53%略高。

风险性：其中的阿尔法为超额收益率，运用该策略能够获得 20.1%的超额收益率。贝塔为该投资组合的系统性风险，该组合系统性风险略高于 1，其组合风险要略大于一般市场风险。

稳定性：稳定性主要从最大回撤情况来体现。所谓最大回撤便是相对于历史最高收益的最大收益减少率。该策略的最大回撤为 9.1%，属于较低的最大回撤率。

综合看来该策略收益率良好，风险中等，盈利稳定性较好。

本文的所有策略测试都建立在该平台上面。

下单/成交时间	证券代码	买/卖	下单/成交数量(股)	下单/成交价格	成交额
2014-08-07	共 3 个订单				
2014-08-08	买 1,526.90, 共 3 个订单				
09:30 / --	600650 锦江投资	--	0 / 0	市价 / 0.00	0.00
09:30 / 09:30	000619 海螺型材	买入	100 / 100	市价 / 6.79	678.60
09:30 / 09:30	000401 冀东水泥	买入	100 / 100	市价 / 8.48	848.30
2014-08-11	买 919.90, 卖 1,530.90, 共 3 个订单				
09:30 / 09:30	600650 锦江投资	买入	100 / 100	市价 / 9.20	919.90
09:30 / 09:30	000619 海螺型材	卖出	100 / 100	市价 / 6.82	681.60
09:30 / 09:30	000401 冀东水泥	卖出	100 / 100	市价 / 8.49	849.30

图 3.7 部分持仓情况显示

该系统还能够在回测详情当中显示各种股票具体的买卖与持仓情况。能够供我们进行具体买卖情况的检测。

4. 股市各类型走势下的机器学习算法效果比较

4.1 股市走势类型简介

股市中曾出现过一类著名的技术分析方法名为缠论[35]。缠论试图通过数学的方法来研究股市。缠论结合高等数学、几何学、逻辑学等诸多学科，将股市作为一个能够被完全分类的对象去进行研究。

正因为缠论对股市进行过完全分类，因此，本文借鉴缠论中对股市的几种完全分类来进行研究。

缠论当中将任何股票的走势都分为上升趋势，下降趋势，以及盘整趋势。在论及三种趋势前有必要提及缠论当中走势中枢的概念，缠论当中对于走势中枢有过严格的数学定义，其数学定义为：

假设 A、B、C 三条线依次是上升、下降、上升，或者下降、上升、下降，A 的高低点分别为 a1，a2；B 的高低点分别为 b1，b2；C 的高低点分别为 c1，c2。那么缠论走势中枢的区间就是[max(a2，b2，c2)，min(a1，b1，c1)]。

如图所示：

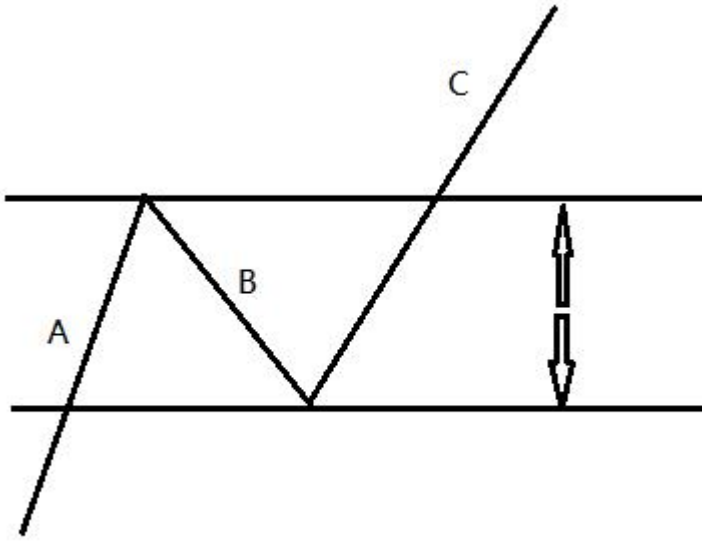


图 4.1 缠论走势中枢示意图

A、B、C 为三种趋势，而两条平行线表示了一个走势中枢。

缠论当中对于趋势的定义为：

在任何级别的任何走势中，某完成的走势类型至少包含两个以上依次同向的缠论走势中枢，则称为趋势。用示意图可表示为：

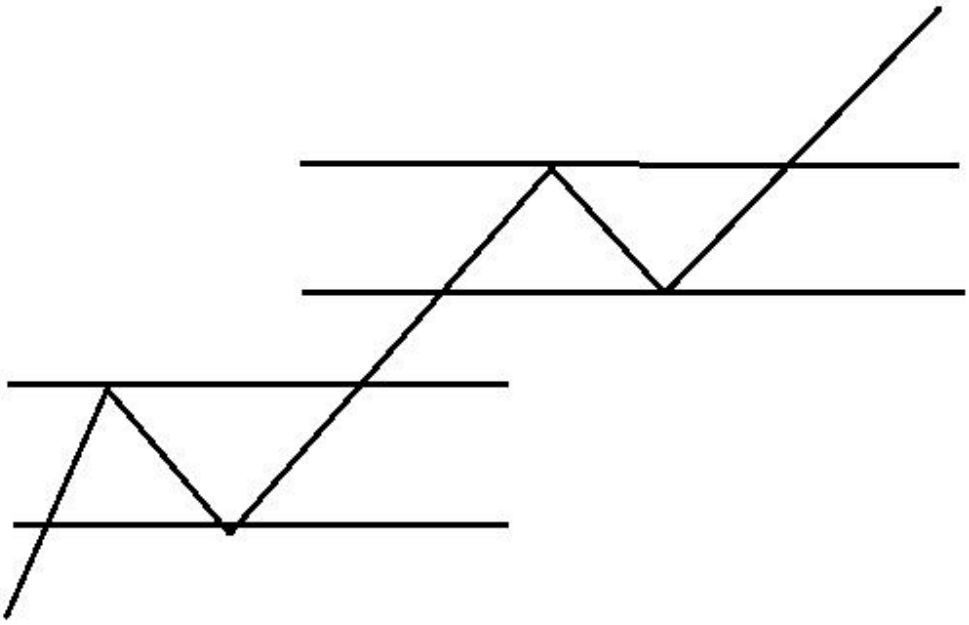


图 4.2 上升趋势示意图

此为上升趋势示意图。下降趋势则为两个或两个以上依次降低的走势中枢。

缠论对于盘整的定义为：

在任何级别的任何走势中，某完成的走势类型只包含一个缠论走势中枢，则称为盘整。用示意图可表示为：

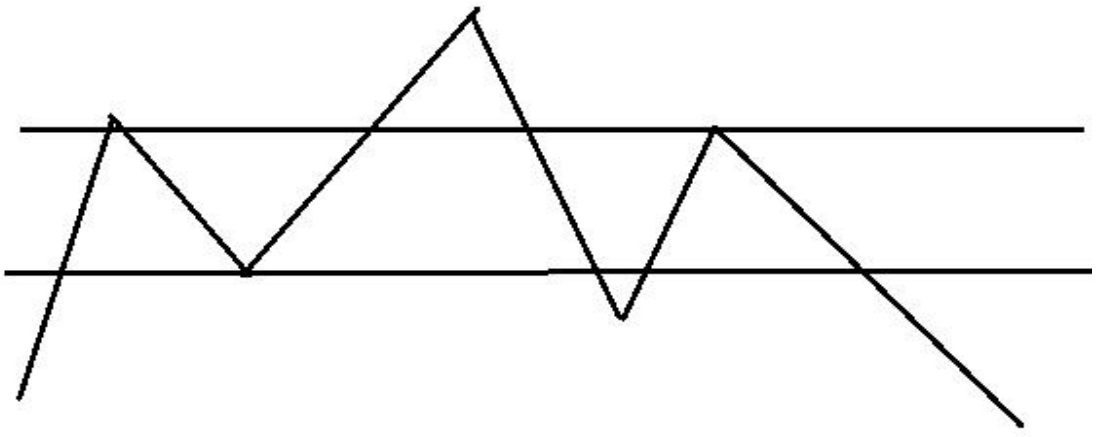


图 4.3 盘整示意图

本文将测试在上升趋势、下降趋势以及盘整走势当中，机器学习各大算法在量化投资当中的表现。

本文重点研究的是中国 A 股，因此，本文选用上证综合指数作为选取三种走势的参考标的。

首先选择上升趋势，如图：

图 4.4 2014 年 11 月 20 日至 2015 年 6 月 15 日上证指数日线图

根据缠论对于上升趋势的定义，选取 2014 年 11 月 20 日至 2015 年 6 月 15 日的上证指数作为上升趋势的走势。该走势包含两个向上的走势中枢，是典型的上升趋势。

其次选择下降趋势，如图：

图 4.5 2011 年 4 月 15 日至 2012 年 1 月 6 日上证指数日线图

根据缠论对于下降趋势的定义，选取 2011 年 4 月 15 日至 2012 年 1 月 6 日的上证指数作为下降趋势的走势。该走势包含两个向下的走势中枢，是典型的下降趋势。

最后选择盘整走势，如图：

图 4.6 2013 年 12 月 5 日至 2014 年 8 月 8 日上证指数日线图

根据缠论对于盘整走势的定义，选取 2013 年 12 月 5 日至 2014 年 8 月 8 日的上证指数的走势作为盘整走势。该段时间内的股价走势都被包含在一个走势中枢当中。

我们将对每种走势都用逻辑回归算法、KNN 算法、朴素贝叶斯算法、SVM 算法、神经网络算法、决策树算法、随机森林算法进行验证，试图找出每种走势下的最佳算法。

我们的输入变量为前一日的开盘价、收盘价、最高价、最低价、成交量、成交金额，而输出变量有两种需要分别训练。第一种情况是预测第二天的涨跌情况，第二种情况是预测第二天是高开还是低开。这两种情况，第一种情况中的涨用 1 表示，跌用 0 表示。第二种情况中的高开用 1 表示，低开用 0 表示。在上升以及下降走势当中，前 1/2 的数据包含了一个完整的走势中枢以及上升或下降阶段，而后 1/2 的走势几乎是前 1/2 走势的再次重现。而对于盘整走势，无论在盘整走势何种阶段，走势都被包含在了一个中枢当中，没有其他的特征，因此，对于三种走势，我们选取的训练集统一为前 1/2 数据，测试集统一为后 1/2 数据。所用的编程语言为 Python。本文的研究对象仅限于中国 A 股市场，因此本文的结论也仅针对于中国 A 股市场而言。机器学习算法在其他市场上的量化投资表现有待进一步研究。

4.2 股市各走势类型效果比较

根据缠论的走势分类，我们将走势分为三种类型，下表为机器学习诸算法在各走势类型中的效果比较。

其中的 DT 代表决策树算法、BP 代表神经网络算法、NB 代表朴素贝叶斯算法、KNN 代表 K 近邻算法、LR 代表逻辑回归算法、RF 代表随机森林算法、SVM 代表支持向量机算法。

我们选取的数据集部分情况如下两表所示：

表 4.1 2013 年 12 月 5 日至 2013 年 12 月 31 日上证指数涨跌预测部分训练集

	开盘价	最低价	最高价	收盘价	成交量（亿）	成交额（亿）	次日涨跌
0	2252	2239	2255	2247	122	1058	跌
1	2242	2228	2248	2237	104	923	跌
2	2241	2231	2249	2238	93	844	跌
3	2239	2231	2250	2237	109	955	跌
4	2228	2193	2228	2204	109	943	涨
5	2199	2194	2214	2102	86	760	涨
6	2188	2183	2204	2196	85	726	跌
7	2197	2159	2201	2160	101	864	跌
8	2161	2146	2165	2151	78	669	跌
9	2150	2142	2156	2148	64	554	跌
10	2153	2126	2159	2127	73	614	跌
11	2128	2082	2131	2084	86	721	涨
12	2089	2068	2099	2089	68	584	跌
13	2094	2079	2111	2092	70	616	涨
14	2095	2088	2107	2106	65	581	跌
15	2102	2070	2102	2073	78	667	涨
16	2074	2069	2110	2101	76	663	跌
17	2108	2095	2112	2097	74	630	涨
18	2090	2087	2120	2115	80	714	跌

以上为 2013 年 12 月 5 日至 2013 年 12 月 31 日的上证指数数据集。也是作为部分训练集。其预测的是大盘第二天的涨跌情况。

而光预测第二天的涨跌还不足够，还需要预测第二天是高开还是低开，这样

才能方便我们做出投资决策。

下表展示的是预测高开与低开的部分数据集情况：

表 4.2 2013 年 12 月 5 日至 2013 年 12 月 31 日上证指数高低开预测部分训练集

	开盘价	最低价	最高价	收盘价	成交量（亿）	成交额（亿）	次日开盘情况
0	2252	2239	2255	2247	122	1058	低开
1	2242	2228	2248	2237	104	923	高开
2	2241	2231	2249	2238	93	844	高开
3	2239	2231	2250	2237	109	955	低开
4	2228	2193	2228	2204	109	943	低开
5	2199	2194	2214	2102	86	760	低开
6	2188	2183	2204	2196	85	726	高开
7	2197	2159	2201	2160	101	864	高开
8	2161	2146	2165	2151	78	669	低开
9	2150	2142	2156	2148	64	554	高开
10	2153	2126	2159	2127	73	614	高开
11	2128	2082	2131	2084	86	721	高开
12	2089	2068	2099	2089	68	584	高开
13	2094	2079	2111	2092	70	616	高开
14	2095	2088	2107	2106	65	581	低开
15	2102	2070	2102	2073	78	667	高开
16	2074	2069	2110	2101	76	663	高开
17	2108	2095	2112	2097	74	630	低开
18	2090	2087	2120	2115	80	714	低开

以上是高低开预测情况。如果我们预测第二天是高开，并且第二天会涨，则我们可以在当日尾盘的时候及时买入，次日继续持有；如果预测第二天是高开，但第二天会跌，则我们可以在当日尾盘的时候及时买入，次日高开以后立刻卖出；如果预测第二天是低开，但是会涨，则考虑在第二天开盘的时候进行买入操作或

者不进行操作，因为中国股市是 T+1 交易，次日开盘买，即使尾盘时上升的，但也有可能次日低开，因此谨慎起见最好不进行操作；如果预测第二天是低开并且会跌，仍然不进行操作。因此我们操作与否的判断基准是高开还是低开，如果是高开我们才进行尾盘时分的买入，如果预测第二天是跌则次日开盘卖出，预测次日涨则继续持有，第二日结束时，如预测第三日是低开则在第二日的尾盘时分进行卖出，如果是预测第三日是高开但会跌则在第三日的开盘卖出，如果仍然预测第三日高开上涨才继续持有。

以上是高低开预测情况对于操作的指导。而我们在进行操作指导前应该先选定恰当的机器学习算法，以下是各机器学习算法在各走势类型当中的具体表现。

首先需要确定机器学习算法对于涨跌预测的准确率，如下表所示：

表 4.3 机器学习各算法对于上证指数各走势的涨跌预测准确率

算法类型 / 走势类型	DT	BP	NB	KNN	LR	RF	SVM
上升趋势	0.62	0.29	0.28	0.39	0.59	0.61	0.72
下降趋势	0.47	0.47	0.43	0.42	0.43	0.57	0.43
盘整走势	0.49	0.53	0.51	0.44	0.53	0.44	0.43

从表中可以分析出，在上升趋势当中，表现最佳的算法是支持向量机算法。该算法能够将涨跌的预测准确率提高到 72%，决策树算法、逻辑回归算法与随机森林算法的表现也较为良好，能够达到 60%左右，其他算法表现欠佳。

而在下降趋势当中，各大算法都表现平平，较为突出的是随机森林算法，能够达到 57%的准确率。

在盘整走势当中，所有机器学习算法的表现都并不优良，因此在盘整走势当中应该谨慎用机器学习算法进行量化投资的指导，此时更应该结合其他技术分析理论进行投资决策。

表 4.4 机器学习各算法对于上证指数各走势的高低开准确率比较

算法类型 / 走势类型	DT	BP	NB	KNN	LR	RF	SVM
上升趋势	0.70	0.57	0.30	0.33	0.51	0.33	0.70

下降趋势	0.59	0.62	0.41	0.62	0.59	0.41	0.59
盘整走势	0.58	0.47	0.41	0.60	0.75	0.60	0.75

上表示机器学习各算法在各走势中对于高低开的准确率比较。

从表中可以看出，在上升趋势当中，决策树算法与支持向量机算法的准确率都较高，能够达到 70%的准确率，朴素贝叶斯算法、KNN 算法以及随机森林算法此时都表现欠佳。

在下降趋势当中，除了朴素贝叶斯与随机森林算法表现欠佳，其他算法都能够达到 60%左右的准确率，神经网络算法能够达到最高的 62%。

在盘整走势当中，逻辑回归算法与支持向量机算法有着非常优秀的表现，能够使得准确率达到 75%。

5. 不同品种下的机器学习诸算法效果比较

5.1 A 股主要品种划分

本文主要以上市 A 股作为研究对象，上市 A 股分为多种品种，每种品种都会有自己相应的特性。本文将上市 A 股分为：工业、农业、服务业、高科技产业四大块。

工业主要包括了建筑材料、有色金属、采掘业、军事工业等块。

农业主要包括了种植业、林业、养殖业等块。

服务业主要包括了交通运输业、航运业、物流等业。

高科技产业主要包括了新能源汽车、无人机等业。

对于每种大行业中的次级行业，每个次级行业里面选取三支股票作为标的。三支股票的选取原则为按照该类股票的权重进行选取。即把该类型股票的权重按照从大到小排列，选取权重最大、权重中等以及权重最小的三支股票作为该行业的代表，以避免所选取的股票具有特殊性而使得最终的适应性度量也具有特殊适应性。计算每个标的股票的准确率，再计算平均数，之后再对每种次级行业计算平均数，作为大行业的准确率评估。

测试时间为包含了上升趋势，下降趋势，盘整走势三种时间段，并且取三者的平均值作为最终的准确率。

对于工业股，选取了翼东水泥、海螺型材等共计 12 支股票。

对于农业股，选取了农发种业、中牧股份等共计 9 支股票。

对于服务业，选取了楚天高速、东莞控股等共计 9 支股票。

对于高科技产业，选取了大唐发电、创业环保等工业 6 支股票。

交易时间为 2014 年 11 月 20 日至 2015 年 6 月 15 日的上升趋势、2011 年 4 月 15 日至 2012 年 1 月 6 日的下降趋势、2013 年 12 月 5 日至 2014 年 8 月 8 日的盘整趋势，共计约 23 个月的交易时间段。

我们选取的自变量为股票的开盘价、收盘价、最高价、最低价、成交量、成交价，因变量有两个，第一个为次日股票的涨跌情况，涨记为 1、跌记为 0。第

二个为次日股票是高开还是低开，高开记为 1，低开记为 0。训练集为每种走势类型的前一半时间，测试集为每种走势类型的后一半时间。

5.2 机器学习各算法在不同品种中的表现

首先需要比较的是机器学习各算法对于四大产业的第二日涨跌预测准确率，如下表所示：

表 5.1 机器学习各算法对于各品种的涨跌预测准确率比较

算法类型 / 品种	DT	BP	NB	KNN	LR	RF	SVM
工业	0.57	0.58	0.63	0.56	0.58	0.52	0.58
农业	0.55	0.54	0.57	0.52	0.56	0.53	0.51
服务业	0.50	0.49	0.55	0.50	0.59	0.48	0.54
高科技产业	0.51	0.54	0.52	0.54	0.58	0.55	0.61

从上表可以看出，用机器学习各大算法对工业、农业、服务业、高科技产业的次日涨跌预测准确率要相对比大盘的涨跌预测的最高准确率相对更低一些。但相比较大盘预测而已，不会出现某类算法非常不适应的情况，普遍都能达到 0.5 以上的准确率。

在工业股当中，各大算法的表现都相对较为优良，其中朴素贝叶斯算法的效果最好，能够将准确率提升到 63%。而在农业股当中，仍然是朴素贝叶斯算法最为优良，逻辑回归算法与之有着非常接近的准确率。在服务业股当中，逻辑回归算法有着 59%的准确率，表现最好，其他算法表现一般。而对于高科技产业股，则支持向量机算法能将准确率提升到 61%，其他的算法表现一般。

其次还需要比较机器学习算法对于各类股票高低开的预测准确率，如下表

表 5.2 机器学习各算法对于各品种的高低开预测准确率比较

算法类型 / 品种	DT	BP	NB	KNN	LR	RF	SVM
工业	0.55	0.59	0.67	0.58	0.63	0.59	0.56
农业	0.59	0.57	0.54	0.57	0.62	0.56	0.53
服务业	0.53	0.53	0.60	0.53	0.64	0.53	0.58

高科技产业	0.54	0.56	0.54	0.55	0.60	0.53	0.63

从上表可以明显看出，逻辑回归算法无论对于任何品种都能够达到 60%以上的准确率。几乎对于所有的品种都是最优的算法。只有在高科技产业股当中支持向量机算法能够将准确率提升到 63%，也仅仅超过逻辑回归算法 3%而已。并且几乎所有的品种在所有算法当中都能够达到 50%以上的准确率。

6. 利用机器学习算法构建自动化交易程序进行回测

前两章已比较了机器学习各大算法在不同走势下的表现，以及比较了各大算法在不同投资品种中的表现。接下来需要根据机器学习算法编写自动化交易程序，实现量化交易，并且查看策略的各项属性表现。

交易规则的原始设定：

我们比较的预测项有次日的高低开以及次日的涨跌情况。如果我们预测次日是高开，那么便需要在尾盘的时候进行买入操作，如果预测次日是低开，则不进行操作，即使预测次日会上涨，但由于存在低开，无法判断次日低开的上涨是否能够弥补低开的缺口，因此保险起见我们不进行操作。因此买入规则的确立就是预测次日高开，则在尾盘买入。在有持仓的情况下，如果预测次日高开上涨，则继续持有；如果预测次日高开下跌，则在次日尾盘时进行卖出操作；如果预测次日低开上涨，则在尾盘进行卖出操作；如果预测次日低开下跌，则在尾盘进行卖出操作。

此是利用机器学习算法对于投资的具体指导，此操作方法目前只能用手工操作进行买卖的实现。如果用 Python 语言在优矿网进行自动化交易程序的编写，由于版本与技术的原因，目前在该网站所编写的程序只能实现在每日刚开盘，也就是 9:30 分的时候网站自动运行一次，因此目前无法实现尾盘自动化交易，只能实现开盘自动化交易。

因此用 Python 语言编写的自动化交易程序的交易逻辑就需要变为：

预测次日涨跌情况，如果预测次日是涨的，则在次日开盘时进行买入操作，如果预测次日是跌的，则在次日开盘时进行卖出操作。

该交易逻辑只考虑了涨跌情况，没有考虑高低开的问题，存在着一些缺陷。但仍能用于检验各大算法的回测情况。

根据之前的结论，我们得知，在上升趋势当中，支持向量机算法能够有最优秀的表现，在下降趋势当中，随机森林算法能够有最优秀的表现，在盘整走势中，各大算法表现平平，逻辑回归算法表现稍微优良一些。因此我们对于三种趋势，我们的股票池就选用包括工业、农业、服务业、高科技产业在内的共计 36 支股票作为股票池进行交易。在现实操作中，我们并不能 100%判定当下究竟是上升

趋势、下降趋势还是盘整趋势，但我们可以根据均线来进行大体走势的研判。具体操作当中，我们可以以 60 日均线作为参考线，当 K 线向上连续穿越 30 日均线以及 60 日均线并且持续三日不跌破，我们可以认为此时开始了上升趋势。当 K 线向下穿越 30 日均线以及 60 日均线并且持续三日不升破，我们可以认为此时开始了下降趋势。当 K 线处于 30 日均线与 60 日均线之间，我们可以认为此时处于盘整走势。这种判断方法并不是 100%的准确，但是能够以大概率的形式拟合到当下的大体走势类型。趋势一旦形成，并不会轻易改变，因此机器学习算法的适应性会在一定时间内有效。

首先测试上升趋势当中，我们选用支持向量机作为交易逻辑进行自动化交易程序的设计，其在上升趋势当中的表现如下：

图 6.1 2015 年 4 月 15 日至 2015 年 6 月 15 日 SVM 策略测试情况

根据缠论，本文选取的上升趋势为 2014 年 11 月 20 日至 2015 年 6 月 15 日，我们取 2014 年 11 月 20 日至 2015 年 4 月 15 日的交易数据作为训练数据，利用支持向量机算法训练出交易模型，再将该模型用于 2015 年 4 月 15 日至 2015 年 6 月 15 日进行交易测试。

从图中可以看出，在上升趋势当中，合理运用机器学习算法，能够超越基准年化收益率。中国股市在该段时间内处于一种火热上涨的阶段，因此即使是基准年化收益率也达到了 157.2%之高。此阶段几乎买各种股票都能将本金翻倍。如果用机器学习算法进行投资指导，能够达到锦上添花的作用。并且可以看到，该投资组合的贝塔仅仅只有 0.89，属于风险较低的投资组合。而其缺点是收益波

动率太高，达到了 51.3%，并不算很稳定。最大回撤虽然位 14.9%，在火热上涨阶段，低于 20%的回撤便不算太高，是能够接受的范围。

接下来需要测试的是在下降阶段当中用机器学习算法进行指导的具体表现。

根据本文之前的结论，在下降当中，我们选用随机森林算法进行自动化交易模型的设计，如下图所示：

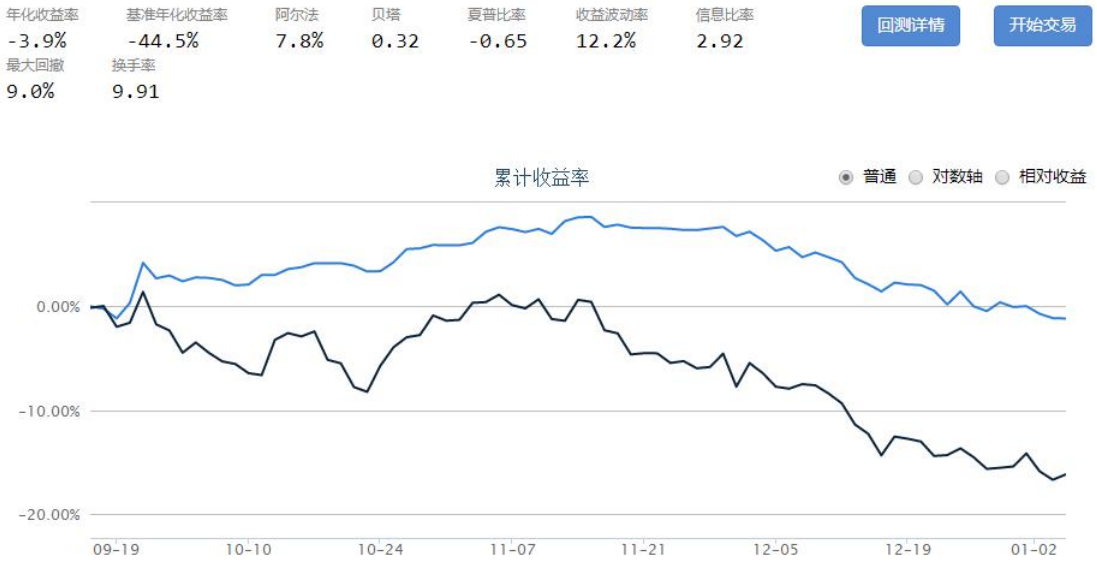


图 6.2 2011 年 9 月 15 日至 2012 年 1 月 6 日随机森林策略测试情况

从图中可以看出，在下降趋势当中，即使是使用机器学习算法，也没有产生正的年化收益。但如果合理使用机器学习算法，能够使得亏损大大降低。此时的年化收益率为-44.5%，但如果使用机器学习算法，能够将年化收益率损失降低到-3.9%的程度，几乎没有很大的亏损。并且从图中可以看出，在 2011 年 10 月待 2011 年 12 月这段时间里面，运用机器学习算法进行投资还产生了较为客观的正收益，如果进行及时止损的处理，即使在下降趋势当中也很可能产生正的收益。

接下来需要测试的是盘整阶段机器学习算法的表现。

根据本文的结论，在盘整当中，逻辑回归算法的表现比较优秀，因此我们选用逻辑回归算法进行盘整阶段的策略设计，如下图：

图 6.3 2014 年 5 月 8 日至 2014 年 8 月 8 日逻辑回归策略

从图中的表现来看，在盘整阶段使用机器学习算法并没有产生太多正面的效果，与基准年化收益率曲线相比，运用机器学习算法的收益率曲线总是与之相比并没有显著的孰高孰低。

而从之前的测试数据也可以看出，在盘整阶段，各大机器学习算法的表现也都一般般，接近 50%的成功率，因此在盘整阶段并不建议投资者运用机器学习算法进行投资指导。

7. 结论与展望

在上文当中，我们根据缠论分出了 3 种股票价格走势，分别是上涨趋势，下跌趋势，和盘整走势。每种走势都测试了用机器学习各大算法对于次日高低开以及次日涨跌情况预测的准确率，根据测试结果显示，对于股票次日涨跌准确率情况预测：

在上涨走势当中，表现最佳的算法是支持向量机算法。该算法能够将涨跌的预测准确率提高到 72%，决策树算法、逻辑回归算法与随机森林算法的表现也较为良好，能够达到 60%左右，其他算法表现欠佳。

而在下降趋势当中，各大算法都表现平平，较为突出的是随机森林算法，能够达到 57%的准确率。

在盘整走势当中，所有机器学习算法的表现都并不优良，因此在盘整走势当中应该谨慎用机器学习算法进行量化投资的指导，此时更应该结合其他技术分析理论进行投资决策。

而对于股票次日高低开的准确率测试结果则显示：

在上升趋势当中，决策树算法与支持向量机算法的准确率都较高，能够达到 70%的准确率，朴素贝叶斯算法、KNN 算法以及随机森林算法此时都表现欠佳。

在下降趋势当中，除了朴素贝叶斯与随机森林算法表现欠佳，其他算法都能够达到 60%左右的准确率，神经网络算法能够达到最高的 62%。

在盘整走势当中，逻辑回归算法与支持向量机算法有着非常优秀的表现，能够使得准确率达到 75%。

之后又分别对于工业股、农业股、服务业股、高科技产业股分别进行了次日涨跌以及次日高低开的准确率预测。

对于次日涨跌预测的准确率测试结果显示：

在工业股当中，各大算法的表现都相对较为优良，其中朴素贝叶斯算法的效果最好，能够将准确率提升到 63%。而在农业股当中，仍然是朴素贝叶斯算法最为优良，逻辑回归算法与之有着非常接近的准确率。在服务业股当中，逻辑回归算法有着 59%的准确率，表现最好，其他算法表现一般。而对于高科技产业股，则支持向量机算法能将准确率提升到 61%，其他的算法表现一般。

而对于次日高低开预测的准确率测试结果显示：

逻辑回归算法无论对于任何品种都能够达到 60%以上的准确率。几乎对于所有的品种都是最优的算法。只有在高科技产业股当中支持向量机算法能够将准确率提升到 63%，也仅仅超过逻辑回归算法 3%而已。并且几乎所有的品种在所有算法当中都能够达到 50%以上的准确率。

本文试图通过机器学习算法构建自动化交易程序，但由于版本与技术的原因，无法在优矿网实现尾盘自动化交易。而用机器学习算法指导操作需要结合对于次日的涨跌预测以及高低开预测，有时会需要在尾盘的时候实现自动化交易，因此本文最大的展望是能够在技术上实现尾盘自动化交易，将次日涨跌预测及高低开预测两相结合指导操作。目前最大的不足之处在于只能实现每日开盘时的自动化操作，因此在自动化交易的程序设计当中只用到了机器学习算法对于次日股价涨跌预测的情况。这也将是今后在这一块内容的研究当中需要重点攻克的难题。

参考文献

[1] Lukac, Brorsen, Irwin. A comparison of twelve technical trading systems with market efficiency implications[J].Station bulletin-Dept.of Agricultural Economics, Purdue University, Agricultural Experiment Station (USA), 1986.

[2] Neftci S N. Naive trading rules in financial markets and wiener-kolmogorov prediction theory: a study of "technical analysis"[J]. Journal of Business, 1991: 512-522.

[3] Chopra N, Lakonishok J, Ritter J. Measuring abnormal performance: Does the market overreact[J]. Journal of Financial Economics, 1992, 32: 136-189.

[4] Jegadeesh N,Titman S. Returns to Buying Winners and Selling Losers: mplications for Stock Market Efficiency.[J]. Journal of Finance，1993，48(48):65-91.

[5] Kim K. Financial time series forecasting using support vector machines[J].Neuro-computing, 2003. 55(1): 300-321.

[6] Khan A U, Bandopadhyaya T K, Sharma S. Comparisons of Stock Rates predicti-on. Accuracy Using Different Technical Indicators with Backpropagation Neural Network and Genetic Algorithm Based Backpropagation Neural Network[C]// Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on. IEEE, 2008:575-580.

[7] Nair B B, Dharini N M, Mohandas V P. A Stock Market Trend Prediction System Using a Hybird Decision Tree-Neuro-Fuzzy System[C]// Advances in Recent Technologies in Communication and Computing (ARTCom), 2010 International Conference on. IEEE, 2010:321-335.

[8] Lee, Jun S, Jeong, et al. Trading Strategies based on Pattern Recognition in Stock Futures Market using Dynamic Time Warping Algorithm[J]. Journal of Conve-rgence Information Technology, 2012, 7(10):140-177.

[9] Ticknor J L. A Bayesian regularized artificial neural network for stock market for-ecasting [J]. Expert Systems with Applications, 2013, 40(14): 5453-5591.

[10] Takeuchi L, Lee Y Y A. Applying Deep Learning to Enhance Momentum Trading

Strategies in Stocks[J].The Journal of Finance,2013,41(16):5-16.

[11] 吴微,陈维强,刘波.用 BP 神经网络预测股票市场涨跌[J].大连理工大学学报, 2001，41(1):8-18.

[12] 彭丽芳,孟志青,姜华等.基于时间序列的支持向量机在股票预测中的应用[J]. 计算技术与自动化,2006, 25(3): 70-93.

[13] 苏治, 傅晓媛. 核主成分遗传算法与 SVR 选股模型改进[J]. 统计研究，2013，30(5):7-12.

[14] 曹正凤,纪宏,谢邦昌.使用随机森林算法实现优质股票的选择[J].首都经济贸易大学学报,2014, 16(2):19-29.

[15] 赵志勇,王峰,李元香.基于深度学习的股票市场预测[J]. 武汉大学学报, 2014 39(1):1-7.

[16] 张炜,范年柏,汪文佳.基于自适应遗传算法的股票预测模型研究[J]. 计算机工程与应用, 2015, 51(4).

[17] 吴耿锋.股票短期预测的一种非线性方法[J].上海:上海投资,1999.

[18] 镇磊.基于高频数据处理方法对 A 股算法交易优化决策的量化分析研究[J]. 中国科学技术大学出版社,2010.

[19] 周琳杰.中国股票市场动量策略赢利性研究[J].世界经济,2002(08):15-27.

[20] 冯平.遗传算法在股票投资技术分析中的应用[J].合肥:预测,2001,第 2 期.

[21] Friedman J H. Greedy Function Approximation: A Gradient Boosting Machine[J]. Annals of Statistics, 2000, 29(5):1189--1232.

[22] Kang J, Liu M H, Ni S X. Contrarian and momentum strategies in the China stock market: 1993-2000[J]. Pacific-Basin Finance Journal, 2002, 10(3):154-231.

[23] Agrawal R, Faloutsos C, Swami A. Efficient similarity search in sequence databases[J]. Lecture Notes in Computer Science, 1993:54-56.

[24] Johan Blokker. The Application of SVM to Algorithmic Trading[J]. CS229 Term Project. Stanford University. 2008.

[25] Sanjay Ghemawat Howard Gobioff and Shun-Tak Leung. The Google File System[J].ACM SIGOPS Operating Systems Review. 2003.

[26] Scharftein D.,Stein J..Herd Behavior and Investment[J].The American Economic

Review，1990：352-432.

[27] 金德宝. 基于支持向量机的股市预测研究[J].浙江大学出版社，2010.

[28] 沈晨. 基于神经网络的股价操纵行为实证研究[J].上海交通大学出版社，2011.

[29] 袁增任.人工神经网络及其应用[J].北京:清华大学出版社,1992.

[30] 膜志纯.利用双 BP 算法提高 BP 网络的泛化能力[J].模式识别与人工智能,1995.

[31] 王东升.神经网络与神经计算机[M].北京:科学出版社,1997, 156-176.

[32] 崔世春.卖空的艺术[M].上海:上海财经大学出版社,2002.

[33] Narasimhan Jegadeesh,Sheridan Titman.Returns to buying winners and selling losers: implication for stock market efficiency[J]. Journal of Finance, 1993(48): 31-42.

[34] LeBaron B. Nonlinear dynamics and economic[J].Journal of Business 1990(55): 1245-1424.

[35]江南小隐.图解缠论[M].中国宇航出版社,2012,21-37.

附录 Python 量化交易部分源代码

```
from sklearn import metrics
import numpy as np
import time
start = '2014-05-08'                          # 回测起始时间
end = '2014-08-08'                            # 回测结束时间
benchmark = 'HS300'                           # 策略参考标准
universe  = ['000401.XSHE'] # 证券池，根据所选股票输入相应代码。
capital_base = 100000                         # 起始资金
freq = 'd'                                   # 策略类型，'d'表示日间策略使用日线回测，'m'表示日内策略使用分钟线
refresh_rate = 1                              # 调仓频率
def random_forest_classifier(train_x, train_y): #定义随机森林算法，其他算法的定义与此类似
    from sklearn.ensemble import RandomForestClassifier
    model = RandomForestClassifier(n_estimators=8)
    model.fit(train_x, train_y)
    return model

def predict(a):                #定义预测模型函数，在交易函数阶段进行调用
    Index=DataAPI.MktEqudGet(tradeDate=u"",secID=u"",ticker=u"000401",beginDate=u"20131205",endDate=u"20140508",isOpen="",field=u"closePrice,openPrice,highestPrice,lowestPrice,turnoverVol,turnoverValue,preClosePrice",pandas="1")#选取2013年12月到2014年5月8日的股票数据作为训练数据。
    x=Index[['openPrice','closePrice','highestPrice','lowestPrice','turnoverVol','turnoverValue']].values
```

```
    y=(Index['closePrice']>Index['openPrice']).values
    X=x[0:len(x)-2]
    Y=y[1:len(x)-1]
    model=random_forest_classifier(X, Y)
    YC=model.predict(a)
    return YC
def initialize(account):                    #初始化交易函数
    pass

def handle_data(account):                   #定义交易函数
#取得开盘价、收盘价、最高价、最低价、成交量、成交额等数据作为自变量。
    hist1 = account.get_attribute_history('openPrice', 30)
    hist2 = account.get_attribute_history('closePrice', 30)
    hist3 = account.get_attribute_history('highPrice', 30)
    hist4 = account.get_attribute_history('lowPrice', 30)
    hist5 = account.get_attribute_history('turnoverVol', 30)
    hist6 = account.get_attribute_history('turnoverValue', 30)
    for stock in account.universe:
        T1=hist1[stock][-1]
        T2=hist2[stock][-1]
        T3=hist3[stock][-1]
        T4=hist4[stock][-1]
        T5=hist5[stock][-1]
        T6=hist6[stock][-1]
        T7=[T1,T2,T3,T4,T5,T6]
#将取得的昨日的开盘价、收盘价、最高价、最低价、成交量、成交额作为输入变量带入已训练好的模型当中进行预测。
        sound=predict(T7)
#根据预测结果发出买卖指令。
```

```
    if sound >0    :
        order(stock, 1000)
    if sound ==0   :
        order_to(stock, 0)
```

Return#以上代码仅适用于量化平台“优矿网”。

致谢

时光飞逝，转眼就到了硕士毕业的时间。在暨南大学的两年时光里让我成长了许多。首先需要特别感谢姜云卢导师在两年时光中对我学习与生活中的许多帮助。姜云卢导师对于机器学习的诸多算法都有丰富的经验，在这方面给予了我非常多的指导。在导师的多次指导下让我完成了硕士毕业论文的撰写。同时还需要感谢暨大经济学院统计系的诸多教授。他们在课堂以及课后都让我在统计学方面的技能得到提升，培养了我对于统计学的兴趣，并且让我决定在以后的人生中要继续进行统计相关的科研与工作。还要感谢我在暨大遇到的各位同学，在学习中通过讨论班的形式相互提升，在生活中也互相帮助，感受到集体的温暖。最后需要感谢我的父母，在生活中给我的大力支持，以及对于我继续统计学方面研究的大力支持。

谢翔

2017 年 4 月 7 日

阅读此文的还阅读了：

26. 机器学习服务比较:谷歌、微软、亚马逊

27. 多侧面递进算法在机器学习中的应用

28. 不同学历新生学习适应性的比较

29. 开源机器学习Weka环境下的数据分类算法实现及应用

30. 机器学习算法应用浅析

31. 基于机器学习算法的网络入侵检测

32. 三种经典作业调度算法适应性比较

33. 中药指纹图谱识别的机器学习算法研究

34. 深度学习算法在智能协作机器人方面的应用

35. 超高效能的量子机器学习算法

36. 基于机器学习的可降解支架检测与分割算法

37. 基于机器学习算法的大数据处理

38. 大数据背景下机器学习算法的综述

39. 机器学习——我们该如何与机器竞争

40. 基于机器学习对优质股的选择

41. Eigenface算法与EBGM算法的适应性比较

42. 大数据下的机器学习算法探讨

43. 网络入侵检测的机器学习算法评估与比较

44. 大数据环境下机器学习算法趋势研究

45. 基于机器学习聚类算法的学习者自动分类研究

46. 机器学习算法原理及效率分析

47. 机器学习算法 在无人驾驶中的应用

48. 理解机器学习从理论到算法

49. 机器学习服务比较：谷歌、微软、亚马逊

50. 机器学习算法在焊接领域中的应用