分类号 密级

中国地质大学（北京）

硕士学位论文

量化投资选股模型的研究与应用

学　　号：2019150006

研 究 生：李 洋

专　　业：数 学

研 究 方 向：金融数学

指 导 教 师：黄光东 副教授

2018 年 5 月

A Dissertation Submitted to

China University of Geosciences for Master Degree

Research and Application of Quantitative Investment Selecting Stock Model

Master Candidate: Li Yang

Major: Mathematics

Study Orientation: Financial Mathematics

Dissertation Supervisor: A.Prof.Huang Guangdong

China University of Geosciences (Beijing)

声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，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，也不包含为获得中国地质大学或其它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了谢意。

签　名：_____日　期：_____

关于论文使用授权的说明

本人完全了解中国地质大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

□公开　□保密（____年）　**(保密的论文在解密后应遵守此规定)**

签　名：_____导师签名：_____日　期：_____

摘　要

量化投资是利用数学模型和计算机程序来预测股票价格的走势，由于其自身的客观性，快速性，纪律性等优点越来越受投资者的钟爱。而且在未完全成熟的中国市场，市场信息不对称和市场失灵现象较多，同时大部分投资者的投资理念相对落后，投资水平也是参差不齐。因此，量化投资模型的研究对于国家经济宏观政策的调整和个人投资都有重要的指导意义。

股票价格走势的预测问题可以作为分类回归问题来研究，而解决分类回归问题在机器学习算法领域已经有不错的解决方法。本文受启发于 XGBoost 算法在各种数据挖掘比赛中的大放异彩，同时在现有的量化投资选股模型的研究上，考虑到有的选股特征因子包含的信息量可能很少，但它又有部分特别关键的信息，如果直接用 XGBoost 算法模型，可能在做特征分裂的时候会忽略掉这些因子。并且特征因子之间是有相关性的，信息也是有重复的。

考虑到上诉问题，本文加入主成分分析方法(PCA)对 XGBoost 算法进行改进，将 PCA 算法与 XGBoost 算法相结合构成新的 P-XGBoost 算法模型，新的 P-XGboost 模型在综合特征选股因子的主要信息的同时，也能降低特征因子维度。同时，基于 P-XGBoost 模型本身的算法特性，其拥有两个特别好的优点。一是在分裂特征因子选取时计算速度更快，样本之间以及特征因子之间无任何相关性，可实现并行计算，与其他模型相比，在计算速度上有很大的提升；二是在算法中加入了正则项，防止了模型出现过拟合的问题，使得模型的泛化能力更强。

最后将 P-XGBoost 模型作为量化投资选股模型运用在股票市场中，并比较了 P-XGBoost 模型和改进前的 XGBoost 模型在股市中的表现结果，提出了 P-XGBoost 模型对股票价格走势预测的有效可行性。

关键词：量化投资，分类回归问题，选股特征因子，P-XGBoost 模型

Abstract

Quantitative investment is the use of mathematical models and computer programs to predict the trend of stock prices. Because of its own objectivity, rapidity, discipline, and other advantages are increasingly favored by investors. Moreover, in the not fully matured Chinese market, there are more market information asymmetries and market failures. At the same time, the investment philosophy of most investors is relatively backward and the level of investment is also uneven. Therefore, the study of quantitative investment models has important guiding significance for the adjustment of national macroeconomic policies and personal investment.

The prediction of the stock price trend can be studied as a classification regression problem, and solving the classification regression problem has a good solution in the field of machine learning algorithms. This article is inspired by the XGBoost algorithm's brilliant performance in various data mining contests. At the same time, in the study of the existing quantitative investment stock selection model, it is considered that some stock selection feature factors may contain very little information, but it There are also some particularly critical pieces of information. If you use the XGBoost algorithm model directly, you may ignore these factors when performing feature splitting. And there are correlations between the feature factors and the information is also repeated.

Considering the appeal problem, this paper adds Principal Component Analysis (PCA) to improve the XGBoost algorithm. The PCA algorithm and XGBoost algorithm are combined to form a new P-XGBoost algorithm model. The new P-XGboost model is used to synthesize feature selection factors. The main information can also reduce the feature factor dimension. At the same time, based on the characteristics of the P-XGBoost model itself, it has two particularly good advantages. One is that when the splitting feature factor is selected, the calculation speed is faster, there is no correlation between the samples and the feature factors, and parallel calculation can be realized. Compared with other models, the calculation speed is greatly improved; Regular terms are added to the algorithm to prevent overfitting of the model and make the model more

generalizable.

Finally, the P-XGBoost model was used as a quantitative investment selection model in the stock market. The P-XGBoost model and the improved XGBoost model were compared in the stock market. The P-XGBoost model was used to forecast the stock price trend. Effective and feasible.

Keywords: Quantified investment，Classified regression problem，Stock selection feature factor，P-XGBoost model

目 录

第一章 引言

1.1 研究背景及意义

股神-巴菲特曾经创造的连续保持32年战胜市场的记录竟然被一个基金经理-詹姆斯西蒙斯打破了，而能让这个基金经理打破巴菲特神话记录的就是量化投资。量化投资是一种投资策略，利用现代统计和数学从大量历史数据中发现和获得超额收益。投资者使用计算机程序严格遵循这些策略构建的量化模型来投资并产生回报。量化投资始于20世纪50年代，最早由马克维茨首次提出，使用具有最佳均值方差的数学方法来选择最优投资组合。但是，量化投资需要复杂的数据处理，直到最近二十年，计算机被广泛使用之后其才得以迅速的发展。

2008 年到 2010 年间，金融危机的到来打击了全球金融市场。几乎所有股票市场的投资者都遭受了重大损失，所有的基金都遭受了或多或少的损失。2010 年以后，经济逐步复苏，但仍不稳定，受到不同程度的冲击。 在这次金融危机中，价值投资和千篇一律的投资似乎是无效的，量化投资却确保了高稳定的地位，并在资本市场取得了良好的效果。比如打败沃伦巴菲特的西蒙斯在 2008 年金融风暴中实现了 160％的量化投资回报。

量化选股在当今的金融行业变得越来越重要，毕竟量化投资的回报率很高。尽管已经有许多成功的量化投资案例，但是人们对它的运作方法依然不是很了解，还处在一个模糊的状态中。可能在量化投资定义中有它是通过计算设计，程序实现等部分，人们会觉得量化投资是计算机活动。其实量化投资是人类的活动而不是计算机，是人类通过大量的实际数据及理论分析研究后，建立交易模型然后编写交易模型的程序在计算机中实现。量化投资通过发现市场错误估计和寻找赢得机会找到超额收益。量化投资在以下几个方面有优势：（1）快速性：数量投资依靠计算机来比人脑更快地挖掘市场信息。（2）纪律性，量化投资是在无人干预的情况下发生的。（3）客观性，量化投资是通过计算机程序对市场数据进行分析，而不受人的主观情绪的影响。从 1998 年到 2008 年，量化基金管理资产从 80 亿美元增加到 1848 美元，年均增长 15％。在全球范围内，量化投资受到越来越多的投资者青睐，量化投资已经在投资领域中有着不可撼动的地位。

近年来，量化投资引发了中国的高潮。特别是 2009 年以后，国内各大基金

公司开始大力建设量化投资团队，并推出量化基金产品。这包括中海量化基金，光大量化基金，上投摩根阿尔法，富国沪深 300 增强，嘉实阿尔法量化基金等等。在整体市场不景气的情况下，量化基金似乎开辟出了另一个投资空间。这种在中国刚刚起步的投资方式看起来很新颖，但在海外，量化投资已经走过了 40 年的历程。在过去的 20 年中，海外量化基金的数量和市值呈爆炸式增长。可以看出，近年来量化投资发展非常迅速。

在股市方面，国内股市发展较晚，非专业个人投资者一直是主要投资者。整体投资者的投资理念相对落后，投资水平也不均衡。但是，对于专业投资者而言，首先可以通过使用专业知识和自己掌握市场更好地探索它们。其次，国内资本市场上的量化投资策略竞争对手相对较少，量化投资可以通过程序更快地开拓市场投资机会，从而避免重视定性判断的传统投资方法有缺陷的能力，这将为未来的发展留出巨大空间。随着时代的发展，越来越多的公司在 A 股市场上市，越来越多的公司上市。上市企业中很难再依靠传统的定性分析选择出具有高投资性质的股票。那么投资者将如何选择具有投资价值的股票是一个围绕专家学者及业内人士的一个复杂的问题。我国的选股模型的研究还处于初级阶段，并且我国的市场信息不对称，市场失灵的现象出现的较多。在这样的不算一个有效的市场中存在很多价格错位或偏离的股票，量化投资股票选择基于所有公共信息量化统计建立相应的数学模型。在计算机上对每只股票进行分析便可自动选出符合数据模型的股票组合，当应对这种不是很有效的市场的时候，可以高效的、系统的挖掘出具有投资意义的股票。在这种情况下，量化投资的研究也更有意义。

1.2 研究现状

1.2.1 国外研究现状

在国外，量化投资方法的发展已经有将近 40 年的历史，所以国外的量化投资研究已经发展的很好了。

Fama 等(1993)研究了影响债券收益和股票的因素，结论表明影响股票收益的因素主要有与账面价值相关和企业规模风险因素以及整个市场风险因素。

Asness(1997)研究中公司的基本面数据的变化与上市公司股票价格变化有着密切相关的关系，基本面数据好，公司股价上升；基本面数据坏，股票价格下跌。在这之后更多的学者的研究成果也证实了这一观点。

Joseph D.Piotroski(2000)从基础财务数据中确定了 9 个财务指标，对个股进行打分，并对其中的财务指标进行评分。如果指标小于临界值，记为 0 分；如果指标大于临界值，则记为 1 分。把股票按照打分的高低排序然后投资组合分数较高的的若干只股票，此类投资组合具有较强的盈利能力和较低的风险，并为投资者投资提供指导。

KJ Kim 使用支持向量机(SVM)来对股票价格指数进行预测，并研究了其方法在股价预测方面的可行性，同时将该方法与神经网络对比，结果表明 SVM 在股票预测方面有不错的前景(KJ Kim，2003)。

Hassan 和 Nath 首次提出原始数据的选择不仅仅只是一维收盘价格，相反，选择一组四维数据集，包括开盘价格，收盘价格，最高价格和最低价格更有说服力。通过识别日常数据模式，查找与当前数据模式类似的历史模式，使用这个规则来预测金融行业的股票价格(Hassan R，Nath B，2005)。

Hassan，Nath 和 Kirley(2007）提出了一种人工神经网络 ANN 来处理基于四维阵列观察到的序列值。原因在于四组数据高度相关，并且通过人工神经网络降低了各组的相关性，以提高识别效果。同时利用遗传算法对模型的初始参数进行改进，使模型能够更准确地识别数据模型，提高预测精度。

Kumar, Pandey 等将和股价相关的指标用来预测股价，同时并将遗传算法加入进去和 SVM 组合，提高单个支持向量机的预测精度(L Kumar，A Pandey，S Srivastava，et al.2008)。

Kazem, Sharifi 等（2013）提出了一种新的基于混沌隐射以及萤火虫算法的支持向量机回归量化模型，并对美国市场个代表股票进行预测，取得了不错的效果。

1.2.2 国内研究现状

相对于欧美国家，中国的量化投资研究起步较晚，发展时间较短。但近年来金融改革仍在继续，政府大力推动社保基金进入A股市场，国有企业和央企降低股票比例，成立创业板等。所有这些都清楚地表明，一个更加高效和有效的市场已经逐渐走向繁荣，市场上的资本运作更加规范。中国尚未完全成熟的市场已经开展了大量的量化投资基础研究工作，大量理论模型已经建立并在市场上得到应用。

陈守东（2003）在《中国股票市场FF多因子模型的比较分析》中使用最小二乘法和广义矩估计方法，证明多因子模型适合中国股票市场。

侯雅文(2007)假定发射函数为单个正态分布，对标普指数单变量做了HMM建模并应用到标准普尔指数预测领域,同时将新建的HMM模型与神经网络预测模型进行了比较，结果显示HMM模型优于神经网络模型。

李嘉裕(2008)用神经网络，决策树等研究方法对股指期货的发展方向和渠道进行分析和预测。其中，量化的理念与传统数据挖掘方法已经被结合起来使用。彭益等人调整了拟合泊松分布和股指的上下波动，发现通过优化相关参数可以得到最佳拟合值，并用来预测股指的走势。

蔡健林（2009）以中国股市实际情况为背景，采用小波和遗传算法两个模型对加拿大皇家银行系统的价值指标选股体系中的股票数据进行了实证研究，结果表明小波算法很适合中国的市场。

李体委着重分析了中国证券市场的特点，特别是股市的投机性质以及引入外资模式时由于不完善的制度造成的干扰。当他研究FF三因素模型时，他发现中国股票收益率削弱了SMB（市值因子）因素的解释能力(李体委，2011)。

丁鹏的一本名叫《量化投资——策略与技术》的书中介绍了关于潘凡对于量化投资的多因子选股方法。其里面对30个常见选股因子的有效性进行验证，并且进行了有效因子的去冗处理，最后建立了一个只有9个有效因子的多因子选股模型。他在研究报告中重点强调了因子有效性分析和有效但冗余因子的剔除这两个部分，用该方法建立的选股模型在2005年到2010年这段时间里取得了很不错的收益(丁鹏，2012)。

周鑫（2013）将2008年的金融危机作为边界，将研究区间分为2003年至2007年和2008年至2012年两部分。然后基于Fama-French三因素模型进行改进，最终得到基于较低风险的新模型，进而增强了模型的解释力度。

段谨怡（2014）基于 Fama-French 三因素模型分析了股市价格波动，发现该模型的有效性可以通过未预期盈余得到充分的证明。并且在之后通过增加流动性指数和在中国被认为是影响股价波动的主要因素-换手率指标来比较这些异常收益的差异。

苏冰在《基于PCA-SVM模型的量化择时研究》中，将PCA和SVM结合建立量化

模型，证明了运用PCA-SVM方法提高了模型的精度，在沪深300上取的了不错的效果表现(苏冰，2015)。

李想在《基于XGBoost算法的多因子量化选股方案策划》中，第一次将XGBoost算法纳入量化选股，建立量化模型并获得了不错效果(李想，2017)。

总的来说，在中国股市要达到有效市场水平之前还有很长的路要走。 资产定价错误的现象也很普遍,但使用量化投资选股方法来建立投资组合是可以产生超额收益的。国内外学者对中国股市使用量化投资模型也是莫衷一是。

1.3 本文研究内容与方法

目前关于量化投资的研究，大多采用了多因子选股模型，当然还包括部分机器学习模型，时间序列模型等方法在内的量化模型。但它们都有自身的一些问题。本文的主要研究内容是在现有的量化投资模型的基础之上，运用最新的机器学习算法模型--XGBoost，并且将XGBoost算法和主成分分析(PCA)算法结合，构成新的P-XGBoost算法模型。并将新算法模型运用在量化选股上，得出能够在中国A股市场取得不错收益并且稳定的量化投资模型。同时，基于投资股票市场和环境的分析和构建的P-XGBoost模型，并在该模型的实现上给出一套理论的方法。最后在对比该模型与改进前的XGBoost模型在选股方面的优劣，从而为量化投资市场提供一个可行的新方向。

本文研究的方法主要是以下几个方面，首先是研究和分析当前中国 A 股市场的现有的一些选股策略和选股量化模型，了解金融市场的一些特性和规律。其次，是研究机器学习的一些算法模型，结合股票市场的特性结合多因子选股模型和机器学习算法模型选择新的算法模型，量化模型的检验需要大量的数据，本文找到数据后会用 python 对数据进行清洗，处理，再把处理好的数据带入自己写的程序模型中，得出结果，并通过图像可视化展示出来。从算法的理论推导和特性分析，数据的预处理，到模型的实现最终得出结果，并与改进前的选股模型相比较，得出模型的可行性。

1.4 本文的创新

1.本文在少有人使用可实现并行化的较新的机器学习算法XGBoost进行量化选股的基础上,以此来构建新的量化选股模型。

2.传统的多因子选股模型进一步发展了Fama-French多因子模型，基本面因

子，技术面因子和行为金融因子的研究结果被应用于股票选择模型，在选出因子后，对多个因子进行有效性检验和冗余因子剔除，最后使用排序打分法或者是多因子回归法对股票价格进行预测。而本文将主成分分析(PCA)与XGBoost算法模型结合，使用主成分分析方法（PCA）将因子处理成不相关因子，再使用XGBoost算法实现,同时PCA得到的新的特征因子是综合信息，也可弥补少量但关键的因子可能被忽略掉的缺点，从而构建成了P-XGBoost新模型，以此来提高XGBoost算法的预测准确率。

3.P-XGBoost模型由于自身算法的特性，相较于传统的量化模型在速度和泛化能力上均有优势，这也可以为量化选股的高频交易提供一个新的思路方向。

1.5 本文章节安排

本文的行文章节安排如下：

第一章是引言,主要介绍了量化投资的研究背景及其意义,国内外研究现状,并列出了本文的研究内容及研究思路方法，本文的创新之处和论文的结构。

第二章是文献的回顾和股票市场的一些相关知识,主要介绍了量化投资的发展情况，介绍目前量化投资选股的一些主要的选股方法和策略，最后分析目前量化选股存在的问题。

第三章是 P-XGBoost 算法模型的理论推导,主要介绍了 P-XGBoost 算法模型的由来，对 PCA 算法，基函数决策树和 XGBoost 算法进行了公式的推导，最后分析 P-XGBoost 算法的优点。

第四章是 P-XGBoost 算法模型在选股中的实现应用,主要介绍了数据的预处理的方法，模型的训练以及模型的评估。

第五章是 P-XGBoost 算法模型的评定,主要介绍了股票市场的风险和收益的评定方法，并对该模型的选股结果进行风险和收益的评价，最后与改进前的模型的结果作对比。

第六章是本文的研究总结分析和展望,总结分析本文的工作情况以及对后续工作的展望。

本章小结：本章主要介绍量化投资的背景及意义，量化投资的国内外的研究现状，提出本文的研究内容和研究方法，本文的创新点，以及本文的行文安排。

第二章 量化投资的相关知识

2.1 量化投资的发展

1952年，马克维兹首次建立了均值-方差模型并将其数学工具引入金融研究。此后，一门新的学科形成，成为数学金融或数学金融学。因此，数学金融是近几十年来出现的一门学科，而它作为一门学科的名称，却只有二十来年的历史。

在马克维兹工作的基础上，Sharpe(1964)、Litner(1965)、Mossin(1966)研究了资产价格的均衡结构，他们研究出的资本资产定价模型(CPAM)在如今依旧是风险衡量模型中的经典。

接下来便是Samuelson与Fama(1965)提出的有效市场假说(EMH)，这个假说包含了三个方面：有效市场，理性投资者和随机游走。该假设成立就意味着在功能齐全，信息畅通的资本市场当中，资产价格的动态规律可以用(半)鞅来表示，在购买和出售股票时，投资者可以快速有效地使用可用信息。所有已知的影响股票价格的因素都已经反映在股票的价格之中，所以，根据这一理论，股票的技术分析是无效的。EMH假说理论成为了60年代以来证券理论研究的基石。

1973年Black和Scholes建立了期权定价理论模型，这是金融衍生品定价方面的重大突破，直到现在仍在现实中有着重要的作用。

之后，Ross在1976年建立了套利定价理论(APT)。在投资实践当中，多因子定价选股模型可以看作是APT理论最典型的代表。

20 世纪 90 年代后，数学金融已经引入了更多的数学工具，其中非线性科学是 Doyne Farmer 博士和 Norman Pachard 博士最具代表性的模型之一。他们系统地阐述了李雅普诺夫指数对于混沌分类的重要性，并且在重构相空间的延迟方面有着重要的贡献。他们建立了不同的量化模型，分别使用决策树，遗传算法，神经网络和其他非线性回归的方法。令人遗憾的是，根据专有合同，瑞士银行集团拥有他们的技术，所以他们投资过程的细节和业绩记录都是其专有的财产。

如今，大部分公司将量化投资与大数据挖掘相结合，使用各种挖掘技术，运用统计，机器学习以及计算机等知识建立量化选股模型，在如今的全部投资中，量化投资占比大约为40%左右，且在投资市场获得了不错的表现。

2.2 量化投资常用方法

以下介绍现在股票市场上部分常用的量化选股方法：

1.多因子模型选股

多因子选股模型是最经典也是目前很多公司最常用的选股方法，该方法采用一系列的可能影响股票价格的有效因子（比如市盈率 PE 等）作为选股的影响标准，然后检验该因子的有效性，去重冗余因子，然后采用回归法或者是打分法得出股票组合。不满足这些因子标准的股票被卖出，满足的进行买入。例如像巴菲特这样的价值投资者就会在低 PE 的时候买入股票，而在 PE 回归的时候卖出该股票。

2.支持向量机(SVM)择时选股模型

SVM 模型作为机器学习的一个重要分支，其基于统计学习理论和结构风险化最小原理，通过将低维空间线性不可分的问题隐射到高维空间中，实现高维空间的线性可分。并且对于其每一步如何实现隐射我们不用清楚，只需要知道隐射函数(核函数)即可，因此其有效地降低了问题的复杂度，也可以一定程度上地避免过拟合的问题。因此支持向量机在量化选股中也有其一席之地，在国外，很多公司也用其实现高频交易。

3.隐式马尔可夫选股模型

隐式马尔可夫(HMM)模型最早是应用在语言识别领域，其核心的三个问题是：1.评估问题(向前-向后算法)。2.解码问题(维特比算法)。3.学习问题(EM 算法)。由于其在语言识别领域出色的表现，于是被应用在量化选股领域。并且通过前人的研究，该模型在量化择时选股中表现不俗。本文也将该模型纳入后文的选股模型实证结果对比。

4.神经网络选股模型

在机器学习领域，神经网络有着举足轻重的地位。神经网络是由大量彼此连接的节点（或者神经元）所组成的运算模型。每个节点代表一个称为激励函数的特定输出函数。每两个节点之间的连接表示信号通过连接的加权值，称为权重，相当于人工神经网络的存储器。网络的输出取决于网络连接的方式，具有不同的权重值和不同的激励功能。 网络本身通常是某种算法或函数的近似值，或者它

可能是逻辑策略的表达式。同时，神经网络又分为 BP 神经网络，深度学习（多层神经网络）等。 其在语音识别和图像识别中表现都非常好，所以也有用神经网络模型进行量化择时选股。

5.聚类算法选股模型

聚类选股模型主要是通过因子特征将股票聚为几类，然后用每一类股票组合进行数据回测得出收益率和风险评估结果，以此来选择用哪一类中的股票组合进行股票的投资。常用的聚类算法有：K-means 聚类，密度聚类，层次聚类，期望最大化聚类等。

6.时间序列类模型

时间序列模型也是量化选股中使用很广泛的。ARMA 模型，ARCH 模型,GARCH 模型等都是选股中最常见的模型。时间序列分析基于从系统观测获得的时间序列数据，并且通过曲线拟合和参数估计建立数学模型的理论和方法，以预测股票的下一阶段。

7.各种策略类选股

在股票市场中，还有一些策略类的选股方法。比如：

风格轮动策略模型：由于投资者交易行为不同(偏好价值股，偏好成长股)，形成了不同的市场风格，利用市场风格的变化，进行轮动投资。

行业轮动策略模型：行业轮动策略模型：该策略认为资产价格受内在价值的影响，而内在价值随宏观经济因素的变化而波动。在一个完整的经济周期中，一些是主导产业，一些是跟随产业。也就是说在不同的经济时期，行业的表业也不同。因此，研究一个周期内行业的轮换顺序，并在轮值开始前对其进行配置，以进行一轮投资。

资金流策略模型：该策略是认为资金流流向的股票在未来一段时间内会上涨，资金流流出的股票在未来一段时间内会下跌，以此作为选股策略。

动量反转策略：动量效应是认为某股票在前一段时间表现良好，则在接下来一段时间内仍然有不错的表现。而反转效应则认为某股票在前一段时间表现差，在接下来的时间有回到均值的需要，即表现好。市场中则有根据动量反转策略选择股票，一般来说牛市选择反转，熊市选择动量。

在现在的股票市场和量化模型研究中，大家都不单纯的使用一种模型进行建

模，而是会对模型进行算法的改进，再结合策略，行业，市场情况进行股票的选择。

2.3 量化投资选股存在的问题

尽管中国近年来量化投资规模迅速增长，但也受到各基金，证券等投资公司的关注。但由于中国的一些特点，在发展过程中遇到了一些瓶颈：首先，量化投资需要大量历史数据库进行回测，但中国股票制度建立较晚，数据量和质量不足;其次，量化投资需要股票完全由市场来控制，或者股票的变化完全是由于人共同的非理性行为而形成的。但是，中国股市受政策影响很大,而且对于因子的选取和参数的确定也没有一个统一的标准，而是根据数据回测的结果来检验的，但有时我们在回测的时候会出现过度拟合从来造成在实际中得不到我们想要的结果。

其次，在模型的实证和检验中，所需要使用到的数据繁多而且不定因素很多多。在实际情况中,财务数据的延迟不一定是一个月,可能会有更长时间的滞后，这样会造成使用未来数据的问题，这样会对模型的评估造成影响。相反，如果财务数据出现的早，由于市场有效性的存在，财务信息也会被市场充分的消化和吸收。所以，选择财务数据的使用时间对我们的模型也是很重要的，或者是将其时间作为参数加入模型。同时，在处理数据的时候，为了处理数据的方便，如果直接选择当月的最后一个交易日的收盘价作为进场的买入价，在实际交易中，这个价格不一定是可以获得的，如果当日交易出现涨停，该股票当日是无法买入的。所以，如何选择数据才更为合理也是需要考虑思索的问题。并且随着覆盖的特征越多，数据量的增多，如何面对大量的计算也是我们需要考虑的问题。

同时，在中国市场大部分人使用的量化投资选股模型也比较单一，可能无法捕捉到影响股票价格的主要因素，如何建立不同的有效选股模型，也是需要研究的问题。

本章小结：本章主要介绍了量化投资的发展历程，目前股票市场上常用的量化投资选股模型，以及分析了目前量化投资选股以及股票市场存在的一些问题。

第三章　P-XGBoost 算法模型理论

P-XGBoost 算法模型是将主成分分析(PCA)和 XGBoost 算法相结合构建成的新模型。本文将使用 PCA 算法将选股因子处理成不相关的新综合指标，新综合指标是尽可能多的包含原指标的主要信息，且每一个新综合指标的特征信息是线性无关的且无冗余信息。以此再结合 XGBoost 算法，在选取分裂点和设置树深和迭代次数的前提下，尽可能多的找到有用的主要信息。同时由于 P-XGBoost 的算法特性，可以选取囊括更多方面的选股因子，如果选取的因子过多，也可进行降维处理。这样在保证速度的前提下，也可一定程度上提高选股模型的准确率。

3.1 主成分分析方法（PCA）

PCA 算法的思想是找出数据最重要的方面，用数据最重要的方面取代原始数据。也就是将 N 维的特征映射到 k 维上（$N \geq k$），构建的 k 维特征是全新的正交特征。这个低维的线性空间被称为主子空间，并且使得投影数据的方差被最大化，即最大方差理论。且第一个特征综合因子被称为第一主成分，方差最大，以此类推后面为第二主成分等。PCA 也是一种特征融合算法，下面是 PCA 的数学推导过程。

3.1.1 PCA 的降维

假设两个向量为 $A=(x_1,y_1)^T$，$B=(x_2,y_2)^T$，则 A,B 的内积为：

$$A \bullet B = x_1 x_2 + y_1 y_2 \tag{3-1}$$

也可以表示为：

$$A \bullet B = |A||B|\cos(\theta) \tag{3-2}$$

其中 θ 为 A,B 向量的夹角。而 A 向量在 B 向量上的投影长度为 $|A|\cos(\theta)$，则 A 与 B 的内积就表示为 A 在 B 上的投影长度和 B 的模的乘积。如果 B 为单位向量，即 B 的模为 1，则 A 与 B 的内积就是 A 在 B 上的投影长度。

而在二维坐标系中，如果一个向量用坐标点表示为 $C=(3,2)^T$，其实坐标点中的 3 是向量在 x 轴上的投影长度，2 则是在 y 轴上的投影长度. 如果以 x 轴和 y 轴

上的单位向量为标准，则向量 C 则可以用 x 轴和 y 轴上的单位向量线性表示：

$$C=3(1,0)^T+2(0,1)^T \tag{3-3}$$

并且所有的二维向量都可以用这种方式表示出来。并且 $(\vec{1,0})$ 和 $(\vec{0,1})$ 是线性无关的一组基。同理，任何的两个线性无关的二维向量也可以成为一组基，所说的线性无关在二维平面内可以直观地以为是不在同一条直线上的两个向量。比如我们不以 x 轴和 y 轴为基轴，比如我们以（$\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}$）^{T}和（$-\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}$）^{T} 的方向为基轴，则 C 在该方向用矩阵表示为：

$$\begin{pmatrix}\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{pmatrix}^T\begin{pmatrix}3\\2\end{pmatrix}=\begin{pmatrix}\frac{5}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix} \tag{3-4}$$

其中矩阵两列则为两个基向量，以此方法可以推广到 m 个 n 维向量。$(a_1,a_2,\cdots a_m)$，$a_t(t=1,2,3\cdots m)$ 为 n 维向量，我们想得到其在 R 个 n 维向量表示的新空间中，则新空间的坐标表示为：

$$\begin{pmatrix}p_1^T\\ p_2^T\\ \vdots\\ p_R^T\end{pmatrix}\begin{pmatrix}a_1 & a_2 & \cdots & a_m\end{pmatrix}=\begin{pmatrix}p_1^Ta_1 & p_1^Ta_2 & \cdots & p_1^Ta_m\\ p_2^Ta_1 & p_2^Ta_2 & \cdots & p_2^Ta_m\\ \vdots & \vdots & \ddots & \vdots\\ p_R^Ta_1 & p_R^Ta_2 & \cdots & p_R^Ta_m\end{pmatrix} \tag{3-5}$$

其中 $p_t^T(t=1,2,\cdots R)$ 为 n 维向量。其中 R 可以小于 n, 并且 R 决定了变化之后的维数。即我们可以将 N 维的数据通过这种变化隐射到更低的维度上，并且变换后的维度由基的数量所决定。因此这种矩阵相乘的表示形式也可以用来表示降维变换，而 PCA 的降维也就是这个过程。

3.1.2 PCA 的方差最大化

上小节是运用线性变化，将高维空间的坐标用低维空间坐标表示。但是我们希望在保留住较多的原数据点的特性同时，以使用较少的数据维度做计算。而保留较多的原数据的信息，则需要使的方差最大。因为数据的方差越大，数据就越分散，方差越小，数据就越集中，而越分散的数据越能保留更多的信息。

假设样本集为 $X=\{x_1,x_2,\cdots,x_m\}$，样本点 x_i 在新空间中的投影为 W^Tx_i，若所有样本点的投影能尽可能分开，则应该使投影后样本点的方差最大化，投影后样本点的方差可以表示为：$\sum_{i=1}^{m}W^Tx_ix_i^TW$，则方差最大即最大化条件问题：

$$\begin{Bmatrix} obj:\max(\sum_{i=1}^{m}W^Tx_ix_i^TW) \\ st:W^TW=I \end{Bmatrix} \tag{3-6}$$

我们构造拉格朗日函数：

$$f(W)=W^TXX^TW+\lambda(I-W^TW) \tag{3-7}$$

我们对 W 求导：

$$\frac{\partial f}{\partial W}=2XX^TW-2\lambda W \tag{3-8}$$

令上式（3-8）等于 0 得：

$$XX^TW=\lambda W \tag{3-9}$$

显然，W 为 XX^T 特征值 λ 所对应的特征向量，所以：

$$W^TXX^TW=\lambda W^TW \tag{3-10}$$

而条件中 $W^TW=I$，所以目标函数最大方差即为最大特征值 $\lambda_1>\lambda_2>\cdots\lambda_n$，以此类推第二第三等主成分。而前 d 个特征值所对应的特征向量组成的 $W=\{w_1,w_2,\cdots w_d\}$，就是我们主成分分析的解。

3.1.3 PCA 的算法过程

（1）去平均化，即每一个特征向量减去各自的平均值。

（2）计算该数据的协方差矩阵。

（3）计算出协方差矩阵的特征向量和特征值。

（4）对特征值由大到小进行排序。

（5）保留最大的 k 个特征向量。

（6）将数据映射到由 k 个特征向量所构成的新空间中。

以上就是 PCA 算法的推导，下面我们介绍 XGBoost 算法，因为 XGBoost 算法的基函数是决策树中的 CART 树，所以在介绍 XGBoost 算法之前，我们将简单的

介绍一下决策树。

3.2 决策树

3.2.1 决策树概论

(1) 决策树的简介

决策树是一种在什么条件下会得到什么值或者是什么类别的类似规则的算法，决策树可以分为两种树，即分类树和回归树。对离散变量做决策的树是分类树，对连续数值变量做决策的树是回归树。

如果不用考虑效率和过拟合的话，所有的样本特征在决策规则下终会将其中的一个样本分到一个终止的叶子类上。事实上，样本中的一些特征在分类中起着决定性的作用。决策树在构造的过程中，就是想找到这些具有决定性作用的特征，然后根据其决定性的程度来构造一棵树，决定性程度最大的那个特征作为树的根节点，然后递归地找出在其各分支下的子数据集中，位列次大的决定性的特征，直到所有数据都有其所属的类。因此，构建决策树的过程本质上是一个根据数据特征对数据集进行分类的递归过程。所有我们要解决的问题也就是当前数据集上哪个特征在划分数据分类时有着决定性的作用。

为了找到决定性的特征以此划分出最佳的结果，我们必须对数据集中的每个特征进行评估。完成评估之后，原始的数据集就被分成了几个不同的数据子集，这些数据子集会在第一个分裂点的所有分支节点上。如果在某个分支下的所有数据是属于同一个类的，则这个分支节点称为一个叶子节点，即确定的类。划分数据子集的算法和划分原始数据集的是相同的，如果数据子集内的数据不属于同一类，则重复上诉划分过程，直到所有相同类的数据在一个叶子节点上。

决策树的生成是一个递归的过程。在决策树的基本算法中导致递归返回有三种情况：(1)当前节点里的样本都是同一个类，则不必再划分；(2)当前属性集为空，或者所有样本在所有特征属性上都具有相同的值导致无法被划分；(3)当前节点不含有样本集，无法进行划分。

决策树学习的关键是如何在每个拆分节点上选择最佳的划分属性。一般情况下，在划分的过程中，我们希望决策树的分支节点上的样本尽可能的是同一类，这样节点的纯度也会越来越高。

（2）决策树的构造

1.特征选择：特征选择指的是从数据的多个特征里面选一个特征作为当前该节点的分裂准则，如何选择特征分裂点有不同的评估准则，因此也有了不同的决策树算法。

2.决策树生成：依据选取的特征评估准则，子节点自上到下递归地生成，在数据集不可分割的时候则停止增长决策树。

3.剪枝：决策树很容易出现过拟合现象，一般情况是需要做剪枝处理的，简化树的结构，树的规模可以缓解过拟合。

（3）决策树所用到的信息论

信息熵

特征集中的数据通常以定性字符串数据的形式出现，并且成为标称数据。在实际计算中，这些数据需要量化为数字。

引入一个概念来衡量一个事物的特征取值的有无序程度：信息熵。Shannon在他的《信息论》中引用了熵，并提出了出名的信息熵。信息熵可以理解为特定信息发生的概率，或者某个事物的不确定性。

事件 u 的信息量可以表示为事件 u 发生概率 p 的单调递减函数：

$$I(u)=\log(\frac{1}{p})=-\log(p) \tag{3-11}$$

两个独立事件的不确定性是有相加的性的，即由两个独立事件产生的不确定性应等于它们各自不确定性的总和：

$$I(p_1+p_2)=I(\mathrm{p}_1)+I(p_2) \tag{3-12}$$

考虑到所有可能的信息源的平均不确定性，使用不确定的期望来统计平均值，可以称为信息熵：

$$H(U)=E[-\log(p_i)]=-\sum_{i=1}^{n}p_i\log(p_i) \tag{3-13}$$

上式 3-13 中，对数的底一般取 2。并且某个特征向量的信息熵值越大，就表明该向量的不确定性程度越大，即混淆程度越大，越复杂。

条件熵

条件熵$H(Y|\mathrm{X})$表示的是在已知随机变量 X 的条件下，随机变量 Y 的不确定性程度，即为在给定的条件 X 下，Y 的条件概率分布的熵对于 X 的期望:

$$H(Y|\mathrm{X})=\sum_{i=1}^{n} p_i H(Y|\mathrm{X}=x_i) \tag{3-14}$$

信息增益

信息增益表示的是在知道特征 X 的信息后，能够使 Y 的不确定性程度减少多少。其定义为:

$$g(D,A)=H(Y)-H(Y|X) \tag{3-15}$$

GINI 指数

GINI 指数是一种通常用于衡量收入不平等的不平等指标，也可用于衡量任何不均衡的分布，是介于 0-1 之间的数，0 表示完全相等，1 表示完全不相等。在对指标进行分类度量时，所包含的类别越混乱，GINI 指数越大（类似于熵的概念）：

$$\mathrm{Gini}(T)=1-\sum_{i=1}^{n} p_i^2 \tag{3-16}$$

其中p_i表示事件i发生的概率。

这样我们就很容易得到 GiniGain(Gini 信息度):

$$GiniGain(T)=\sum_{i=1}^{n}\frac{N_i}{N}Gini(T) \tag{3-17}$$

(4) 决策树的三种算法：

目前实现决策树分裂特征选取时的主要算法有三种，ID3 算法，C4.5 算法,CART 算法，三种算法对应三种决策树。

其中 ID3 算法采用的是信息增益作为分裂特征选择，信息增益越大，代表该特征不确定越多，以此作为分裂特征。C4.5 算法则采用信息增益率作为分裂特征的选取。而 CART 算法则采用 gini 指数的信息增益作为分裂特征的选取。

ID3 算法是一种没有剪枝的过程的且可以用于划分字符串数据集的算法，如果想要规避过拟合问题，可裁剪合并无法生成大量信息增益的相邻叶子节点（比如也可以设置信息增益的阀值）。信息增益的使用实际上具有缺点，即它

偏向于具有大量值的属性。也就是说在训练集中，属性采用的不同值的数量越多，将其用作为拆分属性的可能性就越大，有时这样做是没有意义的。并且 ID3 算法不能处理连续分布的数据特征，suoyi 在此基础上又有了 C4.5 算法，当然 CART 算法也可以处理连续分布的数据特征。

C4.5 是 ID3 的一个改进算法，其拥有 ID3 算法的优点。CC4.5 算法利用信息增益率来选择属性，克服了 ID3 算法在选择属性分裂时偏向于选择取值多的属性的缺点，并且在树的构造中对树进行剪枝，可以完成连续属性的离散化，还能对不完整数据进行处理。C4.5 算法的分类准则容易理解，准确率也高，但是，由于构造树时需要将数据集多次的顺序扫描和排序，这样效率很低。

CART 算法所使用的分裂标准是 Gini 指数，同时它是有后剪枝操作的。尽管 C4.5 算法和 ID3 算法在样本集的学习中是可以提取尽可能多的信息，但由其生成的决策树具有较大的分支和较大的规模。为了减少决策树的规模和提高决策树的生长速度，就有了基于 GINI 指数来选择分裂属性的决策树算法 CART，同时 CART 决策树只能是二叉树。

3.2.2 CART 决策树

(1) CART 两种树的区别

CART 分类树：分类树的输出是样本的类标。使用的是 Gini 指数增益最大作为分裂点的选择。比如一个相亲的例子，决定要不要去见面，分类决策树如下图 3-1 所示：

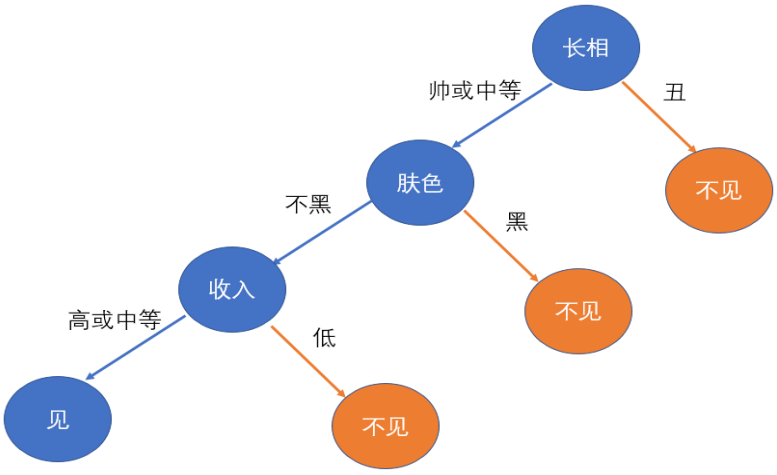

图 3-1 分类决策树样例

上图 3-1 中分类树就候选人分成了见和不见两类。

CART 回归树：回归树的输出是一个数值(比如人的年龄，工资等)。树在划分左右子树时有一个非常重要的量，即给定一个值，以这个值为标准来划分左右两个子树。选择这个给定值的原则是减少分割子树中的混淆程度，如何定义这种混乱级别是设计 CART 算法的关键部分。在回归树中，可以用方差来表示混乱程度，方差越大，越混乱。所以必须找到使切分之后的方差最小的划分值。依然以相亲为例子，但回归树得出的不是见或不见的结果，而是候选相亲人的得分值，回归树决策树如下图 3-2 所示：

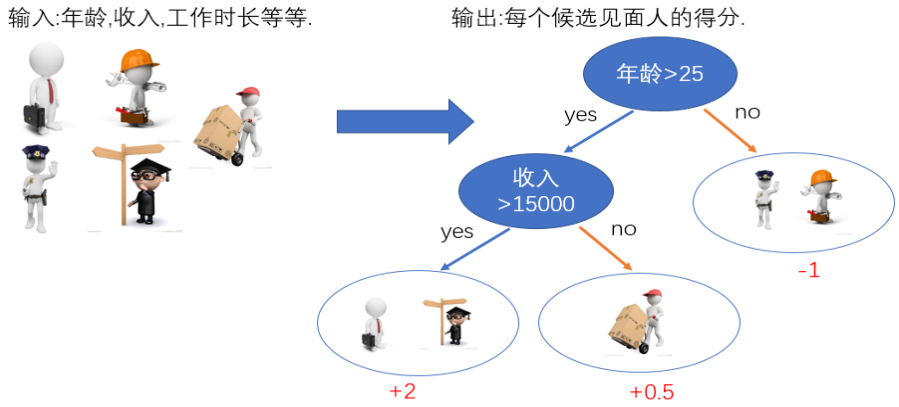

图 3-2 回归树样例

图 3-2 中标红的数值是每个候选者的得分值。

其实分类树和回归树的实质是一样的，分类树只是将样本数据隐射为类别，回归树则是将样本数据隐射预测为实值。由于 XGBoost 算法的基函数采用的是 CART 回归树,所以本文将单独详细介绍 CART 回归树的建立。

(2) CART 回归树的建立

既然 CART 树是决策树，所以一定会有以下两个核心问题：1.如何选择划分点？2.如何决定叶节点的输出值？

在回归树中，使用启发式的方法。假设有n个特征，每个特征可以取S_i个值，然后遍历所有特征，尝试某个特征的所有取值，划分空间直到得到特征 j 的取值 s，使得损失函数被最小化，这样就可以得出一个划分点.目标公式如下所示：

$$\min\left[min_{c_1}\sum_{x_i\in R_1(j,s)}LOSS(y_i,c_1)+min_{c_2}\sum_{x_i\in R_2(j,s)}LOSS(y_i,c_2)\right] \quad (3\text{-}18)$$

其中 LOSS()为损失函数，y_i为实际值，c_1，c_2分别为叶子节点R_1，R_2的输出值。

如果损失函数是平方差，则最小二乘回归树算法为：

输入：训练数据集 T；

输出：回归树 f(x)；

在训练集所在的输入空间中，每个区域被递归地划分为两个子区域，并且确定每个子区域上的输出值以构建二叉树；

(1)选择最佳切分变量 j 和切分点 s 以求得最小值：

$$\min\left[min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right] \quad (3\text{-}19)$$

遍历变量 j,对固定的切分特征 j 扫描切分点 s,选择使公式（3-19）达到最小值的 j 和 s.

(2)使用选定的特征 j 和切分值 s 并得出输出值。因为式（3-19）是关于输出值c_i的二次多项式，所以对于输出值c_i可容易求出。

(3)继续对 2 个子区域调用步骤 1，和步骤 2，直至满足终止条件。

(4)将输入的空间划分为 T 个区域$R_1,R_2,R_3,\dots R_T$,生成决策树：

$$\mathrm{f(x)}=\sum_{t=1}^{T}c_tI(x\in R_t) \quad (3\text{-}20)$$

3.3 XGBoost 算法

XGBoost 全名叫极端梯度提升算法，在最近的几年数据挖掘比赛中大放异彩，其效果显著。

XGBoost 是 2014 年 2 月被研究出来的专注于梯度提升的算法，此算法因其优良的学习效果以及高效的训练速度而获得广泛的关注。在 2015 年的 Kaggle 竞赛获胜的 29 个算法里，XGBoost 算法占了 17 个，相比之下，近年很热的深度神经网络算法则占了 11 个。在 KDDCup 2015 比赛中，使用了 XGBoost 算法的队伍排名排在前十。XGBoost 不仅运行良好，而且速度非常快，XGBoost 的性能相比其他的算法常常出现十倍以上的提高。

XGBoost 算法的三个核心：1.集成(boosting)思想。2.构建目标损失函数。3.求解损失函数。XGBoost 算法的推导我们会分为四部分介绍。XGBoost 算法的基础是集成思想，理解和清楚集成学习对 XGBoost 是很关键的，因此下面第一部

分会介绍集成思想，这也能更好的了解 XGBoost 算法模型。第二部分本文将详细推导构建损失函数，这也是常提到的策略。第三部分中，对目标损失函数进行求解，对公式进行推导，这就是算法。第四部分，也就是模型学习的过程。上面所述的也就是常提到的模型，策略，算法以及模型学习的过程。

3.3.1 Boosting 思想

研究表明单个决策树模型容易出现过度拟合，并且在实践中无法有效应用。为了解决这个问题，便有了集成学习方法的出现。

集成学习的思想主要是使用某些方法学习出多个弱分类器，然后将多个分类器组合在一起进行共同预测。主要问题就是怎么样训练出多个弱分类器和怎么样对这些弱分类器进行组合。而 XGBoost 是以 CART 树进行集成学习组合，其实就是一堆 CART 树，并且 XGBoost 的集成学习思想是使得目标损失函数沿最快的方向不断的减小至到达到终止条件。

XGBoost 使用 CART 树而不是用普通的决策树，首先由于与 CART 树的叶节点相对应的值是实际分数值而不是确定类别，因此这将有助于实现高效的优化算法。其次，XGBoost 成名的原因之一是准确，第二是速度快。之所以快速，也是选择 CART 树是有好处的。那么这样一堆树怎么做预测呢？事实上，每棵树的预测值会被加在一起作为最终的预测值。还是以相亲的那个例子，假定通过输入候选者年龄，收入，每天工作时长情况，进行预测候选者能被见面的得分值，一颗 CART 决策树模型如下图 3-3 所示：

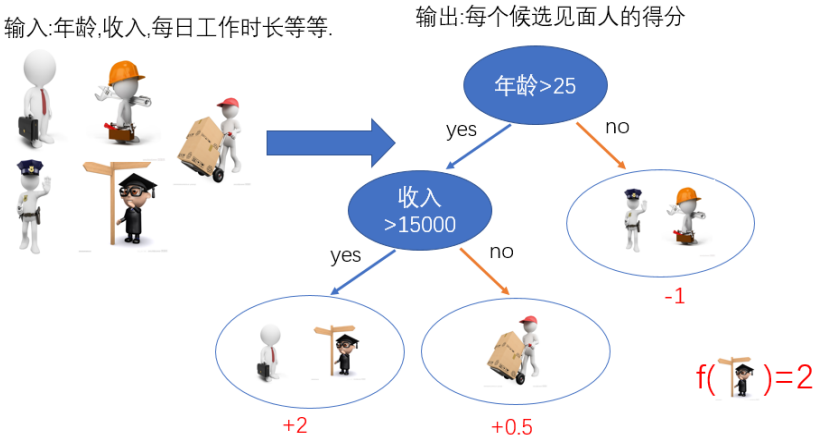

图 3-3　一颗 CART 决策树模型

集成的一堆 CART 决策树模型如下图 3-4 所示：

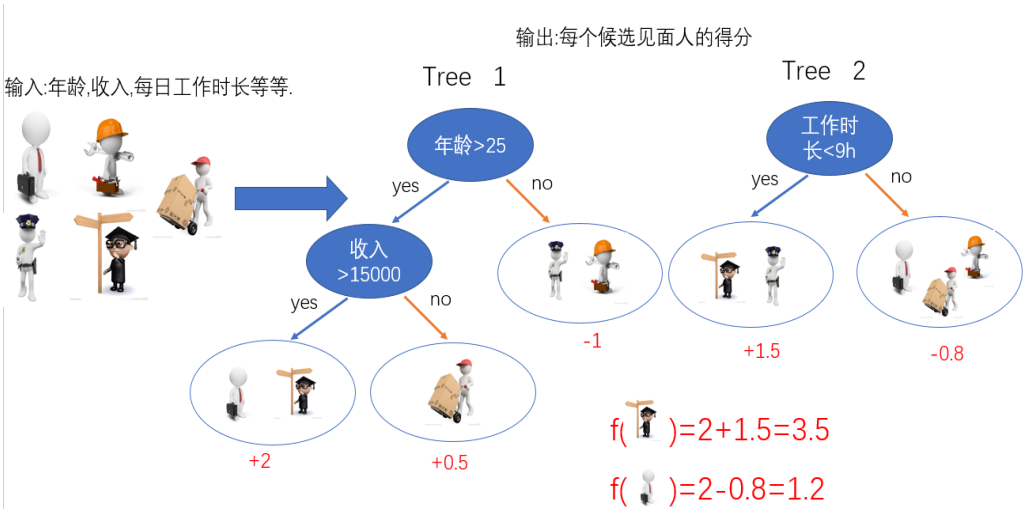

图 3-4 多颗 CART 决策树模型

对比这 2 种决策树模型由图可知，这个小博士在一颗树的得分是 2 分，而在多棵树的得分是 3.5 分。将集成思想推广，得出数学预测模型为：

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{3-21}$$

其中 K 为树的总个数，f_k 表示第 k 颗树的输出函数，$\hat{y}_i$ 表示样本 x_i 的预测结果，这个模型由 K 棵 CART 树组成。

3.3.2 目标损失函数

在介绍 XGBoost 的目标函数之前，简单介绍一下过拟合。过拟合，其实就是指学习的东西太多了，学习的太彻底，把所有的样本数据特征基本全部都学习了，这样模型就学了过多的局部特征，过多的由噪声传过来的假特征，从而使得模型的泛化能力和识别能力变得很差，所以，在用模式识别新的样本的时候就会发现没有几个是正确识别分类的。

XGBoost 损失目标函数同样包含两部分：第一部分就是目标损失函数，第二部分就是正则项，这正则项是由 K 棵树的正则项相加得来的。正则化项的目的是为了限制树的结构和生长从而达到防止模型过拟合的目的，使得模型泛化能力更强，鲁棒性更好，其实就是希望在保证树的准确性的同时也尽可能的让树的结构

简单一些，树的复杂度低一些。

XGBoost 模型的损失函数为：

$$L(\theta)=\sum_{i}^{n} l(y_i,\hat{y}_i) \tag{3-22}$$

其中 y_i 表示真实值，$\hat{y}_i$ 表示预测值，l 表示损失函数，可以是指数损失函数，平方损失函数等等。

XGBoost 模型的正则化函数（树的复杂度）为：

$$\Omega(f_k)=\gamma T_k+\frac{1}{2}\lambda\sum_{j=1}^{T_k} w_j^2 \tag{3-23}$$

其中 T_k 表示第 k 棵树的叶子的节点个数，w_j 表示第 j 个叶子节点的得分数，而 γ 和 λ 是 XGBoost 算法自己定义的，在建立 XGBoost 算法模型时，可以自己设置 γ 和 λ 的值，从公式（3-23）中可以看出，γ 值越大，此时对较多叶子节点的树的惩罚越大，表明越希望可以得到结构简单的树。同样，λ 越大是对叶子节点预测值的惩罚，也可以防止过拟合。

综合上面两式(3-22)和(3-23)，得出 XGBoost 目标函数：

$$\begin{aligned} obj(\theta)&=\sum_{i}^{n} l(y_i,\hat{y}_i)+\sum_{k=1}^{K}\Omega(f_k) \\ &=\sum_{i}^{n} l(y_i,\hat{y}_i)+\sum_{k=1}^{K}(\gamma T_k+\frac{1}{2}\lambda\sum_{j=1}^{T_k} w_j^2) \end{aligned} \tag{3-24}$$

3.3.3 求解

结合集成学习思想和目标损失函数，进行加法训练。算法的目标不再是直接优化整个目标函数，这已被证明是行不通的，而是逐步优化目标函数。首先优化第一棵树，然后优化第二棵树，直到 K 棵树被优化完全。整个加法策略过程如下所示：

初始化(这时模型里是没有树的，它的预测结果是 0)：$\hat{y}_i^{(0)}=0$　　(3-25)

向模型中加入第一棵树：$\hat{y}_i^{(1)}=f_1(x_i)=\hat{y}_i^{(0)}+f_1(x_i)$　　(3-26)

向模型中加入第二棵树：$\hat{y}_i^{(2)}=f_1(x_i)+f_2(x_i)=\hat{y}_i^{(1)}+f_2(x_i)$　　(3-27)

•••

向模型中加入第 t 棵树： $\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)$ (3-28)

其中 $f_k(x_i)$ 表示的是第 k 棵树对样本 x_i 的预测结果，$\hat{y}_i^{(t)}$ 表示综合 t 棵树的模型对样本 x_i 的预测结果。

由上面公式可得，每次将树添加到模型中时，其损失函数都会发生变化。另外在加入第 t 棵树时，前面的第 $t-1$ 棵树已经训练完毕，所以，前面所训练的 $t-1$ 棵树的正则项和训练误差就都变成了已知的常数项。

所以假设加入第 t 棵树，则分解原目标损失函数为：

$$\begin{aligned} obj(\theta) &= \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \\ &= \sum_{i}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_{k=1}^{t-1} \Omega(f_k) + \Omega(f_t) \end{aligned} \tag{3-29}$$

所得到的目标损失函数变成了式（3-29）所示，这里 XGBoost 算法的核心就是将原来的目标函数用泰勒二阶展开式来近似表示，泰勒二阶展开式为：

$$f(x+\Delta x) \approx f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2 \tag{3-30}$$

因此，目标损失函数为：

$$\begin{aligned} obj(\theta) &= \sum_{i}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_{k=1}^{t-1} \Omega(f_k) + \Omega(f_t) \\ &\approx \sum_{i}^{n} (l(y_i, \hat{y}_i^{(t-1)}) + \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} f_t(x_i) + \frac{1}{2} \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} f_t^2(x_i)) \\ &\quad + \sum_{k=1}^{t-1} \Omega(f_k) + \Omega(f_t) \end{aligned} \tag{3-31}$$

令 $g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ ， $h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ ，则由上式（3-31）得：

$$obj(\theta) \approx \sum_{i}^{n} (l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)) + \sum_{k=1}^{t-1} \Omega(f_k) + \Omega(f_t) \tag{3-32}$$

由于在训练第 t 棵树的时候，前面的 $t-1$ 棵树的预测值和树的结构（正则项）变为已知的了，则令其为常数项为 C，则根据上式（3-32）得：

$$\begin{aligned} obj(\theta) &\approx \sum_{i=1}^{n}(g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i))+\Omega(f_t)+C \\ &=\sum_{i=1}^{n}(g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i))+\gamma T_t+\frac{1}{2}\lambda\sum_{j=1}^{T_t}w_j^2+C \end{aligned} \tag{3-33}$$

同时，函数 $f_t(x)$ 是表示的是第 t 棵树上将样本 x 隐射为预测值的函数，而第 t 棵树的预测值与第 t 课树叶子节点的个数和所在的叶子节点位置有关。所以，这里令：

$$f_t(x)=w_{q_t(x)}, w\in R^{T_t}, q_t: R^d \to \{1,2,3,\cdots,T_t\} \tag{3-34}$$

其中 $q_t(x)$ 是将样本 x 隐射到某个叶子节点，比如按照某个特征分裂点为：$q_t(x): m<5; m>=5$，那么对于样本 x，按照该 m 特征的值的情况隐射到相应的叶子节点，其实就是代表树的结构。 w 则是表示节子节点的分数，也就是预测值。

因此，根据上（3-33）与（3-34）两式得：

$$\begin{aligned} obj(\theta) &\approx \sum_{i=1}^{n}(g_i w_{q_t(x_i)}+\frac{1}{2}h_i w_{q_t(x_i)}^2)+\gamma T_t+\frac{1}{2}\lambda\sum_{j=1}^{T_t}w_j^2+C \\ &=\sum_{j=1}^{T_t}[(\sum_{i\in I_j}g_i)w_j+\frac{1}{2}(\sum_{i\in I_j}h_i)w_j^2]+\gamma T_t+\frac{1}{2}\lambda\sum_{j=1}^{T_t}w_j^2+C \\ &=\sum_{j=1}^{T_t}[(\sum_{i\in I_j}g_i)w_j+\frac{1}{2}(\sum_{i\in I_j}h_i+\lambda)w_j^2]+\gamma T_t+C \end{aligned} \tag{3-35}$$

上式（3-35）中 $i\in I_j$ 表示在第 t 棵树上，样本 x_i 被分到第 j 个叶子节点上。

于是，令 $G_j=\sum_{i\in I_j}g_i$， $H_j=\sum_{i\in I_j}h_i$，则目标损失函数为：

$$obj(\theta)\approx\sum_{j=1}^{T_t}[(G_j w_j+\frac{1}{2}(H_j+\lambda)w_j^2]+\gamma T_t+C \tag{3-36}$$

对于第 t 棵 CART 树而言，在做分裂时，树的结构是确定的，因此所有的 G_j 和 H_j 都是确定的，并且一个树上的每个叶子节点的值即 w 也是相互独立的。所以上式则是一个二次项式，优化参数目标和分裂点选取是为了使得目标损失函数最小，所以由上式（3-36）得：

$$w_j^*=-\frac{G_j}{H_j+\lambda} \tag{3-37}$$

$$\min(obj(\theta)) \approx -\frac{1}{2}\sum_{j=1}^{T_t}\frac{G_j^2}{H_j+\lambda}+\gamma T_t+C \tag{3-38}$$

至此，目标损失函数便求出了其最小值。它表明树的结构是有多好，其值越小，表示树结构越好！也就是说，它是衡量第 t 棵 CART 树的结构好坏的评判标准。 有了评判树的结构好坏的准则，就可以先求最佳的树结构，树的结构确定之后，最佳的叶子节点的值也在上面公式中已经求出来了。

3.3.4 CART 树的学习过程

对于一棵树的结构几乎是无限的。因此，仍然需要采取同样的策略，就是逐步学习出最佳的树结构。所以就像将 K 棵树的模型分解成一棵一棵树来学习一样，对于在一棵树上，按树的结构一层一层的节点向下学习，而模型学习的路线则是让目标函数不断减小的方向。

这里以提到过的判断五个人是否喜欢玩游戏为例子。最简单的树结构就是一个节点的树。可以算出这棵单节点的树的好坏程度，也就是上面的目标损失函数，此时叶子节点只有 1 个，所以目标损失函数为：

$$obj_{t1} = -\frac{1}{2}\left(\frac{(g_1+g_2+g_3+g_4+g_5)^2}{h_1+h_2+h_3+h_4+h_5+\lambda}\right)+\gamma+C \tag{3-39}$$

其中 obj_{t1} 代表第 t 棵树的第一个节点的目标损失函数，依此类推。

假设现在按年龄分割这个单节点树，这里有两个问题：

（1） 根据年龄分割是否有效，即是否减少了目标损失函数的值。

（2） 如果可以分割，那么以年龄值多少来分会最大减少目标函数的值。

首先，我们将五个人按照年龄的大小进行排序，按照年龄的依次进行分裂，并以使目标函数减少最多的年龄值为节点的分裂点，假设做出如下图 3-5 所示的节点分裂：

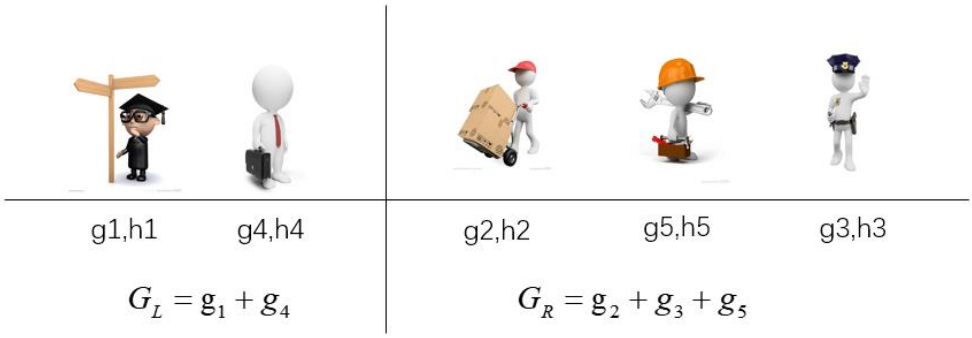

图 3-5 节点分裂

此时对于两个节点的目标损失函数为：

$$obj_{t2} = -\frac{1}{2}(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda}) + 2\gamma + C \tag{3-40}$$

则得出分裂后目标函数的减少值：

$$\begin{aligned} Gain(obj) &= obj_{t1} - obj_{t2} \\ &= \frac{1}{2}(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}) - \gamma \end{aligned} \tag{3-41}$$

$Gain(obj)$ 其实就是原单节点的目标函数和切分后的两个节点的目标函数之差，如果该值是大于零的，并且值越大，则表明切分后的目标函数值越小于单节点的目标函数值，这样就越值得切分。同时，如果上式（3-41）中的左半部分小于右侧的 γ 值，则上式就是负值，表示切分后目标函数反而变大了。而 γ 在这里其实是一个临界值，它的值如果越大，则表明对切分后目标损失函数下降程度要求越严格。之前我们也提到过，这个值我们是可以在 XGBoost 模型中设定的。

3.4　P-XGBoost 模型在股市的优点

P-XGBoost 模型的优点主要分为以下几点：

(1) P-XGBoost 模型运行速度很快

由于 XGBoost 算法的特性，每棵树的目标损失函数只与前一棵树的梯度有关，也就是上文中提到的前一棵树的 g_i 和 h_i，且每个样本 $g_i(h_i)$ 之间是没有任何关系的，这样就可以进行并行的计算，而且在一棵树内部，节点的分裂也不需要像传统的回归树一样去进行复杂计算，也只与前一棵树的 g_i 和 h_i 有关。同时，

XGBoost 的代码是基于 C++写的，在内存优化方面，大部分的内存分配在第一次加载中就完成了，之后便不再涉及动态内存分配的问题。而且在分布式中，XGBoost 的每一个节点都进行过优化，让你可以有效地在更少节点上处理更大的数据集。另一方面，XGBoost 模型训练始终是沿着损失函数的负梯度方向，使得收敛速度加快。并且如果维度很大，使用 PCA 算法也可进行降维处理。

(2) 泛化能力更强

在 P-XGBoost 算法中，PCA 算法将选股因子的有效信息综合构成新的综合指标，且指标直接无相关性，也无冗余变量，同时 XGBoost 算法中目标损失函数加入了正则项来控制模型的复杂度，并且对损失函数的要求只需要二阶可微，XGB 还支持线性的分类器，这大大的破除了很多研究领域的限制。也正因为如此，无论是模型训练本身还是研究领域，P-XGBoost 模型的泛化能力都很强。

从方差偏差的角度来讲，正则项降低了模型的方差，使得学习出来的模型在保证偏差的同时变得更加简单，防止过拟合。此外，如果样本的特征因子有缺失值，P-XGBoost 是可以自动学习并找出它的分裂方向。

综上所述，正是因为 P-XGBoost 算法模型的设计和其特性，对于数据量大反应快的股票市场，P-XGBoost 有其自身的优势，同时这也可以为高频股票交易提供一种新的思路方法和选股策略。

本章小结：本章是本文使用的量化选股模型的理论相关知识，有 PCA，基函数决策树，XGBoost 等算法，其中包括公式的详细推导，理论的阐述，算法优点的缘由，以及构成的新算法模型的优点。

第四章　P-XGBoost 模型在选股中的应用

4.1 股票因子的选取

多因子选股模型，其实就是假设上市公司的财务指标和一些行情指标的会对股票的未来收益率有影响。同时，多因子选股模型认为历史是能够重演的，该模型希望从历史数据中找到有效的影响因子，再通过这些有效影响因子找到那些有投资价值的公司。也就是说，多因子选股模型是基于一定的选择标准，并选择能够克服市场并获得稳定超额收益的投资组合。

在传统的多因子模型中，因子的选取是非常重要的一部分，需要从很多方面来考察一支股票的可投资性，并且市场上有很多的可选因子，在不同的市场环境下，它们或多或少地会对股价走势起到一定的作用。通过分析前人研究的多因子量化选股模型，不难发现，无论选择的因子个数是多少，因子的选取主要可以分为以下几类因子：估值因子，技术面因子，成长型因子，财务质量因子等。但是传统的多因子模型如果过多的选取因子会造成计算量的几何增长，并且由于数据的异常值等情况也可能使得模型产生过拟合或者是欠拟合的情况;但是如果选取的因子过少，又不能完全涵盖能影响股票的全部信息。所以传统的多因子模型的处理方法都是在选取的候选因子中检验每个因子的有效性和剔除无效因子和冗余因子，以此来达到减少计算量的同时也能保证选取的因子对股票都是有效的。而本文的P-XGBoost算法模型有先天处理这些问题的特性和优点，所以可以尽可能多的选取各类的股票因子。同时考虑到数据的可得性，本文选取的因子如下面表所示：

表 4-1 估值类因子

市盈率（PE）	股票每股市价 / 年度每股盈利
市现率（PCF）	股票每股价格 / 每股现金流量
市销率（PS）	股票总市值/主营业务收入
市净率（PB）	每股股价/每股净资产
市盈率相对盈利增长比率（PEG)	市盈率/盈利增长比率

企业价值倍数	企业价值/企业收益
账面市值比（BM）	股东权益/公司市值

表 4-2 成长型因子

总资产收益率（ROA）	(本期 ROA-上期 ROA)/(上期 ROA 的绝对值)
营业收入增长率	(今年营业收入-去年营业收入)/(去年营业收入的绝对值)
净资产收益率增长率	(今年净资产收益率-去年净资产收益率)/(去年净资产收益率的绝对值)
总资产收益率增长率	(今年总资产收益率-去年总资产收益率)/(去年总资产收益率的绝对值)
净利润增长率	(今年净利润-去年净利润)/(去年净利润的绝对值)
经营现金流增长率	(今年经营现金流净值-去年经营现金流净值)/ (去年经营现金流净值的绝对值)
营业利润增长率	(今年的营业利润-去年的营业利润)/(去年的营业利润的绝对值)
每股收益增长率	(本期每股收益-上期每股收益)/(上期每股收益的绝对值)

表 4-3 技术面因子

10 日涨跌比率(ADR)	10 日内上涨股票家数/10 日内下跌股票家数
10 日乖离率(BIAS)	(当日收盘价-10 日平均价)/(10 日平均价)
10 日随机值(RSV)	(当日收盘价-10 日内的最低价)/(10 日内的最高价-10 日内的最低价)
换手率	成交量/流通股本
VR 指标	$VR=(AVS+\frac{1}{2}CVS)\div(BVS+\frac{1}{2}CVS)$
OBV 指标	基期成交量+(涨)今日成交量-(跌)今日成交量

10 日 KDJ 指标	$K_{day10}=\frac{2}{3}K_{day9}+\frac{1}{3}RSV; D_{day10}=\frac{2}{3}D_{day9}+\frac{1}{3}K_{day9};$ $J_{day10}=3K_{day9}-2D_{day9}$
10 日 ROC 指标	(今日的收盘价-10 日前的收盘价)/10 日前的收盘价
MACD 指标	$(EMA_{快}\text{-}EMA_{慢})$ 的 10 日平均值
RVI 指标	(收盘价-开盘价)/(最高价-最低价)
PVI 指标	$PVI_n=PVI_{n-1}\times PV_n;$ 当$vol_n>vol_{n-1}$时,$PV_n=Close_n\div Close_{n-1}$;否则$PV_n=1$
10 日 CMO 指标	(上涨日的收盘价差值和-下跌日的收盘价差值的绝对值之和)/(上涨日的收盘价差值和+下跌日的收盘价差值的绝对值之和)

表 4-4 财务质量因子

净资产收益率（ROE）	税后利润/所有者权益
总资产收益率（ROA）	净利润/平均资产总额
投入资本回报率	息前税后经营利润/投入资本
资产负债率	资产负债率=总负债/总资产
总资产周转率	销售收入/总资产
营业收入	营业所得的收入
净利润	营业收入－成本－营业税
流通市值	流通股数×每股股价

传统的多因子模型需要对因子进行有效性检验和冗余因子的去除，而本文采用的 P-XGBoost 算法模型由于本身算法的优点，每一棵 CART 树不必使用全所有的特征因子，同时是使用 PCA 对原始股票因子进行处理得出新的不相关的综合指标，从而可以直接构建模型。

4.2 数据的预处理

本文选取的是 2006 年 2 月到 2016 年 5 月沪深的所有正常上市且停牌未超过一个月的部分股票,总共 821 支股票，并且这个时间段也基本囊括了中国股票市场的所有状态，时间长度也足够长。本文所用的 P-XGBoost 模型中的特征分裂的特征选取也就是上文所选的股票因子，数据的来源是 WIND 金融数据库，某金融公司数据库等。

由于原始数据并不是模型所需要的格式，且存在数据的缺失，数据存在异常值，数据之前的量纲不统一等等情况，所以需要对原始数据进行预处理。虽然本文所用的 P-XGBoost 算法模型里有针对缺失值的处理方法,但是考虑到处理数据也是需要研究的一部分,所以也会对数据的缺失值进行处理。

(1) 数据的缺失值处理

由于现实世界中的数据异常杂乱,可能由于信息暂时无法获取,信息被遗漏,有些对象的某个或某些属性是不可用的，有些信息被认为是不重要的，获取这些信息的代价太大，系统实时性能要求较高等等原因，数据会出现缺失值的情况。对缺失值的处理要具体问题具体分析，因为属性缺失有时并不意味着数据缺失，缺失本身是包含信息的,所以需要根据不同应用场景下缺失值可能包含的信息进行合理填充而不是一味的删除。本文的原始数据缺失样例如下 4-1 表所示：

表 4-5 缺失数据样例

S_INFO_WINDCODE	TRADE_DT	当日总市值(万元)	市净率	S_VAL_PCF_OCF
000756.SZ	20160220	557,007.03	3.0506	16.0826
002646.SZ	20160220	759,150	3.2317	32.804
601818.SH	20160220	16,571,078.73	0.8438	4.7757
600237.SH	20160220	423,841.54	3.0398	93.0112
002465.SZ	20160220	3,115,631.40	5.0829	
002428.SZ	20160220	1,225,906.24	7.6478	152.4201
600007.SH	20160220	1,519,989.34	2.7696	

000529.SZ	20160220	500,892.10	4.4307	33.7958
000835.SZ	20160220	451,031.49		87.0314

本文对于缺失值的股票数据的处理，如果一支股票的缺失值数据过多，我们就将其直接删除；而如果一支股票的缺失值数据较少，且缺失值的比例也小，本文使用平均值填充法。在该方法中，缺失的属性值是通过该属性在其他对象中的取值求平均值来得到。

(2) 数据的异常值处理

在数据集中存在的不合理的值称为异常值，也称为离群点。处理异常值的主要判别的方法：

1.简单统计量分析(最大值、最小值)

因为本文数据均是数值数据，所以统计分析每个属性的最大值，最小值，每个属性都有数值的规约范围，以此来判断数据是否有异常值。

2.箱型图分析

箱型图提供了可以判断异常值的一个标准，如果某个值大于或小于箱型图中所设定的上下界的数值，这样的值判断为异常值。箱型图如下图 4-1 所示：

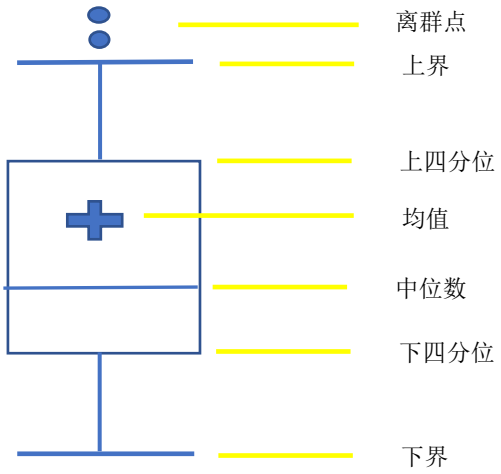

图 4-1 箱型图

在上图 4-1 中，上四分位的数值设为U，表示的是在所有的样本中只有$\frac{1}{4}$的数值大于U；同样的，下四分位的数值设为 L，表示的是在的所有样本中只有$\frac{1}{4}$

的数值小于 L。设上四分位和下四分位之差为 R，即：$R=U-L$。那么，上界为：$U+1.5R$，下界为：$L-1.5R$。

由于本文的异常值较少，所以本文采用直接删除含有异常值的数据，而不再进行修正和当成缺失值来进行插值。

(3) 数据的归一化

数据的标准化是对数据进行缩放，使其落入一个小的特定范围。它通常用于处理一些需要比较和评估的指标，以消除单位数据限制并将其转换为无量纲的数值，便于比较和衡量不同单位或量级的指标。而归一化数据处理，就是将数据全部映射到$[0,1]$区间上。

由于本文的所有数值性数据囊括了股票的多种信息，所以在数值方面差异很大，所以在做模型计算的时候需要将数据无量纲化，对数据进行归一化(标准化)。常见的归一化方法如下：

1. z-score 标准化

z-score 标准化之后的数据的均值为 0，标准差为 1，转化函数如下：

$$x^{*}=\frac{x-u}{\sigma} \tag{4-1}$$

其中u是每个特征数据的均值，σ是标准差。这也是最常见的归一化方法。

2. 离差标准化

通过线性变化将原始数据的结果落在$[0,1]$区间内，转化函数如下：

$$x^{*}=\frac{x-\min}{\max-\min} \tag{4-2}$$

其中 max 和 min 分别为每个特征数据中最大值和最小值，此类归一化方法不适用于一类数据中数据差异特别大的情况，因为容易使结果无限接近与零。

本文采用的归一化方法是离差标准化。处理完的部分数据如下表 4-6 所示：

表 4-6 归一化样例数据

S_INFO_WINDCODE_x	TRADE_DT	S_DQ_OPEN	S_DQ_HIGH	S_DQ_LOW	S_DQ_CLOSE	当日总市值（万元）	流通市值(万元)
000063.SZ	20060104	0.166719	0.172192	0.168358	0.173990	0.010599	0.035196
000063.SZ	20060105	0.173949	0.171076	0.171084	0.174053	0.010602	0.035208
000063.SZ	20060106	0.174451	0.170828	0.171211	0.171458	0.010454	0.034717
000063.SZ	20060109	0.171748	0.170209	0.168992	0.172217	0.010498	0.034861
000063.SZ	20060110	0.170617	0.168474	0.170450	0.171964	0.010483	0.034813

4.3 P-XGBoost 模型的训练

4.3.1 模型的评估

在介绍模型的评估之前，本文先做下面的一个定义。

定义 4.1：

本文定义正例为上涨股票，负例为下跌股票，如果价格不变，考虑到每次交易有手续费的问题，所以也将价格不变的股票定义为下跌股票。所以将股票转换成二分类问题(回归问题可转成分类问题)。同时也作如下定义：

真正例（TP）：真实类别为上涨股票，预测类别为上涨股票。

假正例（FP）：真实类别为下跌股票，预测类别为上涨股票。

假负例（FN）：真实类别为上涨股票，预测类别为下跌股票。

真负例（TN）：真实类别为下跌股票，预测类别为下跌股票。

同时模型的完整评估需要综合来看，不能仅仅只看分类的准确率。本文选取了四个指标来验证和评估 P-XGBoost 模型,也是在下一小节本文模型训练时需要参考的标准。下面是本文的 4 个评估指标:

1.准确率:评估整个模型对样本分类的准确度，这也是最常见的评估标准:

$$p_{准}=\frac{TP+TN}{TP+FP+FN+TN} \tag{4-3}$$

2.召回率:分类器分类正确的正样本个数占全部正样本个数的比例:

$$P_{\text{call}}=\frac{TP}{TP+FN} \tag{4-4}$$

3. F1-score: 精确率与召回率的综合平均值:

$$p_{精确}=(\frac{TP}{TP+FP}) \tag{4-5}$$

$$F1_{score}=2\times\frac{p_{精确}\times p_{call}}{p_{精确}+p_{call}} \tag{4-6}$$

4. AUC: roc 曲线下的面积,值越大，分类器结果越好，介于 0 到 1 之间。

4.3.2 模型的训练

按上文 4.1 小节和 4.2 小节所诉处理完数据之后，将数据代入模型，进行数据的训练和参数的设定。首先是将股票数据因子用主成分分析法(PCA)得出新的综合特征因子。由于本文所选的股票因子只有 35 个，因此本文使用 PCA 方法是主要是为了获的综合的新因子，PCA 后的 18 个综合因子如下图 4-2 所示：

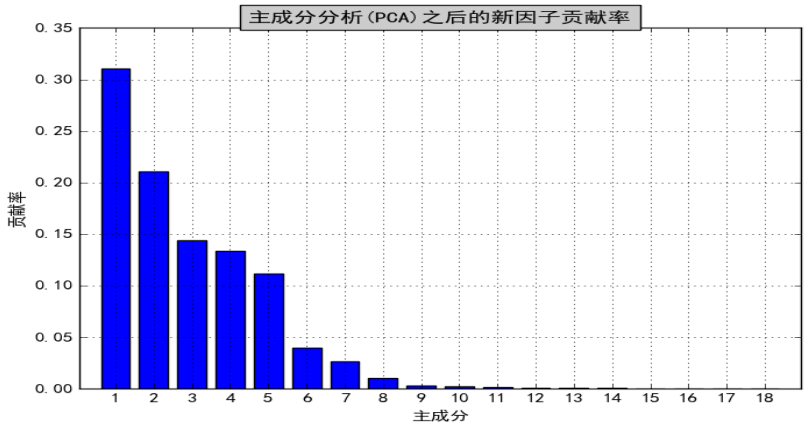

图 4-2 因子贡献率

本文选取的综合因子个数为 18，累计贡献率基本达到了 95%以上，并且在后续算法模型上也有不错的效果。同时需要注意的是，本文将样本分为了训练集和测试集，那么在做 PCA 的时候一定要保证训练集和测试集是在同一个映射特征空间。这样，之前的 35 个股票因子就用这 18 个综合因子来表示了，且因子包含的信息是逐渐递减的。

做完因子转化之后，下面是将数据带入 XGBoost 算法模型中进行训练。而在上文介绍 XGBoost 算法时，XGBoost 的一些参数则需要自己设定，下面是本文用到的 XGBoost 中比较重要的需要设定的参数：

1.eta:类似学习率。一般在[0.01,0.35]间选取。让每一棵树不用学习太多。

2. max_depth:每一棵树的最大深度。一般不宜设置过大。

3.lambda:目标损失函数中正则项叶子得分的系数，用来惩罚损失函数。

4.gamma:目标损失函数中正则项的叶子节点个数的系数，越大代表所需要的树的结构越简单。

5.colsample_bytree: 创建树的时候，从所有的列中选取的比例。

6.nthread: 这个是设置并发执行的信息，设置在几个核上并发。

7.Booster: 基函数模型，有树模型和线性模型。

8.objective:回归分类问题类型。

9.min_child_weight:叶子节点样本权重权重和的最小值。也就是第三章中 XGBoost 算法公式中的G_j,该值越小，对损失函数影响就越小。因此，为了降低复杂度，设置该值的阈值表示对该节点不再分裂。

10.colsample_bytree:建立树时对特征采样的的比例。这里需要说明的是由于我们用的 P-XGBoost 模型，之前对原有因子进行过处理了，所以这里是对特征全部采样，也就是取值为 1.

11.eval_metric:打印训练时模型的准确率或是错误率等信息。

12. tree_number:这个是表示需要的树的棵树。

参数的优化调整在以往的文献中已经有详细表述，就是一一训练设置，比对模型的训练误差，选择合适的参数，这里本文就不再一一赘述了。本文经过调参优化，使用的参数如下表 4-7 所示：

表 4-7 参数选择

Booster: Gbtree	objective: binary:logistic
max_depth: 4	lambda: 15
eta: 0.15	min_child_weight:3
gamma: 20	subsample: 0.8
colsample_bytree:1	nthread:8
eval_metric: error	tree_number:80

经过上述模型的训练和参数的选定，下面则是模型的实现和评估。本文使用的数据训练集和测试集的比例是 0.8：0.2，其中测试集有 163788 个样例，训练集有 655152 个样例，总共有约 81 万个样本。模型的训练准确率如下图 4-3 所示：

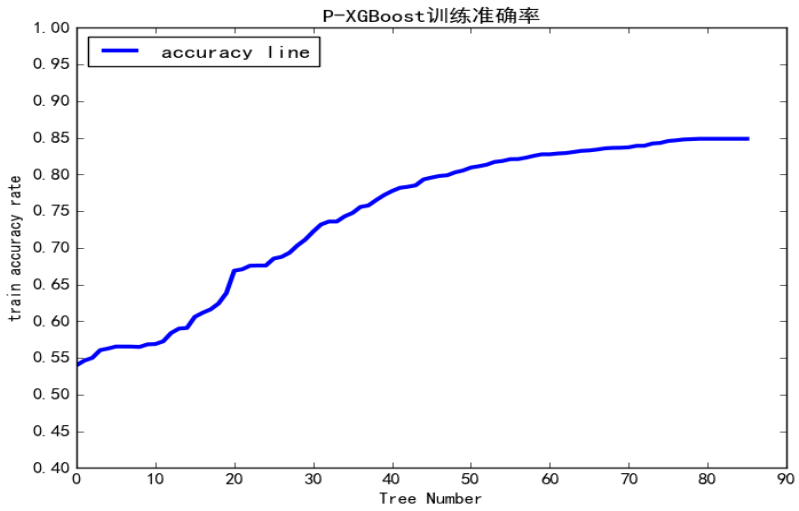

图 4-3 P-XGBoost 模型训练准确率

由上图 4-4 可知，模型在 CART 树达到 80 左右时，训练准确率变化基本可以忽略，所以我们选取的 CART 树为 80 棵。

并得出模型的评估结果如下 4-8 表所示：

表 4-8 模型评估结果

准确率(ACC)	AUC	F1-score	Recall
0.8487	0.8485	0.8729	0.8976

由上表结果可知，模型训练结果很不错，其中分类的准确率更是达到了 84.5%的准确度。同时各分类的样例个数（混淆矩阵）如下图 4-4 所示：

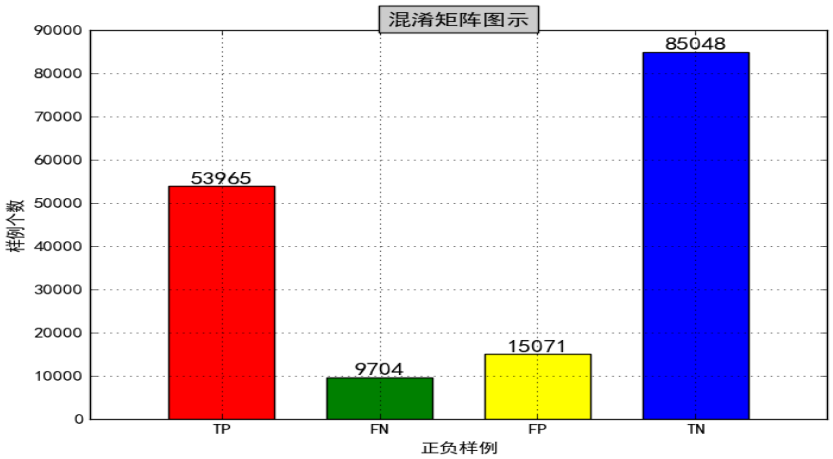

图 4-4 混淆矩阵图

由上图 4-5 可知，测试集的正例是 63669 个样本，负例是 100119 个样本，正例和负例的比约为 0.64：1。这样的比例也是合理的。综上所述，训练的模型已经有不错的效果表现了。

本章小结:本章主要是构建的新的算法模型 P-XGBoost 在股票市场的应用，其中包括对股票特征因子的选取，股票数据的处理，以及模型的训练和评估，在本章完成新模型的实现。

第五章　P-XGBoost 模型的选股表现

在上文中已经构建了 P-XGBoost 模型，并对股票数据进行了的处理和使用，模型的参数设定，以及对模型进行了评估。在这一章，本文使用部分股票数据进行回测，得出在模型下的股票的收益和风险表现的情况，并且与没改进的 XGBoost 模型作比较。下面将分两小节，第一小节介绍量化投资选股的风险和收益评估标准，第二小节是模型的收益率，风险表现与其他模型的比较。

5.1 量化选股中的风险收益评估指标

在量化投资选股中，对模型最直观的评价就是查看用该模型在股票市场所获得的收益以及所面临的风险。本文使用的股票收益评估标准是总收益率和年均复合增长率，而使用的股票风险评估标准是最大历史回撤率，以及收益和风险的综合指标夏普比率，其中夏普比率是表示每承受一单位总风险，会产生多少的超额报酬，以此可以同时对策略的收益与风险进行综合考虑。下面是本文所选 4 个指标的计算公式：

(1) 总收益率：

$$\text{股票总收益率}=\frac{\text{收益额}}{\text{原始投资额}} \tag{5-1}$$

(2) 年化收益率：

$$\text{年化收益率}=\left(\frac{\text{投资收益}}{\text{本金}}+1\right)^{\left(\frac{250}{\text{投资天数}}\right)}-1 \tag{5-2}$$

(3) 最大历史回撤率：

$$\text{最大回撤率}=max\left(1-\frac{\text{当日价值}}{\text{当日之前的最高价值}}\right) \tag{5-3}$$

(4) 夏普比率：

$$\text{夏普比率(sharp)}=\frac{\text{投资收益率-无风险收益率}}{\text{投资组合的标准差}} \tag{5-4}$$

下面本文将以这四个指标来评判 P-XGBsoot 模型，并且将 P-XGBoost 模型与改进前的 XGBoost 模型做比较，有比较才能看出改进之后的模型优势。

5.2 P-XGBoost 模型股票表现

本文使用 2013 年 1 月到 2015 年 3 月的数据进行回测，前面 2006 年到 2012 年的股票数据作为训练集构建模型，并采用单日预测。本文目前没有考虑交易手续费等问题对收益率的影响，并且只针对股票进行买进卖出，没有做股指期货的做空和做多。同时本文认为越接近交易时间的价格变化，越能反应近期股票市场的信息。但是考虑到季报的时间问题，所以本模型每个季度重新训练模型，将上一季度的财务数据加入进训练模型，将最早一季度的数据剔除，之后再对模型重新训练一次。量化择时模型如下图 5-1 所示：

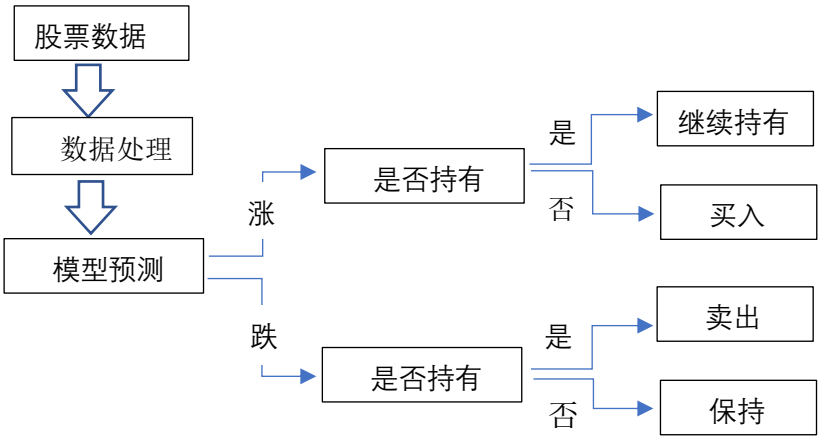

图 5-1 股票量化模型交易流程图

按上图 5-1 的交易流程，回测结果如下表 5-1 所示：

表 5-1 回测结果表现

股票总收益率	121.54%
年化收益率	38.97%
最大历史撤回	26.53%
夏普比率	1.3729

由上表可以看出，本文构造的 P-XGBoost 预测选股模型，取的了不错的收益，在股市中有较强的盈利能力，达到了本文的预期。

5.3 量化选股模型的比较

下面将是将本文的 P-XGBoost 模型和没改进的 XGBoost 模型从模型评估到股票收益进行比较。首先是模型训练的评估方面如下表 5-2 所示：

表 5-2 模型训练评估对比

	P-XGBoost	XGBoost
AUC	0.8485	0.6233
ACC	0.8487	0.7024
Recall	0.8976	0.6777
F1-score	0.8729	0.8008
Time(运行时间):	48.92s	32.66s

由上表 5-2 可以看出，原始的 XGBoost 模型相比较与结合了主成分（PCA）算法之后的 P-XGBoost 模型，准确率由 70.24%上升到了 84.87%，同时包括召回率，F1 分值以及 AUC 都有不同程度的提升,这样也充分证明了 P-XGBoost 算法模型的有效性。但是,对于同等处理好的数据,由于 P-XGBoost 要对数据做主成分，所以运行时间会比原始的 XGBoost 慢一点，同时也会对内存要求更大一些。但是对于主成分之后的数据，因为维度有所降低，使用 P-XGBoost 模型的运行速度会比 XGBoost 的快。同时本文只用了线程计算，并未在分布式上进行实现。

有了模型的优势，接下来将是模型在股票应用时的回测的结果比较，如下表 5-3 所示：

表 5-3 模型回测结果比较

	P-XGBoost	XGBoost
总收益率	121.54%	98.58%
年化收益率	38.97%	32.82%
最大历史回撤	26.53%	28.32%
夏普比率	1.3729	0.8608

累计收益率随时间变化的对比如下图 5-2 所示：

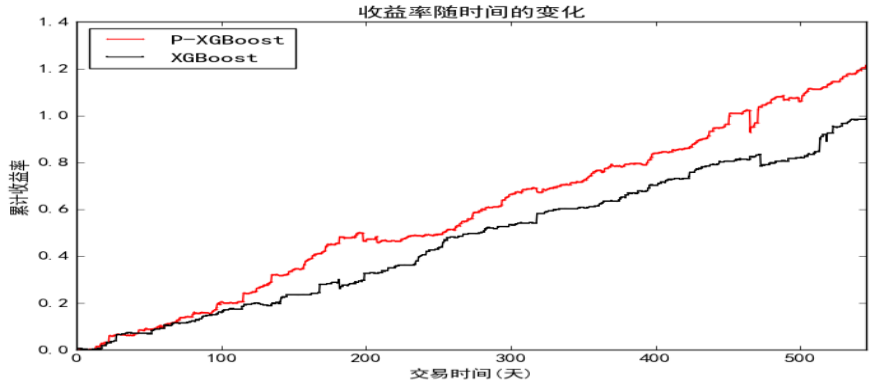

图 5-2 收益率随时间的变化

由上述回测结果可知,改进后的 P-XGBoost 模型在股市上的表现也是优于原始的 XGBoost 模型，也证明了改进后的模型的有效可行性。

本章小结：本章节使用上面构建好的模型，根据量化投资的收益风险准则，运用股票数据进行动态回测，并对改进之后的 P-XGBoost 模型和原始的 XGBoost 模型进行比较。

第六章　总结和展望

6.1 本文主要结论

本文对量化投资选股目前的模型进行了归纳总结，同时分析了现在中国股票市场量化方面的问题。受启发于 XGBoost 在数据挖掘中成功的应用，考虑到 PCA 算法的特性，将 PCA 算法与 XGBoost 算法相结合构成新的 P-XGBoost 算法模型，同时将其作为量化选股模型投入股票市场。

对本文用的模型算法 P-XGBoost 进行了详细的讲解和公式的推导，在公式的推导中，分析了该模型在速度上更快，泛化能力上更强的缘由。并且对数据的处理做了讲解，在其后的验证和训练中表明新模型 P-XGBoost 相比较于改进之前的 XGBoost 模型，在预测准确率，召回率，F1 分值，AUC 等方面均有所提升。

同时在股票数据回测结果中也表明 P-XGBoost 模型在股票市场是可以获得不错的收益，且在收益率，历史最大回撤，夏普比率上也是优于改进前的 XGBoost 模型。证明了 P-XGBoost 模型在股票投资市场的有效可行性。

但是由于中国股票市场具有一定的混沌效应，想要依靠同样的因子和模型进行长期的获利是不现实的，还需要结合目前中国的市场政策以及金融股票的相关知识和嗅觉，实际问题，实际分析建模。

6.2 展望

（1）本文所使用的数据只用了股票和财务数据，我相信这也没有将股票的影响因子全部囊括。在后续的研究中，会加入期货，债券等方面的特征因子，有更加全面的因子相信还会提高模型的预测能力。

（2）本文对模型的训练是对参数一个或一对的单独调试，然后观察训练误差的变化，通过训练误差来选定参数。后续的研究中可以考虑使用遗传算法来进行最佳参数的寻优。

（3）由于本文 P-XGBoost 模型的特性，可以分布式运算，并且算法的设计在特征分裂选取时也非常地迅速。后续可以使用分钟级或者是更高频率的股票数据，进行高频预测，这也为高频交易提供一种新的思路。

相信随着进一步的深入研究，将 P-XGBoost 模型应用到中国股票市场上，量化投资选股将得到更好的效果。

致 谢

随着时间的流逝，三年的硕士生生活马上就要结束了。在这三年里，我的老师，我有幸结识的很多同学，好朋友，他们都给予了我很多无私的帮助和关怀，让我的三年研究生生活更加的丰富难忘。在此我想向他们表达我诚挚的谢意！

首先我要感谢我的导师黄光东副教授，非常荣幸能成为您的学生，也是您让我在三年里完成了我的蜕变，你的言传身教都时刻的影响着我。黄老师在学术在给了我很多的帮助，细心的指导我，为我解开了学习方面的很多困惑，同时您每周进行的讨论班也让我在讲解和自学方面有了很大的提高。黄老师在生活方面开朗，大度，其谦和的为人处事的方式也让我受益匪浅。同时，黄老师还鼓励我们多学习，多思考，多实践。支持我去实习，将学习里的知识与社会相结合，按着自己的发展道路不断前进。黄老师的关怀，信任，严谨的学术风范以及诲人不倦的精神都让我如沐春风，终身难忘。在此，我向您致以我最真诚的敬意与谢意！

同时，也要感谢我身边的同学和朋友，刘凯华，联克强，李油，张海，王栋，王子耀。每当我遇到困难时，你们总是无私的帮助我，生活方面的支持，学术瓶颈的讨论都使我在这几年里感动颇多。我也非常荣幸能认识你们这一群活泼可爱的同学，从考试，实习到找工作，期间的紧张，迷茫和奔波也让我们更加珍惜彼此的友谊。还有同门的师兄弟们，与你们的相处让我这个独自在异地求学的人感到温暖和开心，你们在学术上的讨论也给了我很多的新思路。在此，我对你们表示我深深的谢意。

最后，我要感谢我的父母，远在异地的父母无时无刻的牵挂和关心都是我努力学习奋斗的源泉，父母无私的爱，支持我的每一个决定，无论我在何方，都给予我最真诚的关心。对于父母，我唯有走好今后的每一步，以报父母的恩情。我亦不是一个善于表达情感之人，我愿以实际行动回报父母。

参考文献

[1] 丁鹏. 量化投资-策略与技术. 北京：电子工业出版社,2011.

[2] 朱建平. 经济预测与决策. 厦门：厦门大学出版社，2007.

[3] 商晔. 隐马尔可夫模型参数训练的改进及在股市预测中的应用:[硕士学位论文]. 上海：上海交通大学,2011.

[4] 吴漫君. 基于隐马尔科夫模型的股价走势预测:[硕士学位论文]. 广州：华南理工大学,2011.

[5] A Kazem,E Sharifi,FK Hussain,et al. Support vector regression with chaos-based firefly algorithm for stock market price forecasting.《Applied Soft Computing Journal》,2013,13(2):947-958.

[6] L Kumar,A Pandey,S Srivastava,et al. A Hybrid Machine Learning System for Stock Market Forecasting. 《Proceedings of World Academy of Science Engineering & Technolog》,2008,20:315—318.

[7] 张杨,宋恒. 基于聚类技术的股市基本趋势规律挖掘. 数理统计与管理,2006年04期.

[8] 阎纲. 支持向量机在股市预测中的应用. 科学技术与工程,2008,8(2):507-510.

[9] KEISUKE Y, SUMIO W, et al. Algebraic geometry and stochastic complexity of hidden Markov models. Neurocomputing,2005,69:62－84.

[10] EDMONDO T, MARCO G, et al. Robust Combination of Neural Networks and Hidden Markov Models for Speech Recognition. IEEE Transaction On Neural Networks,2003,14:6.

[11] 杨析，陈展. 中国股市因子资产定价模型实证研究. 数量经济技术经济研究，2003,(12):137-141

[12] 吴世农，许年行. 资产的理性定价模型和非理性定价模型的比较研究-基于中国股市的实证分析. 经济研究，2004,(6):105-116 .

[13] 胡淑兰，魏捷，黄晟. 基于HMM的中国股市状态转换及预测. 统计与决策,2011,第22期（总346期）.

[14] Hassan R and Nath B. StockMarket Forecasting Using Hidden Markov Model: A New Approach. International Conference on Intelligent Systems Design &

Applications, 2005:192-196.

[15] 李体委，Fama-French因子模型的改进和对中国股市收益率的检验:[硕士学位论文]. 济南：山东大学，2011.

[16] MR Hassan,B Nath,M Kirley. A fusion model of HMM, ANN and GA for stock market forecasting.《Expert Systems with Applications》,2007,33 (1):171-180.

[17] 李姝锦,胡晓旭,王聪. 浅析基于大数据的多因子量化选股策略. 经济研究导刊,2016,第17期.

[18] 汪敏. 建立在中国股市的数量化投资模型实证分析:[硕士学位论文]. 上海：复旦大学，2013.

[19] David R. Aronson. Evidence-Based Technical Analysis：Applying the scientific Method and Statistical Inference to Trading Signals. New York：John Wiley & Sons，2007.

[20] 黄恒秋. 基于高频数据的支持向量机量化择时预测模型. 科技经济刊，2016,第13期.

[21] R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working set selection using second order information for training Support Vector Machines. Machine Learning Research,2005(6):1889-1918.

[22] 方浩文. 量化投资发展趋势及其对中国的启示. 宏观管理,2012,第5期.

[23] KJ Kim. Financial time series forecasting using support vector machines.《Neurocomputing》,2003,55(1):307-319.

[24] 詹财鑫. 基于SVM_AdaBoost模型的股票涨跌实证研究:[硕士学位论文]. 广州：华南理工大学，2013.

[25] Eugene F Fama，and Kenneth R French. The Value Premium and the CAPM. Working Paper,2005.

[26] Martin Lettau and Jessica Wachter. Why is Long-Horizon Less Risky? A Duration-Based Explanation of the Value Premium. NBER Working Paper,2015,11144.

[27] S Ramnath，S Rock and P.Shane. The Financial Analyst Forecasting Literature：A Taxonomy with Suggestions for Further Research. International Journal of Forecasting，2008，24:34—75.

[28] 朱宝宪，何治国. 13值和账面是市值比与股票收益关系的实证研究. 金融研究，

200204，71～79.

[29] Wes McKinney. 利用Python进行数据分析. 北京：机械工业出版社,2013.

[30] HYRY Studio. 用 Python 做科学计算. 北京：清华大学出版社，2012.

[31] RP Sheridan，WM Wang，A Liaw，et al. Extreme Gradient Boosting as a Method for Quantitative Structure-Activity Relationships. 《Journal of Chemical Information & Modeling》，2016，56(12):2353.

[32] C Bort Escabias. Tree Boosting Data Competitions with XGBoost. Universitat de Barcelona,2017.

[33] X Ren，H Guo,S Li,S Wang，J Li. A Novel Image Classification Method with CNN-XGBoost Model. International Workshop on Digital Watermarking, 2017:378-390.

[34] A Daffertshofer，CJ Lamoth，OG Meijer，PJ Beek. PCA in studying coordination and variability: a tutorial. 《Clinical Biomechanics》, 2004,19(4):415-428.

[35] 黄艳莹,基于 EMD-XGBoost-AR 模型的网络舆情预测研究:[硕士学位论文]. 广州：广东工业大学,2017.

[36] D Nielsen. Tree Boosting With XGBoost-Why Does XGBoost Win "Every" Machine Learning Competition? 2016.

[37] 范淼,李超. Python 机器学习及实践:从零开始通往 Kaggle 竞赛之路. 北京：清华大学出版社,2016.

[38] T Chen,C Guestrin. XGBoost: A Scalable Tree Boosting System. Acm Sigkdd International Conference on Knowledge Discovery&Data Mining, 2016:785-794.

[39] 李想. 基于 XGBoost 算法的多因子量化选股方案策划:[硕士学位论文]. 上海：上海师范大学,2017.

[40] 凌筱玥. 基于 XGBoost 算法的上证指数预测方案设计研究:[硕士学位论文]. 上海：上海师范大学, 2017.

[41] 张昊 ,纪宏超,张红宇. XGBoost 算法在电子商务商品推荐中的应用.《物联网技术》,2017,7(2):102-104.

[42] 李叶紫,王振友,周怡璐,韩晓卓. 基于贝叶斯最优化的 Xgboost 算法的改进及应用.《广东工业大学学报》,2018(1):23-28.

[43] 苏冰. 基于 PCA-SVM 模型的量化择时研究:[硕士学位论文]. 天津：天津财经大

学,2015.

[44] 史卫峰. PCA 和随机森林在 BARRA 量化对冲模型中的应用研究:[硕士学位论文]. 西安：西安科技大学, 2017.

[45] 胡帅,顾艳,曲巍巍. 基于 PCA-LVQ 神经网络的教学质量评价模型研究.《河南科学》,2015(7):1247-1252.

[46]TP Minka. Automatic choice of dimensionality for PCA. International Conference on Neural Information Processing Systems, 2000 :577-583.

[47] Yi Xinyang,Park Dohyung,Chen Yudong,Caramanis Constantine. Fast Algorithms for Robust PCA via Gradient Descent. eprint arXiv,2016:1605.07784.

[48] N Fitriah,SK Wijaya,MI Fanany,C Badri,M Rezal. EEG channels reduction using PCA to increase XGBoost's accuracy for stroke detection. American Institute of Physics Conference Series, 2017,1862(1):2489-2492.

[49] A Gómez-Ríos,J Luengo,F Herrera. A Study on the Noise Label Influence in Boosting Algorithms: AdaBoost, GBM and XGBoost. Springer International Publishing AG,2017.

[50] 黄宏运,吴礼斌,李诗争. GA 优化的 SVM 在量化择时中的应用.《南京师范大学学报(工程技术版)》,2017,17(1):72-79.

[51] 唐华松,姚耀文. 数据挖掘中决策树算法的探讨. 《计算机应用研究》,2001,18(8):18-19.

[52] ZHANG Hui. CART Decision Tree Classifier Based on Multi-feature of MODIS Data.《Geospatial Information》,2013.

[53] Y Zhang,D Wang,DO Eeis. A Speech Model Cluster Method Based on GBDT Algorithm.《Informatization Research》,2013.

附录 A

作者简介

李洋，男（1991.2-），四川人。2015 年 7 月，毕业于中国地质大学(北京)数学与应用数学专业，获得理学学士学位。2015 年 9 月至 2018 年 6 月在中国地质大学（北京）攻读数学专业硕士学位，研究的方向是金融数学方向。

附录 B

数据处理的部分程序:

```
import pandas as pd
import numpy as np
a=pd.read_csv("F:/研究僧/liyang/price.csv",sep="\t")
b=a.sort(["S_INFO_WINDCODE","TRADE_DT"]).reset_index(drop="index")
b["状态"],b["get"]=get_label(b)
pd_rep2=pd.read_csv("F:/研究僧/liyang/report2.txt")
pd_rep1=pd.read_csv("F:/研究僧/liyang/report1.csv")
jo=pd.merge(pd_rep1,pd_rep2,on=['WIND_CODE',"REPORT_PERIOD"])
pd_pepb=pd.read_csv("F:/研究僧/liyang/PEB.csv")
pd_size=pd.read_csv("F:/研究僧/liyang/size.csv")
jo2=pd.merge(pd_pepb,pd_size,on=["\ufeffS_INFO_WINDCODE","TRADE_DT"])
tft=pd.merge(b,jo2,left_on=["TRADE_DT","S_INFO_WINDCODE"],right_on=["TRADE_DT",
"\ufeffS_INFO_WINDCODE"])
tft["REPORT_PERIOD"]=get_dat(tft)
join_all=pd.merge(tft,jo,left_on=['S_INFO_WINDCODE',"REPORT_PERIOD"],right_on=[
"WIND_CODE","REPORT_PERIOD"])
ffff=join_all.iloc[:,[0,1,2,3,4,5,9,10,11,12,14,15,17,18,19,23,24,25,26,30,31,3
2,33,34,35,7,6]]
dfdf=ffff.dropna(how="all",axis=1)
dfdf.iloc[:,2:-1]=dfdf.iloc[:,2:-1].astype(str).apply(lambda
x:x.str.replace(",","").astype(float))
dfdf.iloc[:,2:-1]=dfdf.iloc[:,2:-1].apply(lambda x :x.fillna(x.mean()))
dfdf=dfdf.drop_duplicates(["S_INFO_WINDCODE_x",'TRADE_DT'])
dfdf.iloc[:,2:-2]=(dfdf.iloc[:,2:-2]- dfdf.iloc[:,2:-2].min())/(dfdf.iloc[:,2:-
2].max()-dfdf.iloc[:,2:-2].min())
```

P-XGBoost 模型的部分程序:

```
import pandas as pd
import xgboost as xgb
import time
from sklearn.cross_validation import train_test_split
from sklearn import metrics
pd_gp=pd.read_csv("F:/研究僧/
liyang/gp3.csv").drop_duplicates(["S_INFO_WINDCODE_x","TRADE_DT"])
arr_cacul=pd_gp.iloc[:,2:-2].values
arr_label=pd_gp["状态"].values
arr_price=pd_gp["get"].values
arr_label=arr_label.astype(int)
arr_gp=pd_gp["S_INFO_WINDCODE_x"].values
train_x,test_x,train_y,test_y=train_test_split(arr_cacul,arr_label,
random_state=0,train_size=0.8)
train_x=pca.transform(train_x)
test_x=pca.transform(test_x)
dtrain=xgb.DMatrix(train_x,label=train_y)
dtest=xgb.DMatrix(test_x)
params={'booster':'gbtree',
    'objective': 'binary:logistic',
    'eval_metric': 'error',
    'max_depth':4,
    'lambda':15,
    'subsample':0.8,
    'gamma':20,
    'colsample_bytree':1,
    'min_child_weight':3,
```

```
    'eta': 0.15,
    'seed':10,
    'nthread':8,
     'silent':1}
watch_list = [(dtrain, 'train')]
bst=xgb.train(params,dtrain,num_boost_round=80,evals=watch_list)
```