基于机器学习算法预测股票走势

闵凉宇 2017310167 信息管理与工程学院 上海财经大学

0. 研究背景及研究意义

金融时间序列数据由于其具有非平稳性，复杂性并且含有大量的噪声，导致传统的统计学方法(比如 ARMA 模型等)对其的预测效果很一般。张丽霞[1]等人在《基于支持向量机方法的金融时间序列研究》一文当中提出使用 SVM 预测金融时间序列数据的方法，并应用在我国上证 180 的指数预测当中，并且取得了较好的效果，本文复现了她的部分工作，并且和其他机器学习模型进行了对比。

从近几年的研究工作来看，机器学习算法在建模预测等工作上对传统统计学方法有着取而代之的趋势。

在大数据时代，机器学习模型对于数据处理有着很好的效果，以利率市场化背景下对利率进行定价的工作来看，传统的方法诸如 VAR，专家模型(5C)，Altman 的 Z 计分模型，ZETA 的信用风险模型，死亡率模型，信用评级等方法，这些模型简单明了，但是计量方法粗糙，对于细节的把握不是很理想。现代信用风险度量方法主要有 Credit Metrics 模型，KMV 模型，Credit Risk+模型等，这些模型在问题刻画上仍然有些一些缺陷，最重要的一点是模型在输入数据存在大量缺失值的时候，模型的效果会偏离实际。孙存一[2]，龚六堂[3]在《大数据思维下的利率定价研究》一文中使用机器学习算法对利率进行分析定价，利用机器学习所能采集到的全样本、全变量构建模型，最大限度地逼近真实利率价格水平，得到的结果对复杂的经济环境有一定的适应性。

同样，机器学习在智能投顾方(Robot-Advisor)面有着很好的应用，在 2016 年 12 月的《证券市场导报》中提到，智能投顾服务在广义上可以分为三个层次：第一个层次是通过大量数据分析提供一般意义上的投资建议，不能满足客户的定制化需求；第二次层次是根据服务对象的特征和偏好，给出定制化的投资建议，但是没有交易的功能；第三个层次是在第二个层次的基础上，为服务对象提供交易服务，包括完全自动交易，人工投资顾问协助交易和自动执行交易，目前来看，一般意义上的智能投顾是指第二个层次到第三个层次服务的智能。

近年来，国际上很多公司都重视将人工智能和金融投资服务相结合。2013 年，新加坡的 Dragon Wealth 公司成立，利用手机移动应用程序提供在线资产管理服务，该公司与大数据及整体解决方案(Crowd Solution)供应商建立战略合作关系，进行信息收集和分析。Dragon Wealth 公司最大的特点是向客户提供与客户投资信息和资产规模类似的同类组(peer group)的投资分析报告。

2015 年 4 月 30 日，韩国三星证券等 10 个券商和 KB 银行等 6 家银行联合推出了 Samsung POP RoboAdvisor、Quarterback、QVRobo Account、i-ROBO α 等智能投顾系统，韩国政府也已经通过“关于活跃智能投顾的方案”，大力支持智能投顾的发展。

相比较发达国家，我国智能投顾发展比较晚，2015 年相继有一些智能投顾理财产品推出。目前，京东金融，蚂蚁聚宝等互联网公司纷纷上线财富管理 APP，推动我国智能投顾事业的

[1] 辽宁工业大学 信息科学与工程学院
[2] 北京大学 光华管理学院
[3] 北京大学 光华管理学院

发展。除了法律制度需要完善之外，智能投顾在我国发展的难点还有如何应对小概率事件，众所周知，量化投资在应对大概率事件时表现很稳定，但是当“黑天鹅”事件发生时，量化策略往往进入无效的状态，比如曾经光大证券的“乌龙指”事件，“市场熔断”事件等。

正是因为当前技术在金融市场上的应用尚存在大量需要改进的空间，这也促使我们再金融科技领域继续探索。

1.工程思路

1.1 获取数据

通过 yahoo 或者 sina 的 API 获取某只股票的选定时间段的数据，主要方法是调用目前的开源包 Tushare，股票选择为中国银行(601988)，数据选取的时间段为 2014-1-1(start_time)至 2017-10-25(end_time)，选择银行股的原因是在上证 50 成分中银行股为主力股票，并且一直以稳定的特性而著称，众所周知，通过平稳时间序列数据学习到的模型更加可靠，预测的结果也相对可信。

1.2 处理数据

主要工作包括数据 index 的处理，标准化处理，训练集和测试集的分离等。数据索引处理主要将原来纯 int 的索引去除，变成以日期顺序的升序 DataFrame。标准化处理就是将 DataFrame 中的列属性数值按照公式 $\frac{X-mean}{std}$ 进行处理，这样对于每个属性而言，所有的数据都集中在 0 附近，并且方差为 1.0，标准化主要应用 sklearn.preprocessing.scale()函数进行实现。

1.3 模型搭建

根据目前课程进度，我们已经学习了线性回归模型，Logisitic Regression 模型，SVM 等预测模型，这个案例中使用 Linear Model 和 SVM 进行训练，得到准确度。

1.4 绘制图像

完成模型的训练之后，使用模型进行预测，根据预测出来的数值进行绘图，评估预测的结果，给预测数据和实际数据一个比较直观的展示、

1.5 效果评估

利用模型预测得到的数值结果和真实数据进行对比，计算模型预测的正确率，评估模型的效果，对模型进行改进处理(主要工作可能就是调整参数)，以求得更好的模型。

1.6 拓展研究

根据实际经验和当前研究进展，对当前的研究进行拓展，力求提高研究成果的实际应用价值。

2.环境搭建

本案例使用的开发 IDE 是 PyCharm 2016.2，主要使用的模块如图 2-1 所示。

```
"""
@author: Liangyu Min
"""

import pandas as pd
import numpy as np

import datetime
import time
# import pandas_datareader as web 该接口出现bad handshake Error
import tushare as ts
import math
import matplotlib.pyplot as plt
from matplotlib import style
from sklearn.model_selection import cross_val_score
from sklearn import preprocessing, cross_validation, svm
from sklearn.linear_model import LinearRegression
```

图 2-1：Demo 中主要使用的模块

其中 pandas，numpy，math 是数据处理模块，datetime 和 time 是时间模块，tushare 是数据获取模块，matplotlib 是绘图模块，sklearn 是机器学习调包模块。

工作平台是 Win 64bit，处理器是 Intel i7-6700HQ，使用 CPU 处理模型，收敛速度在容忍范围之内。

3.模型原理及代码实现

考虑使用两种模型进行预测，第一种是线性回归模型(Linear model)，第二种是支持向量机(SVM)。

3.1 多元线性回归

数据集采用的属性是 close(当日收盘价)，flucate(当日震幅)，price_change(当日最高价 High-当日最低价 Low)，volume(当日交易量)，Label 是滞后的价格，滞后日期为采集数据量的 1%，处理代码如下：

$$forecast\_out = \mathrm{int}(math.ceil(0.01 * len(boc\_df)))$$
$$boc\_df['label'] = boc\_df[forecast\_col].shift(-forecast\_out)$$

在构建多元线性回归模型的过程中，数据需要被标准化处理，调用 sklearn 包中的 preprocessign.sacle(X)函数直接按属性处理数据：

$$X\_scale = preprocessing.scale(X)$$

数据集切分为两块 X_scale 和 X_delay，前者作为实验模块，包括训练集和测验集，后者为预测模块，切分代码如下：

$$X\_delay = X\_scale[-forecast\_out:]$$
$$X\_scale = X\_scale[:-forecast\_out]$$

在数据清洗和切分完成之后，就可以调用 sklearn 包中的相关函数生成多线性回归模型进行预测，即满足如下公式：

$$f(x_i) = \omega^T x_i + b, \text{ 使得 } f(x_i) \simeq y_i$$

3.2 支持向量机

使用 SVM 对数据集划分超平面分类，且要求超平面有最大间隔，即满足约束：

$$\max_{\omega,\mathrm{b}} \frac{2}{\|\omega\|}$$

$$s.t$$

$$y_i(\omega^T\omega_i + b) \geq 1, \mathrm{i} = 1,2,...,m$$

使用不同的核函数对回归得到的精确度进行测量，在本案例中，预测精度主要受到核函数的影响，在实验中选取的核函数如表 3-1 所示：

核函数名称	核函数表达式	其他说明
线性核(Linear)	$k(x_i, x_j) = x_i^T x_j^T$	
多项式核(Poly)	$k(x_i, x_j) = (x_i^T x_j^T)^d$	参数 d 为多项式的次数
高斯核(RBF)	$k(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$	σ >0 为高斯核的带宽(width)
Sigmoid 核	$k(x_i, x_j) = \tanh(\beta x_i^T x_j + \theta)$	tanh 为双曲正切函数, β >0，θ <0

表 3-1 SVM 主要使用的核函数

4.测试结果反馈及拓展

4.1 不同参数下模型的精确度

将 test_size 从 15%-10%进行测试，得到模型不同的精确度如下表 4-1(线性模型)和 4-2(SVM)所示。

Test_size	Accuracy
15%	93.1%
14%	90.1%
13%	90.9%
12%	88.0%
11%	92.97%
10%	92.4%

表 4-1 不同参数下 Linear Model 的精确度

Test_size / Kernal	15%	14%	13%	12%	11%	10%
linear	91.8%	90.1%	91.1%	88.1%	93.0%	92.6%
poly	74.9%	60.7%	73.8%	45.1%	72.2%	76.8%
RBF	93.2%	91.4%	91.2%	88.4%	93.8%	93.6%
sigmoid[4]	-667	-358	-550	-674	-1170	-844

表 4-2 不同参数下 SVM 各核函数的精确度[5]

[4] sigmoid 核函数在本案例中可能没有完成收敛，以至于预测精度完全失灵

[5] 表格中数据每次都会在很小的范围内波动

4.2 模型预测结果及评价

根据训练集得到精确度，同时考虑过拟合的影响，本例中选取 test_size=10%作为线性模型的模型参数，得到预测曲线如图 4-1 所示，图中红色曲线的部分表示历史数据，蓝色曲线部分表示预测数据，具体预测数值和实际数据如表 4-3 所示。

从预测数据和真实数据的对比结果来看，预测结果与实际结果的数据相差在 0.1 元之内，但是变化趋势的判断准确率强差人意，因此模型尚有需要改进的空间。

使用 SVM 模型进行预测，采用核函数 RBF 得到的预测结果与真实结果如表 4-4 所示，总体来看 SVM 的数值预测精确度比线性模型有了一定的提高，但是预测方向错误率较大，因此可以认为 SVM 模型对过去数据有较强的依赖。

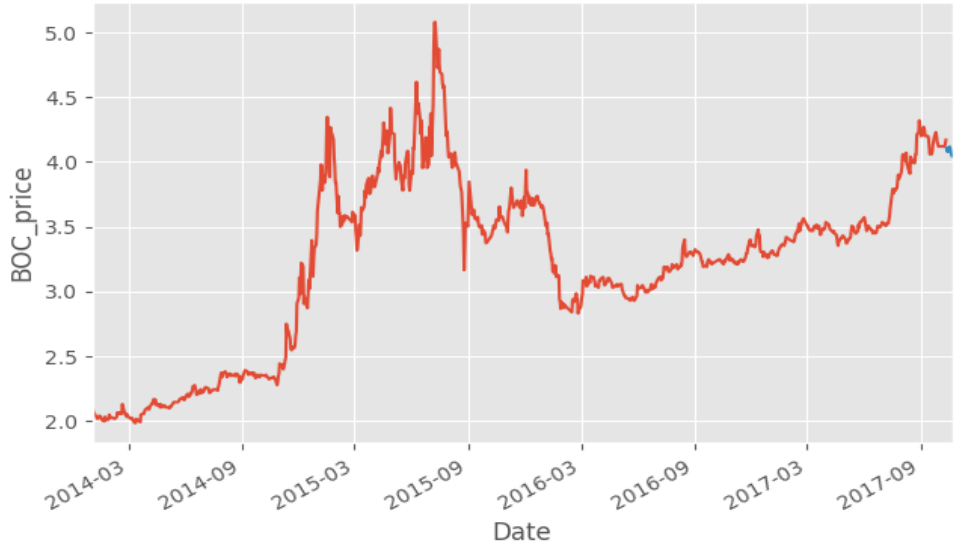

图 4-1 利用线性得到的预测曲线

日期	预测数据	实际数据	数值偏差	方向是否正确
2017-10-12	4.106663	4.18	-0.073337	NULL
2017-10-13	4.081760	4.15	-0.06824	True
2017-10-16	4.088994	4.14	-0.051006	False
2017-10-17	4.081666	4.15	-0.068334	False
2017-10-18	4.108772	4.17	-0.061228	True
2017-10-19	4.118341	4.16	-0.041659	False
2017-10-20	4.105686	4.11	-0.004314	True
2017-10-23	4.077018	4.16	-0.082982	False
2017-10-24	4.055915	4.11	-0.054085	True
2017-10-25	4.045628	4.12	-0.074372	False

表 4-3 线性模型预测结果与实际数据对比

日期	预测数据	实际数据	数值偏差	方向是否正确
2017-10-12	4.160924	4.18	-0.019076	NULL
2017-10-13	4.141712	4.15	-0.008288	True

2017-10-16	4.157471	4.14	0.017471	False
2017-10-17	4.144211	4.15	-0.005789	False
2017-10-18	4.137737	4.17	-0.032263	False
2017-10-19	4.151410	4.16	-0.00859	False
2017-10-20	4.161408	4.11	0.051408	False
2017-10-23	4.109121	4.16	-0.050879	False
2017-10-24	4.122747	4.11	0.012747	False
2017-10-25	4.064636	4.12	-0.055364	False

表 4-4 SVM(RBF)预测结果与实际数据对比

4.3 拓展研究一：蜡烛图

根据本人在资管投行的工作经验，客户往往对数据的可视化有更加的偏好，因此在研究工作完成后，如何构建一个客户易于接受的可视化界面往往成为吸引客户的关键。

在本案例中，将股票日 K 数据绘制成蜡烛图，绘图主要使用 matplotlib 包中的函数，关键是数据类型的转换和图像参数的设置，关键步骤如下所示。

import 包中函数：

```
# 日期参数处理函数
from matplotlib.dates import DateFormatter, WeekdayLocator, DayLocator, MONDAY, date2num, MonthLocator
# 蜡烛图绘制函数
from matpoltlib.finance import candlestick_ohlc
```

根据函数 candlestick_ohlc 的接口

```
def candlestick_ohlc(ax, quotes, width=0.2, colorup='k', colordown='r', alpha=1.0)
```

其中参数 quotes 需要接受一组序列(time, open, high, low, close,…)，在之前的处理中数据一直以 DataFrame 的格式存储，因此需要手动将数据转换成为表格，转换处理过程如下：

```
# 建立空列表
list_data=[]
# 对 DataFrame 进行按行遍历
for i in range(len(boc_df)):
    # 组合成为列表
    list_cont = [boc_df.date[i], boc_df.open[i], boc_df.high[i], boc_df.low[i], boc_df.close[i]]
    list_data.append(list_cont)
```

在图像相关参数设置完成之后，可以得到绘制的蜡烛图如图 4-2 所示，时间刻度选取方面选取主要时间刻度月度时间，每 12 月取一次；次要时间刻度选取为周时间，显示时间角度为倾斜 50 度。

从股价变动的整体趋势来看，本案例的预测结果还是和实际情况有一定的吻合程度的。

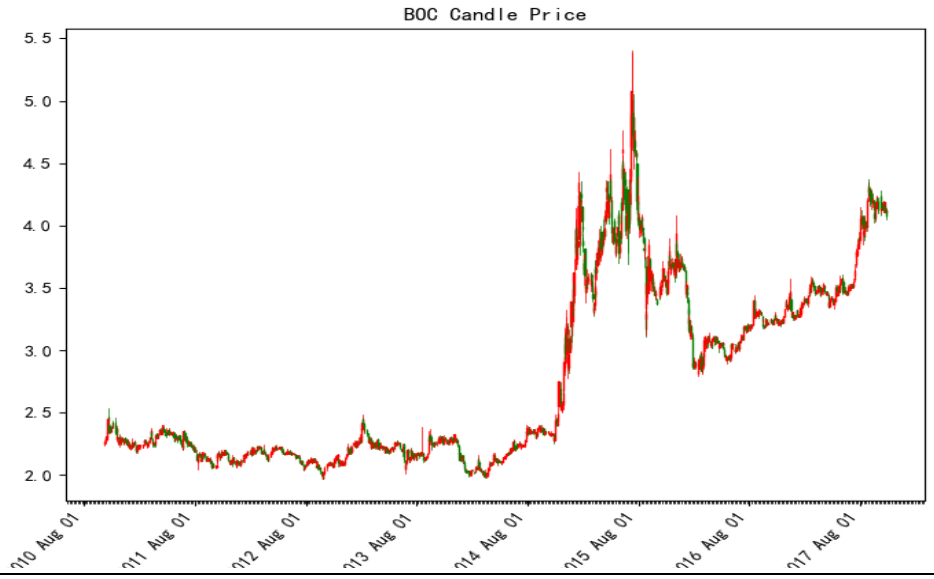

图 4-2 生成股票数据的蜡烛图

4.4 拓展研究二：与传统时间序列模型(ARIMA)对比

在上述实验中，将获得的数据存储在 csv 文件中，本实验将使用导出的数据，利用传统统计分析模型 ARMA 对中国银行未来收盘价进行预测。得到时间序列数据图如 4-3 所示，从

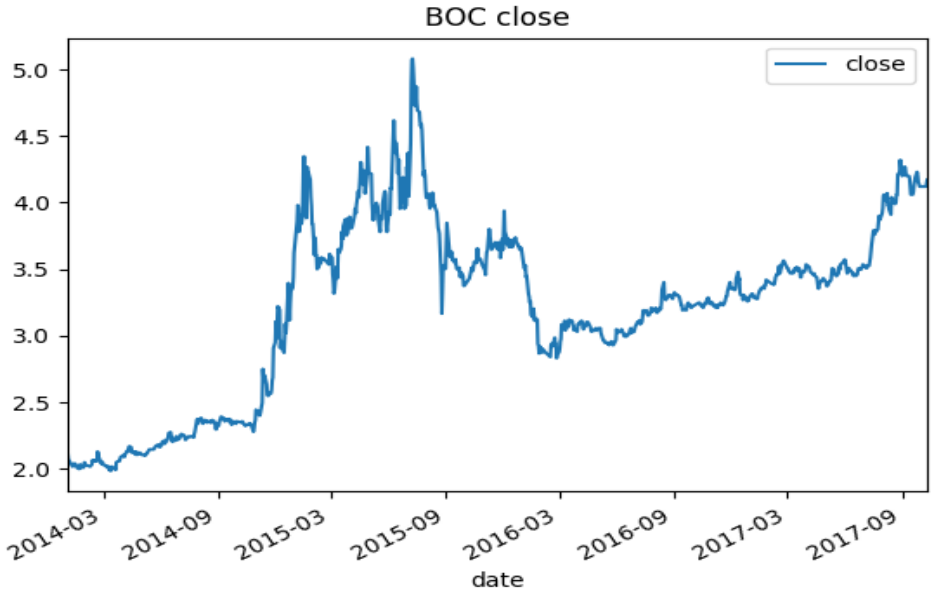

图 4-3 BOC 收盘价的时间序列数据

时间序列数据中很容易看出，该数据不平稳。

利用 ADF 单位根对数据的检测结果如表 4-5 所示，从 ADF 检验数据结果来看，我们需要将数据进行 4 阶滞后可以得到一个比较平稳的时间序列数据，其中统计量为-1.737，可以拒绝原假设，接受备择假设，认为该序列为平稳序列。

利用 Ljung-Box 检验序列是否为白噪声，LjungBox 检验值为 0.0，可以拒绝原假设，接受备择假设，认为序列不是白噪声。

Augmented Dickey-Fuller Results	
Test Statistic	-1.737
P-value	0.412
Lags	4
Trend: Constant	
Critical Values: -3.44 (1%), -2.86 (5%), -2.57 (10%)	
Null Hypothesis: The process contains a unit root.	
Alternative Hypothesis: The process is weakly stationary.	

表 4-5 BOC 时间序列数据 ADF 检验结果

对数据进行一阶差分处理，检测得到 ACF 和 PACF 收敛阶数为 0 和 1，因此认为序列满足模型 ARIMA(0,1,1)，通过残差平稳性检验后，利用该模型进行预测，预测得到数据如表 4-6 所示[6]。

日期	预测数据	实际数据	数值偏差
2017-10-12	4.173168	4.18	-0.006832
2017-10-13	4.175405	4.15	0.025405
2017-10-16	4.177642	4.14	0.037642
2017-10-17	4.179878	4.15	0.029878
2017-10-18	4.182115	4.17	0.012115
2017-10-19	4.184352	4.16	0.024352
2017-10-20	4.186588	4.11	0.076588
2017-10-23	4.188825	4.16	0.028825
2017-10-24	4.191062	4.11	0.081062
2017-10-25	4.193298	4.12	0.073298

表 4-6 ARIMA 预测数据与实际数据对比

从预测结果来看，ARIMA 模型得到的预测结果呈单调递增的状态，数值准确率与线性模型相近。

所以综合实验结果来看，机器学习模型与统计分析模型可以结合使用，以求更加精确的预测结果。

5.错误总结

1. 获取数据包的使用中出现的错误

最初使用 pandas.io.data 接口，直接想从 yahoo 获取 MSFT 的数据，但是发现接口已经被更新。

尝试一，在 anaconda2 中安装了包 pandas_datareader，继续获取数据，但是出现如下错误：

requests.exceptions.SSLError: ("bad handshake: SysCallError(10054, 'WSAECONNRESET')",)

怀疑和大陆链接外网有关系。具体错误信息见图 5-1。

尝试二，使用 Tushare 包获取数据，成功获取数据。

2. 编写绘图函数时遇到时间转换错误

[6] 由于时间关系，这个 ARIMA 模型做的比较匆忙，结果可能不准确

直接调用 last_date.timestamp()方法将当前日期转换成为秒数，遇到 TimeStamp Object has no attribute timestamp 的错误，后来在 stackoverflow 上使用 time 包中的构造方法处理了这个问题，last_sec = time.mktime(last_date.timetuple())

3. 日期索引建立 BUG

使用函数 get_hsit_data()获取到股价数据之后，发现数据是按照时间日期的顺序降序排列，因此当所有数据集训练完成之后，发现预测的不是未来数据，而是起始数据(在本案例中就是 2014-1-1 附近的数据)，后来查找多种解决方案，包括使用 reset_index.sort_values('date')等方法，但是重新建立索引后，在绘图函数中又会出现数值类型不匹配错误，最终改用函数接口 ts.get_k_data(ktype='D')，同时重新建立日期索引 BOC.index = pd.to_datetime(BOC['date'])完成按照日期升序的数据采集工作，但是保存到 csv 文件，会发现此时数据文件中存在了两个 date 列，因此需要将重复的列删除，使用数据框处理函数

$BOC.drop([BOC.columns[0]], axis=1, inplace=True)$ 将非索引的 date 列删除，CSV 问题处理完成，至此 BUG 解决。

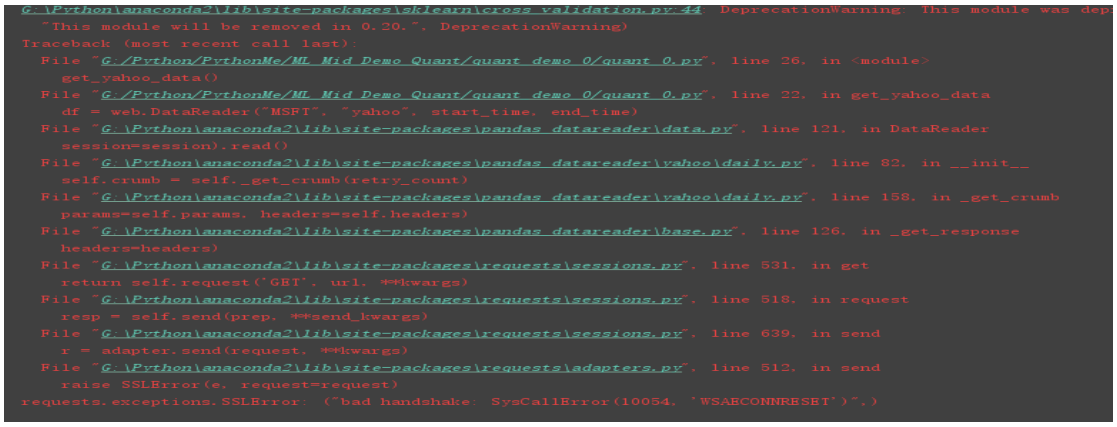

图 5-1：bad handshake Error

6.参考文献

1.机器学习算法可近似性的量化评估分析[J].江树浩
2.基于机器学习的版权金融化价值评估模型研究[J].钟媛
3.基于理性指标的马尔科夫链故事态势预测方法[J].姚宏亮
4.基于支持向量机方法的金融时间序列研究[J].张立霞
5.深度学习的金融实证应用[J].苏治
6.投资组合分析中的指标选择方法[J].郝善勇
7.智能投顾的发展现状及监管建议[J].姜海燕