对单个股票的支持向量机择时策略

杨昀昶[1]

（西安邮电大学经济与管理学院,陕西省西安市 710121）

通信作者联系方式：（Email:yangyunchang001@gmail.com 电话：029-68801085 手机：18292577417）

投稿日期：2018-7-27,

作者简介：

杨昀昶（1997-），男，本科，从事量化投资与市场微观结构理论的研究。

Email:yangyunchang@gmail.com

摘 要 随着量化投资的快速发展，各种新兴的学科也开始在金融领域大放异彩。而量化投资研究中，一个核心的问题是，对股价的预判。如果能够对股价有效的预判，则投资者即可在投资中获得超额的收益。而股价又是一种极度不稳定，又难以预测的时间序列。因此，找到合理的影响股价波动的因素和适当的算法，就是量化投资急需解决的问题。

本文认为，传统的时间序列分析方法中，又大多偏向于线性的分析方法，而忽略了股价波动的非线性因素。因此，本文利用模式识别中常用的一种感知机——支持向量机对股票的相关数据进行建模，以预判下一交易日个股股价是上升还是下跌，并以此为依据进行股票交易。

本文首先介绍了策略概况；其次，以文献综述形式分析了国内外相关的研究进展；再次，介绍了模型基本背景和算法；最后，通过回测分析来评价策略的各方面表现。

策略回测结果较好，但回撤幅度较大，本文认为与当时市场环境存在很大关系。策略 2015 年 1 月 5 日至 2018 年 7 月 30 日的回测结果为：总收益率 159.55%(对比同仁堂股票 31.45%)，年化收益率 53.51%，夏普率 0.899，最大回撤 29.615%。

关键词 支持向量机 感知机 股价预测 量化投资

Abstract With the rapid development of quantitative investment, various emerging disciplines have begun to shine in the financial field. In the quantitative investment research, a core issue is the pre-judgment of stock prices. If the stock price can be effectively pre-judged, the investor can get excess income in the investment. The stock price is an extremely unstable and unpredictable time series. Therefore, finding a reasonable factor affecting stock price volatility and appropriate algorithms is a problem that quantitative investment needs to be solved urgently.

This paper believes that the traditional time series analysis methods are mostly biased towards linear analysis methods, while ignoring the nonlinear factors of stock price fluctuations. Therefore, this paper uses a perceptron, commonly used in pattern recognition, to support the stock machine to model the relevant data of stocks, so as to predict whether the stock price of the stocks will rise or fall in the next trading day,

and use this as a basis for stock trading.

This paper first introduces the strategy overview. Secondly, it analyzes the relevant research progress at home and abroad in the form of literature review. Thirdly, it introduces the basic background and algorithm of the model. Finally, the back analysis provides the evaluation of various aspects of the strategy.

1 策略概述

1.1 研究目的

随着量化投资的发展，传统的量化择时和量化选股模型已经不能满足当前量化投资者对市场预测的精度需要。尤其是传统模型中，线性模型的广泛使用并没有起到明显优于技术分析和基本分析的预测效果。因此，非线性模型开始在量化投资领域逐步成为主流。近年来，随着人工智能、机器学习的发展，大量的非线性方法开始成为量化投资的主流。由于股票价格存在极强的非线性性、不稳定性、混沌性，因此机器学习方法可以较传统时间序列分析方法，更加准确的预测股票价格走势。

传统的机器学习理论主要是 BP 神经网络。但该算法是基于经验风险最小的原则，最终解依赖于初值，因此存在过度学习，过度拟合和局部最优解的问题。而且收敛速度慢，还存在隐网络节点个数难以确定的问题。[1,2]

而近年来，支持向量机(Support Vector Machine) 在理论研究和算法实现方面都取得了突破性进展,并成为克服 “维数灾难”和“过学习”等传统困难的有力手段。关于支持向量机在经济预测中特别是股票价格预测却是刚刚起步, 很多问题有待研究和探索。[3]

而本文就是试图通过支持向量机来进行对单个股票进行择时，得出对下一个交易日的预判。

1.2 策略主要思路和创新点

本策略的主要思路是对单个股票进行涨跌预测。

首先，选定一支股票 s，获取其相关的特征后，再加上相关的标签。

其次，进行 SVM 训练。

最后，利用训练好的模型对该股票下一交易日的涨跌进行预测。

本策略的创新点在于，利用每日的交易数据来预判下一周该股票的涨跌状况。在传统的策略中，通常是用同样频率的数据区预测同样频率下的涨跌，而本文利用信息含量更大的数据去预测信息含量较小的周数据。本文认为这样做，可能能够取得更高的收益。

1.3 研究内容

本文主要内容安排如下：

第一章为策略概述。简单介绍策略进行择时的基本原理，并简单阐述了策略整体思想和整体流程，以及策略的创新点。

第二章为文献综述。通过回顾国内外研究人员利用 SVM 模型进行股价预测的相关文献，整理和分析 SVM 模型的内容及发展现状、相关概念及特点进行了简要的介绍、梳理现有的相关量化策略。

第三章为模型设定。本章通过对策略流程的详细介绍和模型构建，来呈现本策略的具体建模思路和交易流程。

第四章为回测与实证研究。本章通过对策略在广发量化平台上的回测来分析策略的具体表现状况。

第五章为结论。本章阐述了全文的结论以及策略的不足之处和未来完善的方向。

2 文献综述

2.1 SVM 模型在量化投资中的文献综述

Kim(2003)[4]利用 SVM 模型对韩国综合股价指数(KOSPI) 进行研究预测，结果表明 SVM 优于 BPN 和 CBR。HUANG Wei 等(2005)[5]提出结合多元分类的支持向量机模型，对 S&P 500 指数、日本 NIKKEI 225 指数进行研究分析，结果表明 SVM 是金融预测的有效工具，结合多元分类方法会提高预测性能。

彭丽芳等(2006) [6]利用基于时间序列的 SVM 股票预测方法，对 2002 年 3 月 14 日到 8 月 19 日的沙河股份数据进行研究分析，结果表明，与神经网络方法以及时间序列方法相比，SVM 的预测精度更高，在某些非

线性时间序列的 预测中有很好的表现，解决了传统时间序列预测 模型无法解决的非线性问题。

林琦等(2010) [7]利用基于相空间重构的 LS-SVM 模型对股票价格进行预测，发现利用相空间重构对数据进行预处理和贝叶斯优化参数后，再用最小二乘支持向量机进行股价预测可以取得更有效的结果。

丁玲娟 (2012) [8]用小波分析对原始时间序列进行去噪，并对去噪后的序列进行小波分解，得到平稳的小波序列和非平稳的尺度序列,然后对平稳的小波序列建立 ARMA 模型进行预测，对非平稳的尺度序列建立 SVM 模型进行预测，最后整合得到对原始数据的预测结果。研究表明，组合模型完全达到了预想的高精度标准。近几年，投资者关注度和投资者情绪对股票市场的影响引起研究者的广泛关注。

周胜臣等(2013)[9]基于微博搜索和 SVM 对股价进行预测，结果表明其构建的预测模型比传统的时间序列模型具有更好的预测性能和泛化能力。

Francis 和 Lijuan Cao（2003）[10]同时利用神经网络和支持向量机对单只股票的价格走势进行预测，从他们研究的实证结果可以看出，支持向量机的各项性能也是更为优异。

2.2 小结

综上所述，可以看出，SVM 模型在金融价格预测领域的进展则主要在对模型本身的优化、修改以及和其他时间序列分析模型组合，以获取更高的收益。但是，在机器学习中，特征工程才是其核心。因此，选择良好的特征有事半功倍的效果。但过往的文献中很少有利用不同特征作为输入值进行训练的文献。因此，本文试图通过自身对业务的理解，构造出一个更加合理的模型。

3 模型设定

3.1 模型假定

假设 3.1 市场是有摩擦的。买入时佣金万分之三，卖出时佣金万分之三加千分之一印花税, 每笔交易佣金最低扣 5 块钱。但不考虑冲击成本。

假设 3.2 所有证券的价格均是动态复权的。

假设 3.3 策略于每日 9 时整开始运行，不考虑时间复杂度。因此策略在 9 时 30 分前便确定了当日的交易策略，并在开盘时执行该策略。

假设 3.4 不考虑分红和相关的税收。

3.2 支持向量机

支持向量机算法由 Vapnik 等人提出，相关理论至今仍在不断的完善与发展，是一种主要运用于数据挖掘或机器学习的方法[11]。支持向量机算法在模型的复杂性与学习能力之间寻找平衡，弥补了传统神经网络学习算法多项不足，它在解决模式识别和回归问题时，性能优越[12]。

本文将简单介绍 SVM 模型，使用 SVM 进行训练和测试前的准备工作主要包括选择合适的核函数和得到适合样本的最优核函数参数。

假设在线性可分情况下的支持向量机。

定义 3.1 分隔面

存在超平面：

$$f(\mathbf{x}) = \text{sign}(\boldsymbol{\omega}\mathbf{x} + b)$$

当 wx+b 为正时取 1，反之为-1，wx+b=0 时，函数为 0.

对于每一个点x_0，其到分隔面的距离公式为：

公式 3.1 距离公式

$$\frac{1}{\|\boldsymbol{\omega}\|}\left|\boldsymbol{\omega}^T\boldsymbol{x_0} + b\right|$$

而对于支持向量来说，其距离为 0 则分类间隔ρ：

公式 3.2 分类间隔

$$\rho = \frac{1}{\|\boldsymbol{\omega}\|} = -\frac{1}{\|\boldsymbol{\omega}\|} = \frac{2}{\|\boldsymbol{\omega}\|}$$

由于最佳分类平面的间隔最大，因此问题转化为寻找合适的$\boldsymbol{\omega}$，使得$\rho = \frac{2}{\|\boldsymbol{\omega}\|}$最大。因此，该问题近一步转化为：

公式 3.3 最大距离问题

$$\text{Min } \frac{\|\boldsymbol{\omega}\|^2}{2} \quad \text{s.t. } y_i(\boldsymbol{\omega}^T \boldsymbol{x_i} + b) \geq 1$$

考虑到有噪声样本，因此引入松驰变量$\xi_i, i = 1,2,\ldots,n$ 予以解决。因此进一步演化为：

公式 3.4 引入松驰变量的最大距离问题

$$\text{Min } \frac{\|\boldsymbol{\omega}\|^2}{2} + C\sum \xi_i$$
$$\text{s.t. } y_i(\boldsymbol{\omega}^T \boldsymbol{x_i} + b) \geq 1 - \xi_i, \xi_i > 0$$

其中，使$\frac{\|\boldsymbol{\omega}\|^2}{2}$最小可以，提高泛化能力。而$C\sum\xi_i$能使误差尽量小。因此，对于 C：

定义 3.2 惩罚系数

C 是用于平衡正则化部分和经验风险部分的平衡系数。此参数被认为是惩罚参数，当 C 越大时，对错误分类的修正越大。

再利用拉格朗日方法将其转化为对偶问题，最后解得：

公式 3.5 非线性超平面方程

$$\text{f}(\mathbf{x}) = \text{sign}(\sum \alpha_i^* y_i K(\boldsymbol{x_i}, y_i) + \text{b}^*)$$

——$K(\boldsymbol{x_i}, y_i)$为核函数，应满足 Mercer 条件。用核函数来免去高维变换，直接用低维度的参数带入核函数来等价计算高维度的向量的内积，这样可以避免出现“维数灾难”的问题。同时，核函数的引入，也使得 SVM 可以训练出非线性的分隔面。

下面介绍常用的几个核函数形式：

公式 3.6 常见核函数形式

1° 多项式核：$\text{K}(\text{x},\text{y}) = \text{x}\cdot\text{y}$

2° RBF 核：$\text{K}(\text{x},\text{y}) = \text{e}^{-\gamma\|x-y\|^2}$

3° 傅立叶核：

$$\text{K}(\text{x},\text{y}) = \frac{1-q^2}{2(1-2q\cos(x-y))+q^2}$$

本文选用 RBF 核，因为 RBF 核具有良好的性态，在实际应用中表现出了良好的性能。

4 回测与实证研究

4.1 数据说明

本文的策略以同仁堂（SH.600085）为研究对象，选择 2015 年 1 月 5 日至 2018 年 7 月 30 日合共 900 多个交易日的历史数据进行实证研究。数据来源是广发量化平台的 API 接口。

策略主要的变量选择：

变量名称	说明
输入变量	收盘价/均值、现量/均量、最高价/均价、最低价/均价、成交量比值（相对前一日）、区间收益率、区间标准差
输出变量	下一交易周的涨跌

表 4.1 策略主要使用的数据

利用前 252 个交易日的输入变量，即训练集，外加标签，输入进 SVM 模型训练。其后，再用每周三的输入变量值带入预测模型，即可得到下一交易周的预测值。

定义 4.1 标签

假设标签 label 为布尔值。对于证券 s，如果在某个交易日已经收盘后的时刻 t 时，其 t+5 时刻的股价p_{t+5}大于时刻 t 的收盘价p_t，则 label 值记为 1，否则即为 0

$$label = \begin{cases} 1, p_{t+5} > p_t \\ 0, \text{tx} \end{cases}$$

如果模型返回 1，即在这周三买入同仁堂股票，反之，卖出。

以此为交易方法，可以获取超额收益。

下面说明策略使用 SVM 模型时，设定的参数。

参数名称	设定值
惩罚系数 C	1.0
核函数类型	RBF 核
核函数参数	不适用
核函数系数	自动调整
核函数独立项	不适用
概率估计	不启用
启发式收缩方式	是

误差精度 g	0.001
训练使用内存	200MB
多个惩罚系数	否
多线程运算	否
最大迭代次数	不设上限②
伪随机发生器	不启用

表 4.2 参数设定表

4.2 回测结果

本文采用广发量化平台进行回测。

假定交易成本只有买入时佣金万分之三，卖出时佣金万分之三加千分之一印花税，每笔交易佣金最低扣 5 块钱。但不考虑冲击成本。

主要结果：

项目	结果
总收益	159.55%
策略年化收益	31.45%
基准收益	53.51%
Alpha	0.227
Beta	0.522
Sharpe	0.899
Sortino	1.352
Information Ratio	0.629
Algorithm Volatility	0.305
Benchmark Volatility	0.422
胜率	0.500
日胜率	0.498
盈亏比	2.106
盈利次数	21
亏损次数	21
最大回撤	29.615%
最大回撤出现时间	2015-07-13~2015-08-25

表 4.3 策略回测结果概况

从回测结果看，策略大幅跑赢市场，且夏普率为 0.9，也算是较高的水平。策略回撤较大可能是因为没有引入止损导致的。

但总体看，策略是有效的，可以获得长期利润的。

5 结论

5.1 策略结论

策略明显跑赢大市，可以认为本策略是有效的，且可以获得长期复利增长。策略时间复杂度也在合理范围内，因此可以认为本策略可以在正常情况下完成交易。本文亦证明了支持向量机在股价预测领域有较高的预测准确率和强大的容错能力。

5.2 策略不足与展望

策略存在以下不足：

1.没有引入止损，导致回撤较大。

2.仅仅考虑单一股票，容易导致风险过于集中，蒙受损失。

3.仅仅使用最简单的 SVM 模型，过于简单。

针对这些问题，本文认为将来还可以做出如下修正：

1. 引入适当的止损。例如利用隐马尔可夫链算法，预判当前市场处于牛市、熊市亦或是震荡市，当市场环境为震荡市或熊市时止损。

2. 构建股票池，对大量股票进行选股加择时的建模，效果更好。

3. 修正 SVM 参数不断回测实验，得出最优参数。或是更换更深的网络进行机器学习，获得更高的准确率，利用 LSTM，GoogleNet，深度 Q 学习等方法。（由于当前广发平台不支持 tensorflow 模块，因此暂时无法修改）还可以，利用遗传算法将多个模型组合起来使用。亦或是采用并行算法。

参考文献

[1] ZangJia-shu, LiHeng-chao, XiaoXian-ci. ADCT domain quadratic predict or forreal-time prediction of continuous chaoticsignal [J] . Acta Physical Sinica, 2004, 53(3):710-716.

[2] LPMaguire, BRoche, TMM cginnity. Predicting achaotic time series using a fuzzy neural network [J] .

Information Sciences, 1998, 112:125-136
[3] 徐维维,高风.灰色算法在股票价格预测中的应用[J] .计算机仿真, 2007, 24 (11):274-276
[4] KIM K J. Financial time series forecasting using support vector machines[J]. Neuro computing，2003，55 (1 -2) : 307 -319.
[5] HUANG Wei, NAKAMO RIY, WANG Shouyang. Forecasting stock market movement direction with support vector machine[J].Computers & Operations Research， 2005，32(10) : 2513-2522.
[6] 彭丽芳，孟志青，姜华. 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(3) : 88 － 91.
[7] 林琦，吴少雄. 基于相空间重构的 LS-SVM 股票价格预测［J］. 福建工程学院学报，2010(3) : 300 － 303.
[8] 丁玲娟.基于小波分析和 ARMA-SVM 模型的股票指数预测分析［D］. 上海: 华东师范大学，2012.
[9]周胜臣，施询之，瞿文婷等. 基于微博搜索和 SVM 的股市时间序列预测研究［J］. 计算机与现代化，2013（4）：22-26
[10] Francis E.H. Tay, Lixiang Shen and Lijuan Cao. Ordinary Shares, Exotic Methods Financial Forecasting Using Data Mining Techniques[M].World Scientific Publishing, 2003.
[11] 吴亚军.基于非线性方法和 VaR 的均线交易系统研究[D].哈尔滨:哈尔滨工业大学, 2014
[12] 汤凌冰.机器学习在量化投资中的运用研究[M].北京:电子工业出版社,2014.

附录 3 策略代码

```
# 导入函数库
from sklearn import svm
import numpy as np

#初始化
def initialize(context):
    #设置标的
    g.stock = '600085.XSHG'
    #设置基准
    set_benchmark(g.stock)
    #过滤掉 order 系列 API 产生的比 error 级别低的 log
    log.set_level('order', 'error')
    #设置数据长度
    g.days = 22
    #设置定时任务
    run_weekly(trade,3, time='open')
#定时任务函数
def trade(context):
    prediction = svm_prediction(context)
    if prediction == 1:
        cash    = context.portfolio.total_value
        order_target_value(g.stock,cash)
    else:
        order_target_value(g.stock,0)

#结果预测
def svm_prediction(context):
    #获取标的的历史数据
    stock_data = get_price(g.stock, frequency='1d',end_date=context.previous_date,count=252)
    date_value = stock_data.index
    close = stock_data['close'].values
    #用于记录日期的列表
    date_list = []
    # 获取行情日期列表
    #转换日期格式
    for i in range(len(date_value)):
        date_list.append(str(date_value[i])[0:10])

    x_all = []
    y_all = []
    #获取特征变量 x
```

```
    for i in date_list[g.days:-5]:
        features_temp = get_features(context,date=i,count=g.days)
        x_all.append(features_temp)
    #获取特征变量 y
    for i in range(g.days,len(date_list)-5):
        if close[i+5]>close[i]:
            label = 1
        else:
            label = 0
        y_all.append(label)
    x_train = x_all[: -1]
    y_train = y_all[:-1]
    clf = svm.SVC()
    clf.fit(x_train, y_train)
    #进行预测
    prediction = clf.predict(x_all[-1])[0]
    return prediction

#获取特征值
def get_features(context,date,count=252):
    #获取数据
    df_price                                                                                    =
get_price(g.stock,end_date=date,count=count,fields=['open','close','low','high','volume','money','
avg','pre_close'])
    close = df_price['close'].values
    low = df_price['low'].values
    high = df_price['high'].values
    volume = df_price['volume'].values
    #特征变量设置
    #收盘价/均值
    close_mean = close[-1]/np.mean(close)
    #现量/均量
    volume_mean = volume[-1]/np.mean(volume)
    #最高价/均价
    high_mean = high[-1]/np.mean(high)
    #最低价/均价
    low_mean = low[-1]/np.mean(low)
    #成交量比值（相对前一日）
    volume_current = volume[-1]/volume[0]
    #区间收益率
    returns = close[-1]/close[0]
    #区间标准差
    std = np.std(np.array(close),axis=0)
    features = [close_mean,volume_mean,high_mean,low_mean,volume_current,returns,std]
```

return features