基于高斯随机过程的股票预测

摘 要

高斯过程（Gaussian Process）是一种用于建模和推断随机函数的强大工具。它被广泛应用于机器学习、统计建模和贝叶斯优化等领域。而高斯过程回归则是运用高斯过程来进行回归分析，可以通过对数据进行建模来预测连续目标变量，并提供与预测相关的不确定性估计。相比于其他回归分析方法，高斯过程回归是一种非参数的回归分析，不需要对模型结构进行假设。它可以适应各种类型的数据，并且可以对复杂的非线性关系建模，并且其还具有先验灵活性，高斯过程回归允许通过选择不同的先验分布来灵活地调整模型的先验知识。先验可以根据问题的特定领域知识或经验进行定制，从而提供更准确的预测，这也让其在金融分析等领域展现了非凡的潜力，本文即是用高斯随机过程进行股票预测的应用实践。

关键词：高斯过程、随机过程、回归分析

装

订

线

1. 问题背景

股票预测问题是金融领域的一个有趣的研究问题，而历史股价数据是大量时间序列数据，可以认为没有噪音。许多经济学家和数学家已经尝试了许多不同的监督学习方法来预测未来的股票价格，例如遗传算法、竞争性学习等等。

而在近年来的人工智能和机器学习领域，随机回归模型逐渐备受关注，其中高斯过程回归（Gaussian Processes Regression, GPR）更是展现出了强大的潜力。相比于其他回归模型，GPR 是一种非参数的回归方法，它主要具有以下优点：

非线性建模能力：GPR 能够对非线性关系进行建模，可以捕捉到复杂的数据模式和非线性趋势。通过选择合适的核函数，GPR 可以灵活地拟合不同类型的数据。

不确定性估计：GPR 提供了对预测的不确定性的估计。除了给出预测值，GPR 还可以为每个预测提供置信区间，从而量化预测的可靠性。这对于决策制定和风险管理非常有帮助。

先验灵活性：GPR 允许通过选择不同的先验分布来调整模型的先验知识。不同的核函数和超参数选择可以灵活地适应不同的数据特征和先验假设。

GPR 在许多应用领域都得到了广泛的应用，如金融预测、医学诊断、信号处理等。它不仅在理论上具有坚实的基础，而且在实际问题中具有良好的表现。因此，GPR 成为了人工智能和机器学习领域研究的热点之一，并吸引了越来越多的学者的关注和研究。

2. 理论分析

2.1 基本概念

高斯过程是高斯分布的一种推广，它表示了一个函数的概率分布，完全由均值和协方差函数来确定。高斯过程是一组随机变量，其中任意有限个变量的联合分布都是高斯分布。

假设$f(x)$是高斯过程，可以写作

$$f(x) \sim GP(m(\cdot), k(\cdot,\cdot))$$

其中m和k分别是其均值函数和协方差函数

$$m(x) = E[f(x)]$$

$$k(x_1, x_2) = E[(f(x_1) - m(x_1))(f(x_2) - m(x_2))]$$

我们假设有一个训练集 ，$D = \{(x_i, y_i) | i = 1, ...N\}$其中$x$为所有输入的矩阵，$y$为目标值向量。我们还应该假设一个噪声$\varepsilon$：

$$y = f(x) + \varepsilon$$

使用高斯过程进行回归是通过贝叶斯推断来实现的，给定适当的先验和训练数据，目的是得到一个后验分布，。然后，对于新的测试输入，我们可以使用后验分布来得到在给定测试输入和训练数据的条件下的预测分布。

一般情况下，我们要先进行数据标准化，即把高斯分布的均值函数标准化为0。此时，我们令$f=[f(x_1),...f(x_n)]$为训练集中函数值的向量。它们的先验分布为：

$$f\sim GP(0,K(X,X))$$

其中$K(X,X)_{ij}=k(x_i,x_j)$是是使用协方差函数在给定点之间计算得到的协方差矩阵。然后，对于新的测试输入，我们可以使用后验分布来得到在给定测试输入和训练数据的条件下的预测分布。

$$\begin{bmatrix} f \\ f^* \end{bmatrix}\sim N\left(0,\begin{bmatrix} K(X,X) & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix}\right)$$

然后，使用贝叶斯定理，给定训练数据的联合后验为：

$$P(f,f^*|y)=\frac{P(y|f,f^*)P(f,f^*)}{P(y)}=\frac{P(y|f)P(f,f^*)}{P(y)}$$

其中由于f,f^*一般是在给定条件下独立的，满足条件独立公式，所以

$$P(y|f,f^*)=P(y|f)$$

所以我们可以通过将先验分布乘以似然函数来得到后验分布，其中高斯分布的似然函数为：

$$y\sim N(f,\sigma_n^2 I_n)$$

根据高斯过程的性质，给定观测数据的条件下，后验分布仍然是一个高斯分布。我们需要计算后验分布的均值和协方差矩阵。根据贝叶斯定理和高斯分布的性质，后验分布的均值可以通过下式计算：

$$m^*=K^{*^T}(K+\sigma_n^2 I)^{-1}y$$

其中m^*是后验分布的均值，K^*是测试输入X^*和训练输入X之间的协方差矩阵，下面再来计算后验分布的协方差矩阵

$$K^{**}=K^*-K^{*^T}(K+\sigma_n^2 I)^{-1}K$$

其中K^{**}是测试输入之间的协方差矩阵，综上所述，高斯过程回归的后验分布为

$$f^*\sim N(m^*,K^{**})$$

我们可以根据给定的观测数据，计算其后验分布来得到对新输入的预测。

2.2 核函数和超参数

核函数即计算协方差矩阵的函数，不同的核函数具有不同的特性，对训练结果会产生比较大的影响。常用的几种核函数有

1. **Squared Exponential Kernel**（平方指数核函数）：也称为高斯核函数或径向基函数（**Radial Basis Function，RBF**），定义为：

$$K(x,x')=\sigma^2\exp\left(-\frac{||x-x'||}{2l^2}\right)$$

2. **Matérn Kernel**（马特尔核函数）：马特尔核函数是一族核函数，根据不同的参数值可以得到不同程度的光滑性。其中最常用的是 **Matérn 3/2** 和 **Matérn 5/2** 核函数。**Matérn 3/2** 核函数的定义为：

$$K(x,x')=\sigma^2\left(1+\frac{\sqrt{3}\,||x-x'||}{l}\right)\exp\left(-\frac{\sqrt{3}\,||x-x'||}{l}\right)$$

而**Matérn 5/2核函数定义为：**

$$K(x,x')=\sigma^2\left(1+\frac{\sqrt{5}\,||x-x'||}{l}+\frac{5\,||x-x'||}{3l^2}\right)\exp\left(-\frac{\sqrt{5}\,||x-x'||}{l}\right)$$

3.**Rational Quadratic Kernel**（有理二次核函数）定义为：

$$K(x,x')=\sigma^2\left(1+\frac{||x-x'||^2}{2\alpha l}\right)^{-\alpha}$$

其中α是形状参数，控制了核函数的光滑程度

4. **Periodic Kernel**（周期核函数）：适用于具有周期性模式的数据，定义为：

$$K(x,x')=\sigma^2\exp\left(-\frac{2\sin^2\left(\frac{\pi}{p}||x-x'||\right)}{l^2}\right)$$

其中 p 是周期参数

这些核函数在高斯过程回归中用于度量不同输入之间的相似性和协方差，从而影响预测的平滑性、振荡性和拟合能力。根据具体的数据特点和问题需求，选择合适的核函数非常重要。此外，有时还可以通过组合多个核函数或引入更复杂的核函数来增加模型的灵活性和精度。

超参数（Hyperparameters）是在建立模型之前需要人为设定的参数，用于控制模型的学习

和复杂度。与模型的参数不同，超参数不是通过训练数据自动学习得到，而是由人工设定或通过经验调整。超参数的选择对模型的性能和泛化能力具有重要影响。

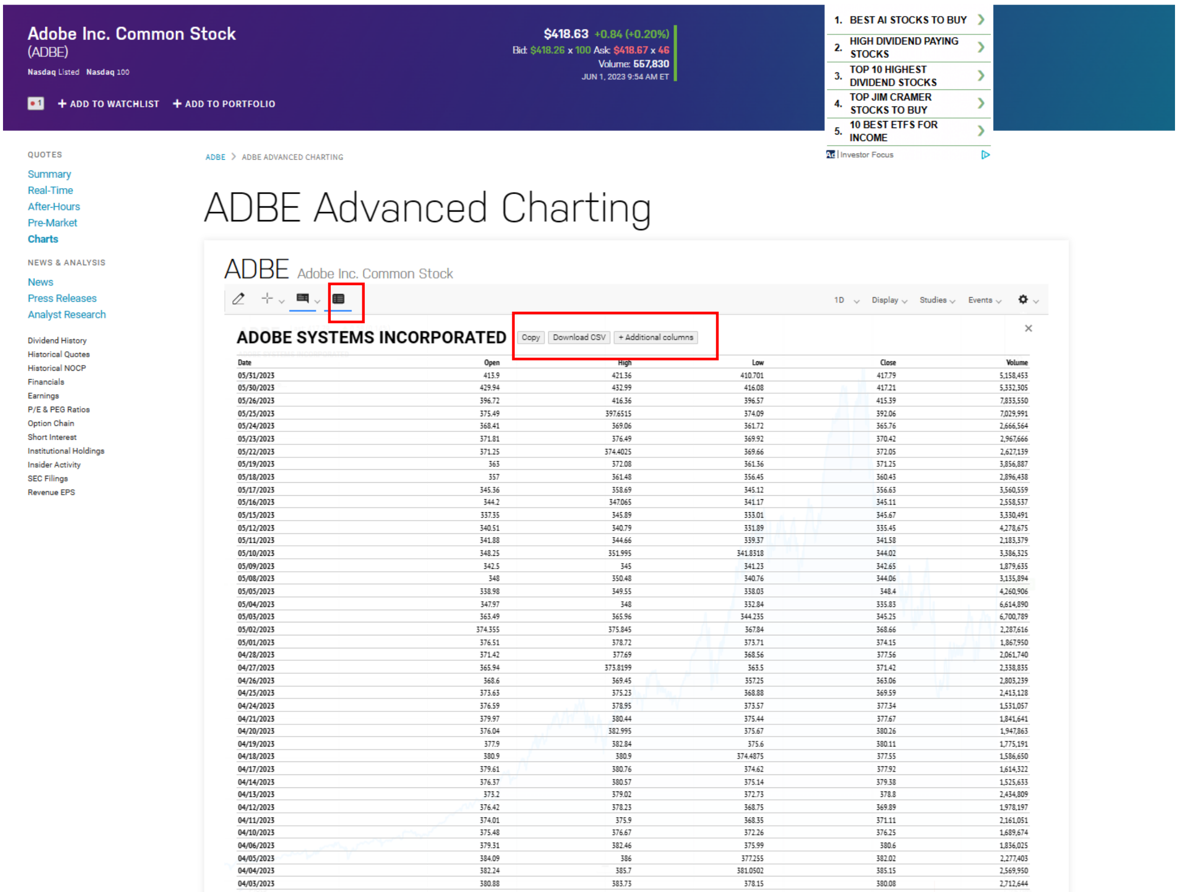
在高斯过程回归中，超参数包括噪声中方差的参数，以及核函数中的一些参数。

3. 模型实现

本项目采用交叉验证法,将原始数据集划分为训练集和测试集,由训练集数据训练模型获得最佳核函数以及最优超参数,从而获得最优拟合模型。再通过测试集评估模型的精确度及预测股票价格,获得了非常可观的预测效果。

3.1 数据导出和预处理

本次股票预测我选择了和[5]中一样的几只股票，数据来源于http://www.nasdaq.com/，数据的导出流程主要是进入股票的chart界面，然后选择Table表格模式显示，然后点击Download CSV即可下载股票的数据表。



Date	Open	High	Low	Close	Volume
05/31/2023	413.9	421.36	410.701	417.79	5,158,453
05/30/2023	429.94	432.99	416.08	417.21	5,332,305
05/26/2023	396.72	416.36	396.57	415.39	7,833,550
05/25/2023	375.49	397.6515	374.09	392.06	7,029,991
05/24/2023	368.41	369.06	361.72	365.76	2,666,564
05/23/2023	371.81	376.49	369.92	370.42	2,967,666
05/22/2023	371.25	374.4025	369.66	372.05	2,627,139
05/19/2023	363	372.08	361.36	371.25	3,856,887
05/18/2023	357	361.48	356.45	360.43	2,896,438
05/17/2023	345.36	358.69	345.12	356.63	3,560,559
05/16/2023	344.2	347.065	341.17	345.11	2,558,537
05/15/2023	337.35	345.89	333.01	345.67	3,330,491
05/12/2023	340.51	340.79	331.89	335.45	4,278,675
05/11/2023	341.88	344.66	339.37	341.58	2,183,379
05/10/2023	348.25	351.995	341.8318	344.02	3,386,325
05/09/2023	342.5	345	341.23	342.65	1,879,635
05/08/2023	348	350.48	340.76	344.06	3,135,894
05/05/2023	338.98	349.55	338.03	348.4	4,260,906
05/04/2023	347.97	348	332.84	335.83	6,614,890
05/03/2023	363.49	365.96	344.235	345.25	6,700,789
05/02/2023	374.355	375.845	367.84	368.66	2,287,616
05/01/2023	376.51	378.72	373.71	374.15	1,867,950
04/28/2023	371.42	377.69	368.56	377.56	2,061,740
04/27/2023	365.94	373.8199	363.5	371.42	2,338,835
04/26/2023	368.6	369.45	357.25	363.06	2,803,239
04/25/2023	373.63	375.23	368.88	369.59	2,413,128
04/24/2023	376.59	378.95	373.57	377.34	1,531,057
04/21/2023	379.97	380.44	375.44	377.67	1,841,641
04/20/2023	376.04	382.995	375.67	380.26	1,947,863
04/19/2023	377.9	382.84	375.6	380.11	1,775,191
04/18/2023	380.9	380.9	374.4875	377.55	1,586,650
04/17/2023	379.61	380.76	374.62	377.92	1,614,322
04/14/2023	376.37	380.57	375.14	379.38	1,525,633
04/13/2023	373.2	379.02	372.73	378.8	2,434,809
04/12/2023	376.42	378.23	368.75	369.89	1,978,197
04/11/2023	374.01	375.9	368.35	371.11	2,161,051
04/10/2023	375.48	376.67	372.26	376.25	1,689,674
04/06/2023	379.31	382.46	375.99	380.6	1,836,025
04/05/2023	384.09	386	377.255	382.02	2,277,403
04/04/2023	382.24	385.7	381.0502	385.15	2,569,950
04/03/2023	380.88	383.73	378.15	380.08	2,712,644

获取到的数据包含多个表项，如下左图，表格显示了每一个交易日该只股票开盘价、最高最低价以及收盘价和成交量，我们这里把收盘价作为每支股票的实际价值，删去无关项得到新的表格。

	A	B	C	D	E	F
1	Date	Open	High	Low	Close	Volume
2	05/31/202	413.9	421.36	410.701	417.79	5,158,453
3	05/30/202	429.94	432.99	416.08	417.21	5,332,305
4	05/26/202	396.72	416.36	396.57	415.39	7,833,550
5	05/25/202	375.49	397.6515	374.09	392.06	7,029,991
6	05/24/202	368.41	369.06	361.72	365.76	2,666,564
7	05/23/202	371.81	376.49	369.92	370.42	2,967,666
8	05/22/202	371.25	374.4025	369.66	372.05	2,627,139
9	05/19/202	363	372.08	361.36	371.25	3,856,887
10	05/18/202	357	361.48	356.45	360.43	2,896,438
11	05/17/202	345.36	358.69	345.12	356.63	3,560,559
12	05/16/202	344.2	347.065	341.17	345.11	2,558,537
13	05/15/202	337.35	345.89	333.01	345.67	3,330,491

	A	B	C	D
1	05/31/2023	417.79		
2	05/30/2023	417.21		
3	05/26/2023	415.39		
4	05/25/2023	392.06		
5	05/24/2023	365.76		
6	05/23/2023	370.42		
7	05/22/2023	372.05		
8	05/19/2023	371.25		
9	05/18/2023	360.43		
10	05/17/2023	356.63		
11	05/16/2023	345.11		
12	05/15/2023	345.67		
13	05/12/2023	335.45		
14	05/11/2023	341.58		

由于我们的模型需要按时间顺序排列，但是表格中第一项日期的数据格式在excel中无法更改，所以用到了python的csv库进行处理


```
Datadeal.py

import csv
from datetime import datetime

# 打开 CSV 文件并读取数据
with open('股票名.csv', 'r') as file:
    reader = csv.reader(file)
    data = list(reader)


# 按照日期升序排序数据
data_sorted = sorted(data, key=lambda x: datetime.strptime(x[0],
'%m/%d/%Y'))


# 迭代每一行数据并转换日期格式
for row in data_sorted:
    # 使用 strptime 函数解析原始日期字符串
    date = datetime.strptime(row[0], '%m/%d/%Y')
    # 使用 strftime 函数将日期格式化为新的格式
    row[0] = date.strftime('%Y-%m-%d')


# 将转换后的数据写回到 CSV 文件中
with open('股票名_sorted_and_converted.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerows(data_sorted)
```

	A	B	C	D
1	2016-1-4	60.31		
2	2016-1-5	60.39		
3	2016-1-6	59.54		
4	2016-1-7	55.98		
5	2016-1-8	55.06		
6	2016-1-11	55.23		
7	2016-1-12	54.79		
8	2016-1-13	50.87		
9	2016-1-14	52.65		
10	2016-1-15	49.4		
11	2016-1-19	47.59		
12	2016-1-20	47.9		
13	2016-1-21	47.87		
14	2016-1-22	49.03		

运行的结果如上图，我们得到了可以作为训练的六只股票的六个 csv 表格。

然后我们需要对数据进行标准化，为减少计算量，我们假设进行GPR的数据的均值为0，即gpr的meanfunction设置为0，这样就不必考虑meanfunction的参数。但实际上因为股票价格不能为负数，所以其均值不可能为0。针对此，我们预先对训练集数据进行去均值化处理，即

$$\mathbf{TrainY = TrainY - avg(TrainY),}$$

预测出未来的股票价格testYpd后再把其加上avg(TrainY)即可

在代码中的实现为figrgp中的Standardize参数设置为true，matlab就可以自动对输入进行标准化。

3.2 超参数估计与核函数选择

代码中训练高斯过程回归模型的代码及参数为：

```
%gpr model
gprMdl = fitrgp(trainX, trainY, ...
    'KernelFunction','matern32','BasisFunction','pureQuadratic',...
    'FitMethod','sr','PredictMethod','fic', ...
    'Standardize',true,'ComputationMethod','v', ...
    'ActiveSetMethod','likelihood','Optimizer','quasinewton', ...
    'OptimizeHyperparameters','auto');
```

具体参数的解释如下：

- trainX：训练数据的输入特征，一个包含训练样本的矩阵或表格。每行代表一个样本，每列代表一个特征。
- trainY：训练数据的输出目标。
- 'KernelFunction'：核函数的选择，这里使用的是Matérn核函数，具体是matern32。Matérn核函数是一种常用的核函数，可以用于建模光滑和非光滑的函数。
- 'BasisFunction'：基函数的选择，这里使用的是pureQuadratic，表示使用二次多项式作为基函数。

- 'FitMethod'：拟合方法的选择，这里使用的是'SR'，表示使用置信域方法（Sequential Replacement）进行模型拟合。

- 'PredictMethod'：预测方法的选择，这里使用的是'FIC'，表示使用快速交互预测法（Fast Interactive Predictions）进行预测。

- 'Standardize'：是否对输入数据进行标准化，选择为true，将对输入数据进行标准化处理。

- 'ComputationMethod'：计算方法的选择，这里使用的是'V'，表示使用特征向量方法（vectorized method）进行计算，以提高计算效率。

- 'ActiveSetMethod'：活动集方法的选择，这里使用的是'likelihood'，表示使用似然方法来选择活动集。

- 'Optimizer'：优化器的选择，这里使用的是'quasinewton'，表示使用拟牛顿法进行优化。

- 'OptimizeHyperparameters'：是否自动优化超参数，默认为'auto'，表示自动选择最优的超参数。

通过调用fitrgp函数，将训练数据trainX和trainY输入模型中，并根据给定的参数进行模型构建。

而超参数的选择我采用的是先利用likelihood函数来预先估计超参数，自动选择最优的超参数，以省去调参的繁琐过程。当然这样可能不一定会得到最精准的结果，但是由于根据先验计算后验，耗费时间过长，所以我考虑采用更加好的核函数和迭代进行弥补。

选择**Matern32**作为核函数是因为具有不同于常用的高斯核函数，此函数具有平滑度和非光滑度之间的折中特性，对于股票价格这种随机性较强的时间序列数据，其中既包含平滑的趋势成分，也包含非光滑的波动成分，所以用此核函数预测比较合理，能够较好的预测股票发展的趋势。相比较之下，周期性核函数显然不符合股票的特性，而平方指数核函数又太过平稳和光滑，不太符合股票涨跌的速度，遂选择**Matern32**作为核函数。

而超参数的选择我采用的是先利用likelihood函数来预先估计超参数，自动选择最优的超参数，以省去调参的繁琐过程。当然这样可能不一定会得到最精准的结果，但是由于根据先验计算后验，耗费时间过长，所以我考虑采用更加好的核函数和迭代进行弥补。

以下是训练模型的全部matlab代码：

```
clc;clear;close all;
%data process
data1 = readcell('xxx（股票名）_sorted_and_converted.csv');
stock1 = cell2mat(data1(:,2));
```

```
% 把 2016-1-4 到 2023-4-28 的数据作为训练集
trainX = (1:1843)';
trainY = stock1(1:1843);

%把 2023-5-1 到 2023-5-31 的数据作为测试集
testX = (1844:1865)';
testYreal = stock1(1844:1865);

% 高斯过程回归的训练
%gpr model
gprMdl = fitrgp(trainX, trainY, ...
    'KernelFunction','matern32','BasisFunction','pureQuadratic',...
    'FitMethod','sr','PredictMethod','fic', ...
    'Standardize',true,'ComputationMethod','v', ...
    'ActiveSetMethod','likelihood','Optimizer','quasinewton', ...
    'OptimizeHyperparameters','auto');
[testYpd,~,limit] = predict(gprMdl,testX);
Lower=limit(:,1);
Upper=limit(:,2);%testYpd 预测值，limit 为上限和下限
%计算误差
erravg=sum(abs(testYpd-testYreal)./testYreal)/length(testYreal);
disp('平均绝对误差为');disp(erravg);
% 计算测试集实际值在上下限的概率
y3=(testYreal-Lower>0)&(Upper-testYreal>0);
errarea=sum(y3)/length(y3);
disp('实际值在预测上下限区间的概率为');disp(errarea);
%作图
figure;
plot(trainX,trainY,'b');xlabel('时间/天');ylabel('收盘价格/美元')
hold on;
plot(testX,testYreal,'b');
plot(testX,testYpd,'m');
fill([testX;flipud(testX)], [Lower;flipud(Upper)],[0.93333, 0.83529,
0.82353],'edgealpha', '0', 'facealpha', '.5');
legend('train','testreal','testpd','uncertainty');
```

4. 结果分析

4.1 结果可视化

本次实验选取了[5]中的几种股票，分别为ADBE，ADSK，MSFT，ORCL，SAP，VRSN。我们选取2016-1-4至2023-4-28的股票数据作为训练集，选取2023-5-1至2023-5-31的数据作为测试集，训练与预测结果如下：

装 订 线

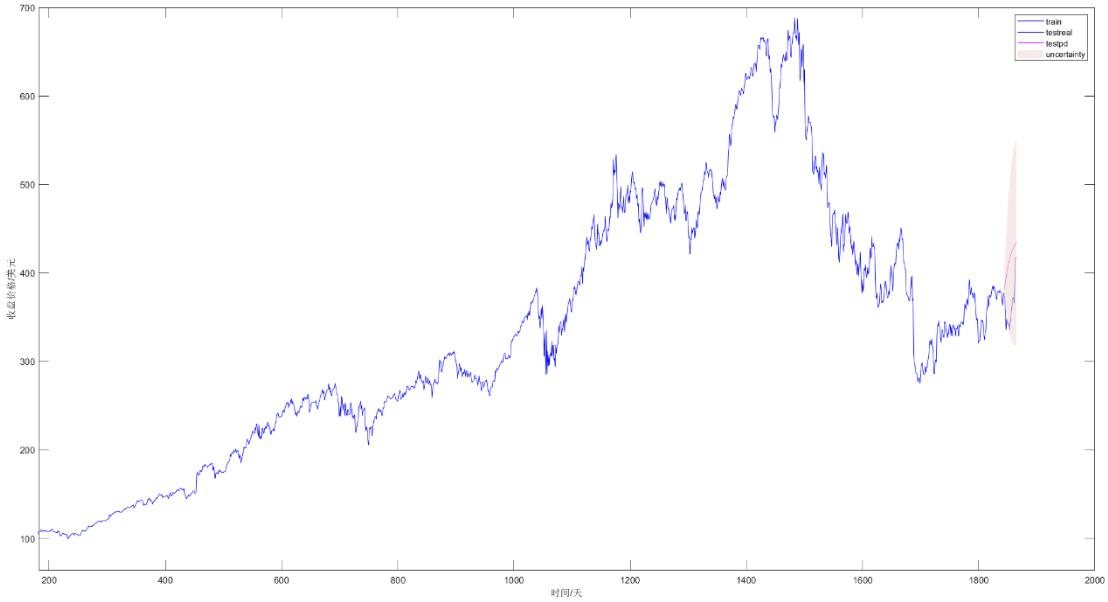


图 1 ADBE

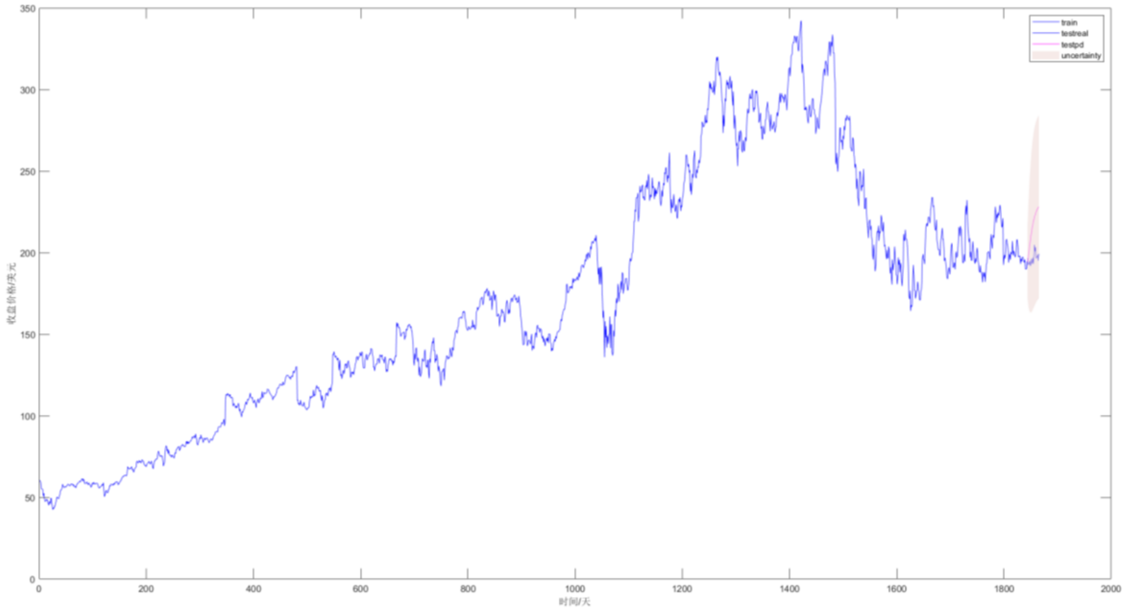


图 2 ADSK

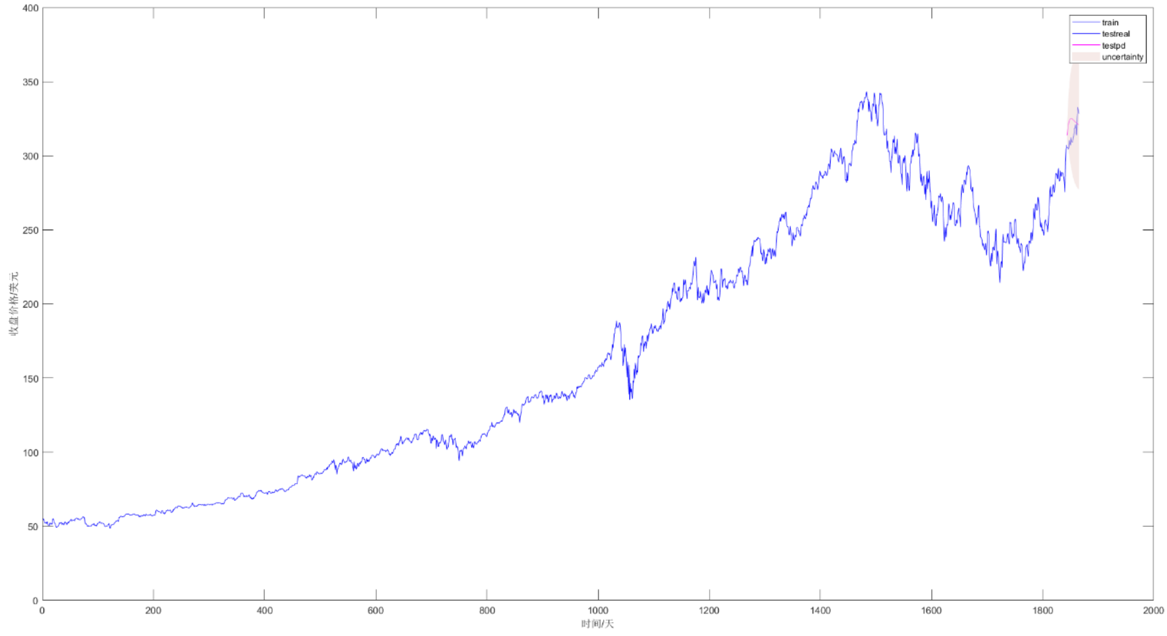


图 3 MSFT

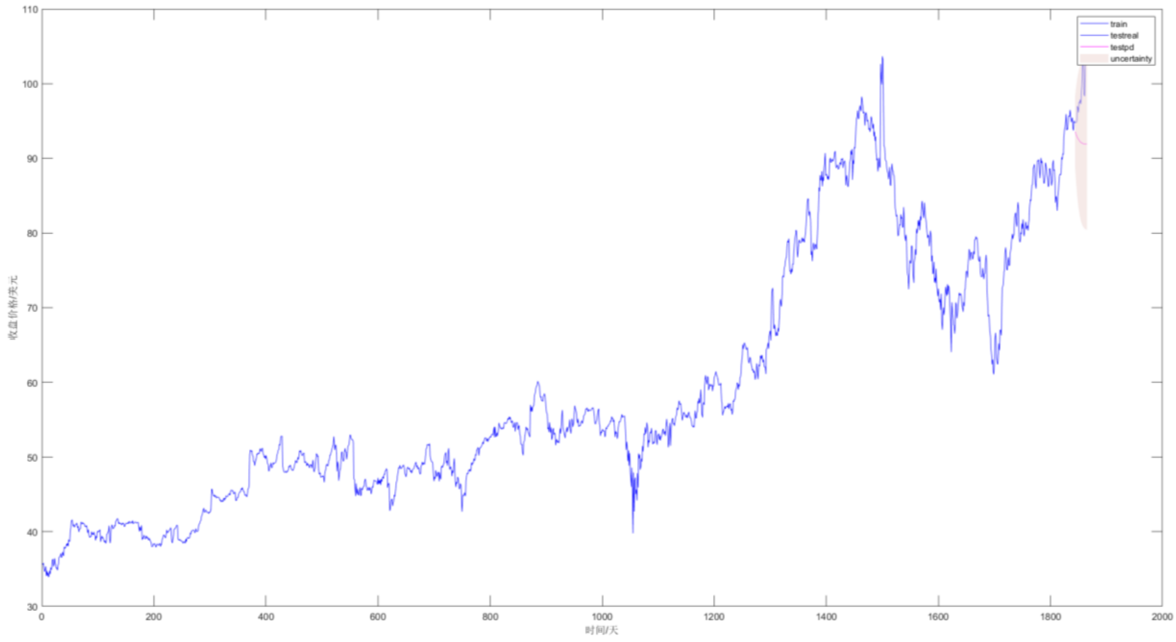


图 4 ORCL

装 订 线

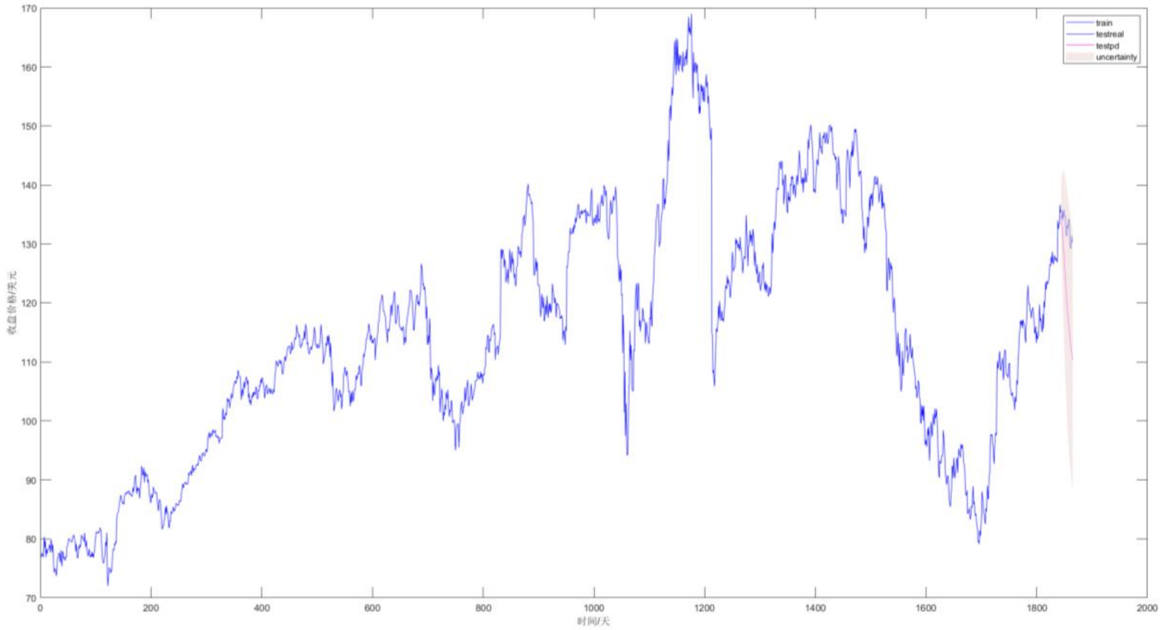


图 5 SAP

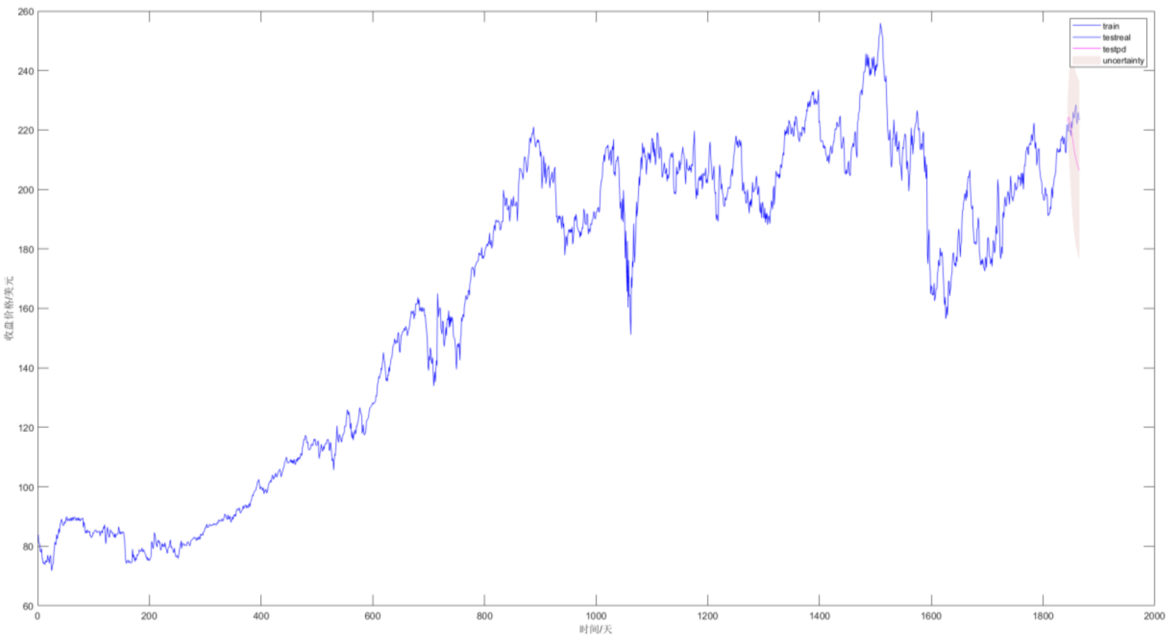


图 6 VRSN

装 订 线

不过如果对股票进行短期预测，那么准确度会更高，以下是我采用的是 2022 年 11 月 1 日到 2023 年 4 月 28 日作为训练值预测 2023 年五月的股票价格，以两只股票为例展示了短期预测的结果：

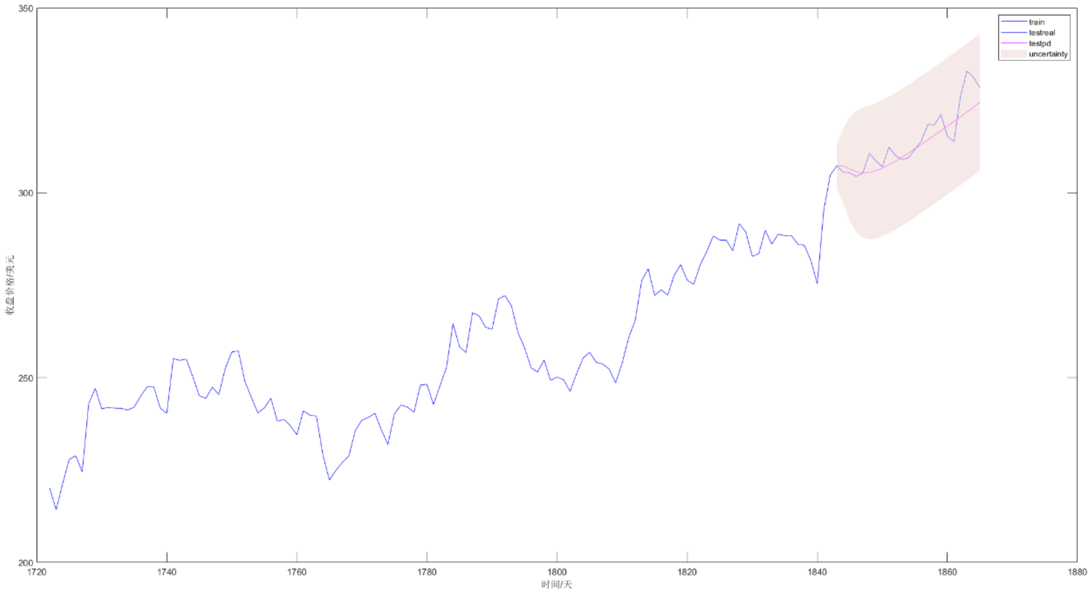


图 7 MSFT 短期预测

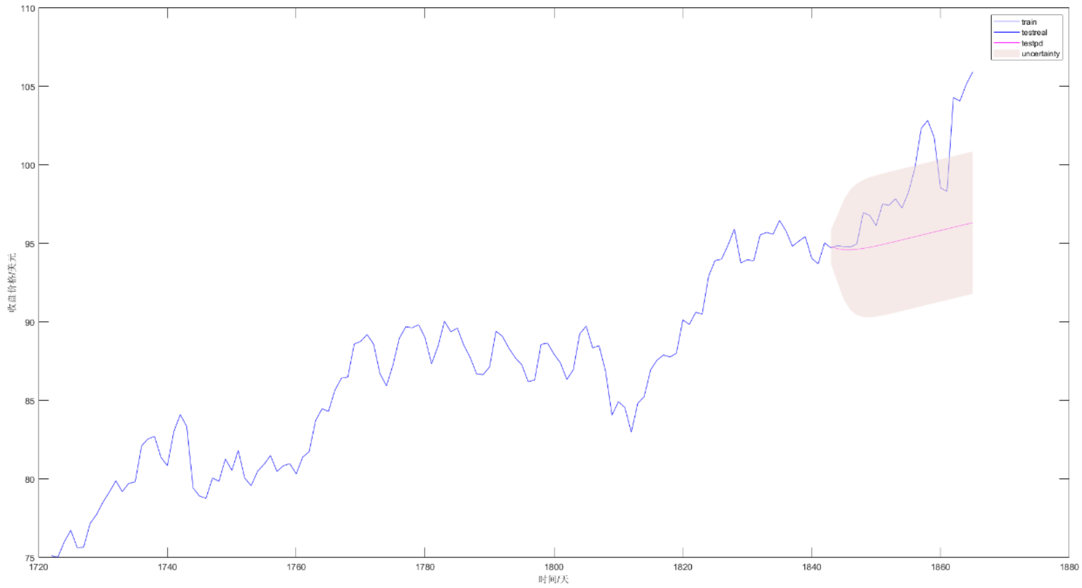


图 8 ORCL 短期预测

4.2 误差分析

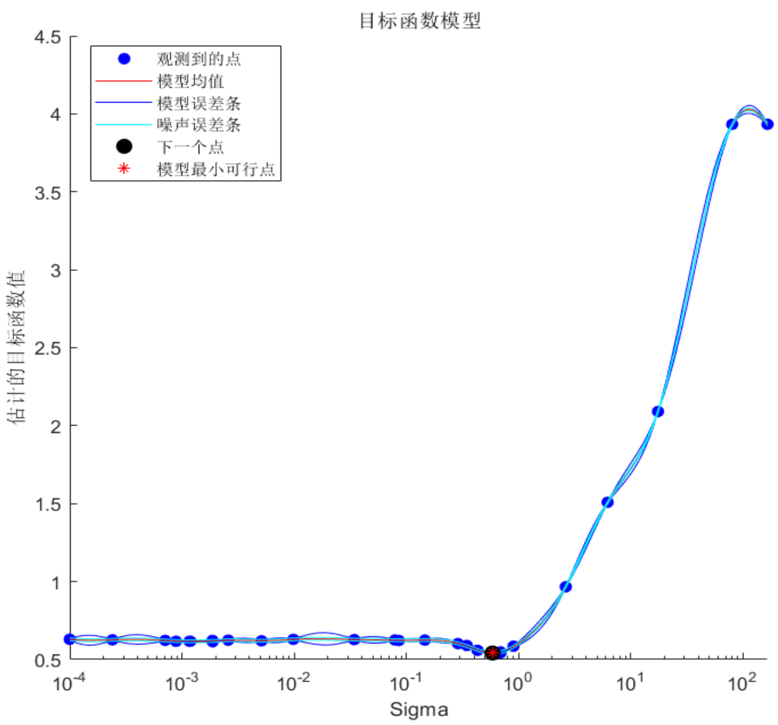
本模型在预测准确点的同时本实验还可以预测上限和下限，给预测增添了更多可参考的价值，并可求出平均绝对误差和实际值在预测上下限区间的概率（即图上实际值的曲线有多少在红色范围内）。

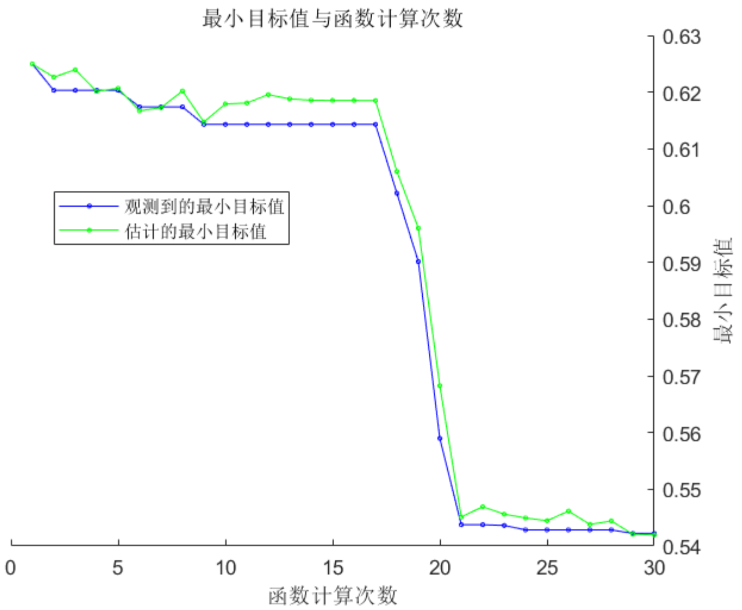
平均绝对误差为实际值与预测值绝对差除以实际值，最后对测试集数据求平均得到，误差越小越好。实际值在预测上下限区间的概率最大为1，越接近1说明测试集中预测准确的值越多。

本实验预测的平均绝对误差全部小于0.1，并且实际值在预测上下限区间内的概率均大于0.9，说明预测的准确度较高。

4.3 迭代次数与时间复杂度

经过参考文献的阅读以及资料查阅，目前主流的方法为了克服训练结果对于超参数初值的敏感性，大多数选择先用训练集先多生成一些初始值，再进行算法迭代以选择出最合理的超参数和核函数。在[5]和[6]中，这种迭代进行了大约100次，耗费资源比较大，但是本项目用fitrgp函数，自动控制超参数迭代，可自动优化不合适的超参数，无需重复优化，减少了迭代次数和时间。大约只需要30次迭代就可以得出预测值。其中一支股票的计算结果如下：






```
|========================================================================================|
| Iter | Eval   | Objective:   | Objective  | BestSoFar  | BestSoFar  |          Sigma |
|      | result | log(1+loss)  | runtime    | (observed) | (estim.)   |                |
|========================================================================================|
|    1 | Best   |       4.8646 |     70.648 |     4.8646 |     4.8646 |         92.827 ||
|    2 | Best   |       2.1689 |     91.308 |     2.1689 |     2.8808 |         3.1597 |
|    3 | Best   |       2.1256 |      236.1 |     2.1256 |     2.5361 |      0.0027029 |
|    4 | Accept |       2.1307 |     258.97 |     2.1256 |     2.4267 |       0.047772 |
|    5 | Accept |       2.1311 |     259.66 |     2.1256 |     2.1259 |      0.0033172 |
|    6 | Accept |       2.1317 |     250.07 |     2.1256 |     2.1263 |        0.67369 |
|    7 | Accept |         2.16 |     213.76 |     2.1256 |      2.126 |     0.00010024 |
|    8 | Best   |        2.121 |      223.4 |      2.121 |      2.121 |     0.00043196 |
|    9 | Best   |       1.9464 |     129.99 |     1.9464 |     1.9477 |         1.5091 |
|   10 | Accept |       2.1348 |     230.94 |     1.9464 |      1.948 |       0.016254 |
|   11 | Best   |       1.9373 |     124.98 |     1.9373 |     1.9426 |         1.5325 |
|   12 | Accept |       1.9526 |     138.98 |     1.9373 |     1.9459 |         1.5357 |
|   13 | Accept |       1.9422 |     144.83 |     1.9373 |     1.9449 |         1.5133 |
|   14 | Best   |       1.9367 |     125.47 |     1.9367 |     1.9432 |         1.5007 |
12        previousCallingContextType = 'unknown';
|   15 | Accept |       1.9517 |     242.47 |     1.9367 |     1.9445 |         1.4515 |
|   16 | Accept |       1.9392 |     147.09 |     1.9367 |     1.9434 |         1.6022 |
|   17 | Accept |       1.9724 |     129.38 |     1.9367 |     1.9468 |         1.6277 |
|   18 | Accept |       1.9407 |     143.41 |     1.9367 |     1.9457 |         1.4216 |
|   19 | Accept |       5.8127 |     52.835 |     1.9367 |     1.9459 |         522.82 |
|   20 | Accept |       3.2025 |     50.435 |     1.9367 |     1.9456 |         17.863 |
|========================================================================================|
| Iter | Eval   | Objective:   | Objective  | BestSoFar  | BestSoFar  |          Sigma |
|      | result | log(1+loss)  | runtime    | (observed) | (estim.)   |                |
|========================================================================================|
|   21 | Best   |       1.9299 |     139.95 |     1.9299 |      1.942 |         1.3752 |
|   22 | Best   |       1.9295 |     140.51 |     1.9295 |     1.9391 |         1.3406 |
900              fvec      = (dvec + tau^2) - sum(umat.^2,1)';
|   23 | Accept |       1.9547 |     168.16 |     1.9295 |     1.9416 |         1.3145 |
|   24 | Accept |        2.566 |     70.941 |     1.9295 |     1.9413 |         7.7279 |
|   25 | Accept |       2.1267 |     263.84 |     1.9295 |     1.9416 |        0.16562 |
|   26 | Accept |       2.1172 |     209.19 |     1.9295 |     1.9417 |     0.00019408 |
|   27 | Accept |       2.1481 |     222.95 |     1.9295 |     1.9418 |       0.001001 |
|   28 | Accept |       2.1458 |     229.16 |     1.9295 |     1.9419 |      0.0080886 |
|   29 | Accept |       5.8127 |     51.577 |     1.9295 |     1.9412 |         231.43 |
|   30 | Accept |       4.1334 |     46.013 |     1.9295 |     1.9407 |         37.854 |
```

装
订
线

```
优化完成。
达到 MaxObjectiveEvaluations 30。
函数计算总次数: 30
总历时: 4820.9007 秒
总目标函数计算时间: 4807.0138

观测到的最佳可行点:
Sigma
______

1.3406

观测到的目标函数值 = 1.9295
估计的目标函数值 = 1.9409
函数计算时间 = 140.5125

估计的最佳可行点(根据模型):
Sigma
______
1.4216

估计的目标函数值 = 1.9407
估计的函数计算时间 = 148.4889

平均绝对误差为
0.0424
实际值在预测上下限区间的概率为
1
```

5. 总结

本项目是我初涉随机过程领域的第一次实践，在前期通过理论学习与推导让我感受到了数学之美与随机的魅力。在实践中，不论是数据处理还是代码编写和模型调试，都让我在实践中充分理解了高斯过程回归的全过程以及各步骤的含义。本模型运用高斯过程回归的方法对股票市场的6支股票都作出了可接受的预测结果，股票价格预测的准确性是令人满意的。

高斯过程回归模型是机器学习算法之一,这也侧面说明当今机器学习等算法进入金融、医疗等各大领域的合理性，由于机器学习算法相对于传统数学方法计算方便，并且带有启发式的学习，能给投资者提供长期或短期的预测与决策，是非常强有力的工具。

当然本项目也存在一些局限性，比如由于进行了自动超参数调整，导致结果可能会对核函数的选择比较敏感，而核函数的选择也是仅仅根据其特性和股票特点进行选择，并没有尝试所有核函数并进行算法迭代找到最适合的核函数。

不过总的来说，本项目是一次比较成功的股票预测和高斯过程回归的实践。

装 订 线

装 订 线

参考文献

[1] C.K. I. Williams, C.E. Rasmussen “Gaussian Processes for Regression”

[2] M.Ebden, " Gaussian Processes for Regression An Quick Introduction".

[3] H.M.Wallace, “Introduction to Gaussian Process Regression”

[4] Z. Ghahramani, ” A Tutorial on Gaussian Processes”

[5] M.T. Farrell, et al, “Gaussian Process Regression Models for Predicting Stock Trends”.

[6] Long-term Stock Market Forecasting using Gaussian Processes

[7] J.M, Tomczak, et, al, “Gaussian process regression as a predictive model for Quality-of-Service in Web service systems”

[8] Y. Altmann, et,al, “Nonlinear spectral unmixing of hyperspectral images using Gaussian processes”.

[9] H. Topa, et,al, “Gaussian process modelling of multiple short time series”.

[10] Chuong B. Do, ”GaussianProcesses” [11] C.J. Paciorek, ”NONSTATIONARY GAUSSIAN PROCESSES FOR REGRESSION AND SPATIAL MODELING”