实验报告

xuan

一、问题背景

近年来，人工智能和机器学习领域的学者将更多的目光投向了“随机回归模型”，高斯过程回归（Gaussian Processes Regression，GPR）就是其中的典型代表。 GPR 采用如下模型

$$Y = g(X) + N$$

其中 g(X) 是一个和 X 存在紧密联系的高斯过程的样本轨道片段。

而股票价格数据最能反映股票市场的波动变化,股价的预测也成为了投资者们一直关注的热门问题，也是 GPR 的一种典型应用，对于股票交易商来说是非常重要的工具。现有的预测模型在短期预测中显示出有效的结果，然而其准确性在长期预测中会下降。 本实验将应用 GPR 来预测股市趋势，并选择了参考文献[5]和[6]中的股票来验证所提出的模型，旨在改进其预测性能和时间复杂度等, 从而帮助投资者更好地进行长期投资或验证他们的投资决策。

二、理论分析

设 x 为收盘时间，f 为收盘时间对应的收盘价格，先假设无噪声，且对股票收盘价格进行去均值的预处理。则有

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

其中 K=K(x,x)为协方差矩阵，选择不同的核函数来拟合导致 K 不同，如常见的 SE 核：$k_{ij} = \sigma_f^2 \exp(-\frac{(x_i - x_j)^T (x_i - x_j)}{2\rho^2})$

现用股票的前一段收盘时间和收盘价格作为训练集 x,f；股票的后一段收盘时间作为测试集 x^*,要预测测试集x^**对应的收盘价格*f^*

则有$\begin{bmatrix} f \\ f^* \end{bmatrix} \sim N(0, \begin{bmatrix} K(x,x) & K(x,x^*) \\ K(x^*,x) & K(x^*,x^*) \end{bmatrix})$

由贝叶斯后验公式，$f^* \sim N(\mu^*, \sigma^*)$

其中$\mu^* = K(x,x^*)^T K^{-1} f, \sigma^* = K(x^*,x^*) - K(x,x^*)^T K^{-1} K(x,x^*)$

有噪音的情况下，即 y = f(x) + n,其中$n \sim N(0, \sigma_n^2)$

$$\mu^* = K(x,x^*)^T (K + I\sigma_n^2)^{-1} f, \sigma^* = K(x^*,x^*) - K(x,x^*)^T (K + I\sigma_n^2)^{-1} K(x,x^*)$$

即训练模型的超参数为σ_f、ρ、σ_n

三、实现方法

为减少代码量，本实验使用 matlab 作为编程工具，使用 matlab 自带的 fitgpr 函数来拟合和训练高斯过程回归模型。核心代码如下：

```
testYreal = stock1(1186:1227);
gprMdl = fitrgp(trainX, trainY, ...
    'KernelFunction','matern32','BasisFunction','pureQuadratic',...
    'FitMethod','sr','PredictMethod','fic', ...
    'Standardize',true,'ComputationMethod','v', ...
    'ActiveSetMethod','likelihood','Optimizer','quasinewton');
[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);
```

3.1 数据选择与处理

本实验引入高斯过程回归模型对股票价格做预测研究,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。

其中选取的股票为[5]和[6]中国外学生们研究过的几种股票，分别为 adbe, adsk, msft, orcl, sap, vrsn；以及 starbuck 和 Hewlett-Packard（HP）。因网站限制，已无法获取当时学生研究时选取的数据（2002 年-2011 年），所以只能选择近五年（2017 年-2022 年）的数据来研究并作横向对比。

为方便起见，先在 excel 中对原始数据进行预处理，只保留收盘数据作为训练和预测的股票价格，且把所有数据按照日期升序排列。考虑到其准确性在长期预测中会下降，统一把 2017 年 1 月-2021 年 9 月的数据作为训练集，把 2021 年 10 月-2021 年 11 月的数据作为测试集。

数据集表格形式如图：

	A	B
1	01-18-2017	108.79
2	01-19-2017	109.79
3	01-20-2017	110.71
4	01-23-2017	110.97
5	01-24-2017	113.72
6	01-25-2017	114.25
7	01-26-2017	112.88
8	01-27-2017	113.99
9	01-30-2017	113.82

A 列为收盘日期，B 列为股票收盘价格

3.2 去均值化

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

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

预测出未来的股票价格 testYpd 后再把其加上$\mathrm{avg(TrainY)}$即可

在代码中的实现为 figprg 中的 Standardize 参数设置为 true,则 matlab 将分别通过列均值和标准差对预测器数据的每一列进行居中和缩放。

3.3 超参数的估计

Fitgpr 的活动集选择为 likelihood,即基于对数似然选择的回归子集。ComputationMethod 参数为'v'，即使用基于 V 方法的方法，在使用回归子集（'sr'）和完全独立的条件（'fic'）近似方法的前提下，来计算参数估计的对数似然度和梯度，能提供对数似然梯度的更快计算。虽然根据原理 gp 是不用估计参数的，可以直接训练学习，根据先验计算后验，但耗费时间较长，所以我先利用 likelihood 函数来预先估计超参数（即 kernel 里的σ_{f}、ρ、σ_n），这也会导致问题变得初值敏感，所以会用迭代和适当的核函数来选取更好的初值。

3.4 核函数的选择

Matlab 中给出的核函数有很多，但只有不带周期项的核函数才会对初值不太敏感，即以下几种：

3.4.1Squared Exponential Kernel（SE）

$$\mathrm{k_{ij}} = {\sigma_{\mathrm{f}}}^2 \exp\left(-\frac{(\mathrm{x_i}-\mathrm{x_j})^T\ (\mathrm{x_i}-\mathrm{x_j})}{2\rho^2}\right)$$

3.4.2Matern 3/2

$$\mathrm{k_{ij}} = {\sigma_{\mathrm{f}}}^2(1+\frac{\sqrt{3}r}{\rho})\exp\left(-\frac{\sqrt{3}r}{\rho}\right)$$

其中$\mathrm{r} = \sqrt{(\mathrm{x_i}-\mathrm{x_j})^T\ (\mathrm{x_i}-\mathrm{x_j})}$

3.4.3Matern 5/2

$$\mathrm{k_{ij}} = {\sigma_{\mathrm{f}}}^2(1+\frac{\sqrt{5}r}{\rho}+\frac{5r^2}{3\rho^2})\exp\left(-\frac{\sqrt{5}r}{\rho}\right)$$

其中$\mathrm{r} = \sqrt{(\mathrm{x_i}-\mathrm{x_j})^T\ (\mathrm{x_i}-\mathrm{x_j})}$

3.4.4Rational Quadratic Kernel

$$\mathrm{k_{ij}} = {\sigma_{\mathrm{f}}}^2\left(1+\frac{r^2}{2\alpha\rho^2}\right)^{-\alpha}$$

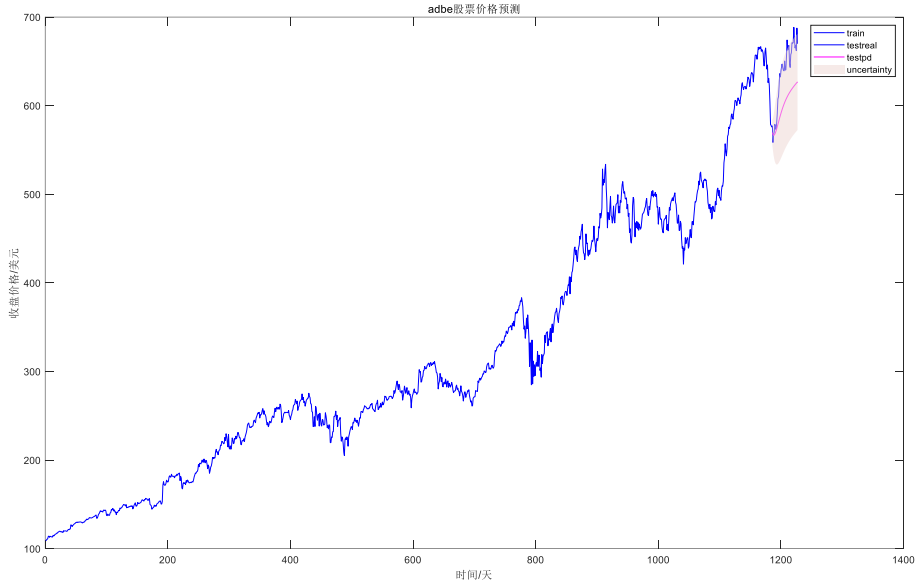
其中$r = \sqrt{(x_i - x_j)^T\ (x_i - x_j)}$

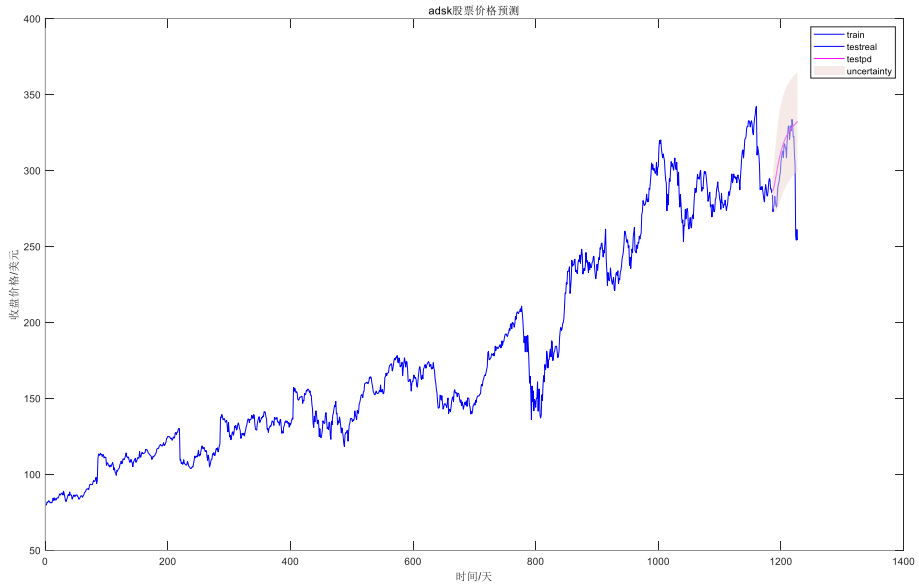
经过训练发现 Matern 3/2 的训练效果最好，预测结果更准确，且预测函数不会太平滑，所以本实验选择 Matern 3/2 作为核函数

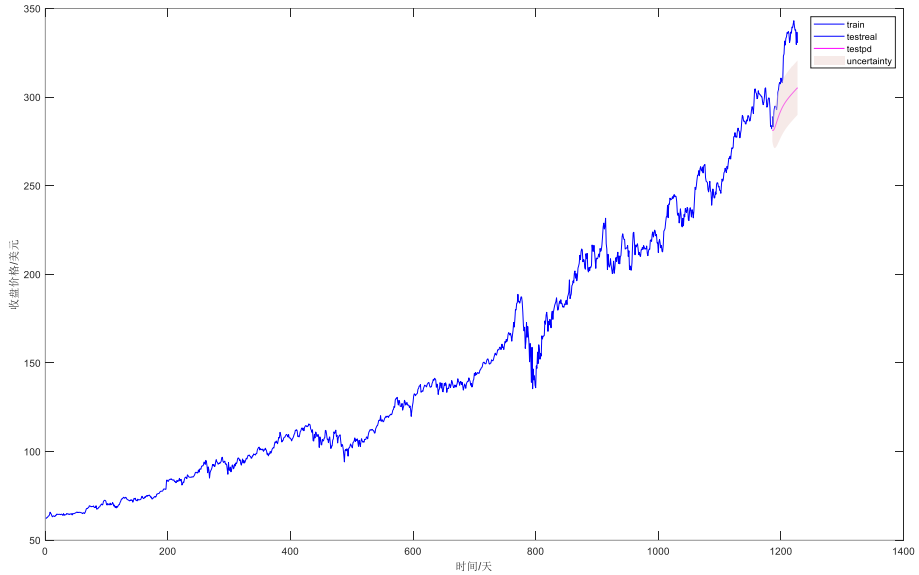
四、与其他方法对比

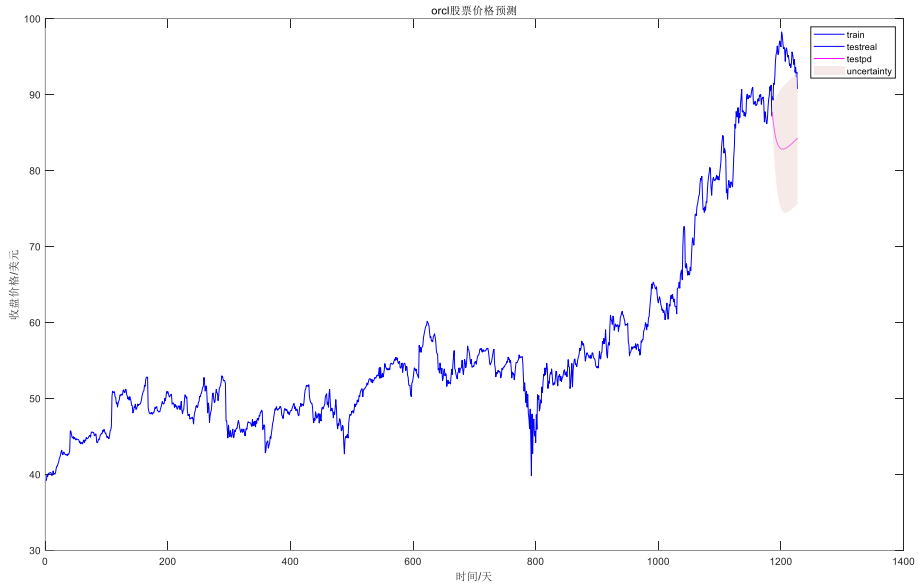
4.1 预测性能

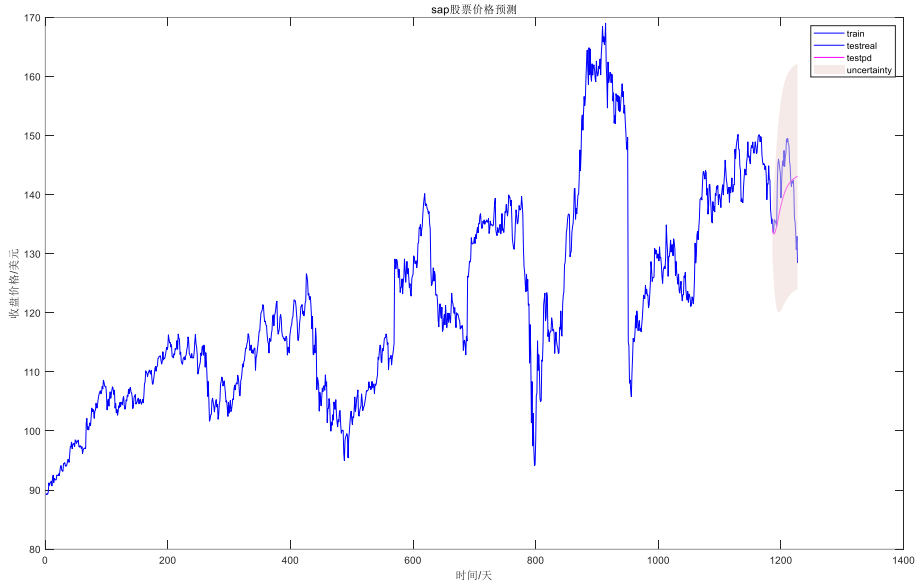
以下为 adbe, adsk, msft, orcl, sap, vrsn 的股票收盘价格预测，其中时间轴起点代表 2017 年 1 月，蓝线代表实际数据（训练集和测试集），红线代表预测的收盘价格，红色阴影代表股票价格的波动范围。显然，与[5]中的 figure1 预测图对比，预测的价格更准确，预测性能更好。

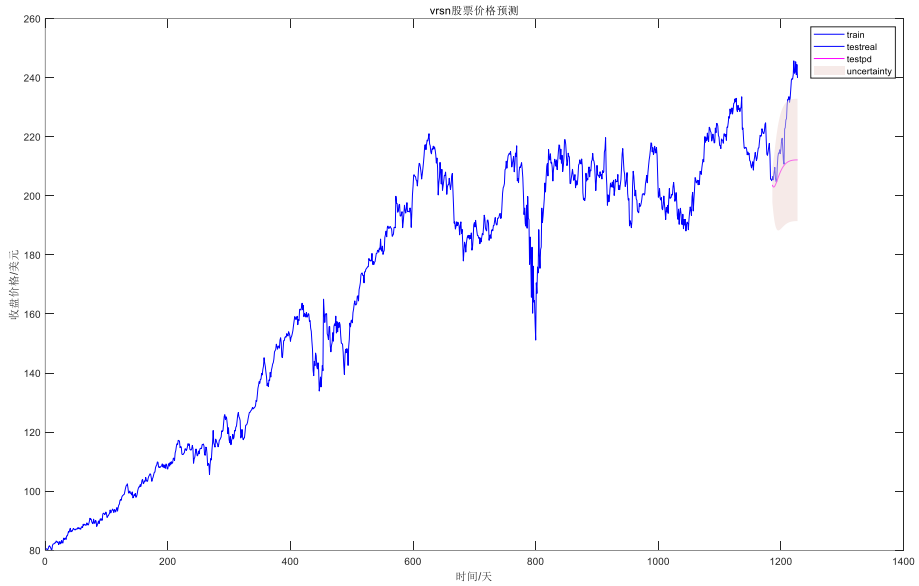



adsk股票价格预测
train
testreal
testpd
uncertainty
收盘价格/美元
时间/天
400
350
300
250
200
150
100
50
0
200
400
600
800
1000
1200
1400

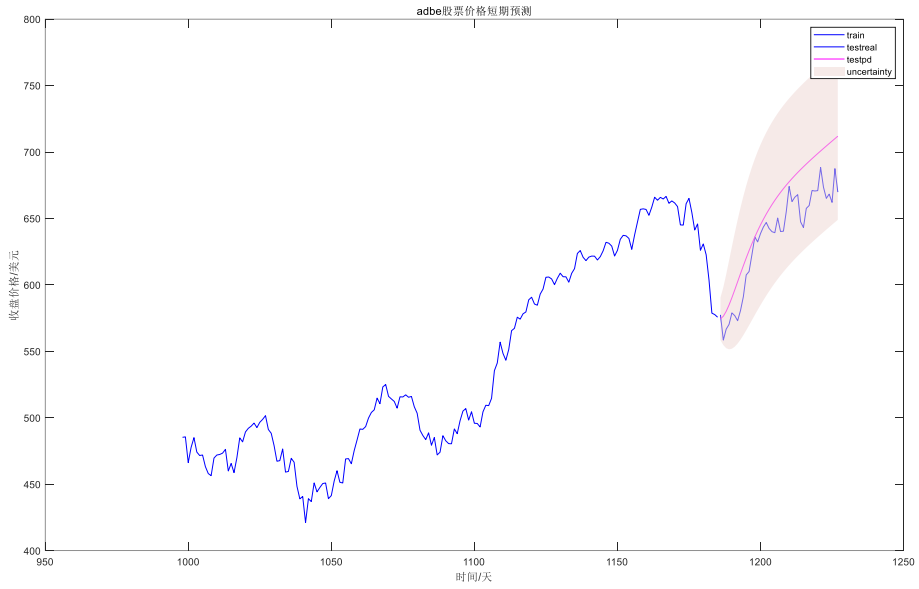

train
testreal
testpd
uncertainty
收盘价格/美元
时间/天
350
300
250
200
150
100
50
0
200
400
600
800
1000
1200
1400

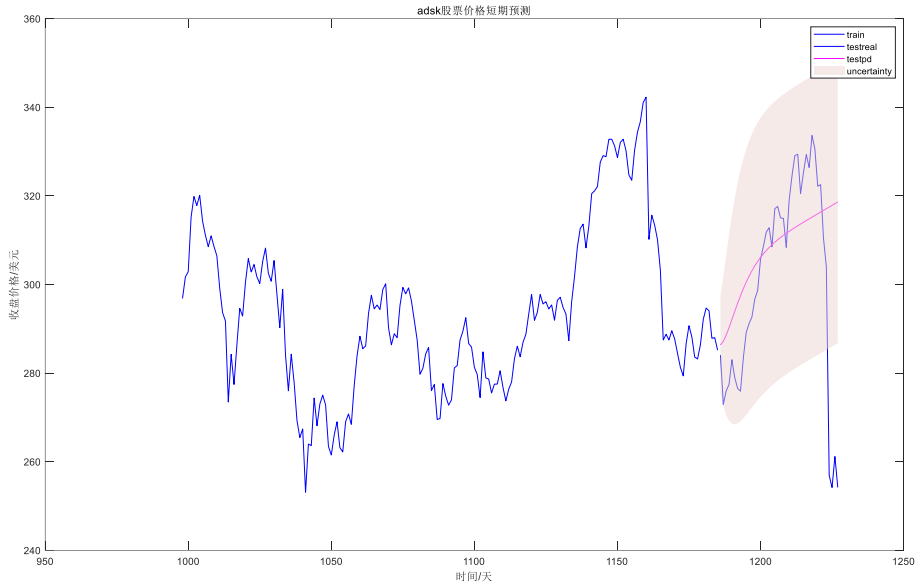

orcl股票价格预测
train
testreal
testpd
uncertainty
收盘价格/美元
时间/天
100
90
80
70
60
50
40
30
0
200
400
600
800
1000
1200
1400

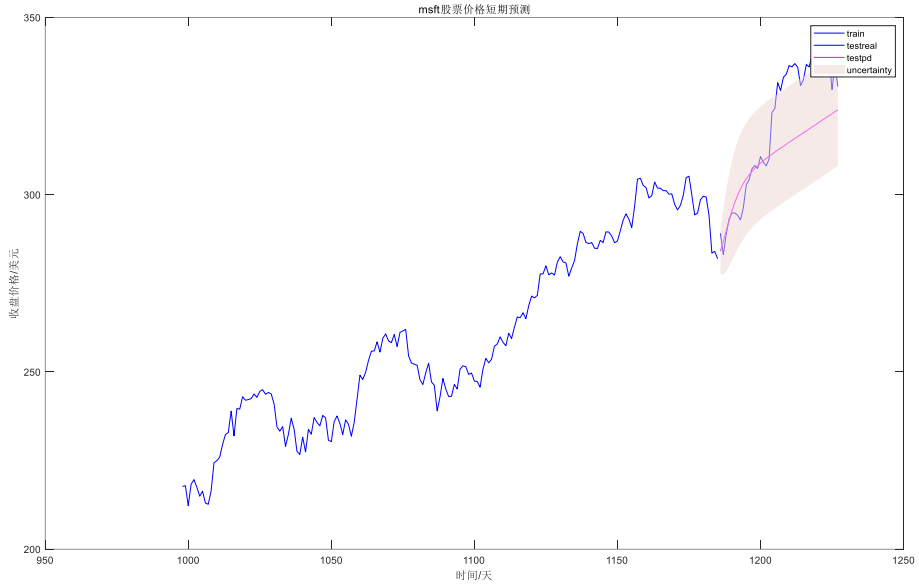

sap股票价格预测
train
testreal
testpd
uncertainty
收盘价格/美元
时间/天
170
160
150
140
130
120
110
100
90
80
0
200
400
600
800
1000
1200
1400

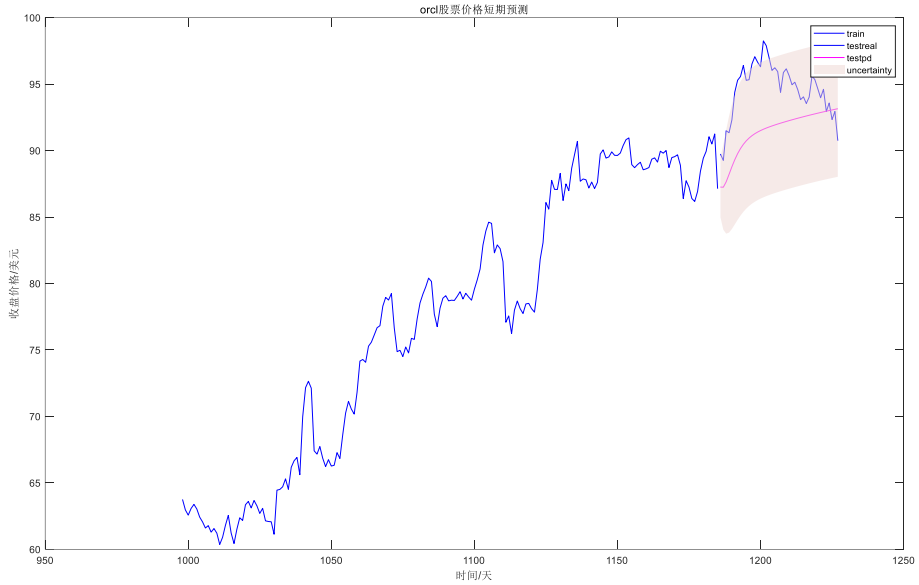


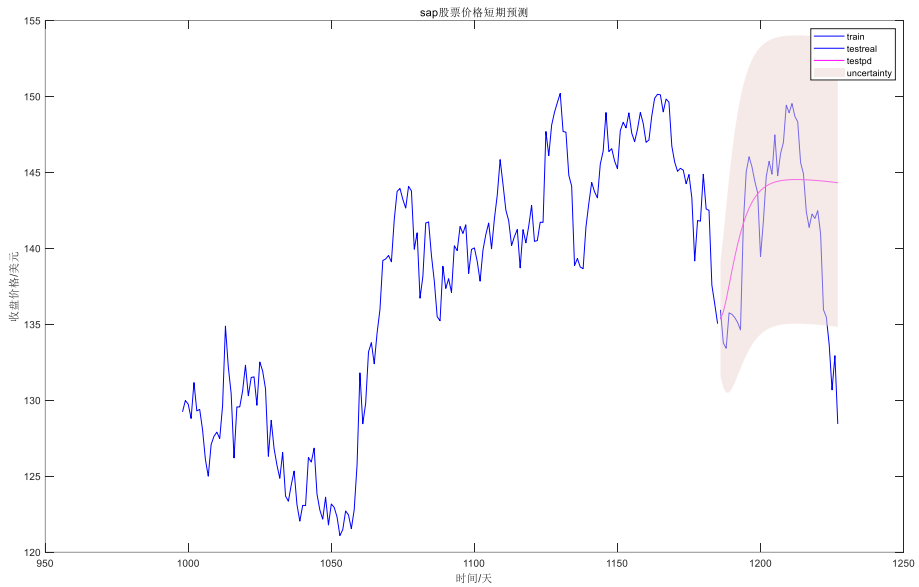
若对其作短期预测，即训练集数据只有 2021 年 1 月到 9 月，则准确度会更高，结果如下：

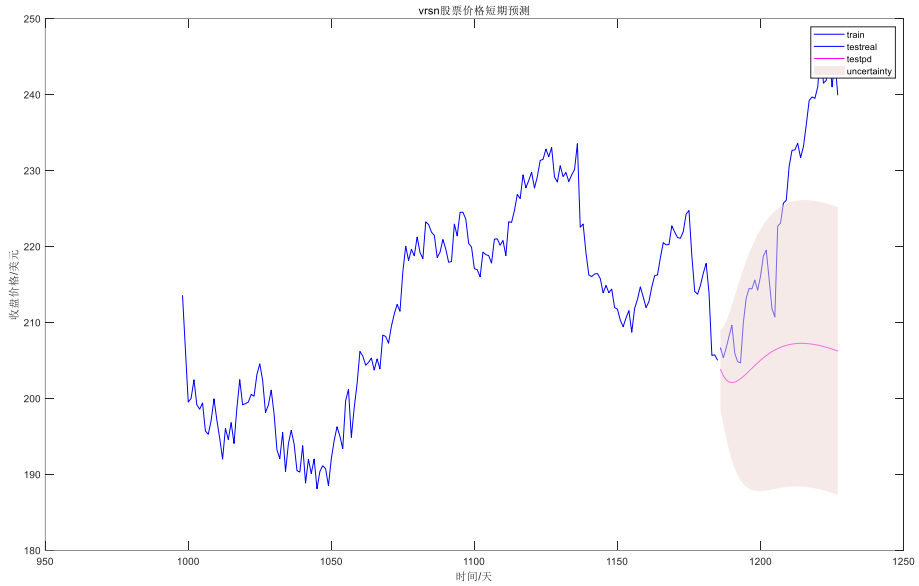



adsk股票价格短期预测
train
testreal
testpd
uncertainty
收盘价格/美元
时间/天
360
340
320
300
280
260
240
950
1000
1050
1100
1150
1200
1250

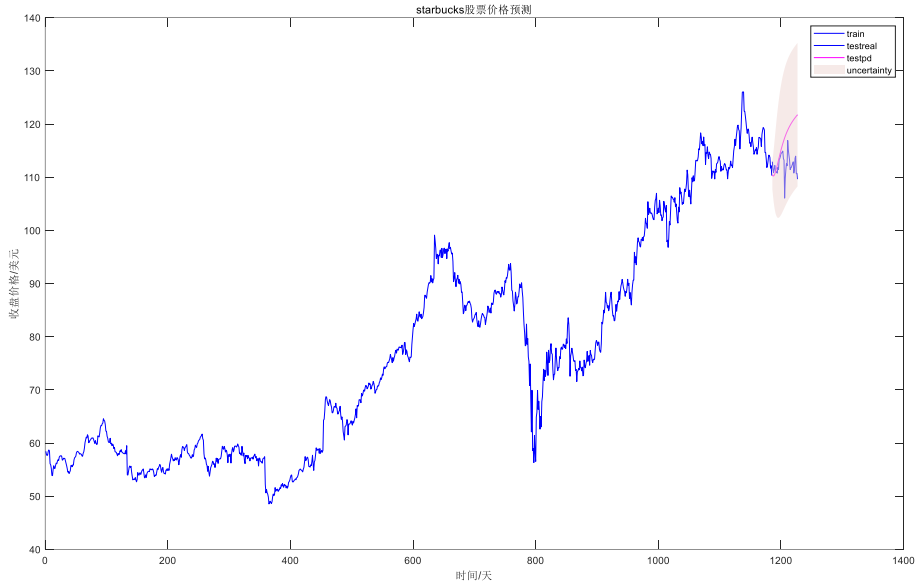

msft股票价格短期预测
train
testreal
testpd
uncertainty
收盘价格/美元
时间/天
350
300
250
200
950
1000
1050
1100
1150
1200
1250

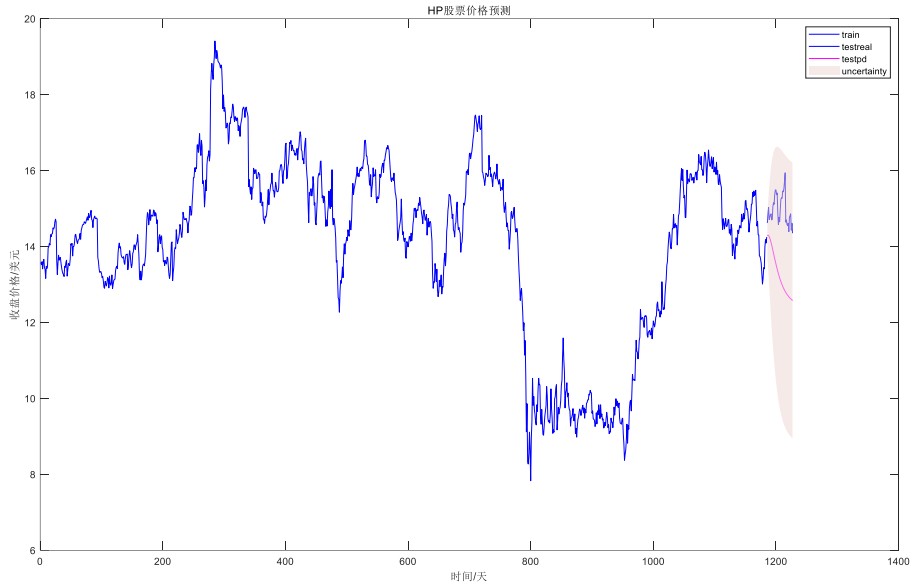

orcl股票价格短期预测
train
testreal
testpd
uncertainty
收盘价格/美元
时间/天
100
95
90
85
80
75
70
65
60
950
1000
1050
1100
1150
1200
1250


sap股票价格短期预测
train
testreal
testpd
uncertainty
收盘价格/美元
时间/天
155
150
145
140
135
130
125
120
950
1000
1050
1100
1150
1200
1250

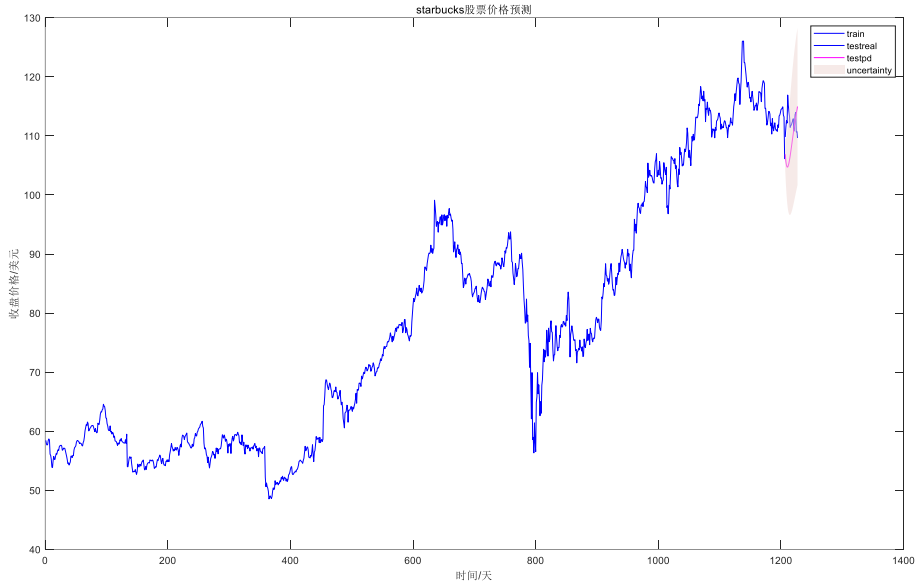


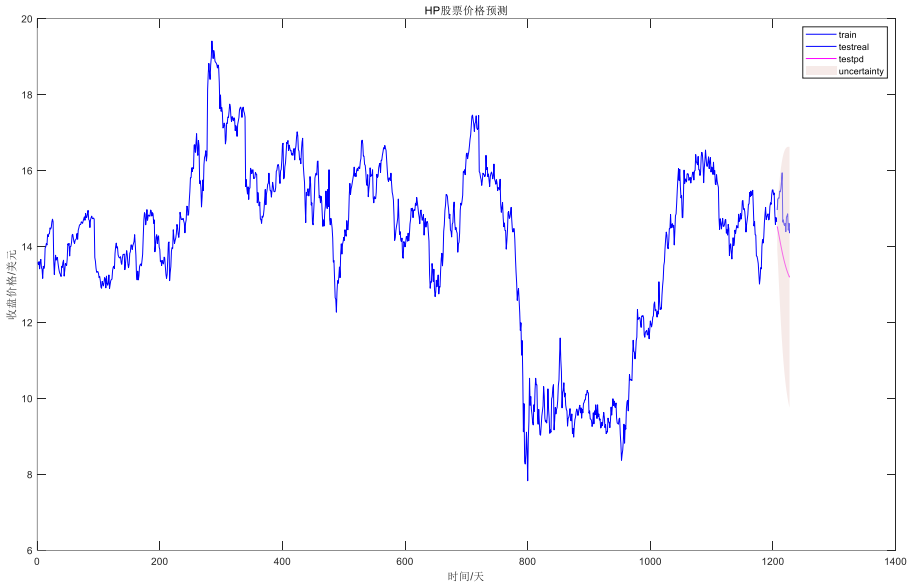
对于[6]中的 starbucks 和 HP 股票，我也使用两种方法做了预测，一是以 2017 年 1 月到 2021 年 9 月的数据作为训练集，预测 2021 年 10 月和 11 月的股票价格；结果如下：





二是以 2017 年 1 月到 2021 年 10 月的数据作为训练集，预测 2021 年 11 月的股票价格，结果如下：





可知方法二对股票价格的变动更敏感

4.2 时间复杂度

考虑到股票市场每时每刻的变动很大而 gpr 预测股票的计算量较大，有必要优化计算方法，减少时间复杂度，从而对股票市场的变动保持敏锐的反应。
训练集计算数据量大了之后，需优化协方差矩阵的计算。本实验用基于秩 1 的拟牛顿近似法来优化超参数的估计（即 fitgpr 的参数 Optimizer 设置为 quasinewton）。

牛顿法每次迭代的时间复杂度是$O\left(\log\log\frac{1}{\varepsilon}\right)$,但其每次迭代所需花销太大，即每次迭代都需要求 Hessian 矩阵并对其求逆，时间复杂度达到了O（n^3）。而拟牛顿法就是为解决牛顿法运行时间太长的问题，直接近似 Hessian 矩阵的逆。本实验中，Hessian 矩阵是密集且对称的，时间复杂度为$O(n^2)$,降低了时间复杂度。

4.3 误差区间

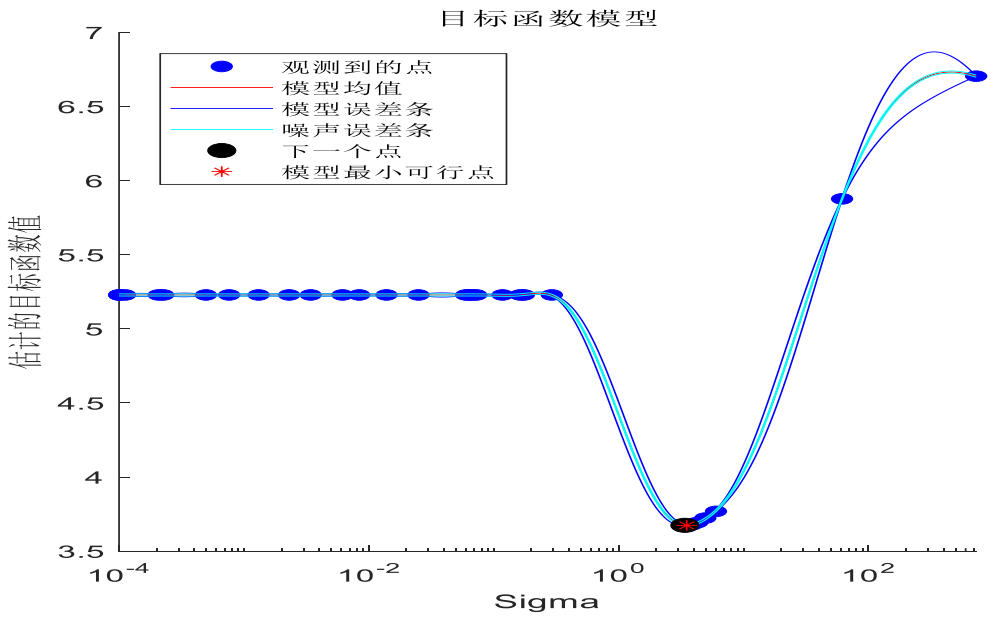
在预测准确点的同时本实验还可以预测上限和下限，给预测增添了更多可参考的价值，并可求出平均绝对误差和实际值在预测上下限区间的概率（即图上实际值的曲线有多少在红色范围内）。
平均绝对误差为实际值与预测值绝对差除以实际值，最后对测试集数据求平均得到，误差越小越好。实际值在预测上下限区间的概率最大为 1，越接近 1 说明测试集中预测准确的值越多。

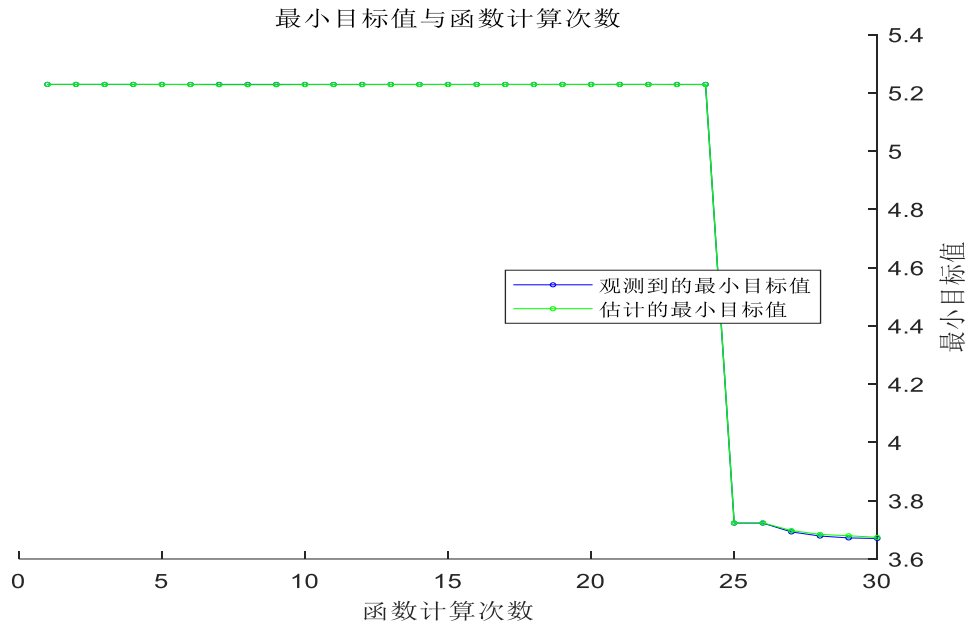
```
平均绝对误差为
    0.0557

实际值在预测上下限区间的概率为
    0.8810
```

以上几支股票的平均绝对误差都小于 0.1，且实际值在预测上下限区间的概率达到了 80%以上。(除去偏差特别大的不合理数据)

4.4 迭代次数和时间

为了尽可能克服超参数初值的敏感性，目前的主流解决方法是在具体的超参数的合理取值范围内先随机多次生成一些初始值，然后进行算法迭代，再比较最后得到的似然函数，从而选取出最合适的核函数和超参数。
[5]和[6]中为了得到较理想的结果需要迭代的次数是 100 次，迭代时间较长。而本实验通过 fitgpr 中的 OptimizeHyperparameters 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。以短期预测为例，结果如下:





Iter	Eval result	Objective: log(1+loss)	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	Sigma
21	Accept	6.7048	1.0699	5.2288	5.2286	728.18
22	Accept	5.2288	2.5476	5.2288	5.2286	0.29159
23	Accept	5.2288	2.5204	5.2288	5.2281	0.1741
24	Accept	5.2288	2.588	5.2288	5.2283	0.16313
25	Best	3.7232	1.8357	3.7232	3.7234	4.9388
26	Accept	3.7686	2.0772	3.7232	3.7241	6.0088
27	Best	3.6928	2.1562	3.6928	3.6976	4.2537
28	Best	3.6787	2.158	3.6787	3.6844	3.8735
29	Best	3.6723	1.9778	3.6723	3.6797	3.6069
30	Best	3.6704	2.2554	3.6704	3.6743	3.4512

优化完成。
达到 MaxObjectiveEvaluations 30。
函数计算总次数: 30
总历时: 72.5887 秒
总目标函数计算时间: 60.8392

观测到的最佳可行点:

Sigma
3.4512

观测到的目标函数值 = 3.6704
估计的目标函数值 = 3.6741
函数计算时间 = 2.2554

迭代次数只需要 30 次就能达到理想的结果

五、总结

通过本次大作业，我充分了解并实践了高斯过程回归的方法在股票价格预测方面的应用。预测结果表明,高斯过程回归的方法对纳斯达克股票市场的 8 支股票都呈现出可接受的预测结果，能够有效地应用于股票价格预测。而高斯过程回归模型是机器学习算法之一,本实验论证了使用高斯过程回归模型预测股票价格的可行性,理论上也论证了机器学习方法在股票市场的应用。由于该模型计算方便，能给投资者提供长期或短期的预测与决策。
但本实验也有许多局限性，一是直接使用 matlab 提供的 fitgpr 函数，其参数都是限制好的，无法自己很方便地调参，导致结果的精确性降低，如果使用 matlab 的 GPML 开源代码包来自己调参的话应该结果会更好。二是核函数的选取也并不是最合适的，未来可以尝试选取 wilson 的 SM 核来拟合，功能性更强大。三是股票数据的选取不算最好，希望未来能用更多的股票在这个模型上进行测试。

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, ”Gaussian Processes” [11] C.J. Paciorek, ”NONSTATIONARY GAUSSIAN PROCESSES FOR REGRESSION AND SPATIAL MODELING”