

信息分析与预测实践 课程设计

专业 信息管理与信息系统

班 级 信管 1611

指导老师 苏锦河

项 目 名 称： 基于SVR与ARIMA的时间序列预测研究

组长： 201621124025 刘佳昇 成 绩

目录

[基于SVR与ARIMA的时间序列预测研究 3](#_Toc13268995)

[一、 背景 4](#_Toc13268996)

[二、 主要技术介绍 5](#_Toc13268997)

[2.1 MSE均方误差 5](#_Toc13268998)

[2.2 ADF单位根检验 5](#_Toc13268999)

[2.3 ACF、PACF自相关系数与偏自相关系数 5](#_Toc13269000)

[2.4 AIC准则 5](#_Toc13269001)

[2.5 ARIMA模型 6](#_Toc13269002)

[2.5.1 自回归模型AR 6](#_Toc13269003)

[2.5.2 移动平均模型MA 6](#_Toc13269004)

[2.5.3 自回归移动平均模型ARMA 6](#_Toc13269005)

[2.6 SVR模型 7](#_Toc13269006)

[2.6.1 SVR模型概览 7](#_Toc13269007)

[2.6.2核函数 7](#_Toc13269008)

[三、 案例分析 9](#_Toc13269009)

[3.1 数据集 9](#_Toc13269010)

[3.1.1 数据预处理 10](#_Toc13269011)

[3.1.2 平稳性评估及平稳化处理 10](#_Toc13269012)

[3.2 ARIMA模型 11](#_Toc13269013)

[3.2.1 ARIMA构建流程图 11](#_Toc13269014)

[3.2.2 参数预估及检验 11](#_Toc13269015)

[3.2.3 模型检验 14](#_Toc13269016)

[3.3 SVR模型 14](#_Toc13269017)

[3.3.1 获取时间序列数据 14](#_Toc13269018)

[3.3.2 关于核函数的选取 15](#_Toc13269019)

[3.3.3 C的取值 15](#_Toc13269020)

[3.3.4 Gamma取值 16](#_Toc13269021)

[3.3.5 选取最优的C与gamma 18](#_Toc13269022)

[3.3.6 模型拟合 19](#_Toc13269023)

[四、 结论 19](#_Toc13269024)

[五、 参考文献 20](#_Toc13269025)

# 基于SVR与ARIMA的时间序列预测研究

**摘要：**ARIMA模型与SVR模型各有优缺点，但由于分别对线性模型与非线性模型处理分别具有优势，他们之间存在优势互补。本文通过同例样本分别使用这两种不同模型的优缺点进行预测，探究这两种模型的预测优劣。本文将数据进行同等预处理后，分别采用两个模型的最优参数进行预测。最后将预测值与结果通过评估均方误差MSE的方式评估模型优劣。

**关键词 ：**ARIMA、SVR、时间序列预测、MSE均方误差

# 背景

近年来,我国空气质量整体加速恶化趋势明显,极端大气污染事件频繁发生,京津冀、珠三角、长三角、关中地区等城市经济带尤为显著,最典型且影响最大的地区为京津冀区域,近期根据环保部发布的数月全国重点区域和74个城市空气质量状况月报显示,京津冀地区空气质量最差,平均达标天数比例为27.4%,低于全国32.7个百分点,全国污染最严重的10个城市中,京津冀地区占8个(http://www.cnemc.cn)。最主要的原因是京津冀区域集聚了大量的水泥、钢铁、炼油石化等高污染产业和遍布各地的无组织零散高危害产业,它们产生的大气污染物排放量非常巨大,而当地地形和气候系统又不利于污染扩散。在2013年1月,我国中东部地区发生了罕见的连续高强度的霾污染天气,造成大量航班延误、高速公路封闭、呼吸道疾病患者涌向医院急诊室。本次霾污染事件范围涉及10省市自治区,受害人口达8亿以上,其中污染最严重的是京津冀区。据统计,整个1月(1月1~31日),共计22天PM2.5超过新修订的《环境空气质量标准》中24 h平均浓度限值二级标准(75 μg/m3),27天超过一级标准(35 μg/m3),只有4天晴好天气,此次霾污染事件的详细分析结果参见文献[1]。在随后的几个月内,京津冀区域仍反复出现了数次严重的霾污染事件。霾污染事件的频繁发生为我国环境危机拉响了警报,解决经济发展与大气环境污染之间的矛盾势在必行。预测工作在解决大气污染问题中有这举足轻重的作用，所以，笔者从UCI数据库中查找2010年1月至2010年7月中的每一小时的PM2.5指数数据共计5606条。并将其对应成时间序列，分别通过ARIMA模型与SVR模型进行预测分析。

# 主要技术介绍

## 2.1 MSE均方误差

MSE用于衡量预测值和真实值之间的离差度其公式如下：

公式2.1.1

其中，是预测值，为真实值，m为数据大小

## 2.2 ADF单位根检验

单位根检验是指检验序列中是否存在[单位根](https://baike.baidu.com/item/%E5%8D%95%E4%BD%8D%E6%A0%B9/10927387)，因为存在单位根就是非平稳时间序列。单位根就是指单位根过程，可以证明，序列中存在单位根过程就不平稳，会使回归分析中存在[伪回归](https://baike.baidu.com/item/%E4%BC%AA%E5%9B%9E%E5%BD%92/4934408)。平稳是自回归模型ARMA的必要条件，因此对于时间序列，首先要保证应用自回归的n阶差分序列是平稳的。

## 2.3 ACF、PACF自相关系数与偏自相关系数

通常在时间序列分析中，采用自相关函数（ACF）、偏自相关函数（PACF）来判别ARMA(p,q)模型的系数和阶数。自相关函数(ACF)描述时间序列观测值与其过去的观测值之间的线性相关性。偏自相关函数(PACF)描述在给定中间观测值的条件下时间序列观测值与其过去的观测值之间的线性相关性。其计算公式如下：

公式2.3.1

## 2.4 AIC准则

**AIC：赤池信息准则（Akaike Information Criterion，AIC）是我们常用的判断ARIMA模型优劣的方法，其计算公式如下**

公式 2.4.1

其中k为模型参数个数，n为样本数量，L为似然函数

## 2.5 ARIMA模型

### 2.5.1 自回归模型AR

自回归模型首先需要确定一个阶数p，表示用几期的历史值来预测当前值。p阶自回归模型的公式定义为：

公式2.5.1.1

上式中yt是当前值,u是常数项,p是阶数 γi是自相关系数,Et是误差。

### 2.5.2 移动平均模型MA

移动平均模型关注的是自回归模型中的误差项的累加 ，q阶自回归过程的公式定义如下：

公式2.5.2.1

移动平均法能有效地消除预测中的随机波动。

### 2.5.3 自回归移动平均模型ARMA

自回归模型AR和移动平均模型MA模型相结合，我们就得到了自回归移动平均模型ARMA(p,q)，计算公式如下：

公式2.5.3.1

## 2.6 SVR模型

### 2.6.1 SVR模型概览

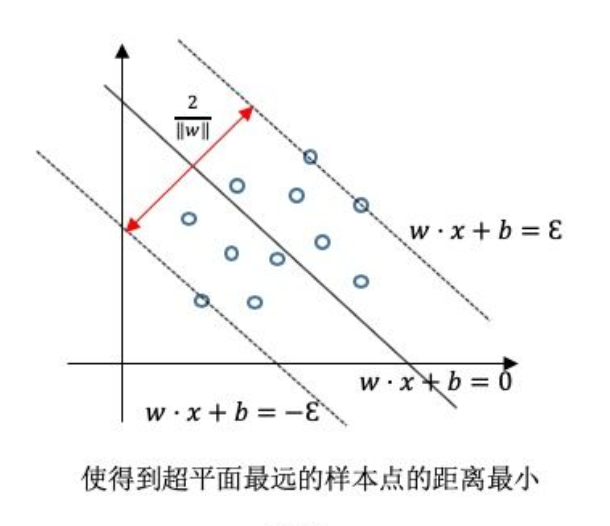


图1 SVR概要

SVR问题可以形式化为

公式2.6.1.1

其中为超平面的最小距离，C为正则化常数

为函数如下

公式2.6.1.2

可以理解为当点落在距离超平面的|z|中时，不损失，若落在之外，则执行相应损失处理

### 2.6.2核函数

核函数可以使得数据被映射到高维空间中，使其变得线性可分。更加有利于运算，本文主要用到高斯核函数（RBF），其公式如下：

公式2.6.2.1

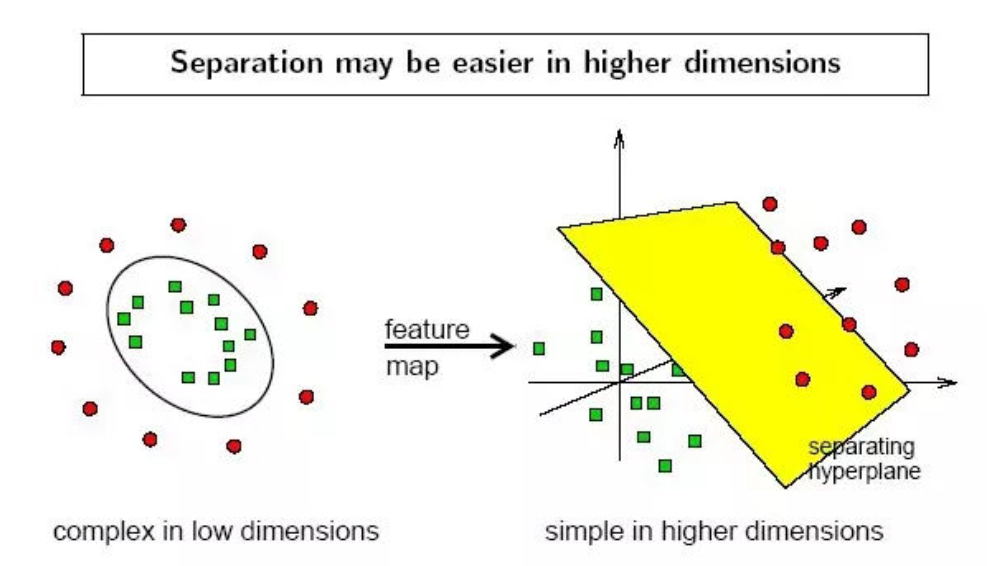


图2 核函数映射说明

# 案例分析

## 3.1 数据集

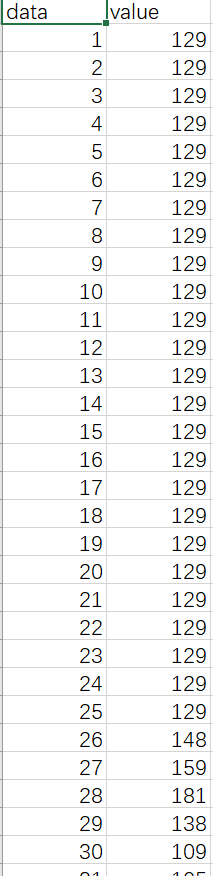


图3 数据预览

本文选举电商销量对应时间序列数据，作数据散点图，观察得数据无明显季节性

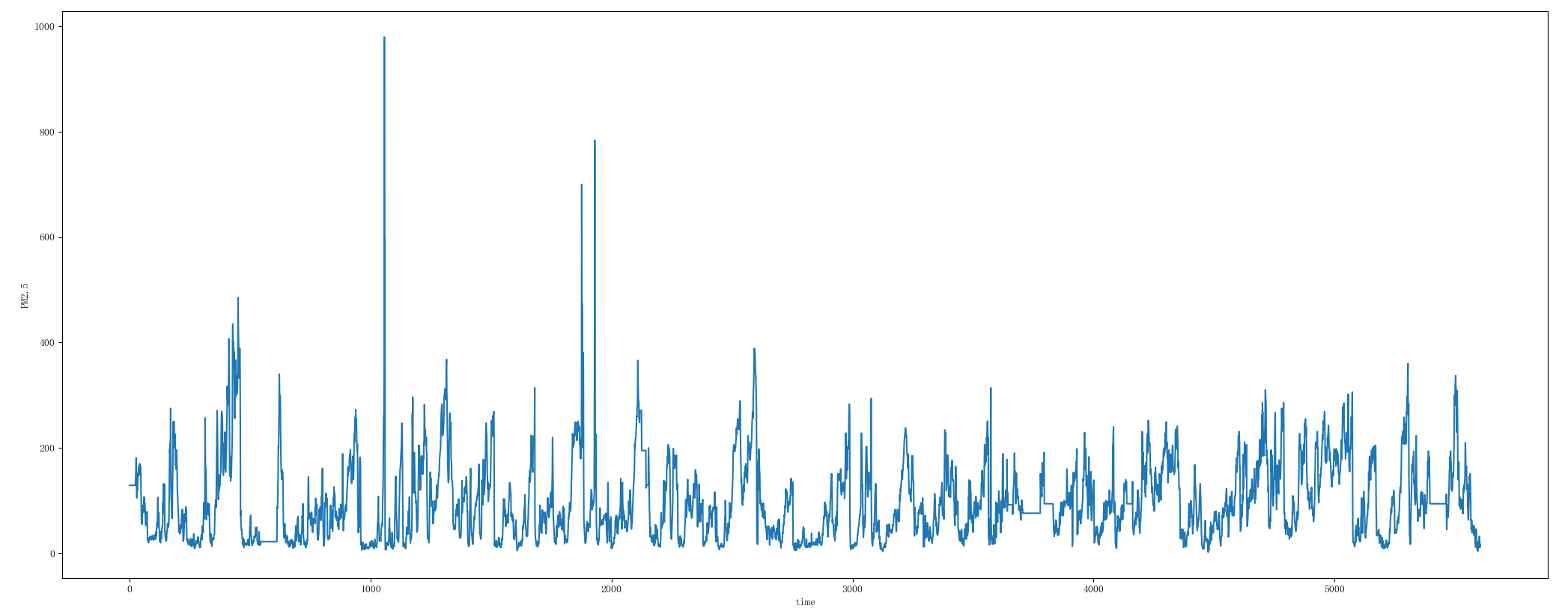


图4 2010年1月至2011年7月每小时PM2.5指数时序图

### 3.1.1 数据预处理

样本区间是北京市2010年1月-2010年7月期间每小时的PM2.5指数，共计5606条，记为时间序列。对数据分析，由于预测的数据需要保证在未来一段时间的精度，保留时间序列的连续性，所以需要剔除异常值。其中缺失数据用前后均值代替

### 3.1.2 平稳性评估及平稳化处理

由散点图可以看出该序列在0附近随机波动，波动具有稳定性没有明显的趋势变动，数据为平稳时间序列。

由于散点图带有一定的主观性，需要采用统计检验方法加以判断验证，因此对序列做单位根检验(ADF)，检验统计量结果如表所示。

|  |  |
| --- | --- |
| 表1 ADF检验结果 | |
| 1% | -3.4315174335991756 |
| 5% | -2.862055891650023 |
| 10% | -2.567044607646165 |
| 检验统计量 | -14.750283397457396 |

可以看出检验统计量小于1%、 5%、10%显著水平下的临界值（图所示），因而序列为平稳时间序列

**此部分代码如下**：

'''对时间序列ADF检验'''  
train=read\_csv('../data/PRSA\_data\_ff.csv', header=0, parse\_dates=[0], index\_col=0, squeeze=*True*)  
result = ts.adfuller(train, 1)  
print(result)

## 3.2 ARIMA模型

### 3.2.1 ARIMA构建流程图

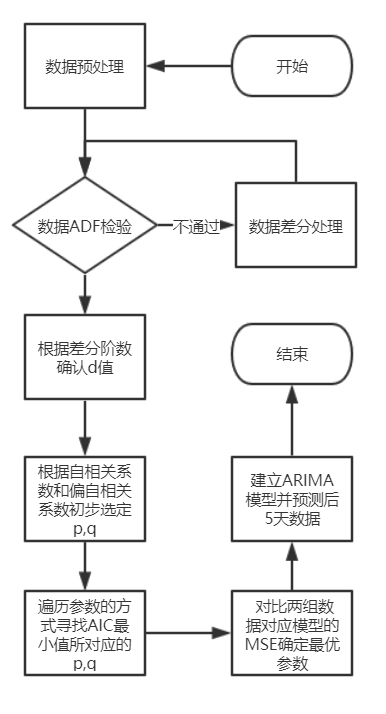


图5 模型建立流程图

### 3.2.2 参数预估及检验

|  |  |  |
| --- | --- | --- |
| 表2 ARMA模型选择 | | |
| ACF | PACF | 模型 |
| 拖尾 | P阶截尾 | AR(p) |
| Q阶截尾 | 拖尾 | MA(q) |
| 拖尾 | 拖尾 | ARMA(p,q) |

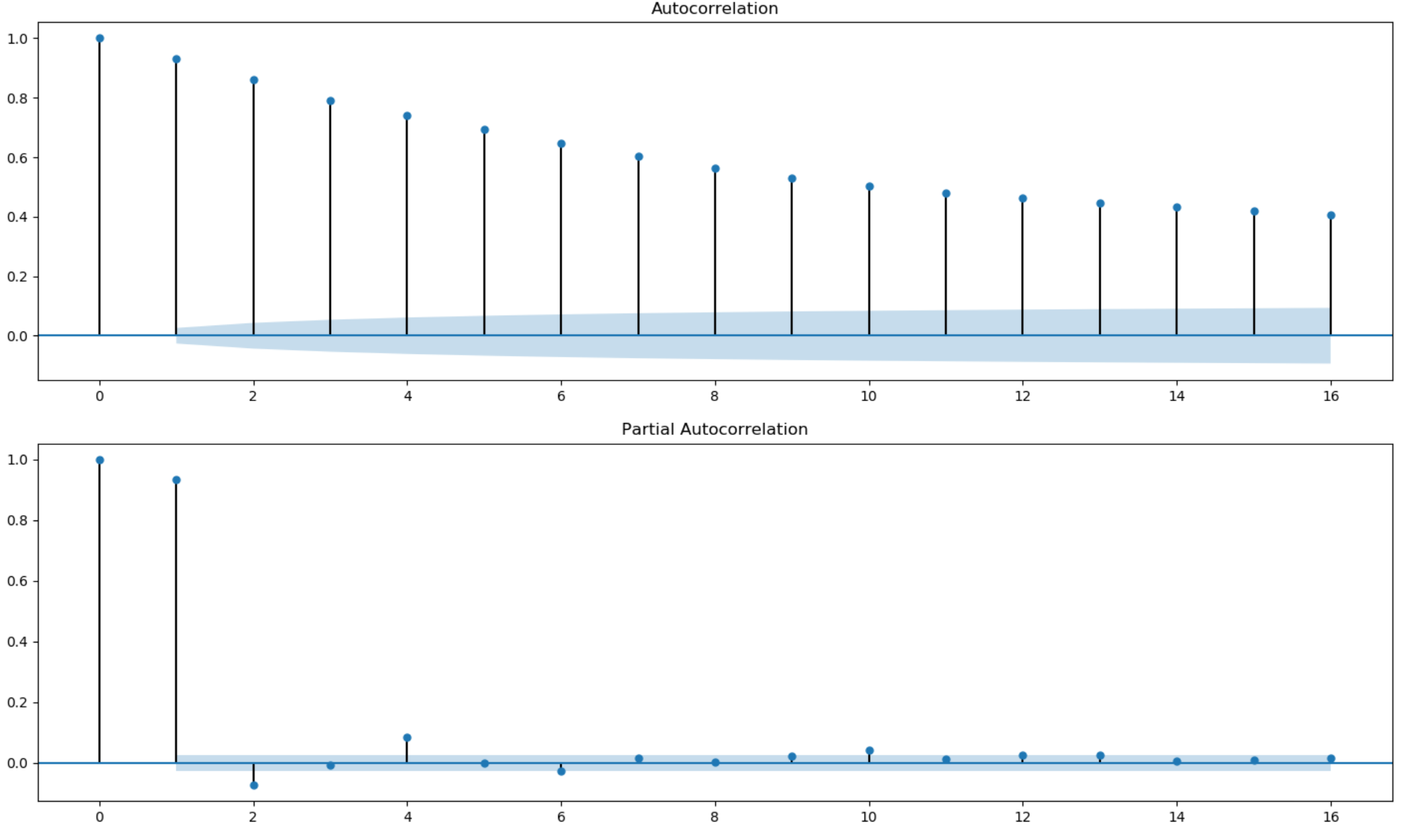


图6 时间序列的的自相关图和偏相关图

画出自相关图和偏自相关图，从图中可以看出，PAC序列1、2阶偏自相关系数超出±2倍估计标准差，2阶以后偏自相关系数在±2倍估计标准差以内，并且迅速减少至0，即偏自相关函数2阶以后截尾；同理，PAC序列超出5%样本相关系数落在±2倍估计标准差以外，即自相关函数扫尾，结合表可初步确定p=1或2，q=0。而采用AIC准则遍历AIC最小值得出p=7，q=5。

综上候选模型为ARIMA(1,0,0),ARIMA(2,0,0)，ARIMA(7,0,5)。

为检验参数预估准确性，尝试拟合候选模型ARIMA(1,0,0),ARIMA(2,0,0)，ARIMA(7,0,5)，最终根据MSE准则，计算不同p,q值组合所对应的MSE值，评价模型优劣。下表为候选模型MSE值。

|  |  |
| --- | --- |
| 表3 候选模型MSE值 | |
| ARIMA(1,0,0) | 0.080 |
| ARIMA(2,0,0) | 0.079 |
| ARIMA(7,0,5) | 0.092 |

由表可看出，当p=1，q=0时，MSE值为0.080；当p=2，q=0时，MSE值为0.079；当p=7，q=5时，MSE值为0.092。当p=2时，MSE值较小，所以确立模型为ARIMA(2,0,0 )。

**相关代码如下：**

'''通过AIC准则寻找最优参'''  
*def* findC(*series*):  
 temp = 1000000  
 ansp = 0  
 ansq = 0  
 ansd = 0  
 *for* p *in* range(0, 8):  
 *for* q *in* range(0, 8):  
 # if p+q!=0:  
 *try*:  
 testModel = ARIMA(*series*, order=(p, 0, q))  
 testModel\_fit = testModel.fit(disp=0)  
 aic = testModel\_fit.aic  
 *if* aic < temp:  
 temp = aic  
 ansp = p  
 ansq = q  
 ansd = 0  
 *except*:  
 *continue  
 return* ansp,ansd,ansq

'''比较三个候选参数'''  
series = read\_csv('../data/PRVA\_data\_ff.csv', header=0, parse\_dates=[0], index\_col=0, squeeze=*True*)  
X=preprocessing.scale(series.values)  
mse = buildArima.evaluate\_arima\_model(X,(1,0,0))  
print("p=1,d=0,q=0 mse= %.3f" %mse)  
mse = buildArima.evaluate\_arima\_model(X,(2,0,0))  
print("p=2,d=0,q=0 mse= %.3f" %mse)  
mse = buildArima.evaluate\_arima\_model(X,(7,0,5))  
print("p=6,d=0,q=0 mse= %.3f" %mse

### 3.2.3 模型检验

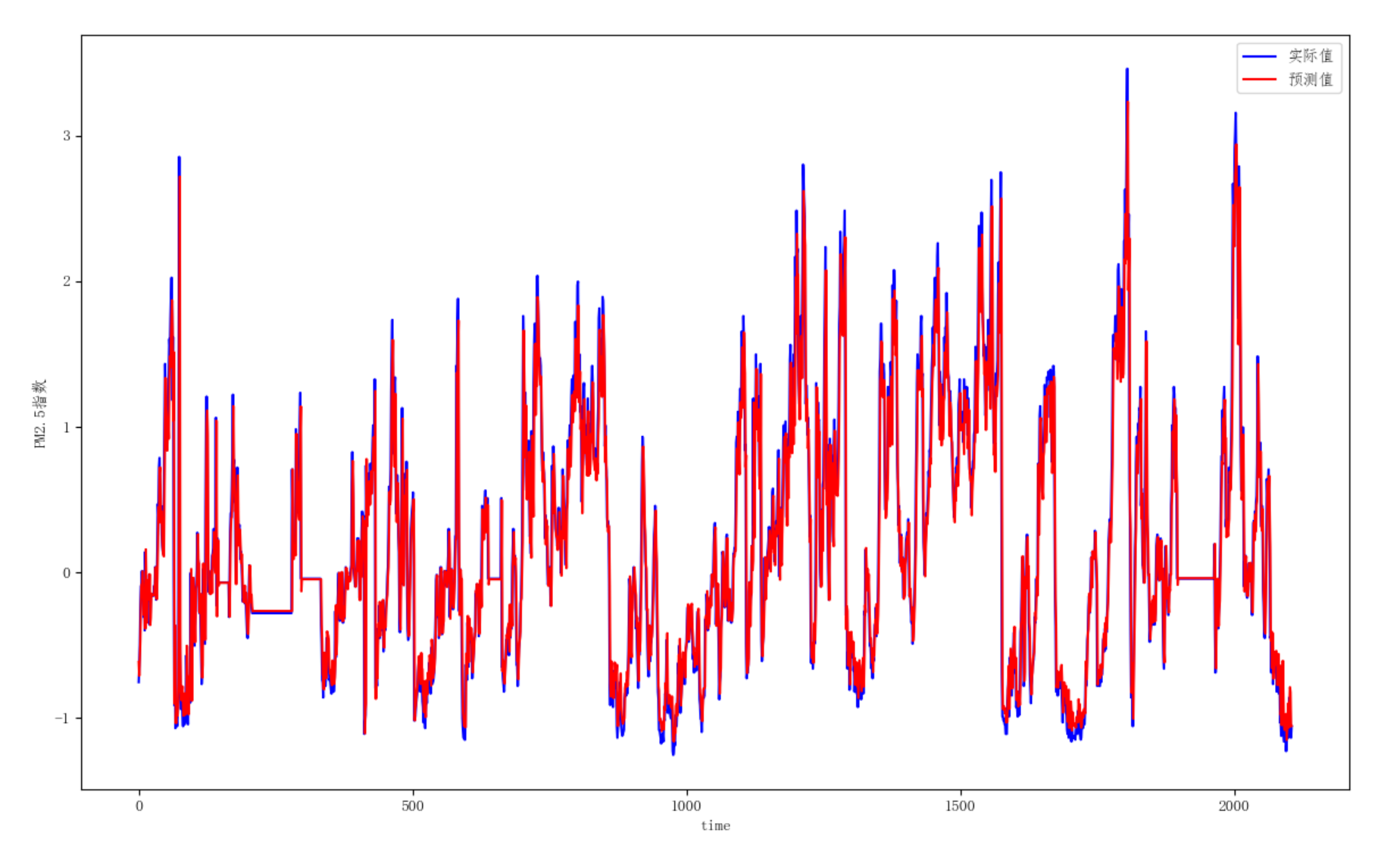
选取出最优参数之后，将模型拟合并采用测试样本前3500条数据预测后2106条数据

图7 ARIMA拟合图

拟合结果MSE = 0.079

## 3.3 SVR模型

### 3.3.1 获取时间序列数据

*def* read\_csv(*path*):  
 csv\_data = pd.read\_csv(*path*) # 读取训练数据  
 print(csv\_data.data.size)  
 data=[]  
 value = []  
 *for* i *in* range(0,csv\_data.data.size):  
 data.append(i)  
 value.append(csv\_data.value[i])  
 *return* data,value

利用pandas中的read\_csv读取数据，将时间序列存储至date，销量存储值value并返回

svr\_rbf = SVR(kernel='rbf', C=*c\_parameter*, gamma=*gamma\_paramenter*)

利用sklearn.Svm中的SVR函数构建SVR模型对象，需要三个参数分别是kernel、gamma以及C

### 3.3.2 关于核函数的选取

这其中kernel参数指定要在算法中使用的内核类型。它必须是'linear'，'poly'，'rbf'，'sigmoid'，'precomputed'或者callable之一。如果没有给出，将使用'rbf'。如果给出了callable，则它用于预先计算内核矩阵。在这里我选择了rbf核函数

### 3.3.3 C的取值

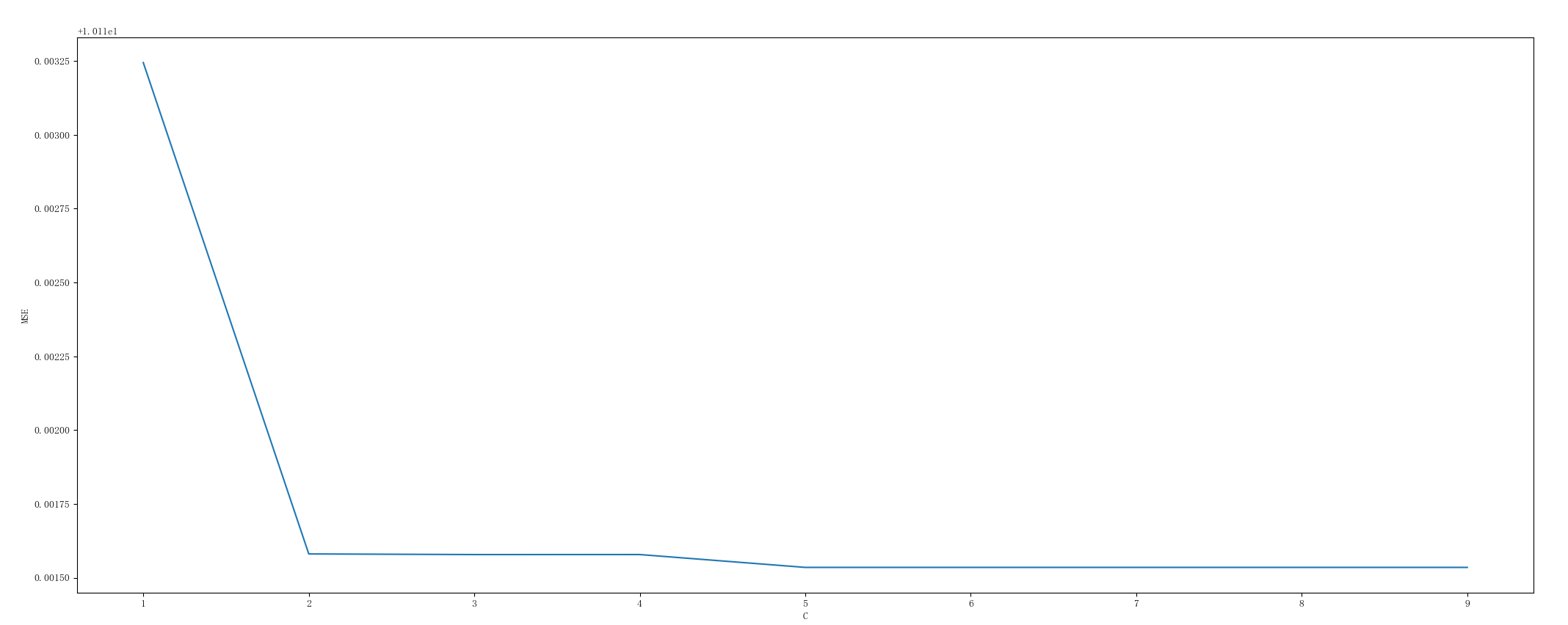
C是错误惩罚参数，这里暂且取值为1.0，c越高，说明越不能容忍出现误差,容易过拟合。C越小，容易欠拟合。C过大或过小，泛化能力变差。

下表为通过测试咋在gamma恒定为1的情况下各C取值情况下的MSE值。

表4 各C下的MSE值

|  |  |
| --- | --- |
| C | MSE |
| 1 | 10.113244672732614 |
| 2 | 10.111580874893928 |
| 3 | 10.111578700166211 |
| 4 | 10.11157878281847 |
| 5 | 10.111535301021904 |
| 6 | 10.11153530028908 |
| 7 | 10.111535300287187 |
| 8 | 10.111535300287148 |
| 9 | 10.111535300287148 |

**下图为可视化结果**



**图8 可视化结果**

可以看到在gamma恒定为1的情况下，随着惩罚指数变大的情况下，MSE值由开始迅速下降后趋于平稳。

此时将gamma 恒定为10，再测试一遍

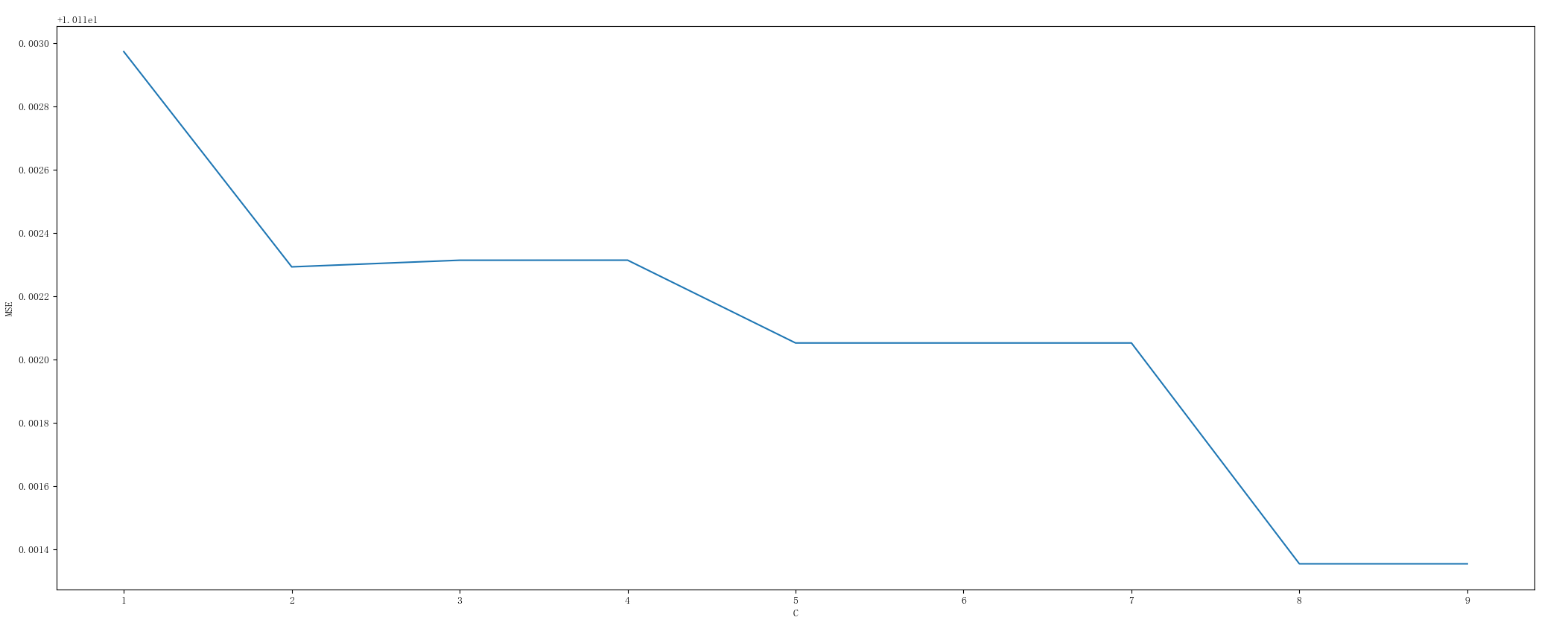


图9 可视化结果

可以看到随着C的增大MSE值在逐渐减少，最后趋于平稳，

**可以得出结论：**惩罚系数C不可取的过小，过小会导致模型无法正常拟合，也不可取得过大。且MSE值随着C的增大而下降

### 3.3.4 Gamma取值

Gamma是选择RBF函数作为kernel核函数后，gamma为该函数自带的一个参数。隐含地决定了数据映射到新的特征空间后的分布，gamma越大，支持向量越少，gamma值越小，支持向量越多。支持向量的个数影响训练与预测的速度。

下表为通过测试咋在C恒定为1的情况下各Gamma取值情况下的MSE值

表4 各Gamma下的MSE值

|  |  |
| --- | --- |
| Gamma | MSE |
| 1 | 10.113244672732614 |
| 2 | 10.1105363229753 |
| 3 | 10.110972572903398 |
| 4 | 10.111644376066465 |
| 5 | 10.114195377422826 |
| 6 | 10.111597573787769 |
| 7 | 10.111918132996959 |
| 8 | 10.111730007586536 |
| 9 | 10.111617677757136 |

**将表中数据可视化结果如下**

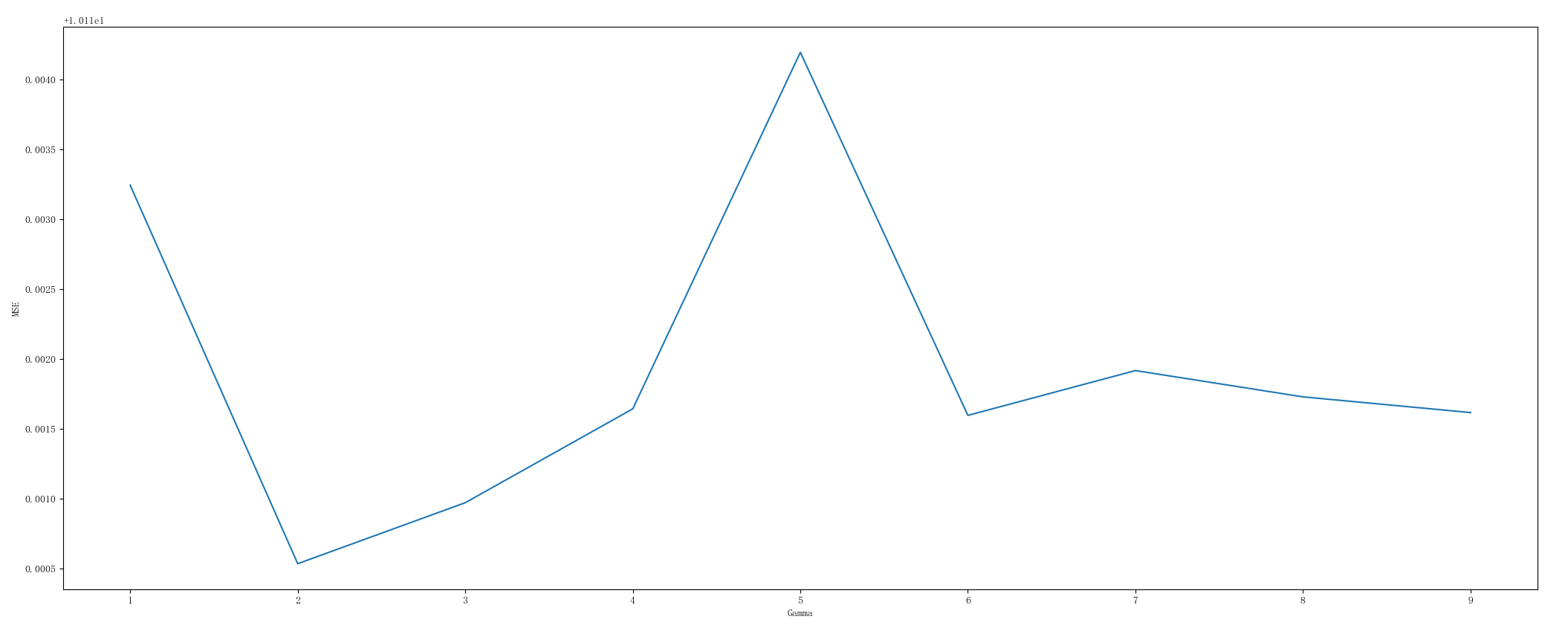


图10 可视化结果

若此时将C的取值增大至10

**可视化结果如下**

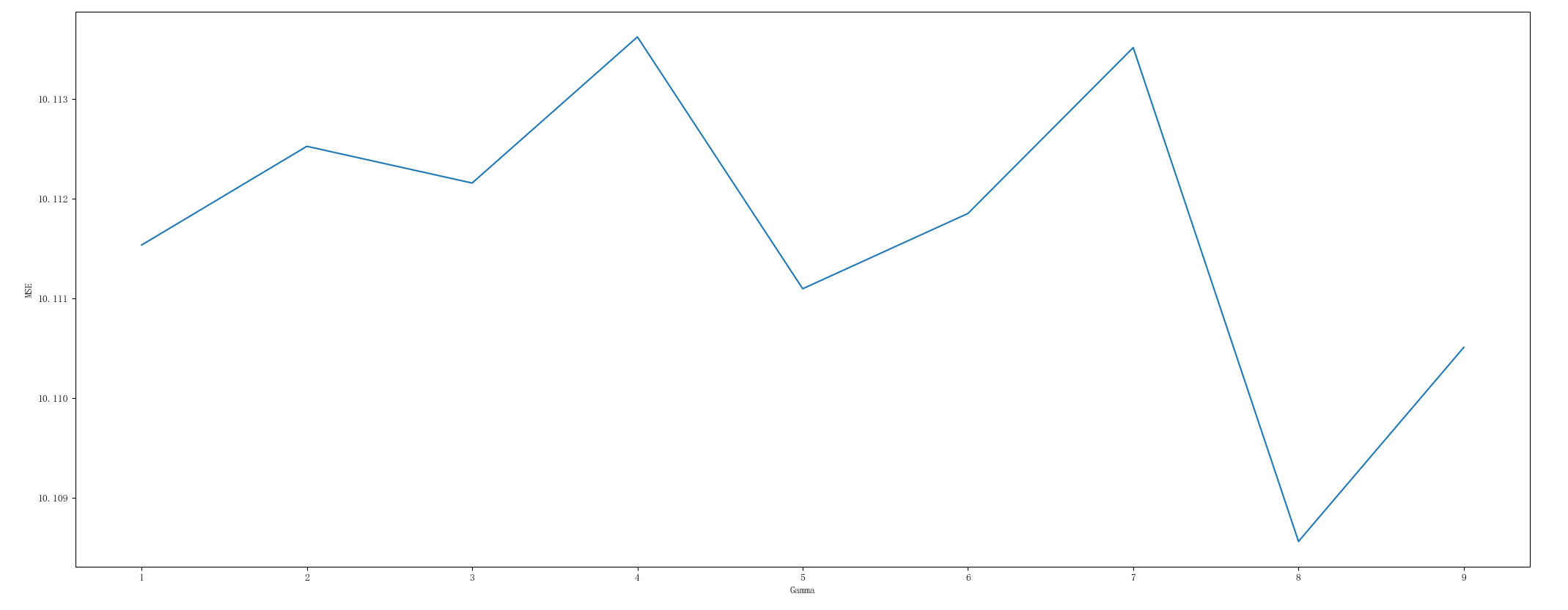


图11 可视化结果

C增大至100

**可视化结果如下**

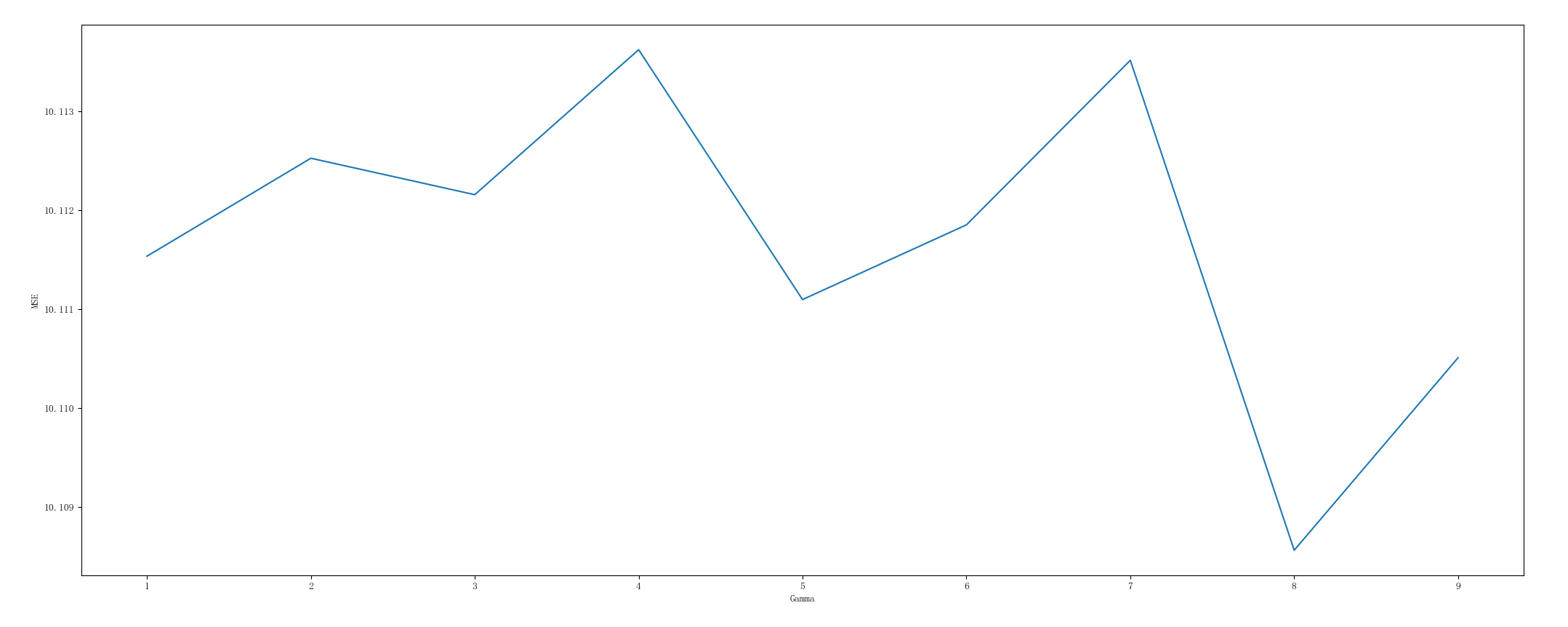


图12 可视化结果

可以看到图中MSE值随着gamma取值的增大而增大后又减小。呈周期性起伏。且不随着C的增大而发生规律变化。

**可以得出结论：**预测结果的准确度随着gamma取值的增大而均匀起伏，且gamma对结果的影响不随着C的改变而改变。这将导致gamma的取值变得异常困难。

此部分代码如下：

*def* testC(*gamma*):  
 print("when Gamma=%d "%*gamma*)  
 cs= []  
 mses = []  
 *for* c *in* range(1,10):  
 X\_data, Y\_data, X\_prediction, y\_prediction, error, mse = sv.svm\_timeseries\_prediction(data, value, *gamma*, c)  
 print("C= %.3f" %c)  
 cs.append(c)  
 print("mse = %.3f" %mse)  
 mses.append(mse)  
 plt.plot(cs,mses)  
 plt.axis([0,9,10,30])  
 plt.xlabel('C')  
 plt.ylabel('MSE')  
 plt.show()  
  
*def* testGamma(*c*):  
 print("when c=%d "%*c*)  
 gammas = []  
 mses = []  
 *for* gamma *in* range(1,10):  
 X\_data, Y\_data, X\_prediction, y\_prediction, error, mse = sv.svm\_timeseries\_prediction(data, value, gamma, *c*)  
 print("Gamma= %d" %gamma)  
 print("mse = %.3f" %mse)  
 gammas.append(gamma)  
 mses.append(mse)  
 plt.plot(gammas, mses)  
 plt.axis([0, 9, 0, 30])  
 plt.xlabel('Gamma')  
 plt.ylabel('MSE')  
 plt.show()

### 3.3.5 选取最优的C与gamma

经过上述步骤初步探究了C与gamma取值对于结果的影响，这一步就要选择最优的参数组合了。由于gamma与结果优劣的不确定性，我们通过遍历比较MSE的方式可以得出最佳的参数组合

**此部分代码如下：**

data,value = read\_csv("../data/PRSA\_data\_ff.csv")  
  
value=preprocessing.scale(value)  
temp\_mse = 10000 #mse初始值 默认无限大  
c\_weight = range(90,100) #c的取值范围  
gamma\_weight = range(90,100) #gamma的取值范围  
  
*for* c\_paramenter *in* c\_weight:  
 *for* gamma\_paramenter *in* gamma\_weight:  
 X\_data,Y\_data,X\_prediction,y\_prediction,error,mse = svm\_timeseries\_prediction(data,value,c\_paramenter,gamma\_paramenter)  
 *if*(mse<temp\_mse):  
 temp\_mse = mse  
 temp\_c = c\_paramenter  
 temp\_gamma = gamma\_paramenter

最终的出的最优参数为C=99，gamma = 99，此参数组合得出的MSE值为10.111

### 3.3.6 模型**拟合**

采用前30份数据进行训练后对后40份数据进行预测，并求出拟合值。

可视化结果如下

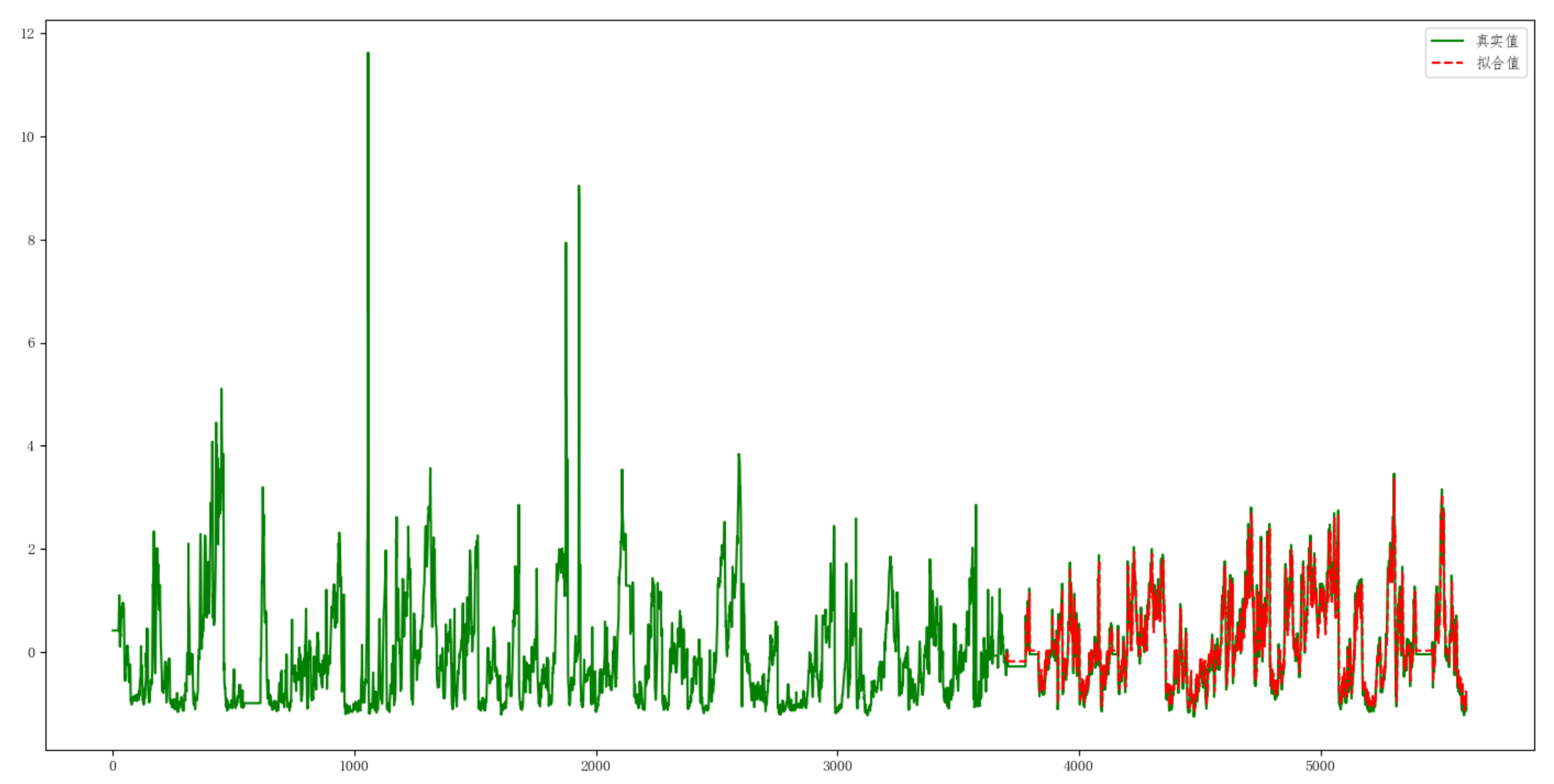


图13 SVR拟合结果

MSE为0.010，比最优参数的ARIMA模型低

# 结论

在同一数据，且进行平稳性检验的前提下SVR模型得出的拟合值为10.111，ARIMA模型得出的拟合值为0.079，综合得出结论：本案例中，在相同平稳时间序列的情况下，经过同同样的归一化处理后ARIMA模型的拟合值误差更小。也就是说ARIMA模型对于规模较大的时间序列数据，有更优的预测效果。

# 参考文献

1. sklearn官方API文档

<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>

1. UCI数据集

[http://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data#](http://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data)

1. ARIMA模型原理

<https://blog.csdn.net/qq_35495233/article/details/83514126>

1. SVR简明教程

<https://zhuanlan.zhihu.com/p/38896196>