集美大学

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
一、 背景 4
二、 主要技术介绍 5
2.1 MSE 均方误差 5
2.2 ADF 单位根检验 5
2.3 ACF、PACF 自相关系数与偏自相关系数 5
2.4 AIC 准则 5
2.5 ARIMA 模型 6
2.5.1 自回归模型 AR 6
2.5.2 移动平均模型 MA 6
2.5.3 自回归移动平均模型 ARMA 6
2.6 SVR 模型 7
2.6.1 SVR 模型概览 7
2.6.2 核函数 7
三、 案例分析 9
3.1 数据集 9
3.1.1 数据预处理 10
3.1.2 平稳性评估及平稳化处理 10
3.2 ARIMA 模型 11
3.2.1 ARIMA 构建流程图 11
3.2.2 参数预估及检验 11
3.2.3 模型检验 14
3.3 SVR 模型 14
3.3.1 获取时间序列数据 14
3.3.2 关于核函数的选取 15
3.3.3 C 的取值 15
3.3.4 Gamma 取值 16
3.3.5 选取最优的 C 与 gamma 18
3.3.6 模型拟合 19
四、 结论 19
五、 参考文献 20

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

一、 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。

二、 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：

$$\frac{1}{m}\sum_{i=1}^{m}(y_i-\hat{y}_i)^2$$

公式 2.1.1

其中，y_i是预测值，$\hat{y}_i$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，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。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：

$$\mathrm{ACF(k)}=\rho_k=\frac{Cov(y_t,y_{t-k})}{Var(y_t)}$$

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

$$\mathrm{AIC}=2\mathrm{k}-2\ln(\mathrm{L})$$

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：

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-i} + \epsilon_t$$

公式 2.5.1.1

上式中 y_t 是当前值，u 是常数项，p 是阶数 γ_i 是自相关系数，E_t 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：

$$y_t = \mu + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$$

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

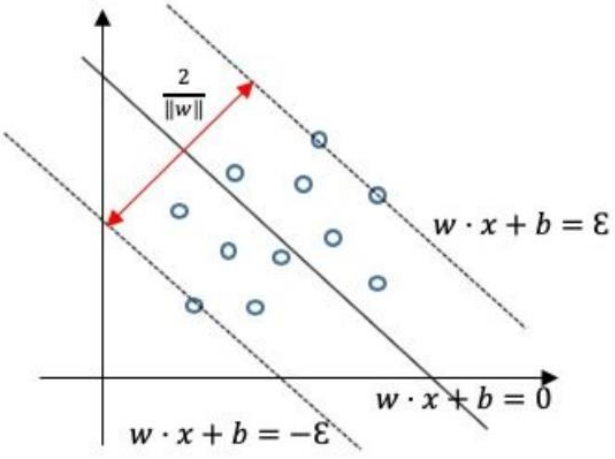
自回归模型 AR 和移动平均模型 MA 模型相结合，我们就得到了自回归移动平均模型 ARMA(p,q)，计算公式如下：

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{i-1} + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$$

公式 2.5.3.1

2.6　SVR 模型

2.6.1 SVR 模型概览



使得到超平面最远的样本点的距离最小

图 1　SVR 概要

SVR 问题可以形式化为

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m} l_\epsilon(f(x_i) - y_i)$$

公式 2.6.1.1

其中$\min_{w,b} \frac{1}{2}||w||^2$为超平面的最小距离，C 为正则化常数

l_ϵ为 ε-不敏感损失 函数如下

$$l_\epsilon(z) = \begin{cases} 0, & if\,|z| \le \epsilon \\ |z| - \epsilon, & otherwise \end{cases}$$

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：

$$\mathrm{k}(\mathrm{x}_i, \mathrm{x}_j) = \exp\left(-\frac{\left||x_i - x_j|\right|^2}{2\sigma^2}\right) \qquad \sigma > 0$$

公式 2.6.2.1

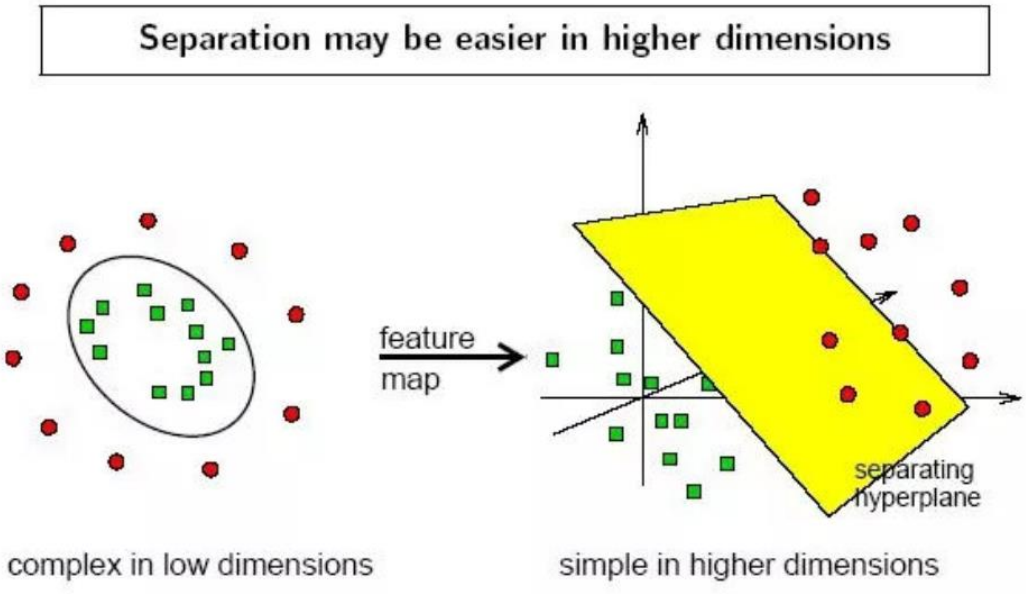


图 2 核函数映射说明

三、 案例分析

3.1 数据集

data	value
1	129
2	129
3	129
4	129
5	129
6	129
7	129
8	129
9	129
10	129
11	129
12	129
13	129
14	129
15	129
16	129
17	129
18	129
19	129
20	129
21	129
22	129
23	129
24	129
25	129
26	148
27	159
28	181
29	138
30	109

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

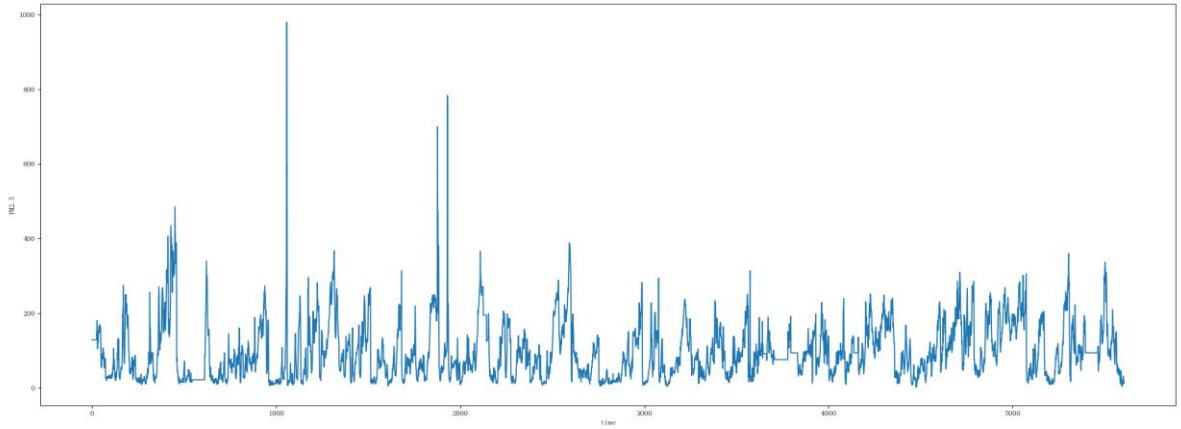


图 4 2010 年 1 月至 2011 年 7 月每小时 PM2.5 指数时序图

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$\{x_t\}$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

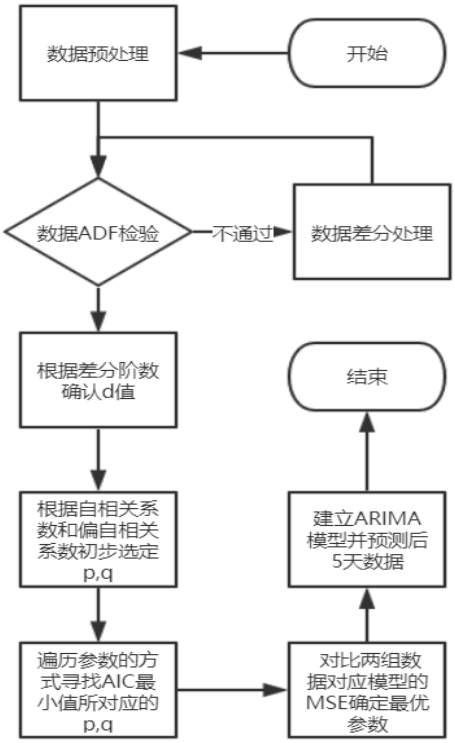


图 5 模型建立流程图

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)

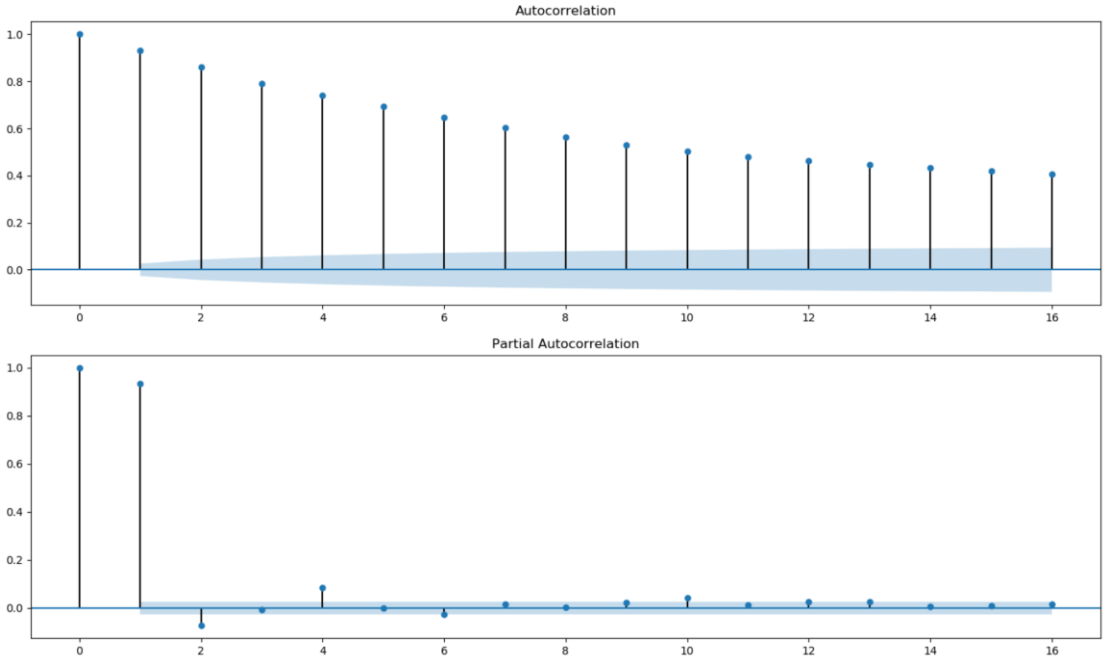


图 6 时间序列的的自相关图和偏相关图

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

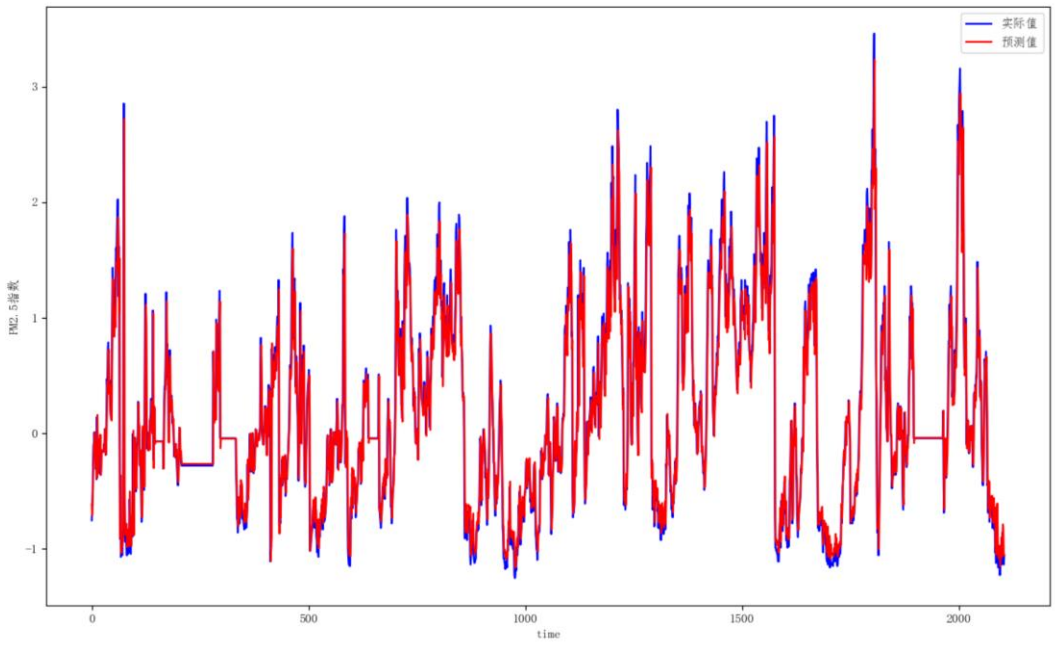


图 7 ARIMA 拟合图

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

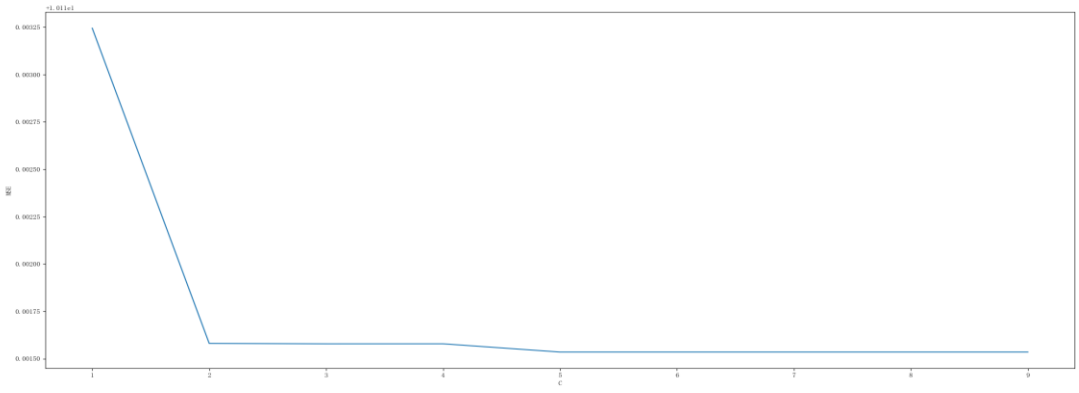


图 8 可视化结果

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

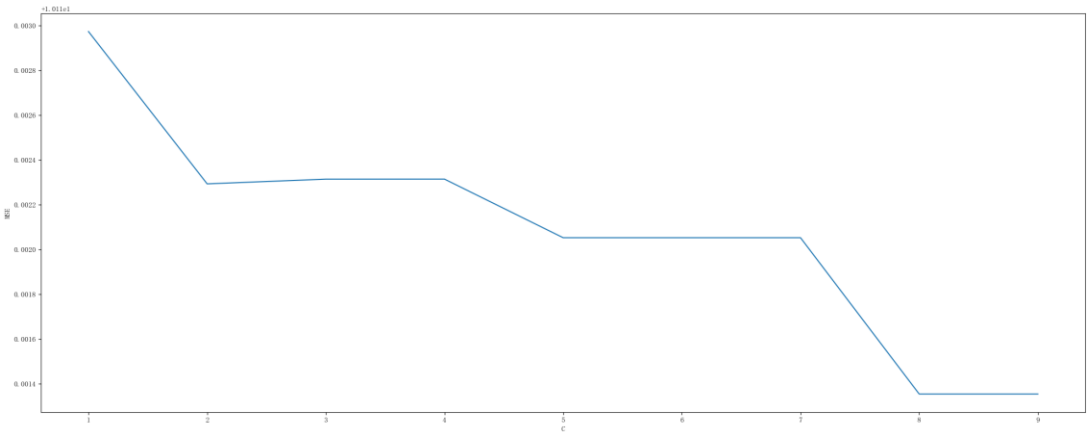


图 9 可视化结果
可以看到随着 C 的增大 MSE 值在逐渐减少，最后趋于平稳，
可以得出结论：惩罚系数 C 不可取的过小，过小会导致模型无法正常拟合，也不可取得过大。且 MSE 值随着 C 的增大而下降

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

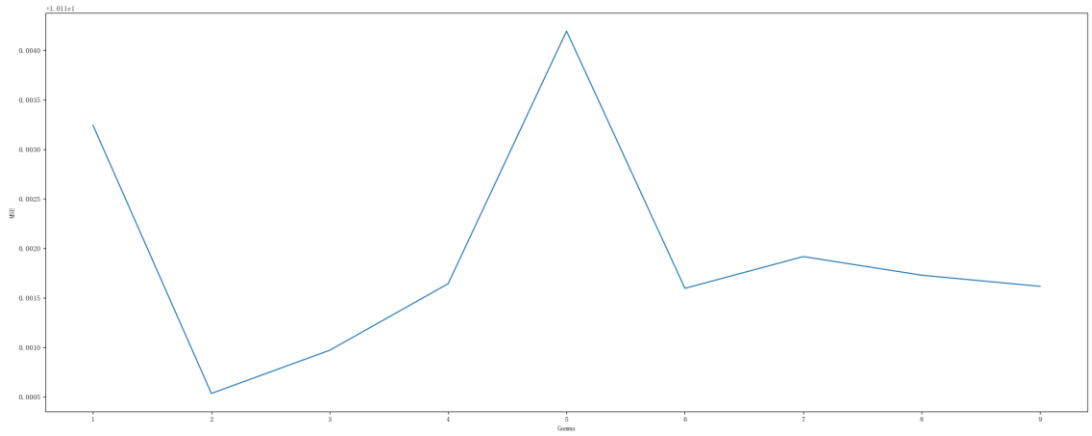


图 10 可视化结果

若此时将 C 的取值增大至 10

可视化结果如下

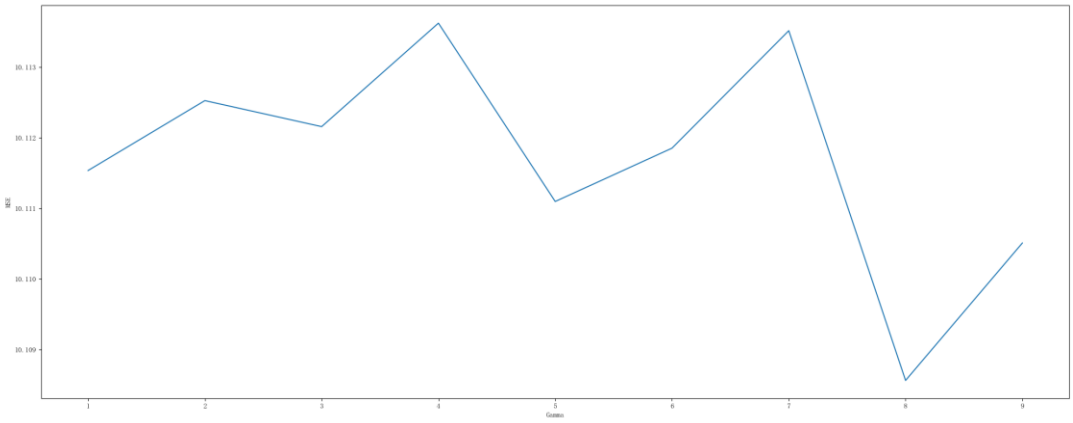


图 11 可视化结果

C 增大至 100

可视化结果如下

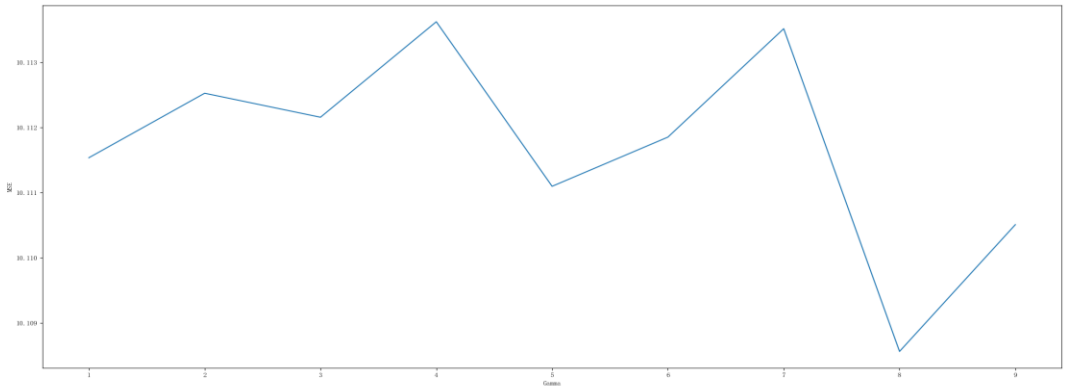


图 12 可视化结果

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_paramente
r)
        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

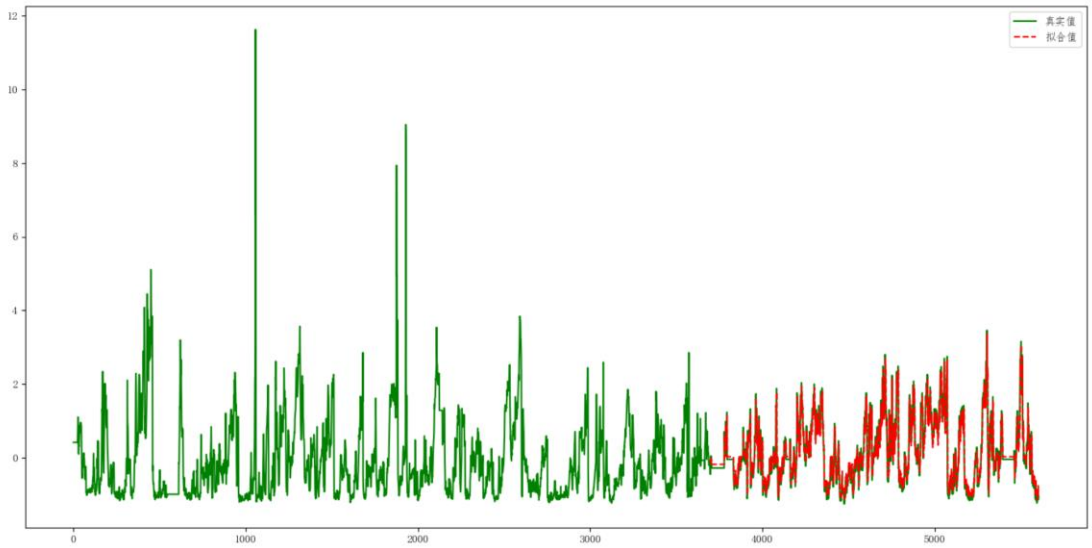


图 13 SVR 拟合结果

MSE 为 0.010，比最优参数的 ARIMA 模型低

四、 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。

五、 参考文献

【1】 sklearn 官方 API 文档
https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

【2】 UCI 数据集
http://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data#

【3】 ARIMA 模型原理
https://blog.csdn.net/qq_35495233/article/details/83514126

【4】 SVR 简明教程
https://zhuanlan.zhihu.com/p/38896196