高级量化交易技术

闫涛
科技有限公司
北京
{yt7589}@qq.com

第 1 章　时间序列基本特性

Abstract

在本章中我们将讨论时间序列的基本特性，包括自相关性和平稳性。

1　自相关性

时间序列的自相关性是指时间序列过去与未来存在某种关系，是我们时间序列预测的基础。主要用自协方差函数（Autocovariance Function, AF）、自相关系数函数（Autocorrelation Coefficent Function, ACF）和偏自相关系数函数（Partial Autocorrelation Coefficient Function, PACF）来描述。

1.1　随机变量统计量

随机变量 X 其取值为 x 的均值定义为：

$$E(x) = \mu \tag{1}$$

方差定义为：

$$\sigma^2(x) = E\left[\left(x-\mu\right)^2\right] \tag{2}$$

其中 $\sigma(x)$ 为标准差。对于两个随机变量 x 和 y，其协方差可以定义为：

$$\sigma(x,y) = E\left[\left(x-\mu_x\right)\left(y-\mu_y\right)\right] \tag{3}$$

在实际应用中，我们不可能知道真实的均值，只能使用统计量，因此协方差可以定义为：

$$Cov(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y}) \tag{4}$$

随机变量 x 和 y 的相关系数定义为：

$$\rho(x,y) = \frac{E[(x-\mu_x)(y-\mu_y)]}{\sigma_x\sigma_y} = \frac{\sigma(x,y)}{\sigma_x\sigma_y} \tag{5}$$

采用统计量的表示方法为：

$$Cor(x,y) = \frac{Cov(x,y)}{std(x)\times std(y)} \tag{6}$$

以上我们讨论的都是不同随机变量之间的关系，对于时间序列来说，我们可以把从不同时间点开始的子时间序列，视为不同的随机变量，那么我们就可以定义自协方差、自相关系数函数和偏自相关系数函数了。

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

1.2 时序序列平稳性

时序信号 x_t 的均值定义为：

$$E(x_t) = \mu(t) \tag{7}$$

时间序列的均值与所考虑的时间点有关。我们可以把时间序列上每个时间点都视为一个独立的时间变量，但是对于时间序列而言，每个时间点的随机变量只有一个观测值，怎么求出均值呢？在实际应用中，我们会将时间序列中的趋势信号（上涨或下跌）、季节性信号等从时间序列中去除掉，对于剩下的残差序列，我们可以视为其各个时间点上的随机变量的均值是不变的，于是就可以使用下面的公式来计算均值：

$$\bar{x} = \frac{1}{N}\sum_{t=1}^{N} x_t \tag{8}$$

由此我们引入平稳时间序列的概念，对于平稳时间序列，其各个时间点上对应的随机变量的均值相等。时间序列的方差可以定义为：

$$\sigma^2(t) = E[(x_t - \mu_t)^2] \tag{9}$$

根据上面平稳时间序列的定义，各个时间点对应的随机变量的均值不变，则式9可以化间为：

$$\sigma^2(t) = E[(x_t - \mu)^2] \tag{10}$$

我们同时规定，平稳时间序列各个时间点对应的随机变量的方差也不变，则10可进一步化简为：

$$Var(x_t) = \frac{1}{N-1}\sum_{t=1} N(x_t - \bar{x})^2 \tag{11}$$

1.3 自协方差

在讨论自协方差之前，我们首先要定义二阶平稳性。根据上一节定义，平稳时间序列是指各个时间点对应的随机变量的均值和方差相同。二阶平稳性是指在这一基础上，不同时间点对应的随机变量的相关系数只与时间相隔（lag）相关。注意：以下我们讨论的各种性质，均以此为前提。对于 lag=k 的自协方差定义为：

$$C_k = E[(x_t - \mu)(x_{t+k} - \mu)] \tag{12}$$

1.4 自相关系数函数 ACF

由于自协方差的大小与随机变量的大小有关，无法准确衡量其间的关系，因此我们引入自相关系数函数 ACF：

$$\rho_k = \frac{C_k}{\sigma^2} \tag{13}$$

由定义可知：

$$\begin{aligned} \rho_0 = \frac{C_0}{\sigma^2} &= \frac{E[(x_t - \mu)(x_t - \mu)]}{\sigma^2} \\ &= \frac{E[(x_t - \mu)^2]}{\sigma^2} = \frac{\sigma^2}{\sigma^2} = 1 \end{aligned} \tag{14}$$

在实际应用中，我们都是处理的离散数据点，则自协方差可以定义为：

$$c_k = \frac{1}{N}\sum_{t=1}^{N}(x_t - \bar{x})(x_{t+k} - \bar{x}) \tag{15}$$

自相关系数函数 ACF 可以定义为：

$$r_k = \frac{c_k}{c_0} \tag{16}$$

1.5 自相关性举例

下面我们以上证综指时间序列为例，来看自相关系数函数 ACF 和偏自相关系数函数 PACF 的求法和作图，程序代码如下所示：

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from matplotlib.font_manager import FontProperties
from statsmodels.tsa import stattools
from statsmodels.graphics import tsaplots

class Chp023(object):
    def __init__(self):
        self.name = 'Chp022'
        # 数据文件格式: 编号 日期 星期几 开盘价 最高价
        # 最低价 收益价 收益
        self.data_file = 'data/pqb/chp023_001.txt'

    def startup(self):
        print('第23章: 时间序列基本性质')
        data = pd.read_csv(self.data_file, sep='\t', index_col='Trddt')
        sh_index = data[data.Indexcd==1]
        sh_index.index = pd.to_datetime(sh_index.index)
        sh_return = sh_index.Retindex
        print('时间序列长为: N={0}'.format(len(sh_return)))
        acfs = stattools.acf(sh_return)
        print(acfs)
        pacfs = stattools.pacf(sh_return)
        print(pacfs)
        tsaplots.plot_acf(sh_return, use_vlines=True, lags=30)
        plt.show()
        tsaplots.plot_pacf(sh_return, use_vlines=True, lags=30)
        plt.show()
```

Listing 1: 时间序列基本性质

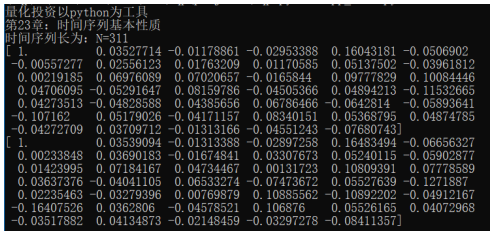
数据文件格式为：

Figure 1: 数据文件格式

Indexcd	Trddt	Daywk	Opnindex	Hiindex	Loindex	Clsindex	Retindex
1	2014/1/2	4	2112.126	2113.11	2101.016	2109.387	-0.003115
1	2014/1/3	5	2101.542	2102.167	2075.899	2083.136	-0.012445
1	2014/1/6	1	2078.684	2078.684	2034.006	2045.709	-0.017967
1	2014/1/7	2	2034.224	2052.279	2029.246	2047.317	0.000786
1	2014/1/8	3	2047.256	2062.952	2037.11	2044.34	-0.001454
1	2014/1/9	4	2041.773	2057.196	2026.446	2027.622	-0.008178
1	2014/1/10	5	2023.535	2029.297	2008.007	2013.298	-0.007064
1	2014/1/13	1	2014.978	2027.181	2000.404	2009.564	-0.001855

其运行结果为：

Figure 2: 运行结果



在图2中，第一个数据为自相关系数函数 ACF 各期的值，而第二个数组为偏自相关系数函数 PACF 各期的值。判断时间序列是否具有自相关性，可以看除 ACF 和 PACF 中，除第一个元素外，有没有显著超过阈值的元素，阈值定义为：

$$threshold = \frac{1.96}{\sqrt{N}} = \frac{1.96}{\sqrt{311}} = 0.11 \tag{17}$$

式17中的 N 为时间序列样本数，在本例中，共有 311 条记录，故 N=311，所以其阈值为 0.11 左右。由于 $acf[4] = 0.16 > 0.11$ 所以可以推断其具有自相关性，同时 $pacf[4] = 0.165 > 0.11$ 也可以推断其具有自相关性。我们还可以通过图形的方式形像的表示出来，自相关系数函数图如所示：

Figure 3: 自相关系数函数图

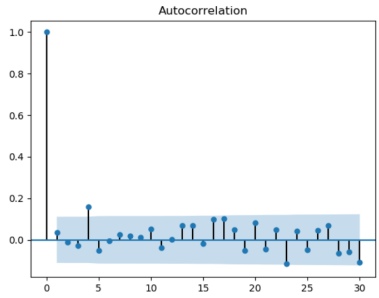
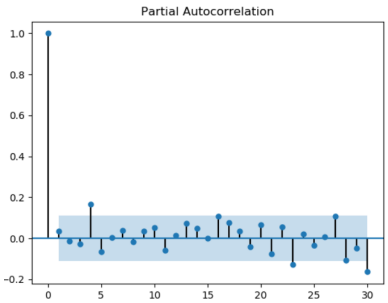


图3中蓝色区域的界限为 $[-\frac{1.96}{\sqrt{N}}, \frac{1.96}{\sqrt{N}}] = [-0.11, 0.11]$，除第 1 项外，其他项如果超出蓝色区域则说明此时间序列具有自相关性。
偏自相关系数函数图为：

Figure 4: 偏自相关系数函数图



1.6 白噪声和随机游走

1.6.1 残差序列定义

我们要对任意时间序列 y_t 进行建模，我们的模型为 $\hat{y}_t$，残差序列 x_t 可以定义为：$x_t = y_t - \hat{y}_t$，我们的任务就是使残差时间序列中每一个时间点对应的随机变量互相独立，没有自相关性，即满足独立同分布（Independent and Identical Distribution,I.I.D）条件。如果各个随机变量 $x_t \sim \mathbb{N}(0, \sigma^2)$，则称其为高斯白噪声。

1.6.2 差分运算符

为了后续讨论问题方便，我们首先定义 BSO 运算符：

$$Bx_t = x_{t-1} \quad B^n x_t = x_{t-n} \tag{18}$$

我们定义差分运算符为：

$$\begin{aligned} \nabla x_t &= x_t - x_{t-1} = (1 - B)x_t \\ \nabla x_t^n &= \left(x_t - x_{t-n}\right)^n = (1 - B)^n x_t \end{aligned} \tag{19}$$

1.6.3 白噪声定义

对于时间序列 $\{w_t, t = 1, 2, 3, ..., N\}$，满足 $\forall t \quad w_t \sim \mathcal{N}(0, \sigma^2)$ 且 $\forall i \neq j \quad Cor(w_i, w_j) = 0$，则其为白噪声时间序列。
下面我们来看白噪声的二阶属性：

$$\begin{aligned} \mu &= E(w_t) = 0 \\ \gamma_k = Cor(w_t, w_{t+k}) &= \begin{cases} 1 & if \quad k = 0 \\ 0 & if \quad k \neq 0 \end{cases} \end{aligned} \tag{20}$$

1.6.4 随机游走

随机游走（Random Walk）时间序列是指 x_t 可以定义为：$x_t = x_{t-1} + w_t$，其中 w_t 为白噪声时间序列。随机游走时间序列可以表示为：

$$\begin{aligned} x_t = x_{t-1} + w_t &= Bx_t + w_t \\ x_t = x_{t-1} + w_t &= x_{t-2} + w_{t-1} + w_t \\ &..... \\ x_t &= w_1 + w_2 + ... + w_{t-1} + w_t \end{aligned} \tag{21}$$

所以随机游走时间序列可以看作是多个白噪声时间序列的叠加。下面我们来看随机游走时间序列的均值和协方差：

$$\begin{aligned} \mu &= 0 \\ \gamma_k(t) = Cov(x_t, x_{t+k}) &= t\sigma^2 \end{aligned} \tag{22}$$

我们再来看随机游走序列的自相关系数函数 ACF：

$$\begin{aligned} \rho_k(t) &= \frac{Cov(x_t, x_{t+k})}{\sqrt{Var(x_t) \cdot Var(x_{t+k})}} \\ &= \frac{t\sigma^2}{\sqrt{t\sigma^2(t+k)\sigma^2}} = \frac{1}{\sqrt{1 + \frac{k}{t}}} \end{aligned} \tag{23}$$

在通常情况下，t 比 k 要大得多，因此 ρ_k 会比较接近于 1。
下面我们来模拟一个随机游走时间序列信号，程序如下所示：

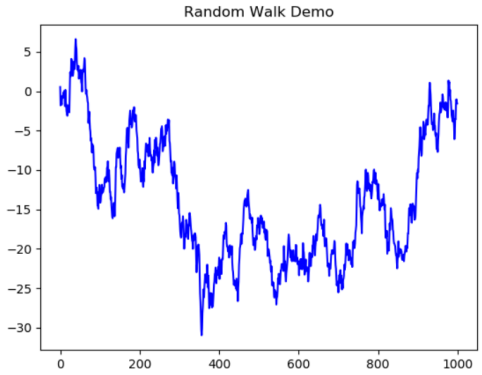
```
    def random_wale_demo(self):
        '''
        随机游走时间序列建模示例
        '''
        w = np.random.standard_normal(size=1000)
        x = w
        for t in range(1, len(w)):
            x[t] = x[t-1] + w[t]
        plt.plot(x, c='b')
        plt.title('Random Walk Demo')
        plt.show()
        acfs = stattools.acf(x)
        print(acfs)
        tsaplots.plot_acf(x, use_vlines=True, lags=30)
        plt.show()
        # 拟合随机游走信号
        r = []
        for t in range(1, len(x)):
            r.append(x[t] - x[t-1])
        rd = np.array(r)
        plt.plot(rd, c='r')
        plt.title('Residue Signal')
        plt.show()
        rd_acfs = stattools.acf(rd)
        print(rd_acfs)
        tsaplots.plot_acf(rd, use_vlines=True, lags=30)
```

```
27          plt.show()
```

Listing 2: 随机游走过程模拟

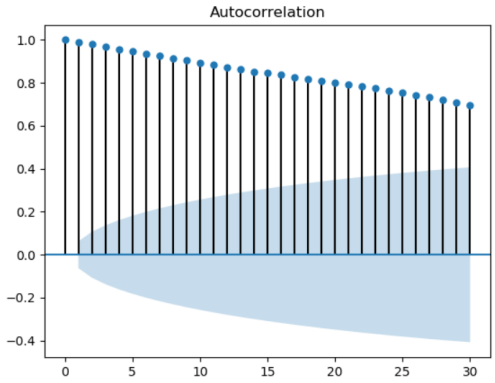
我们首先通过 $x_t = x_{t-1} + w_t$ 生成一个随机游走信号，该信号图形如下所示：

Figure 5: 随机游走时间序列信号



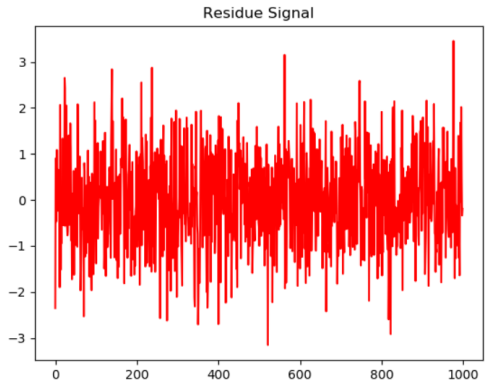
由图5可以看出，其非常像是一个股票收盘价的走势图，这也是为什么有些人说股票走势是随机游走过程了。接着我们求出该时间序列的自相关系数函数 ACF 及其自相关图，如下所示：

Figure 6: 随机游走时间序列信号



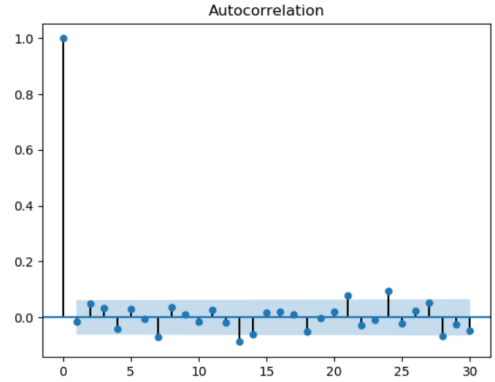
由图可以看出，其具有极强的自相关性，所有 ACF 值均位于蓝色置信区间之外。
我们知道 $x_t - x_{t-1} = w_t$，而 w_t 是白噪声时间序列信号，这实际上模拟了实际应用过程，我们把 x_t 视为实际的金融信号，而 x_{t-1} 为我们建模的信号，将两个信号相减，得到残差信号，如果残差信号是白噪声信号，就可以认为我们建模是合理的。下面来看我们得到的残差信号：

Figure 7: 残差时间序列信号



计算并绘制 ACF 如下所示：

Figure 8: 残差自相关系数函数



程序的运行结果如下所示：

Figure 9: 程序运行结果

```
量化投资以python为工具
第23章：时间序列基本性质
[1.          0.98914742 0.97903129 0.96783309 0.95632931 0.94596796
 0.93489265 0.92427096 0.91497217 0.90461824 0.89396843 0.88323481
 0.87235068 0.86216894 0.85285975 0.84479971 0.83634853 0.82763235
 0.81879369 0.81058584 0.80226346 0.79341307 0.78342443 0.77367788
 0.76487486 0.75417161 0.74313034 0.73174875 0.71939583 0.7078342
 0.6965237  0.6861672  0.6760541  0.66486422 0.65380632 0.64342133
 0.63339216 0.62341977 0.6128232  0.60296489 0.59324119]
[ 1.00000000e+00 -1.48521639e-02  5.06985700e-02  3.21625338e-02
 -4.24342295e-02  2.86756161e-02 -4.41871354e-03 -7.18986605e-02
  3.66200199e-02  1.21850369e-02 -1.59723171e-02  2.68915024e-02
 -1.94434230e-02 -8.78575550e-02 -5.94851957e-02  1.62981893e-02
  1.93119858e-02  8.95837565e-03 -4.99698205e-02 -3.38587649e-03
  2.00536805e-02  7.92334588e-02 -2.79786314e-02 -8.84736489e-03
  9.60908594e-02 -2.21958482e-02  2.21536791e-02  5.16706070e-02
 -6.80076985e-02 -2.63520397e-02 -4.69741376e-02  7.80032912e-04
  7.86023537e-02 -2.33883445e-02 -5.47155569e-02 -2.65466308e-02
 -3.87128530e-03  2.09699285e-02 -5.18812130e-02 -2.84822052e-02
  1.58564831e-02]
```

下面我们以上证综指收益率为例，来看随机游走模型是否可以很好的拟合这个时间序列，程序如下所示：

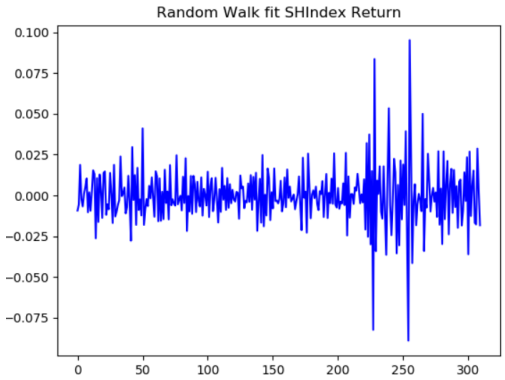
```
    def random_walk_fit(self):
        data = pd.read_csv(self.data_file, sep='\t', index_col='Trddt')
        sh_index = data[data.Indexcd==1]
        sh_index.index = pd.to_datetime(sh_index.index)
```

```
        sh_return = sh_index.Retindex
        print('时间序列长为: N={0}'.format(len(sh_return)))
        r = []
        for t in range(1, len(sh_return)):
            r.append(sh_return[t] - sh_return[t-1])
        rd = np.array(r)
        plt.plot(rd, c='b')
        plt.title('Random Walk fit SHIndex Return')
        plt.show()
        rd_acfs = stattools.acf(rd)
        print(rd_acfs)
        tsaplots.plot_acf(rd, use_vlines=True, lags=30)
        plt.show()
```

Listing 3: 随机游走拟合上证综指收益率

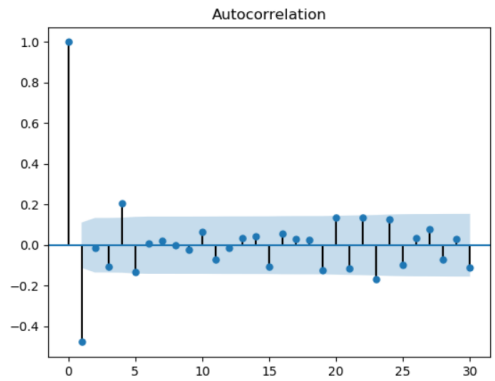
其残差图像为:

Figure 10: 残差图形



自相关系数函数 ACF 图形:

Figure 11: 自相关系数函数 ACF



由图11所示，在 1、4 时间点，明显超出置信范围，因此随机游走过程不能很好的拟合上证综指收益率时间序列信号。程序的运行结果如下所示:

Figure 12: 程序运行结果

```
量化投资以python为工具
第23章：时间序列基本性质
时间序列长为：N=311
[ 1.00000000e+00 -4.76495633e-01 -1.54901215e-02 -1.06305387e-01
  2.07607804e-01 -1.33423618e-01  7.48593453e-03  2.06096879e-02
 -6.05232517e-04 -2.38979190e-02  6.74201467e-02 -6.91058174e-02
 -1.32615416e-02  3.58256731e-02  4.58501980e-02 -1.05971579e-01
  5.84057765e-02  2.86644902e-02  2.44271741e-02 -1.21337912e-01
  1.34541909e-01 -1.13442293e-01  1.34590834e-01 -1.67581384e-01
  1.28908918e-01 -9.57150010e-02  3.63403946e-02  8.10977064e-02
 -7.23949376e-02  2.91387907e-02 -1.08475502e-01  1.30374612e-01
 -1.13879167e-01  8.04183633e-02 -1.16678698e-02  4.48580346e-02
 -8.87779106e-02  6.77491004e-02 -9.87007838e-03 -1.07688287e-03
 -4.10958841e-02]
```

第 2 章 ARIMA 模型

Abstract

在本章中我们将首先讲述自回归模型 AR(p)，接着讲述移动平均 MA(q)，最后讲解 ARMA(p,q)，然后将其泛化为 ARIMA(p,d,q)，分别将这些模型用于实际金融时间序列数据拟合。aqt001.py

2 ARIMA 模型

2.1 稳定性和模型选择标准

2.1.1 强稳定性

在我们以前的讨论中，我们说如果一个时间序列各个时间点所对应的随机变量，只要均值和方差不变，就是平稳时间序列。下面我们对强平稳性进行定义。
对于一个时间序列 $\{x_t\}$，如果对于 $\forall t_i, m$，两个序列：$x_{t_1}, x_{t_2}, ..., x_{t_N}$ 和 $x_{t_1+m}, x_{t_2+m}, ..., x_{t_N+m}$ 的统计特性完全相同，则说明该时间序列为强平稳特性。

2.1.2 模型选择标准

我们将用 AIC 来进行模型选择，AIC 的全称为：Akaike Information Criterion，我们通常会选择 AIC 值较小的模型。在实际应用中，还可以使用 BIC 来进行模型选择，BIC 的全称为 Bayes Information Criterion。在本章中我们只用 AIC 来进行模型选择。
假设统计模型的似然函数有 k 参数，最大似然值为 L，则 AIC 定义为：

$$AIC = -2\log(L) + 2k \tag{24}$$

由式24可知，最大似然值越大或者参数越少，AIC 的值越小，模型就越是好模型。

2.2 自回归模型

2.2.1 背景

我们可以扩展随机游走模型，使当前时间点数据不仅依赖前一时间点的值，同时还依赖前 p 个时间点的值，是这 p 个值的线性组合，这就得到了自回归模型。

2.2.2 模型定义

自回归模型 AR(p) 是随机游走模型的扩展，p 阶自回归模型定义为：

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + ... + \alpha_p x_{t-p} + w_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + w_t \tag{25}$$

其中 $\{w_t\}$ 为白噪声，$\alpha_i \in R$ 且 $\alpha_p \neq 0$。
当 $p=1$ 且 $\alpha_1 = 1$ 时，自回归模型就退化为随机游走模型。
为后续讨论方便，我们定义如下运算符：

$$\theta_p(B)x_t = (1 - \alpha_1 B - \alpha_2 B^2 - ... - \alpha_p B^p)x_t = w_t \tag{26}$$

有了上述模型之后，我们就可以直接拿来作预测，如下所示：

$$\begin{aligned} \hat{x}_t &= \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + ... + \alpha_p x_{t-p} \\ \hat{x}_{t+1} &= \alpha_1 \hat{x}_t + \alpha_2 x_{t-1} + ... + \alpha_p x_{t-p+1} \end{aligned} \tag{27}$$

然后依此类推，可以求出其后 n 个时间点的预测值。

2.2.3 二阶特性

我们首先定义特性方程：

$$\theta_p(B) = 0 \tag{28}$$

解这个方程得到的解的绝对值必须大于才是平稳序列。我们可以举几个实例，首先是随机游走序列：

随机游走 根据定义 $x_t = x_{t-1} + w_t$ 我们可以得到 $\alpha_1 = 1$，其特性方程为 $\theta = 1 - B = 0$，其解为 $B = 1$，因为其解的绝对值不大于 1，所以其不是平稳模型。

1 阶自回归 我们假设自回归模型为 $x_t = \frac{1}{4}x_{t-1} + w_t$，其中 $\alpha_1 = \frac{1}{4}$，其特性方程为 $\theta = 1 - \frac{1}{4} = 0$，其解为 $B = 4$，该解绝对值大于 1，所以其是平稳模型。

2 阶自回归模型 我们假设自回归模型为 $x_t = \frac{1}{2}x_{t-1} + \frac{1}{2}x_{t-2} + w_t$，其特性方程为 $\theta_2(B) = \frac{1}{2}(1 - B)(B + 2) = 0$，则其解为 $B = 1, -2$，其中一个解为单位根，其绝对值不大于 1，因此本模型不是平稳模型。虽然本模型不是平稳模型，但是其他 2 阶自回归模型是完全有可能是平稳模型的。

二阶特性 均值、自协方差、自相关系数定义如下所示：

$$\begin{aligned} \mu_x &= E(x_t) = 0 \\ \gamma_k &= \sum_{i=1}^{p} \alpha_i \gamma_{k-i}, \quad k > 0 \\ \rho_k &= \sum_{i=1}^{p} \alpha_i \rho_{k-i}, \quad k > 0 \end{aligned} \tag{29}$$

2.2.4 相关图

2.2.5 模拟数据

下面我们来模拟一个 AR(2) 的时间序列，我们的数据生成和拟合代码如下所示：

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from matplotlib.font_manager import FontProperties
from statsmodels.tsa import stattools
from statsmodels.graphics import tsaplots
from statsmodels.tsa.arima_model import ARIMA

class Aqt001(object):
    def __init__(self):
        self.name = 'Aqt001'

    def startup(self):
        print('ARMA模型...')
```

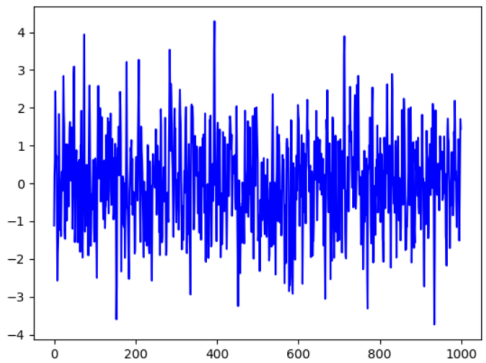
```
        self.simulate_ar2()

    def simulate_ar2(self):
        print('模拟AR(2)')
        alpha1 = 0.666
        alpha2 = -0.333
        wt = np.random.standard_normal(size=1000)
        x = wt
        for t in range(2, len(wt)):
            x[t] = alpha1 * x[t-1] + alpha2 * x[t-2] + wt[t]
        plt.plot(x, c='b')
        plt.show()
        ar2 = stattools.ARMA(x, (2, 0)).fit(disp=False)
        print('p={0} **** {1}; q={2}***{3}; {4} - {5} - {6}'.format(
                    ar2.k_ar, ar2.arparams, ar2.k_ma, ar2.maparams,
                    ar2.aic, ar2.bic, ar2.hqic)
        )
        arima2_0_0 = ARIMA(x, order=(2, 0, 0)).fit(disp=False)
        print('ARIMA: p={0} **** {1}; q={2}***{3}; {4} - {5} - {6}'. \
                    format(arima2_0_0.k_ar, arima2_0_0.arparams,
                    arima2_0_0.k_ma, arima2_0_0.maparams,
                    arima2_0_0.aic, arima2_0_0.bic,
                    arima2_0_0.hqic)
        )
        resid = arima2_0_0.resid
        # 绘制ACF
        acfs = stattools.acf(resid)
        print(acfs)
        tsaplots.plot_acf(resid, use_vlines=True, lags=30)
        plt.title('ACF figure')
        plt.show()
        pacfs = stattools.pacf(resid)
        print(pacfs)
        tsaplots.plot_pacf(resid, use_vlines=True, lags=30)
        plt.title('PACF figure')
        plt.show()
```

Listing 4: AR 数据模拟和拟合示例

我们首先生成一个 1000 个数据点的均值为 0 方差为 1 的白噪声数据，然后根据 $x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + w_t = 0.666 \times x_{t-1} - 0.333 \times x_{t-2} + w_t$ 公式，生成拟合数据。我们绘制该模拟数据图像，接着我们分别用 ARMA 和 ARIMA 进行拟合，求出系数，然后计算出模拟选择的参数：AIC、BIC、HQIC 的值，最后我们求出模型拟合的残差，并绘制出残差自相关系数函数 ACF 和偏自相关系数函数 PACF 的图像。模拟数据图像为：

Figure 13: AR2 模拟生成数据图



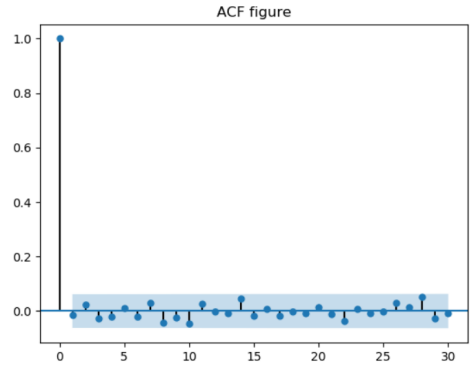
无论是 ARMA 还是 ARIMA 拟合，我们都可以得到较为正确的数据，同时残差的 ACF 和 PACF 也表明其是白噪声序列，运行结果如下所示：

Figure 14: 程序运行结果

```
量化投资以python为工具
ARMA模型...
模拟AR(2)
p=2 **** [ 0.67734879 -0.30934048]; q=0***[]; 2805.45882423338 - 2825.0898453493082 - 2812.919982104708
ARIMA: p=2 **** [ 0.67734879 -0.30934048]; q=0***[]; 2805.45882423338 - 2825.0898453493082 - 2812.919982104708
[ 1.00000000e+00 -1.27594914e-02  2.33980378e-02 -2.83973503e-02
 -1.99624524e-02  1.17663389e-02 -1.95196686e-02  2.85292757e-02
 -4.28202635e-02 -2.52670740e-02 -4.48889855e-02  2.56210627e-02
 -3.07874134e-03 -6.99614103e-03  4.61211026e-02 -1.70347399e-02
  6.79310491e-03 -1.76508823e-02 -3.03059320e-03 -7.93521381e-03
  1.47847033e-02 -1.00313054e-02 -3.58293988e-02  6.74092169e-03
 -8.77538910e-03 -4.66989248e-04  2.93222689e-02  1.34644761e-02
  5.12237137e-02 -2.72023000e-02 -8.38201436e-03 -2.17939430e-02
 -5.25994727e-02 -4.20756414e-02 -1.87352437e-02  1.33258139e-02
  6.19913054e-03  4.16316144e-03  1.61043531e-02 -1.14240579e-02
 -1.32598149e-02]
[ 1.          -0.01277226  0.0232856  -0.0279126  -0.02130531  0.0126565
 -0.01921062  0.0265783  -0.04146117 -0.02846307 -0.04359029  0.02540431
 -0.00479445 -0.00996858  0.04490654 -0.01295893  0.00138853 -0.01396058
 -0.00845372 -0.00942079  0.01628618 -0.01232774 -0.03515738  0.00746495
 -0.00265584 -0.00863967  0.03102144  0.01237213  0.04962798 -0.02321584
 -0.01278117 -0.02126148 -0.05607669 -0.04583023 -0.01970885  0.01196684
  0.01507584  0.00205992  0.01972234 -0.01629268 -0.02133968]
```

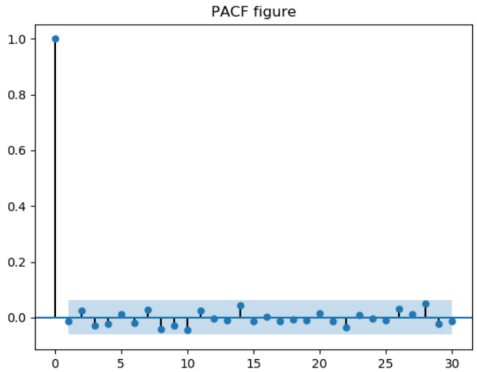
残差的自相关系数函数 ACF 图：

Figure 15: 残差的自相关系数函数 ACF 图



残差的偏自相关系数函数 PACF 图：

Figure 16: 残差的偏自相关系数函数 PACF 图



由 ACF 和 PACF 图可以看出，我们残差是比较典型的随机白噪声序列，由此可见我们拟合还是很好的。

2.2.6 金融数据拟合

2.2.7 预测

3 汇总

f000014 c000005 e000030

参考文献：I.MLearning [1999]—A.NikolaosAI [1999]—Bakry et al. [2015]

References

Andreas Nikolaos A.NikolaosAI. *A Book He Wrote*. His Publisher, Erewhon, NC, 1999.

Amr Bakry, Mohamed Elhoseiny, Tarek El-Gaaly, and Ahmed M. Elgammal. Digging deep into the layers of cnns: In search of how cnns achieve view invariance. *CoRR*, abs/1508.01983, 2015. URL http://arxiv.org/abs/1508.01983.

Ivan Marc I.MLearning. Some related article I wrote. *Some Fine Journal*, 99(7):1–100, January 1999.