时序分析(5) -- ARMA(p,q)模型

如无特殊说明，本系列文章中的数据将使用2012~2017年，分别代表国内股票、香港股票、国内债卷和国内货币的四个指数数据。

> 前两篇文章我们分别探讨了AR模型和MA模型对时序数据进行建模，这一节我们主要讨论ARMA模型。从名字中我们可以推知，ARMA模型是AR模型和MA模型的一种组合。

首先我们介绍ARMA模型的基本概念：

AutoRegression Moving Average Models - ARMA(p,q)

- AR模型是尝试捕捉和解释金融交易市场的动量和均值反转效果
- MA模型是尝试捕捉和解释在白噪声项中所观测到的振荡效果，这些震荡可以被理解为影响所观测过程的非预期事件造成的影响，例如超额收益等。

ARMA模型就是这两者的联合，它的主要缺点是忽略了在金融市场时序数据中经常可见的波动聚簇现象(Volatility Clustering)，模型公式如下：

$$
\begin{aligned}
y_t &= \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \ldots + \alpha_p y_{t-p} + \omega_t + \beta_1 \omega_{t-1} + \ldots + \beta_q \omega_{t-q} \\
&= \sum_{i=1}^{p} \alpha_i y_{t-i} + \omega_t + \sum_{i=1}^{q} \beta_i \omega_{t-i}
\end{aligned}
$$

In [1]:

```
import warnings
warnings.simplefilter('ignore')
```

1. 导入python包

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

from fintechtools.backtest import *
from fintechtools.datasource import *
#from fintechtools.SimuMultiTest import *
#from lib.portfolio import DailySimulator
#from lib.experiment import Experiment

import statsmodels.formula.api as smf
import statsmodels.tsa.api as smt
import statsmodels.api as sm
import scipy.stats as scs
from arch import arch_model
#sns.set_context("talk")
import matplotlib
import matplotlib as mpl
from matplotlib.ticker import FuncFormatter
mpl.style.use('classic')

plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['font.serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
import seaborn as sns
sns.set_style("whitegrid",{"font.sans-serif":['simhei', 'Arial']})
sns.set_context("talk")

#zhfont1 = matplotlib.font_manager.FontProperties(fname='C:\Users\ktwc37\Documents\ZNTG\notebooks\S7


%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

2. 读入数据

In [3]:

```
start = '2012-01-01'
end = '2017-02-05'
```

In [4]:

```
indexs = pd.read_excel('./data/华夏指数.xlsx')
indexs_pv = indexs.pivot_table(index='日期', columns= '简称', values= '收盘价(元)')
indexs_pv.index = pd.to_datetime(indexs_pv.index, unit='d')
```

In [5]:

```
indexs_pv.columns = ['国内债券', '国内股票', '香港股票', '国内货币']
indexs_pv = indexs_pv[['国内债券', '国内股票', '国内货币', '香港股票']]
indexs_pv.fillna(axis=0,method='bfill',inplace=True)
indexs_sub = indexs_pv.loc[start:end,]
```

国内债卷：中债综合财富(总值)指数
国内股票：中证全指
香港股票：恒生指数
国内货币：货币基金

In [6]:

```
indexs_sub.head()
```

Out[6]:

	国内债券	国内股票	国内货币	香港股票
日期				
2012-01-04	141.5160	2571.951	1166.7726	18727.31
2012-01-05	141.5501	2513.699	1166.9696	18813.41
2012-01-06	141.7277	2527.247	1167.1185	18593.06
2012-01-09	141.8669	2619.638	1167.5058	18865.72
2012-01-10	142.0118	2713.529	1167.6330	19004.28

In [7]:

```
indexs_logret = indexs_sub.apply(log_return).dropna()
```

In [8]:

```
indexs_logret.head()
```

Out[8]:

	国内债券	国内股票	国内货币	香港股票
日期				
2012-01-05	0.000241	-0.022909	0.000169	0.004587
2012-01-06	0.001254	0.005375	0.000128	-0.011782
2012-01-09	0.000982	0.035906	0.000332	0.014558
2012-01-10	0.001021	0.035214	0.000109	0.007318
2012-01-11	0.000188	-0.002115	0.000113	0.007740

In [9]:

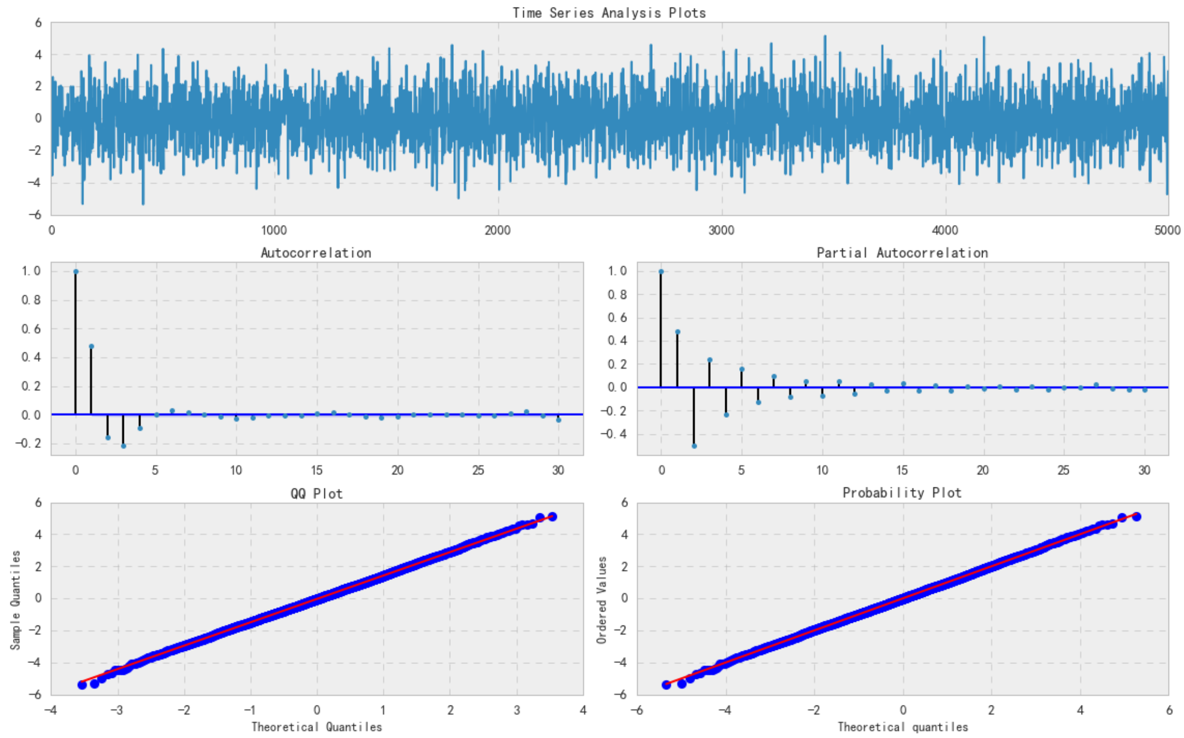
```
def tsplot(y, lags=None, figsize=(16, 10), style='bmh'):
    if not isinstance(y, pd.Series):
        y = pd.Series(y)
    with plt.style.context(style):
        fig = plt.figure(figsize=figsize)
        #mpl.rcParams['font.family'] = 'Ubuntu Mono'
        layout = (3, 2)
        ts_ax = plt.subplot2grid(layout, (0, 0), colspan=2)
        acf_ax = plt.subplot2grid(layout, (1, 0))
        pacf_ax = plt.subplot2grid(layout, (1, 1))
        qq_ax = plt.subplot2grid(layout, (2, 0))
        pp_ax = plt.subplot2grid(layout, (2, 1))

        y.plot(ax=ts_ax)
        ts_ax.set_title('Time Series Analysis Plots')
        smt.graphics.plot_acf(y, lags=lags, ax=acf_ax, alpha=0.5)
        smt.graphics.plot_pacf(y, lags=lags, ax=pacf_ax, alpha=0.5)
        sm.qqplot(y, line='s', ax=qq_ax)
        qq_ax.set_title('QQ Plot')
        scs.probplot(y, sparams=(y.mean(), y.std()), plot=pp_ax)

        plt.tight_layout()
    return
```

模拟ARMA(2,3)过程

下面我们以$\beta = [0.5, -0.3,$ $\alpha = [0.5, -0.25]$来模拟一个ARMA(2,2)过程

In [15]:

```
# Simulate an ARMA(2, 2) model with alphas=[0.5,-0.25] and betas=[0.5,-0.3]
max_lag = 30

n = int(5000) # lots of samples to help estimates
burn = int(n/10) # number of samples to discard before fit

alphas = np.array([0.5, -0.25])
betas = np.array([0.5, -0.3])
ar = np.r_[1, -alphas]
ma = np.r_[1, betas]

arma22 = smt.arma_generate_sample(ar=ar, ma=ma, nsample=n, burnin=burn)
_ = tsplot(arma22, lags=max_lag)
```



现在我们对此模拟数据用ARMA(2,2)进行建模

In [16]:

```
mdl = smt.ARMA(arma22, order=(2, 2)).fit(
    maxlag=max_lag, method='mle', trend='nc', burnin=burn)
print(mdl.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 5000
Model:                     ARMA(2, 2)   Log Likelihood               -7115.477
Method:                           mle   S.D. of innovations              1.004
Date:                Thu, 12 Jul 2018   AIC                          14240.955
Time:                        19:10:30   BIC                          14273.541
Sample:                             0   HQIC                         14252.375

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1.y        0.5859      0.052     11.335      0.000       0.485       0.687
ar.L2.y       -0.2640      0.015    -17.544      0.000      -0.294      -0.235
ma.L1.y        0.4138      0.053      7.807      0.000       0.310       0.518
ma.L2.y       -0.3684      0.049     -7.542      0.000      -0.464      -0.273
                                    Roots
=============================================================================
                 Real           Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1           1.1096           -1.5989j            1.9462           -0.1534
AR.2           1.1096           +1.5989j            1.9462            0.1534
MA.1          -1.1790           +0.0000j            1.1790            0.5000
MA.2           2.3022           +0.0000j            2.3022            0.0000
-----------------------------------------------------------------------------
```

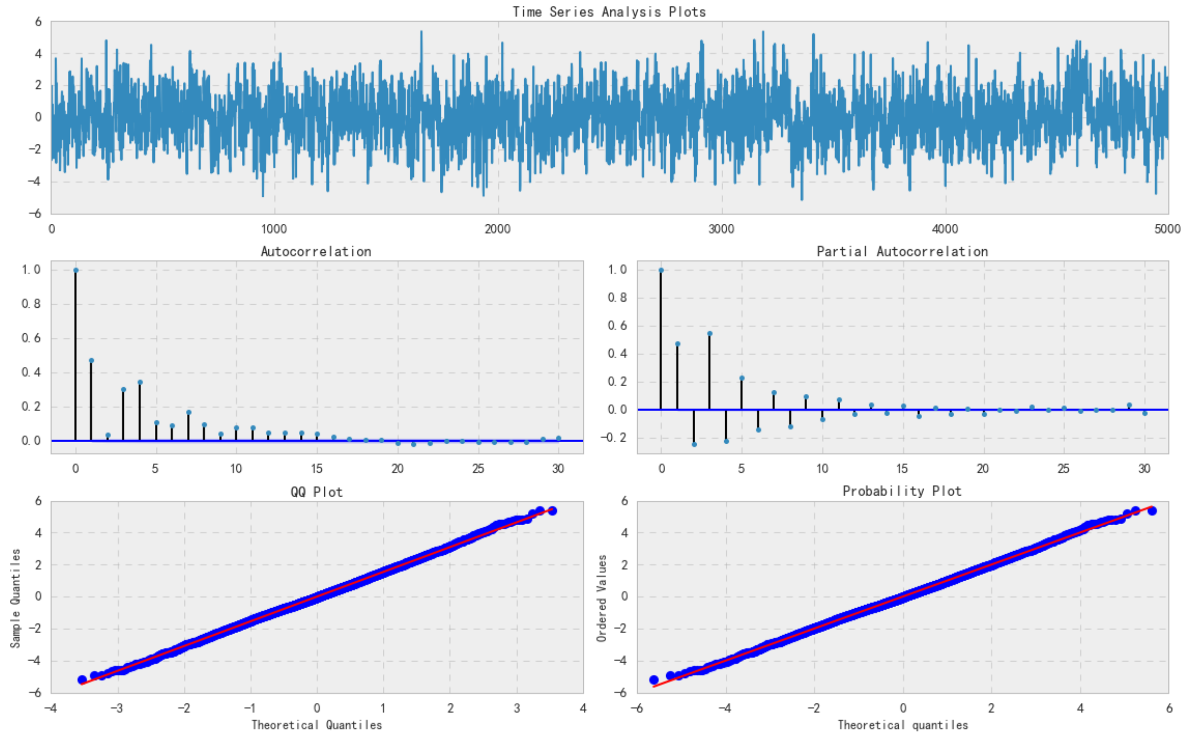
显然，ARMA模型准确的回归出了模拟数据设定的参数。

下面，我们模拟一个ARMA(3,2)过程，然后用ARMA模型来建模，搜索参数并给出赤池信息量(Akaike Information Criterion)AIC最低的p,q。

In [17]:

```
# Simulate an ARMA(3, 2) model with alphas=[0.5,-0.25,0.4] and betas=[0.5,-0.3]

max_lag = 30

n = int(5000)
burn = 2000

alphas = np.array([0.5, -0.25, 0.4])
betas = np.array([0.5, -0.3])

ar = np.r_[1, -alphas]
ma = np.r_[1, betas]

arma32 = smt.arma_generate_sample(ar=ar, ma=ma, nsample=n, burnin=burn)
_ = tsplot(arma32, lags=max_lag)

# pick best order by aic
# smallest aic value wins
best_aic = np.inf
best_order = None
best_mdl = None

rng = range(5)
for i in rng:
    for j in rng:
        try:
            tmp_mdl = smt.ARMA(arma32, order=(i, j)).fit(method='mle', trend='nc')
            tmp_aic = tmp_mdl.aic
            if tmp_aic < best_aic:
                best_aic = tmp_aic
                best_order = (i, j)
                best_mdl = tmp_mdl
        except: continue


print('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)

aic: 14266.72269 | order: (3, 2)
```



非常好，我们准确的找到了p,q。

In [18]:

```
print(best_mdl.summary())
```

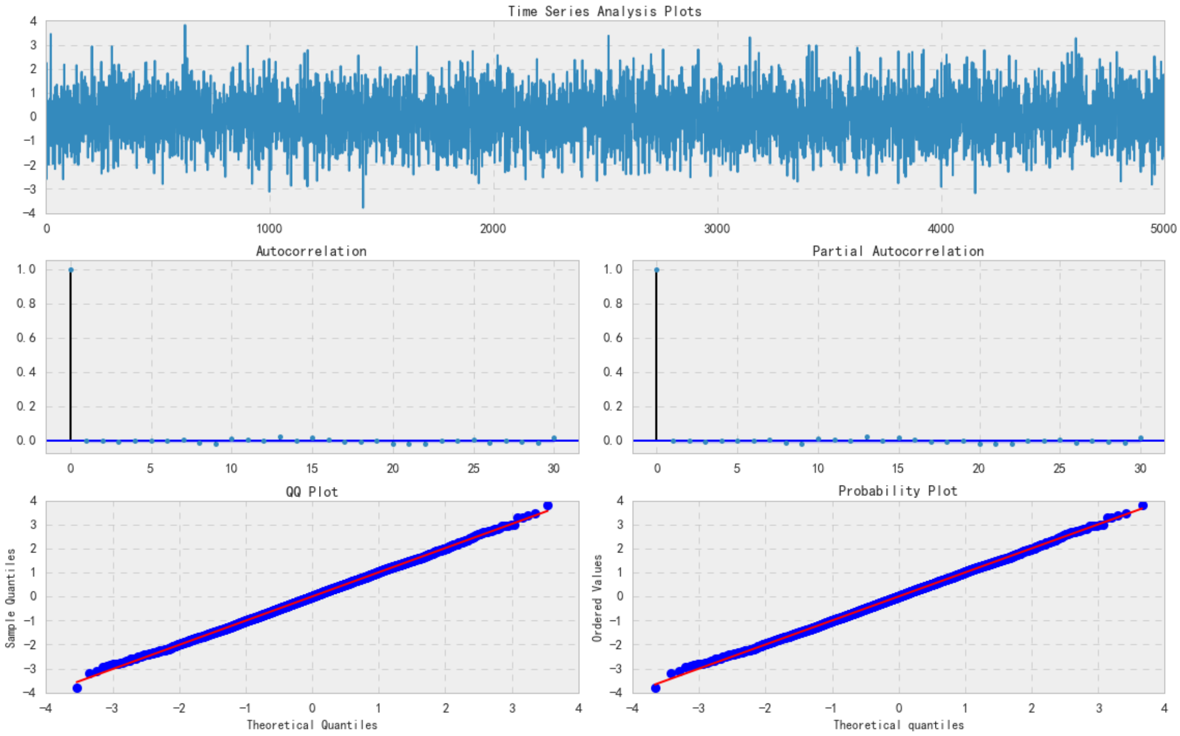
```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 5000
Model:                     ARMA(3, 2)   Log Likelihood               -7127.361
Method:                           mle   S.D. of innovations              1.006
Date:                Thu, 12 Jul 2018   AIC                          14266.723
Time:                        19:11:51   BIC                          14305.826
Sample:                             0   HQIC                         14280.428

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1.y        0.5258      0.030     17.367      0.000       0.466       0.585
ar.L2.y       -0.2770      0.016    -17.226      0.000      -0.309      -0.245
ar.L3.y        0.4151      0.014     30.284      0.000       0.388       0.442
ma.L1.y        0.4752      0.033     14.563      0.000       0.411       0.539
ma.L2.y       -0.3211      0.031    -10.259      0.000      -0.382      -0.260
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.2300           -0.0000j            1.2300           -0.0000
AR.2           -0.2813           -1.3710j            1.3996           -0.2822
AR.3           -0.2813           +1.3710j            1.3996            0.2822
MA.1           -1.1736           +0.0000j            1.1736            0.5000
MA.2            2.6535           +0.0000j            2.6535            0.0000
-----------------------------------------------------------------------------
```

我们也准确地找到了参数α, β

残差plot

In [19]:

```
_ = tsplot(best_mdl.resid, lags=max_lag)
```



残差与高斯白噪声非常拟合。

下一步，我们要尝试使用ARMA模型对四个指数数据建模。

- 国内股票收益率
  以ARMA建模， 优化目标为Max Likelihood Estimation，得到阶数为(3,2)

In [20]:

```
best_aic = np.inf
best_order = None
best_mdl = None
Y = indexs_logret['国内股票']
rng = range(5)
for i in rng:
    for j in rng:
        try:
            tmp_mdl = smt.ARMA(Y, order=(i, j)).fit(method='mle', trend='nc')
            tmp_aic = tmp_mdl.aic
            if tmp_aic < best_aic:
                best_aic = tmp_aic
                best_order = (i, j)
                best_mdl = tmp_mdl
        except: continue


print('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
```

```
aic: -6601.86081 | order: (3, 2)
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
```

In [21]:

```
print(best_mdl.summary())
```

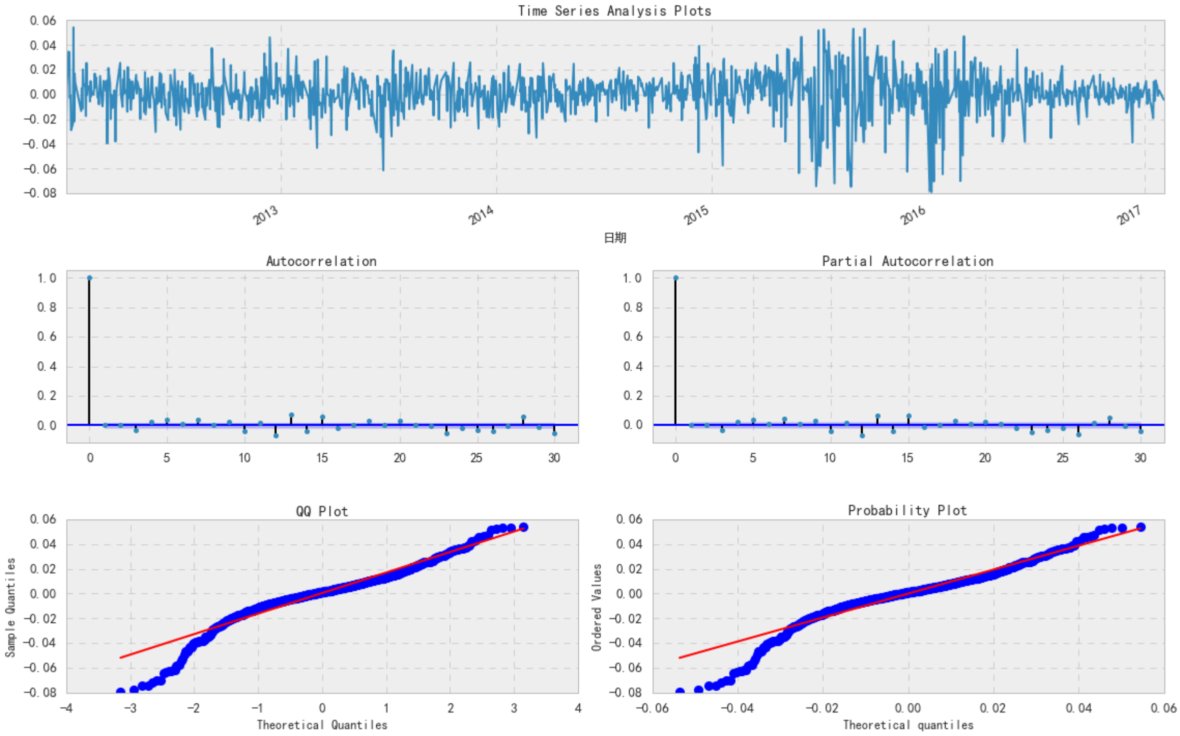
```
                              ARMA Model Results
==============================================================================
Dep. Variable:                   国内股票   No. Observations:                 1234
Model:                     ARMA(3, 2)   Log Likelihood                3306.930
Method:                           mle   S.D. of innovations              0.017
Date:                Thu, 12 Jul 2018   AIC                          -6601.861
Time:                        19:29:11   BIC                          -6571.153
Sample:                    01-05-2012   HQIC                         -6590.309
                         - 02-03-2017
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1.国内股票     0.3035      0.033      9.106      0.000       0.238       0.369
ar.L2.国内股票    -0.9876      0.014    -68.697      0.000      -1.016      -0.959
ar.L3.国内股票     0.1147      0.029      3.950      0.000       0.058       0.172
ma.L1.国内股票    -0.2264      0.019    -12.219      0.000      -0.263      -0.190
ma.L2.国内股票     0.9566      0.017     57.387      0.000       0.924       0.989
                                    Roots
=============================================================================
                 Real           Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            0.0957           -1.0131j            1.0176           -0.2350
AR.2            0.0957           +1.0131j            1.0176            0.2350
AR.3            8.4200           -0.0000j            8.4200           -0.0000
MA.1            0.1183           -1.0155j            1.0224           -0.2315
MA.2            0.1183           +1.0155j            1.0224            0.2315
-----------------------------------------------------------------------------
```

从拟合的模型指标上看，一阶参数都比较显著。

残差Plot

In [22]:

```
_ = tsplot(best_mdl.resid, lags=max_lag)
```



从QQ-plot上看，残差并非正态分布，所以说明模型拟合并不是非常完美。

- 香港股票收益率
  以ARMA建模
  优化目标为Max Likelihood Estimation，拟合参数为(2,2)

In [25]:

```
Y = indexs_logret['香港股票']
rng = range(5)
for i in rng:
    for j in rng:
        try:
            tmp_mdl = smt.ARMA(Y, order=(i, j)).fit(method='mle', trend='nc')
            tmp_aic = tmp_mdl.aic
            if tmp_aic < best_aic:
                best_aic = tmp_aic
                best_order = (i, j)
                best_mdl = tmp_mdl
        except: continue


print('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)

aic: -7640.79631 | order: (2, 2)

d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
```

In [26]:

```
print(best_mdl.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                   香港股票   No. Observations:                 1234
Model:                     ARMA(2, 2)   Log Likelihood                3825.398
Method:                           mle   S.D. of innovations              0.011
Date:                Thu, 12 Jul 2018   AIC                          -7640.796
Time:                        19:32:56   BIC                          -7615.206
Sample:                    01-05-2012   HQIC                         -7631.170
                         - 02-03-2017
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1.香港股票    -0.1710      0.005    -37.855      0.000      -0.180      -0.162
ar.L2.香港股票    -0.9924      0.004   -239.949      0.000      -1.001      -0.984
ma.L1.香港股票     0.1804      0.002     73.455      0.000       0.176       0.185
ma.L2.香港股票     1.0000      0.005    201.497      0.000       0.990       1.010
                                    Roots
=============================================================================
                 Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1          -0.0862          -1.0001j            1.0038           -0.2637
AR.2          -0.0862          +1.0001j            1.0038            0.2637
MA.1          -0.0902          -0.9959j            1.0000           -0.2644
MA.2          -0.0902          +0.9959j            1.0000            0.2644
-----------------------------------------------------------------------------
```

回归系数都比较显著

- 国内债券收益率
  以ARMA建模
  优化目标为Max Likelihood Estimation，得到(3,1).

In [27]:

```
Y = indexs_logret['国内债券']
rng = range(5)
for i in rng:
    for j in rng:
        try:
            tmp_mdl = smt.ARMA(Y, order=(i, j)).fit(method='mle', trend='nc')
            tmp_aic = tmp_mdl.aic
            if tmp_aic < best_aic:
                best_aic = tmp_aic
                best_order = (i, j)
                best_mdl = tmp_mdl
        except: continue


print('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)

aic: -14105.28291 | order: (3, 1)
```

In [28]:

```
print(best_mdl.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                   国内债券   No. Observations:                 1234
Model:                     ARMA(3, 1)   Log Likelihood                7057.641
Method:                           mle   S.D. of innovations              0.001
Date:                Thu, 12 Jul 2018   AIC                         -14105.283
Time:                        19:35:52   BIC                         -14079.693
Sample:                    01-05-2012   HQIC                        -14095.657
                         - 02-03-2017
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1.国内债券     1.3945      0.031     44.862      0.000       1.334       1.455
ar.L2.国内债券    -0.3340      0.048     -6.961      0.000      -0.428      -0.240
ar.L3.国内债券    -0.0629      0.029     -2.138      0.033      -0.121      -0.005
ma.L1.国内债券    -0.9845      0.012    -79.206      0.000      -1.009      -0.960
                                    Roots
=============================================================================
                 Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.0045           +0.0000j            1.0045            0.0000
AR.2            1.9219           +0.0000j            1.9219            0.0000
AR.3           -8.2389           +0.0000j            8.2389            0.5000
MA.1            1.0157           +0.0000j            1.0157            0.0000
-----------------------------------------------------------------------------
```

回归系数都比较显著

- 国内货币收益率
  以ARMA建模
  优化目标为Max Likelihood Estimation，得到(3,4).

In [30]:

```
Y = indexs_logret['国内货币']
rng = range(5)
for i in rng:
    for j in rng:
        try:
            tmp_mdl = smt.ARMA(Y, order=(i, j)).fit(method='mle', trend='nc')
            tmp_aic = tmp_mdl.aic
            if tmp_aic < best_aic:
                best_aic = tmp_aic
                best_order = (i, j)
                best_mdl = tmp_mdl
        except: continue


print('aic: {:6.5f} | order: {}'.format(best_aic, best_order))
```

```
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
d:\Anaconda3\lib\site-packages\statsmodels\base\model.py:496: ConvergenceWarning: Ma
ximum Likelihood optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)

aic: -19062.72857 | order: (3, 4)
```

In [31]:

```
print(best_mdl.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                   国内货币   No. Observations:                 1234
Model:                     ARMA(3, 4)   Log Likelihood                9539.364
Method:                           mle   S.D. of innovations              0.000
Date:                Thu, 12 Jul 2018   AIC                         -19062.729
Time:                        19:37:52   BIC                         -19021.784
Sample:                    01-05-2012   HQIC                        -19047.327
                         - 02-03-2017
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1.国内货币     1.5461   6.79e-05   2.28e+04      0.000       1.546       1.546
ar.L2.国内货币    -1.4877   6.74e-05  -2.21e+04      0.000      -1.488      -1.488
ar.L3.国内货币     0.9415   1.32e-05   7.14e+04      0.000       0.942       0.942
ma.L1.国内货币    -1.7193      0.026    -66.755      0.000      -1.770      -1.669
ma.L2.国内货币     1.6906      0.042     40.473      0.000       1.609       1.772
ma.L3.国内货币    -1.1665      0.041    -28.474      0.000      -1.247      -1.086
ma.L4.国内货币     0.2377      0.025      9.532      0.000       0.189       0.287
                                    Roots
=============================================================================
                 Real           Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.0000           -0.0000j            1.0000           -0.0000
AR.2            0.2900           -0.9889j            1.0306           -0.2046
AR.3            0.2900           +0.9889j            1.0306            0.2046
MA.1            0.2808           -1.0674j            1.1037           -0.2091
MA.2            0.2808           +1.0674j            1.1037            0.2091
MA.3            1.0468           -0.0000j            1.0468           -0.0000
MA.4            3.2994           -0.0000j            3.2994           -0.0000
-----------------------------------------------------------------------------
```

回归系数都比较显著。

总结

本文展示了采用Python语言为四个指数时序数据进行ARMA建模，介绍了ARMA模型的基本概念和AR、MA模型的联系，并使用ARMA模型对四指数数据进行建模。