用 Python 进行时间序列分析与预测

目录

引言 2
基础 2
线性模型 4
对数线性模型 4
自回归模型 AR（p） 5
移动平均模型 MA（q） 6
自回归移动平均模型 ARMA（p，q） 8
自回归综合移动平均模型 ARIMA（p，d，q） 11
自回归条件异方差模型 ARCH（p） 12
广义自回归条件异方差模型 GARCH（p，q） 13
时间序列分析预测以 SARIMA 为例 14
    数据集 14
    求解最优参数 15
    模型检验 15
    模型预测 15
代码 16
参考文献 16

引言

本文主要利用 Python 进行时间序列分析常见算法的运算和展示。系统得介绍了时间序列分析常见算法（AR、MA、ARMA、ARIMA、SARIMA、ARCH、GARCH）及其之间的联系与区别。时间序列分析试图理解过去并预测未来。通过时间序列分析技术，我们可以更好地了解已经发生的事情，并对未来做出更好，更有利的预测。

基础

时间序列是按时间顺序索引（列出或绘制图形）的一系列数据点。
平稳性是我们关注的重点。平稳的时间序列易于预测，因为我们可以假设未来的统计属性与当前的统计属性相同或成比例。我们在时间序列分析中使用的大多数模型都假设协方差平稳性。这意味着这些模型预测的描述性统计量（例如均值，方差和相关性）仅在时间序列稳定时才是可靠的，否则就无效。
我们一般遇到的大多数时间序列并不是固定不变的。因此，时间序列分析需要我们确定要预测的序列是否平稳，如果不是，我们必须找到方法对其进行变换以使其平稳（比如差分）。
自相关：本质上，当我们对时间序列建模时，我们将序列分解为三个部分：趋势，季节性/周期性和随机性。随机分量称为残差或误差。这只是我们的预测值和观察值之间的差异。序列相关是指时间序列模型的残差（误差）相互关联时的情况。
白噪声是最简单的时间序列模型之一。根据定义，作为白噪声过程的时间序列具有连续不相关的误差，并且这些误差的预期平均值等于零。如果时间序列模型成功地捕获了数据性质，模型的残差将变得类似于白噪声过程。因此，时间序列分析实际上是在尝试将模型拟合到某种时间序列模型，以使残差序列与白噪声无法区分。
我们模拟白噪声过程并进行可视化。下面我们用 Python 编写了一个函数，用于绘制时间序列并直观地分析序列相关性。

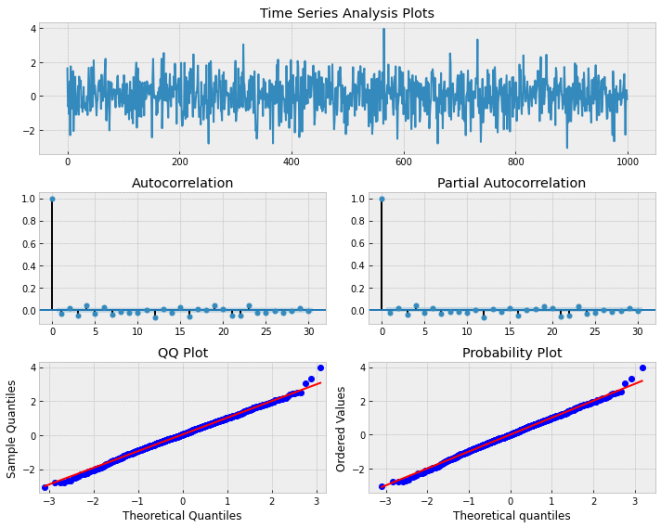

图 1 白噪声序列

我们可以看到该序列似乎是随机的，并且以零为均值。自相关（ACF）和部分自相关（PACF）图也表明没有明显的序列相关。下面，我们可以看到 QQ 和概率图，其将我们的数据分布与正态分布进行了比较。显然，我们的数据是随机分布的，并且看起来应该遵循高斯白噪声。
随机游走定义如下：随机游走序列是非平稳的，因为观察值之间的协方差是时间相关的。如

果我们建模的时间序列是随机游走，则其无法预测。
接着，我们用函数模拟随机游走，从标准正态分布中采样。

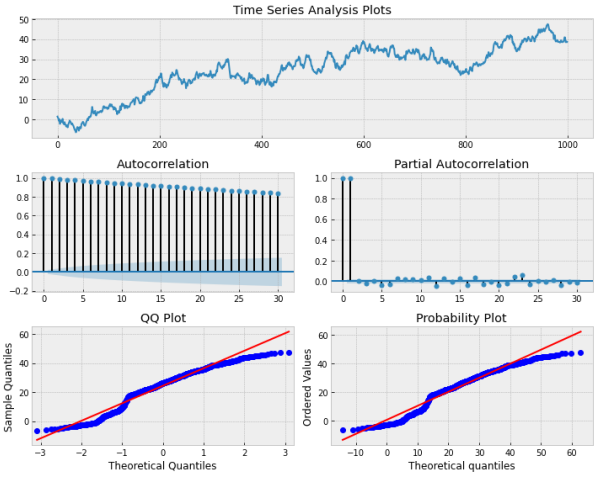

图 2 随机游走序列

显然，该随机游走序列不平稳。然而由于：

$$X_t = X_{t-1} + W_t$$

$$X_t - X_{t-1} = W_t$$

因此，随机游动序列的一阶差分等于白噪声过程。所以，我们据此对随机游走序列进行一阶差分。检验如下，其显然是白噪声过程。

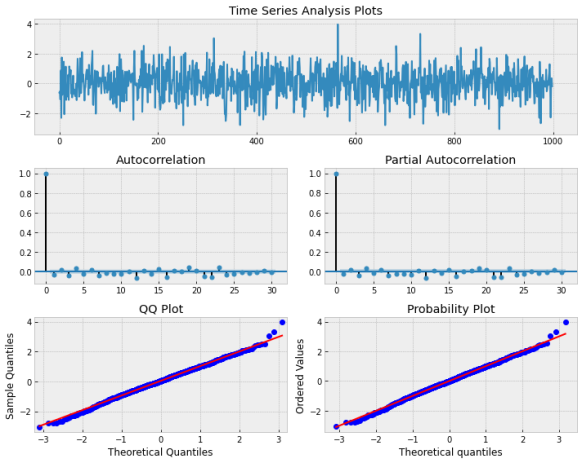

图 3 差分后的随机游走序列

刚刚我们所处理的时间序列为自己生成的数据。接着，我们在实际数据集上进行试验。我们通过雅虎财经的 API 获取标准普尔 500 指数的 2007~2015 年的数据。对其做一阶差分，其结果如下：

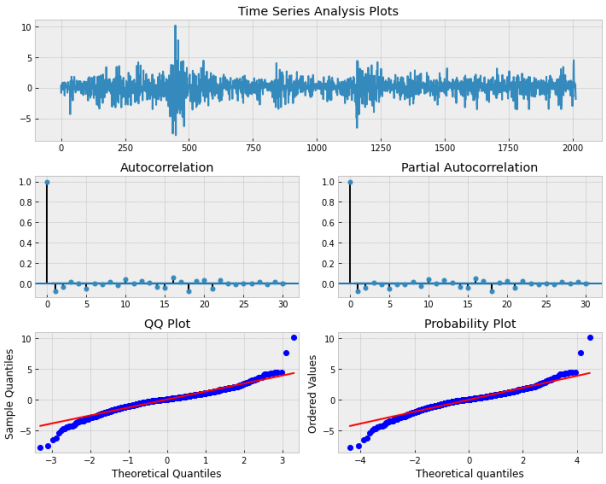

图 5 SPY 数据进行一阶差分后

该差分结果很像白噪声。但是，根据 QQ 图和概率图的形状，表明该序列接近正态分布，然而存在重尾性。这意味着应该有更好的模型来描述实际的价格变化过程。

线性模型

线性模型（又称趋势模型）表示可以使用直线绘制的时间序列。基本公式为：

$$y_t = b_0 + b_1 t + \epsilon_t$$

在此模型中，因变量的值由 beta 系数和自变量时间确定。接下来，我们使用人工生成数据集进行模拟，如下图。

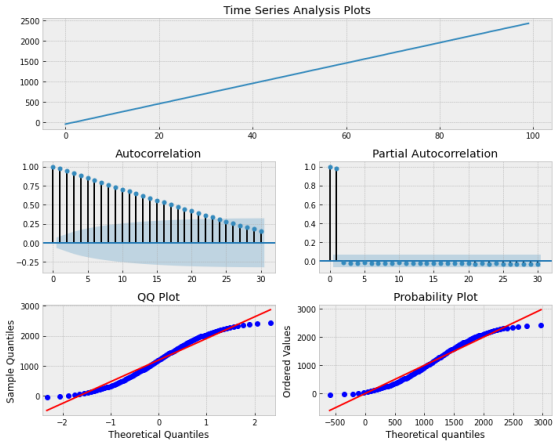

图 6 线性模型

在这里，我们可以看到模型的残差是相关的，并且呈现拖尾性。分布大致符合高斯分布。在使用该模型进行预测之前，我们必须考虑消除该序列中存在的明显的自相关。其 PACF 的显着性表明自回归模型可能是合适的。

对数线性模型

该模型与线性模型相似，我们进行类似的分析：

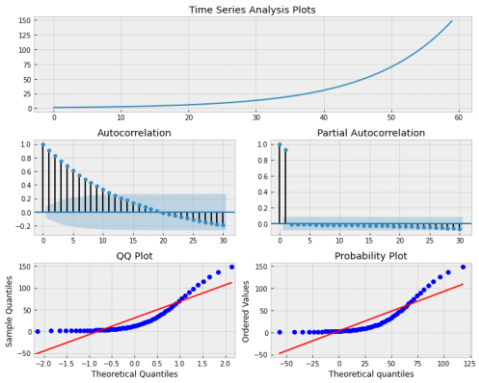

图 7 指数模型

自回归模型 AR（p）

当因变量根据自身的一个或多个滞后值进行回归时，该模型称为自回归模型。如下所示：

$$x_t = \alpha_1 x_{t-1} + \ldots + \alpha_p x_{t-p} + \omega_t$$

$$= \sum_{i=1}^{P} \alpha_i x_{t-i} + \omega_t$$

p 表示模型中使用的滞后变量的数量。例如，AR（2）模型或二阶自回归模型如下所示：

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \omega_t$$

其中，α 是系数，ω 是白噪声项。在 AR 模型中，α 不能等于零。接下来，我们模拟一个 α =0.6 的 AR（1）模型：

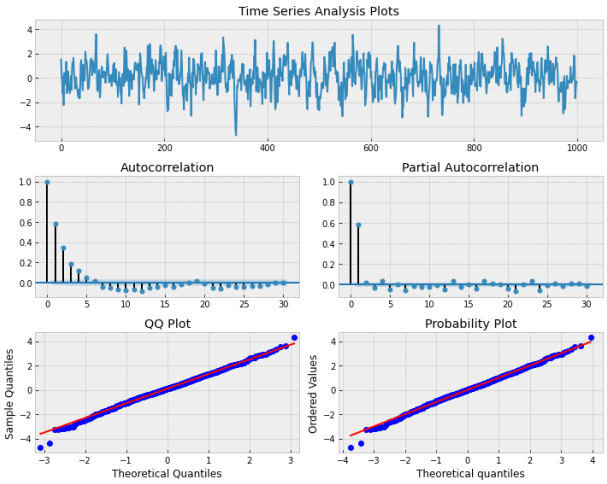

图 8 AR 模型

不出所料，我们的模拟 AR（1）模型的分布是正态的。如 PACF 图所示，其存在显著的偏自相关性，一阶截尾。

现在，我们可以使用 Python 的 statsmodels 包拟合 AR（p）模型。首先，进行模拟数据的估计，随后返回估计的 alpha 系数。然后，我们查看拟合模型是否会选择正确的滞后项。如果 AR 模型正确，则估算的 alpha 系数将接近我们的真实 alpha 值 0.6，阶数将等于 1。

```
alpha estimate: 0.58227 | best lag order = 1

true alpha = 0.6 | true order = 1
```

看起来估计得很好。接下来我们用 alpha_1 = 0.666 和 alpha_2 = -0.333 来模拟 AR（2）过程。

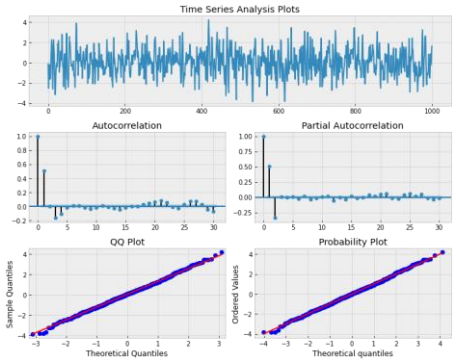

图 9 AR(2)数据

让我们看看是否可以通过模型拟合得到正确的参数。

```
coef estimate: 0.6760 -0.3393 | best lag order = 2

true coefs = [0.666, -0.333] | true order = 2
```

效果不错。

移动平均模型 MA（q）

MA（q）模型与 AR（p）模型非常相似。不同之处在于，MA（q）模型是过去的白噪声误差项的线性组合，而不是像 AR（p）模型那样的过去观测值的线性组合。MA（q）模型的公式为：

$$x_t = \omega_t + \beta_1\omega_{t-1} + \ldots + \beta_p\omega_{t-p}$$

$$= \omega_t + \sum_{i=1}^{P} \beta_i\omega_{t-i}$$

w 是白噪声，E（w_t）= 0。接下来，我们使用 beta = 0.6 生成人工数据集，进行模型可视化。

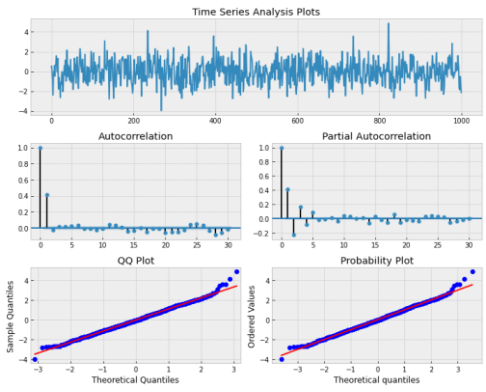

图 10 MA 模型

ACF 显示一阶截尾，这表明 MA（1）模型适用于我们的模拟序列。PACF 具有拖尾性。我们现在尝试将 MA（1）进行模型的拟合。

ARMA Model Results

Dep. Variable:	y	No. Observations:	1000
Model:	ARMA(0, 1)	Log Likelihood	-1390.513
Method:	mle	S.D. of innovations	0.972
Date:	Fri, 12 Jun 2020	AIC	2785.025
Time:	11:32:24	BIC	2794.841
Sample:	0	HQIC	2788.756

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1.y	0.5874	0.026	22.762	0.000	0.537	0.638

Roots

	Real	Imaginary	Modulus	Frequency
MA.1	-1.7024	+0.0000j	1.7024	0.5000

图 11 MA（1）模型摘要

该模型能够正确估计系数，0.58 接近我们的真实值 0.6。并且 95%置信区间确实包含真实值。接着，让我们尝试模拟一个 MA（3）序列，然后使用 ARMA 函数进行三阶 MA 模型拟合，观察是否可以进行正确的估计。Beta 1-3 分别等于 0.6、0.4 和 0.2。

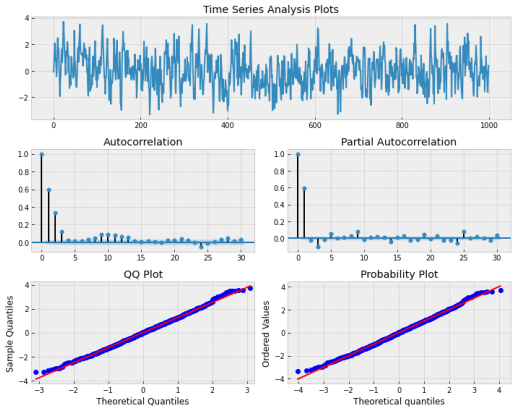

图 12 模拟的 MA（3）

ARMA Model Results

Dep. Variable:	y	No. Observations:	1000
Model:	ARMA(0, 3)	Log Likelihood	-1427.038
Method:	mle	S.D. of innovations	1.008
Date:	Fri, 12 Jun 2020	AIC	2862.075
Time:	13:12:30	BIC	2881.706
Sample:	0	HQIC	2869.536

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1.y	0.6025	0.031	19.322	0.000	0.541	0.664
ma.L2.y	0.4060	0.034	11.806	0.000	0.339	0.473
ma.L3.y	0.1683	0.031	5.420	0.000	0.107	0.229

Roots

	Real	Imaginary	Modulus	Frequency
MA.1	-0.1714	-1.6856j	1.6943	-0.2661
MA.2	-0.1714	+1.6856j	1.6943	0.2661
MA.3	-2.0700	-0.0000j	2.0700	-0.5000

图 13 模型摘要

该模型能够有效地估计实际系数。95%置信区间还包含 0.6、0.4 和 0.3 的真实参数值。现在，我们尝试利用 MA（3）模型拟合 SPY 数据。

ARMA Model Results

Dep. Variable:	SPY	No. Observations:	2013
Model:	ARMA(0, 3)	Log Likelihood	5756.951
Method:	mle	S.D. of innovations	0.014
Date:	Fri, 12 Jun 2020	AIC	-11505.902
Time:	13:12:30	BIC	-11483.472
Sample:	0	HQIC	-11497.669

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1.SPY	-0.0959	0.022	-4.314	0.000	-0.139	-0.052
ma.L2.SPY	-0.0737	0.023	-3.256	0.001	-0.118	-0.029
ma.L3.SPY	0.0274	0.022	1.260	0.208	-0.015	0.070

Roots

	Real	Imaginary	Modulus	Frequency
MA.1	-2.8909	-0.0000j	2.8909	-0.5000
MA.2	2.7906	-2.2012j	3.5543	-0.1063
MA.3	2.7906	+2.2012j	3.5543	0.1063

图 14 模型摘要

让我们看一下模型残差。ACF 和 PACF 都快速收敛。然而在 QQ 图上有重尾性，使其不是预测未来 SPY 指数的最佳模型。

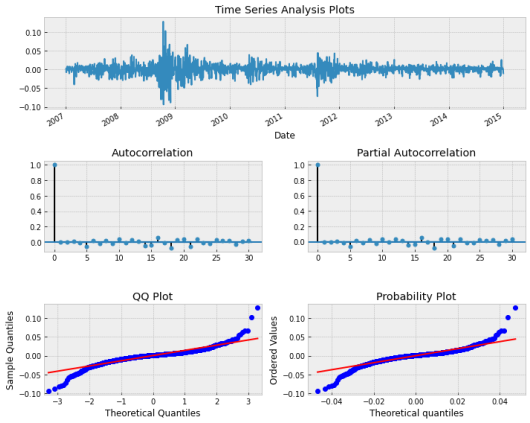

图 15 SPY MA（3）模型残差

自回归移动平均模型 ARMA（p，q）

ARMA 模型是 AR（p）和 MA（q）模型之间的合并。如果是在量化金融中，AR（p）模型试图捕获在交易市场中的动量和均值回归效应。MA（q）模型试图捕获以白噪声方式观察到的冲击效果，这些影响可以被认为是意外事件。模型公式为：

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \omega_t + \beta_1 \omega_{t-1} + \beta_2 \omega_{t-2} \ldots + \beta_q \omega_{t-q}$$

$$= \sum_{i=1}^{P} \alpha_i x_{t-i} + \omega_t + \sum_{i=1}^{Q} \beta_i \omega_{t-i}$$

我们用给定的参数模拟一个 ARMA（2，2）流程，然后拟合一个 ARMA（2，2）模型并查看它是否可以正确估计这些参数。将 alpha 设置为[0.5，-0.25]，将 beta 设置为[0.5，-0.3]。

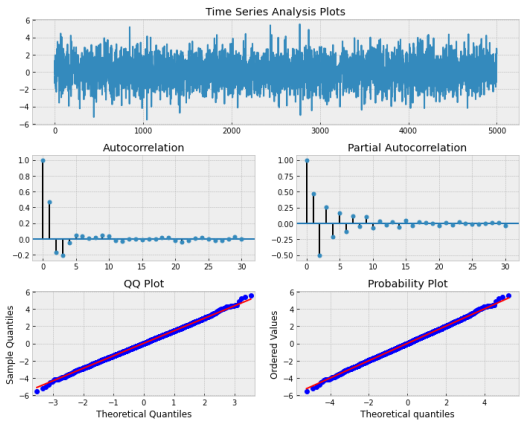

图 16 模拟的 ARMA（2，2）

ARMA Model Results

Dep. Variable:	y	No. Observations:	5000
Model:	ARMA(2, 2)	Log Likelihood	-7076.176
Method:	mle	S.D. of innovations	0.996
Date:	Fri, 12 Jun 2020	AIC	14162.352
Time:	13:12:33	BIC	14194.938
Sample:	0	HQIC	14173.773

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1.y	0.4730	0.051	9.338	0.000	0.374	0.572
ar.L2.y	-0.2645	0.015	-17.489	0.000	-0.294	-0.235
ma.L1.y	0.5224	0.052	10.089	0.000	0.421	0.624
ma.L2.y	-0.2699	0.047	-5.684	0.000	-0.363	-0.177

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	0.8943	-1.7267j	1.9446	-0.1739
AR.2	0.8943	+1.7267j	1.9446	0.1739
MA.1	-1.1867	+0.0000j	1.1867	0.5000
MA.2	3.1219	+0.0000j	3.1219	0.0000

图 17 ARMA（2，2）拟合结果

该模型估计到了正确的参数，并且真实参数包含在 95%的置信区间内。

接下来，我们模拟一个 ARMA（3，2）模型。之后，我们循环遍历 p，q 的非平凡数量的组合，将 ARMA 模型拟合我们的模拟序列。我们根据哪种模型产生最低的 AIC 来选择最佳组合。

```
aic: 14212.01026 | order: (3, 2)
```

通过拟合得到的 AIC 以及对应的模型参数，可以发现其拟合到了正确参数。

ARMA Model Results

Dep. Variable:	y	No. Observations:	5000
Model:	ARMA(2, 2)	Log Likelihood	-7076.176
Method:	mle	S.D. of innovations	0.996
Date:	Fri, 12 Jun 2020	AIC	14162.352
Time:	14:49:57	BIC	14194.938
Sample:	0	HQIC	14173.773

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1.y	0.4730	0.051	9.338	0.000	0.374	0.572
ar.L2.y	-0.2645	0.015	-17.489	0.000	-0.294	-0.235
ma.L1.y	0.5224	0.052	10.089	0.000	0.421	0.624
ma.L2.y	-0.2699	0.047	-5.684	0.000	-0.363	-0.177

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	0.8943	-1.7267j	1.9446	-0.1739
AR.2	0.8943	+1.7267j	1.9446	0.1739
MA.1	-1.1867	+0.0000j	1.1867	0.5000
MA.2	3.1219	+0.0000j	3.1219	0.0000

图 18　拟合的 ARMA（3，2）模型摘要

我们看到拟合结果输出了正确的 p、q 参数，并且正确估计了模型的参数。但是注意 MA.L1.y 的真实系数 0.5 几乎超出了 95%的置信区间。下面我们观察模型的残差。显然，这是一个白噪声过程，因此我们确信这是最好的模型。

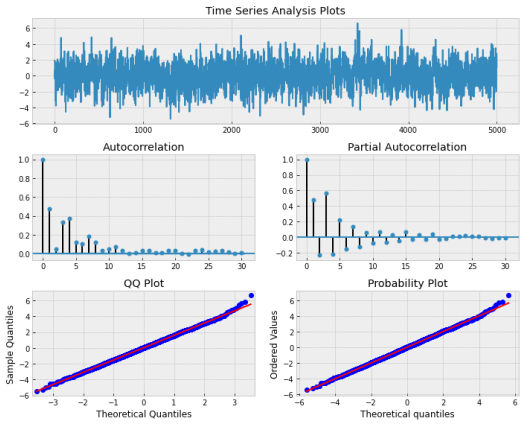

图 19 ARMA（3，2）最佳模型残留白噪声

接下来，我们将 ARMA 模型应用到 SPY 数据集（之前的章节提到过）上。根据遍历 p 和 q，我们找到了最低 AIC 下的最优参数组合：

```
aic: -11520.47028 | order: (4, 3)
```

可以得到最优参数为 4,3。我们总结模型。

ARMA Model Results

Dep. Variable:	SPY	No. Observations:	2013
Model:	ARMA(4, 3)	Log Likelihood	5768.235
Method:	mle	S.D. of innovations	0.014
Date:	Fri, 12 Jun 2020	AIC	-11520.470
Time:	14:57:19	BIC	-11475.611
Sample:	0	HQIC	-11504.004

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1.SPY	-0.6735	0.617	-1.092	0.275	-1.882	0.535
ar.L2.SPY	-0.9994	0.510	-1.960	0.050	-1.998	-0.000
ar.L3.SPY	0.0118	0.650	0.018	0.986	-1.261	1.285
ar.L4.SPY	-0.0526	0.060	-0.870	0.384	-0.171	0.066
ma.L1.SPY	0.5810	0.618	0.940	0.347	-0.631	1.793
ma.L2.SPY	0.8708	0.452	1.928	0.054	-0.015	1.756
ma.L3.SPY	-0.1262	0.597	-0.211	0.833	-1.297	1.045

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	-0.3687	-0.9387j	1.0085	-0.3096
AR.2	-0.3687	+0.9387j	1.0085	0.3096
AR.3	0.4810	-4.2977j	4.3245	-0.2323
AR.4	0.4810	+4.2977j	4.3245	0.2323
MA.1	-0.3693	-0.9492j	1.0185	-0.3091
MA.2	-0.3693	+0.9492j	1.0185	0.3091
MA.3	7.6361	-0.0000j	7.6361	-0.0000

图 20　SPY 最佳模型模型

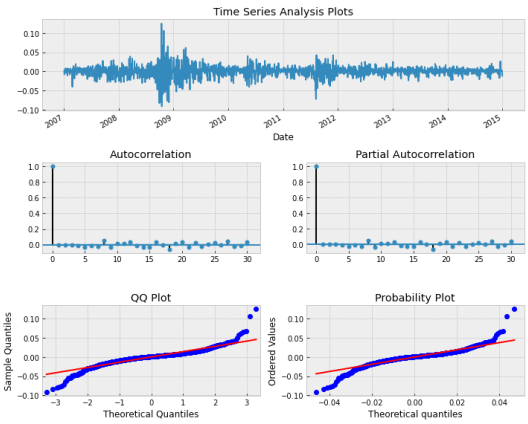

图 21　SPY 最佳模型残差

ACF 和 PACF 没有显示出显著的自相关。QQ 和概率图显示残差近似高斯分布，但呈重尾分布。然而该模型的残差看起来不像白噪声。原因可能是模型未捕获的明显的条件异方差（条件波动）的部分（2009 年和 2012 年）。

自回归综合移动平均模型 ARIMA（p，d，q）

ARIMA 是 ARMA 模型类别的自然扩展。如前所述，许多时间序列不是平稳的，但是可以通过差分使它们平稳。当我们对高斯随机游走进行一阶差分并证明它等于白噪声时，即是完成了这一过程。换句话说，我们进行了非平稳随机游走，并通过一阶差分将其转换为平稳的白噪声。

在下面的示例中，我们遍历（p，d，q）阶的平凡组合，以找到最佳的 ARIMA 模型以适合 SPY 数据。我们使用 AIC 评估每个模型。我们得到最优参数。

```
aic: -11520.47028 | order: (4, 0, 3)
```

最优模型的差分项为 0。下面，我绘制了模型残差。结果基本上与我们上面拟合的 ARMA（4，3）模型相同。显然，这个 ARIMA 模型也没有解释该序列的条件波动率。

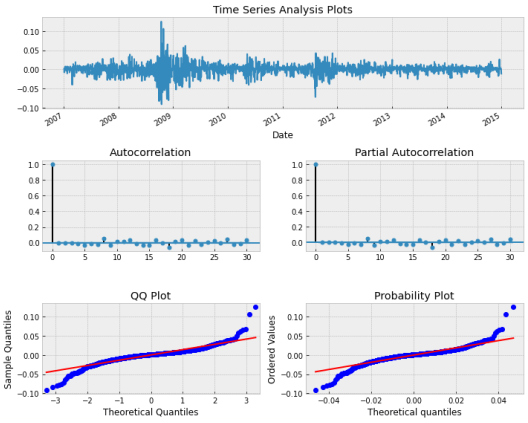

图 22 SPY 最佳模型残差

现在，我们可以对未来的数据进行简单的预测。除了预测准确值，我们还进行一个区间预测。

	forecast	lower_ci_95	lower_ci_99	upper_ci_95	upper_ci_99
2014-12-31	0.001367	-0.025642	-0.034129	0.028376	0.036863
2015-01-01	-0.000068	-0.027192	-0.035715	0.027056	0.035579
2015-01-02	0.000230	-0.026954	-0.035495	0.027413	0.035954
2015-01-03	0.000454	-0.026736	-0.035280	0.027644	0.036187
2015-01-04	-0.000608	-0.027798	-0.036342	0.026582	0.035126

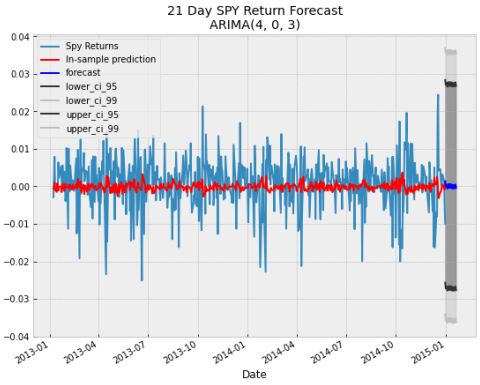

图 23 SPY 数据预测

自回归条件异方差模型 ARCH（p）

ARCH（p）模型可以简单地看作是应用于时间序列异方差的 AR（p）模型。也可以看做，我们在时间 t 处的方差取决于先前方差。以下为 ARCH(1)模型。

$$Var(y_t|y_{t-1}) = \sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2$$

我们可以将模型表示为：

$$y_t = \sigma_t \epsilon_t, \text{ with } \sigma_t = \sqrt{\alpha_0 + \alpha_1 y_{t-1}^2}, \text{ and } \epsilon_t \sim iid(0,1)$$

根据上式，我们可以模拟 ARCH(1)模型：

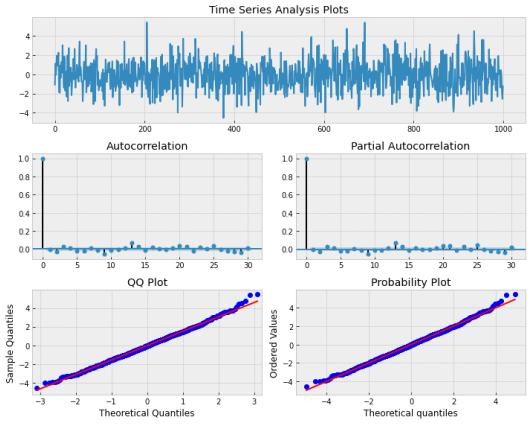

图 24 模拟 ARCH（1）过程

广义自回归条件异方差模型 GARCH（p，q）

简而言之，GARCH（p，q）是应用于时间序列异方差的 ARMA 模型，即它具有自回归项和移动平均项。AR（p）对残差的方差（误差平方）建模。基本的 GARCH（1，1）公式为：

$$\epsilon_t = \sigma_t w_t$$
$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

w 是白噪声，而 alpha 和 beta 是模型的参数。而且 alpha_1 + beta_1 必须小于 1，否则模型不稳定。我们可以在下面模拟 GARCH（1，1）过程，参数为 0.2,0.5 和 0.3。

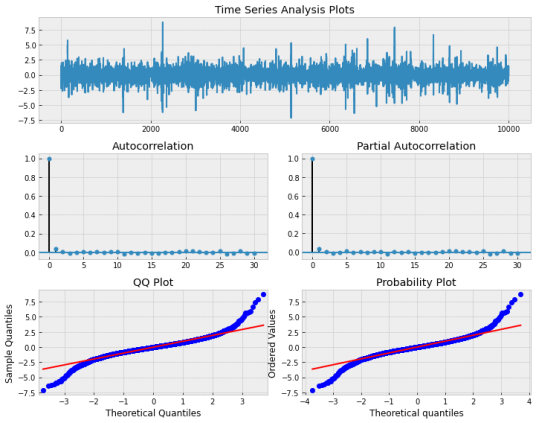

图 25 模拟 GARCH（1，1）过程

接下来，我们看看是否可以使用 GARCH（1，1）模型对以上序列进行参数估计。在这里，我们利用 ARCH 包中的 arch_model 函数。经过试验发现估计的参数值均落入置信区间内。

```
Iteration:      5,   Func. Count:     38,   Neg. LLF: 12311.793683614378
Iteration:     10,   Func. Count:     71,   Neg. LLF: 12238.592658753043
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 12237.30326731947
            Iterations: 13
            Function evaluations: 89
            Gradient evaluations: 13
                  Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:                      y   R-squared:                      -0.000
Mean Model:             Constant Mean   Adj. R-squared:                 -0.000
Vol Model:                      GARCH   Log-Likelihood:               -12237.3
Distribution:                  Normal   AIC:                           24482.6
Method:            Maximum Likelihood   BIC:                           24511.4
                                        No. Observations:                10000
Date:                Fri, Jun 12 2020   Df Residuals:                    9996
Time:                        14:55:22   Df Model:                           4
                            Mean Model
==============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
------------------------------------------------------------------------------
mu         -6.7225e-03  6.735e-03     -0.998      0.318 [-1.992e-02,6.478e-03]
                          Volatility Model
==============================================================================
                 coef    std err          t      P>|t|  95.0% Conf. Int.
------------------------------------------------------------------------------
omega          0.2021  1.043e-02     19.383  1.084e-83 [  0.182,  0.223]
alpha[1]       0.5162  2.016e-02     25.611 1.144e-144 [  0.477,  0.556]
beta[1]        0.2879  1.870e-02     15.395  1.781e-53 [  0.251,  0.325]
==============================================================================

Covariance estimator: robust
```

图 26 GARCH 模型拟合摘要

现在我们使用 SPY 数据做一个示例。流程如下：

1. 通过 ARIMA（p，d，q）模型的组合进行迭代，选择最优模型。
2. 根据具有最低 AIC 的 ARIMA 模型选择 GARCH 模型。
3. 将 GARCH（p，q）模型拟合到我们的 SPY 数据集。
4. 检查模型残差和平方残差以进行自相关。

首先，得到最优参数：

```
aic: -5255.50660 | order: (3, 0, 2)
```

模型残差：

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	2.5765e-03	9.672e-05	26.640	2.327e-156	[2.387e-03, 2.766e-03]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	0.0000	3.794e-06	0.000	1.000	[-7.437e-06, 7.437e-06]
alpha[1]	0.9998	8.765e-04	1140.591	0.000	[0.998, 1.001]
alpha[2]	3.6372e-05	2.946	1.235e-05	1.000	[-5.773, 5.773]
alpha[3]	1.8346e-05	1.746	1.051e-05	1.000	[-3.422, 3.422]
alpha[4]	1.7942e-05	2.254	7.960e-06	1.000	[-4.418, 4.418]
beta[1]	3.3583e-05	2.947	1.140e-05	1.000	[-5.775, 5.775]
beta[2]	1.7564e-05	1.748	1.005e-05	1.000	[-3.425, 3.425]
beta[3]	1.7188e-05	2.255	7.623e-06	1.000	[-4.419, 4.419]

Distribution

	coef	std err	t	P>|t|	95.0% Conf. Int.
nu	45.5025	274.011	0.166	0.868	[-4.915e+02, 5.826e+02]

Covariance estimator: robust

图 27 GARCH 模型

其对应的 GARCH 模型。我们可以看出，下方图形类似于白噪声，经过检查 ACF 和 PACF 可以看出，我们已经实现了良好的模型你和，由于平方残差中没有明显的正相关。

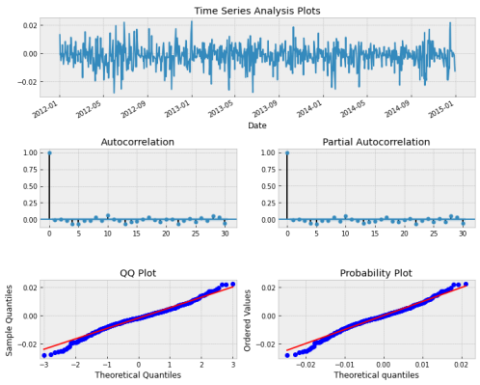

图 28 GARCH（3，2）模型残差

时间序列分析预测以 SARIMA 为例

数据集

在这里，我们采取了某城市电力负荷月度数据集，并将其划分为训练集和测试集。

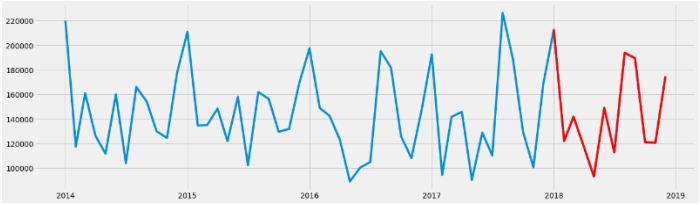

图 29 数据集

求解最优参数

在处理季节性影响时，我们利用季节性 ARIMA，表示为 ARIMA(p,d,q)(P,D,Q)s。这里，(p, d, q)是上述非季节性参数，而(P, D, Q)适用于时间序列的季节分量。s 是时间序列的周期（季度为 4，年度为 12 等等）。由于涉及诸多参数，所以我们采用枚举法求得最优的模型参数（在 R 语言中，实现了模型的自动定阶，但并未移植到 Python 中，所以这里我们手动求解）。

我们使用“网格搜索”来迭代地探索参数的不同组合。对于参数的每个组合，我们拟合一个新的季节性 ARIMA 模型，并评估其整体质量。

通过训练求解出最优参数：AIC 最小时：ARIMA(1, 1, 1)x(1, 1, 0, 12)12 - AIC:515.925486295917

训练过程：

```
ARIMA(1, 0, 1)x(0, 0, 1, 12)12 - AIC:40159.484179181
ARIMA(1, 0, 1)x(0, 1, 0, 12)12 - AIC:778.8224319925813
ARIMA(1, 0, 1)x(1, 0, 0, 12)12 - AIC:825.3497808222438
ARIMA(1, 0, 1)x(1, 0, 1, 12)12 - AIC:16779.754010292527
ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:535.2831508085185
ARIMA(1, 1, 0)x(0, 0, 0, 12)12 - AIC:1115.078702503965
ARIMA(1, 1, 0)x(0, 0, 1, 12)12 - AIC:15357.591048616969
ARIMA(1, 1, 0)x(0, 1, 0, 12)12 - AIC:788.5252169551662
ARIMA(1, 1, 0)x(1, 0, 0, 12)12 - AIC:790.5223007266907
ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:12050.212155163801
ARIMA(1, 1, 0)x(1, 1, 0, 12)12 - AIC:517.5534960386747
ARIMA(1, 1, 1)x(0, 0, 0, 12)12 - AIC:1083.9691175643518
ARIMA(1, 1, 1)x(0, 0, 1, 12)12 - AIC:45302.59377845227
ARIMA(1, 1, 1)x(0, 1, 0, 12)12 - AIC:760.0380196582735
ARIMA(1, 1, 1)x(1, 0, 0, 12)12 - AIC:800.6115102534033
ARIMA(1, 1, 1)x(1, 0, 1, 12)12 - AIC:36792.047471260645
ARIMA(1, 1, 1)x(1, 1, 0, 12)12 - AIC:515.9254862959178
```

图 30 网格搜索

模型检验

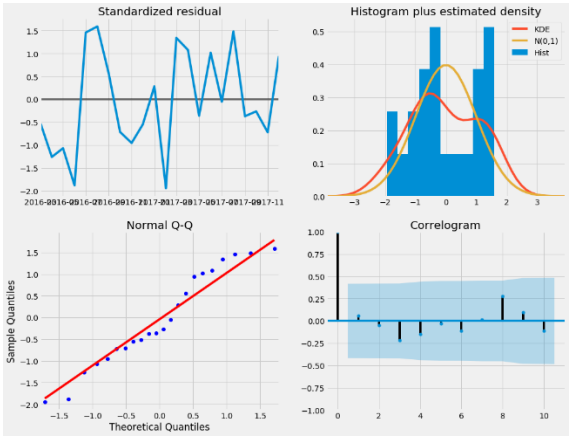

图 31 模型检验

残差近似为白噪声序列，QQ 图显示近似服从正态分布，模型检验通过。

模型预测

在模型预测方面。我们先在训练集上进行预测，并检查其误差。预测 2017 年的月度数据：

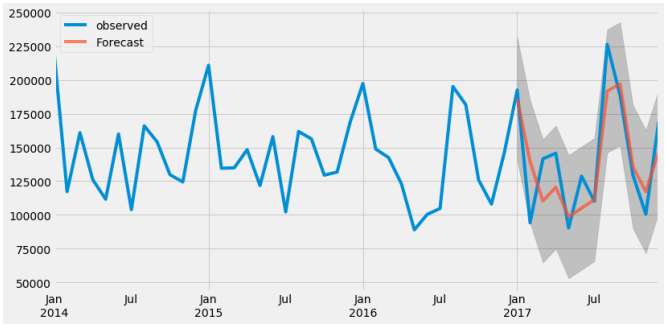

图 31 在训练集上预测

得到误差 RMSE：23113.32，MAPE:14.5%。其预测效果不错，于是我们在测试集上预测 2018 年的月度数据：

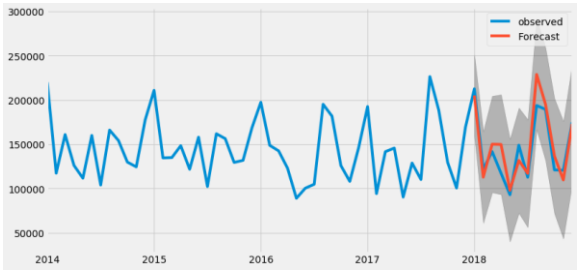

图 31 在测试集上预测

在测试集上得到误差 RMSE：16424.76，MAPE:9.13%。可以看出，SARIMA 获得了非常好的效果。

代码

本文的代码均可以在我的 Github 上获取：

其他章节：

https://github.com/stxupengyu/time-series-analysis/blob/master/time_series_analysis_prediction.ipynb

SARIMA 章节：

https://github.com/stxupengyu/SARIMA/blob/master/ts-predict-lord.ipynb

参考文献

http://iacs-courses.seas.harvard.edu/courses/am207/blog/lecture-17.html

http://www.seanabu.com/2016/03/22/time-series-seasonal-ARIMA-model-in-python/

https://zhuanlan.zhihu.com/p/35128342

https://www.cnblogs.com/foley/p/5582358.html

https://blog.csdn.net/qifeidemumu/article/details/88782550