13 GARCH 估计日方差率

使用申明*

2021 年 2 月 26 日

目录

1 简介

广义自回归条件异方差 (Generalized Auto-Regressive Heteroscedasticity, GARCH) 模型是由 Bollerslev 在 1986 年提出的一种估计市场变量日方差率的模型。对于广义的 GARCH(p,q) 模型，σ_n^2 是由最近的 p 个 u_i^2 观察值，最近的 q 个日方差率 σ_i^2 和一个常数项组成。这里我们只考虑 GARCH(1,1) 模型，并简记为 GARCH，其中日方差率的表示为

$$\sigma_n^2 = \omega + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2, \quad \alpha + \beta < 1 \, . \tag{1}$$

相较于 EWMA 模型日方差率的更新估计只涉及一个待定参数 λ，GARCH 模型中有 3 个待定参数。将 ω 表示为 $V_L(1-\alpha-\beta)$ 后，我们可以看到 GARCH 模型对日方差率的估计实际上是用前一天的 u_i^2, σ_i^2 和长期方差 V_L 加上不同权重后求和得出。

考虑我们有市场变量从第 0 天到第 N 天每天末的数值为 $S_0, S_1, ..., S_N$，

$$u_n = \frac{S_n - S_{n-1}}{S_{n-1}}, \quad , u_1 = \frac{S_1 - S_0}{S_0}, \quad u_0 = 0 \, . \tag{2}$$

$$\sigma_n^2 = \omega + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2, \quad \sigma_2^2 = u_1^2, \quad \sigma_1^2 = \sigma_0^2 = 0 \, . \tag{3}$$

$$\omega, \alpha, \beta = \mathrm{Arg}\min_{\omega,\alpha,\beta} \, Loss(\omega, \alpha, \beta), \quad Loss(\omega, \alpha, \beta) = \sum_{i=2}^{N} \left(\ln \sigma_i^2 + \frac{u_i^2}{\sigma_i^2} \right) \, . \tag{4}$$

模型中最佳参数 ω, α, β 的选取同 EWMA 模型选取最佳 λ 的方法相似。选择的 ω, α, β 应该使上面的 $Loss(\omega, \alpha, \beta)$ 达到极小。这里我们可以用两步穷举来计算这些参数，先将 V_L 设为 $\frac{1}{N}\sum_{i=1}^{N} u_i^2$，$\omega = V_L(1-\alpha-\beta)$，将参数数量简化为两个，找出最佳的 α_0, β_0。然后在 $\omega_0 = (1-\alpha_0-\beta_0)$, α_0, β_0 附近进一步细分穷举找出最佳参数 ω_1, α_1, β_1。

2 GARCH 估计日方差率步骤

1. 由 $S_0, S_1, ..., S_N$ 计算出 $u_0^2, u_1^2, ..., u_N^2$, $u_n = \frac{S_n - S_{n-1}}{S_{n-1}}$, $u_0 = 0$, $u_1 = \frac{S_1 - S_0}{S_0}$.

2. 写一个计算 $Loss$ 的函数，输入 $\{u_0^2, u_1^2, ..., u_N^2\}$, ω, α, β，输出 $Loss$ 数值。由于 $Loss(\omega, \alpha, \beta) = \sum_{i=2}^{N} \left(\ln \sigma_i^2 + \frac{u_i^2}{\sigma_i^2} \right)$, $\sigma_i^2 = \omega + \alpha u_{i-1}^2 + \beta \sigma_{i-1}^2$，计算时不需要保存所有的 σ_i^2，只需要不断更新并加入到总的 $Loss$。

3. 计算出 $V_L = \frac{1}{N} \sum_{i=1}^{N} u_i^2$，让 α 在 [0, 0.5]，β 在 [0.5, 1] 区间各自均匀地取 M1 个点，且要求 $\alpha + \beta < 1$。然后 $\omega = V_L(1 - \alpha - \beta)$，依次计算出所有的 α 和 β 下的 $Loss$。记下当 $Loss$ 取极小时的 α_1，β_1 和 $\omega_1 = V_L(1 - \alpha_1 - \beta_1)$。

4. 让 α, β, ω 在 $[\alpha_1 - 0.025, \alpha_1 + 0.025]$, $[\beta_1 - 0.025, \beta_1 + 0.025]$, $[0.5\omega_1, 1.5\omega_1]$ 区间内各自均匀地取 M2 个点。依次计算出所有取值下的 $Loss$。让使得 $Loss$ 取极小值时的 α, β, ω 为我们计算出的 GARCH 模型最佳参数。

5. 使用上面计算出的最佳 α, β, ω，由 GARCH 模型 $\sigma_n^2 = \omega + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2$ 计算出 $S_0, S_1, ..., S_N$ 对应的日方差率估计值 $\sigma_0^2 = 0$, $\sigma_1^2 = 0$, $\sigma_2^2 = u_1^2, ..., \sigma_{N+1}^2$。

3 步骤 Python 代码实现

```
import numpy as np

def get_loss(U2, omega, alpha, beta):
    sigma2 = U2[1]
    loss = 0
    for i in range(2, len(U2)):
        loss += np.log(sigma2)+U2[i]/sigma2
        sigma2 = omega+alpha*U2[i]+beta*sigma2
    return loss

def GARCH_optimal_parameters(data, M1=80, M2=30):
    N = len(data)-1     # data: S_0, S_1, ..., S_N .
    U2 = [0]*(N+1)

    for i in range(1, N+1):
        U2[i] = (data[i]-data[i-1])/data[i-1]
        U2[i] = U2[i]*U2[i]

    VL = np.average(U2[1:])
    min_loss = float("inf")
    loss = None
    opt1_omega = None
    opt1_alpha = None
    opt1_beta = None
    for i in range(M1):
        beta = 0.5+i*0.5/M1
        for j in range(M1):
            alpha = 0.01+j*0.5/M1
            if alpha+beta >= 1:
                continue
            omega = VL*(1-alpha-beta)
            loss = get_loss(U2, omega, alpha, beta)
            if loss < min_loss:
                min_loss = loss
                opt1_omega = omega
                opt1_alpha = alpha
                opt1_beta = beta
```

```
    print("Step1: \n VL= ",VL)
    print("Optimal alpha, beta = ", opt1_alpha, opt1_beta)
    print("Omega = VL(1-alpha-beta)= ", opt1_omega)
    print("Total loss:  ", min_loss)
    print("\n")

    min_loss = float("inf")
    loss  = None
    opt2_omega = None
    opt2_alpha = None
    opt2_beta = None
    for i in range(M2):
        beta = opt1_beta-0.025+0.05*i/M2
        for j in range(M2):
            alpha = opt1_alpha-0.025+0.05*j/M2
            if alpha+beta>=1:
                continue
            for k in range(M2):
                omega = 0.5*opt1_omega+k*opt1_omega/M2
                loss  = get_loss(U2, omega, alpha, beta)
                if loss < min_loss:
                    min_loss = loss
                    opt2_omega = omega
                    opt2_alpha = alpha
                    opt2_beta = beta
    print("Step 2/ Final result:  ")
    print("Optimal omega, alpha, beta = ", opt2_omega, opt2_alpha, opt2_beta)
    print("Total loss:  ", min_loss)

    return opt2_omega, opt2_alpha, opt2_beta

def GARCH_predict(data, omega, alpha, beta):
    N = len(data)-1      # data: S_0, S_1, ..., S_N
    U2 = None
    variances = [0.0, (data[1]-data[0])**2/data[0]/data[0]]
    for i in range(2, N+1):
        U2 = (data[i]-data[i-1])/data[i-1]
        U2 = U2*U2
        variances.append(omega+alpha*U2+beta*variances[-1])

    return variances
```

4 计算示例

我们使用 John Hull 网站上的 GARCH 示例数据 ($http://www2.rotman.utoronto.ca/\sim hull/data/GARCHCALCSS\&P500.xls$)。

如果我们用 EWMA 模型，可以得到最佳的 $\lambda = 0.937$，对应的 $Loss = -10192.50707$。如下，我们使用 GARCH 模型的话，会计算得到最佳参数为 $\omega = 1.4060 \times 10^{-6}$，$\alpha = 0.08417$，$\beta = 0.90875$，对应的 $Loss = -10228.21197$。可见从最大似然估计的角度看，GARCH 模型给出的结果要好于 EWMA。

```
if __name__ =="__main__":
```

```
data = np.genfromtxt("GARCHCALCSS&P500.txt", skip_header=1, usecols=(1))
omega, alpha, beta = GARCH_optimal_parameters(data, 80, 30)
# variances = GARCH_predict(data, omega, alpha, beta)
```

Output:

Step 1:

Optimal alpha, beta = 0.975, 0.89375

Omega = VL(1-alpha-beta) = 2.109 x 10^-6

Total loss = -10225.05837

Step 2/ Final result:

Optimal omega, alpha, beta = 1.406 x 10^-9, 0.0841667, 0.90875

Total loss: -10228.21197

5 参考资料

参考文献

[1]《期权、期货及其他衍生产品》(原书第 9 版)第 23 章，John C. Hull 著，王勇、索吾林译，机械工业出版社，2014.11 。