《Principles of Financial Engineering》 Project

熊雄

20234213001

1 数据选取

从https://www.investing.com/etfs/chinaamc-china-50获取上证50ETF（代码：510050）最近80个交易日的数据，数据的日期从 2023.8.11 ~ 2023.12.8。部分原始数据显示如下：

Date	Price	Open	High	Low	Vol.	Change %
12/08/2023	2.31	2.31	2.32	2.30	578.43M	0.30%
12/07/2023	2.31	2.31	2.32	2.29	845.84M	-0.43%
12/06/2023	2.32	2.32	2.33	2.31	724.29M	0.00%
12/05/2023	2.32	2.36	2.36	2.32	966.83M	-2.03%
12/04/2023	2.37	2.39	2.39	2.37	602.13M	-0.75%
⋮	⋮	⋮	⋮	⋮	⋮	⋮
08/15/2023	2.62	2.62	2.63	2.6	749.61M	0.00%
08/14/2023	2.62	2.62	2.63	2.6	701.91M	-0.76%
08/11/2023	2.64	2.71	2.71	2.64	1.32B	-2.37%

2 估计历史波动率

2.1 模型选取

实际中，方差值常常会被拉到长期平均值水平，这种现象被称为均值回归。GARCH(1,1)模型有均值回归的特性，而 EWMA 没有均值回归特性，从理论上讲，GARCH(1,1)比 EWMA 更具有诱人之处。

在本文，我们采用GARCH(1,1)方法来估计波动率，其中GARCH(1,1)模型的形式如下：

$$\sigma_n^2 = \gamma V_L + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2, \tag{1}$$

其中V_L为长期平均方差，σ_n表示第n天的波动率估计值（在第$n-1$天末估计），u_{n-1}表示最近一天内变化的百分比，$\gamma, \alpha, \beta \in (0,1)$为常数，满足$\alpha + \beta + \gamma = 1$。

我们将GARCH(1,1)模型写成

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

其中$\omega = \gamma V_L$。因为在估计模型的参数的时候，一旦估计出ω, α和β后，我们可以由$\gamma = 1 - \alpha - \beta$来计算$\gamma$，而长期方差$V_L = \omega / \gamma$。

2.2 估计参数

我们利用极大似然方法（MLE）估计GARCH(1,1)的参数，定义$v_i := \sigma_i^2$为第i天的方差估计值，假设u_i在方差上的条件概率分布为正态分布，参数的最佳估计应该使得下式最大化

$$\prod_{i=1}^{m}\left[\frac{1}{\sqrt{2\pi v_i}}e^{-\frac{u_i^2}{2v_i}}\right], \tag{3}$$

取对数后，式(3)最大化与以下表达式最大化等价

$$\sum_{i=1}^{m}\left[-\ln v_i - \frac{u_i^2}{v_i}\right], \tag{4}$$

可以采用迭代法来求取使得式(4)达到最大化的解。

利用Stata可以很快估计GARCH(1,1)参数，具体do文档代码见附录5.1，得到输出如下

```
ARCH family regression

Sample: 1 thru 80                               Number of obs     =         80
                                                Wald chi2(.)      =          .
Log likelihood = 267.9723                       Prob > chi2       =          .

------------------------------------------------------------------------------
             |                 OPG
      Change | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Change       |
       _cons |  -.0019808   .0009875    -2.01   0.045    -.0039164   -.0000453
-------------+----------------------------------------------------------------
ARCH         |
        arch |
         L1. |  -.0746986   .0793357    -0.94   0.346    -.2301937    .0807965
             |
       garch |
         L1. |  -.7473422    .431669    -1.73   0.083    -1.593398    .0987135
             |
       _cons |   .0001399   .0000385     3.63   0.000     .0000644    .0002154
------------------------------------------------------------------------------
```

即模型收敛，具体分析如下

1. Change部分：这部分提供了ARCH族模型中Change变量的系数估计结果。
    - _cons：表示Change变量的截距项，估计值为-0.0019808，标准误差为0.0009875。截距项的p值为0.045，可能接近显著性水平。
2. ARCH部分：这部分提供了ARCH模型中的系数估计结果。
    - arch L1：表示ARCH模型中的滞后项，即ARCH(1)。系数估计值为-0.0746986，标准误差为0.0793357，p值为0.346，不显著。
3. GARCH部分：这部分提供了GARCH模型中的系数估计结果。

- garch L1：表示GARCH模型中的滞后项，即GARCH(1)。系数估计值为-0.7473422，标准误差为0.431669，p值为0.083，可能接近显著性水平。
- _cons：表示GARCH模型中的截距项，估计值为0.0001399，标准误差为0.0000385，p值小于0.001，显著。

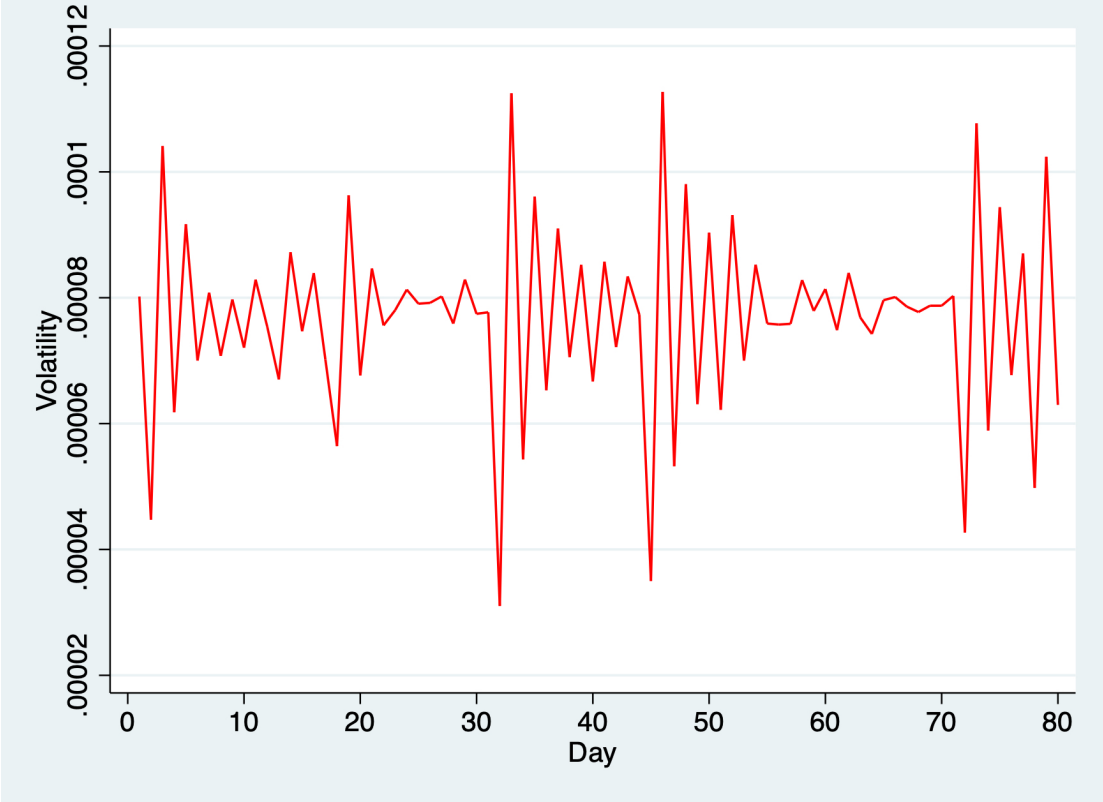
于是GARCH(1,1)模型的参数估计为

$$\omega \approx 0.0001399, \quad \alpha \approx -0746986, \quad \beta \approx -7473422. \tag{5}$$

$$\gamma = 1 - \alpha - \beta = 0.8220408, \quad V_L = \omega/\gamma \approx 0.0001702. \tag{6}$$

2.3 利用Stata估计波动率

具体do文档代码见附录5.1，绘制图形如下：



3 计算美式看跌期权的价格

3.1 二叉树方法

二叉树方法既可用于欧式期权定价，也可用于美式期权定价。我们首先将期权的期限分成许多长度为Δt的很小的时间区间，并且假定在每一个时间区间里股票价格从开始的S变成两个新价格$uS(u > 1)$和$dS(d < 1)$中的一个。价格上涨的概率记为p，价格下跌的概率记为$1 - p$。

3.1.1 参数

敲定价格 $K = S_0 = 2.31$。在Stata估计中，最后一天估计的波动率为0.000063，因此利用二叉树来计算期权价格时的年波动率时采用该估计值。在本文中，采用十年期中债国债收益率2.677%作为无风险利率r的估计。假设期权到期时间为一个月，即$T = 1/12$。

3.1.2 计算步骤

1. 计算风险中性概率p^*，上涨和下跌的倍数u和d:

$$p^* = \frac{e^{(r-q)\Delta t - d}}{u-d}, \quad u = e^{\sigma\sqrt{\Delta t}}, \quad d = e^{-\sigma\sqrt{\Delta t}}. \tag{7}$$

2. 产生所要求的二叉树，在每个点上比较执行与不执行获得的收益的最大值。

3. 利用风险中性概率和贴现率将第二步的结果在每棵二叉树上向前贴现得到最终的价格。

我们利用Matlab实现该过程，具体函数代码见附录5.2，

3.2 计算结果

我们利用Matlab实现，具体代码见附录5.3。最后计算美式看跌期权价格为 **0.023631（元）** 。

4 利用 Delta 对冲进行风险管理

4.1 生成Delta矩阵

期权的Delta为期权价格与标的股票价格变化的比率，即$\Delta f/\Delta S$，其中ΔS为股票价格的微小变化，Δf为相应的期权价格的微小变化。在Δt时刻，当股票价格为S_0u的时候，期权价格的估计值为$f_{1,1}$；当股票价格为S_0d的时候，期权价格的估计值为$f_{1,0}$，这说明当$\Delta S = S_0u - S_0d$时，$\Delta f = f_{1,1} - f_{1,0}$，因此$\Delta t$在$\Delta$近似值为
$\Delta = (f_{1,1} - f_{1,0})/(S_0u - S_0d)$.

我们利用Matlab可以生成Delta矩阵如下

0	-0.3874	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Δt	-0.4771	-0.3008	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$2\Delta t$	-0.5735	-0.3841	-0.2205	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$3\Delta t$	-0.6721	-0.4784	-0.2931	-0.1505	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$4\Delta t$	-0.7672	-0.5803	-0.3800	-0.2091	-0.0939	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$5\Delta t$	-0.8529	-0.6845	-0.4798	-0.2838	-0.1371	-0.0522	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$6\Delta t$	-0.9241	-0.7842	-0.5883	-0.3750	-0.1957	-0.0805	-0.0249	0	0	0	0	0	0	0	0	0	0	0	0	0
$7\Delta t$	-0.9779	-0.8721	-0.6993	-0.4813	-0.2725	-0.1216	-0.0408	-0.0096	0	0	0	0	0	0	0	0	0	0	0	0
$8\Delta t$	-1.0000	-0.9424	-0.8043	-0.5980	-0.3687	-0.1796	-0.0657	-0.0168	-0.0027	0	0	0	0	0	0	0	0	0	0	0
$9\Delta t$	-1.0000	-0.9888	-0.8947	-0.7171	-0.4830	-0.2583	-0.1037	-0.0290	-0.0050	-0.0004	0	0	0	0	0	0	0	0	0	0
$10\Delta t$	-1.0000	-1.0000	-0.9639	-0.8279	-0.6102	-0.3603	-0.1599	-0.0494	-0.0094	-0.0008	0	0	0	0	0	0	0	0	0	0
$11\Delta t$	-1.0000	-1.0000	-0.9980	-0.9189	-0.7400	-0.4849	-0.2400	-0.0825	-0.0174	-0.0017	0	0	0	0	0	0	0	0	0	0
$12\Delta t$	-1.0000	-1.0000	-1.0000	-0.9819	-0.8582	-0.6258	-0.3489	-0.1350	-0.0319	-0.0034	0	0	0	0	0	0	0	0	0	0
$13\Delta t$	-1.0000	-1.0000	-1.0000	-1.0000	-0.9503	-0.7694	-0.4873	-0.2152	-0.0577	-0.0070	0	0	0	0	0	0	0	0	0	0
$14\Delta t$	-1.0000	-1.0000	-1.0000	-1.0000	-0.9968	-0.8946	-0.6485	-0.3318	-0.1027	-0.0142	0	0	0	0	0	0	0	0	0	0
$15\Delta t$	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-0.9797	-0.8125	-0.4901	-0.1791	-0.0290	0	0	0	0	0	0	0	0	0	0
$16\Delta t$	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-0.9462	-0.6835	-0.3035	-0.0590	0	0	0	0	0	0	0	0	0	0
$17\Delta t$	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-0.8805	-0.4935	-0.1201	0	0	0	0	0	0	0	0	0	0
$18\Delta t$	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-0.7514	-0.2446	0	0	0	0	0	0	0	0	0	0
$19\Delta t$	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-0.4980	0	0	0	0	0	0	0	0	0	0

4.2 不提前执行，$S_T > K$

举例说明：考虑全是上涨的情况下的路径。由于价格一直上涨，因此期权最后处于深度虚值的状态，Delta 接近于 0，对Delta矩阵的每一个点，例如Δt时的 Delta 约为-0.3874，因此应该买入 0.3874 份50ETF进行对冲；$2\Delta t$ 时的 Delta约为-0.3008，Delta值变小， 因此应该卖出$0.3874 - 0.3008 = 0.0866$份50ETF进行对冲；以此类推。

4.3 不提前执行，$S_T < K$

举例说明：考虑全是下跌的情况下的路径。由于价格一直下跌，因此期权最后处于深度实值的状态，Delta 接近于 0，对Delta矩阵的每一个点，例如Δt时的 Delta 约为-0.3874，因此应该买入 0.3874 份50ETF进行对冲；$2\Delta t$ 时的 Delta约为-0.4771，Delta值变大， 因此应该买入$0.4771 - 0.3874 = 0.0897$份50ETF进行对冲；以此类推。

4.4 在到期日之前提前执行

对任意样本路径，在满足执行期权可以获得正的收益时执行该期权即可。

5 附录

5.1 利用Stata实现的估计模型参数和波动率与绘图

```
clear
cd "/Users/xiong/Library/Mobile Documents/com~apple~CloudDocs/研一课程/金融工程原理/project"
import excel data_clear.xls, firstrow clear
save data_clear.dta, replace
use data_clear.dta
keep if i <= 80
replace i = 81 - i
replace Change = -.0085 in 71 //分红
tsset i
arch Change, arch(1) garch(1)
predict h, variance
graph twoway line  h i, xlabel(0(10)80)  xtitle("Day") ytitle("Volatility") lc(red)
```

5.2 利用Matlab实现的二叉树方法定价函数

```
function [option_price, sample_paths] = american_option_pricing(S, K, r, sigma, T, N, sample_paths)
% 计算二叉树参数
    dt = T / N;
    u = exp(sigma * sqrt(dt));
    d = 1 / u;
    p = (exp(r * dt) - d) / (u - d);

% 构建二叉树
    tree = zeros(N+1);
    for i = 1:N+1
        for j = 1:i
            tree(i,j) = S * (u^j) * (d^(i-j));
        end
```

```
    end

% 计算期权价值和样本路径
    option_values = zeros(N+1);
    if sample_paths
        paths = zeros(N+1);
        paths(1,1) = S;
    end

    for j = 1:N+1
        option_values(N+1,j) = max(K - tree(N+1,j) , 0);
        if sample_paths
            paths(N+1,j) = tree(N+1,j);
        end
    end

% 逐步回溯计算期权价值和样本路径
    for i = N:-1:1
        for j = 1:i
            exercise_value = max(K - tree(i,j) , 0);
            continuation_value = (p * option_values(i+1,j+1) + (1-p) *
option_values(i+1,j)) * exp(-r * dt);
            option_values(i,j) = max(exercise_value, continuation_value);
            if sample_paths
                paths(i,j) = tree(i,j);
            end
        end
    end

    if sample_paths
        option_price = option_values(1,1);
        sample_paths = paths;
    else
        option_price = option_values(1,1);
    end
end
```

5.3 利用Matlab实现的二叉树方法函数

```
clear;clc;
%% Parameters
S = 2.31;
K = 2.31;
r = 0.02677;  % rist-free
sigma = sqrt(0.000063) * sqrt(252);  % year volatility
T = 1/12;  % year
N = 20;  % step

%% Calculate
[option_price, option_values, sample_paths] = american_option_pricing(S, K, r,
sigma, T, N, true);
disp("美式看跌期权价格: " + num2str(option_price));
```

```
disp("样本路径:");
 disp(sample_paths);
```

5.4 利用Matlab实现的风险管理

```
%% Delta-Hedge
delta = zeros(N, N);

for i = 1: N
    for j = 1: i
        delta(i,j) = (option_values(i+1, j+1) - option_values(i+1, j)) / (sample_paths(i+1, j+1) - sample_paths(i+1, j));
    end
end
delta
```