



# 风险平价模型测试

汇报人：李楷

# 目录

COMPANY

01 定义及产生原因

02 实证结果

## 定义以及产生原因

### 背景



传统60/40的投资组合在资本配置上可能看起来是平衡的，但从风险配置的角度来看，它是高度集中的，股票贡献了大多数的风险。



风险平价模型（Risk Parity）试图通过将资金分配到更广泛的类别，如股票、政府债券、信贷相关证券和通胀对冲（包括实物资产、商品、房地产和通胀保值债券）等，来均衡风险。风险平价投资组合的定义是低相关性的子资产、低股票风险和被动管理。

# PART 02

## 实验过程与实 证结果

# 研究方法



单类资产j 的边际风险贡献定义如下：

$$MRC_j = \frac{\partial \sigma_p}{\partial w_j} = \frac{(V * w)_j}{\sigma_p}$$

其中，

$w_j$  表示第j个资产的权重

$V$  表示资产的协方差矩阵

$\sigma_p = \sqrt{w * V * w^T}$  表示组合风险

# 研究方法



单类资产j 的风险贡献定义如下：

$$RC_j = w * MRC_j = \frac{w_j(V * w)_j}{\sigma_p}$$

需要优化的损失函数和约束条件定义如下：

$$J(x) = (\sum_{i=1}^n \sum_{j=1}^n (w_i(V * w))_i - w_j(V * w)_j)^2$$

$$\min J(x)$$

$$s. t. \sum_i w_i = 1$$

$$1 \geq w_i \geq 0$$

# 数据处理与求解

## 1. 选取数据

使用朝阳永续EXCEL插件获取产品：宏锡2号作为商品期货资产的代表；明泓稳健增长1期作为量化多头的代表。

## 2. 统一数据频率

因为各个子基金的披露频率不同,选取复权净值序列根据披露日期的交集，求得协方差矩阵。

3. 设定样本内外分界线为2019-06-01，将样本内净值序列代入，定义风险预算权重均为0.5（等权）。

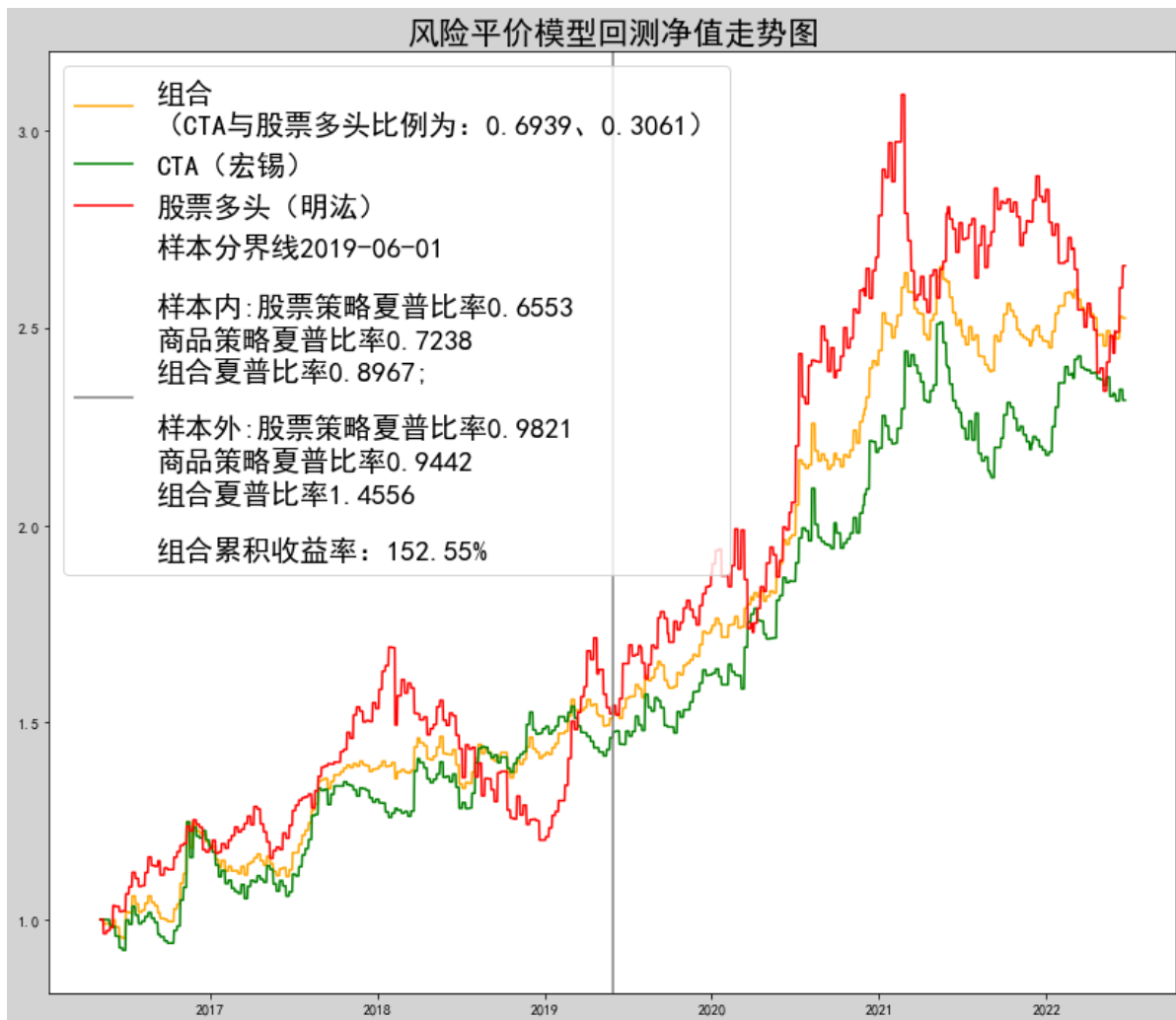
4. 使用SLSQP序列最小二乘优化器求数值解，得到新资产权重。

5.组合分样本内外回测，使用按日填充的数据进行收益回测，避免遗漏不在数据交集内的收益。



# 风险平价结果：

由于股票波动更大，在样本内得到风险平价模型的最终资产权重CTA比股票多头结果为0.6939与0.3061，并回测到2022-06-22



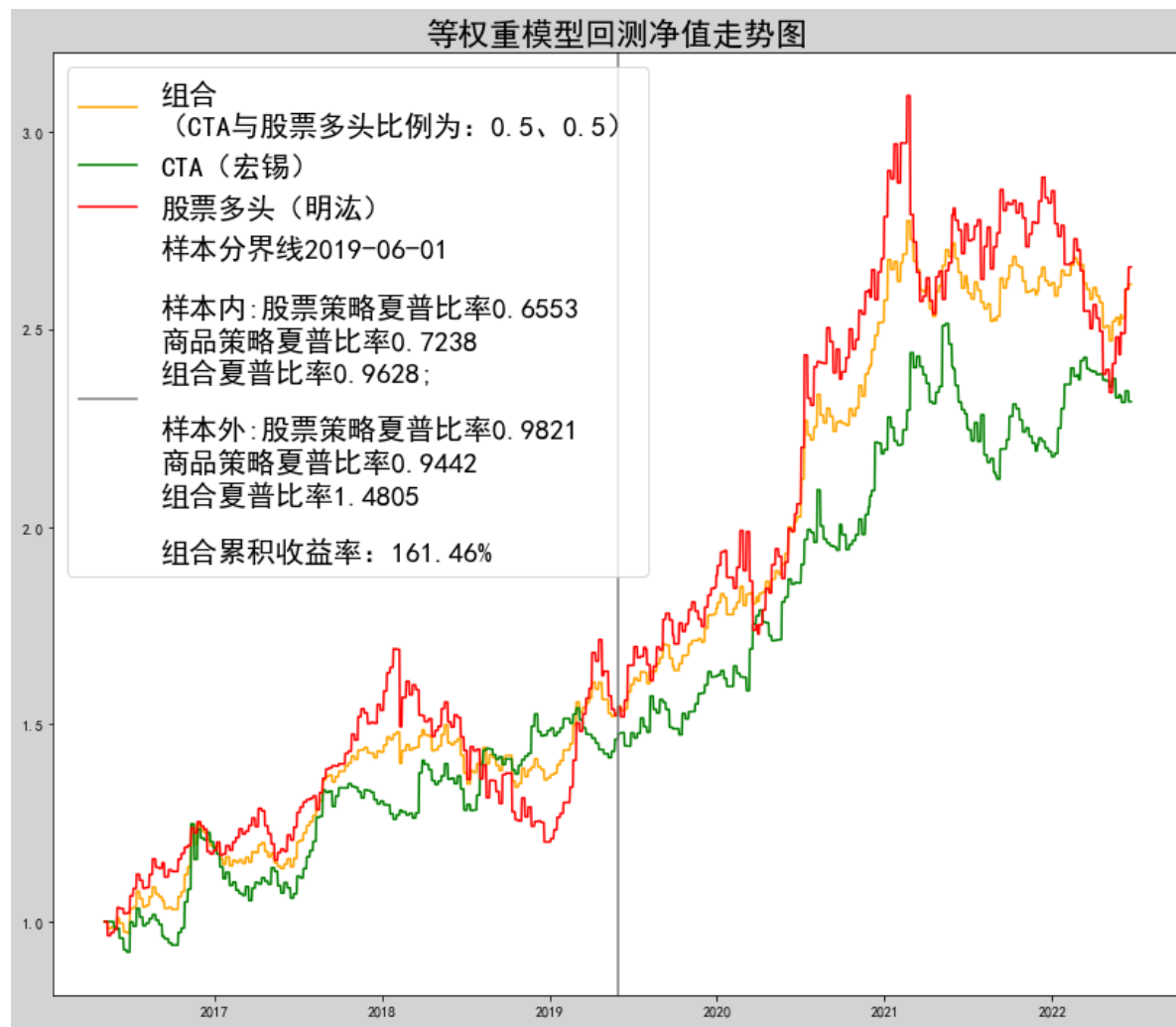
无论是在样本内还是样本外

资产组合的配置相比单一资产具有更高的夏普比率，  
主要是由于 股票策略和CTA策略的表现在某些时刻低相  
关、负相关。

(夏普比率取周复权净值变化，并按52周年化)

# 对比等权重模型结果：

可能由于选取样本的原因，无论是在样本内还是样本外，风险平价模型各项指标并没能战胜等权重模型。



### 附录：核心代码

```
def calculate_portfolio_var(w,V):  
    w=np.matrix(w)  
    return (w*V*w.T)[0,0]
```

```
def calculate_risk_contribution(w,V):  
    w=np.matrix(w)  
    sigma=np.sqrt(calculate_portfolio_var(w,V))  
    Cov_array=V*w.T  
    MRC_array=np.multiply(Cov_array,w.T)/sigma  
    return MRC_array
```

### 附录：核心代码

```
def risk_budget_objective(x,pars):  
    #  
    V=pars[0]#  
    x_t=pars[1]  
    sig_p=np.sqrt(calculate_portfolio_var(x,V))  
    risk_target=np.asmatrix(np.multiply(sig_p,x_t))  
    MRC_array=calculate_risk_contribution(x, V)  
    J=np.sum(np.square(MRC_array-risk_target.T))  
    return J  
  
def total_weight_constraint(x):  
    return np.sum(x)-1  
  
def long_only_constraint(x):  
    return x
```

### 附录：核心代码

```
def calcu_w(x,V):#x表示风险预算权重
    w0=[0.5,0.5]#,0.25,0.25]
    x_t=x#
    cons=({'type':'eq','fun':total_weight_constraint},
          {'type':'ineq','fun':long_only_constraint})
    res=minimize(risk_budget_objective,w0,args=[V,x_t],method='SLSQP',constraints=cons,options={'disp':True})
    w_rb=np.array(res.x)
    #w_rb=np.asmatrix(res.x)
    return w_rb
```

谢谢  
THANKS

