Return Entropy Portfolio Optimization

Paper

- A Maximum Entropy Model for Large-scale Portfolio Optimization

  → Publisher : IEEE

- Diversified portfolios with different entropy measures

  → Publisher : Applied Mathematics and Computation

Mean Variance Portfolio Optimization

Min : $w^T \Omega w$

s.t

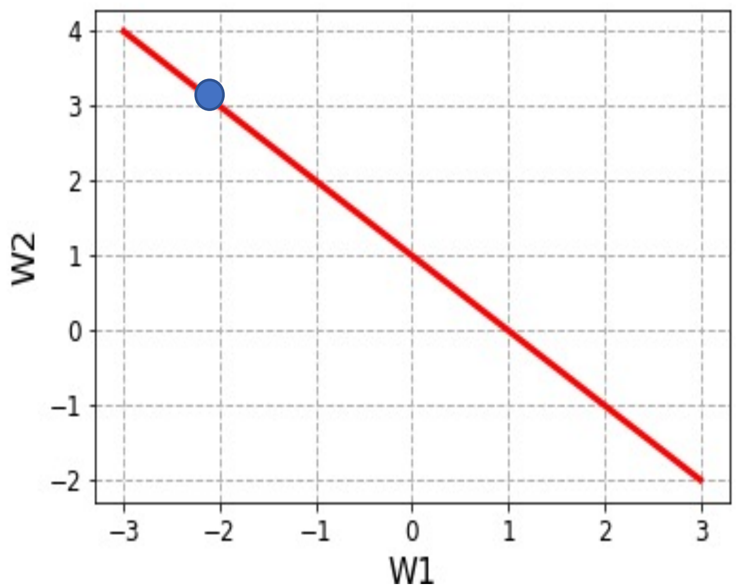
$$\begin{cases} \sum_{i=1}^{n} r_i w_i = E \quad , i = 1,2 \ldots.. n \\ \sum_{i=1}^{n} w_i = 1 \quad , i = 1,2 \ldots.. n \end{cases}$$

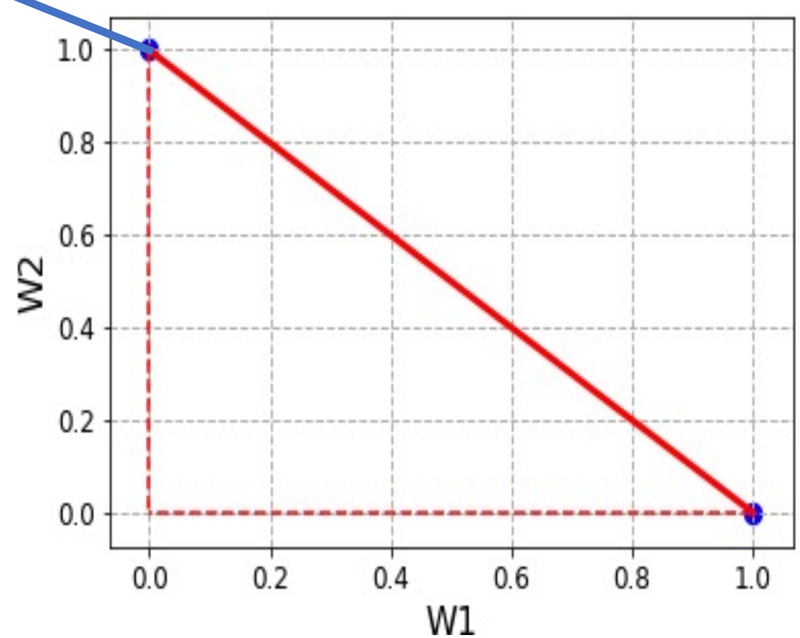
Min : $w^T \Omega w$

s.t

$$\begin{cases} \sum_{i=1}^{n} r_i w_i = E \quad , i = 1,2 \ldots.. n \\ \sum_{i=1}^{n} w_i = 1 \quad , i = 1,2 \ldots.. n \\ w_i \geq 0, \quad i = 1,2 \ldots.. n \end{cases}$$

Corner solution





Return Entropy Portfolio Optimization

Min : $w^T \Omega w$

s.t
$$\begin{cases} \sum_{i=1}^{n} r_i w_i = E \quad , i = 1,2 \ldots\ldots n \\ \sum_{i=1}^{n} w_i = 1 \quad , i = 1,2 \ldots\ldots n \\ w_i \geq 0, \quad i = 1,2 \ldots\ldots n \end{cases}$$

Min : $w^T \Omega w - \underbrace{[-\sum_{i=1}^{n} w_i * \ln(w_i)]}_{\text{Information Entropy}}$

s.t
$$\begin{cases} \sum_{i=1}^{n} r_i w_i = E \quad , i = 1,2 \ldots\ldots n \\ \sum_{i=1}^{n} w_i = 1 \quad , i = 1,2 \ldots\ldots n \end{cases}$$

Information Entropy

Example of Cost function in Machine Learning :

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h(x_i' \theta) - yi)^2 + \underbrace{Rgularization\ term}_{\text{Prevent Overfitting}}$$

Prevent Overfitting

Maximum Entropy Proof

Max : $-\sum_{i=1}^{n} w_i \ln w_i$

$s.t. \quad \sum_{i=1}^{n} w_i = 1$

$\Rightarrow$ Min : $w^T \Omega w - [-\sum_{i=1}^{n} w_i * \ln(w_i)]$

Lagrange $\Rightarrow$ $L(\mathrm{w}, \lambda) = -\sum_{i=1}^{n} w_i \ln w_i + (\lambda + 1)(\sum_{i=1}^{n} w_i - 1)$

F.O.C $\Rightarrow$

$$\begin{cases} \dfrac{\partial L}{\partial w_i} = -(\cancel{1} + \ln w_i) + (\lambda + \cancel{1}) = 0 \Rightarrow \ln w_i = \lambda \Rightarrow w_i = e^{\lambda} \\ \dfrac{\partial L}{\partial \lambda} = \sum_{i=1}^{n} w_i - 1 = 0 \Rightarrow \sum_{i=1}^{n} e^{\lambda} - 1 = 0 \Rightarrow ne^{\lambda} = 1 \end{cases}$$

$$\Rightarrow \begin{cases} w_i = e^{\lambda} \\ ne^{\lambda} = 1 \end{cases} \Rightarrow w_i = e^{\lambda} = \frac{1}{n} \Rightarrow$$

When Wi = $\frac{1}{n}$,
We can Maximize objective Function !

Mean Variance Portfolio Optimization

主要架構

```
def objective_function(w):
    w_tp=w.transpose()
    return np.dot(np.dot(w_tp,cov),w)

def equality_constraint_1(w):
    w_tp=w.transpose()
    return 1-np.dot(w_tp,ones)

def equality_constraint_2(w):
    w_tp=w.transpose()
    return u-np.dot(w_tp,returns)
```

Min : $w^T \Omega w$

s.t $\sum_{i=1}^{n} r_i w_i = E \quad , i = 1,2 \ldots . . n$

$$\sum_{i=1}^{n} w_i = 1 \quad , i = 1,2 \ldots . . n$$

$$w_i \geq 0, \quad i = 1,2 \ldots . . n$$

```
ones=np.ones((cov.shape[0],1))
u=0.01
bounds=[(0,1000),(0,1000),(0,1000),(0,1000),(0,1000)]
w0=[1,1,1,1,1]
constraint_1={'type': 'eq','fun':equality_constraint_1}
constraint_2={'type': 'eq','fun':equality_constraint_2}
constraint=[constraint_1,constraint_2]
result=minimize(objective_function,w0,method='SLSQP',bounds=bounds,constraints=constraint)
w=result['x']
```

Data(2020/01~09)

1. 0050
2. 2330
3. 3406
4. 1101
5. 2882

```
PS C:\Users\qw930\Desktop\python> python -u "c
-----------------mean-varianve-portfolio-opti
weights1:  0.0000000000000000000
weights2:  0.2302911443728159202
weights3:  0.0000000000000000008
weights4:  0.1083614004883177023
weights5:  0.6613474551388665024
portfolio variance :  0.0428854200171160З
```

Mean Variance Portfolio Optimization modified by entropy

主要架構

```
def objective_function(w):
    w_tp=w.transpose()
    entropy_part=np.dot(w_tp,np.log(w))
    return np.dot(np.dot(w_tp,cov),w) + entropy_part

def equality_constraint_1(w):
    w_tp=w.transpose()
    return 1-np.dot(w_tp,ones)

def equality_constraint_2(w):
    w_tp=w.transpose()
    return u-np.dot(w_tp,returns)
```

$$\text{Min}: w^T \Omega w - \left[-\sum_{i=1}^{n} w_i * \ln(w_i)\right]$$

$$\text{s.t} \quad \sum_{i=1}^{n} r_i w_i = E \quad ,i = 1,2 \ldots..n$$

$$\sum_{i=1}^{n} w_i = 1 \quad ,i = 1,2 \ldots..n$$

```
ones=np.ones((cov.shape[0],1))
u=0.01
bounds=[(0,1000),(0,1000),(0,1000),(0,1000),(0,1000)]
w0=[1,1,1,1,1]
constraint_1={'type': 'eq','fun':equality_constraint_1}
constraint_2={'type': 'eq','fun':equality_constraint_2}
constraint=[constraint_1,constraint_2]
result=minimize(objective_function,w0,method='SLSQP',bounds=bounds,constraints=constraint)
w=result['x']
```

```
------------------use-entropy-modified-mean-v
weights1:  0.1682574722399195200
weights2:  0.0676868028906073005
weights3:  0.1400762290186602366
weights4:  0.2730539358447347520
weights5:  0.3509255600060781632
portfolio variance :  0.05950352128795654
```

Code

Max Entropy with variance inequality constraint

主要架構

```
def objective_function(w):
    w_tp=w.transpose()
    entropy_part=np.dot(w_tp,np.log(w))
    return entropy_part

def equality_constraint_1(w):
    w_tp=w.transpose()
    return 1-np.dot(w_tp,ones)

def equality_constraint_2(w):
    w_tp=w.transpose()
    return u-np.dot(w_tp,returns)

def inequality_constraint(w):
    w_tp=w.transpose()
    return 0.05950352128795654-np.dot(np.dot(w_tp,cov),w)
```

```
ones=np.ones((cov.shape[0],1))
u=0.01
bounds=[(0,1000),(0,1000),(0,1000),(0,1000),(0,1000)]
w0=[1,1,1,1,1]
constraint_1={'type': 'eq','fun':equality_constraint_1}
constraint_2={'type': 'eq','fun':equality_constraint_2}
constraint_3={'type':'ineq','fun':inequality_constraint}
constraint=[constraint_1,constraint_2,constraint_3]
result=minimize(objective_function,w0,method='SLSQP',bounds=bounds,constraints=constraint)
w=result['x']
```

```
-------------------objective fnc 是 entropy part--
variance 主觀設定不能大於 0.05950352128795654
weights1:  0.16838850951603386
weights2:  0.06764023148163482
weights3:  0.14008148317495730
weights4:  0.27295652137979514
weights5:  0.35093325444757889
```

Min : $w^T \Omega w - [-\sum_{i=1}^{n} w_i * \ln(w_i)]$

等價?

Max: $-\sum_{i=1}^{n} w_i * \ln(w_i)$

s.t $\sum_{i=1}^{n} r_i w_i = E \quad , i = 1,2 \ldots\ldots n$

$\sum_{i=1}^{n} w_i = 1 \quad , i = 1,2 \ldots\ldots n$

$w^T \Omega w \leq Var$

Code

Max Entropy with variance inequality constraint

主要架構

```
def objective_function(w):
    w_tp=w.transpose()
    entropy_part=np.dot(w_tp,np.log(w))
    return entropy_part

def equality_constraint_1(w):
    w_tp=w.transpose()
    return 1-np.dot(w_tp,ones)

def equality_constraint_2(w):
    w_tp=w.transpose()
    return u-np.dot(w_tp,returns)

def inequality_constraint(w):
    w_tp=w.transpose()
    return 0.04288542001711603-np.dot(np.dot(w_tp,cov),w)
```

Max: $-\sum_{i=1}^{n} w_i * \ln(w_i)$

s.t $\sum_{i=1}^{n} r_i w_i = E \quad ,i = 1,2 \ldots..n$

$\sum_{i=1}^{n} w_i = 1 \quad ,i = 1,2 \ldots..n$

$w^T \Omega w \leq Var$

```
ones=np.ones((cov.shape[0],1))
u=0.01
bounds=[(0,1000),(0,1000),(0,1000),(0,1000),(0,1000)]
w0=[1,1,1,1,1]
constraint_1={'type': 'eq','fun':equality_constraint_1}
constraint_2={'type': 'eq','fun':equality_constraint_2}
constraint_3={'type':'ineq','fun':inequality_constraint}
constraint=[constraint_1,constraint_2,constraint_3]
result=minimize(objective_function,w0,method='SLSQP',bounds=bounds,constraints=constraint)
w=result['x']
```

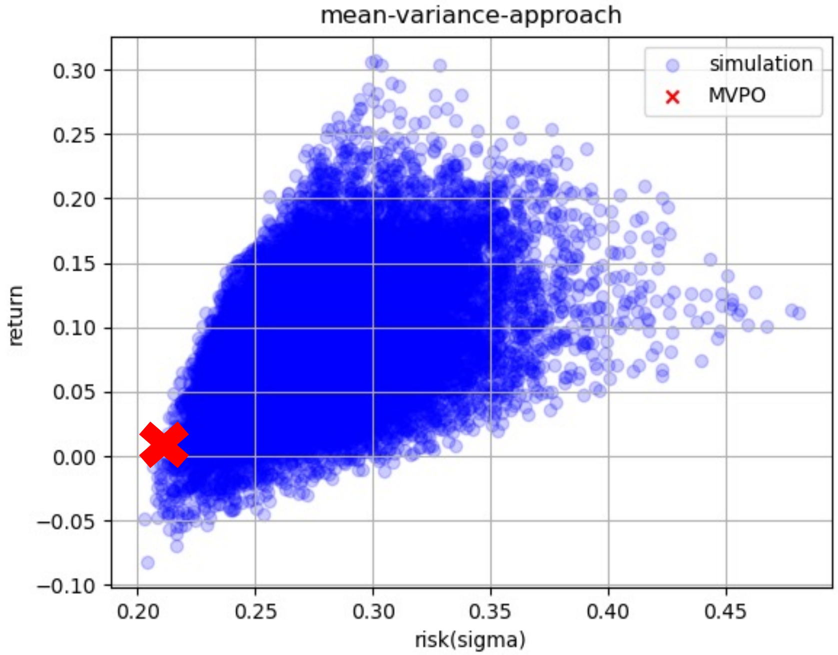
```
-----------------objective fnc 是 entropy part-----
若variance 主觀設定不能大於 的是0.04288542001711603(
weights1:  0.00000000000000000004
weights2:  0.2301415624581996666
weights3:  0.00000000000000000099
weights4:  0.1094756977255134656
weights5:  0.6603827398162869233
```

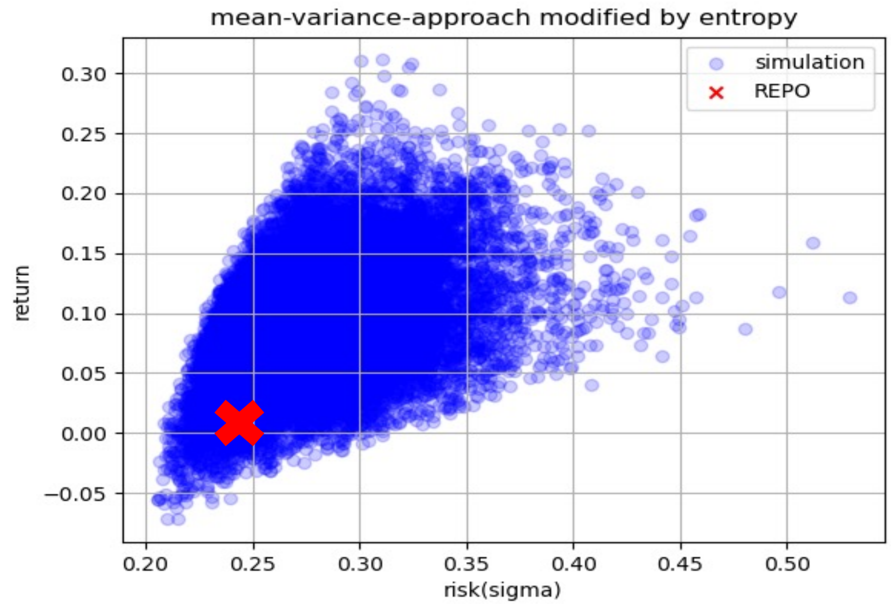
回到MVPO！

Comparison

Min : $w^T \Omega w$

Min : $w^T \Omega w - [-\sum_{i=1}^{n} w_i * \ln(w_i)]$





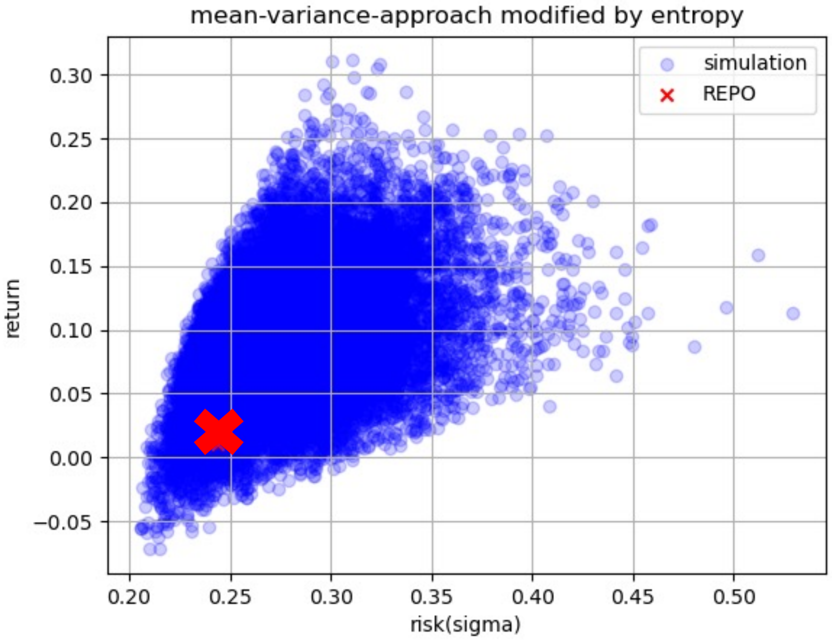

```
PS C:\Users\qw930\Desktop\python> python -u "c:\Users\qw930\Desktop\python\????\??
-----------------mean-varianve-portfolio-optimization-approach------------------
weights1:  0.0000000000000000000
weights2:  0.2302911443728159202
weights3:  0.0000000000000000008
weights4:  0.1083614004883177023
weights5:  0.6613474551388665024
portfolio risk :  0.20708795237076452
portfolio return :  0.009999999998754297
```

```
-----------------objective fnc 是 entropy part---------------
variance 主觀設定不能大於 0.05950352128795654
weights1:  0.1683885095160338186
weights2:  0.0676402314816348382
weights3:  0.1400814831749573053
weights4:  0.2729565213797951473
weights5:  0.3509332544475788906
portfolio risk :  0.24393363076694946
portfolio return :  0.009999999999989635
portfolio entropy :  1.4794094170712746
```

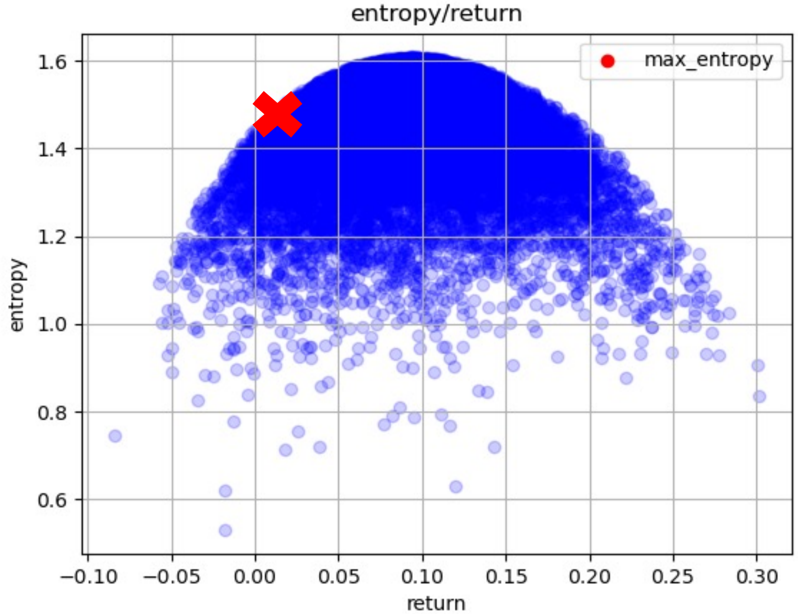
Summary

$$\text{Min}: w^T \Omega w - \left[-\sum_{i=1}^{n} w_i * \ln(w_i)\right]$$

Portfolio risk and return



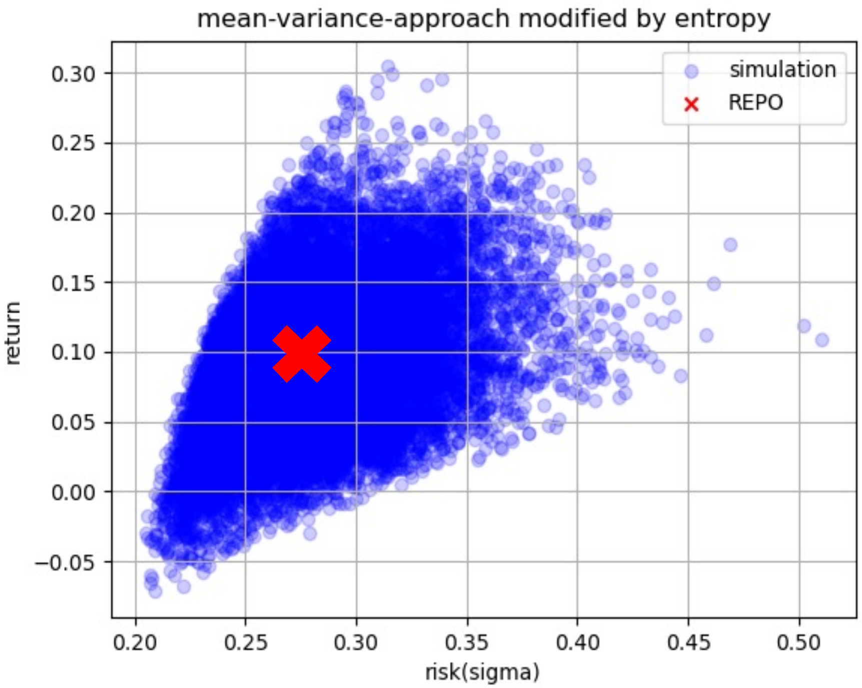
Portfolio entropy and return

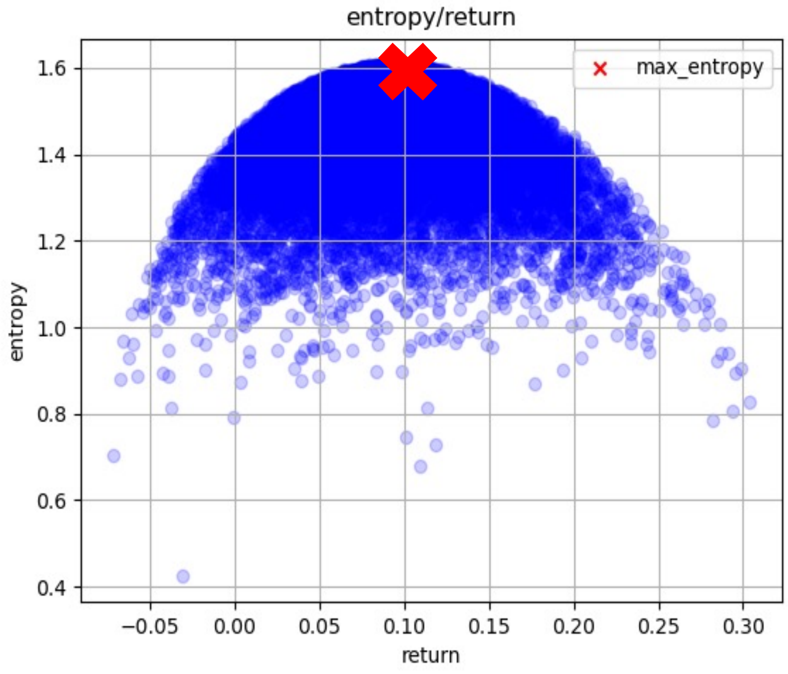



```
-----------------objective fnc 是 entropy part---------------------------
variance 主觀設定不能大於 0.05950352128795654
weights1:  0.1683885095160338186
weights2:  0.0676402314816348382
weights3:  0.1400814831749573053
weights4:  0.2729565213797951473
weights5:  0.3509332544475788906
portfolio risk :  0.24393363076694946
portfolio return :  0.009999999999989635
portfolio entropy :  1.4794094170712746
```

Summary

Without any Constraint






```
PS C:\Users\qw930\Desktop\python> python -u "c:\Users\qw930\Desktop\python\
-----------------objective fnc 是 entropy part---------------------------
weights1:  0.2003935561575734547
weights2:  0.2116943159889726478
weights3:  0.2012406209921142608
weights4:  0.1945957577684544748
weights5:  0.1920757490928860223
portfolio risk :  0.2789236655243451
portfolio return :  0.10000001055302542
portfolio entropy :  1.6088654873854584
```

Thank you for your Listening~

報告人：清華大學計量財務金融學系 , 江晨立