

Informetric Approach to Portfolio Optimization

This project compares the mean-variance based portfolio optimization with maximized entropy based portfolio optimization. Empirically, mean-variance approach often leads to extreme weights concentrated on a few assets, as a result, the out-sample performance often very poor. Using cross-entropy as the objective function, we demonstrate that entropy based approach offers diversification benefits and superior out-sample performance.

Step 1: Import Equity Return Data

The following dataset contains monthly return of 10 industry sectors. Health, utilities and others are chosen for this demo.

The following data came from Professor Kenneth French (<http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/>)

```
clc
clear

[A, B] = importdata('10_Industry_Portfolios.csv');

Date = A.data(:,1);
Data = A.data(:,end-2:end); % pick 3 industry

%%
% in-sample data
startindex = find(200001 == Date);
endindex = find(201601 == Date);

Date = Date(startindex:endindex,:);
Data = Data(startindex:endindex,:);

AssetReturns = mean(Data)'
```

```
AssetReturns =

    0.6832
    0.8193
    0.4456
```

```
AssetCov = cov(Data)
```

```
AssetCov =

    16.1013    7.7834    13.1114
    7.7834    18.2606    10.6404
    13.1114    10.6404    28.5164
```

```
AssetVar = diag(AssetCov);
AssetCorr = corrcoef(Data);
nAsset = length(AssetReturns);
```

```
% out-sample data
startindex2 = find(201602 == Date);
endindex2 = find(201610 == Date);
Date2 = Date(startindex2:endindex2,:);
Data2 = Data(startindex2:endindex2,:);
```

Step 2: Visualize Efficient Frontier

Compare efficient frontier based on mean-variance optimization and max entropy.

- **traditional MV framework:**

$$\min \pi' \Sigma \pi$$

$$\text{s.t. } \pi' \hat{\mu} = \mu_0 \text{ and } \pi' 1_N = 1$$

- **max entropy approach (without variance as constraint)**

$$\max - \pi' \ln \pi$$

$$\text{s.t. } \pi' \hat{\mu} = \mu_0 \text{ and } \pi' 1_N = 1$$

- **max entropy approach (with variance as constraint)**

$$\max - \pi' \ln \pi$$

$$\text{s.t. } \pi' \hat{\mu} \geq \mu_0 \text{ and } \sqrt{\pi' \hat{\Sigma} \pi} \leq \sigma_0 \text{ and } \pi' 1_N = 1$$

```
% min variance portfolio weight
[ret_mv,sig_mv, wgt_mv] = EfficientFrontier_MV(100, AssetCov, AssetReturns);

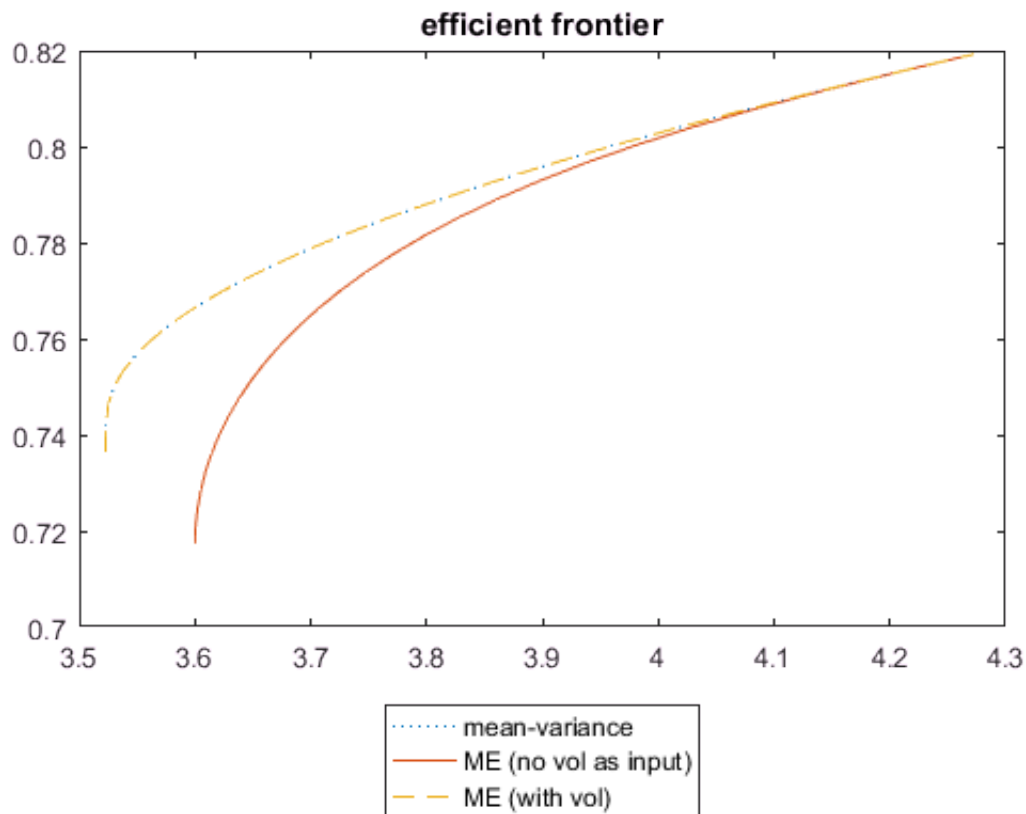
minVarWgt = wgt_mv(1,:)'
```

```
minVarWgt =
```

```
0.5407
0.4350
0.0243
```

```
[ret_me,sig_me, wgt_me] = EfficientFrontier_ME(100, AssetCov, AssetReturns);
[ret_me2,sig_me2, wgt_me2] = EfficientFrontier_ME_withVol(100, AssetCov, AssetReturns);

plot(sig_mv,ret_mv,':',sig_me,ret_me,sig_me2,ret_me2,'--')
legend('mean-variance','ME (no vol as input)','ME (with vol)','Location','southoutside')
title('efficient frontier')
```



As we can see from the plot, without second moment as constraint, the max entropy approach yields a frontier that is lower than ME's frontier. After adding in the second moment as constraint, the max entropy approach has exactly same frontier as MV approach.

Step 3: Visualize Entropy Contour Curves

In the context of entropy based portfolio optimization, portfolio diversification can also be defined as

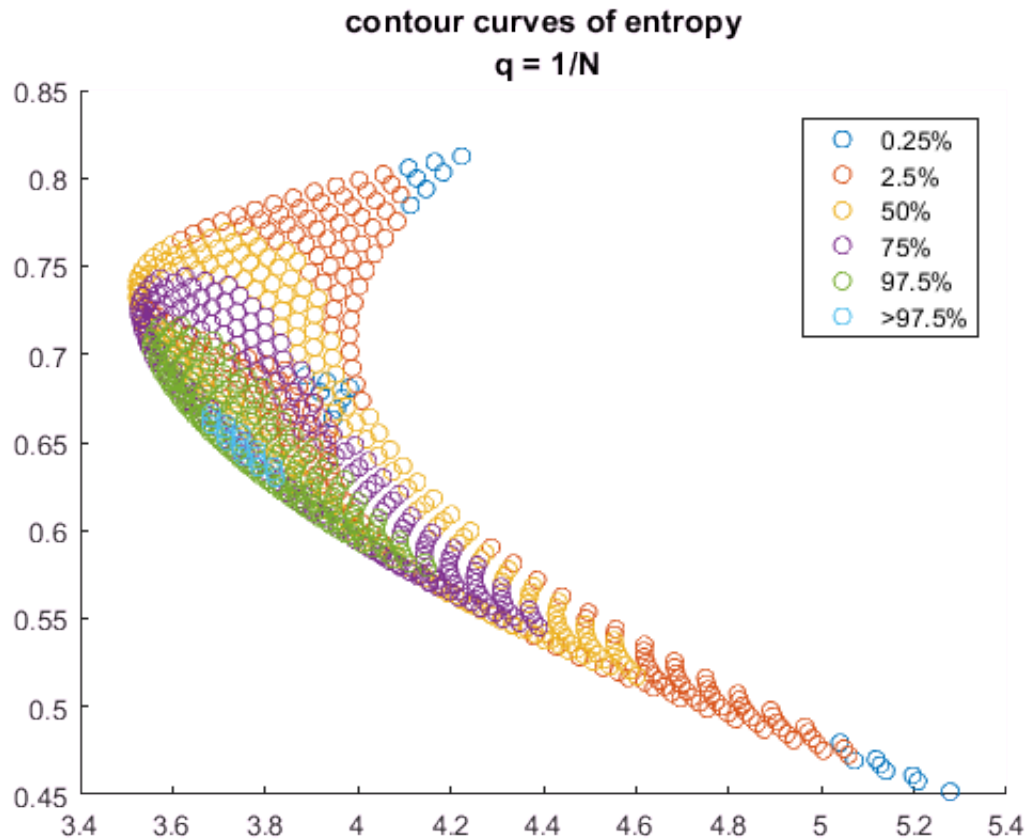
$$CE(\pi | q) = \sum_{i=1}^N \pi_i \ln \left(\frac{\pi_i}{q_i} \right)$$

where π_i are the weight of each asset, and q_i are the prior weight of each asset.

- Example 1: use equal weights as the prior weights

$$q = \frac{1}{N}$$

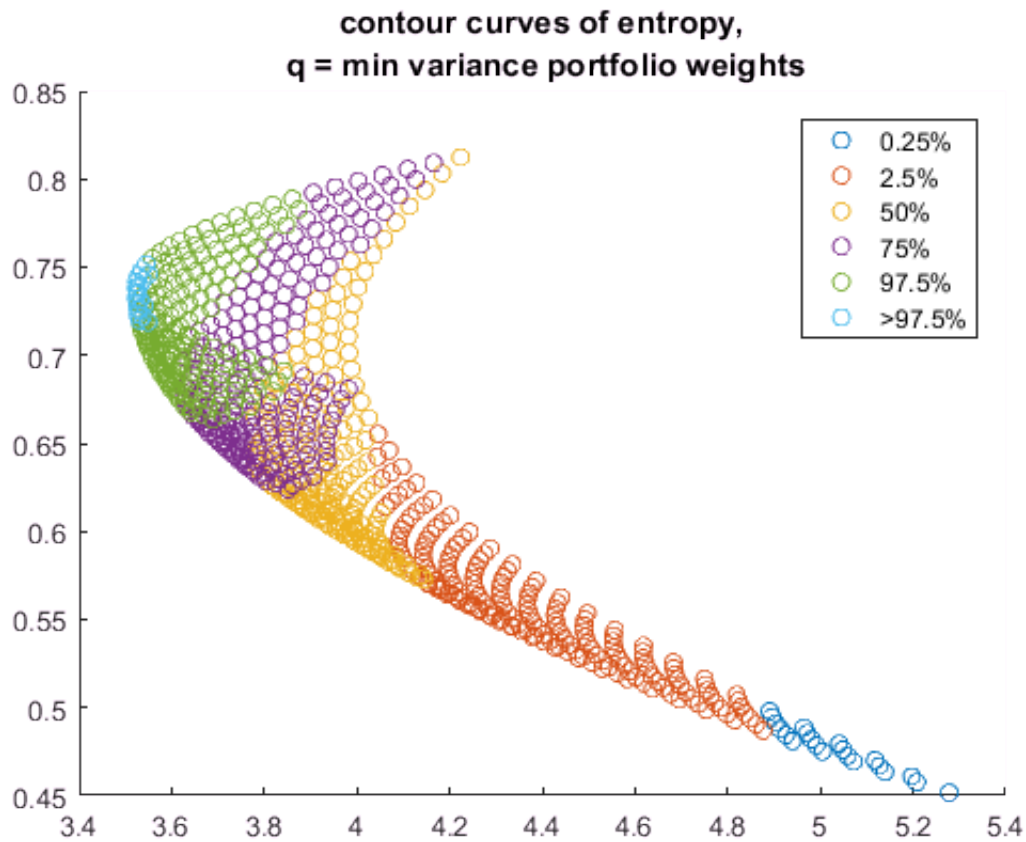
```
% prior weight = 1/N
plotEntropyContour(Data)
title({'contour curves of entropy', ' q = 1/N'})
```



- **Example 2: use min variance portfolio weights as the prior weights**

$$q = \pi_{\min\text{Vol}}$$

```
% prior weight = min variance
plotEntropyContour(Data,minVarWgt)
title({'contour curves of entropy',' q = min variance portfolio weights'})
```



Step 4: Solve for Optimal Portfolio Weights for Given Risk Aversion (λ)

For given level of risk aversion, in the traditional MV framework, we can obtain the optimal portfolio weights by maximizing the following quadratic utility

$$\max EU(\pi, R, \lambda) = \max \left[\pi' \mu_0 - \frac{\lambda}{2} \pi' \Sigma_0 \pi \right]$$

subject to

$$\pi \geq 0 \text{ and } \pi' \mathbf{1}_N = 1$$

```
lambda = 0.06; % risk-aversion coefficient.
```

```
% optMV : mean-variance solution
```

```
[optMV_ret ,optMV_sig ,optMV_wgt]=Opt_MeanVariance(Data,lambda);
```

```
optMV_util = optMV_wgt'*mean(Data)' - 0.06*optMV_wgt'*cov(Data)*optMV_wgt/2
```

```
optMV_util = 0.3790
```

Bera and Park (2004) proposed an approach to incorporate the estimation errors in mean and covariance by resampling method. By re-simulating multivariate stock returns B times from the empirical distribution using the sample mean (μ_0) and sample covariance (Σ_0), one can measure the imprecision of the sample moments in terms of expected utility (e.g, quadratic utility) generated by MV approach. For each of $b = 1, 2, \dots, B$ times, we follow these steps:

1. resample the return distribution \tilde{R}_b , estimate the mean $\tilde{\mu}_b$ and covariance $\tilde{\Sigma}_b$
2. using MV to solve for optimal portfolio weights $\tilde{\pi}_b$ and corresponding expected utility $\tilde{\xi}_b$ (quadratic utility)

After doing this B times (e.g, 1000 times), we can obtain a non-parametric kernel density for expected utility distribution

```
% find the distribution of expected utility

psi = simExpectedUtility(Data,lambda);
[f,xi] = ksdensity(psi); % pdf
plot(xi,f)
title({'non-parametric density for expected utility',' (from re-sampling)'})
xlabel('expected utility')
ylabel('density')
```

For a given level of confidence, which can be interpreted as investor's strength of belief in estimation, one can obtain the optimal portfolio weights by the following minimization problem:

$$\min \text{CE}(\pi | q) = \sum_{i=1}^N \pi_i \ln \left(\frac{\pi_i}{q_i} \right)$$

subject to

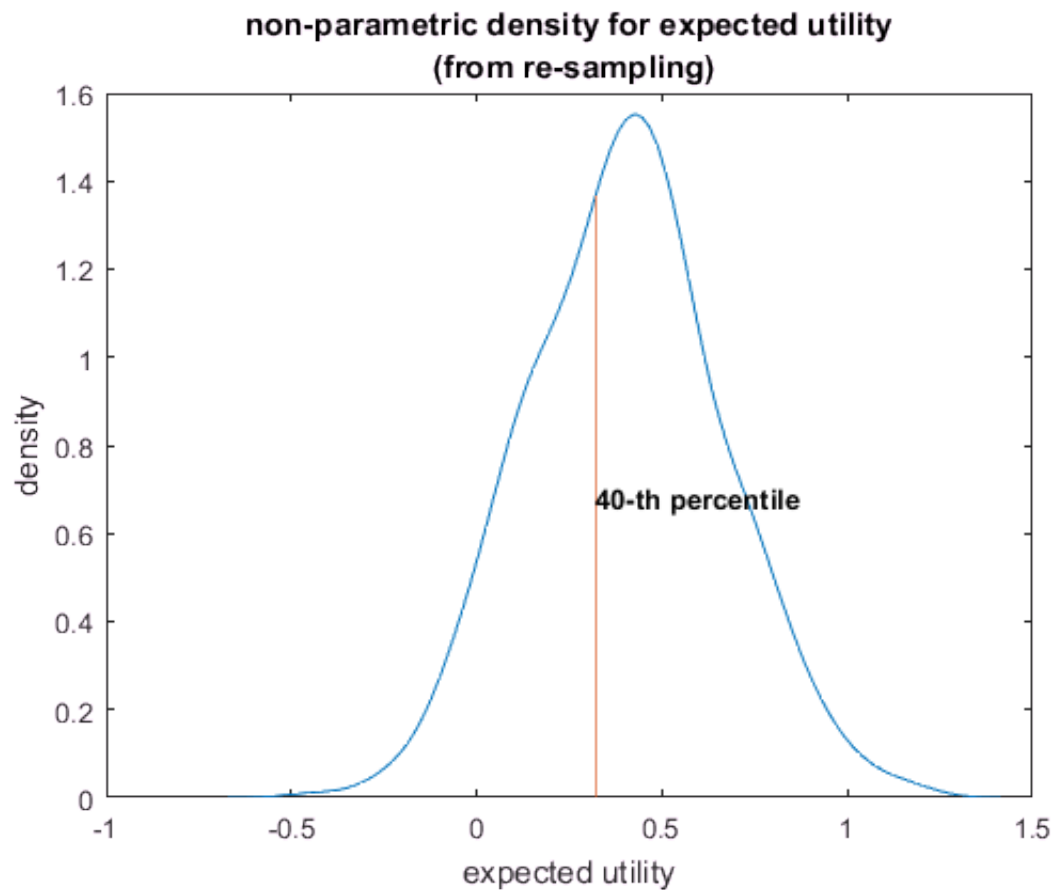
$$\text{EU}(\pi, R, \lambda) \geq \xi_\alpha \text{ and } \pi' \mathbf{1}_N = 1$$

where ξ_α corresponds to the α^{th} percentile of the expected utility distribution.

```
% alpha the percentile (confidence level)
alpha = 0.4;
[F,Xi] = ksdensity(psi,'function','cdf'); %cdf
[~, index]=find(F<=alpha);% percentile
psi_alpha = Xi(index(end)) % expected utility at 40-th percentile
```

```
psi_alpha = 0.3196
```

```
f_alpha = f(xi==psi_alpha);
hold on
plot([psi_alpha psi_alpha],[0 f_alpha])
text(psi_alpha, f_alpha/2, '40-th percentile', 'fontweight', 'bold')
hold off
```



Based on the given level of confidence ξ_α , we minimize the cross-entropy to get the optimal portfolio weights

```
% optCE cross-entropy solution (q = 1/n)
[optCE_ret, optCE_sig, optCE_wgt]=Opt_CrossEntropy(Data,lambda,psi_alpha);
optCE_util = optCE_wgt'*mean(Data)' - 0.06*optCE_wgt'*cov(Data)*optCE_wgt/2
```

```
optCE_util = 0.3196
```

The expected utility based on cross-entropy optimal weights (ξ_α^{CE}) is not very far away from the expected from MV approach at α^{th} percentile (ξ_α^{MV})

```
% optCE cross-entropy solution (q = min variance )
[optCE_ret2, optCE_sig2, optCE_wgt2]=Opt_CrossEntropy(Data,lambda,psi_alpha,minVarWgt);
optCE2_util = optCE_wgt2'*mean(Data)' - lambda*optCE_wgt2'*cov(Data)*optCE_wgt2/2
```

```
optCE2_util = 0.3643
```

Step 5: Visualize Shrinkage Effect

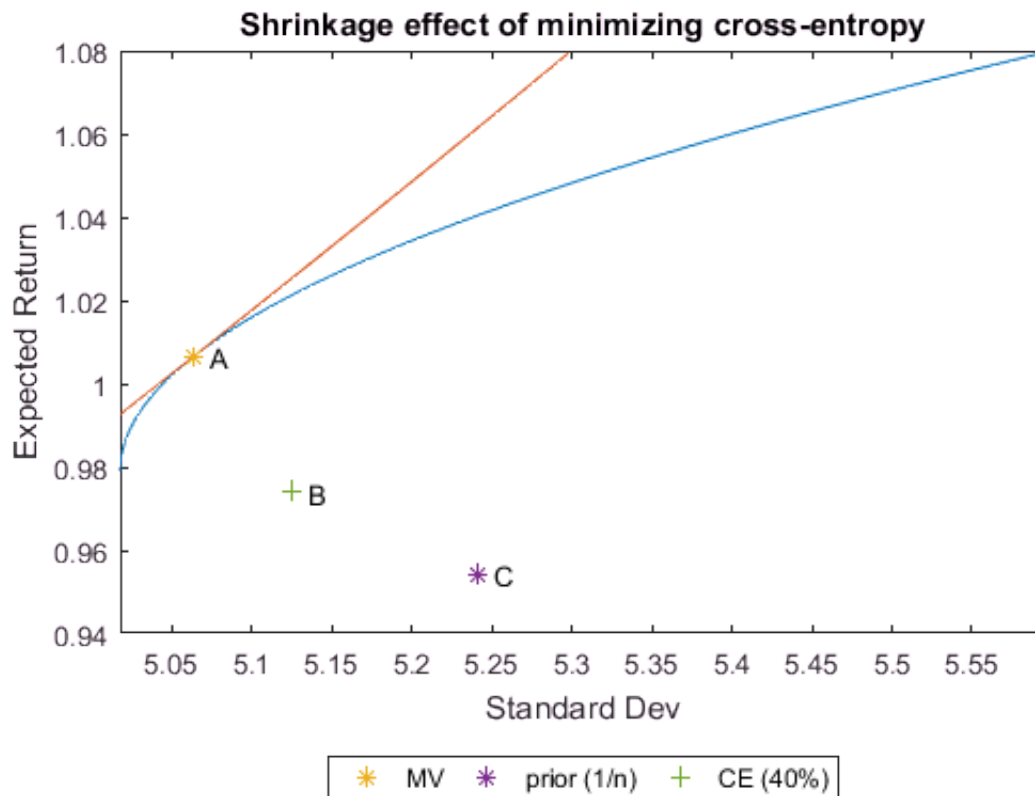
Note that minimization of CE shrinks MV efficient portfolio (point A) towards more diversified portfolio (equally weighted, point C). The degree of shrinkage depends on the investor's degree of uncertainty aversion (confidence level).

```
% equal weights portfolio, the prior weight (q = 1/n)
pi = ones(nAsset,1)/nAsset;
eq_sig = sqrt(pi'*AssetCov*pi);
eq_ret = pi'*AssetReturns;

% Utility Curve
% U = R - lambda * V^2/2
for i = 1:length(ret_mv)
    mvSig(i) = sqrt(2*(ret_mv(i) - optMV_util)/lambda);
end

plot(sig_mv,ret_mv)
hold on
plot(mvSig,ret_mv)
h1=plot(optMV_sig,optMV_ret,'*');
h2=plot(eq_sig,eq_ret,'*');
h3=plot(optCE_sig,optCE_ret,'+');
xlabel('Standard Dev')
ylabel('Expected Return')
legend([h1 h2 h3 ],{'MV','prior (1/n)','CE (40%)'},'Location','southoutside','Orientation','horizontal')
xlim([min(sig_mv) max(sig_mv)])
title('Shrinkage effect of minimizing cross-entropy')
hold off

text(optMV_sig+1/100,optMV_ret,'A')
text(eq_sig+1/100,eq_ret,'C')
text(optCE_sig+1/100,optCE_ret,'B')
```

Step 6: Compare Emperical Performance (no short-selling)

(warning: this will take a little bit time)

To evaluate the performance , we use Sharpe ratio as the measure based on a "rolling window" scheme. Consider $W = 24$ months, and estimate parameter values over each W and all asset models.

The average of in-sample Sharpe ratio can be calculated as:

$$SR_{in} = \frac{1}{T - W} \sum_{t=W}^T \frac{\hat{\pi} \mu_t}{\sqrt{\hat{\pi} \Sigma_t \hat{\pi}}}$$

Following the rolling window scheme, based on the optimal weights estimated from window $[t-W+1, t]$, the next period return can be calculated as

$$\hat{\mu}_{t+1} = \hat{\pi}_t R_{t+1}$$

and the outsample Sharpe ratio can be calculated:

$$\mu = \frac{1}{T - W} \sum_{t=W}^T \hat{\mu}_t$$

$$\sigma^2 = \frac{1}{T-W} \sum_{t=W}^T (\hat{\mu}_t - \mu)^2$$

$$SR_{out} = \frac{\mu}{\sigma}$$

```
[nDates, ~]= size(Data);
```

```
T = nDates;
```

```
W = 24; % 24-months rolling window
```

```
alpha = 0.4; % given confidence level (percentile of Exepcted utility)
```

```
for i = W:T-1
```

```
    tmp = Data(i-W+1:i,:); % in-sample data:[i-W+1:i]
```

```
    tmpRet = mean(tmp)';
```

```
    tmpSig = cov(tmp);
```

```
    psi = simExpectedUtility(tmp,lambda);
```

```
    [F,Xi] = ksdensity(psi,'function','cdf'); %cdf
```

```
    [~, index]=find(F<=alpha);% percentile
```

```
    psi_alpha = Xi(index(end)); % expected utility at alpha-th percentile
```

```
    [optCE_ret, optCE_sig, optCE_wgt]=Opt_CrossEntropy(tmp,lambda,psi_alpha);
```

```
    [optMV_ret, optMV_sig, optMV_wgt]=Opt_MeanVariance(tmp,lambda);
```

```
    % in-sample
```

```
    optCE_wgt_all(:,i-W+1)=optCE_wgt;
```

```
    optMV_wgt_all(:,i-W+1)=optMV_wgt;
```

```
    SR_in.CE_all(i-W+1) = (optCE_wgt'*tmpRet)/sqrt(optCE_wgt'*tmpSig*optCE_wgt); % Sharpe Rat
```

```
    SR_in.MV_all(i-W+1) = (optMV_wgt'*tmpRet)/sqrt(optMV_wgt'*tmpSig*optMV_wgt); % Sharpe Rat
```

```
    %out-sample
```

```
    mu_t.CE_all(i-W+1) = optCE_wgt'*Data(i+1,:)';
```

```
    mu_t.MV_all(i-W+1) = optMV_wgt'*Data(i+1,:)';
```

```
end
```

```
SR_in.CE = mean(SR_in.CE);
```

```
SR_in.MV = mean(SR_in.MV);
```

```
SR_in
```

```
SR_in =
```

```
    CE: 0.3631
```

```
    MV: 0.3706
```

```
SR_out.CE = mean(mu_t.CE)/std(mu_t.CE);
```

```
SR_out.MV = mean(mu_t.MV)/std(mu_t.MV);
```

```
SR_out
```

```
SR_out =
```

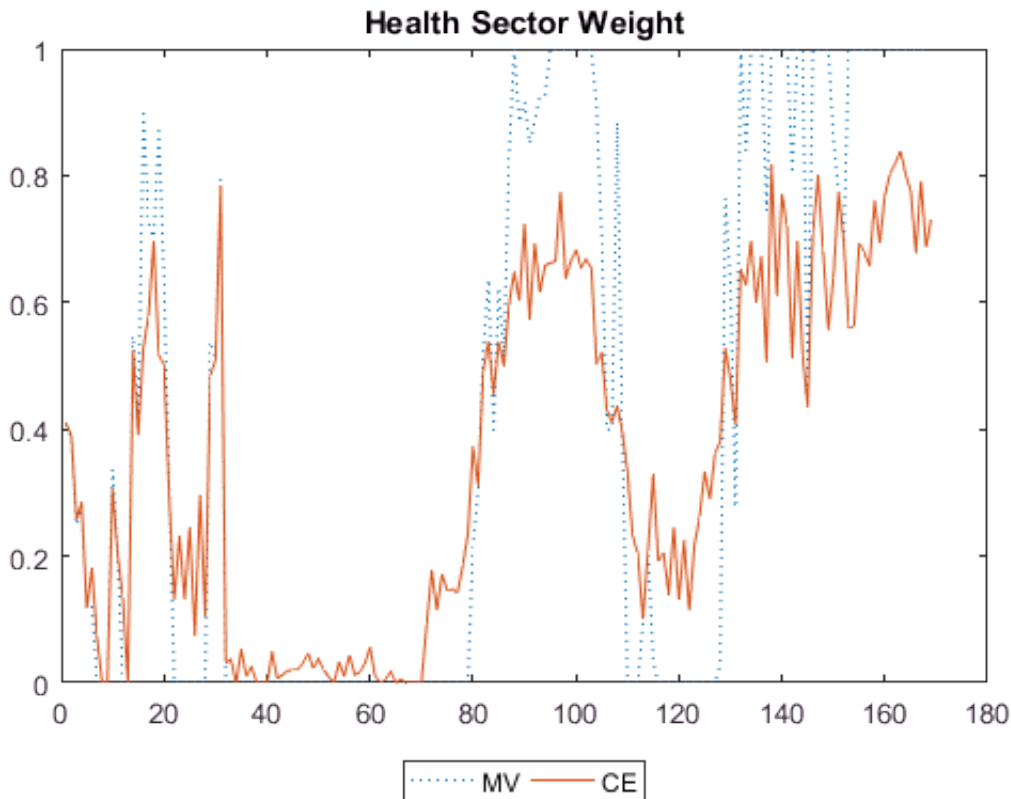
```
    CE: 0.1782
```

```
    MV: 0.1676
```

```

plot(optMV_wgt_all(1,:), ':')
hold on
plot(optCE_wgt_all(1,:))
hold off
title('Health Sector Weight')
legend({'MV', 'CE'}, 'Location', 'southoutside', 'Orientation', 'horizontal')

```



Step 7: Compare Empirical Performance (*with short-selling*)

Following generalized cross entropy method proposed by Golan, Judge and Miller (1996), we can allow for negative weights. For each asset weight (P_i) has a discrete distribution $\{P_{i1}, P_{i2}, P_i, \dots, P_{iM}\}$ on a set of equally spaced discrete points $Z = \{z_1, z_2, z_3, \dots, z_M\}$. Also, for each prior weight (q_i) over z , define a discrete prior probability distribution $w_i = \{w_{i1}, w_{i2}, w_i, \dots, w_{iM}\}$. The portfolio weights can be represented by

$$\pi = Zp = \begin{bmatrix} z' & 0 & 0 & 0 & 0 \\ 0 & z' & 0 & 0 & 0 \\ 0 & 0 & z' & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & z' \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ \vdots \\ p_N \end{bmatrix}$$

The objective function is a generalized cross-entropy type of minimization problem

$$\min \sum_{i=1}^N \sum_{j=1}^M p_{ij} \ln \left(\frac{p_{ij}}{w_{ij}} \right)$$

subject to

$$\pi \mu - \frac{\lambda}{2} \pi \Sigma \pi \geq \xi_{\alpha}$$

$$p_i \mathbf{I}_m = 1, \quad i = 1, 2, \dots, N$$

$$\pi \mathbf{I}_N = 1$$

For example, let's consider 11 equal weighted discrete points $z = \{-1, -0.8, -0.6, \dots, 1\}$, while $w_i = \{w_{i1}, w_{i2}, w_{i3}, \dots, w_{iM}\}$ can be obtained using maximum entropy principle by

$$\max - \sum_{m=1}^M w_{im} \ln(w_{im})$$

subject to

$$\sum_{m=1}^M z w_{im} = q_i$$

$$\sum_{m=1}^M w_{im} = 1$$

```
%% with short
[nDates nAsset]= size(Data);

T = nDates;
W = 24; % 24-months rolling window
alpha = 0.4; % given confidence level (percentile of Exepcted utility)

for i = W:T-1
    tmp = Data(i-W+1:i,:);
    tmpRet = mean(tmp)';
    tmpSig = cov(tmp);
    psi = simExpectedUtility_Short(tmp,lambda);
    [F,Xi] = ksdensity(psi,'function','cdf'); %cdf
    [~, index]=find(F<=alpha);% percentile
    psi_alpha = Xi(index(end)); % expected utility at alpha-th percentile
    [optCE_short_ret, optCE_short_sig, optCE_short_wgt]=Opt_CrossEntropy_Short(tmp,lambda,psi_
    [optMV_short_ret, optMV_short_sig, optMV_short_wgt]=Opt_MeanVariance_Short(tmp,lambda);
```

```

optCE_short_wgt_all(:,i-W+1)=optCE_short_wgt;
optMV_short_wgt_all(:,i-W+1)=optMV_short_wgt;

SR_in.CE_short_all(i-W+1) = (optCE_short_wgt'*tmpRet)/sqrt(optCE_short_wgt'*tmpSig*optCE_short_wgt);
SR_in.MV_short_all(i-W+1) = (optMV_short_wgt'*tmpRet)/sqrt(optMV_short_wgt'*tmpSig*optMV_short_wgt);

%out-sample
mu_t.CE_short(i-W+1) = optCE_short_wgt'*Data(i+1,:);
mu_t.MV_short(i-W+1) = optMV_short_wgt'*Data(i+1,:);
end

SR_in.CE_short = mean(SR_in.CE_short_all);
SR_in.MV_short = mean(SR_in.MV_short_all);
SR_in

```

```

SR_in =
    CE: 0.3631
    MV: 0.3706
CE_short: 0.4027
MV_short: 0.3907

```

```

SR_out.CE_short = mean(mu_t.CE_short)/std(mu_t.CE_short);
SR_out.MV_short = mean(mu_t.MV_short)/std(mu_t.MV_short);
SR_out

```

```

SR_out =
    CE: 0.1782
    MV: 0.1676
CE_short: 0.1816
MV_short: 0.1670

```

```

plot(optMV_short_wgt_all(1,:),':')
hold on
plot(optCE_short_wgt_all(1,:))
hold off
title('Health Sector Weight (with short)')
legend({'MV (with short)','CE(with short)'},'Location','southoutside','Orientation','horizontal')
ax = gca;
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';

```



Conclusion:

Entropy based portfolio optimization approach offers additional diversification benefits and by avoid placing extreme weights on a few assets, the out-sample performance is superior than mean-variance approach (in terms of Sharpe ratio).