

[量化学堂-金融市场]Beta对冲

对冲, 市场中性

iQuant

17年5月 #1

导语：本文介绍了因子模型、对冲以及Beta因子的相关内容，并针对如何进行市场风险对冲给出了具体的案例。

因子模型

因子模型是通过其他若干项资产回报的线性组合来解释一项资产回报的一种方式，因子模型的一般形式是：

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

这看起来很熟悉，因为它正是多元线性回归模型。

什么是beta?

一项资产的 β 是该资产收益率与其他资产收益率通过上述模型回归拟合的 β 。比如，我们用回归模型 $Y_{gzmt} = \alpha + \beta X_{benchmark}$ 来描述贵州茅台收益率相对于沪深300回归的 β 值，如果我们使用模型 $Y_{gzmt} = \alpha + \beta_1 X_{benchmark} + \beta_2 X_{wly}$ ，那么就会出现两个 β ，一个是贵州茅台对沪深300的风险暴露，一个是贵州茅台对五粮液的风险暴露。

通常而言， β 更多地指该资产相对于基准指数的风险暴露，即只相对于市场基准的一元线性回归所得到的回归系数。

什么是对冲?

如果我们确定我们的投资组合的回报与市场的关系如下面公式所示：

$$Y_{portfolio} = \alpha + \beta X_{hs300}$$

于是，我们可以建立沪深300空头头寸来对冲市场风险，对冲的市值为 $-\beta V$ ，如果我们持有多头组合的市值是 V 。因为我们多头组合的收益为 $\alpha + \beta X_{hs300}$ ，沪深300对冲空头的收益为 $-\beta X_{hs300}$ ，于是我们最终的收益为 $\alpha + \beta X_{hs300} - \beta X_{hs300} = \alpha$ ，于是我们的收益来源只有 α ，而与市场系统风险没有关系。

风险暴露

一般而言，beta描述的是持有资产所承担的系统风险敞口这一概念。如果一项资产相对沪深300基准指数具有较高的 β 暴露水平，那么在市场上涨时，它的表现将会很好，当市场下跌时，它表现很差。高 β 对应于高系统风险（高市场风险），意味着你的投资更具有波动性。

在BigQuant，我们重视尽可能没有系统风险暴露的市场中性策略。这意味着策略中的所有回报都在模型的 α 部分，而与市场无关。因为这意味着该策略与市场系统风险无关，不管是牛市还是熊市，它都具有稳定的业绩表现。市场中性策略对于拥有大量现金池的机构（银行、保险、公募基金等）最具吸引力。

风险管理

减少因子风险暴露的过程称为风险管理。 对冲是在实践中进行风险管理的最佳方式之一。

本文通过具体案例来了解如何做到市场风险对冲的，我们使用贵州茅台和基准沪深300来构建我们的投资组合，将沪深300的权重设为 $-\beta$ （由于持有基准空头头寸）。

附件：市场风险对冲的具体案例

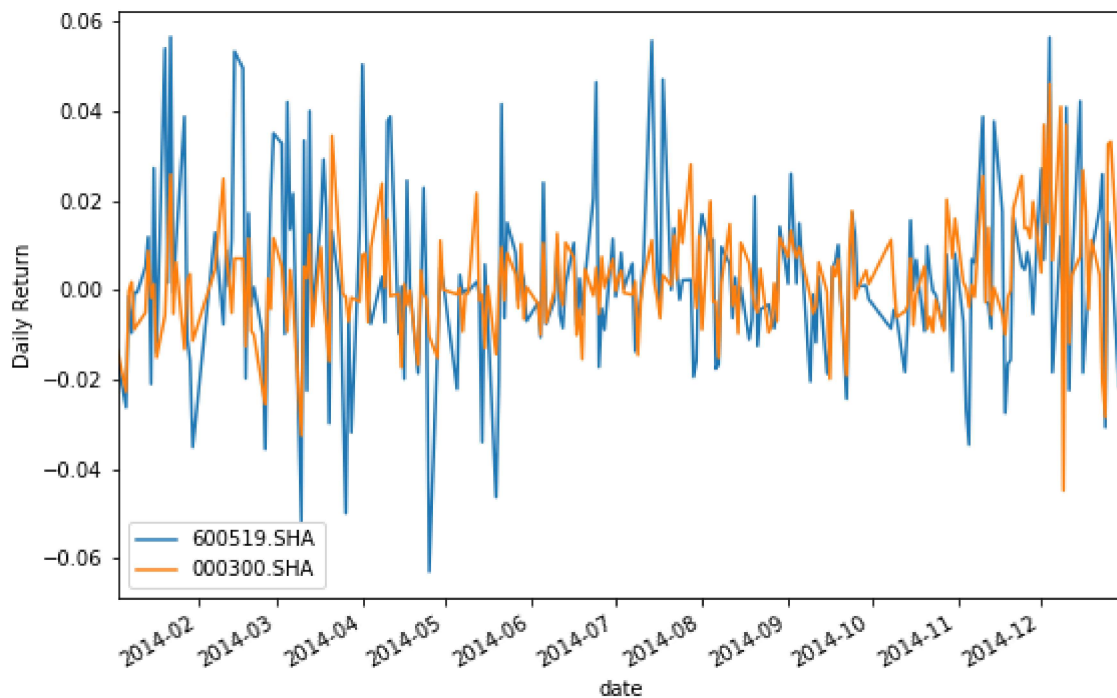
 克隆策略

```
# 导入相应的模块
import numpy as np
from statsmodels import regression
import statsmodels.api as sm
import matplotlib.pyplot as plt
import math
```

```
# 获取一段时间的股票数据
start_date = '2014-01-01'
end_date = '2015-01-01'
asset = D.history_data('600519.SHA', start_date, end_date, fields=['close']).set_index('date')['close']
benchmark = D.history_data('000300.SHA', start_date, end_date, fields=['close']).set_index('date')['close']
asset.name = '600519.SHA'
benchmark.name = '000300.SHA'

# 计算收益率
r_a = asset.pct_change()[1:]
r_b = benchmark.pct_change()[1:]

# 绘制
r_a.plot(figsize=[9,6])
r_b.plot()
plt.ylabel("Daily Return")
plt.legend();
```



现在我们可以通过回归求出 $\alpha(\alpha)$ 和 $\beta(\beta)$

```

X = r_b.values
Y = r_a.values
x = sm.add_constant(X)

def linreg(x,y):
    # 增加一个常数项
    x = sm.add_constant(x)
    model = regression.linear_model.OLS(y,x).fit()
    # 再把常数项去掉
    x = x[:, 1]
    return model.params[0], model.params[1]

alpha, beta = linreg(X,Y)
print('alpha: ' + str(alpha))
print('beta: ' + str(beta))

```

alpha: 0.00116253939056

beta: 0.672934653004

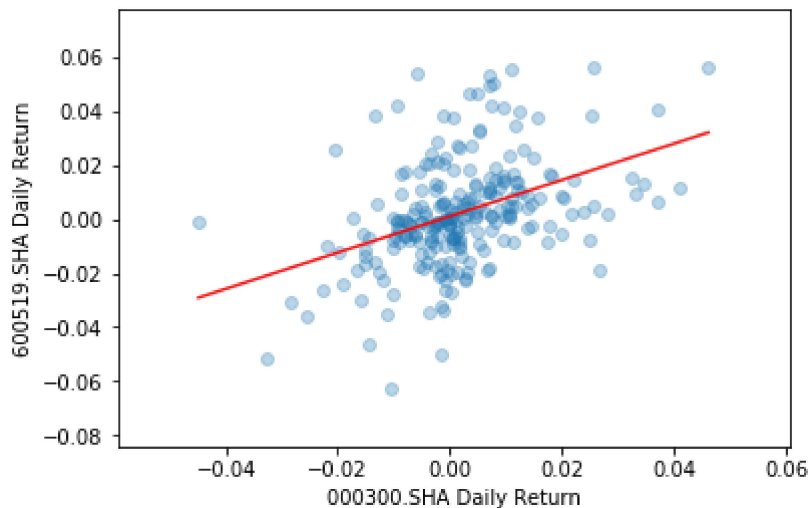
```

X2 = np.linspace(X.min(), X.max(), 100)
Y_hat = X2 * beta + alpha

plt.scatter(X, Y, alpha=0.3) # 画出原始数据散点
plt.xlabel("000300.SHA Daily Return")
plt.ylabel("600519.SHA Daily Return")

# 增加一条红色的回归直线
plt.plot(X2, Y_hat, 'r', alpha=0.9);

```



风险暴露

一般而言，beta描述的是持有资产所承担的 **系统风险敞口**这一概念，用 β 表示。如果一项资产相对沪深300基准指数具有较高的 β 暴露水平，那么在市场上涨时，它的表现将会很好，当市场下跌时，它表现很差。高 β 对应于高系统风险（高市场风险），意味着你的投资更具有波动性。

在BigQuant，我们重视尽可能没有系统风险暴露的 **市场中性策略**。这意味着策略中的所有回报都在模型的 α 部分，而与市场无关。因为这意味着该策略与市场系统风险无关，不管是牛市还是熊市，它都具有稳定的业绩表现。市场中性策略对于拥有大量现金池的机构（银行、保险、公募基金等）最具吸引力。

风险管理

减少因子风险暴露的过程称为 **风险管理**。对冲是在实践中进行风险管理的最佳方式之一。

通过案例来了解如何做到市场风险对冲的

现在我们已经知道要对冲多少市值，让我们看看它如何影响我们的收益。我们使用贵州茅台和基准沪深300来构建我们的投资组合，将沪深300的权重设为 $-\beta$ （由于持有基准空头头寸）

```
# 构建一个市场中性组合
portfolio = -1*beta*r_b + r_a
portfolio.name = "600519.SHA + Hedge"

# 绘制各自的收益曲线
portfolio.plot(alpha=0.9,figsize=[9,6])
r_b.plot(alpha=0.5);
r_a.plot(alpha=0.5);
plt.ylabel("Daily Return")
plt.legend();
```



看来组合（贵州茅台+沪深300）的收益和贵州茅台走势相当接近。我们可以通过计算两者的平均回报率 and 风险（收益率的标准差）来量化其表现的差异

```
print("means: ", portfolio.mean(), r_a.mean())
print("volatilities: ", portfolio.std(), r_a.std())
```

```
means:  0.0011625392362475395  0.002370904665440321
```

```
volatilities:  0.01785176992416382  0.019634943455457687
```

可以看出，我们以收益率为代价降低了波动，在降低风险的同时，收益也相应降低了。接下来，我们来检查一下 α 是否与以前一样，而 β 已被消除

```
P = portfolio.values
alpha, beta = linreg(X,P)
print('alpha: ' + str(alpha))
print('beta: ' + str(beta)) # alpha 和以前仍然一样 beta 已经被消除,beta几乎为0
```

```
alpha: 0.00116253937709
```

```
beta: -1.24623534062e-09
```

请注意，我们使用历史数据构建了 **市场中性策略**。我们可以通过在不同的时间框架内验证资产和对冲投资组合的 α 和 β 值来检查其是否仍然有效

```
# 得到过去一年得到的alpha 和beta值
start_date = '2014-01-01'
end_date = '2015-01-01'
asset = D.history_data('600519.SHA', start_date, end_date, fields=['close']).set_index('date')['close']
benchmark = D.history_data('000300.SHA', start_date, end_date, fields=['close']).set_index('date')['close']
r_a = asset.pct_change()[1:]
r_b = benchmark.pct_change()[1:]
X = r_b.values
Y = r_a.values
historical_alpha, historical_beta = linreg(X, Y)
print('Asset Historical Estimate:')
print('alpha: ' + str(historical_alpha))
print('beta: ' + str(historical_beta))

# 获取下一年的数据:
start_date = '2015-01-01'
end_date = '2015-06-01'
asset = D.history_data('600519.SHA', start_date, end_date, fields=['close']).set_index('date')['close']
benchmark = D.history_data('000300.SHA', start_date, end_date, fields=['close']).set_index('date')['close']
asset.name = '600519.SHA'
benchmark.name = '000300.SHA'
# 重复前面的过程来计算alpha 和beta值
r_a = asset.pct_change()[1:]
r_b = benchmark.pct_change()[1:]
X = r_b.values
Y = r_a.values
alpha, beta = linreg(X, Y)
print('Asset Out of Sample Estimate:')
print('alpha: ' + str(alpha))
print('beta: ' + str(beta))

# 构建对冲投资组合来计算alpha、beta
portfolio = -1*historical_beta*r_b + r_a
P = portfolio.values
alpha, beta = linreg(X, P)
print('Portfolio Out of Sample:')
print('alpha: ' + str(alpha))
print('beta: ' + str(beta))
```



```
# 绘制图形
portfolio.name = "600519.SHA + Hedge"
portfolio.plot(alpha=0.9,figsize=[9,6])
r_a.plot(alpha=0.5);
r_b.plot(alpha=0.5)
plt.ylabel("Daily Return")
plt.legend();
```

Asset Historical Estimate:

alpha: 0.00116253939056

beta: 0.672934653004

Asset Out of Sample Estimate:

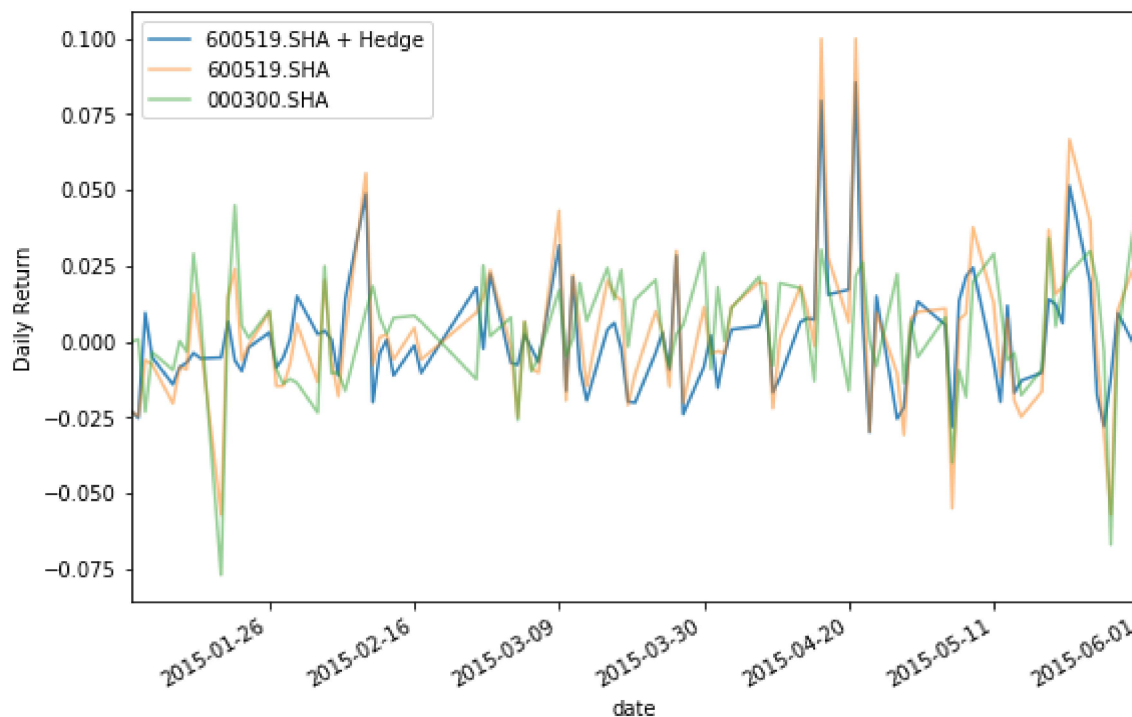
alpha: 0.00020366206079

beta: 0.866552969103

Portfolio Out of Sample:

alpha: 0.000203662008879

beta: 0.193618313006



从上图可以看出，对冲后的收益降低了，但波动性也降低了。历史估计出的贝塔值在样本外的一年中是有效的，将资产的贝塔值0.673通过对冲降低到了0.193，也就是说降低了2/3，这样的对冲效果是比较明显的，而且也反映出历史的贝塔值是有效的，当然，要做到更好的效果，可以采取滚动估计贝塔的方法。

本文由BigQuant宽客学院推出，版权归BigQuant所有，转载请注明出处。

🔗 社区干货与精选整理（持续更新中...）

