投资咨询业务资格：证监许可【2012】669 号

工具详解系列一：
卡尔曼滤波最优性原理和迭代逻辑

金融工程组：
研究员：

盛博文
从业资格号 F03107915
投资咨询号 Z0021449

孔如玉
从业资格号 F03108272
投资咨询号 Z0021459

报告要点

本篇报告回答了 1 个问题：如果卡尔曼滤波的输入是配对品种的价格序列，为何其输出可以是两者的动态对冲比率和截距修正项。

摘要

在此前专题报告《**配对交易专题（二）：卡尔曼滤波在价差套利中的应用（基于 Backtrader 视角）——专题报告 20240124**》中，我们以日内 30 分钟线为颗粒度，在若干金融期货品种上应用卡尔曼滤波器——基于“配对品种价格序列”的输入，输出配对品种的“动态对冲比率和截距修正项”，目的是结合配对品种的价格序列来生成价差，以此作为价差套利的底层数据，进而适配于各类进出场规则以生成相应建平仓信号。

区别于上述报告中**直接调取高度集成化的 pykalman 安装包**，我们将于本篇报告详细拆解该工具的最优性原理及代码的运行迭代逻辑，具体对应如下要点的探讨：卡尔曼增益为何是其最优性原理的最终“落地”结果？卡尔曼滤波基于前后 2 个时点的状态拟合如何代码实现？

风险因子：本报告中所涉及的资产配比和模型应用仅为回溯举例，并不构成推荐建议。

目录

图表目录

一、卡尔曼滤波应用与配对交易

配对交易作为一种经典的统计套利策略，其通常的做法是去观察历史回溯区间中走势较为相近的金融资产价格，以捕捉两者在未来出现异常偏离时候的套利机会。它的本质是做多低估品种、做空高估品种，而“走势较为相近”可以使用“满足协整关系”进行具体刻画。

比如 2024 年上半年，权益市场相继在年初遭遇微小盘流动性踩踏和二季度受到“国九条”规范，整体缺乏明确的单边行情，彼时套利类策略受到了市场的关注，其中就包括大小盘轮动或配对。在针对(IF, IM)的配对中，如果 IF 与 IM 的价格震荡中枢分别是 3000 点和 5000 点，在市值中性的要求下可以得知 1 份的 IM 应当对应 1.6 份的 IF。粗略地讲，这里“1.6 倍”的对冲比率中枢是可以基于 IF 与 IM 历史价格走势、通过线性回归（即“协整法”）实现，然而协整法使用的前提——“满足协整关系”并不被大部分金融时间序列所满足。这将使得虽然线性回归可以“硬算”，但是会出现伪回归或者后续均值回归的逻辑不成立。

基于上述背景，**不需要协整关系作为前提**、仅依赖前后两个状态的卡尔曼滤波便是上述线性回归较好的替代（相关文献可参考 **a new approach to linear filtering and prediction problems**, Rudolf Emil Kalman, 1960），python 中也有高度集成的安装包（如 pykalman）。下图依次展示了类 KalmanFilter 的调用语句以及卡尔曼滤波应用于(T, TS)、(IF, IM)和(IF, IC)主连 30min 线的结果——红线对应动态对冲比率、灰线对应截距修正，通常我们以 **“副腿价格-动态对冲比率*主腿价格-截距修正项”** 作为价差定义。

图表1：类 KalmanFilter 直接调用释义

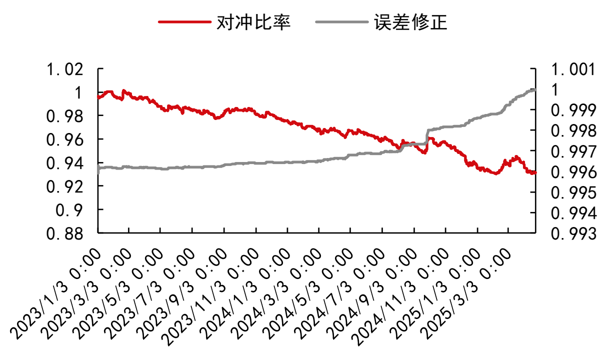
```python
from pykalman import KalmanFilter

kf = KalmanFilter(n_dim_obs,n_dim_state,
    initial_state_mean,initial_state_covariance,
    transition_matrices,observation_matrices,
    observation_covariance,transition_covariance)

state_means, state_covs = kf.filter(...)
```

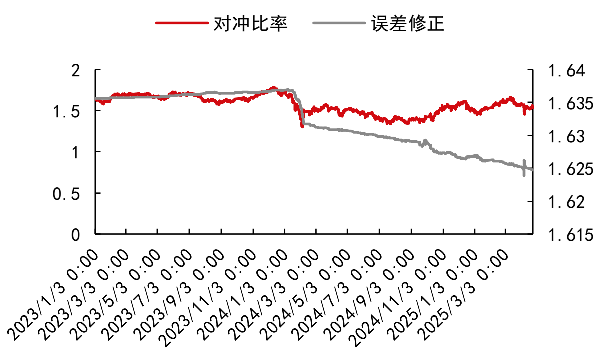
资料来源：中信期货研究所

图表2：卡尔曼滤波应用于 T 与 TS 配对交易



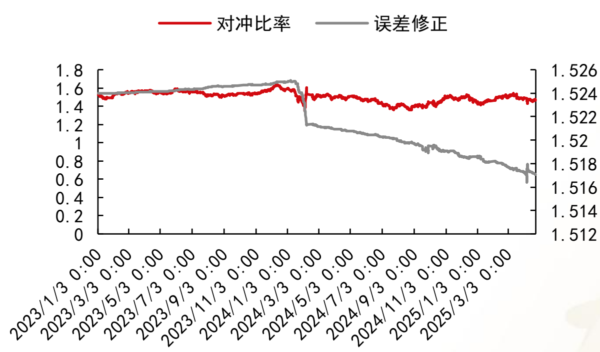
资料来源：同花顺 iFind、中信期货研究所

图表3：卡尔曼滤波应用于 IF 与 IM 配对交易



资料来源：同花顺 iFind、中信期货研究所

图表4：卡尔曼滤波应用于 IF 与 IC 配对交易



资料来源：同花顺 iFind、中信期货研究所

在此前专题报告**《配对交易专题（二）：卡尔曼滤波在价差套利中的应用（基于 Backtrader 视角）——专题报告 20240124》**中，大家较为关注直接调用函数 KalmanFilter.filter()的底层逻辑：如果卡尔曼滤波的输入是配对品种的价格序列，为何其输出可以是两者的动态对冲比率和截距修正项。

接下来我们将对此从两方面进行详细分析——卡尔曼滤波的最优性原理和卡尔曼滤波循环迭代逻辑。

二、卡尔曼滤波的相关理论

（一）应用场景与迭代细节

1. 问题背景：引入一个线性系统状态空间模型

考虑如下离散时间线性系统：

$$x_k = Ax_{k-1} + Bu_k + \omega_k, \omega_k \sim \mathcal{N}(0, Q), \tag{1.1}$$

$$z_k = Hx_k + v_k, v_k \sim \mathcal{N}(0, R), \tag{1.2}$$

其中：x_k表示系统的状态向量（如位置、速度），A表示状态转移矩阵，B表示控制输入矩阵，u_k表示控制输入，ω_k表示过程噪声，z_k表示测量值（如雷达高度计测量得到的位置），H表示测量矩阵，v_k表示测量噪声。

目的：我们希望找到一个对状态向量x_k的估计值$\hat{x}_k$，使得真实值与估计值间的误差最小，这里衡量误差最小的模式将采用“均方误差”进行刻画。

应用：落地到跨品种套利中，区分于经典的使用配对品种收盘价的历史时间序列去做线性回归，此过程中配对的**动态对冲比率**及**截距修正项**都可以作为上述离散系统的状态向量进行估计。

2. 循环迭代 1：预测（predict）——使用上一时刻的状态估计值来预测当前状态

本部分主要量化“**在没有测量数据时，我们对状态估计的不确定性**”。

假设在$k-1$时刻，我们已经得到对系统状态向量x_{k-1}的估计值$\hat{x}_{k-1}$（注意，回测最初需要对初始状态向量x_0及**后续若干变量**进行人为初始化），同时该估计值$\hat{x}_{k-1}$具有误差

$$\epsilon_{k-1} =_{(def)} x_{k-1} - \hat{x}_{k-1} \,,$$

及与其相对应的均方误差

$$MSE_{k-1} =_{(def)} E[\epsilon_{k-1}\epsilon_{k-1}^T] \ .$$

来到k时刻，我们利用（当前可获取的）上一时刻针对系统状态向量的估计值$\hat{x}_{k-1}$来预测当前的状态向量x_k，该新估计值对应

$$\hat{x}_k^- =_{(def)} A\hat{x}_{k-1} + Bu_k, \tag{2.1}$$

同时，计算当前这个（暂时的）新估计值$\hat{x}_k^-$的均方误差

$$\begin{aligned} MSE_k^- &= E[\epsilon_k^-(\epsilon_k^-)^T] = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] \\ &= \mathrm{E}[(Ax_{k-1} + Bu_k + \omega_k - A\hat{x}_{k-1} - Bu_k)(Ax_{k-1} + Bu_k + \omega_k - A\hat{x}_{k-1} - Bu_k)^T] \quad (\text{使用}(1.1) + (2.1)) \\ &= \mathrm{E}[(A(x_{k-1} - \hat{x}_{k-1}) + \omega_k)(A(x_{k-1} - \hat{x}_{k-1}) + \omega_k)^T] = AMSE_{k-1}A^T + Q, \end{aligned} \tag{2.2}$$

最后一个等号成立是因为$x_{k-1}-\hat{x}_{k-1}$与ω_k是**相互独立的**、且后者服从均值为 0 的正态分布，我们也把(2.2)称为均方误差矩阵的更新。

3. 循环迭代 2：更新（correct/update）

本部分想要实现：在k时刻，基于上述阶段性结果、**再额外得到新的测量数据**（即z_k，注意该值对应配对中副腿在k时刻的价格、属于传入数据）后，我们构造一个新的残差项$(z_k-H\hat{x}_k^-)$，并希望寻找得到卡尔曼增益K_k，能使得如下形式的新估计值

$$\hat{x}_k =_{(def)} \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{3.1}$$

对k时刻真实的状态向量x_k的均方误差MSE_k最小。

我们具体给出这个误差及均方误差的表达式：

$$\epsilon_k = x_k - \hat{x}_k =_{(3.1)} x_k - \hat{x}_k^- - K_k(z_k - H\hat{x}_k^-),$$

将（1.2）式代入上式，

$$\begin{aligned}\epsilon_k &= x_k - \hat{x}_k =_{(3.1)} x_k - \hat{x}_k^- - K_k(z_k - H\hat{x}_k^-)\\ &=_{(1.2)} x_k - \hat{x}_k^- - K_k(Hx_k + \nu_k - H\hat{x}_k^-)\\ &= (x_k - \hat{x}_k^-) - K_kH(x_k - \hat{x}_k^-) - K_k\nu_k\\ &= (I - K_kH)(x_k - \hat{x}_k^-) - K_k\nu_k ,\end{aligned}$$

然后，上式两边取期望并同时计算均方误差，

$$\begin{aligned}MSE_k = E[\epsilon_k\epsilon_k^T] &= E\{[(I-K_kH)(x_k - \hat{x}_k^-) - K_k\nu_k][(I-K_kH)(x_k - \hat{x}_k^-) - K_k\nu_k]^T\}\\ &= (I-K_kH)MSE_k^-(I-K_kH)^T + K_kRK_k^T,\end{aligned} \tag{3.2}$$

最后一个等号成立是因为$x_k - \hat{x}_k^-$与ν_k是相互独立的，且后者服从均值为 0 的正态分布，我们也把（3.2）称为均方误差矩阵的更新。

要使得MSE_k取得最小值，只需对K_k求导即可：

$$\frac{\partial MSE_k}{K_k} = -2MSE_k^-H^T + 2K_k(HMSE_k^-H^T + R),$$

令上式为零，即可得到：

$$K_k = MSE_k^-H^T(HMSE_k^-H^T + R)^{-1}, \tag{3.3}$$

此即卡尔曼增益。

（二）要点提炼

我们把上述推导精炼提取为如下“两两融合”的概念。对于**离散线性系统**：

$$x_k = Ax_{k-1} + Bu_k + \omega_k, \omega_k \sim \mathcal{N}(0,Q),$$

$$z_k = Hx_k + \nu_k, \nu_k \sim \mathcal{N}(0,R),$$

卡尔曼滤波基于最小均方误差的原理，结合预测和更新两个步骤，持续修正对于状态变量的估计，进而达到精度提升的目的。

状态预测（当前状态的预测、均方误差的更新）：

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k,$$
$$MSE_k^- = AMSE_{k-1}A^T + Q,$$

步骤更新（卡尔曼增益的计算、当前状态的更新、均方误差的更新）：

$$K_k = MSE_k^- H^T (HMSE_k^- H^T + R)^{-1},$$
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-),$$
$$MSE_k = (I - K_kH)MSE_k^-(I - K_kH)^T + K_kRK_k^T.$$

三、卡尔曼滤波的应用举例

（一）一维单变量

由浅入深，先给出一维单变量卡尔曼滤波的例子（如金融时间序列的价格走势、物体匀速直线运动等）。

图表5：一维单变量卡尔曼滤波代码（部分）示例

```
class KalmanFilter:
    def __init__(self, A, B, H, Q, R, M, x):
        """初始化卡尔曼滤波器"""
        self.A = A   # 状态转移矩阵
        self.B = B   # 控制矩阵
        self.H = H   # 观测矩阵
        self.Q = Q   # 过程噪声协方差
        self.R = R   # 观测噪声协方差
        self.M = M   # 估计均方误差
        self.x = x   # 初始状态

    def predict(self, u=np.zeros((1,1))):
        """ 预测步骤 """
        self.x = np.dot(self.A, self.x) + np.dot(self.B, u)  # 预测状态
        self.M = np.dot(np.dot(self.A, self.M), self.A.T) + self.Q  # 预测均方误差
        return self.x

    def update(self, z):
        """ 更新步骤 """
        K = np.dot(np.dot(self.M, self.H.T), np.linalg.inv(np.dot(self.H, self.M).dot(self.H.T) + self.R))
        self.x = self.x + np.dot(K, (z - np.dot(self.H, self.x)))  # 更新状态估计
        self.M = np.dot((np.eye(self.M.shape[0]) - np.dot(K, self.H)), self.M)  # 更新均方误差
        return self.x

A = np.array([[1]])  # 状态转移矩阵
B = np.array([[0]])  # 控制矩阵
H = np.array([[1]])  # 观测矩阵
Q = np.array([[1e-5]])  # 过程噪声协方差
R = np.array([[1e-2]])  # 观测噪声协方差
M = np.array([[1]])  # 估计均方误差
x = np.array([[0]])  # 初始状态
```

资料来源：中信期货研究所

从上例中，我们对卡尔曼滤波有以下几点发现：**较强的递归性**——只依赖于当前状态更新，无需保留全部历史数据；**估计的最优性**——在状态系统线性且服从（多维）正态分布的前提下，能够达到“均方误差最小”意义上的最优状态估计。

即便本例只涉及一维单变量，但我们也可预感到其计算**较大程度依赖于矩阵计算**。为了对此有进一步的感知，我们也给出含二元状态向量离散系统的例子。

（二）二元变量

本例涉及两个状态变量——参数β和参数ε（这里暂不对其赋予具体含义，注意后者区分于前述误差ϵ），假设仅前者可以观测、后者没有测量值，我们使用卡尔曼滤波对此两者进行估计。

图表6：二元变量卡尔曼滤波（部分）示例

```
A = np.array([[1, dt],[0, 1]])   # 状态转移矩阵
H = np.array([[1, 0]])   # 观测矩阵
Q = np.array([[1e-4, 0],[0, 1e-4]])   # 过程噪声协方差
R = np.array([[0.01]])   # 测量噪声协方差
x = np.array([[1], [0]])  # 初始状态估计：对冲比率 1，截距修正 0
M = np.eye(2)  # 初始均方误差

# 模拟真实运行轨迹（略）

for _ in range(N):
    # 模拟真实运行 + 噪声
    x_true = A @ x_true + np.random.multivariate_normal([0, 0], Q).reshape(-1, 1)
    z = H @ x_true + np.random.normal(0, np.sqrt(R[0, 0]), (1, 1))

    true_x.append(x_true.flatten())
    measurements.append(z.item())

    # ==== 卡尔曼滤波（注意上下唯一有联系的地方在于变量 z） ====
    # 预测
    x_pred = A @ x    # 注意这个 x 并非每轮循环保持不变，它在下方是会更新的
    M_pred = A @ M @ A.T + Q

    # 卡尔曼增益
    K = M_pred @ H.T @ np.linalg.inv(H @ M_pred @ H.T + R)

    # 更新
    x = x_pred + K @ (z - H @ x_pred)
    M = (np.eye(2) - K @ H) @ M_pred
```

资料来源：中信期货研究所

从上述卡尔曼滤波最优性证明中可以得知，在每个时刻 k 我们都致力于寻找最优的卡尔曼增益K_k，希望使得均方误差MSE_k最小。

当系统包含多维（≥ 2）向量需要估计时，MSE_k成为一个矩阵。由于矩阵本身没有“大小”，所以我们需要选择一种方式对它进行度量（就好比我们前面用“均方误差最小”来量化“估计值对真实值最好的观察”这一表述），最常见的做法就是使用矩阵MSE_k的迹。一方面是因为MSE_k的对角线元素对应每个状态变量的误差的方差，另一方面是因为它的迹（每个状态变量误差方差的和）等价于每个状态变量的误差的最小二范数，即

$$\min_{\mathrm{k}} E[\epsilon_k^2] = \min_{\mathrm{k}} tr(MSE_k) = \min_{k} tr(\epsilon_k \epsilon_k^T).$$

（三）理解 pykalman 包的 KalmanFilter 类

在 KalmanFilter 这个类中，核心是封装函数_fiter()。该函数的内部设计结构清晰，主要拆分为调用两部分：负责“当前状态预测、均方误差更新”的函数_filter_predict()和负责“卡尔曼增益的计算、当前状态的更新、均方误差的更新”的函数_filter_correct()。

封装函数_fiter()中包含了针对全体时刻的 for 循环，每一步循环中完成上述“预测阶段”和“校正阶段”，算作完整的一次更新。

以下给出上述 3 个关键函数中若干变量名与卡尔曼最优性原理公式中若干符号的对应关系。

图表7：KalmanFilter 类中涉函数_filter()若干变量释义

变量名	公式对应字母	首发归属函数	其他
transition_matrices	A	__init__，需初始化	/
observation_matrices	H	__init__，需初始化	单腿期货价作基准（如 IF）
transition_covariance	Q	__init__，需初始化	/
observation_covariance	R	__init__，需初始化	/
transiton_offsets	ω_k	__init__，需初始化	/
observation_offsets	ν_k	__init__，需初始化	/
initial_state_mean	np.zeros(2)	__init__，需初始化	/
initial_state_covariance	np.ones((2,2))	__init__，需初始化	/
current_state_covariance	MSE_{k-1}	_filter_predict()	/
predicted_state_covariance	MSE_k^-	_filter_predict()	/
predicted_state_mean	$\hat{x}_k^-$	_filter_predict()	配对时具体为$\left(\hat{\beta}_k^-, \hat{\varepsilon}_k^-\right)^T$
predicted_observartion_mean	$H\hat{x}_k^- + \nu_k$	_filter_correct()	$(\mathrm{CLOSE}_{k,IF}, 1) * \left(\hat{\beta}_k^-, \hat{\varepsilon}_k^-\right)^T + \nu_k$
observation	z_k	_filter_correct()	单腿期货价作观测（如 IM）

资料来源： 公开资料、中信期货研究所

基于上述表格，我们可参考如下步骤来理解类 KalmanFilter 的运行先后顺序：

1. 定义 transition_covariance 和 observation_covariance 分别作为预测误差 Q 和观测误差 R（均需初始化），同时将基准合约（如（IF, IM）中的主腿期货 IF）的价格转为三维数组 observation_matrices；
2. 将参数传入类 KalmanFilter，使用函数_determine_dimensionality()来确定每个状态的维度以及观测数据的维度，该函数将基于前述处理结果和传入参数（若有）来最终确定变量 n_dim_state 和 n_dim_obs 的取值；
3. 将传入类 KalmanFilter 的所有参数存储在字典 arguments 中；
4. 将观察合约（如（IF, IM）中的副腿期货 IM）的价格数据 observation、n_dim_state、n_dim_obs 和字典 arguments 等作为参数传入函数 filter()；
5. 使用函数_parse_observations()将 observation 转化为二维数据，并连同字典 arguments 中的所有值作为参数传入_filter()函数；
6. 包含基准合约价格的 observation_matrices 时间序列作为 H 序列，观察合约价格序列为 z 序列，对冲比率和截距修正项作为估计状态变量（需初始化）。

从上述对应图表中，我们对卡尔曼滤波应用于配对交易有如下几点观察：

1. **同一时刻估计的状态向量有两个：**配对交易中，使用卡尔曼滤波估计的系统状态向量在每个时刻同步运行的包含有动态对冲比率β和截距修正项ε（注意区分于前述定义误差所用ϵ）；
2. **“主/副腿价格”与“基准/观察”的一一对应：**以（IF, IM）的配对为例，初始化类 KalmanFilter 时，我们将主腿 IF 价格（拼接全为 1 的列进行形式调整）作为基准项进行针对形参 observation_matrices 的赋值，再在调用该类的函数 filter()时将副腿 IM 价格作为观察项进行针对函数 filter()内置函数_filter()的形参 observations 的赋值；更具象一些，$(CLOSE_{k,IF},1)$与$CLOSE_{k,IM}$分别对应上述（3.1）式中的 H与z_k；
3. **循环迭代过程无需系统状态向量β和ε的真实值：**卡尔曼滤波的最优性证明最终是对真实值与估计值的均方误差关于卡尔曼增益求一阶导，令其为零，从而得到卡尔曼增益的具体表达式；但在实际迭代过程中，我们**无需也无法**得到状态向量β和ε的真实值，在“最优性”得以证明后，每一步迭代就是使用卡尔曼增益作为系数将基于“预测”和“修正”的结果两两融合，得到的结果就是“最贴近”β和ε的（未知）真实值。

四、总结

详细了解上述“最优性原理及循环迭代”后，返璞归真，我们理解卡尔曼滤波仅需记住这样一个例子：

假设某座高山海拔为$\boldsymbol{y}$，当前存在两种观测方式如下——雷达高度计测量得$\boldsymbol{y_1}$和气压计测量得$\boldsymbol{y_2}$，这两个观测值皆有误差——$\boldsymbol{bias_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)}$ 和 $\boldsymbol{bias_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)}$。我们希望通过权重因子$K$进行组合$\hat{\boldsymbol{y}} =_{(def)} \boldsymbol{y_1} + K(\boldsymbol{y_2} - \boldsymbol{y_1})$来达到对真实值$\boldsymbol{y}$的最优估计。

基于上述背景，我们采用最小方差的方式，即将$var(\hat{\boldsymbol{y}} - \boldsymbol{y})$关于其自变量$K$求导并令其为零，得到$K$并代入得到$\hat{\boldsymbol{y}}$，我们也可以简单地证明后者相较于$\boldsymbol{y_1}$和$\boldsymbol{y_2}$而言，与真实值$\boldsymbol{y}$的误差来得“更小”、估计更准确。

免责声明

中信期货有限公司

深圳总部

地址：深圳市福田区中心三路 8 号卓越时代广场（二期）北座 13 层 1301-1305、14 层

邮编：518048

电话：400-990-8826

传真：(0755)83241191

网址：http://www.citicsf.com