

第一个问题：理解不同，修改在

```
# %% feat generate

def ACF_func(s, lag=5):
    n = s.shape[0]

    def get_corr(s1, s2):
        return np.corrcoef(s1, s2)[0, 1]

    s_acf = np.array([get_corr(s[i - 2 * lag: i - lag], s[i - lag: i])
                      for i in range(2 * lag, n)])
    return s_acf
```

此处修改 $3 * s_acf$ 即可

第二个问题：

$$\begin{aligned} 0 < (D_i - D_{i-1}) < \theta \cdot D_i \\ 0 < (D_i - D_{i-1}) / D_i < \theta \end{aligned} \quad (1)$$

定义 $\hat{\theta} = (D_i - D_{i-1}) / D_i$ ，则 $0 \leq \hat{\theta} \leq \theta$

上述两式等价，与代码描述一致

```
# 扩散关系  $0 < \text{value} < \theta$ 
df['sig_theta'] = (df['difuse'] - df['difuse'].shift(1)) / df['difuse']
# 状态描述 在方差范围内
df['sig_F'] = (df['close'] - df['mu']).apply(np.abs) / df['sigma']

# 超参数可调
def get_buy_signal(df, acf=0.7, F=1.4, theta=0.1):
    df['sig_o1'] = df['acf'] < acf
    df['sig_o2'] = (df['sig_theta'] > 0) & (df['sig_theta'] < theta)
    df['sig_o3'] = df['sig_F'] < F
    return df
```

第三个问题：同理， $(D_i - D_{i-1}) \leq 0$ 等价于 $\hat{\theta}$ 小于 0

```
def get_sell_signal(df):
    df['sig_s1'] = df['sig_theta'] < 0
    return df
```

第四个问题：

(原文翻译-fast部分)

4. 通过三个条件描述了一种新的平衡态。第一个条件对应于反转- $P > u$ ，(看跌转换后)， $P < u$ (看涨转换后)。第二个条件表示波动和扩散压缩的减少。最后，必须通过以下简短的记忆条件来验证运输模型的合格性： $ACF(TO, T0) \ll ACF X$ 。

由此，与之前满足的两个条件一致。我这边疏忽了，需要去掉第三个条件，已更正。

```
74
75 def get_sell_signal(df, acf=0.7, F=1.3):
76     df['sig_s1'] = df['acf'] < acf
77     df['sig_s2'] = df['difuse'] < 0
78     df['sig_s3'] = df['sig_F'] < F
79     return df
80
```

删除