

Strategy_7: (..\策略回测\VIX\strategy_7)

参考研报:

20210913-中信期货-基于铜期权隐含波动率的交易策略研究

测试1: 基于平值期权变化 (test_atm_no_position_control_check_n)

测试平值期权检视周期: 5, 6, 7, 8

交易信号:

每周期判断一次平值期权是否发生变化, 发生变化产生入场信号, 卖出看涨和看跌期权

仓位管理

使用VIX_CALL, VIX_PUT (test_atm_position_control_check_n):

使用VIX (test_atm_position_control_check_n_use_vix)

- 调仓周期为5, 即每隔5天进行下述的波动率判断, 进行调仓

原研报:

当铜期权波动率 $\leq 18\%$ 时, 仓位为原来的 1 倍;

当 $18\% < \text{铜期权波动率} \leq 22\%$ 时, 仓位为原来的 1.5 倍;

当 $22\% < \text{铜期权波动率} \leq 28\%$ 时, 仓位为原来的 2 倍;

当 $28\% < \text{铜期权波动率}$, 仓位为原来的2.5倍

适应不同品种, 设置波动率的布林带:

- 上轨: 波动率均值 + 一单位标准差; 向上突破上轨, 仓位为原来的2.5倍
- 中轨: 波动率均值; 向上突破中轨, 仓位为原来的2倍
- 下轨: 波动率均值 - 一单位标准差; 向上突破下轨, 仓位为原来的1.5倍

```
def _calc_change_atm_signal(self, data, check_time):  
  
    # data['quote_date'] = pd.to_datetime(data['quote_date'])  
    df_merge = pd.merge(  
        data, self.atm_opts_symbol[['quote_date', 'symbol_change_1']], on=['quote_date']  
    )  
  
    df_merge['signal'] = 0  
    df_merge['out'] = 0  
    count = 0  
    for i in range(len(df_merge)):  
        if count == check_time:  
            if df_merge.loc[i, 'symbol_change_1'] == 1:  
                df_merge.loc[i, 'signal'] = 1  
            else:  
                df_merge.loc[i, 'signal'] = 0  
                count = 0  
            count += 1  
  
    return df_merge
```

测试2：基于均线的趋势跟踪 (test_ma)

交易信号：

。看涨期权：

- 标的收盘价的短均线下穿长均线，卖出看涨期权
- 其余时间持仓

。看跌期权：

- 标的收盘价的短均线上穿长均线，卖出看跌期权
- 其余时间持仓

仓位管理：

使用VIX (test_ma_position_control_use_vixuse_vix)

使用VIX_CALL, VIX_PUT (test_ma_position_control)

- 调仓周期为5，即每隔5天进行下述的波动率判断，进行调仓

适应不同品种，设置波动率的布林带：

- 上轨：波动率均值 + 一单位标准差；向上突破上轨，仓位为原来的2.5倍
- 中轨：波动率均值；向上突破中轨，仓位为原来的2倍
- 下轨：波动率均值 - 一单位标准差；向上突破下轨，仓位为原来的1.5倍

2 usages

@staticmethod

```
def calc_in_ma(data, signal_type, long_ma_col, short_ma_col):

    data['signal'] = 0
    data['out'] = 0

    for i in range(2, len(data)):

        if signal_type == 'put':

            if ((data.loc[i-1, short_ma_col] < data.loc[i-1, long_ma_col])
                & (data.loc[i, short_ma_col] > data.loc[i, long_ma_col])):
                data.loc[i, 'signal'] = 1

            else:
                continue

        if signal_type == 'call':

            if ((data.loc[i - 1, short_ma_col] > data.loc[i - 1, long_ma_col])
                & (data.loc[i, short_ma_col] < data.loc[i, long_ma_col])):
                data.loc[i, 'signal'] = 1

            else:
                continue

    return data[['signal', 'out']]
```