

▶ 赛题：股指期货高频交易

股指期货作为金融衍生品的核心工具，其高频预测与交易能力广泛应用于各类策略，如对冲，套利，择时等。在股票策略领域，可以利用其进行系统性风险动态对冲，基差管理，暴露优化等。

同时，股指期货的高频预测和交易是一项非常有挑战的任务。存在噪声大，信号弱，微观结构复杂，实时性要求高，成本敏感，模型容易过拟合等诸多问题。解决这些问题，将对各类策略的表现有显著的增益。

欢迎有能力的你加入信弘股指期货高频交易大赛！

X ASSET
MANAGEMENT

数据集：比赛提供股指期货高频行情数据（L2数据）

数据存储方式

```
./train      ./test
20220722    20241008
20220725    ...
20220726    20241129
...
20240930
```

./train/20220722

```
IC_2208_L1_M.csv  IC_2212_L3.csv  IC_2209_L2.csv  IC_2303_L4.csv
IF_2208_L1_M.csv  IF_2212_L3.csv  IF_2209_L2.csv  IF_2303_L4.csv
IH_2208_L1_M.csv  IH_2212_L3.csv  IH_2209_L2.csv  IH_2303_L4.csv
IM_2208_L1_M.csv  IM_2212_L3.csv  IM_2209_L2.csv  IM_2303_L4.csv
```

数据说明

- 数据分为训练集和测试集，分别按照以交易日命名的文件夹存储；按照交易日区分样本内外；
- 每天的数据按照合约保存为多个csv文件，csv文件命名规则: 标的_交割月_交割月顺序_是否主力(标记M为主力合约)；
- 交割月顺序: L1-L4分别为近月/次月/次季月/远季月合约；

数据集：行情数据示例

例 ./train/20220722/IC_2208_L1_M.csv

	TRADINGTIME	SYMBOL	OPENPRICE	LASTPRICE	HIGHPRICE	LOWPRICE	SETTLEPRICE	PRESETTLEPRICE	CLOSEPRICE	PRECLOSEPRICE	...	CHANGERATIO	CONTINUESIGN	POSITIONCHANGE	AVERAGEPRICE	ORDERRATE	ORDERDIFF	AMPLITUDE
0	2022-07-22 09:29:00.400	IC2208	6328.2	6328.2	6328.2	6328.2	0.0	6337.0	0.0	6310.2	...	-0.0014	ICL1	0.0	6328.200	0.1852	20.0	0.0000
1	2022-07-22 09:30:00.400	IC2208	6328.2	6327.2	6329.0	6325.0	0.0	6337.0	0.0	6310.2	...	-0.0015	ICL1	-17.0	6328.160	0.5942	41.0	0.0006
2	2022-07-22 09:30:00.900	IC2208	6328.2	6327.4	6329.0	6325.0	0.0	6337.0	0.0	6310.2	...	-0.0015	ICL1	-20.0	6328.126	-0.3889	-14.0	0.0006
3	2022-07-22 09:30:01.400	IC2208	6328.2	6327.0	6329.0	6325.0	0.0	6337.0	0.0	6310.2	...	-0.0016	ICL1	-30.0	6327.992	0.6216	46.0	0.0006
4	2022-07-22 09:30:01.900	IC2208	6328.2	6327.0	6329.0	6325.0	0.0	6337.0	0.0	6310.2	...	-0.0016	ICL1	-34.0	6327.859	0.5506	49.0	0.0006
...
28043	2022-07-22 14:59:58.900	IC2208	6328.2	6245.2	6347.0	6176.6	0.0	6337.0	0.0	6310.2	...	-0.0145	ICL1	1.0	6259.509	0.0000	0.0	0.0269
28044	2022-07-22 14:59:59.400	IC2208	6328.2	6245.2	6347.0	6176.6	0.0	6337.0	0.0	6310.2	...	-0.0145	ICL1	0.0	6259.509	-0.0435	-1.0	0.0269
28045	2022-07-22 14:59:59.900	IC2208	6328.2	6244.8	6347.0	6176.6	0.0	6337.0	0.0	6310.2	...	-0.0145	ICL1	1.0	6259.507	-0.2381	-5.0	0.0269
28046	2022-07-22 15:00:00.400	IC2208	6328.2	6244.8	6347.0	6176.6	0.0	6337.0	6244.8	6310.2	...	-0.0145	ICL1	0.0	6259.507	-0.2727	-6.0	0.0269
28047	2022-07-22 15:29:23.900	IC2208	6328.2	6244.8	6347.0	6176.6	6233.2	6337.0	6244.8	6310.2	...	-0.0145	ICL1	0.0	6259.507	-0.2727	-6.0	0.0269

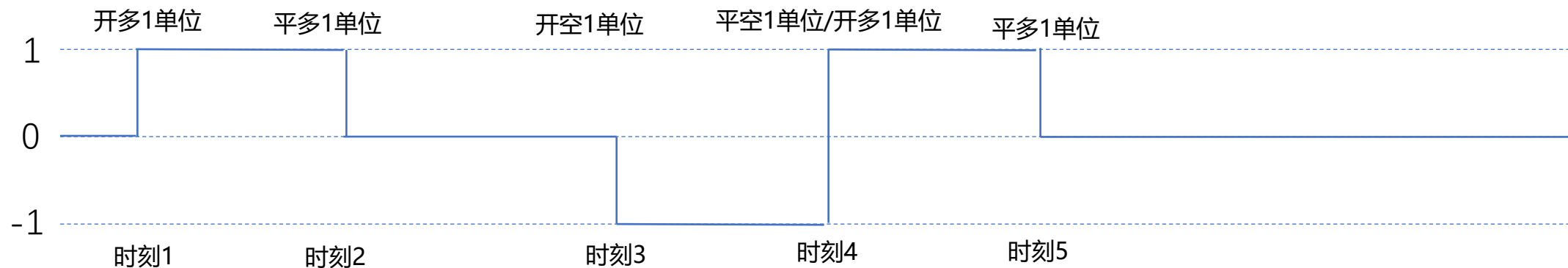
28048 rows × 56 columns

1. I2行情数据包含交易时间戳，标识，盘口信息，价量数据等特征。数据类型包含datetime，str，int，float;
2. 具体的字段说明见“字段说明.csv”文档;
3. 数据集中不包含预先构造好的label，label需各参赛队自行构造;

交易规则说明

在测试期间，每个交易日内，每个可交易时间点上，对每个主力合约标的（IC，IF，IH，IM），可以在多或空方向上最多持仓1个单位。为简化交易规则，当仓位变化即进行开仓/平仓操作时，会使用主力合约上下一个Tick的对手方1档价格作为成交价。开仓/平仓操作的手续费皆按照0.23‰（万分之0.23）计算。每个交易日开收盘时仓位为须为0，如果收盘时仓位不为0，会在收盘时进行强制平仓。

交易序列示例



各参赛队利用提供的股指期货行情数据构建交易策略，策略类型不限。策略目标是生成测试交易期间，每个交易日4种股指期货主力合约对应的全天仓位。各参赛队须提交策略代码，并保证策略可在测试环境正常运行。

策略要求

1. 策略只生成每天的主力合约对应的仓位；
 2. 策略生成的仓位文件命名需与./test文件夹下对应的行情文件名对齐；
 3. 生成的仓位时间戳须与对应的行情时间戳对齐；
 4. 每个时间戳上的合约仓位取值为[0,1,-1]，分别对应空仓，持有一个单位的多头，持有一个单位的空头；仓位变化对应交易发生；
 4. 每个交易日起始和结束时间戳上的仓位回测时置为0；
 6. 策略生成仓位时，只能使用到当前时间戳为止的行情数据，向前最多可使用前一天的行情数据，可以使用到其它合约的数据，包括全部主力/非主力合约；
- 策略可使用gpu，注意提交策略时在测试环境调试通过；
7. 当前./test文件夹在最终测试环节会被替换；
 8. 测试交易日定义为./test文件夹下第2个交易日起的全部交易日；
 9. 运行策略，保存结果至./positions文件夹；
 10. 比赛提供示例策略代码，供各参赛队提交任务参考。

结果提交格式说明

结果提交格式示例

行情数据格式

./test/20241009

```
IC_2410_L1_M.parquet IF_2410_L1_M.parquet IH_2410_L1_M.parquet IM_2410_L1_M.parquet
IC_2411_L2.parquet IF_2411_L2.parquet IH_2411_L2.parquet IM_2411_L2.parquet
IC_2412_L3.parquet IF_2412_L3.parquet IH_2412_L3.parquet IM_2412_L3.parquet
IC_2503_L4.parquet IF_2503_L4.parquet IH_2503_L4.parquet IM_2503_L4.parquet
```

./test/20241009/IC_2410_L1_M.parquet

	SYMBOL	OPENPRICE	LASTPRICE	HIGHPRICE	LOWPRICE	SETTLEPRICE	PRESET
TRADINGTIME							
2024-10-09 09:29:00.000	IC2410	6013.0	6013.0	6013.0	6013.0	0.0	
2024-10-09 09:30:00.500	IC2410	6013.0	6008.0	6016.0	6000.0	0.0	
2024-10-09 09:30:01.000	IC2410	6013.0	6000.8	6016.0	5993.0	0.0	
2024-10-09 09:30:01.500	IC2410	6013.0	5992.8	6016.0	5990.0	0.0	
2024-10-09 09:30:02.000	IC2410	6013.0	5990.0	6016.0	5985.8	0.0	
...	
2024-10-09 14:59:58.500	IC2410	6013.0	5792.2	6108.2	5690.0	0.0	
2024-10-09 14:59:59.000	IC2410	6013.0	5792.8	6108.2	5690.0	0.0	
2024-10-09 14:59:59.500	IC2410	6013.0	5792.8	6108.2	5690.0	0.0	
2024-10-09 15:00:00.000	IC2410	6013.0	5790.8	6108.2	5690.0	0.0	
2024-10-09 15:26:37.000	IC2410	6013.0	5790.8	6108.2	5690.0	5843.8	

28556 rows × 53 columns

对应的仓位保存格式

./position/20241009

```
IC_2410_L1_M.csv IF_2410_L1_M.csv IH_2410_L1_M.csv IM_2410_L1_M.csv
```

./positions/20241009/IC_2410_L1_M.csv

	TRADINGTIME	position
0	2024-10-09 09:29:00.000	0
1	2024-10-09 09:30:00.500	0
2	2024-10-09 09:30:01.000	0
3	2024-10-09 09:30:01.500	0
4	2024-10-09 09:30:02.000	0
...
28551	2024-10-09 14:59:58.500	0
28552	2024-10-09 14:59:59.000	0
28553	2024-10-09 14:59:59.500	0
28554	2024-10-09 15:00:00.000	0
28555	2024-10-09 15:26:37.000	0

28556 rows × 2 columns

- 时间戳对齐
- position取[0,1,-1]

运行./backtest.py文件对预测生成的仓位进行回测，最终使用**夏普比率**作为评估指标；

单笔交易的收益率

$$\text{sign}_i = \begin{cases} +1, \\ -1 \end{cases} = BTO$$

$$\text{pnl}_i = (P_{c,i} - P_{o,i}) \cdot \text{sign}_i$$

$$\text{fee}_i = (P_{o,i} + P_{c,i}) \cdot \text{FEE_RATE}$$

$$\text{net_pnl}_i = \text{pnl}_i - \text{fee}_i$$

$$r_i = \frac{\text{net_pnl}_i}{P_{o,i}}$$

合约k日收益率

$$\text{cum_return}_k = \sum_{i=1} r_i$$

d日收益率

$$R_d = \frac{1}{4} \sum_{j=1}^4 r_{d,j}$$

夏普比率

$$\text{Sharpe} = \frac{\sqrt{N} \bar{R}}{\sigma_R}$$

其中, N = 252

结果评估方式

在测试环境，使用python ./backtest.py 遍历行情和持仓进行回测，生成回测结果。回测结果包含交易明细和收益统计，及策略在回测期间的夏普比率。

交易明细./backtest/20241120/IF_2412_L1_M.csv

	time_o	action_o	price_o	time_c	action_c	price_c	sign	pnl	fee	net_pnl	return	cum_return
0	2024-11-20 10:36:05	STO	3956.6	2024-11-20 11:03:07.500	STC	3977.0	-1	-20.4	0.182473	-20.582473	-0.005202	-0.005202
1	2024-11-20 11:03:08	BTO	3977.2	2024-11-20 15:00:00.000	BTC	3979.8	1	2.6	0.183011	2.416989	0.000608	-0.004594

收益统计./backtest/all_rets.csv

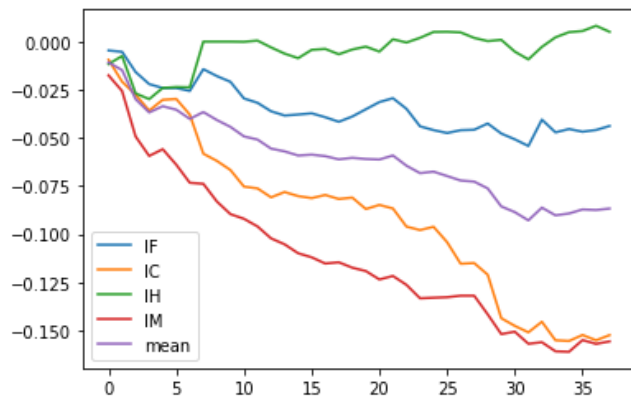
	IF	IC	IH	IM	mean
20241009	0.032501	0.013310	0.033011	0.028400	0.026806
20241010	-0.016476	0.000000	-0.008913	-0.044773	-0.017540
20241011	0.003374	0.002391	0.005795	0.002345	0.003476
20241014	-0.001256	0.009864	-0.008003	0.021246	0.005463
20241015	0.014638	0.008530	0.014588	0.006225	0.010995
20241016	-0.009202	-0.010335	-0.007833	-0.007159	-0.008632
20241017	0.010833	0.009804	0.012454	0.009802	0.010723
20241018	0.032237	0.026804	0.025752	0.033244	0.029510
20241021	-0.010438	-0.011045	0.000000	-0.009337	-0.007705
20241022	-0.001606	0.000302	0.000000	-0.002161	-0.000866
20241023	-0.003449	-0.010431	-0.002541	-0.010771	-0.006798
20241024	0.000005	0.000000	-0.000343	0.000000	-0.000085
20241025	-0.002566	0.000334	-0.003967	0.006927	0.000182
20241028	0.000000	-0.000899	0.000547	0.002442	0.000523
20241029	0.005507	0.004074	0.005297	0.007492	0.005592
20241030	0.005772	-0.005928	0.005666	-0.008451	-0.000735

最终结果

Backtest Result:
annual_ret: 0.7200139970084014
sharpe: 4.117084561254175

随机生成仓位的策略 strategy_2.py

```
def generate_signals(df):  
    df = df.copy()  
    # 随机生成[0, 1, -1]之间的数, 取到0的概率是0.98, 取到1的概率是0.01, 取到-1的概率是0.01  
    df['position'] = np.random.choice([0, 1, -1], size=len(df), p=[0.98, 0.01, 0.01])  
    ## 注意仓位的时间戳和主力合约的时间戳要对齐!  
    return df['position']
```



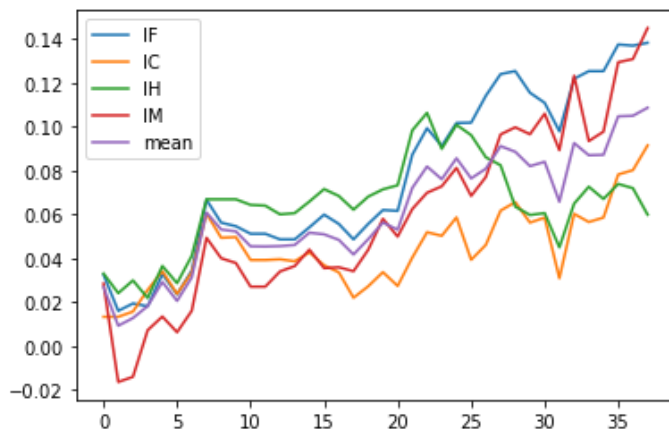
回测结果

Backtest Result:
annual_ret: -1.5589714081981738
sharpe: -56.73633680000261

详见示例策略代码。

一个简单的趋势突破策略 strategy.py

```
def generate_signals_optimized(  
    df: pd.DataFrame,  
  
    window: int = 7200,           # ≈ 60-minute rolling window for local highs/lows  
    ema_period: int = 1080,       # trend filter (≈ 0.15xwindow ≈ 9 min)  
    vol_multiplier: float = 1.8,   # breakout must occur on ≥1.8x平均成交量  
    atr_period: int = 240,        # ATR look-back (≈ 0.033xwindow ≈ 2 min)  
    stop_loss_mult: float = 1.0,   # SL = entry ±1xATR  
    take_profit_mult: float = 3.0, # TP = entry ±3xATR  
    time_stop: int = 21600,       # flat exit after 3xwindow bars (≈ 3 hr)  
    use_obi: bool = True,         # 默认启用盘口不平衡因子  
    obi_threshold: float = 0.15,  # |OBI| 阈值  
  
    bid_col: str = "BUYVOLUME01",  
    ask_col: str = "SELLVOLUME01",  
    high_col: str = "HIGHPRICE",  
    low_col: str = "LOWPRICE",  
    close_col: str = "LASTPRICE",  
    vol_col: str = "TRADEVOLUME",  
)-> pd.Series:  
    """High-frequency breakout strategy with trend/VWAP/OBI filters & ATR risk control.  
  
    Returns  
    -----  
    pd.Series  
        Position series aligned with *df* (1=long, -1=short, 0=flat).  
    """
```



回测结果

Backtest Result:
annual_ret: 0.7200139970084014
sharpe: 4.117084561254175

- 在训练周期内，各参赛队在测试环境中运行各自的策略脚本，对当前的测试集生成最终仓位，将策略脚本和对应生成仓位结果保存在各自根目录的 ./final_submit 文件夹下；
- 最终提交的 ./final_submit 须包含：
 - 1.strategy.py 策略脚本
 - 2. ./positions 当前测试集生成仓位
 - 2.策略其它依赖项（可选）
 - 3.必要的运行说明（可选）

- future_L2 行情数据
 - train 训练集行情数据
 - test 测试集行情数据
- becktest.py 回测脚本
- strategy.py 预测脚本（交易策略）示例
- 字段说明.csv 行情文件字段说明

以上内容会在比赛开始时（7月1日）发布