

📖 目录

📁 收藏

- 策略思路
- 行业轮动
- 优化

🔒 锁定 💬 交流

大小盘轮动和行业轮动策略



周嘉俊 (当前离线)

📄 关注

发布于 2021-06-26 12:36 (编辑于 2021-09-04 13:36)

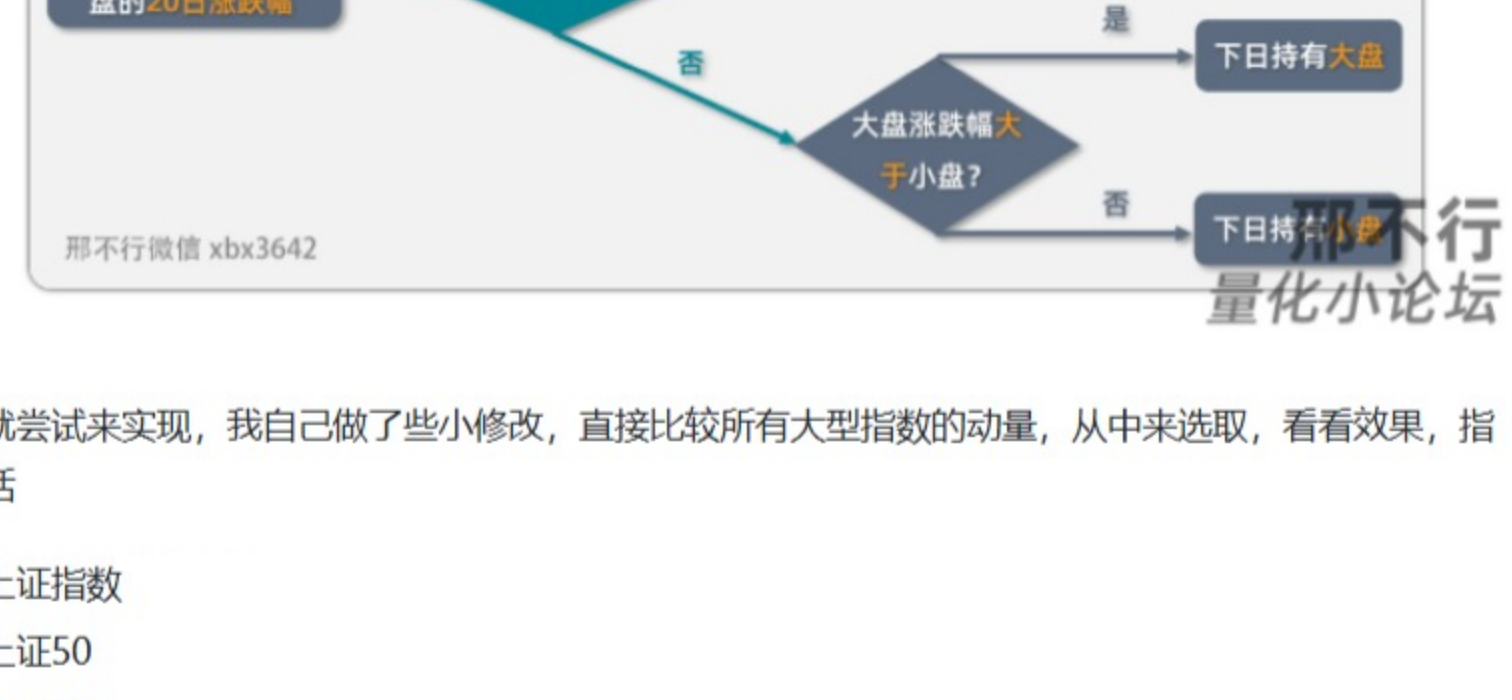
#心得分享📖

上周突然想起那大公众号挺久之前提过一个不错的大小盘轮动的思路，一直没能找到时间去实现，虽然那大说可以抄他去拿代码，但是我还是想自己来实现，一是巩固和提升自己的编程水平，二是能更好的理解策略原理（虽然思路比较简单），

有兴趣的朋友可以看看原文，链接：[抱团股会一直涨？无脑执行大小盘轮动策略，轻松赢赢指数5倍【附Python代码】](#)

策略思路

在两只指数之间寻找动量更强的一个，做为持仓



下面就尝试来实践，我自己做了些小修改，直接比较所有大型指数的动量，从中来选取，看看效果，指数包括

- 上证指数
- 上证50
- 中证500
- 中证1000
- 沪深300
- 创业板指
- 创业板50
- 科创板50

```
import pandas as pd
import os
import glob
import itertools
import numpy as np
import matplotlib.pyplot as plt

# 指数文件保存目录
index_folder_path = "指数文件夹"

# 需要轮动的指数列表
index_list = list(map(lambda x: os.path.basename(x).split("-")[0], glob.glob(os.path.join(index_folder_path, "*.csv"))))

def signal(x):
    """ 策略信号 """
    x = np.array(x)
    if len(x) > 0:
        return np.nan
    else:
        return index_list[np.argsort(x)[-np.isnan(np.sort(x))]-1]]

def calculate(params):
    pos_df = pd.DataFrame()
    n = int(params[0])

    for index in index_list:
        index_file_path = os.path.join(folder_path, f"{index}.csv")
        index_df = pd.read_csv(index_file_path, encoding="utf-8", parse_dates=["交易日期"])
        index_df["动量"] = index_df["收盘价"].pct_change(n)
        index_df = index_df[["交易日期", "股票代码", "涨跌幅", "动量"]]
        index_df.rename(columns={"股票代码": "index", "涨跌幅": f"{index}涨跌幅", "动量": f"{index}动量"}, inplace=True)
        if pos_df.empty:
            pos_df = index_df
        else:
            pos_df = pd.merge(pos_df, index_df, on="交易日期", how="outer")

    # 删除动量为空
    pos_df.dropna(subset=[f"{index}动量" for index in index_list, how="all", inplace=True)
    pos_df["下二日持仓标的"] = pos_df[[f"{index}动量" for index in index_list]].apply(lambda x: x.idxmax(), axis=1)
    pos_df["持仓涨跌幅"] = pos_df["下二日持仓标的"].shift(1)
    pos_df["持仓涨跌幅"] = np.nan
    for index in index_list:
        pos_df.loc[pos_df["持仓标的"] == index, "持仓涨跌幅"] = pos_df[f"{index}涨跌幅"]

    # 手续费
    pos_df.loc[pos_df["持仓标的"] != pos_df["持仓标的"].shift(1), "持仓涨跌幅"] = pos_df["持仓涨跌幅"]
    pos_df.loc[pos_df["持仓标的"] != pos_df["持仓标的"].shift(1), "持仓涨跌幅"] = pos_df["持仓涨跌幅"]

    # 交易时间
    pos_df = pos_df[pos_df["交易日期"] >= "2018-01-01"]

    # 计算指数
    for index in index_list:
        pos_df[index] = (pos_df[f"{index}涨跌幅"] + 1).cumprod()

    # 删除未使用的字段
    pos_df.drop(columns=[f"{index}涨跌幅" for index in index_list] + [f"{index}动量" for index in index_list], inplace=True)

    # 资金曲线
    pos_df["持仓涨跌幅"].fillna(value=0, inplace=True)
    pos_df["策略"] = (pos_df["持仓涨跌幅"] + 1).cumprod()

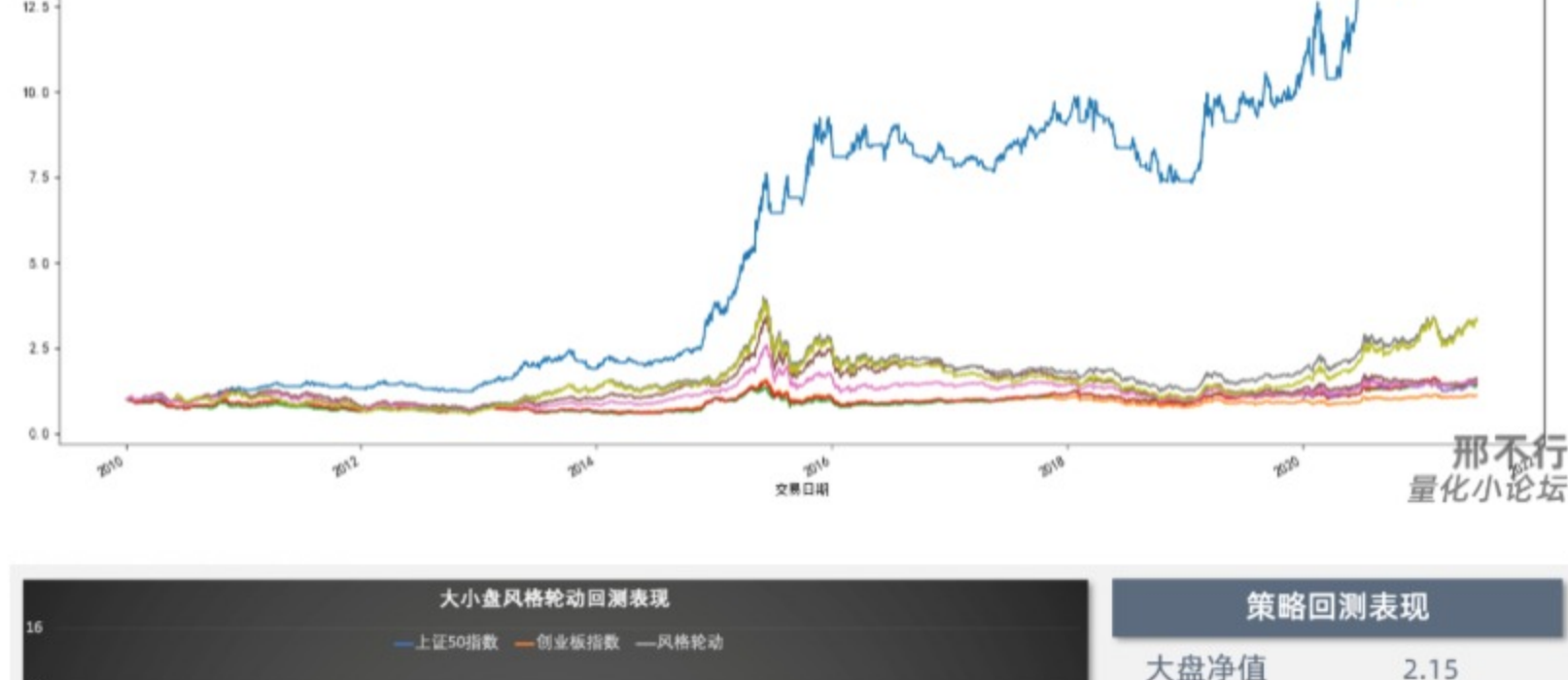
    return pos_df

def show(params):
    """ 查看策略 """
    pos_df = calculate(params)
    pos_df.set_index(keys=["交易日期"], inplace=True)
    print(pos_df)

    pos_df["策略"] + [index for index in index_list].plot()
    plt.legend(loc="best")
    plt.show()

if __name__ == "__main__":
    show((1,))
```

资金曲线



策略名称	资金净值	年化收益	最大回撤	回撤起点	回撤终点
大盘净值	2.15				
小盘净值	3.57				
策略净值	11.71→13.15				
年化收益	26.21%→27.61%				
最大回撤	29.13%→27.32%				
回撤起点	2018-01-24				
回撤终点	2018-01-24				

可以看到，几乎和那大的资金曲线走势相同，那策略代码应该是没什么大问题的了。

科创加油!

全部持仓

查看已清仓证券

股票/市值	持仓/可用	市价/成本价	盈亏/盈亏占比
科创 50		1.595	
		1.491	6.98%

以上全是留持仓

那不行
量化小论坛