

百天计划之38篇，关于“AI智能量化投研平台”搭建。

今天重点要完成可转债“双低”策略的数据补齐。

## 01 转债一些基础知识

我们先来回顾一下“双低策略”以及它需要什么数据。

双低值 = 转债价格 + 转股溢价率 \* 100。

排除：已公告强赎的和一年内到期的。

转债价格(衡量债性)

：这个就是转债每天的收盘价，这个最直接，我们已经入库了。

转股溢价率（衡量股性） =  $\frac{\text{转债价格}}{\text{转股价值}} - 1 = \frac{\text{转债价格}}{(100/\text{转股价} * \text{正股价})} - 1$ 。

看起来有点复杂，但理解一下就比较简单：

$100/\text{转股价}$  = 可以转成“几张股票”。

张数 \* 正股价，就是转成股票后“市值”是多少。

所以转股溢价率就是 用多少（X）钱（转债价格）买入市场上值多少价值（Y）的东西，比如100块买入120块的东西，溢价率 =  $100/120 - 1$  为负的。

溢价的意思就是“买亏”的多少，为负的最好，为零就是不亏不赚，所以溢价肯定越低越高，越低则股性越强。

综上：双低就是寻找当前市场上债性强，股性也强的前N支。

我们还需要补充的数据是转股价，是否公告强赎，以及到期日。

## 02 转股价

从tushare拉取转股价入mongo：

```
def get_cb_price_chg(code):
```

```
# ????  
df = pro.cb_price_chg(**{  
    "ts_code": code,  
    "limit": "",  
    "offset": ""  
}, fields=[  
    "ts_code",  
    "bond_short_name",  
    "publish_date",  
    "change_date",  
    "convert_price_initial",  
    "convertprice_bef",  
    "convertprice_aft"  
])  
return df  
  
def update_all_cb_price_chg():  
    items = list(get_db()['bond_basic'].find({}, {'ts_code':  
1, '_id': 0}))  
    if items and len(items) == 0:  
        logger.error("????????")  
        return  
    for i, item in enumerate(items):  
        code = item['ts_code']  
        logger.debug("{}-{}-{}".format(i, code, i / len(item  
s)))  
  
        df = get_cb_price_chg(code)  
        df['_id'] = df['ts_code'] + '_' + df['change_date']  
        print(df.tAIl())  
        write_df('bond_price_chg', df)  
  
if __name__ == '__main__':  
    update_all_cb_price_chg()
```

代码比较简单，这里不做增量更新了，直接全量同步即可。

这就是做转债的好处，因为总共400支，不像主动型基金，你一搞1万多支，同步数据是一个大麻烦。

从mongo入qlib备查。

我们需要把上面这个表变成 转股价时间序列。

我们从mongo读出来的数据如下：

我们需要把转股价变成一列，即如果为Nan，则使用initial初始值替换。

这里主要涉及pandas

.Dataframe处理数据的技巧了，做量化投资，pandas的熟练使用会让你事半功倍。

处理之后就是我们的转股价：

```
def get_price_chg(code):
    def check_nan(x):
        if pd.isnull(x['convertprice_aft']):
            return x['convert_price_initial']
        return x['convertprice_aft']

    df = get_db()['bond_price_chg'].find({'ts_code': code},
{'change_date': 1,

'convertprice_bef': 1,

'convertprice_aft': 1,

'convert_price_initial': 1,

'_id': 0}
)
df = pd.DataFrame(list(df))
print(df)
```

```
    # convert price aft convert price initia
1
    df['chg_price'] = df.apply(lambda x: check_nan(x), axis=
1)
    print(df)
return df
```

序列合并：

```
df_all = pd.concat([df, df_price_chg], axis=1)
df_all.sort_index(inplace=True, ascending=True)
df_all.fillna(method='ffill', inplace=True)
df_all.sort_index(ascending=True, inplace=True)
df_all.dropna(inplace=True)
```

很重要的一行代码“前向填充ffill”，这样就把这种不规则数据，填充成了时间序列，方便我们后续按天计算。

同理，把正股价查询并合并进来：

数据准备好了，写入csv可以转成qlib数据库。

### 03 从mongo到csv到qlib数据库

```
def build_all(path):
    # path.unlink()
    codes = get_db()['bond_daily'].distinct(key='ts_code')
    for i, code in enumerate(codes):
        logger.debug('?????{ }???Csv-{} , {}'.format(code, i,
i / len(codes)))
        df = get_data_from_mongo(code)
        df.rename(columns={'trade_date': 'date', 'vol': 'vol
```

```
ume'}, inplace=True)
    print(df)
    df.to_csv('{} / {}'.format(path, code), index=False)
)
    #break
    dump = DumpDataAll(csv_path=DATA_DIR_CSV_BOND, qlib_dir=
DATA_DIR_QLIB_BOND)
    dump.dump()
```

综合步骤，qlib关于可转债的数据库就准备好了。

明天可以就“双低”策略进行回测。

转债列表筛选及与正股数据整合：qlib+fastapi

可转债列表页与日频交易数据呈现：fastapi+antV G2

fastapi定时任务，增量构建可转债交易数据入mongo和qlib

飞狐，科技公司CTO，用AI技术做量化投资；以投资视角观历史，解时事；专注个人成长与财富自由。