deepshare.net
深度之眼

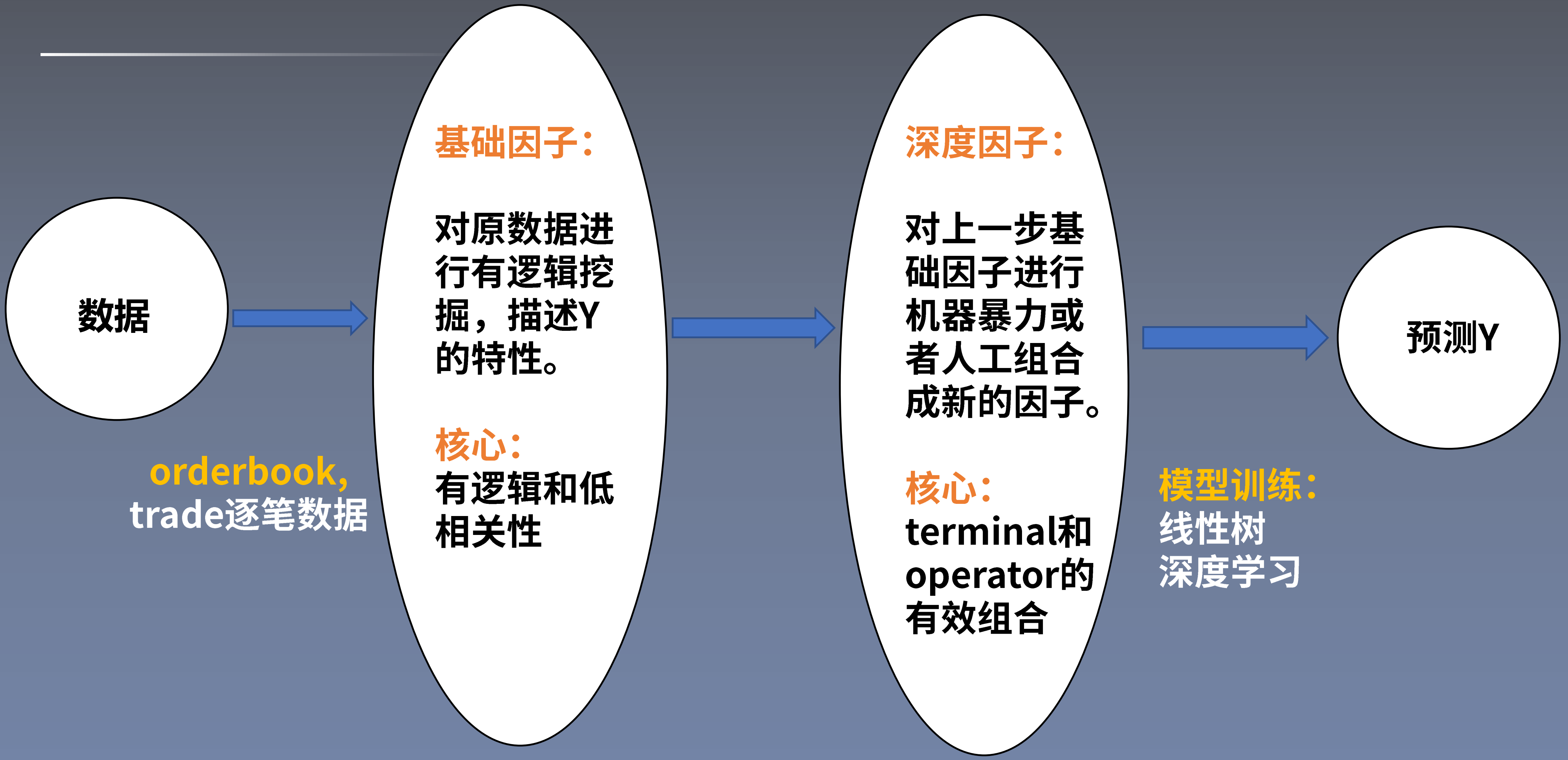
Kaggle-量化波动率预测 高频因子挖掘套路(一)

导师：Shr

deepshare.net
深度之眼

目 录

高频因子挖掘套路-概述


数据
orderbook,
trade逐笔数据
基础因子：
对原数据进行有逻辑挖掘，描述Y的特性。
核心：
有逻辑和低相关性
深度因子：
对上一步基础因子进行机器暴力或者人工组合成新的因子。
核心：
terminal和operator的有效组合
模型训练：
线性树
深度学习
预测Y

2、baseline feature讲解

baseline feature讲解

deepshare.net
深度之眼

book_wap{i}:对于每一档，将买卖挂单拼起来，按照挂单量，对买卖挂单价格进行加权

book_wap_mean：将不同档的挂单均价取均值

book_wap_diff: 不同档挂单均价的差

book_price_spread: 买一卖一挂单价的偏离度，除以bid1+ask1，为了不同时间不同股票指标的可比性。

bid_spread

ask_spread

total_volume

total_volume_imbalance

```python
def feature_row(book):
    # book_wap1 生成标签
    for i in [
            1,
            2,
    ]:
        # wap
        book[f'book_wap{i}'] = (book[f'bid_price{i}'] * book[f'ask_size{i}'] +
                                book[f'ask_price{i}'] *
                                book[f'bid_size{i}']) / (book[f'bid_size{i}'] +
                                                         book[f'ask_size{i}'])
    # mean wap
    book['book_wap_mean'] = (book['book_wap1'] + book['book_wap2']) / 2

    # wap diff
    book['book_wap_diff'] = book['book_wap1'] - book['book_wap2']

    # other orderbook features
    book['book_price_spread'] = (book['ask_price1'] - book['bid_price1']) / (
        book['ask_price1'] + book['bid_price1'])
    book['book_bid_spread'] = book['bid_price1'] - book['bid_price2']
    book['book_ask_spread'] = book['ask_price1'] - book['ask_price2']
    book['book_total_volume'] = book['ask_size1'] + book['ask_size2'] + book[
        'bid_size1'] + book['bid_size2']
    book['book_volume_imbalance'] = (book['ask_size1'] + book['ask_size2']) - (
        book['bid_size1'] + book['bid_size2'])
    return book
```

baseline feature讲解

对price(成交价格），size(成交量），order_count(成交笔数)做简单的统计性质

```
def feature_agg(book, trade):
    """
    """
    # 聚合生成特征
    book_feats = book.columns[book.columns.str.startswith('book_')].tolist()
    trade_feats = ['price', 'size', 'order_count', 'seconds_in_bucket']

    trade = trade.groupby(['time_id', 'stock_id'])[trade_feats].agg(
        ['sum', 'mean', 'std', 'max', 'min']).reset_index()

    book = book.groupby(['time_id', 'stock_id'])[book_feats].agg(
        [lambda x: realized_volatility(log_return(x))]).reset_index()

    # 修改特征名称
    book.columns = ["".join(col).strip() for col in book.columns.values]
    trade.columns = ["".join(col).strip() for col in trade.columns.values]
    df_ret = book.merge(trade, how='left', on=['time_id', 'stock_id'])
    return df_ret
```

deepshare.net
深度之眼

3、有逻辑的base feature

有逻辑的base feature

一、本质是对基础数据（orderbook，trade book)的重新聚合和再加工。

----统计意义的feature: mean,std,skew,kurtosis,autocorr...

----行为金融学、经济学意义的feature：MACD,KDJ,RSI,ATR,CCI....

二、要求：强逻辑性和低相关性，便于后期feature再加工，以及最终的因子组合和模型训练。

trade_df - DataFrame

Index	time_id	seconds_in_bucket	price	size	order_count
478	5	587	1.0013365	109	5
479	5	588	1.0012634	701	9
480	5	589	1.0013491	1	1
481	5	590	1.0011635	378	17
482	5	591	1.0010926	104	8
483	5	592	1.0011038	11	2
484	5	593	1.001175	451	14
485	5	594	1.0013336	456	11
486	5	597	1.001436	105	3
487	5	598	1.0014776	62	1
488	5	599	1.0015239	1410	37
489	11	2	1.000469	980	20
490	11	3	1.0003726	283	4
491	11	4	1.0004969	100	2
492	11	5	1.0002782	1319	36
493	11	6	1.0003312	2	1
494	11	12	1.0003312	5	2
495	11	16	1.000326	8	2
496	11	17	1.0003681	112	2
497	11	23	1.0004554	171	2

Format | Resize | Background color | Column min/max | Save and Close | Close

book_df - DataFrame

Index	time_id	seconds_in_bucket	bid_price1	ask_price1	bid_price2	ask_price2	b
586	5	586	1.0012206	1.0012634	1.0011778	1.0013063	106
587	5	587	1.0011778	1.0012634	1.0011349	1.0013063	109
588	5	588	1.0012634	1.0013491	1.0012206	1.0013919	43
589	5	589	1.0013491	1.0014347	1.0013063	1.0014776	11
590	5	590	1.0010921	1.0011349	1.0010493	1.0011778	17
591	5	591	1.0011349	1.0011778	1.0010921	1.0012206	18
592	5	592	1.0011349	1.0011778	1.0010921	1.0012206	103
593	5	593	1.0012206	1.0013063	1.0011778	1.0013491	11
594	5	594	1.0013491	1.0014347	1.0013063	1.0014776	201
595	5	595	1.0013491	1.0014347	1.0013063	1.0014776	101
596	5	596	1.0013491	1.0014347	1.0013063	1.0014776	102
597	5	597	1.0013919	1.0014776	1.0013491	1.0015204	251
598	5	598	1.0014347	1.0014776	1.0013919	1.0015204	1
599	5	599	1.001606	1.0016917	1.0015632	1.0017345	13
600	11	0	1.0005382	1.000621	1.0004969	1.0006624	567
601	11	1	1.0005382	1.000621	1.0004969	1.0006624	467
602	11	2	1.0003312	1.0003726	1.0002898	1.000414	200
603	11	3	1.0003726	1.0004554	1.0003312	1.0004969	110
604	11	4	1.0004969	1.0005796	1.0004554	1.000621	120
605	11	5	1.000207	1.0003312	1.0001656	1.0003726	29
606	11	6	1.0002898	1.000414	1.0002484	1.0004554	31
607	11	7	1.0002898	1.000414	1.0002484	1.0004554	112
608	11	8	1.0002898	1.000414	1.0002484	1.0004554	212
609	11	9	1.0003312	1.0004554	1.0002898	1.0004969	7
610	11	10	1.0003312	1.000414	1.0002898	1.0004554	5

Format | Resize | Background color | Column min/max | Save and Close | Close

eg： 对orderbook 和tradebook按时间聚类做feature

有逻辑的base feature

deepshare.net 深度之眼

1、统计意义的feature:

----（对价格，量，rolling波动率）mean,std,skew,kurtosis,

---- 单时间序列的autocorr，多变量之间的corr，cov

---- 多变量之间OLS，取beta，residual，rsquare

有逻辑的base feature

deepshare.net 深度之眼

2、行为金融学、经济学意义的feature

前期准备：将orderbook，tradebook做成分成N等分，做成高开低收的candle

原始trade book

f_data - DataFrame

Index	time_id	seconds_in_bucket	price	size	order_count	
5	5	583	1.0015204	103	3	1
6	5	585	1.0013713	176	5	1
7	5	586	1.0012482	17	4	1
8	5	587	1.0013365	109	5	1
9	5	588	1.0012634	701	9	7
0	5	589	1.0013491	1	1	1
1	5	590	1.0011635	378	17	3
2	5	591	1.0010926	104	8	1
3	5	592	1.0011038	11	2	1
4	5	593	1.001175	451	14	4
5	5	594	1.0013336	456	11	4
6	5	597	1.001436	105	3	1
7	5	598	1.0014776	62	1	6
8	5	599	1.0015239	1410	37	1
9	11	2	1.000469	980	20	
0	11	3	1.0003726	283	4	
1	11	4	1.0004969	100	2	
2	11	5	1.0002782	1319	36	
3	11	6	1.0003312	2	1	
4	11	12	1.0003312	5	2	
5	11	16	1.000326	8	2	
6	11	17	1.0003681	112	2	
7	11	23	1.0004554	171	2	
8	11	29	1.0006624	22	3	
9	11	38	1.0007032	304	5	
0	11	39	1.0007038	100	1	
1	11	42	1.0006379	1230	17	
2	11	44	1.0005796	100	1	
3	11	48	1.0005796	206	3	
4	11	49	1.000621	150	4	
5	11	50	1.000621	1	1	

制作candle

df_newxx - DataFrame

Index	time_id	groupi	high	low	vwap
46	5	46	1.0025833	1.0021628	1.0023029
47	5	47	1.0025766	1.0022913	1.002451
48	5	48	1.0025244	1.0022964	1.0024309
49	5	49	1.0025285	1.0023769	1.0024573
50	5	50	1.0024265	1.0021122	1.00232
51	5	51	1.002684	1.0021594	1.0025313
52	5	52	1.0024792	1.0022484	1.0023437
53	5	53	1.0023341	1.0019438	1.0020955
54	5	54	1.0023341	1.0020393	1.0021572
55	5	55	1.0022484	1.0020343	1.0021057
56	5	56	1.0022645	1.0015153	1.0019772
57	5	57	1.0016953	1.001507	1.0016237
58	5	58	1.0015204	1.0012482	1.001372
59	5	59	1.0015239	1.0010926	1.0012882
60	11	0	1.0004969	1.0002782	1.0003896
61	11	1	1.0003681	1.000326	1.0003418
62	11	2	1.0006624	1.0004554	1.0005589
63	11	3	1.0007038	1.0007032	1.0007036
64	11	4	1.0006379	1.0005796	1.0006045
65	11	5	1.000621	1.0003726	1.000477
66	11	6	1.0003146	1.0001947	1.0002619
67	11	7	1.0002484	1.000207	1.0002346
68	11	8	1.0001656	1.0000414	1.0001236
69	11	9	1.000207	1.0001656	1.0001862
70	11	10	1.0001656	1.0001342	1.00015
71	11	11	1.000207	1.0001656	1.0001824
72	11	12	1.0001652	1.0000414	1.0000826
73	11	13	1.000207	1.0000414	1.0001067
74	11	14	1.000414	1.0002668	1.0003566
75	11	15	1.000414	1.0002898	1.0003519
76	11	16	1.0004554	1.000414	1.0004337

每个timeid的feature

df_candle - DataFrame

Index	time_id	pricehigh	pricelow	pricemean	pricec
0	5	1.002684	0.9993021	1.0016363	0.99930
1	11	1.0008256	1.0000414	1.0004425	1.00046
2	16	0.9999488	0.9990271	0.99943537	0.99969
3	31	0.99936575	0.9977599	0.9985353	0.99917
4	62	0.9997715	0.99843085	0.9991682	0.99967
5	72	1.0001986	0.9960454	0.9980557	0.99964
6	97	0.9991845	0.9962798	0.9979797	0.99881
7	103	1.00161	0.9977888	0.999546	0.99979
8	109	1.001397	0.9979097	0.9993093	0.99832
9	123	1.005232	0.99918526	1.0026085	1.00002
10	128	1.000243	0.99902296	0.99970514	0.99958
11	146	1.0076256	0.99715334	1.0044739	0.99745
12	147	1.0000803	0.9978843	0.9991385	0.99998
13	152	1.0020149	0.99966496	1.0010903	1.00051
14	157	1.0027605	0.9959067	0.9990804	0.99888
15	159	1.0012605	0.99863654	0.99979824	1.00006
16	169	1.0000621	0.99819994	0.9991211	0.99968
17	207	1.0120848	1.0025293	1.0072232	1.00501
18	211	0.9987515	0.99759156	0.99818367	0.99798
19	213	1.004052	1.0010984	1.0027717	1.00176
20	218	1.0010206	0.9997052	1.0004039	1.00059
21	227	0.99845	0.9961465	0.9972642	0.99790
22	229	1.0009702	1.000258	1.0006566	1.00034
23	232	0.99923325	0.9898533	0.9939624	0.99833
24	250	1.00047	0.9994046	0.9999243	0.99978
25	254	1.0090561	0.99993443	1.0050482	1.00113
26	256	1.0006862	0.9983601	0.9992897	0.99878
27	266	1.0014136	0.9988692	1.0003943	1.00079
28	273	1.0003512	0.99936795	0.99989307	0.99981
29	289	0.99970937	0.99709356	0.99820346	0.99925

有逻辑的base feature

2、行为金融学、经济学意义的feature

1）MACD，KDJ等简单的指标，TA-LIB包可以实现

2）描述波动的指标：

价格的波动和成交量扩大收缩有密切关系，所以用价量构建描述波动的指标

简单的如ts_corr(price,volume)

-----AD指标

将成交量用价格加权，计算成交量的动量

AD = SUM((close-low)-(high-close))/(high-low)*volume)

有逻辑的base feature

deepshare.net 深度之眼

2、行为金融学、经济学意义的feature

------能量潮（OBV）

将成交量数量化，做成趋势线

价格上涨时成交量总是放大的，下跌时可能放大也可能收缩

obv = sum(if(close>preclose,volume,

if (close<preclose,-volume,0),0)

-----klinger成交量摆动指标

均价x = (close+high+low)/3

sum(if x>x_pre，volume, if (x<x_pre, -volume),0)

有逻辑的base feature

2、行为金融学、经济学意义的feature

-----平均真实波动幅度均值（ATR）

指标越高，趋势改变的可能性越高

atr1 = (max(max(high-low,abs(preclose-high)),abs(preclose-low))

atr = ma(atr1, n)

n:时间参数

------乖离率（BIAS）

bias = (close/ma(close,n)) - 1: 描述距离均线的偏离程度

有逻辑的base feature

deepshare.net
深度之眼

2、行为金融学、经济学意义的feature

------相对波动率指数（RVI）

UP = if (close>pre_close, std(close),0)

DOWN = if (close<=pre_close, std(close),0)

AUP = SMA(UP,N,1)

ADOWN = SMA(DOWN,N,1)

RVI = AUP/(AUP+ADOWN)

有逻辑的base feature

2、行为金融学、经济学意义的feature

3）流动性指标

波动率天然和市场/个股的流动性相关性很大。

流动性更高，单位成交量带来的return的变化越大。

描述流动性：

BARRA，盘口买卖价差、深度、宽度，订单不平衡

有逻辑的base feature

2、行为金融学、经济学意义的feature

BARRA中，对流动性的定义：

短，中，长期的换手率

Liquidity

Definition: *0.35 · STOM + 0.35 · STOQ + 0.30 · STOA*

STOM — *Share turnover, one month*

Computed as the log of the sum of daily turnover during the previous 21 trading days,

$$STOM = \ln\left(\sum_{t=1}^{21}\frac{V_t}{S_t}\right), \qquad \text{(A9)}$$

where V_t is the trading volume on day t, and S_t is the number of shares outstanding.

STOQ — *Average share turnover, trailing 3 months*

Let $STOM_\tau$ be the share turnover for month τ, with each month consisting of 21 trading days. The quarterly share turnover is defined by

$$STOQ = \ln\left[\frac{1}{T}\sum_{\tau=1}^{T}\exp(STOM_\tau)\right], \qquad \text{(A10)}$$

where $T = 3$ months.

STOA — *Average share turnover, trailing 12 months*

Let $STOM_\tau$ be the share turnover for month τ, with each month consisting of 21 trading days. The annual share turnover is defined by

$$STOA = \ln\left[\frac{1}{T}\sum_{\tau=1}^{T}\exp(STOM_\tau)\right], \qquad \text{(A11)}$$

where $T = 12$ months.

The Liquidity factor is orthogonalized with respect to Size to reduce collinearity.

有逻辑的base feature

deepshare.net 深度之眼

2、行为金融学、经济学意义的feature

-------交易量订单量不平衡性

bidi,aski:每档买卖价格，bidv_i, askv_i每档买卖量

**深度：depth imbalance

$$f = sum((bidv\_i - askv\_i)/(bidv\_i+askv\_i))$$

**宽度: height imbalance

$$f = sum(((bid\_i\_t1 - bid\_i\_t0)-(ask\_i\_t1 - ask\_i\_t0))/ ((bid\_i\_t1 - bid\_i\_t0)+(ask\_i\_t1 - ask\_i\_t0)))$$

有逻辑的base feature

deepshare.net
深度之眼

2、行为金融学、经济学意义的feature

**买卖压力:

根据每一档价格距离买卖中间价（bid1,ask1的均值）的距离，给买卖委

托量一个权重：midprice = (aski + bidi)/2

press_ask = sum(askv_i*(midprice - aski)/sum(midprice - aski))

press_bid = sum(bidv_i*(midprice - bidi)/sum(midprice - bidi))

f = log(press_ask) - log(press_bid)

有逻辑的base feature

deepshare.net
深度之眼

2、行为金融学、经济学意义的feature

-----流动性模型

Amihud非流动性模型：

f = sum(abs(收益率)/成交额)/n

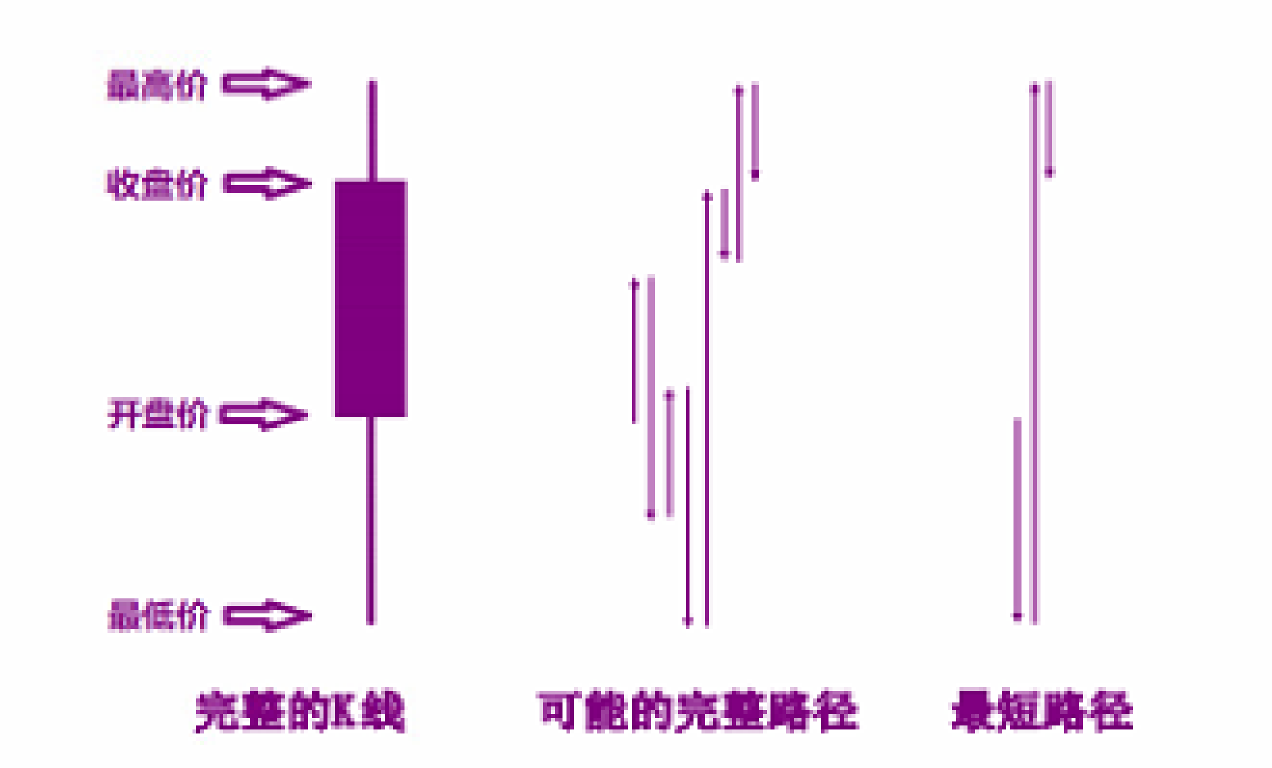
描述每份 成交额，对于价格走过的路程

高频化：

最短路径：

x=2（high - low） - abs(open - close)

f = sum(x/成交额)



deepshare.net
深度之眼

4、深层次的feature

深层次的feature

---------基于base feature 进行深层次挖掘

---------本质上是terminal和operator的有机结合

---------算法：网格搜索，遗传规划，神经网络

deepshare.net

深度之眼

联系我们:

电话: 18001992849

邮箱: service@deepshare.net

Q Q: 2677693114

公众号

客服微信