

# 智能投顾：理论与实践

October 27, 2016

# Contents

|          |                       |          |
|----------|-----------------------|----------|
| <b>1</b> | <b>介绍</b>             | <b>2</b> |
| <b>2</b> | <b>实践出真知</b>          | <b>5</b> |
| 2.1      | ETF 费用一览 . . . . .    | 5        |
| 2.2      | Python Code . . . . . | 6        |

# Chapter 1

## 介绍

所谓的智能投顾：

在国外的情况：

优势：手续费便宜，避税

在国内发展的情况：

在国内发展相关法律法规禁止。

2010 年，第一家智能投顾公司 Betterment 成立于纽约。截至 2016 年 10 月末，该公司管理的资产规模约 60 亿美元。

截至 2016 年 10 月，目前最大的智能投顾公司是著名的 Vanguard 基金公司，管理规模约 410 亿美元。

目前全部的智能投顾公司管理的资产规模约 3,000 亿（参见图 1.1）。

定义

发展

展望

visual 图片

选取国内外的资产

画出 efficient frontier

给出配置组合

## Robo-Advisor Launch Timeline

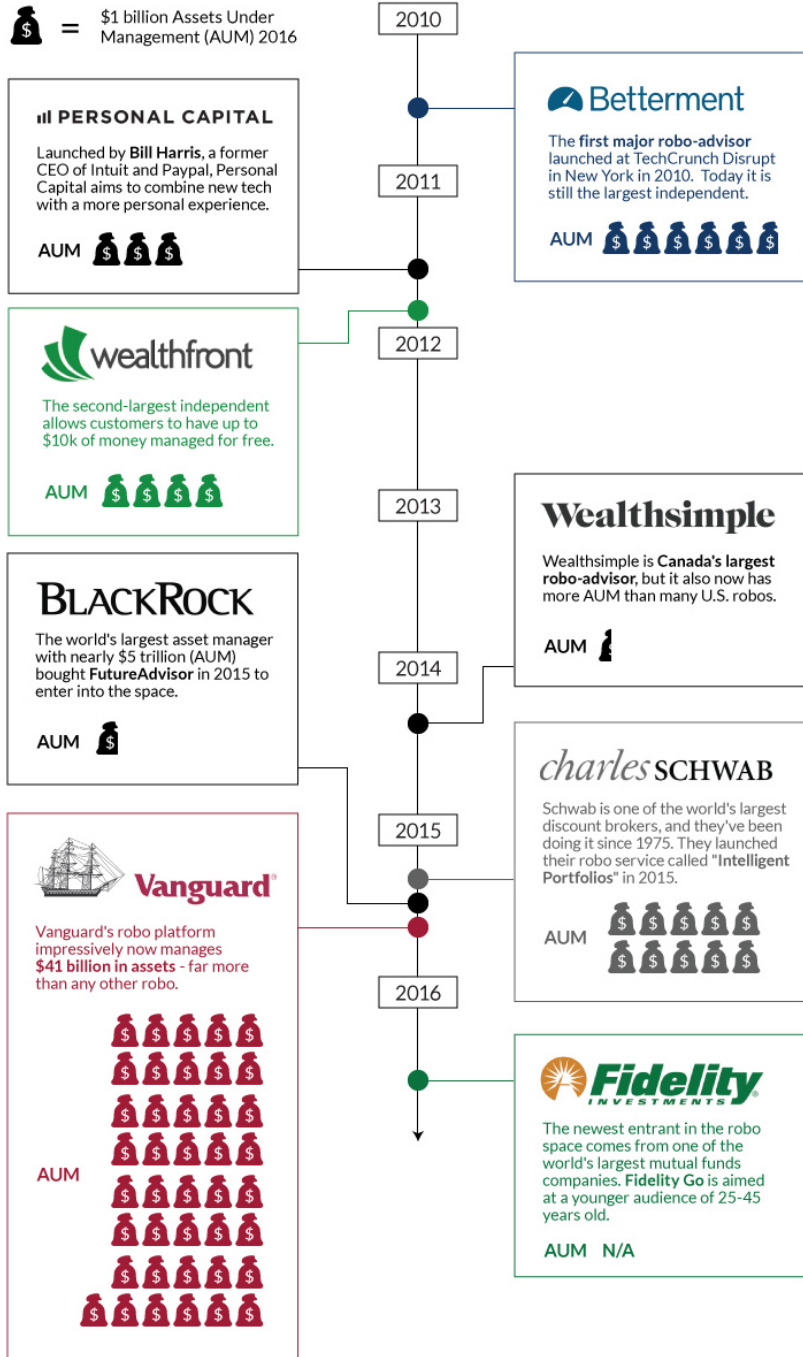


Figure 1.1: 智能投顾发展一览

(图片来源: <http://www.visualcapitalist.com/robo-advisor-arms-race/>)



## Chapter 2

# 实践出真知

本节构建资产组合。算法的理论基础是资本资产定价模型 CAPM (TODO 链接)，同时参考 wealthfront (TODO 链接) 的介绍。算法主要通过 Python (TODO 链接) 实现，代码部分参考了通联数据 (TODO 链接)，海外资产的数据由 Yahoo Finance (TODO 链接) 提供，国内资产的数据由 TuShare (TODO 链接) 提供。

该算法的不足：(1) 估算不同资产间的相关性；(2) 预测资产的年化收益率。事实上，上述两个问题是理论界与业界共同关心的主要问题。(TODO 链接) 本文主要用历史收益率来估算。

主要的资产组合都是 ETF。(TODO 解释为什么要用 etf) 手续费较低，流通性好，可以日内交易。顺便提一下，华泰证券<sup>1</sup> ETF 交易费最低 0.1 元！如果用来少量的搭建组合，还是比较划算。其他券商还是最低 5 元。

### 2.1 ETF 费用一览

(TODO 各 etf 的管理费、规模、成立日期，管理人)

国内的 ETF 普遍比较坑，QDII 组合费用（管理费 + 托管费）在 1.00% 左右，海外的约在 0.10% 左右。

---

<sup>1</sup> 此处不是广告！该券商未以任何形式提供赞助 -\_-

## 2.2 Python Code

代码:

Listing 2.1: Download the data

```
1 # -*- coding: utf-8 -*-
2 '''
3 Modified on Fri, Sep 23, 2016
4 Author: Haifeng XU
5 Email: 78112407@qq.com
6
7 df.shape
8 df.describe()
9 df.info()
10 '''
11
12 import numpy as np
13 import tushare as ts
14 import pandas as pd
15 import pandas_datareader.data as web
16 import datetime
17
18 _start_date = '2007-01-01'
19 _end_date = '2016-10-01'
20
21 ## ~~~~~ ##
22 def _initial_index_cn() :
23     index_list = []
24     index_name = []
25
26     index_list.append( '150151' )    ## HS300A
27     index_name.append( 'HS300A' )
28
29     index_list.append( '150152' )    ## HS300B
30     index_name.append( 'HS300B' )
31
32     index_list.append( '159920' )    ## 恒生ETF
33     index_name.append( '恒生ETF' )
```

```

34
35 index_list.append( '160125' )    ## 南方香港
36 index_name.append( '南方香港' )
37
38 index_list.append( '160416' )    ## 石油黄金
39 index_name.append( '石油黄金' )
40
41 index_list.append( '160717' )    ## 恒生H股
42 index_name.append( '恒生H股' )
43
44 index_list.append( '161116' )    ## 易基黄金
45 index_name.append( '易基黄金' )
46
47 index_list.append( '161210' )    ## 国投新兴
48 index_name.append( '国投新兴' )
49
50 index_list.append( '161714' )    ## 招商金砖
51 index_name.append( '招商金砖' )
52
53 index_list.append( '161815' )    ## 银华通胀
54 index_name.append( '银华通胀' )
55
56 index_list.append( '162411' )    ## 华宝油气
57 index_name.append( '华宝油气' )
58 ## 标普美国行业指数系列之油气开采及生产行业指数
59 ## (S&P Select Industry Oil & Gas Exploration & Production)
60 ## SPSIOP is the index being traced
61 ## XOP is an ETF tracing SPSIOP
62 ## 指数成分股的入选必须满足以下条件:
63 ## 1、成份股是标普美国全市场指数的成员;
64 ## 2、成份股属于GICS定义的油气二级行业分类;
65 ## 3、成份股市值大于5亿美金,或市值大于4亿美金;
66 ## 4、且交易量年换手率大于150%。
67
68 index_list.append( '164701' )    ## 添富贵金
69 index_name.append( '添富贵金' )
70
71 index_list.append( '164815' )    ## 工银资源

```



```

72     index_name.append( '工银资源' )
73
74     index_list.append( '165510' )    ## 信诚四国
75     index_name.append( '信诚四国' )
76
77     index_list.append( '165513' )    ## 信诚商品
78     index_name.append( '信诚商品' )
79
80     index_list.append( '510300' )    ## HS300 ETF
81     index_name.append( '沪深300 ETF' )
82
83     index_list.append( '510500' )    ## 500 ETF
84     index_name.append( '中证500 ETF' )
85
86     index_list.append( '510900' )    ## H股ETF
87     index_name.append( 'H股ETF' )
88
89     index_list.append( '511860' )    ## MoneyFund
90     index_name.append( 'MoneyFund' )
91
92     index_list.append( '513030' )    ## 德国30
93     index_name.append( '德国30' )
94
95     index_list.append( '513100' )    ## 纳指ETF
96     index_name.append( '纳指ETF' )
97
98     index_list.append( '513500' )    ## 标普500
99     index_name.append( '标普500' )
100
101     return index_list, index_name
102 ## ..... ##
103
104
105 ## ~~~~~ ##
106 def _initial_index_us() :
107     '''
108     Initialization of US assets
109     '''

```

```

110
111     index_list = []
112     index_name = []
113
114     index_list.append( 'AAPL' )
115     index_name.append( 'Apple' )
116
117     index_list.append( 'GOOG' )
118     index_name.append( 'Google' )
119
120     index_list.append( 'GLD' )
121     index_name.append( 'Gold' )
122
123     index_list.append( 'SPY' )
124     index_name.append( 'S&P 500' )
125
126     index_list.append( 'USO' )
127     index_name.append( 'USO' )
128
129     index_list.append( 'XOP' )
130     index_name.append( 'XOP' )
131
132     return index_list, index_name
133 ## ..... ##
134
135
136 ## ***** ##
137 def _initial_index( country_code ):
138     '''
139     -----
140     This function just insert the asset code and name
141     -----
142     '''
143
144     index_list = []
145     index_list_name = []
146
147     if country_code == 'cn' :

```

```

148         index_list, index_list_name = _initial_index_cn()
149
150     if country_code == 'us' :
151         index_list, index_list_name = _initial_index_us()
152
153     return index_list, index_list_name
154 ## ..... ##
155
156
157 ## ~~~~~ ##
158 ## 'my_ptf_cn' records the close price of the assets
159 ## in 'index_list_cn['
160 ## ***** ##
161 def _download_data( country_code ) :
162     '''
163     -----
164
165         Parameters
166     -----
167
168     index: refers to the code of assets.
169     index_name: refers to the name of assets.
170
171     -----
172
173         Return
174     -----
175
176     DataFrame: which contains the close price of the assets
177                 in "index"
178     '''
179
180     ## create an empty pd, which would be returned
181     df = pd.DataFrame()
182
183     ## assets to be added
184     index, index_name = _initial_index( country_code )
185
186     '''
187
188     The data from CN is supplied by TuShare.
189     Please refer to
190         http://tushare.org/index.html for more details.
191
192     '''

```

```

186 注意：CN 与 US 的价格日期是相反的，我把 CN 的顺序调整了，
187 与 US 保持一致。即 tail() 是最新的数据。
188
189 ## append the close price
190 if country_code == 'cn' :
191     for i in xrange( len( index ) ) :
192         df[ index[i] ] = ts.get_hist_data( index[ i ] ,
193                                             start = _start_date
194                                             ,
195                                             end = _end_date
196                                             )[ 'close' ]
197
198 The data from US is supplied by Yahoo-finance. Please refer to
199 http://pandas-datareader.readthedocs.io/en/latest/remote\_data.html#yahoo-finance
200 for more details.
201
202 Notice that 'Adj Close' price is what we want.
203
204 ## append the close price
205 if country_code == 'us' :
206     for i in xrange( len( index ) ) :
207         df[ index[i] ] = web.DataReader( index[ i ] ,
208                                         'yahoo',
209                                         start = _start_date ,
210                                         end = _end_date )[
211         'Adj Close' ]
212
213 ## replace inf and NA with zero
214 df[ df == np.inf ] = 0
215 df.fillna( 0, inplace = True )
216
217 ## rename the columns
218 df.rename( columns = dict( zip( index, index_name ) ),
219           inplace = True )
220
221 ## make sure the tail() is the latest data

```

```

220     df = df.sort_index( ascending = True )
221
222     ## make sure the index is of type datetime
223     df.index = pd.to_datetime( df.index )
224
225     ## test
226     print df.shape
227
228     return df
229 ## ..... ##
230
231
232 ## ~~~~~ ##
233 ## calculate the correlation of the assets
234 ## first of all, we calculate the changed ratio of price
235 ## the input needs to be a DataFrame containing the close price
236 ## ***** ##
237 def _price_change( tmp ) :
238     '''
239     -----
240
241     Parameters
242
243     -----
244
245     tmp: a dataframe containing the price.
246         Notice that the tail is the latest date
247
248     -----
249
250     Return
251
252     -----
253
254     DataFrame: a dataframe containing the change of price
255     '''
256
257     df = tmp.copy()
258     # df[1:] = 1.0 * df[1:].values / df[:-1].values - 1
259     df = df / df.shift(1) * 1.0 - 1.0
260
261     ## the first row should be zero
262     df[:1] = 0

```

```

258     ## replace inf and NA with zero
259     df[ df == np.inf ] = 0
260     df.fillna( 0, inplace = True )
261
262     return df
263     ## ..... ##
264
265
266     ## ~~~~~ ##
267     ## We construst my portfolio according to 'index_list_cn'.
268     ## Each entry records the close price of the assets.
269     ## ***** ##
270     def _get_my_ptf():
271         '''
272         return my portfolio
273         '''
274
275         df      = pd.DataFrame()
276         df_cn   = pd.DataFrame()
277         df_us   = pd.DataFrame()
278
279         df_cn = _download_data( country_code = 'cn' )
280         df_us = _download_data( country_code = 'us' )
281
282         '''
283         Caution: join = 'outer' or 'inner'
284         http://pandas.pydata.org/pandas-docs/stable/merging.html
285         '''
286         df = pd.concat( [df_cn, df_us],
287                         axis = 1,
288                         join = 'inner' )
289
290         ## replace inf and NA with zero
291         df[ df == np.inf ] = 0
292         df.fillna( 0, inplace = True )
293
294         return df
295     ## ..... ##

```

```

296
297 my_ptf = _get_my_ptf()
298 my_ptf.to_csv( 'etf_close_price.csv' )
299
300 my_ptf_rtn = _price_change( my_ptf )
301 my_ptf_rtn.to_csv( 'etf_rtn.csv' )
302
303 ## well, this is a bonus for users...
304 print('well, good job!')
305
306 '''
307 import matplotlib.pyplot as plt
308 fig = plt.figure()
309 ax = fig.add_subplot(1,1,1)
310 ax.plot( randn( 1000 ).cumsum() )
311 ax.set_xticks( [0, 50, 100] )
312 ax.set_xticklabels( )
313 ax.set_title( )
314 ax.set_xlabel( )
315 ax.legend(loc='best')
316 '''
317
318 ## ~~~~~ ##
319 ## END of the code
320 ## ***** ##
321 ## ..... ##

```