

Proyecto Algoritmos

Sebastian Guerrero Salinas

11 Febrero 2020

1. Introducción

A continuación se ofrece un breve análisis de los resultados obtenidos con el algoritmo *Cross-Sectional Equity* de Quantopian, dichos resultados obtenidos del algoritmo se seleccionaron y analizaron con tres variables más de las que el algoritmo tenía.

Este algoritmo demuestra el concepto de equidad a corto y largo plazo. Utiliza una combinación de factores para construir una clasificación de valores en un universo líquido negociable. Luego gasta mucho tiempo en los valores mejor clasificados y poco en los valores más bajos. Este algoritmo fue desarrollado como parte de la serie de lecturas de Quantopian.

2. Descripción del algoritmo

Al comienzo del código, se define el tamaño máximo de posición que se puede mantener para cualquier stock determinado. Si bien son parte importante del código, estas líneas iniciales se pueden cambiar a voluntad en caso de encontrar unos valores mejores para el desempeño del programa. Cabe aclarar que el optimizador necesita algo de margen para funcionar, es decir, si el máximo es demasiado pequeño, el optimizador puede estar demasiado limitado.

Seguido de esto, se pone una función central llamada automáticamente una vez al comienzo de un backtest. Se usa esta función para inicializar el estado u otra contabilidad. El parámetro de esta función es un objeto que se puede usar para almacenar el estado que desea mantener en su algoritmo. Dicho parámetro se pasa automáticamente a *initialize*, *before_trading_start*, *handle_data* y cualquier función que se ejecute a través de *schedule_function*. Este parámetro proporciona el atributo de cartera, que se puede utilizar para recuperar información sobre las posiciones actuales. En la siguiente parte se adjunte el *pipeline* para los factores del modelo de riesgo que queremos neutralizar en el paso de optimización. La cadena *risk_factors* se usa para recuperar la salida del *pipeline* en *before_trading_start*, a continuación se programa la función de *rebalance* y se registran las variables de cartera al final del día.

La siguiente adición es la definición de una función que crea y devuelve una tubería o *pipeline*. La tubería que devuelve la función representa el cálculo que nos gustaría realizar en los activos que pasan por la pantalla de canalización. Los factores que creamos aquí se basan en datos fundamentales y un promedio móvil de datos de opinión. Se usa el conjunto de *winorized* para nuestros valores de factores para disminuir el impacto de los valores atípicos.

En la siguiente parte, la parte que variará del algoritmo es la que combina nuestros factores winorizados, puntuando *z* para igualar su influencia. Se crean filtros que representen las cestas superiores e inferiores de acciones mediante nuestro sistema combinado de clasificación. Los usaremos como nuestro universo comercializable cada día. La producción final de nuestra cartera solo debe incluir las 300 existencias superiores / inferiores según nuestros criterios, seguido se crea la tubería.

Se define la Función central opcional llamada automáticamente antes de la apertura de cada día de mercado con dos parámetros, el primero es nuevamente un contexto y el segundo es un objeto que proporciona métodos para obtener datos de precio y volumen, verifica si existe una seguridad y verifica la última vez que se negoció una seguridad. Se llama a *algo.pipeline_output* para obtener la nota: este es un marco de datos donde el índice es el SID de todos los valores que pasan por mi pantalla y las columnas son los factores agregados al objeto de canalización anterior. Se crea un dataframe que contendrá todas nuestras cargas de riesgo.

Posteriormente se incluye una función programada para ejecutarse todos los días al cierre del mercado para registrar información de estrategia para luego trazar el número de posiciones a lo largo del tiempo. Se realiza un llamado al comienzo de cada mes para reequilibrar las listas largas y cortas. Se define una función programada para ejecutarse una vez todos los lunes a las 10 a.m. ET para reequilibrar las listas largas y cortas. Seguido de ello definimos nuestro objetivo para la API de Optimize. Hemos seleccionado *MaximizeAlpha* porque creemos que nuestra clasificación de factores combinados es proporcional a los rendimientos esperados. Esta rutina optimizará el retorno esperado de nuestro algoritmo, yendo por mucho tiempo con el mayor retorno esperado y corto por el más bajo. Se define la lista de restricciones entre las cuales está la restricción *RiskModelExposure* para hacer uso de las restricciones predeterminadas del modelo de riesgo. Se pone una nueva restricción, con esta restricción, exigimos que ninguna posición pueda compensar más de *MAX_SHORT_POSITION_SIZE* en el lado corto y no más de *MAX_LONG_POSITION_SIZE* en el lado largo. Esto garantiza que no concentremos demasiado nuestra cartera en un valor o en un pequeño subconjunto de valores.

Finalmente reúnen todas las piezas que definimos anteriormente pasándolas a la función *algo.order_optimal_portfolio*. Esto maneja toda nuestra lógica de pedido, asignando pesos apropiados a los valores en nuestro universo para maximizar nuestro alfa con respecto a las restricciones dadas.

3. Modificación del algoritmo

A continuación, como se había introducido, se va a modificar el algoritmo añadiendo más variables, explicando los parámetros que se tuvieron en cuenta para usar dichas nuevas variables y finalmente analizando los resultados obtenidos:

Primero se dará una corta descripción de las variables nuevas:

3.1. Total Revenue

Representa las ganancias totales de la compañía dadas como una suma total de las ganancias descritas en el reporte de las compañías sobre ingresos. Este valor se calcula así $\text{Total Revenue} = \text{Net Interest Income} + \text{Non-Interest Income}$. Se implementa en esta modificación del algoritmo ya que en un escenario donde una empresa ha mostrado ganancias altas, y que es consistente, se vuelve bastante llamativa para invertir a largo plazo.

3.2. Growth Score

Representa el crecimiento específico de la acción a comprar. Un valor alto en general del crecimiento de este puntaje de crecimiento indica que se espera el valor de la acción suba rápidamente. Teniendo en cuenta esto es bueno tenerla dentro de las variables nuevas ya que ofrece un conocimiento de cuáles acciones tienen un futuro prometedor en cuanto a ganancias y porcentajes positivos.

3.3. Sustainable Growth Rate

Esta variable ayudará a tener conocimiento de si la acción tiene retornos al inversor atractivos para el comportamiento que buscamos. Traducido y más claramente explicado, esto quiere decir que si compramos una acción con una buena razón de crecimiento sostenible, lo más probable es que la acción que se está comprando y analizando nos de retornos más que sobresalientes a lo largo de la inversión o estudio financiero.

3.4. Working Capital per Share

Se utilizó teniendo en cuenta los bienes y deudas del stock de acciones que se vaya a adquirir dividido entre el número de acciones que se encuentren en el mercado. Es relevante, útil e importante para saber si la acción que se está comprando no es un desperdicio de inversión y realmente tiene un valor para el mercado y para las estrategias propias de cada comprador.

4. Explicación de la modificación

Con los resultados que veremos en la sección siguiente notamos que los retornos inicialmente tienen un comportamiento tranquilo, esto es debido a que la mayoría de las variables utilizadas junto a las existentes previamente en el algoritmo, toman mucho más tiempo en mostrar rendimientos. A pesar de dicha dilatación inicial, se conserva siempre el comportamiento en el rango positivo con una pendiente de crecimiento que se extiende a más años la cual nos generará ganancias más significativas. Otro aspecto interesante es que se cuenta

con un porcentaje de retorno sobresaliente al igual que con una recurrencia que permite mantener ganancias a través del tiempo con la excepción de algún suceso extraordinario. El aspecto a resaltar es la propia recurrencia del buen comportamiento del algoritmo al pasar el tiempo y se debe tener más presente como fuente de un mejor punto de partida para analizar el cual para empezar sería el comportamiento ideal en un algoritmo de trading. Es por lo anterior que desde la perspectiva del autor de este informe, resulta mucho más útil tener un crecimiento lento y seguro que rápido e impredecible con retornos elevados solo hasta el final de los 2 años y medio que dura en ejecución el algoritmo y que es el tiempo propuesto.

5. Nuevos resultados

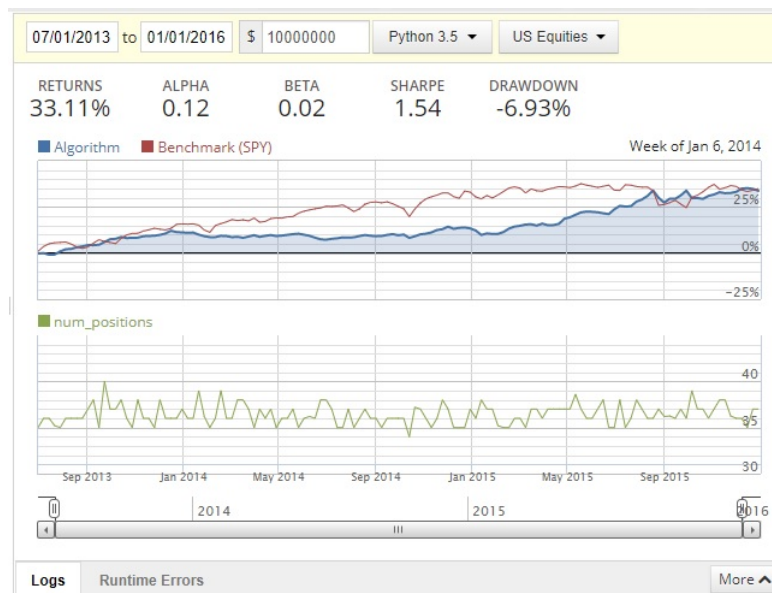


Figura 1: Resultados versión modificada del algoritmo