Reporte semanal: Septiembre 14-23

Jorge Ballote

23 de septiembre de 2021

1. Introducción

A continuación se presentan los resultados de la última semana. En esta ocasión se ha incluído unas conclusiones generales, de todo el proyecto.

2. Objetivos de la semana

Esta semana fue destinada a lo siguiente.

1. Mejorar el porcentaje de participación en el mercado utilizando símbolos con poca correlación.
2. Creación de una Interfaz Gráfica para recibir señales

3. Desarrollo de los objetivos

3.1. Uso de diferentes símbolos

Hasta ahora, se han realizado experimentos unicamente analizando datos del S&P500, en particular con el símbolo SPY. Sin embargo, cómo se ha visto en los reportes anteriores, para alcanzar un porcentaje de accuracy alto, se requiere tener cierto umbral T^+ y T^-. Esto implica que algunos días no se obtengan señales (ni de compra, ni de venta). Para aumentar la participación, lo que haremos será incrementar la cantidad de símbolos que analizamos. Sin embargo, ¿Qué símbolos son los apropiados?.

Optimizar miles de hiperparámetros es una tarea demasiado costosa, sobre todo asumiendo que cada cierto tiempo deberían optimizarse, por consiguiente es necesario hacer una selección adecuada y reducida de los símbolos con los que se trabajará. Por otro lado, hay empresas que no llevan mucho tiempo en el mercado y esto podría implicar una muestra muy baja de información ara entrenar nuestros modelos. Para seleccionar los nuestros símbolos, lo que se hizo fue tomar los 511 de NASDAQ que comenzaron a cotizar en el mercado desde al menos el año 2000. Se seleccionó una lista donde cada pareja de elementos tuvieran baja correlación.

3.1.1. Correlación Pearson

En estadística se utiliza para medir la correlación lineal que tienen dos variables aleatorias. Dadas n muestras $\{(x_i, y_i)\}_{i=1}^n$, la correlación se obtiene de la siguiente manera:

$$r = \frac{\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^n (x_i - \overline{x})^2}\sqrt{\sum_{i=1}^n (y_i - \overline{y})^2}} \tag{1}$$

y de manera resumida el valor máximo de r es $r = 1$, cuando se tiene una correlación positiva completa. Si $r = 0$ no existe ninguna correlación y si $r = -1$ existe una correlación negativa perfecta. Por consiguiente nosotros buscamos una correlación donde $|r| \to 0$.

3.1.2. Los símbolos seleccionados

Para obtener la lista de símbolos, seguimos un algoritmo muy sencillo. Empezamos con una lista L que contenía únicamente a SPY. Iteramos en nuestros 511 símbolos y añadimos aquellos que tengan una correlación menor a $u = 0.145025$ con todos los elementos de L.

3.2. Interfaz Gráfica

Este ha sido un proyecto desarrollado en python 3.9. La documentación se encuentra disponible en github: `https://github.com/ealjkj/Financial-Analysis`.

4. Metodología

Corrimos nuestro algoritmo genético durante 20 generaciones por experimento. Con una población de tamaño 30. La probabilidad de crossover fue seleccionada de 0.9 y la de mutación de 0.5. Se utilizó un algoritmo elitista, donde las 5 mejores soluciones permanecían siempre como miembros de la población. Tal como se describió, la métrica que nos interesa es $\text{mín}(5p, 1)(2a - 1)$, de modo que esa será la métrica a optimizar en nuestro algoritmo genético. En particular, tomando p : 'part above' y a : 'acc above'

Para la presentación de resultados, usaremos 2 formas de evaluar nuestro algoritmo.

1. **Desempeño en los últimos días:** Simplemente, ¿Si hubiésemos usado nuestro algoritmo en los últimos d días, ¿Cuántas veces hubiésemos acertado?

2. **K-Cross Validation** El k-cross validation consiste en particionar nuestro conjunto de datos en K partes diferentes. De modo que entrenamos k veces nuestro modelo, utilizando $k - 1$ conjuntos de la partición para el entrenamiento y el conjunto restante para la validación. Finalemente, se suele tomar la media de los k resultados obtenidos

Para la optimización de hiperparámetros se utilizaron $n - 365$. Para la prueba del modelo, se utilizaron 365 días de nuestro vector de precios. Cada posible cromosoma de nuestro algoritmo, indujo un conjunto de traces, los cuales fueron utilizados para conseguir la puntuación del k-cross validation con $k = 4$.

Las métricas a consideran serán las utilizadas en el reporte previo. Incluyendo el criterio 2.

5. Resultados

Después de 25 generaciones, nuestra Fitness function alcanzó un valor máximo de 0.16403. Al hacer la prueba de los últimos 365 días se obtienen muy buenos resultados, lo cuál sugiere que los parametros obtenidos son capaces de generalizar.

Hiperparámetros	
trace_size	95
height	20
width	95
grid_type	wins
operation_type	W/L
alpha_plus	0.825204
alpha_minus	0.686667
eliminate_noise_thold	0.000251377

$\Rightarrow$

Resultados	
criteria2_above	**0.232**
accuracy	53.75
participation	47.39
acc_above	61.60
part_above	30.68
acc_below	39.34
part_below	16.71

5.1. Alteración: $s \to \sin(s)$

Hiperparámetros	
trace_size	79
height	4
width	79
grid_type	wins
operation_type	W/L
alpha_plus	0.00423783
alpha_minus	0.131058
eliminate_noise_thold	0.0091428

$\Rightarrow$

Resultados	
criteria2_above	**0.219**
accuracy	50.70
participation	96.71
acc_above	60.97
part_above	44.93
acc_below	41.79
part_below	51.78

En el proceso de entrenamiento, obtuvo en promedio `criteria2 = 0.123`. En la prueba de los últimos 365 días, se obtuvo que `criteria2 = 0.219`.

5.2. Alteración: $s \to \cos(s)$

Hiperparámetros	
trace_size	97
height	52
width	97
grid_type	earnings
operation_type	W/T
alpha_plus	0.171751
alpha_minus	0.0240919
eliminate_noise_thold	0.0394345

$\Rightarrow$

Resultados	
criteria2_above	**0.133**
accuracy	50.99
participation	96.71
acc_above	56.69
part_above	69.58
acc_below	36.36
part_below	27.12

En el proceso de entrenamiento, obtuvo en promedio `criteria2_above = 0.123`. En la prueba de los últimos 365 días, se obtuvo que `criteria2_above = 0.133`.

5.3. Alteración: $s \to \tan(s)$

Hiperparámetros	
trace_size	45
height	95
width	45
grid_type	wins
operation_type	W/L
alpha_plus	0.0358503
alpha_minus	0.103658
eliminate_noise_thold	0.0391943

$\Rightarrow$

Resultados	
criteria2_above	**0.133**
accuracy	50.86
participation	95.34
acc_above	56.66
part_above	65.75
acc_below	37.96
part_below	29.58

. . . . . .

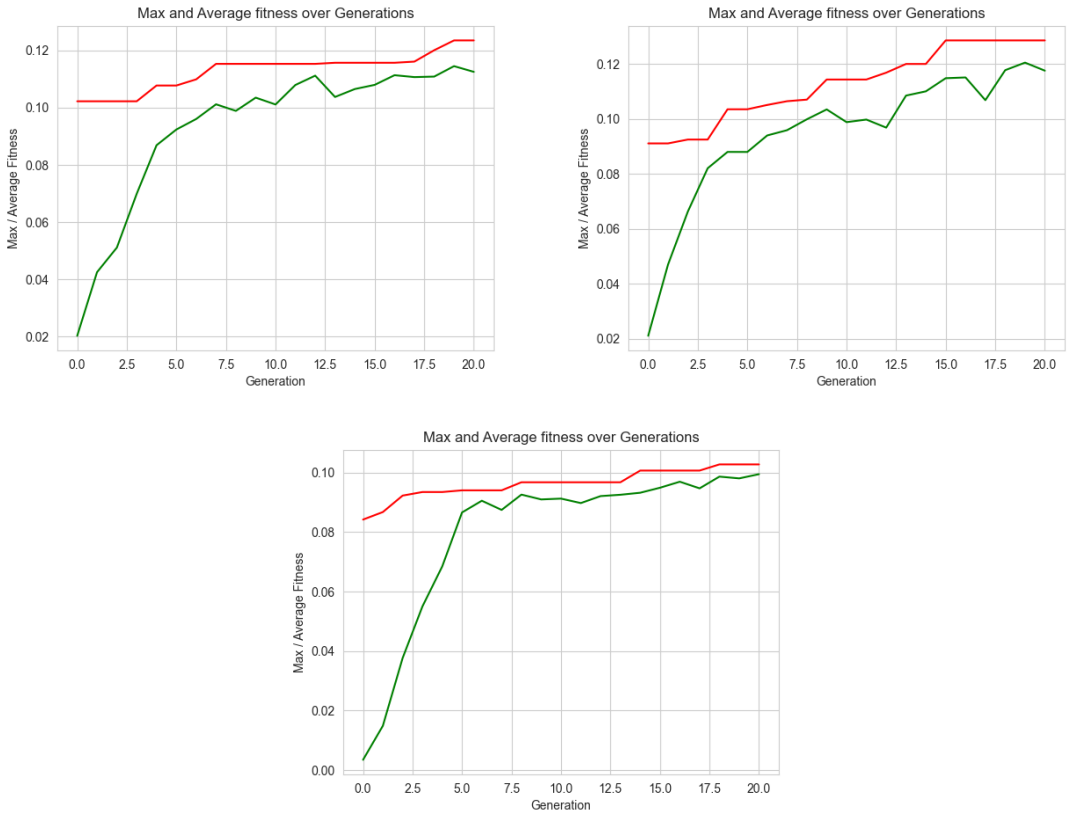

Figura 1: Gráficas de evolución con los operadores sin, cos, tan, respectivamente. La línea roja representa el valor máximo de la fitness function y la línea verde es la media de la población.

6. Conclusiones

Pese a que no se documentaron los experimentos anteriores, es importante destacar que sin la implementación del k-cross validation, la optimización resulta inútil. Lo importante no es que obtenga resultados muy buenos en nuestro conjunto de entrenamiento, sino que pueda generalizar con datos que no ha visto.

Por otro lado, en términos generales, la implementación de las funciones trigonométricas no parecen aportar demasiado aunque cabe destacar que la función $\sin(x)$ parece tener resultados similares. Tal vez puedan resultar útiles en caso de querer tener varias señales.

Los algoritmos genéticos demostraron ser muy eficientes para optimizar hiperparámetros. En la mayoría de los casos, los resultados en el conjunto de prueba incluso superaban a los del conjunto de entrenamiento. Esto sólo sugiere que se generaliza bien, pero no debe asumirse que el desempeño será mejor siempre en el conjunto de prueba.

Algo interesante es que en la práctica si nuestra fitness function se basa en criteria2_above, los resultados mejoran con respecto a una fitness function basada en el simple criteria2, cuando

en teoría ésta última fitness function, debería optimizar simultáneamente el criteria2_above y el criteria2_below.