Proyecto Estadística y Optimización

Jeremy Joel Aguilar Marin, Irene Barba La Orden, Lucía Benages Guijarro, Fabián Calvo Castillo, Cristhian Torrico Castellon

2025-12-15

2. Predicción del Rendimiento

En todas las series temporales se observa que los residuos en el tiempo oscilan alrededor de cero sin patrones visibles, tampoco muestran tendencia ni cambios bruscos de varianza.

El ACF de residuos, la mayoría de los lags están dentro de las bandas de confianza y no hay autocorrelación significativa. Y el PACF de residuos es igual que la ACF: sin picos relevantes. No hay estructura remanente por modelar.

Para poder hacer un pronóstico de los rendimientos futuros de los 5 activos del CSV stock_returns_trains_2.csv hemos empleado modelos ARIMA univariados, ajustados mediante la función `auto.arima` (correspondiente a la librería: `forecast`). Esto lo que nos quiere decir es que si el rendimiento de hoy depende un poco del rendimiento de ayer, el modelo ARIMA lo detecta, ya que nuestro modelo ARIMA lo que hace es detectar los patrones en el tiempo, todo y que estos patrones sean o no insignificantes. Además, hay que tener en cuenta que ARIMA NO sobreajusta, no se inventa patrones inexistentes. Lo explicado con anterioridad es adecuado para series financieras con autocorrelación débil.

A partir de esto, pudimos obtener dos variables fundamentales:

- El rendimiento esperado del activo (μ_t), es decir, lo que el modelo anticipa que podría ocurrir
- La desviación estándar (σ_t), que refleja el nivel de riesgo asociado a esa predicción.

Para poder comprobar todo estuviera bien ajustado, hemos procedido a comprobar los residuos con dos pruebas estadísticas habituales en series temporales:

- Ljung–Box (mediante Box.test) –> que detecta si queda autocorrelación no explicada
- ARCH (mediante ArchTest de la libreria *FinTS*) –> revisa si existe heterocedasticidad

Con esto podemos verificar que nuestro trabajo es razonable, y gracias a esto podemos verificar los apartados 2,3 y 4.

En el apartado 2, usamos la función `getPred()`, la cual nos ajusta un modelo ARIMA con los datos del activo para posteriormente actualizar los datos con la información más reciente hasta el día $t-1$ (con el fin de tener los datos mas actuales posibles). A partir de este modelo (ya actualizado) se obtiene:

- una predicción del rendimeinto del activo del siguiente día mañana ($\hat{\mu}_t$)
- una estimación de la incertidumbre del modelo ($\hat{\sigma}_t$) –> para saber que tan segura es nuestra predicción

A continuación, tenemos la función `getPred_ts()`. Esta lo que hace es repetir el proceso de predicción día a día y activo por activo. Su procesos es sencillo:

1. Empieza en el primer día del periodo test

2. Calcula la predicción de los activos
3. Guarda los datos
4. Hace lo mismo con el siguiente día, e incorpora toda la información.

Despues "construye" dos matrices:

- `mu_hat` –> Predicción del rendimiento del activo para el día t
- `se_hat` –> Desviación estándar estimada para el día t

Finalmente, se calcula el RMSE para evaluar la calidad de las predicciones. En nuestro caso, el RMSE obtenido fue de 0.0175, lo que indica que, en promedio, el modelo comete un error del 1.75% en la predicción de los rendimientos diarios. Este nivel de error es razonable para datos financieros, que suelen presentar alta volatilidad.

3. Utilidad Media-Varianza

A continuación se describen las funciones utilizadas para obtener los pesos óptimos, α_t, para cada activo en $t = T + 1, ..., T + r$ y el rendimiento acumulado total. Tendremos en cuenta dos casos: (3.1) permitiendo posiciones cortas y (3.2) restringiéndolas ($\alpha_t \geq 0$). Para ello se debe optimizar la función utilidad media-varianza, U_{MV}, para obtener estos pesos y así obtener el mejor rendimiento mitigando el riesgo.

Si nos fijamos en la función utilidad media-varianza,

$$U_{MV} = \alpha_t^T \mu_t - \frac{\gamma}{2} \alpha_t^T \Sigma_t \alpha_t,$$

μ_t corresponden a los valores esperados de los rendimientos calculados en el apartado anterior; Σ_t es una matriz calculada a partir de la función `getSigmaMV` (como se indica en el enunciado); γ es el coeficiente de aversión al riesgo (que controla el peso que se le da a la varianza).

Ahora bien, considerando el primer caso (permitiendo posiciones cortas), para obtener un peso en un periodo t utilizamos la función `getAlphaMV`. Esta función ha sido determinada a partir del método de los multiplicadores de Lagrange bajo la restricción $\sum_i \alpha_{ti} = 1$. Y para calcular los pesos para cada activo, hemos utilizado la función `getAlpha_ts` (proporcionada) donde sus variables de entrada son `mus` (μ_t), `sigs` (se_{hat}), γ (que hemos fijado a 20, penalizando fuertemente el riesgo), `getSigmaFunc` que es `getSigmaMV`, `getAlphaFunc` es `getAlphaMV`, y `Xtrain` y `Xtest` los rendimientos para train y test. También, dentro de `getAlpha_ts` se llama a la función `getAlpha` que combina las funciones `getAlphaMV` y `getSigmaMV`. Por último, hemos utilizado la función `getChecks` para verificar que se cumple la restricción mencionada anteriormente ($\sum_i \alpha_{ti} = 1$).

Y para calcular el rendimiento total hemos utilizado la función `getRet` (proporcionada). En este caso hemos obtenido un rendimiento total de 2.95.

Por otro lado, para el caso de posiciones largas (3.2), hemos realizado exactamente el mismo procedimiento, excepto para el cálculo de α_t ya que tenemos que tener en cuenta que solo se consideran valores positivos (además de la restricción de la suma de todos los pesos). Hemos utilizado la función `getAlphaMVPos` donde se ha hecho uso de `solve.QP` del paquete `quadprog`, que permite resolver problemas de optimización cuadrática sujeta a restricciones lineales.

En este caso hemos obtenido un rendimiento de 0.62. A diferencia del caso anterior donde habíamos obtenido un 2.95. Puede ser debido a que al restringir únicamente a pesos positivos, limitamos más las combinaciones posibles (elimina la posibilidad de de neutralizar riesgo con cortos) y, por tanto, reduce las posibilidades de obtener una mejor rentabilidad.

4. Log-Utilidad

En esta sección trataremos de optimizar nuestra cartera de inversiones respecto de la función de utilidad logarítmica, que reconoce la naturaleza compuesta del rendimiento del portafolio. Concretamente se aproximará la función por Taylor a segundo orden, obteniendo:

$$U_{log}(\alpha_t; \mu_t; \Sigma_t) \approx log(1 + \alpha_t^T \mu_t) - \frac{\gamma}{2} \frac{\alpha_t^T \Sigma_t \alpha_t}{(1 + \alpha_t^T \mu_t)^2}$$

Escogeremos, al igual que en el apartado anterior, un valor para el parámetro de riesgo $\gamma = 20$, y realizaremos también la misma estimación para la matriz Σ_t.
También se permitirá en este caso utilizar posiciones cortas respecto de los activos, es decir, valores de α_{ti} negativos, pero están sujetos a la restricción de que deben sumar 1: $\sum_i \alpha_{ti} = 1$.
Para realizar la optimización en primer lugar redefiniremos la función U_{log} añadiendo un término de penalización, que minimizará la función si no se respeta $\sum_i \alpha_{ti} = 1$, la función queda entonces con la forma:

$$U_{log}^{penal}(\alpha_t; \mu_t; \Sigma_t) \approx log(1 + \alpha_t^T \mu_t) - \frac{\gamma}{2} \frac{\alpha_t^T \Sigma_t \alpha_t}{(1 + \alpha_t^T \mu_t)^2} - \rho(\Sigma_i \alpha_{ti} - 1)^2$$

Tras realizar la optimización se verificará que esta restricción se verifica y si no se aumentará el parámetro de penalización ρ.
Para encontrar el α óptimo se utilizará la función `optim` de R, el proceso tendrá dos fases: en un primer lugar utilizaremos el método *Nelder-Mead* partiendo del punto $\alpha^T = (0.2, 0.2, 0.2, 0.2, 0.2)$ para encontrar un primer óptimo y con ello una región factible, después mejoraremos el óptimo encontrado aplicando el método *BFGS* sobre el resultado. Este proceso se incluye en la función `getAlphaLog` que se aplicará sobre todo el periodo de test con la función `getAlpha_ts` y tras ello evaluaremos su rendimiento con `getRet` y la utilidad relativa logarítmica con `Ulog_rel`.
Obtenemos para nuestro periodo de test un rendimiento de 4.5954 y una utilidad relativa de 7.8182.

5. Resultados

Los resultados de las predicciones en estos casos nunca son exactos porque en mercados financieros los retornos son muy ruidosos, y el valor esperado suele ser más suave. En este caso, en la imagen de la $figura1$, se observa cómo en los 5 activos la línea azul (predicha) es más estable, pues ARIMA filtra el ruido de alta frecuencia. Concretamente, en los activos 1, 4 y 5 se observan cambios de nivel, ARIMA se está adaptando lentamente (esto puede ser debido a alguna dependencia temporal débil). Y los activos 2 y 3 son los más cercanos a ruido blanco. Su retorno retorno real oscila al rededor de 0, y su predicción es más aplanada. En la imagen de la $figura2$ se observa la representación de cómo se asiganan los pesos promedio de inversión en cada activo en el periodo de testeo bajo el modelo de utilidad media-varianza con posiciones cortas permitidas.

El activo 5 tiene el peso promedio más alto (~0.8), es decir, posee un rendimiento esperado alto y una volatilidad controlad, el modelo confía bastante en este activo. Los activos 1 y 4 presentan pesos positivos moderados (~0.4), son buenos, pero no hay tanta convicción. Los activos 2 y 3 muestran características menos atractivas (ya sea por bajo rendimiento esperado o alta volatilidad relativa), lo que lleva al modelo a asignarles posiciones cortas.

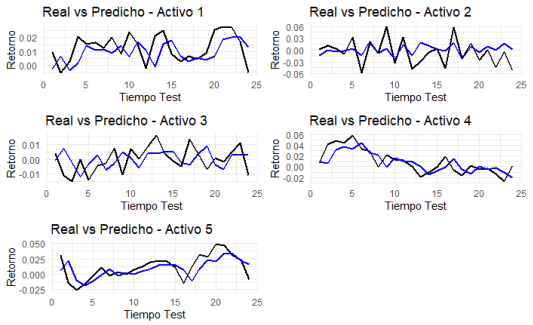

Figure 1: Real vs Predicha

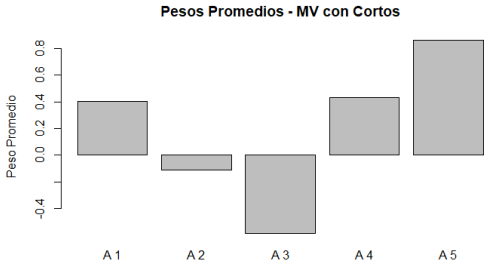

Figure 2: Pesos Promedios

La gráfica siguiente ilustra cómo ha cambiado el portafolio creado bajo la norma de utilidad logarítmica, cuyo objetivo es optimizar el crecimiento compuesto del capital. Este método, a diferencia de la utilidad media-varianza, castiga con severidad las pérdidas y promueve tácticas más conservadoras en situaciones de volatilidad alta. Para calcular el valor acumulado, se utilizan los rendimientos ponderados por los pesos óptimos en cada momento y, posteriormente, se suman los factores de crecimiento. La curva final muestra una tendencia ascendente, acelerándose al final del periodo, lo cual sugiere que el modelo fue capaz de identificar oportunidades de inversión con buena rentabilidad y modificar dinámicamente la distribución de activos en función del riesgo y el rendimiento esperado.

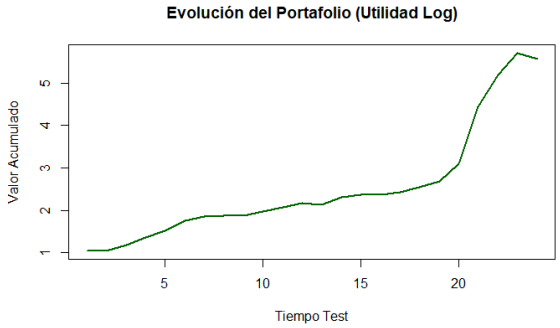

Figure 3: Uitilidad Log

Resumen Resultados Finales

RMSE	Rend. UMV ($\alpha \in \mathbb{R}$, $\gamma = 20$)	Urel MV ($\alpha \in \mathbb{R}$, $\gamma = 20$)	Rend. UMV ($\alpha \in \mathbb{R}^+$, $\gamma = 20$)	Urel MV ($\alpha \in \mathbb{R}^+$, $\gamma = 20$)	Rend. Ulog ($\alpha \in \mathbb{R}$, $\gamma = 20$)	Urel log ($\alpha \in \mathbb{R}$, $\gamma = 20$)
0.0175	2.9515	7.0117	0.6212	3.1182	4.5954	7.8182

Distribución Trabajo

Apartado 2: Jeremy Joel Aguilar Marin. Apartado 3: Irene Barba La Orden - Fabián Calvo Castillo. Apartado 4: Lucía Benages Guijarro. Apartado 5: Cristhian Torrico Castellon.