



Universidad de San Andrés

Escuela de Administración y Negocios

Magíster en Finanzas

**Aplicaciones de aprendizaje reforzado para el arbitraje
estadístico**

Autor: Matias Data

DNI: 35.380.450

Director de Trabajo Final de Graduación: Manuel Maurette

Codirector de Trabajo Final de Graduación: Elsa Cortina

Junio, 2023

Índice

1. Introducción	1
2. Revisión literaria	3
2.1. Arbitraje estadístico	3
2.2. Aprendizaje reforzado	5
3. Marco teórico	7
3.1. Aprendizaje estadístico	7
3.1.1. Análisis de componentes principales	7
3.1.2. Regresión por componentes principales	10
3.1.3. El método de <i>Bootstrap</i>	10
3.2. Aprendizaje reforzado	13
3.2.1. Programación dinámica estocástica	13
3.2.2. El algoritmo de programación dinámica	15
3.2.3. El método de <i>Rollout</i>	16
3.2.4. Anticipación multipaso	17
3.2.5. El método del horizonte móvil	18
3.2.6. Problemas de información parcial de los estados	20
3.3. Arbitraje estadístico	22
3.3.1. Portafolios de arbitraje	22
3.3.2. Neutralidad a los factores y al mercado	24
3.3.3. Señales de arbitraje	25
3.3.4. Políticas de asignación de posiciones	27
3.3.5. Apalancamiento, costos de transacción y <i>P&L</i>	28
4. Datos	32
5. Metodología	33
5.1. Estimación de los parámetros de los modelos	33
5.1.1. Estimación del modelo de factores	33

5.1.2. Estimación del modelo de series de tiempo	35
5.2. Estrategias de aprendizaje reforzado	36
5.2.1. Planteamiento del problema de programación dinámica estocástica . . .	36
5.2.2. La estrategia <i>naive rollout</i>	40
5.2.3. La estrategia <i>rollout with bootstrap</i>	41
6. Resultados	43
6.1. Resultados en un entorno de simulado	43
6.2. Resultados por <i>back-testing</i> en datos históricos	50
7. Conclusiones y nuevas líneas de investigación	61
A. Demostraciones	66



Universidad de
San Andrés

Índice de figuras

1.	Autovectores de la descomposición PCA para los movimientos en <i>U.S. Treasury yields</i> para el período 2005-2023.	8
2.	Gráfico de $(X_t)_t$, una serie de tiempo $AR(1)$ simulada con $\mu = 0, \phi = 0,9, \sigma_\nu = 0,005$, para 1000 días.	43
3.	Retorno acumulado de las estrategias para un activo simulado $AR(1)$ con $\phi = 0,9, \sigma_\nu = 0,005$, exposición por activo de $\Lambda = 2$ dólares, costo de transacción $\xi = 0,0005$ y sin costos de tenencia (i.e. $\delta = 0$).	44
4.	Distribución de retornos individuales acumulados obtenidos en una ventana de $H = 10$ días para $N = 10000$ activos simulados $AR(1)$ con $\mu = 0, \phi = 0,9, \sigma_\nu = 0,005$ desconocidos (es decir, estimados por los algoritmos), exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costos de tenencia $\delta = 0,0001$	45
5.	Retorno acumulado en 60 días para $N = 1000$ activos simulados $AR(1)$ $\phi = 0,9, \sigma_\nu = 0,005$, exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costo de tenencia $\delta = 0,0001$	46
6.	Regresión por núcleos del retorno acumulado en 10 días en función de X_0 estandarizado para cada estrategia. Para $N = 10000$ activos simulados $AR(1)$ $\phi = 0,9, \sigma_\nu = 0,005$, exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costo de tenencia $\delta = 0,0001$	47
7.	Clasificación por redes neuronales del control elegido en función de X_0 para cada estrategia. Para $N = 10000$ activos simulados $AR(1)$ $\phi = 0,9, \sigma_\nu = 0,005$, exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costo de tenencia $\delta = 0,0001$	48
8.	Regresión del retorno esperado en función de ϕ para cada estrategia. Para $N = 1000$ activos simulados $AR(1)$ para cada ϕ con $\sigma_\nu = 0,005$, exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costo de tenencia $\delta = 0,0001$	49
9.	Regresión del retorno esperado en función de σ_ν para cada estrategia. Para $N = 1000$ activos simulados $AR(1)$ para cada σ_ν con $\phi = 0,9$, exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costo de tenencia $\delta = 0,0001$	49

10.	Retorno acumulado de los portafolios de cada estrategia con $K = 5$ factores de PCA, sin costos de transacción, para un <i>back-testing</i> desde el 2 de enero de 2014 al 18 de mayo de 2023.	52
11.	Retorno acumulado de los portafolios de cada estrategia con $K = 8$ factores de PCA, sin costos de transacción, para un <i>back-testing</i> desde el 2 de enero de 2014 al 18 de mayo de 2023.	52
12.	Retorno acumulado de los portafolios de cada estrategia con $K = 10$ factores de PCA, sin costos de transacción, para un <i>back-testing</i> desde el 2 de enero de 2014 al 18 de mayo de 2023.	53
13.	Retorno acumulado de los portafolios de cada estrategia con $K = 15$ factores de PCA, sin costos de transacción, para un <i>back-testing</i> desde el 2 de enero de 2014 al 18 de mayo de 2023.	53
14.	Retorno acumulado de los portafolios de cada estrategia con $K = 8$ factores de PCA, con costo de transacción $\xi = 0,0002$ (i.e. 2 <i>bps</i>) y β calculado por OLS, en un <i>back-test</i> desde el 2 de enero de 2014 al 18 de mayo de 2023.	55
15.	Retorno acumulado de los portafolios de cada estrategia con $K = 8$ factores de PCA, sin costo de transacción ($\xi = 0$) y β estimados con OLSTF, en un <i>back-test</i> desde el 2 de enero de 2014 al 18 de mayo de 2023.	56
16.	Retorno acumulado de los portafolios de cada estrategia con $K = 8$ factores de PCA, con costo de transacción de 2 <i>bps</i> y β estimados con OLSTF, para un <i>back-testing</i> desde el 2 de enero de 2014 al 18 de mayo de 2023.	57
17.	Retorno acumulado de los portafolios de cada estrategia con $K = 8$ factores de PCA, sin costo de transacción ($\xi = 0$), con regresión robusta de series de tiempo (<i>Huber loss</i>) y β estimados con OLS, para un <i>back-testing</i> desde el 2 de enero de 2014 al 18 de mayo de 2023.	59
18.	Retorno acumulado de los portafolios de cada estrategia en un <i>back-test</i> desde el 2 de enero de 2003 al 30 de diciembre de 2011. Se utilizan $K = 8$ factores de PCA, sin costo de transacción de y β estimados por OLS.	60

Índice de cuadros

1.	Estrategia de Avellaneda y Lee (2010)	28
2.	Número de operaciones por estrategia en la simulación de la figura 4, retorno medio (μ) y volatilidad (σ) por activo, y <i>Sharpe ratio</i> aproximado de 10 días de las estrategias para 500 activos.	45
3.	Retorno acumulado y <i>Sharpe ratio</i> mensual por estrategia en la simulación de la figura 5.	46
4.	Resultados por número de factores de PCA (K), estrategia, <i>Sharpe ratio</i> (SR), retorno medio (\bar{R}_{II}), retorno excedente medio (μ), y volatilidad (σ) en términos anuales de los portafolios en un <i>back-test</i> desde el 2 de enero de 2014 al 18 de mayo de 2023. Sin costos de transacción, es decir $\xi = 0$, $\delta = 0$	51
5.	Resultados para $K = 8$ factores de PCA por estrategia para las estrategias con costo de transacción $\xi = 0,0002$ (i.e. 2 <i>bps</i>) y β calculado por OLS. <i>Sharpe ratio</i> (SR), retorno medio (\bar{R}_{II}), retorno excedente medio (μ), y volatilidad (σ) en términos anuales de los portafolios en un <i>back-test</i> desde el 2 de enero de 2014 al 18 de mayo de 2023.	55
6.	Resultados por estrategia para $K = 8$ factores de PCA para costos de transacción (ξ) de 0 <i>bps</i> y 2 <i>bps</i> , y β estimados con OLSTF. <i>Sharpe ratio</i> (SR), retorno medio (\bar{R}_{II}), retorno excedente medio (μ), y volatilidad (σ) en términos anuales de los portafolios para un <i>back-testing</i> desde el 2 de enero de 2014 al 18 de mayo de 2023.	56
7.	Resultados para $K = 8$ factores de PCA por estrategia para los modelos con regresión robusta de series de tiempo (<i>Huber loss</i>). <i>Sharpe ratio</i> (SR), retorno medio (\bar{R}_{II}), retorno excedente medio (μ), y volatilidad (σ) en términos anuales de los portafolios en un <i>back-test</i> desde el 2 de enero de 2014 al 18 de mayo de 2023. Sin costos de transacción, es decir $\xi = 0$, $\delta = 0$, y β estimados por OLS.	58

8. Resultados para $K = 8$ factores de PCA por estrategia. *Sharpe ratio* (SR), retorno medio (\bar{R}_{Π}), retorno excedente medio (μ), y volatilidad (σ) en términos anuales de los portafolios para un *back-testing* desde el 2 de enero de 2003 al 30 de diciembre de 2011. Sin costos de transacción, es decir $\xi = 0$, $\delta = 0$, y β estimados por OLS. 59



Universidad de
San Andrés

Resumen

El objetivo de esta tesis es el de aplicar técnicas de aprendizaje reforzado para generar estrategias de *trading* basadas en el arbitraje estadístico para el mercado de *large cap U.S. Equities*. En particular, se analiza si las estrategias obtenidas mejoran a las estrategias presentes en la literatura, tanto en un entorno de simulación como en datos históricos.



Universidad de
San Andrés

1. Introducción

La definición formal de un arbitraje consiste en la existencia de un portafolio que obtenga un beneficio siempre no negativo, positivo con probabilidad mayor a cero, y cuya inversión inicial sea a costo cero o negativo. Informalmente, un arbitraje constituye una oportunidad de obtener dinero gratis (“*money for nothing*”). Equivalentemente, si se tienen dos portafolios A y B tales que el *payoff* del portafolio A es mayor o igual al de B , y con probabilidad positiva es mayor, entonces el precio del portafolio A es mayor al de B . La ausencia de arbitraje es el concepto fundamental de las finanzas modernas. Una noción más débil es la ley del único precio (*Law of one price, LOOP*) que nos dice que dos portafolios con igual *payoff* en todo estado deben tener igual precio. Si este no fuera el caso, se podría comprar el portafolio barato y vender el caro obteniéndose una ganancia instantánea sin riesgo alguno. Este supuesto básico resulta equivalente a la existencia de una representación por *betas*, y a la existencia de un retorno en la frontera eficiente (ver Cochrane 2009, capítulo 6).

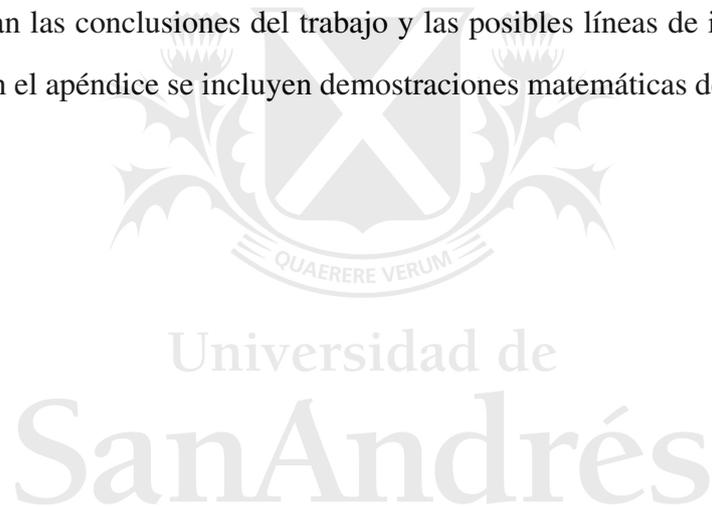
Ahora bien, en la vida real rara vez uno encuentra oportunidades de arbitraje o violaciones de la ley del único precio verdaderamente operables. La mayoría de las mal llamadas “oportunidades de arbitraje” (e.g. *statistical arbitrage, corporate/risk arbitrage, volatility arbitrage, etc.*) que algunos actores del mercado (e.g. *hedge funds, investment advisors, etc.*) explotan no constituyen un arbitraje en el sentido estricto de la definición. En la práctica, usualmente construyen estrategias cuyo *payoff* es positivo con alta probabilidad, pero en general hay una probabilidad no nula de perder dinero. El arbitraje estadístico (*statistical arbitrage*) constituye uno de estos estilos de inversión, que se basa en el uso de señales construidas mediante métodos estadísticos para encontrar oportunidades de inversión con retornos esperados altos usualmente con portafolios que se construyen neutrales al mercado.

El aprendizaje reforzado (*Reinforcement Learning, RL*) es una rama del aprendizaje estadístico (*Machine Learning/Statistical Learning, ML*) cuyo objetivo es optimizar problemas de decisión multi-período usualmente estocásticos. Las estrategias de arbitraje estadístico que se conocen de la literatura en general son *greedy* y solo intentan optimizar el retorno a un día, sin tener en cuenta el retorno o costos de transacción que se originarán en los días subsiguientes al abrir una posición.

La afirmación principal de esta tesis es que es posible mejorar las estrategias de arbitraje

estadístico conocidas mediante metodogías del aprendizaje reforzado. Considerando una ventana de tiempo y las decisiones futuras a realizar en esta ventana, es posible encontrar estrategias con rendimientos estadísticamente superiores a las estrategias básicas *greedy*.

Este trabajo se estructura de la siguiente forma: en la sección 2 se hace una revisión de la literatura, abarcando los trabajos del arbitraje estadístico y del aprendizaje reforzado que presentan técnicas generales de estas áreas; en la sección 3 se explican los métodos de aprendizaje estadístico y aprendizaje reforzado que se utilizan en esta tesis, y las estrategias del arbitraje estadístico que se usan como base para entrenar nuevas estrategias; en la sección 4 se detalla la forma en la que se obtuvieron los datos históricos que se usan para evaluar las estrategias; en la sección 5 se explican los nuevos métodos introducidos en esta tesis así como los detalles de la estimación de los modelos; la sección 6 contiene los resultados empíricos obtenidos. Por último, la sección 7 detallan las conclusiones del trabajo y las posibles líneas de investigación futura. Por completitud, en el apéndice se incluyen demostraciones matemáticas de algunos resultados utilizados.



2. Revisión literaria

2.1. Arbitraje estadístico

El Arbitraje Estadístico (*Statistical Arbitrage*) constituye una de las estrategias más populares en *hedge funds* cuyo *trading* es sistémico o algorítmico (ver Avellaneda 2013). Se origina en los 90s con la estrategia de *pairs trading* en Morgan Stanley, basada en encontrar activos cuyo *spread* sea estacionario y abrir posiciones *long-short* para operar el *spread* entre las mismas.

El trabajo de Avellaneda y Lee (2010) es fundacional dentro del área ya que presenta una formulación general que siguen este tipo de estrategias e introduce una estrategia simple basada en el análisis de componentes principales (*Principal Components Analysis, PCA*) y el modelo de Ornstein-Uhlenbeck (OU). De acuerdo a este artículo las principales características de Arbitraje estadístico son:

- i) Las señales de *trading* se obtienen de forma sistemática, es decir basado en reglas no discrecionales.
- ii) Se aproxima a ser neutral al mercado o, más generalmente, neutral a los factores de riesgo identificados.
- iii) Se basa en utilizar métodos estadísticos para identificar y explotar retornos excedentes.

En un primer paso se aplica PCA para “replicar” un activo mediante un portafolio de activos, o alternatively se utilizan *Exchange Traded Funds* (ETFs) sectoriales para construir un portafolio *long-short*, llamado portafolio de arbitraje, cuyo retorno esperado (incondicional) sea cero. Luego se plantea que en el largo plazo este portafolio debería tener retorno cero y por lo tanto, las desviaciones del retorno de este portafolio respecto a su media deben ser temporales y revertirse en el tiempo. Se modelan entonces los retornos de este portafolio con un modelo de tiempo continuo de Ornstein-Uhlenbeck. Al discretizarlo se obtiene un modelo autoregresivo de series de tiempo $AR(1)$. La estrategia de Avellaneda y Lee abre posiciones en este portafolio cuando su retorno acumulado se desvia de la media de forma significativa y las cierra cuando este revierte cercano a la media.

Otra condición que considera esta estrategia para decidir si abrir una posición es que la reversión a la media sea lo suficientemente rápida, esto se implementa mediante una restric-

cion de la forma $\kappa > \kappa_{crit}$, donde el parámetro κ es la velocidad de reversión del modelo de Ornstein-Uhlenbeck. Otro aporte importante del trabajo es cómo elegir la cantidad de factores de PCA, esto puede ser de forma estática o dinámica, ya que como observan durante las crisis el número de factores que explican un porcentaje alto de la varianza de los activos disminuye considerablemente. También introducen una mejora de la estrategia básica basada en considerar los datos de volumen de operación, lo que llaman *trading-time*. Las señales son re-escaladas de acuerdo al volumen con el que se operaron respecto del volumen medio, cuando operan con un volumen alto se disminuye su intensidad. Por lo tanto comprar un activo que cae fuertemente pero con alto volumen se ve desalentado.

En el artículo de Yeo y Papanicolaou (2017) se introduce una mejora a la estrategia de Avellaneda y Lee que filtra los activos basado en medidas de bondad de ajuste (*goodness-of-fit*). específicamente, utilizan el R^2 de la regresión de series de tiempo como medida de bondad de ajuste, si este no es suficientemente alto se restringe el activo para operar durante el periodo. En la práctica implementan esto descartando el 50 % de los activos con peores valores del R^2 . Otra forma en la que se puede implementar es simplemente una restricción para cada activo de la forma $R^2 > R^2_{crit}$.

Por último mencionamos el reciente *pre-print* de Pelger et al. (2021) que también desarrollan un marco conceptual para el arbitraje estadístico e introducen metodologías de Aprendizaje Profundo (*Deep Learning*) para construir estrategias. El marco conceptual coincide con las tres etapas del trabajo de Avellaneda y Lee, aunque ellos lo formalizan con mayor generalidad. Los tres pasos consisten en:

- i) Identificar activos similares para generar portafolios de arbitraje.
- ii) Extraer señales de desviación temporal de los activos similares.
- iii) Una política de asignación de posiciones en los portafolios de arbitraje.

En cuanto al primer punto utilizan nuevamente modelos de *asset pricing*, específicamente el modelo de 5 factores de Fama-French (ver Fama y French 1993; Fama y French 2015), factores basados en PCA, y factores latentes condicionales estimados con análisis de componentes principales instrumental (IPCA) (ver Kelly et al. 2019). En el segundo paso consideran como referente para comparar el modelo de Avellaneda y Lee (2010), al que modifican con una fun-

ción de asignación más general que la del cuadro 1 mediante un modelo de *Redes Neuronales Feedforward* (FNN). El segundo modelo que consideran es uno no paramétrico basado en la Transformada Rápida de Fourier donde luego se utiliza una FNN para filtrar los coeficientes pequeños removiendo el “ruido” en la representación. El tercer y principal modelo que introducen es uno que obtiene las señales utilizando Redes Neuronales Convolucionales (CNN) y transformadores, que forman una de las arquitecturas más exitosas para reconocimiento de patrones como visión computacional y procesamiento de lenguaje natural (NLP). Para el problema de asignación de pesos utilizan nuevamente FNN, que se plantea en todos los casos como un problema de optimización que se resuelve para datos de los primeros cuatro años que no se usan para evaluar el rendimiento del algoritmo, es decir la evaluación del algoritmo se hace *out of sample*. Una observación importante es que, al menos en las estrategias que estudian, lo más importante es la extracción de señales a la hora de mejorar el desempeño del algoritmo. Los distintos modelos de factores que se utilizan y la asignación de los pesos son importantes pero no fundamentales.

Por último, destacamos también los artículos de Boyd et al. (2013) y Boyd et al. (2017), que si bien no son particulares a estrategias de arbitraje estadístico, especifican una estructura flexible en la que plantear un problema de inversión multi-período y resolverlo mediante métodos de optimización convexa.

2.2. Aprendizaje reforzado

El aprendizaje reforzado tiene como objetivo encontrar soluciones aproximadas problemas de decisión multi-período estocásticos. Un problema con estas características en general puede ser resuelto de forma exacta en principio mediante el método de programación dinámica. Sin embargo, salvo en problemas pequeños o en los casos particulares que admiten soluciones analíticas, el método de programación dinámica resulta computacionalmente inabordable. Es en este caso que se aplican métodos de resolución aproximada para encontrar políticas subóptimas que tengan un buen rendimiento en la práctica. Es por eso que algunos autores se refieren al área como Programación dinámica aproximada (*Approximate Dynamic Programming*). Las técnicas utilizadas en RL incorporan ideas clásicas de la programación dinámica en conjunto métodos del Aprendizaje estadístico (*Statistical Learning*) o Aprendizaje de máquinas (*Machi-*

ne Learning, ML). El Aprendizaje reforzado profundo (*Deep Reinforcement Learning, DRL*) hace referencia al uso de aproximaciones por redes neuronales profundas para las funciones de valor o políticas.

Los libros de Bertsekas (2012), Bertsekas y Tsitsiklis (1996), Bertsekas (2019) y Bertsekas (2022a) dan una introducción al área desde la programación dinámica. Los autores son referentes del área en la que han probado importantes resultados teóricos de convergencia de varios métodos. La notación que utilizan es propia de la teoría de control y difiere de la utilizada mayoritariamente en el área.

La otra referencia que usamos es el libro de Sutton y Barto (2018), que es el libro de referencia canónico del área. Este libro tiene un enfoque más computacional ya que como la mayoría de los autores del tema, viene de las ciencias de la computación. El énfasis está aquí en explicar los algoritmos prestando atención a su implementación y evaluar eficiencia en la práctica más allá de los fundamentos teóricos.

También se consideran los libros de referencia del área de ML como Hastie et al. (2009) y Bishop (2006) para los detalles de implementación de métodos de regresión, clasificación y de redes neuronales utilizados en los algoritmos de Aprendizaje Reforzado. Para el método *bootstrap* se consultan las referencias Efron y Tibshirani (1986) y Efron y Tibshirani (1994).

3. Marco teórico

3.1. Aprendizaje estadístico

3.1.1. Análisis de componentes principales

Sea $X \in \mathbb{R}^{N \times p}$ una matriz de $N \times p$ que consiste de N observaciones (filas) de p variables o regresores (columnas). Toda matriz posee una descomposición por valores singulares (SVD) que la factoriza como

$$X = UDV^T. \quad (3.1)$$

donde $U \in \mathbb{R}^{N \times p}$, $D, V \in \mathbb{R}^{p \times p}$ con U y V matrices con columnas ortonormales y $D = \text{diag}(d_1, \dots, d_p)$ diagonal con $d_1 \geq d_2 \geq \dots \geq d_p \geq 0$.

Recordamos que la matriz de covarianzas muestral está dada por $S = X^T X / N$, y salvo escalar se sigue de la ecuación 3.1 que su descomposición por autovalores está dada por

$$X^T X = V D^2 V^T. \quad (3.2)$$

Los autovectores v_j (i.e. las columnas de V) son las direcciones de las *componentes principales* de X (Jolliffe 2005).

Dado un vector de pesos $w \in \mathbb{R}^p$, este define una nueva variable que es una combinación lineal de las p variables originales, cuyas observaciones están dadas por $z = Xw$ y su varianza muestral es

$$\text{var}[z] = z^T z / N = w^T X^T X w / N.$$

En particular, las variables $z_j = Xv_j$ se llaman las *componentes principales* de X . Se sigue de la factorización 3.2 que la primera componente principal $z_1 = Xv_1$ tiene la mayor varianza muestral entre todas las direcciones posibles (es decir entre w normalizados). Más aún, su varianza está dada por

$$\text{var}[z_1] = v_1^T V D^2 V^T v_1 / N = d_1^2 / N.$$

Se sigue de la ecuación 3.1 que la primera componente principal $z_1 = Xv_1$ también puede expresarse como $z_1 = u_1 d_1$. Por lo tanto u_1 es la primera componente principal normalizada. Análogamente, la segunda componente principal $z_2 = Xv_2$ es la variable con mayor varianza restringida a $\langle z_1 \rangle^\perp$, el complemento ortogonal de z_1 , y así sucesivamente.

Para ilustrar la solidez de este método aplicamos PCA a las tasas (*yields*) de los bonos del Tesoro de los Estados Unidos (*U.S. Treasuries*). Primero consideramos la matriz $Y \in \mathbb{R}^{218 \times 10}$ de *yields* para períodos mensuales desde enero 2005 hasta febrero 2023 y vencimientos de 3 meses a 30 años, obtenidas de *Federal Reserve Economic Data (FRED)*. Luego tomamos las diferencias mensuales en las *yields* y los estandarizamos, llamando X a la matriz resultante. Aplicamos PCA en X y graficamos los primeros tres autovectores obtenidos en la Figura 1. Notablemente, con solo 3 autovectores podemos explicar el 70 % de la varianza de las *yields*.

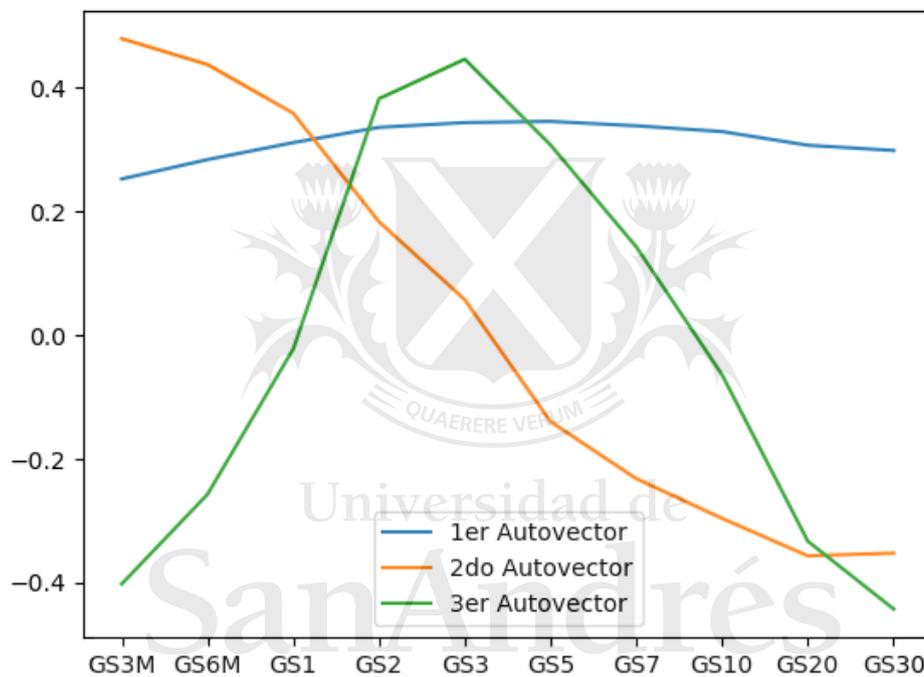


Figura 1: Autovectores de la descomposición PCA para los movimientos en *U.S. Treasury yields* para el período 2005-2023.

Se observa claramente que el primer autovector tiene casi el mismo peso para todos los vencimientos. Esto nos dice que lo que explica la mayor parte de la varianza en los cambios de tasas es un aumento/caída de tasas a lo largo de toda la curva (i.e. para todos los vencimientos). Esto se conoce como un desplazamiento en paralelo (*parallel shift*) de la curva de tasas. El segundo autovector tiene pesos fuertemente positivos (respectivamente negativos) en la parte corta (resp. en la parte larga) de la curva. Las tasas se mueven en esta dirección cuando aumentan las cortas y disminuyen las largas, o viceversa (salvo un signo) cuando disminuyen las cortas y aumen-

tan las largas. Estos movimientos se conocen como aplanamiento (*flattening*) y empinamiento (*steepening*) de la curva de tasas. Estos dos tipos de movimientos son por lejos los más comunes y estudiados en el análisis de la curva de tasas, y es interesante observar que aquí los hemos redescubierto de una forma puramente estadística. Más aún, con PCA confirmamos empíricamente que estos movimientos explican la mayor parte de la varianza en los movimientos de curva. De hecho, de los autovalores asociados sabemos que el desplazamiento paralelo explica aproximadamente el 40 % de la varianza y los aplanamientos/empinamientos un 20 % de esta. El tercer autovector nos muestra un aumento (resp. una caída) de tasas en los vencimientos intermedios (2 a 10 años) y una caída de las tasas cortas y largas, y explica cerca de un 10 % de la varianza.

Consideremos los retornos de p acciones en una ventana de tiempo de T días, esto define una matriz $R = (R_{ti}) \in \mathbb{R}^{T \times p}$, donde por ejemplo T puede ser 180 días y p podría ser 500 para el caso de los componentes del índice S&P 500. En realidad nos importan los retornos excedentes por sobre la tasa libre de riesgo, para la cual tomamos la *Fed Funds rate* o de un fondo *money market* y le restamos al retorno de la acción el retorno a un día en esta tasa. Las componentes principales de una matriz en general dependen de la escala de las columnas, así que primeramente estas se estandarizan $X_{ti} = (R_{ti} - \mu_i) / \sigma_i$ restándole la media y dividiendo por el desvío estándar muestral. Más aún dado que $T \ll p$, la matriz de covarianza muestral resulta singular en este caso, con lo cual se la regulariza sumándole la identidad por una constante pequeña

$$\tilde{S} = S + \gamma I,$$

e.g. $\gamma \approx 10^{-9}$. Esto vuelve a la matriz de covarianzas definida positiva, es numéricamente más estable para ser invertida y tiene los mismos autovectores que V modificando solamente los autovalores a los que se suma γ .

La componente principal $u_j = (1/d_j)Xv_j$ son los retornos excedentes estandarizados del llamado j -ésimo *eigenportfolio*. En otras palabras, hay un portafolio con pesos dados por $w_j = (w_{1j}, \dots, w_{pj})$ tal que u_j son sus retornos estandarizados. Esto se sigue de que u_j tiene varianza uno y media cero, luego:

$$u_j = (1/d_j)Xv_j = (1/d_j) \sum_i v_{ij} X_i = \sum_i \left(\frac{v_{ij}}{d_j \sigma_i} \right) (R_i - \mu_i) = \left(\sum_i w_{ij} R_i \right) - \mu(u_j). \quad (3.3)$$

Esto es, los pesos del j -ésimo *eigenportfolio* son $w_{ij} = v_{ij} / (d_j \sigma_i)$.

3.1.2. Regresión por componentes principales

Es muy común querer regresar observaciones de una variable $y \in \mathbb{R}^N$ por variables explicativas dadas por $X \in \mathbb{R}^{N \times p}$, aún cuando tenemos demasiadas variables (p) o muy pocas observaciones (N). En este caso, la regresión por cuadrados mínimos ordinarios (*Ordinary Least Squares*, *OLS*) puede resultar inestable (si $N < p$ no hay solución única) y suele tener un mal comportamiento fuera de muestra pues sobreajusta a los datos (fenómeno conocido como *overfitting*). En estos casos puede utilizarse PCA como un método de regularización y de reducción de dimensión donde nos quedamos con las primeras K componentes principales z_1, \dots, z_K de X para K menor que N . Luego se hace una regresión de y sobre estas variables que son ortogonales por construcción, con lo cual es una suma de regresiones univariadas

$$\hat{y} = \bar{y} \mathbf{1} + \sum_{j=1}^K \hat{\beta}_j z_j,$$

donde $\hat{\beta}_j = \langle y, z_j \rangle / \langle z_j, z_j \rangle$ y $\mathbf{1}$ denota al vector de unos N dimensional. El supuesto que subyace en esta regresión es que las componentes principales, las direcciones que explican la mayor varianza de las X , también explicarán en gran medida la varianza de y . Este método se conoce como regresión por componentes principales (*Principal Components Regression*, *PCR*).

3.1.3. El método de *Bootstrap*

El método de *bootstrap* es una técnica general para medir la precisión de un estadístico. Supongamos dadas observaciones $Z = (z_1, \dots, z_N)$ independientes idénticamente distribuidos (iid), donde z_i ($i = 1, \dots, n$) sigue la distribución F con F desconocida. Sea $S(Z)$ un estadístico, es decir una función de la muestra. La idea central del *bootstrap* es la de hacer muestreo con reposición de la muestra original Z , del mismo tamaño que la muestra original. Este procedimiento se repite B veces (por ejemplo $B = 100$ ó $B = 1000$), obteniendo muestras

$$Z^{(b)} = (z_1^{(b)}, \dots, z_N^{(b)}), \quad b = 1, \dots, B.$$

Luego de las muestras *bootstrap* se puede estimar cualquier aspecto de la distribución de $S(Z)$. Debemos calcular $S(Z^{(b)})$ para cada b y esto nos da una forma de simular una distribución aproximada de $S(Z)$. Por ejemplo, el estimador *bootstrap* de la varianza de $S(Z)$ es

$$\frac{1}{B-1} \sum_{b=1}^B (S(Z^{(b)}) - \bar{S})^2, \quad (3.4)$$

donde $\bar{S} = \sum_{b=1}^B S(Z^{(b)})/B$. Damos ahora una explicación más detallada de por qué el *bootstrap* funciona. La muestra original define una distribución \hat{F} que asigna probabilidad $1/N$ a cada observación z_i (si hay repeticiones como ocurre en modelos discretos, notamos que la probabilidad para un valor z observado k veces es k/N). La distribución \hat{F} se llama la distribución empírica de la muestra. Cuando estimamos la varianza siguiendo la ecuación 3.4 simplemente estamos haciendo Monte Carlo para generar B muestras de tamaño N que se distribuyen i.i.d. \hat{F} . Por la Ley de los grandes números

$$\frac{1}{B-1} \sum_{b=1}^B (S(Z^{(b)}) - \bar{S})^2 \rightarrow \text{Var}[S(Z^*)], \quad (3.5)$$

donde $Z^* = (z_1^*, \dots, z_N^*)$ es i.i.d. \hat{F} . Si pudieramos simular la distribución F sin costo, aún sin conocerla, entonces *bootstrap* no es necesario y sería absurdo utilizarlo ya que estamos reemplazando F por \hat{F} sin necesidad. Pero cuando lo único que tenemos es una muestra, *bootstrap* nos permite reutilizarla para estimar errores estándar de estimadores, intervalos de confianza, etc. Esto es crucial cuando no se conocen fórmulas cerradas para estos. La desventaja que tiene *bootstrap* es que es computacionalmente costoso ya que requiere re-estimar el estadístico $S(Z)$ B veces.

El *bootstrap* no solo funciona en el caso de muestras i.i.d., puede aplicarse para un modelo probabilístico general. Por ejemplo el caso en que $z_i = (x_i, y_i)$ y se estima un modelo funcional $Y = f(X) + \varepsilon$ donde ε es un error con media cero independiente de X . En general podemos pensar que hay un modelo probabilístico P que genera las observaciones $Z = (z_1, \dots, z_N)$. Con las observaciones lo primero que se hace es estimar el modelo probabilístico, obteniendo \hat{P} . En el caso particular del modelo funcional uno estima una función \hat{f} de acuerdo a algún criterio. Dado el modelo estimado, nos interesa conocer información de la distribución de por ejemplo un parámetro $\hat{\theta} = S(Z)$ estimado que determina (parcialmente) \hat{P} . Es importante aclarar que aquí debe estimarse el modelo completo P , y no hay una técnica general pero si soluciones ad-hoc para cada caso. El *bootstrap* procede en segunda instancia a obtener muestras a partir del modelo estimado \hat{P} . Este segundo paso puede hacerse de forma paramétrica o no paramétrica.

Supongamos que queremos aplicar *bootstrap* en el caso de un modelo de regresión lineal:

$$y_i = \beta^T x_i + \varepsilon_i,$$

donde los errores ε_i se asume que siguen una distribución F desconocida con $E(\varepsilon_i) = 0$ y los x_i son constantes (no aleatorios). Supongamos que β se estima por el método de cuadrados mínimos ordinarios, luego $\hat{\beta}$ está dado por

$$\hat{\beta} = (X^T X)^{-1} X^T y.$$

Para poder aplicar *bootstrap* especificar el modelo probabilístico total, por lo tanto debemos poder estimar F . Si bien no conocemos los errores, podemos calcular los errores estimados o residuos

$$\hat{\varepsilon}_i = y_i - \hat{\beta}^T x_i.$$

Luego, podemos aproximar la distribución F por la distribución empírica de los residuos, es decir \hat{F} es:

$$\mathbb{P}_{\hat{F}}(\varepsilon = \hat{\varepsilon}_i) = 1/N.$$

Luego tenemos una estimación del modelo completo $P = (\beta, F)$ dada por $\hat{P} = (\hat{\beta}, \hat{F})$. Luego para generar una muestra *bootstrap* tomamos

$$y_i^{(b)} = \hat{\beta}^T x_i + \hat{\varepsilon}_i^{(b)}, \quad i = 1, \dots, N, \quad (3.6)$$

donde $\hat{\varepsilon}_i^{(b)}$ se generan haciendo muestreo con reposición de los residuos, es decir con la distribución \hat{F} . Si nuestro interés es la distribución del estimador $\hat{\beta}$, podemos reestimarlos para cada muestra *bootstrap* por cuadrados mínimos

$$\hat{\beta}^{(b)} = (X^T X)^{-1} X^T y^{(b)}.$$

Notamos que esto puede hacerse de forma vectorizada para todas las muestras *bootstrap* a la vez concatenando los $y^{(b)}$ en una matriz con B columnas. Si queremos estimar la varianza de $\hat{\beta}_j$, la calculamos como en la ecuación 3.4 para los $\hat{\beta}_j^{(b)}$. Si suponemos que los errores siguen una distribución normal $\varepsilon_i \sim N(0, \sigma^2)$ entonces alternativamente podíamos estimar $\hat{\sigma}^2$ por

$$\hat{\sigma}^2 = \frac{1}{N-p} \sum_{i=1}^N (y_i - \hat{\beta}^T x_i)^2, \quad (3.7)$$

y luego obtener las muestras *bootstrap* siguiendo 3.6 pero simulando $\hat{\varepsilon}_i^{(b)} \sim N(0, \hat{\sigma}^2)$. Este es un método *bootstrap* completamente paramétrico, mientras que el anterior es semi-paramétrico (es solo paramétrico para obtener $\hat{\beta}$, pero no para \hat{F}). También podríamos haber aplicado el

método *bootstrap* básico donde directamente hacemos muestreos con reposición de la muestra (z_1, \dots, z_N) , este método no tiene ningún supuesto extra sobre los datos salvo que se generan de forma independiente idénticamente distribuida. La elección de que método *bootstrap* utilizar depende fundamentalmente de la confianza que tenemos en el modelo.

Notamos que por la definición de 3.6 se obtiene que la varianza de los $\hat{\beta}^*$ obtenidos por *bootstrap* es

$$\text{Var}[\hat{\beta}^*] = (X^T X)^{-1} \sigma_{\hat{F}}^2, \quad (3.8)$$

como para cualquier modelo lineal. Pero $\sigma_{\hat{F}}^2$ es simplemente

$$\sigma_{\hat{F}}^2 = \frac{1}{N} \sum_{i=1}^N \hat{\varepsilon}_i^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{\beta}^T x_i)^2,$$

que solo difiere de $\hat{\sigma}^2$ de 3.7 en una constante. Con *bootstrap* estimamos 3.8 por Monte Carlo (haciendo infinitas muestras, obtendríamos 3.8 exactamente), y como vemos esta fórmula es muy similar al estimador *plug-in* usual de la varianza de $\hat{\beta}$ dado por $(X^T X)^{-1} \hat{\sigma}^2$.

De hecho, se puede pensar a la distribución de los $\hat{\beta}^*$ obtenidos por *bootstrap* como un método aproximado de obtener una distribución a posteriori de β con un *prior* no informativo en el sentido Bayesiano (ver Hastie et al. 2009, Capítulo 8). La ventaja es que esta distribución se obtiene más fácilmente, sin tener que especificar formalmente un *prior* y sin tener que aplicar métodos costosos para obtener muestras a posteriori (i.e. métodos de *Markov Chain Monte Carlo*). Es por eso que la distribución *bootstrap* puede pensarse como una posteriori del pobre, obtenida a bajo costo.

3.2. Aprendizaje reforzado

3.2.1. Programación dinámica estocástica

Introducimos un problema de decisión multi-período estocástico general, ya que el objetivo del Aprendizaje reforzado es resolver de forma aproximada esta clase de problemas. Consideramos en primer lugar los problemas de horizonte finito ya que son matemáticamente menos complejos y además son utilizados en este trabajo. Sea N la cantidad de períodos o pasos del problema. En el período k -ésimo se tiene un estado x_k que sintetiza toda la información relevante para el problema en adelante. A su vez se tiene un control u_k , que debe pertenecer al conjunto

de “controles admisibles” $U_k(x_k)$. Por último, en cada período hay una variable aleatoria w_k cuya distribución puede depender de x_k y u_k (pero no de los estados o controles anteriores) que determina el siguiente estado de acuerdo a la ecuación de dinámica:

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, \dots, N - 1. \quad (3.9)$$

Por otra parte, se tiene una función de costo del período $g_k(x_k, u_k, w_k)$ y una función de costo final $g_N(x_N)$. El problema informalmente consiste en encontrar los controles óptimos en cada estado y período de forma tal de minimizar el costo total. Para formalizar esta idea debemos introducir el concepto de una política.

Definición 3.2.1. Una política μ es una tupla de funciones $(\mu_0, \dots, \mu_{N-1})$ que asignan a cada estado x_k un control $\mu_k(x_k) \in U_k(x_k)$ (i.e. satisface las restricciones de los controles), para cada período $k = 0, \dots, N - 1$.

Observación 3.2.2. Esta definición es necesaria en el caso estocástico ya que al aplicar un control no sabemos en qué estado vamos a estar en el siguiente período, por lo tanto tenemos que definir qué vamos a hacer en cada estado posible. En el caso determinístico es suficiente definir una secuencia de controles (u_0, \dots, u_{N-1}) y el problema consiste en encontrar un óptimo entre estas secuencias. Una política es un objeto más general que una secuencia de controles ya que se adaptan al estado en el que se encuentra el problema en cada período.

El problema de la Programación dinámica estocástica (*Stochastic Dynamic Programming*, *SDP*) consiste en:

$$\min_{\mu_0, \dots, \mu_{N-1}} \mathbb{E} \left[\sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) + g_N(x_N) \right] \quad (3.10)$$

$$\text{sujeto a } \mu_k(x_k) \in U_k(x_k), \quad x_{k+1} = f_k(x_k, \mu_k(x_k), w_k)$$

para un x_0 dado. Aquí mín se refiere al problema de minimización, aunque el mínimo puede o no alcanzarse, más formalmente deberíamos hablar del ínfimo del problema. Dada una política μ , se tiene una función de costo de la política J^μ dada por:

$$J^\mu(x_0) = \mathbb{E} \left[\sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) + g_N(x_N) \right]$$

para cada estado x_0 . Una política μ^* es óptima si resuelve el problema 3.10, es decir:

$$J^{\mu^*}(x_0) = J^*(x_0) = \min_{\mu \in \Pi} J^\mu(x_0)$$

para todo estado x_0 , donde Π denota al conjunto de todas las políticas y J^* denota a la función de costo óptimo del problema.

3.2.2. El algoritmo de programación dinámica

Esta clase de problemas puede resolverse en teoría por el algoritmo de programación dinámica (DP). Se define $J_N^*(x_N) = g_N(x_N)$ es decir como el costo final, y para cada $k = 0, \dots, N - 1$ sea

$$J_k^*(x_k) = \min_{u_k \in U_k(x_k)} \mathbb{E} [g_k(x_k, u_k, w_k) + J_{k+1}^*(f_k(x_k, u_k, w_k))] \quad (3.11)$$

entonces J_0^* es la función de costo óptimo del problema. Más aún la política $\mu_k^*(x_k)$ está dado por el control u_k que hace que se alcance el mínimo en la ecuación 3.11. La dificultad está en que, en la práctica, este problema de minimización estocástica puede ser muy difícil o costoso de resolver y hay que recurrir a aproximaciones. Aún en el caso de un espacio de estados y controles finitos, el algoritmo de Programación dinámica requiere una cantidad exponencial de operaciones en función de la cantidad de períodos y rápidamente se vuelve inmanejable en general. Salvo contados casos en los que la esperanza y la minimización tienen fórmulas cerradas analíticas, este método no es suficiente para atacar el problema.

Notamos que el algoritmo de DP hace una inducción hacia atrás (*backward induction*) partiendo desde la función de costo final y encontrando de forma recursiva las funciones de costo óptimo del problema empezando en cada paso. El algoritmo se basa en el principio de optimalidad que nos dice que la cola de una política óptima, es decir $(\mu_k^*, \dots, \mu_{N-1}^*)$, es óptima para el subproblema de cola partiendo desde período k . En cada paso que resolvemos la ecuación 3.11 podemos encontrar también la política óptima como:

$$\mu_k^*(x_k) = \arg \min_{u_k \in U_k(x_k)} \mathbb{E} [g_k(x_k, u_k, w_k) + J_{k+1}^*(f_k(x_k, u_k, w_k))] \quad (3.12)$$

Alternativamente se pueden calcular las funciones de costo $J_0^*, \dots, J_{N-1}^*, J_N^*$ anticipadamente, lo que suele llamarse *off-line* y luego a la hora de controlar calcular *on-line* la política de acuerdo a la ecuación 3.12.

3.2.3. El método de *Rollout*

Un método que permite mejorar una política (o más generalmente una heurística) es el algoritmo de *Rollout*. En caso de tener una política suficientemente buena, el algoritmo de *rollout* puede obtener una mejora sustancial y cercana a óptima. Supongamos dada una política $\mu = (\mu_0, \dots, \mu_{N-1})$ y denotemos $J_k^\mu(x_k)$ a la función de costo de la política desde el estado x_k , es decir

$$J_k^\mu(x_k) = \mathbb{E} \left[\sum_{s=k}^{N-1} g_s(x_s, \mu_s(x_s), w_s) + g_N(x_N) \right]. \quad (3.13)$$

Se define al *rollout* de la política μ como:

$$\tilde{\mu}_k(x_k) = \arg \min_{u_k \in U_k(x_k)} \mathbb{E} [g_k(x_k, u_k, w_k) + J_{k+1}^\mu(f_k(x_k, u_k, w_k))] \quad (3.14)$$

Es decir, el *rollout* de una política elige el control óptimo en este período asumiendo que se utilizará la política μ desde el siguiente período en adelante. La política μ solo se usa a la hora de calcular el costo del futuro, pero nunca es utilizada al aplicar el *rollout*. En el período siguiente ($k+1$) para calcular la política $\tilde{\mu}_{k+1}(x_{k+1})$ se vuelve a buscar el mínimo en la ecuación 3.14 con J_{k+2}^μ en lugar de J_{k+1}^μ .

Para aplicar el *rollout* en la práctica hay que lidiar con tres aproximaciones en simultáneo. Primero debemos conocer las funciones $J_k^\mu(x_k)$ de costo de la política μ y esto en general requiere de simulaciones de Monte Carlo para calcular la esperanza en 3.13. Luego está la esperanza en la ecuación 3.14 que también en general se calcula por Monte Carlo. Esto puede hacerse de forma conjunta al paso anterior siempre que el horizonte (N) del problema no sea demasiado largo, estimando directamente lo que se llama el Q -factor de la política μ definido formalmente por:

$$Q_k^\mu(x_k, u_k) = \mathbb{E} [g_k(x_k, u_k, w_k) + J_{k+1}^\mu(f_k(x_k, u_k, w_k))].$$

El Q -factor de una política nos dice cual es el costo de aplicar un control u_k en el estado x_k y en adelante aplicar la política μ . Para estimarlo podemos simplemente aplicar el control u_k en el paso k y de ahí en adelante seguimos aplicando la política μ en M simulaciones. Por esperanzas iteradas y la Ley de los grandes números se obtiene que

$$\frac{1}{M} \sum_{i=1}^M \left(g_k(x_k, u_k, w_k^{(i)}) + \sum_{s=k+1}^N g_s(x_s^{(i)}, \mu_s(x_s^{(i)}), w_s^{(i)}) + g_N(x_N^{(i)}) \right) \longrightarrow Q_k^\mu(x_k, u_k) \quad (3.15)$$

donde la flecha denota convergencia en probabilidad (o más aún, convergencia casi segura).

Una vez resuelta esta aproximación según la cual obtendremos Q -factores aproximados $\tilde{Q}_k^\mu(x_k, u_k)$, debemos resolver el problema de minimización de la ecuación 3.14. En el caso que los controles formen un conjunto finito esta minimización puede hacerse de forma exhaustiva, aunque a veces esto puede ser muy costoso. En tal caso métodos más avanzados como *Monte Carlo Tree Search (MCTS)* se utilizan para reducir el costo computacional de las simulaciones de Monte Carlo, explorando más profundamente los controles más promisorios a cuenta de los menos prometedores. En otros casos el espacio de controles puede ser continuo y entonces hay que aplicar métodos de Programación no-lineal (*Non-Linear programming*) como el método del gradiente, método de Newton, etc.

El algoritmo de *rollout* es muy útil ya que puede ser utilizado *on-line*, esto es, cuando estamos en el paso k -ésimo en el estado x_k , resolvemos de forma aproximada la ecuación 3.14 para elegir el control a utilizar. Una propiedad formal que tiene el *rollout* es la de *policy improvement*, es decir que mejora a la política base sobre la cual lo aplicamos, al menos cuando la minimización se hace de forma exacta.

Proposición 3.2.3. Sea $\mu = (\mu_0, \dots, \mu_{N-1})$ una política para el problema 3.10 y denotemos por $\tilde{\mu} = (\tilde{\mu}_0, \dots, \tilde{\mu}_{N-1})$ al *rollout* de esta política, entonces

$$J_k^{\tilde{\mu}}(x_k) \leq J_k^\mu(x_k), \quad k = 0, \dots, N.$$

Demostración. Ver apéndice A. □

3.2.4. Anticipación multipaso

El método de *rollout* se conoce también como anticipación de un paso (*one-step lookahead*), ya que los costos futuros se calculan usando la política base a partir del siguiente período. Una generalización importante es minimizar sobre ℓ pasos hacia adelante para recién en adelante usar la política base μ . Este método se conoce como anticipación multipaso (*multistep lookahead*). El problema a resolver es

$$\min_{u_k, \tilde{\mu}_{k+1}, \dots, \tilde{\mu}_{k+\ell-1}} \mathbb{E} \left[g_k(x_k, u_k, w_k) + \sum_{m=k+1}^{k+\ell-1} g(x_m, \tilde{\mu}_m(x_m), w_m) + J_{k+\ell}^\mu(x_{k+\ell}) \right]. \quad (3.16)$$

con x_k dado. Notamos que debido a que el problema es estocástico, en el período $k + 1$ en adelante debemos especificar políticas y no simplemente controles ya que no sabemos en que estado vamos a estar en el futuro. Esto aumenta la complejidad de resolver este problema fuertemente. Para simplificar esto se pueden utilizar heurísticas como Equivalente certero (*Certainty Equivalence, CE*) en el cual se reemplazan los w_s por sus valores medios para “apagar” la aleatoriedad del problema, ya que aquí los siguientes estados solo dependen de los controles y son finitos si estos lo son. En algunos casos cuando las w_s sean variables aleatorias discretas y tomen finitos valores y los controles también lo sean, es posible aplicar *multistep lookahead* sin especificar las políticas subsiguientes completamente, alcanza con especificarlas sobre los finitos estados posibles para x_s , con $s = k + 1, \dots, k + \ell - 1$. La complejidad del problema en general crece de forma exponencial con la cantidad ℓ de pasos en el futuro aún en los ejemplos estocásticos más simples, con lo cual ℓ suele ser pequeño. Al igual que en el *rollout*, a la hora de controlar el problema se aplica el control u_k que realiza el mínimo en 3.16, se descartan los controles $\tilde{\mu}_{k+1}, \dots, \tilde{\mu}_{k+\ell-1}$, y se pasa al estado x_{k+1} donde se vuelve a aplicar *multistep lookahead*.

Cuando no hay alternativa a encontrar las políticas $\tilde{\mu}_{k+1}, \dots, \tilde{\mu}_{k+\ell-1}$ para todo el espacio de estados en general esto pasa a ser un método *off-line*, ya que usualmente requiere aplicar métodos de regresión o clasificación para encontrar estas funciones. Por ejemplo, para hacer *2-step lookahead* una forma de resolver este problema de forma aproximada es la siguiente: Primero hacer M simulaciones de $x_{k+1}^{(i)}$ encontrando el control $u_{k+1}^{(i)}$ para cada una de ellas por *rollout*. Luego por métodos de clasificación se ajusta una función $\mu_{k+1}(x_{k+1}, r)$ de acuerdo a un modelo paramétrico (e.g. lineal *feature-based*, redes neuronales, etc.) según el parámetro r . Una vez obtenida $\tilde{\mu}_{k+1} = \mu_{k+1}(-, \tilde{r})$ se procede a aplicar rollout resolviendo el siguiente problema:

$$\tilde{\mu}_k(x_k) = \arg \min_{u_k} \mathbb{E} [g_k(x_k, u_k, w_k) + g(x_{k+1}, \tilde{\mu}_{k+1}(x_{k+1}), w_{k+1}) + J_{k+2}^\mu(x_{k+\ell})]. \quad (3.17)$$

3.2.5. El método del horizonte móvil

Hasta ahora hemos desarrollado la teoría bajo la hipótesis de un horizonte finito, esto es donde el número de períodos N es finito. Pero por otra parte existen muchos problemas cuya formulación es naturalmente de horizonte infinito, i.e. con una cantidad de períodos no acotadas. En muchos casos existe un estado terminal al cual una vez que se llega no hay costo y el problema termina, aunque no sabemos cuantos pasos necesitamos para llegar hasta este, como

por ejemplo en un juego de ajedrez. En otros casos no hay estados terminales y se suele descontar los costos para garantizar bajo ciertas condiciones la buena definición del problema. El problema de horizonte infinito suele formularse entonces como:

$$\min_{\mu} \mathbb{E} \left[\sum_{k=0}^{\infty} \alpha^k g(x_k, \mu(x_k), w_k) \right] \quad (3.18)$$

$$\text{sujeto a } \mu(x_k) \in U(x_k), x_{k+1} = f(x_k, \mu(x_k), w_k)$$

con x_0 dado y donde $\alpha \in (0, 1]$. Notamos que el costo y ecuación de la dinámica ya no dependen del tiempo, son estacionarias. Por último, si bien el problema a veces posee un horizonte finito N , este puede ser demasiado grande ($N \gg 0$) y el problema resulta demasiado complejo o costoso de abordar de forma directa.

Ahora bien, la teoría del horizonte finito puede aplicarse aún en todos estos casos mediante el enfoque del horizonte móvil, también conocido como aproximación de costo final. Simplemente se toma una cantidad de pasos H y se impone una función de costo final $g_H(x_{k+H})$ y para cada paso k se resuelve el problema truncado hasta H pasos en el futuro:

$$\min_{\mu_k, \dots, \mu_{k+H-1}} \mathbb{E} \left[\sum_{s=k}^{k+H-1} g_s(x_s, \mu_s(x_s), w_s) + g_H(x_{k+H}) \right] \quad (3.19)$$

$$\text{sujeto a } \mu_s(x_s) \in U_s(x_s), x_{s+1} = f_s(x_s, \mu_s(x_s), w_s)$$

con x_k dado. Una vez que uno encuentra el control (aproximadamente) óptimo $u_k^* = \mu_k^*(x_k)$, se descartan todas las políticas posteriores ($\mu_{k+1}^*, \dots, \mu_{k+H-1}^*$), se ejecuta este control, se obtiene el estado siguiente $x_{k+1} = f_k(x_k, u_k^*, w_k)$ y se vuelve a plantear el problema truncado en $k+1$ con el horizonte ahora H pasos adelante (i.e. en $k+1+H$). Es por esto que se conoce a este método como horizonte móvil, ya que en cada período se plantea un problema con un horizonte a H pasos en el futuro que se va moviendo en paralelo con el paso del tiempo.

Notamos que en el caso de horizonte infinito donde la dinámica y el costo son estacionarias, dado que usamos la misma función de costo final g_H entonces el enfoque de horizonte móvil produce una política estacionaria μ , es decir, la misma en todos los pasos. En el caso de horizonte finito con $N \gg 0$ tendremos el problema de 3.19 salvo para los últimos pasos cuando $N - k < H$ en cuyo caso se puede resolver el problema original con g_N ya que el horizonte ya está lo suficientemente cerca.

En general, el enfoque del horizonte móvil suele funcionar bien si la distribución de probabilidad del estado H pasos adelante es aproximadamente independiente del estado y control

actual, o alternativamente si esta se concentra alrededor de estados de costo bajo. En tales casos al truncar el problema no se lo altera demasiado, ya que en el primer caso el costo final será aproximadamente una constante fija sumada, mientras que en el segundo caso el costo final es un término insignificante.

3.2.6. Problemas de información parcial de los estados

Hasta ahora implícitamente asumimos que el estado del problema x_k es observable, dado que nuestro control es función del estado actual, i.e. $u_k = \mu_k(x_k)$. Ahora bien, en muchos casos este supuesto es demasiado fuerte y ya que existe incerteza sobre cual es el estado real subyacente al problema. Podemos plantear esto de la siguiente forma: en cada tiempo se observa y_k que nos da información de cual es el estado subyacente x_k , por ejemplo $y_k = x_k + \varepsilon_k$ puede ser una medición con error del estado x_k . Estos problemas son substancialmente más complejos en general y casi siempre se resuelven de forma subóptima o aproximada. Se conoce a este tipo de procesos como procesos de decisión de Markov parcialmente observables (*Partially observable Markov decision process, POMDP*).

Por otra parte, los problemas de información parcial pueden ser reformulados en un problema de información perfecta (i.e. donde observo el estado) reemplazando el estado del problema x_k con la distribución de probabilidad de que el estado sea x_k condicional a todas las observaciones

$$b_k(x_k) = \mathbb{P}(x_k | y_0, \dots, y_k) = \mathbb{P}(x_k | I_k)$$

donde I_k denota el conjunto de información hasta el período k -ésimo. Esta probabilidad puede ser calculada, al menos teóricamente, cuando el modelo probabilístico que relaciona las observaciones con los estados se encuentra especificado, y funciona como estado para un problema reformulado. Estos estados se conocen como *belief states*. Notamos que aún cuando el conjunto de estados x_k es finito, el conjunto de *belief states* es infinito, es el $(n - 1)$ -simplex $\Delta^{n-1} = \{(p_1, \dots, p_n) | \sum p_i = 1, p_i \geq 0\}$ donde n es el número de estados, ya que está formado por todas las distribuciones posibles sobre los estados de x_k .

Consideremos un caso particular que es más simple pero suficiente para nuestras necesidades. Este es el caso de un sistema con un parámetros desconocido pero constante. Este ejemplo se basa en la sección 6.7 del libro Bertsekas (2022a) o el artículo Bertsekas (2022b). Supon-

gamos que x_k es perfectamente observable pero se tiene un parámetro (multidimensional) θ desconocido del cual depende la dinámica del problema según

$$x_{k+1} = f_k(x_k, \theta, u_k, w_k). \quad (3.20)$$

Si bien no conocemos θ , tenemos una distribución “a priori” sobre este parámetro y podemos estimarlo en base a la información de los estados y controles

$$I_k = (x_0, \dots, x_k, u_0, \dots, u_{k-1})$$

Sea $b_k(\theta)$ la probabilidad de θ condicional a la información, i.e.

$$b_k(\theta) = \mathbb{P}(\theta | I_k) = \mathbb{P}(\theta | x_0, \dots, x_k, u_0, \dots, u_{k-1})$$

Entonces podemos encontrar $b_{k+1}(\theta)$ por Bayes:

$$b_{k+1}(\theta) = \mathbb{P}(\theta | x_0, \dots, x_{k+1}, u_0, \dots, u_k) \propto \mathbb{P}(x_{k+1} | x_k, u_k, \theta) b_k(\theta) \quad (3.21)$$

usando que la distribución de x_{k+1} solo depende de x_k, u_k cuando el parámetro θ está dado. Salvo una constante (que no depende de θ) esto nos permite determinar la distribución de $b_{k+1}(\theta)$ y nos define en conjunto con la dinámica de x_k una dinámica para (x_k, b_k) que es el *belief state* en este caso.

Supongamos que tenemos una política $\mu^\theta = (\mu_0^\theta, \dots, \mu_{N-1}^\theta)$ adaptada a cada valor de θ , que puede ser óptima o subóptima para el problema con θ conocido. El problema con θ conocido es en general más simple y puede atacarse con anterioridad, aunque debemos resolverlo para todo θ posible. Podemos entonces aplicar *rollout* para encontrar una política para el problema con θ desconocido de acuerdo a

$$\tilde{\mu}_k(x_k, b_k) = \arg \min_{u_k \in U_k(x_k)} \mathbb{E}_\theta \left[\mathbb{E}_{w_k} \left[g_k(x_k, \theta, u_k, w_k) + J_{k+1}^{\mu^\theta}(f_k(x_k, \theta, u_k, w_k)) | x_k, u_k, \theta \right] | I_k \right] \quad (3.22)$$

Notamos que la esperanza de adentro es respecto a w_k condicional a que el parámetro sea igual a θ , mientras que la esperanza de afuera integra a θ sobre todos los valores posibles, condicional a la información hasta el período k -ésimo, es decir de acuerdo a la distribución $b_k(\theta)$. Observamos que no es necesario conocer la distribución de θ para poder aplicar este método de *rollout*, alcanza con poder simular valores de θ que sigan (aproximadamente) su

distribución, esto en general es bastante más fácil en el contexto Bayesiano. Para estimar los Q -factores dentro de la ecuación 3.22 podemos simular M parámetros $\theta^{(i)}$, aplicar el control u_k en el primer paso y luego aplicar la política $\mu^{\theta^{(i)}}$ en adelante similar a como hicimos en 3.15, obtenemos un estimador consistente

$$\frac{1}{M} \sum_{i=1}^M \left(g_k(x_k, \theta^{(i)}, u_k, w_k^{(i)}) + \sum_{s=k+1}^N g_s(x_s^{(i)}, \theta^{(i)}, \mu_s^{\theta^{(i)}}(x_s^{(i)}), w_s^{(i)}) + g_N(x_N^{(i)}) \right) \quad (3.23)$$

$$\longrightarrow Q_k^\mu(x_k, b_k, u_k)$$

donde $Q_k^\mu(x_k, b_k, u_k)$ denota la esperanza dentro de la minimización en la ecuación 3.22.

3.3. Arbitraje estadístico

En esta subsección introducimos la metodología general de una estrategia de Arbitraje estadístico siguiendo a Avellaneda y Lee (2010) y Pelger et al. (2021) y explicamos en detalle la estrategia de Avellaneda y Lee (AL) ya que sirve como base de nuestro enfoque.

Siguiendo el marco conceptual de Pelger et al. (2021), el método del arbitraje estadístico consiste en: (1) Identificar activos similares para generar portafolios de arbitraje. (2) Extraer señales de desviación temporal de los activos similares. (3) Una política de asignación de posiciones en los portafolios de arbitraje. Vamos a tratar estas tres etapas o pasos por separado.

3.3.1. Portafolios de arbitraje

Consideremos un mercado compuesto por N activos operables. Para cada activo i se denota por $R_{i,t}$ el retorno del activo a tiempo t , incluyendo pagos de dividendos. Esto es,

$$R_{i,t} = \frac{S_{i,t} + D_{i,t} - S_{i,t-1}}{S_{i,t-1}},$$

donde $S_{i,t}$ es el precio del activo i a tiempo t , y $D_{i,t}$ el pago de dividendos en $(t - 1, t]$.

Se asume que existe un activo libre de riesgo operable, cuyo retorno a tiempo t es $R_{f,t}$. En la práctica se utiliza como tasa libre de riesgo la dada por la *Fed Funds Rate*. Alternativamente, se puede tomar la tasa de un *Treasury bill* a un mes, o la tasa de un *money-market fund*. El retorno excedente de un activo luego está dado por

$$R_i^e = R_i - R_f.$$

Se elige primero un modelo de factores $f = (f_1, \dots, f_K)^\top$ que capturan el riesgo sistemático de los activos. Luego se aplica este modelo para los retornos excedentes del activo i -ésimo,

$$R_i^e = \sum_{j=1}^K \beta_{ij} f_j + \varepsilon_i = \beta_i^\top f + \varepsilon_i, \quad (3.24)$$

donde $\mathbb{E}[f_j \varepsilon_i] = 0$, $\mathbb{E}[\varepsilon_i] = 0$. Notamos que permitimos que los β cambien con el tiempo y dependan de la información hasta el tiempo $t - 1$, es decir la ecuación 3.24 cambia en el tiempo, no se considera que los retornos sean estacionarios o i.i.d, lo que dificulta su estimación e implementación en la práctica pero no en teoría. La notación atemporal en esta ecuación se debe solo a una simplificación notacional.

Los modelos de factores utilizados pueden ser varios y usualmente provienen de modelos de *asset pricing*. Avellaneda y Lee (2010) utilizan factores construidos por PCA o también factores dados por ETF sectoriales. Pelger et al. (2021) utilizan factores basados en PCA, factores de Fama-French (ver Fama y French 1993; Fama y French 2015), y factores latentes condicionales estimados con análisis de componentes principales instrumental (IPCA) (ver Kelly et al. 2019).

Se asume que estos factores son retornos excedentes de activos o portafolios operables. Si llamamos $R_t^e = (R_{1,t}^e, \dots, R_{N,t}^e)^\top$ al vector de retornos excedentes de los N activos considerados en nuestro mercado, entonces el hecho de que los factores sean operables quiere decir que existe una matriz $w_{t-1}^F \in \mathbb{R}^{N \times K}$ tal que

$$f_{j,t} = \sum_{i=1}^N w_{ij}^F R_{i,t}^e \quad (3.25)$$

$$f_t = (f_{1,t}, \dots, f_{K,t})^\top = (w_{t-1}^F)^\top R_t^e.$$

Es decir, la columna k -ésima de w_{t-1}^F nos dice cuáles son los pesos a tiempo t de los activos en el k -ésimo factor. Por ejemplo, en el caso de factores dados por PCA, los factores son los *eigenportfolios* y los pesos son los que calculamos en la ecuación 3.3. En caso de que algunos de los factores no sean operables siempre se los puede reemplazar por sus portafolios imitadores (*factor-mimicking portfolios*, ver Cochrane 2009, sección 6.3) proyectándolos ortogonalmente en el espacio de retornos excedentes. Luego en teoría este supuesto es innecesario.

Se define luego el portafolio de arbitraje i -ésimo como el residuo de la regresión en la ecuación 3.24,

$$\varepsilon_{i,t} = R_{i,t}^e - \sum_{j=1}^K \beta_{ij,t-1} f_{j,t} = R_{i,t}^e - \beta_{i,t-1}^\top f_t = R_i^e - \beta_{i,t-1}^\top (w_{t-1}^F)^\top R_t^e. \quad (3.26)$$

Luego los residuos son también retornos excedentes de un portafolio operable donde se está *long* un dólar del activo i -ésimo y *short* β_{ij} dólares en el factor j -ésimo para $j = 1, \dots, K$. Se suele llamar portafolio replicador del activo i -ésimo al dado por los pesos $(w_{t-1}^F) \beta_{i,t-1} \in \mathbb{R}^N$. Podemos dar luego una ecuación lineal para todos los portafolios de arbitraje de forma conjunta en términos de R_t^e ,

$$\varepsilon_t = R_t^e - \beta_{t-1} (w_{t-1}^F)^\top R_t^e = \Phi_{t-1} R_t^e, \quad (3.27)$$

donde $\Phi_{t-1} = \mathbb{1}_N - \beta_{t-1} (w_{t-1}^F)^\top \in \mathbb{R}^{N \times N}$. La matriz Φ_{t-1} es la proyección ortogonal sobre el complemento ortogonal del espacio de los factores.

3.3.2. Neutralidad a los factores y al mercado

Supongamos dado el modelo de factores como en la ecuación 3.24. Dado un portafolio Π definido por las posiciones netas $(Q_i)_{i=1}^N$ en dólares cada activo, decimos que el portafolio es neutral al factor j -ésimo si la sensibilidad del portafolio (i.e. el β) respecto a este factor es cero, es decir

$$\beta_j^\Pi = \sum_{i=1}^N \beta_{ij} Q_i = 0.$$

Decimos que el portafolio es neutral al mercado si es neutral a todos los factores. Implícitamente, estamos asumiendo que el modelo de factores da el precio de los activos, es decir el factor de descuento estocástico (*stochastic discount factor, SDF*) se encuentra en el espacio de los factores, i.e. $m = a + b^\top f$ o al menos se tiene esto de forma aproximada. Si un portafolio es neutral al mercado entonces sus retornos excedentes se deben exclusivamente a la exposición al riesgo idiosincrático de los activos pues,

$$\begin{aligned} R_\Pi^e &= \sum_{i=1}^N Q_i R_i^e = \sum_{i=1}^N Q_i \left[\sum_{j=1}^K \beta_{ij} f_j + \varepsilon_i \right] = \\ &= \sum_{j=1}^K \left[\sum_{i=1}^N \beta_{ij} Q_i \right] f_j + \sum_{i=1}^N Q_i \varepsilon_i = \sum_{i=1}^N Q_i \varepsilon_i. \end{aligned}$$

Los portafolios de arbitraje se construyen de forma tal sean neutrales al mercado, es decir a todos los factores del modelo. Es decir, supongamos que se tiene una exposición *long* al portafolio

de arbitraje

$$\varepsilon_i = R_i^e - \beta_i^\top f = R_i^e - \sum_{j=1}^K \beta_{ij} f_j.$$

Más precisamente, se compra 1 dólar *long* el activo i -ésimo y β_{ij} dólares *short* el factor f_j (que en sí es un portafolio operable por 3.25). Entonces el β_j de este portafolio es cero para todo j ya que los residuos son ortogonales a los factores por definición de la proyección ortogonal, o por linealidad de los β y usando que el β_j de f_j es uno.

Los portafolios de arbitraje son entonces proyecciones sobre el espacio ortogonal a los factores, y por lo tanto para un modelo de *asset pricing* exacto, su retorno esperado debería ser cero, es decir $\mathbb{E}[\varepsilon_{i,t}] = 0$. Notamos que para un modelo de *asset pricing* exacto debe valer que $\mathbb{E}_{t-1}[\varepsilon_{i,t}] = 0$ para todo t lo que descarta la existencia de arbitrajes, pero esto en general es difícil de cumplir para modelos empíricos ya que no conocemos el modelo de los retornos reales y este probablemente puede ir cambiando en el tiempo lo que dificulta su especificación. Sin embargo, cuando el modelo es aproximado, es posible tener desviaciones temporales de los residuos en el corto plazo con reversión a la media en el mediano o largo plazo. También, los residuos deben estar débilmente correlacionados, ya que si no habría algún factor sistemático de riesgo no observado entre ellos.

Estas son las dos heurísticas sobre las que se basa el Arbitraje estadístico: encontrar residuos cuyo retorno esperado condicional a la información a tiempo $t - 1$ sea distinto a cero, es decir $\mathbb{E}_{t-1}[\varepsilon_{i,t}] \neq 0$, y abrir posiciones sobre un gran número de residuos débilmente correlacionados para obtener un retorno positivo con alta probabilidad.

3.3.3. Señales de arbitraje

El segundo paso consiste en extraer señales de arbitraje a partir de la serie de tiempo de los retornos de portafolios de arbitraje. Las señales de arbitraje en general se basan en modelos de series de tiempo paramétricos o no paramétricos. Se considera como *input* de las señales a los últimos L retornos,

$$\varepsilon_{i,t-1}^L = (\varepsilon_{i,t-L}, \dots, \varepsilon_{i,t-1})$$

La señal de arbitraje es una función $\theta : \mathbb{R}^L \rightarrow \mathbb{R}^p$ que asigna una señal $\theta(\varepsilon_{i,t-1}^L) = \theta_{i,t-1} \in \mathbb{R}^p$ que retiene toda la información relevante de la secuencia *input*. Es decir, estamos bajo el supuesto implícito que la señal de arbitraje es un estadístico suficiente para extraer una estrategia.

También estamos asumiendo que la serie de tiempo sigue una distribución estacionaria condicional a los últimos L retornos. Por último, se utiliza el mismo modelo de señal para todos los portafolios de arbitraje (i.e. para todos los activos), luego bajo la misma señal la estrategia debe ser igual. Notamos por θ_{t-1} a la señal aplicada a todos los portafolios de arbitraje, es decir $\theta_{t-1} = (\theta_{1,t-1}, \dots, \theta_{N,t-1}) \in \mathbb{R}^{N \times p}$. Podemos pensar a la señal de arbitraje como todas las características (*features*) relevantes de la serie de tiempo que serán utilizados en el siguiente paso para definir nuestra asignación de posiciones. Es decir, dada una serie de L lags x ,

$$\theta(x) = (\theta_1(x), \dots, \theta_p(x)),$$

donde $\theta_\ell(x)$ me define el *feature* ℓ -ésimo de la señal. Por ejemplo, en Pelger et al. (2021) se utiliza un filtro lineal invariante en el tiempo,

$$\theta_\ell = \sum_{j=1}^L W_j x_j,$$

es decir $\theta(x) = Wx$. En ese artículo utilizan como W a la transformada de Fourier discreta primeramente, así como luego filtros dados por una CNN. Alternativamente, se podría usar como filtro a una transformada de *wavelets*.

Explicamos ahora la señal que utilizan Avellaneda y Lee (2010) basados en un modelo de tiempo continuo de Ornstein-Uhlenbeck para los residuos acumulados. Se consideran los últimos L residuos acumulados (o integrados de forma discreta):

$$X_k = \sum_{l=1}^k \varepsilon_{i,t-L-1+l}, \quad k = 1, \dots, L. \quad (3.28)$$

Se modelan a los residuos acumulados como observaciones discretas de un proceso OU. Esto es $X(t)$ satisface a la siguiente ecuación diferencial estocástica (*Stochastic Differential Equation, SDE*, ver Shreve 2004, sección 6.2):

$$dX(t) = \kappa(\mu - X(t))dt + \sigma dW(t). \quad (3.29)$$

para $\kappa > 0$ y $W(t)$ un Movimiento Browniano. La siguiente proposición nos da una fórmula para la solución de esta SDE, y en particular determina la distribución del proceso a cada tiempo.

Proposición 3.3.1. Sea $X(t)$ un proceso de Ornstein-Uhlenbeck como en la ecuación 3.29, entonces su solución está dada por:

$$X(t) = \mu + e^{-\kappa t}(X(0) - \mu) + \sigma e^{-\kappa t} \int_0^t e^{\kappa s} dW(s) \quad (3.30)$$

En particular, $X(t)$ es normal con varianza $\frac{\sigma^2(1-e^{-2\kappa t})}{2\kappa}$.

Demostración. Ver apéndice A. □

Notamos que si en la ecuación 3.30 se hace tender t a infinito entonces $X(t)$ es normal con media μ y varianza $\sigma_{eq}^2 = \sigma^2/2\kappa$. Esta varianza se llama la varianza de equilibrio o de largo plazo. En el largo plazo, si $X(t)$ se encuentra muy lejos de su media μ sabemos que con alta probabilidad volverá hacia esta.

Los parámetros de modelo se estiman discretizando la ecuación con un modelo de series de tiempo $AR(1)$. Luego la señal de arbitraje de Avellaneda y Lee se define por:

$$\theta^{AL}(\varepsilon_{i,t-1}^L) = (\hat{\mu}, \hat{\kappa}, \hat{\sigma}^2, X_L, R^2), \quad (3.31)$$

donde agregamos el R^2 siguiendo a Yeo y Papanicolaou (2017) para filtrar las señales de acuerdo a esta medida de bondad de ajuste.

3.3.4. Políticas de asignación de posiciones

La política de asignación define los pesos w_i (*weights*) de los portafolios de arbitraje basado en las señales que se extraen de ellos. En general está dada por una función $w^\varepsilon : \mathbb{R}^p \rightarrow \mathbb{R}$ donde $w^\varepsilon(\theta_{i,t-1})$ es el peso asignado al portafolio de arbitraje i -ésimo. En Pelger et al. (2021) primero se asigna una familia de funciones $w^\varepsilon \in W$, y luego se elige w^ε que maximiza el *Sharpe ratio* o maximiza el retorno esperado (condicional a la información a tiempo $t - 1$) para un nivel de varianza dado, o más generalmente una función de utilidad cóncava. Notamos que aquí la política solo considera el retorno a un día, y se tienen en cuenta los costos de transacción respecto de la asignación del día anterior. La familia de funciones de asignación W en general está dada de forma paramétrica, es decir se tienen $w^\varepsilon(\theta_{i,t-1}, \phi)$ donde ϕ es el parámetro sobre el que luego se optimiza para elegir los pesos. Denotamos por $w_{t-1}^\varepsilon \in \mathbb{R}^N$ a la asignación de pesos para todos los portafolios de arbitraje en forma conjunta lugar a lugar. Asociada a esta tenemos una posición en los activos $w_{t-1} = (w_{1,t-1}, \dots, w_{N,t-1}) \in \mathbb{R}^N$, que está dada por $\Phi_{t-1}^\top w_{t-1}^\varepsilon$. Para que la maximización tenga sentido en general se suele imponer alguna restricción de apalancamiento (*leverage*). Pelger et al. (2021) deciden normalizar los pesos de los activos en la norma 1, es decir

$$w_{t-1}^\top = \frac{(w_{t-1}^\varepsilon)^\top \Phi_{t-1}}{\|(w_{t-1}^\varepsilon)^\top \Phi_{t-1}\|_1}. \quad (3.32)$$

En Avellaneda y Lee (2010) la función que asigna los pesos es más simple y está basada umbrales o límites de acuerdo a los cuáles se abren o cierran posiciones. Se define el *score* como:

$$s = \frac{X_L - \hat{\mu}}{\hat{\sigma}/\sqrt{2\hat{\kappa}}}$$

El *score* mide la distancia del residuo acumulado al equilibrio (la media) medido en desvíos estándar de la misma, ya que $\hat{\sigma}/\sqrt{2\hat{\kappa}}$ es un estimador del desvío estándar de equilibrio o largo plazo.

La estrategia de Avellaneda y Lee (2010) consiste en comprar (resp. *sell short*) el portafolio de arbitraje *i*-ésimo cuando el *score* es bastante negativo (resp. positivo) y cerrar la posición cuando se revierta cercano al equilibrio. En más detalle, la estrategia es la siguiente:

abrir una posición *long* si $s < -s_{ol}$ → cerrar una posición *long* si $s > -s_{cl}$
 abrir una posición *short* si $s > s_{os}$ → cerrar una posición *short* si $s < s_{cs}$

Cuadro 1: Estrategia de Avellaneda y Lee (2010)

donde los valores de corte los determinan de forma empírica fuera de la muestra, y son los mismos para todos los activos. Estos se toman en el artículo como $s_{ol} = s_{os} = 1,25$, $s_{cs} = 0,75$ y $s_{cl} = 0,5$. Más aún, se impone una condición de la forma $\kappa > \kappa_{crit}$ para restringir solo a portafolios de arbitraje cuya reversión a la media sea lo suficientemente rápida. Por último siguiendo a Yeo y Papanicolaou (2017) se impone una restricción de la forma $R^2 > R_{crit}^2$, es decir si no se cumple esto el peso es cero.

Falta distribuir el capital disponible entre todos los activos que se desean operar, en este caso las 500 acciones que componen al índice S&P 500. En el artículo se distribuye con igual peso entre todos los activos disponibles y solo se ajusta el apalancamiento (*leverage*) que se desea para el portafolio.

3.3.5. Apalancamiento, costos de transacción y P&L

Supongamos que se tiene un capital (*equity*) E_{t-1} que en principio está invertido en un *money-market fund* y por lo tanto su retorno en el período t es $R_{f,t}$ que es conocido a tiempo $t - 1$. Como en la subsección 3.3.2, se denota por $(Q_{i,t-1})_{i=1}^N$ a las posiciones netas en dólares cada activo antes del período t . Estas posiciones se mantendrán durante el período t , luego del

período en cada activo obtenemos

$$Q_{i,t-1}^+ = Q_{i,t-1} + Q_{i,t-1}(R_{i,t} - R_{f,t}) = Q_{i,t-1}(1 + R_{i,t} - R_{f,t}),$$

dólares en el activo i . $Q_{i,t-1}^+$ es lo que llamamos la posición *post-trade* en el activo i . Las cantidades $Q_{i,t}$ y $Q_{i,t-1}^+$ en general difieren, ya que luego del período t se hace un cambio de las posiciones del portafolio, la diferencia entre estas dos cantidades es lo que determina los costos de transacción.

Supongamos que hay un costo de transacción fijo $\xi > 0$ que incluye las comisiones del *broker*, *spreads* entre compra y venta, más el posible impacto del portafolio en el mercado que aquí no se modela. Por otra parte, se asume que hay un *spread* de $\delta > 0$ dado por comisiones del *broker* para posiciones apalancadas. Cuando se vende en corto un activo riesgoso y se invierte el dinero en el activo libre de riesgo, en realidad el *broker* nos paga $R_{f,t} - \delta$. Similarmente cuando se quiere comprar un activo de forma apalancada el *broker* nos pedirá una tasa de $R_{f,t} + \delta$ (i.e. la *margin rate*). Este δ se conoce como costo de tenencia (*holding cost*). Asumiendo que siempre mantenemos la exposición neta menor que el *equity* podemos suponer que el costo de las posiciones *long* es cero dado que podemos directamente adquirir el activo con el capital disponible. Más precisamente, se definen la exposición *long* y *short* respectivamente por

$$L_{t-1} = \sum_{i=1}^N \max(Q_{i,t-1}, 0), \quad S_{t-1} = \sum_{i=1}^N \min(Q_{i,t-1}, 0).$$

Se define la exposición neta del portafolio como $L_{t-1} + S_{t-1}$, y la exposición bruta como

$$L_{t-1} + |S_{t-1}| = \sum_{i=1}^N |Q_{i,t-1}| = \|Q_{t-1}\|_1.$$

Estamos bajo el supuesto de que la exposición neta es menor que el *equity*, es decir

$$\sum_{i=1}^N Q_{i,t-1} = L_{t-1} + S_{t-1} \leq E_{t-1}, \quad (3.33)$$

o alternativamente $L_{t-1} \leq E_{t-1} + |S_{t-1}|$, lo que nos dice que siempre tenemos capital (nuestro *equity* original más el que proviene de las posiciones *short*) suficiente para adquirir nuestra exposición *long* sin pedir prestado al *broker*.

En la práctica se toman $\xi = 0,0005$ y $\delta = 0,0001$, es decir 5 bps de transacción y 1 bps diario para posiciones *short*, consistente con Pelger et al. (2021) y Boyd et al. (2017). Notamos que

en Avellaneda y Lee (2010) y Yeo y Papanicolaou (2017) se simplifica el problema tomando $\delta = 0$. Luego, la ecuación de beneficios y pérdidas del portafolio (*profit and loss, P&L*) en cada período está dada por

$$E_t = (1 + R_{f,t})E_{t-1} + \sum_{i=1}^N Q_{i,t-1}(R_{i,t} - R_{f,t}) - \xi \sum_{i=1}^N |Q_{i,t-1} - Q_{i,t-2}^+| + \delta \sum_{i=1}^N \min(Q_{i,t-1}, 0) = (1 + R_{f,t})E_{t-1} + Q_{t-1}^\top R_t^e - \xi \|Q_{t-1} - Q_{t-2}^+\|_1 - \delta \|\min(Q_{t-1}, 0)\|_1. \quad (3.34)$$

El primer término en la ecuación 3.34 se debe a que el *equity* nos devenga la tasa libre de riesgo (a un día o el período que se considere), el segundo término es el retorno por las posiciones en activos, el tercer término son los costos de transacción y el último se debe a los costos de tenencia en posiciones short.

El apalancamiento (*leverage*) del portafolio se define como el cociente de la exposición bruta sobre el *equity*, es decir

$$\Lambda_{t-1} = \frac{\sum_{i=1}^N |Q_{i,t-1}|}{E_{t-1}}. \quad (3.35)$$

Existen regulaciones sobre el apalancamiento, por ejemplo la Regulación T de la Reserva Federal establece que este no puede ser mayor a 2 para ciertos fondos, así como también límites impuestos por el *broker*.

Se evalúa la estrategia de dos formas, primero como un fondo *long-short* neutral al mercado para un nivel de *leverage* dado, por ejemplo Pelger et al. (2021) usan $\Lambda = 1$ de acuerdo a la ecuación 3.32. Por otra parte Avellaneda y Lee (2010) evalúan la estrategia con un *leverage* de $\Lambda = 2 + 2 = 4$, es decir 2 veces el capital en las posiciones *long* y *short* respectivamente. Otra alternativa es un lo que se llama un fondo 130 – 30, muy usual dentro de los fondos administrados. Más precisamente se tiene 30% del capital en posiciones *short* y 130% *long*, es decir $L = 1,3E$ y $|S| = 0,3E$. Esto puede ser simplemente invertir el capital en un índice (e.g. S&P 500) y usar este *equity* como colateral de un portafolio *long-short* neutral al mercado.

Los pesos w_i (*weights*) se relacionan con las posiciones según

$$w_{i,t-1} = \frac{Q_{i,t-1}}{E_{t-1}} \quad \text{ó} \quad Q_{i,t-1} = w_{i,t-1}E_{t-1}.$$

Notamos que los *weights* en general no suman uno, esto ocurre para un portafolio *long-only* sin apalancamiento, aunque por nuestro supuesto en 3.33 la suma de los *weights* está acotada por

uno. Observamos que el *leverage* definido en 3.35 se puede calcular como la norma uno de los *weights*, es decir

$$\Lambda_{t-1} = \sum_{i=1}^N |w_{i,t-1}| = \|w_{t-1}\|_1.$$

Por último notamos que para relacionar posiciones (resp. pesos) en los portafolios de arbitraje con posiciones (resp. pesos) en los activos, estos se relacionan a través de la matriz Φ_{t-1} definida en 3.27. En detalle, si damos posiciones $Q_{t-1}^\varepsilon \in \mathbb{R}^N$ para los portafolios de arbitraje, entonces las posiciones en los activos está dada por

$$Q_{t-1} = \Phi_{t-1}^\top Q_{t-1}^\varepsilon.$$

Esto puede verse por

$$Q_{t-1}^\top R_t^e = (Q_{t-1}^\varepsilon)^\top \Phi_{t-1} R_t^e = (Q_{t-1}^\varepsilon)^\top \varepsilon_t,$$

donde usamos la ecuación 3.27 en la última igualdad. Luego, el capital se puede asignar en los portafolios de arbitraje por

$$Q_{i,t-1}^\varepsilon = s_i \frac{\Lambda_{t-1} E_{t-1}}{O_{t-1} \|\Phi_{i,t-1}\|_1}, \quad (3.36)$$

donde $u_i \in \{-1, 0, 1\}$ es la señal dada por el *score* en el Cuadro 1, y $O_{t-1} \leq N$ es el número de posiciones abiertas (*long* o *short*) a tiempo $t-1$ y $\|\Phi_{i,t-1}\|_1$ es la norma uno de la fila i -ésima de Φ_{t-1} . De esta forma se obtiene una estrategia con apalancamiento menor o igual Λ_{t-1} ya que se tiene la siguiente cota,

$$\|Q_{t-1}\|_1 = \|\Phi_{t-1}^\top Q_{t-1}^\varepsilon\|_1 \leq \|\Phi_{t-1}^\top\|_1 \|Q_{t-1}^\varepsilon\|_1 \leq \left(\sum_{i=1}^N |s_i| \right) \frac{\Lambda_{t-1} E_{t-1}}{O_{t-1}} = \Lambda_{t-1} E_{t-1},$$

donde usamos que la norma $\|\Phi_{t-1}^\top\|_1$ es el máximo de las norma uno de las filas de Φ_{t-1} .

Para evaluar las estrategias se utiliza el *Sharpe ratio* ex post anualizado, que se define como el cociente del retorno excedente medio anualizado sobre su volatilidad (i.e. su desvío estándar). Específicamente, dado el *equity* E_t por la ecuación 3.34 entonces,

$$R_{\Pi,t} = \frac{E_t - E_{t-1}}{E_{t-1}},$$

$$\hat{\mu} = \frac{252}{T} \sum_{t=1}^T (R_{\Pi,t} - R_{f,t}),$$

$$\hat{\sigma} = \sqrt{\frac{252}{T-1} \sum_{t=1}^T ((R_{\Pi,t} - R_{f,t}) - \hat{\mu}/252)^2},$$

$$SR = \frac{\hat{\mu}}{\hat{\sigma}}.$$

4. Datos

El objetivo propuesto es evaluar estas estrategias en el mercado de *U.S. equities large-cap*, dado por las componentes del índice S&P 500. Este índice se compone de aproximadamente 500 acciones de compañías listadas en Estados Unidos que poseen el mayor nivel de capitalización. En abril de 2022 el capital necesario para ingresar a esta lista era de \$14.6B (*billions* o miles de millones de dólares).

Ahora bien, el listado de componentes es dinámico y se actualiza periódicamente de acuerdo a las compañías que salen del índice pues valor de mercado cae, son disueltas o adquiridas por otra empresa (e.g. Under Armour, First Republic Bank, SVB Bank, Twitter, etc.) y las empresas cuyo valor sube (e.g. Tesla, Moderna, T-Mobile US, etc.) y pasan a formar parte del índice.

Dado que este listado dinámico no se conoce a ciencia cierta ya que es propiedad de *S&P 500 Dow Jones Indices*, se utilizan ciertos supuestos simplificadores. En realidad, nuestra principal motivación para usar *large-cap equities* se debe a que siendo las más importantes, suelen ser también las que poseen mercados con mayor profundidad y operan altos volúmenes diarios. Se evalúan las estrategias con las acciones que actualmente forman parte del índice S&P 500 y que tengan durante los últimos 10 años al menos un valor de mercado suficiente para ser *mid-cap*. Esto es para evitar que se opere en acciones *small-cap* donde nuestro modelo y sus supuestos pueden dejar de ser realistas por la falta de liquidez en estas acciones individuales. Por otra parte, como las estrategias son neutrales al mercado, el impacto que pueden tener los sesgos generados (e.g. *survivorship bias*) por estas simplificaciones debe ser marginal. Esto nos da un mercado de $N = 451$ acciones sobre las cuáles evaluaremos el rendimiento de las estrategias por *backtesting* en los últimos 10 años. Se obtienen los datos de precio de cierre ajustado por dividendos y *stock splits* de estos activos a través de Yahoo! Finance. Para la tasa libre de riesgo se utiliza la *Fed Funds Effective Rate* que se obtiene de FRED. Si bien no es una tasa operable en forma directa, puede accederse a un retorno similar a través de fondos *money market*.

5. Metodología

5.1. Estimación de los parámetros de los modelos

Dado que las estrategias que se introducen siguen la formulación del arbitraje estadístico introducido en la subsección 3.3, debemos especificar cómo se estiman estos modelos en la práctica.

5.1.1. Estimación del modelo de factores

El modelo lineal de la ecuación 3.24 se estima por ejemplo por OLS obteniendo $\hat{\beta}_{t-1} \in \mathbb{R}^{N \times K}$ y por lo tanto residuos $\hat{\varepsilon}_i$ para cada activo. En nuestra implementación usamos una ventana móvil de $T = 252$ retornos pasados para obtener los factores de PCA, es decir aproximadamente un año. Para estimar β_{t-1} se utiliza una ventana de $L = 60$ días. Esta configuración coincide con la utilizada en Avellaneda y Lee (2010) y Pelger et al. (2021). Se usan ventanas móviles de estimación relativamente cortas porque los retornos de un pasado distante no son siempre económicamente relevantes y no reflejan el comportamiento presente de los activos. Es decir, se cree que el modelo no es estacionario, y para resolver esto se aplica esta heurística simple pero muy utilizada en la práctica.

Alternativamente, se pueden aplicar métodos de regresión adaptativos (e.g. *Flexible Least Squares*, *Exponentially Weighted Recursive Least Squares*, etc., ver Tsagaris 2010; Montana et al. 2009) que permiten que los $\hat{\beta}_t$ se vayan ajustando con el tiempo a los datos más recientes.

Por otra parte, cuando algunos de los factores no son operables y se usan *factor-mimicking portfolios* entonces también se debe estimar la proyección ortogonal de estos en el espacio de retornos, esto en general requiere de una regresión con regularización pues usualmente se tiene que $N \gg T$.

En la implementación de este trabajo utilizamos factores dados por PCA los cuales son utilizados en una regresión por OLS. Esto es equivalente a aplicar PCR como explicamos en la subsección 3.1.2.

Al estimar los factores por PCA y realizar una regresión por OLS para calcular los $\hat{\beta}_t$, encontramos que los portafolios de arbitraje generan demasiados costos de transacción por rebalanceo al tener posiciones abiertas. Esto se debe a que las matrices Φ_t y Φ_{t-1} difieran de forma signifi-

ficativa, ya que la fila i -ésima determina el costo de transacción al tener una posición abierta en el portafolio de arbitraje i -ésimo. Ahora bien,

$$\Phi_t = \mathbb{1}_N - \hat{\beta}_t(w_t^F)^\top,$$

donde w_t^F son los pesos obtenidos por PCA. Por lo tanto, este costo puede ser grande debido a cambios en los $\hat{\beta}_t$ estimados en cada tiempo, o debido a cambios en los pesos de PCA w_t^F . Más específicamente, el costo de transacción base del portafolio i -ésimo está dado por

$$\xi \|\Phi_{i,t} - \Phi_{i,t-1}\|_1 = \xi \|\hat{\beta}_{i,t}^\top(w_t^F)^\top - \hat{\beta}_{i,t-1}^\top(w_{t-1}^F)^\top\|_1,$$

donde $\Phi_{i,t}$ (resp. $\hat{\beta}_{i,t}$) denota la fila i -ésima de la matriz en cuestión.

En una primera instancia consideramos que la causa de los costos se debía principalmente a los pesos w_t^F de PCA y por lo tanto pensamos en alternativas a estos. Una alternativa posible es utilizar *Sparse Principal Component Analysis (SPCA)* de Zou et al. (2006) en lugar de PCA. SPCA encuentra aproximaciones \tilde{w}_t^F ralas (*sparse*, es decir con muchos ceros) de los pesos w_t^F mediante una modificación del método *lasso* (o ℓ_1) adaptado al problema de PCA. La idea es que estos nuevos factores descarten los pesos más pequeños que resulten en “*noise trading*”. En la práctica esta idea no funcionó ya que aumentó los costos de transacción, probablemente debido a que estamos agregando dos nuevas aproximaciones, con lo cual fue descartada. También se considero la metodología de *Interpretable Sparse Proximate Factors* de Pelger y Xiong (2022), pero también resultó en un aumento de costos de transacción.

Luego consideramos una idea más simple pero efectiva, a la hora de estimar los β_t en la regresión agregarle una penalidad de acuerdo a los costos de transacción. Más específicamente, en lugar de OLS se plantea el siguiente problema,

$$\min_{\beta_t} \left(\frac{1}{L} \sum_{s=t-L+1}^t \|R_s^e - \beta_t(w_t^F)^\top R_s^e\|_2^2 \right) + \lambda \|\beta_t(w_t^F)^\top - \beta_{t-1}(w_{t-1}^F)^\top\|_{1,1}, \quad (5.1)$$

donde en el segundo término $\|\cdot\|_{1,1}$ denota a la norma uno “*element-wise*”, es decir la norma uno de la matriz vista como un vector aplanado. El parámetro λ es un parámetro de regularización que especifica el *trade-off* entre el *mean squared error (MSE)* y el “costo de transacción promedio”. Notamos que el costo de transacción con este método en general tampoco es cero, ya que al cambiar w_t^F y w_{t-1}^F el término de regularización no puede hacerse arbitrariamente pequeño. Sin embargo, eligiendo un λ adecuado se obtiene una reducción significativa de los costos de

transacción. Este es un problema convexo y por lo tanto puede ser resuelto de forma eficiente, se utiliza el *solver* ECOS (Domahidi et al. 2013) de CVXPY (Steven Diamond y Boyd 2016) que utiliza un método de punto interior. Denotamos a este método *Ordinary Least Squares with Trading Frictions (OLSTF)*.

5.1.2. Estimación del modelo de series de tiempo

Explicamos ahora cómo estimar los parámetros del modelo de Ornstein-Uhlenbeck de la ecuación 3.29. Tenemos que la solución está dada por la ecuación 3.30. Podemos discretizar esta ecuación en pasos de duración $\Delta t = 1$ (es decir, un día) o más generalmente de acuerdo a la frecuencia de nuestras observaciones. Se obtiene por lo tanto una serie de tiempo autoregresiva $AR(1)$:

$$X_t = \mu + \phi(X_{t-1} - \mu) + \nu_t, \quad t = 1, \dots, L, \quad (5.2)$$

donde $\phi = e^{-\kappa}$ y $\nu_t \sim N(0, \sigma_\nu^2)$ con $\sigma_\nu^2 = \frac{\sigma^2(1-e^{-2\kappa})}{2\kappa}$. Un proceso $AR(1)$ se puede estimar con una regresión de series de tiempo por OLS o también por otros métodos más sofisticados.

Se plantea la siguiente regresión:

$$X_t = a + bX_{t-1} + \nu_t, \quad t = 1, \dots, L,$$

con la sustitución $a = \mu(1 - \phi)$, $b = \phi$. Luego, con los estimadores por OLS de \hat{a} , \hat{b} , $\hat{\sigma}_\nu^2$ obtenemos estimadores *plug-in* de los parámetros del proceso de Ornstein-Uhlenbeck por:

$$\hat{\kappa} = -\log(\hat{b}), \quad \hat{\mu} = \frac{\hat{a}}{1 - \hat{b}}, \quad \hat{\sigma}^2 = \frac{\hat{\sigma}_\nu^2 2\hat{\kappa}}{1 - \hat{b}^2}.$$

Alternativamente, la regresión de la ecuación 5.2 se puede hacer por métodos robustos cuando no se piensa que los errores sean normales. La motivación detrás de esto son las colas pesadas de los retornos diarios de acciones, o presencia de retornos atípicos (*outliers*). Un método posible es cambiar la función de pérdida cuadrática utilizada en OLS por la función de pérdida de Huber (*Huber loss*) dada por:

$$L(x) = \begin{cases} \frac{1}{2}x^2 & \text{si } x \leq \delta \\ \delta(|x| - \frac{\delta}{2}) & \text{si } x > \delta \end{cases} \quad (5.3)$$

donde δ es un parámetro. El comportamiento lineal de esta función para errores mayores a δ hace que se reduzca significativamente la sensibilidad de las regresiones a la presencia *outliers*.

Este tipo de regresiones puede resolverse de forma eficiente reformulando al problema de minimización como uno convexo con restricciones o también a través de *recursive weighted least squares*. Se utiliza el valor por defecto $\delta = 1,35$ para determinar la región cuadrática de L .

5.2. Estrategias de aprendizaje reforzado

5.2.1. Planteamiento del problema de programación dinámica estocástica

Para aplicar las técnicas del aprendizaje reforzado, se debe especificar el problema en este lenguaje, es decir se debe definir un estado, control, función de dinámica, función de costos, tipo de horizonte, etc. Esto requiere de tomar un gran número de decisiones de modelado, supuestos y simplificaciones a la hora de plantear el problema como uno de programación dinámica estocástica.

En un primer enfoque, nuestro punto de partida es la estrategia, o más bien política en el lenguaje del aprendizaje reforzado, de Avellaneda y Lee (2010). Los factores del modelo son construidos por PCA y nuestro modelo de series de tiempo es un $AR(1)$ como en AL. En el problema de SDP que se plantea, se considera a un solo portafolio de arbitraje a la vez, por lo tanto tomamos como control del problema un peso normalizado a $\{-1, 0, 1\}$, que es nuestro espacio de controles del problema. Es decir,

$$u_t \in U_t(x_t) = \{-1, 0, 1\}.$$

Esto es, si $u_t = 1$ entonces compramos *long* un dólar en el portafolio de arbitraje, si es -1 se vende *short* y si es 0 no se tiene una posición abierta. Se podría plantear directamente el problema de optimización conjunta para todos los portafolios de arbitraje. Sin embargo, esto hace que el espacio de controles sea enorme, este tiene 3^N controles y además se debe conocer la dinámica conjunta de todos los portafolios de arbitraje, con lo cual plantearlo o resolverlo con estas técnicas es muy complejo.

Por simplicidad de la notación, vamos a suponer que nos encontramos al final del día t , habiendo observado R_t^e , nuestro estado x_t reflejará esta información. En ese momento debemos elegir nuestro control u_t para el día siguiente, y esperar a observar la aleatoriedad dada por w_t que tiene la información de retornos del día siguiente. Vamos a suponer que todas las operaciones se realizan al cierre del mercado. Esto es una idealización ya que en la práctica no se opera

instantáneamente y exactamente a ese precio.

Recordamos que dada la serie de residuos $\varepsilon_{i,t}^L = (\varepsilon_{i,t-L+1}, \dots, \varepsilon_{i,t-1}, \varepsilon_{i,t})$ se extrae la señal $\theta^{AL}(\varepsilon_{i,t}^L)$ de acuerdo a la ecuación 3.31, estimando los parámetros del proceso $AR(1)$. Esta señal es suficiente para filtrar, calcular el *score* y aplicar la política de AL en base a ella. El problema es que si uno se queda solo con $\theta^{AL}(\varepsilon_{i,t}^L)$, no se puede calcular el estado al día siguiente, ya que es necesaria la serie de los residuos. Esto no es una obstrucción real, ya que simplemente se puede usar a la serie de los últimos L residuos como parte del estado y extraer $\theta^{AL}(\varepsilon_{i,t}^L)$ a partir de este. Más aún, al formular el estado en términos de los residuos esta formulación resulta agnóstica respecto al modelo elegido y puede utilizarse con otros modelos. Ahora bien, la función de costo del problema tiene en cuenta a los costos de transacción y tenencia, con lo cual es necesario conocer el control o peso anterior en el portafolio de arbitraje (es decir, u_{t-1}). También se necesita conocer Φ_t, Φ_{t-1} para traducir estos pesos en portafolios en pesos de los activos de acuerdo a 3.3.5. Por lo tanto, se toma como estado del problema a

$$x_t = (\varepsilon_{i,t}^L, u_{t-1}, \Phi_t, \Phi_{t-1}). \quad (5.4)$$

El costo de transacción de esta posición es

$$\xi \|Q_t - Q_{t-1}^+\|_1 \approx \xi \|Q_t - Q_{t-1}\|_1 = \xi \|u_t \Phi_t^\top e_i - u_{t-1} \Phi_{t-1}^\top e_i\|_1,$$

donde hacemos la simplificación $Q_{t-1}^+ \approx Q_{t-1}$ (i.e. se ignora el retorno de la posición en un día) y donde e_i denota el i -ésimo vector de la base canónica de \mathbb{R}^N . Notamos que $\Phi_t^\top e_i = \Phi_{i,t}$ es la fila i -ésima de la matriz Φ_t , y nos dice cuáles son los pesos de cada activo en el portafolio de arbitraje i -ésimo. Observamos que el costo de transacción no es cero aún cuando no haya un cambio de posición si Φ_t y Φ_{t-1} difieren, aunque en general este costo por rebalanceo suele ser un orden de magnitud menor.

Se puede simplificar el costo de transacción si se asume que $\Phi_t = \Phi_{t-1}$, es decir ignorando los costos de rebalanceo debido al cambio en los factores, luego el costo es cero si no hay cambios en el control. Por lo tanto tenemos

$$\xi \|Q_t - Q_{t-1}^+\|_1 \approx \xi \|\Phi_{i,t}\|_1 |u_t - u_{t-1}|.$$

Más aún, se puede simplificar $\|\Phi_{i,t}\|_1$ reemplazándolo por un valor promedio, ya que esperamos que estas normas sean relativamente estables en el tiempo. Si bien esto hace que nuestro

problema sea aproximado, esta es solo una aproximación que se hace con el fin de encontrar la estrategia, a la hora de operar vamos a considerar los costos del portafolio completo de forma exacta según la ecuación 3.34.

El costo de tenencia de este portafolio de arbitraje está dado por

$$\delta \| \text{mín}(Q_t, 0) \|_1 = \delta \| u_t \text{mín}(\Phi_t^\top e_i, 0) \|_1 = \delta |u_t| \| \text{mín}(\Phi_{i,t}, 0) \|_1,$$

donde $\text{mín}(\Phi_{i,t}, 0)$ es la posición *short* del portafolio de arbitraje i -ésimo. Aquí también se puede simplificar esta constante por un valor promedio. También es esperable que este sea aproximadamente la mitad que $\|\Phi_{i,t}\|_1$, ya que siendo portafolios neutrales al mercado esperamos que aproximadamente las posiciones *long* y *short* tengan pesos similares. Luego,

$$\delta \| \text{mín}(Q_t, 0) \|_1 \approx \frac{\delta}{2} |u_t| \|\Phi_{i,t}\|_1$$

El retorno a un día del portafolio de arbitraje dado el control u_t está dado por

$$u_t \varepsilon_{i,t+1} = u_t [R_{i,t+1}^e - \beta_{i,t}^\top (w_t^F)^\top R_{t+1}^e] = u_t e_i^\top \Phi_t R_{t+1}^e,$$

de acuerdo a las ecuaciones 3.26 y 3.27.

Luego la función de costo del problema simplificado se define como:

$$g_t(x_t, u_t, w_t) = -U \left(u_t w_t - \|\Phi_{i,t}\|_1 (\xi |u_t - u_{t-1}| + \frac{\delta}{2} |u_t|) \right),$$

donde $U : \mathbb{R} \rightarrow \mathbb{R}$ es una función de utilidad cóncava (e.g. $U(x) = \ln(1+x)$, $U(x) = x - \gamma x^2$ para $\gamma \geq 0$, etc.), y donde la componente estocástica del problema está dada por el retorno del portafolio de arbitraje $w_t = \varepsilon_{i,t+1}$, que es desconocida al cierre del día t en el cual debemos tomar nuestra posición para el día $t+1$.

Ahora bien, dado que el costo del problema a tiempo t depende de $\|\Phi_{i,t}\|_1$, para poder simular este problema deberíamos poder calcular $\|\Phi_{i,s}\|_1$ para $s = t+1, t+2, \dots$ pero estos dependen de retornos futuros. El supuesto que hacemos es que $\|\Phi_s\|_1$ se mantiene estable, es decir no cambia significativamente, en una ventana de días pequeña respecto al período de estimación. Esto es, $\|\Phi_{i,s}\|_1 = \|\Phi_{i,t}\|_1$ para $s = t, t+1, \dots, t+H$. Empíricamente, los portafolios de arbitraje que se construyen tienen una exposición bruta de aproximadamente 4 dólares, con dos dólares *long* y dos *short*.

Notamos que para el problema simplificado, el estado x_t en la ecuación 5.4 puede simplificarse y tomarlo como $x_t = (\varepsilon_{i,t}^L, u_{t-1})$. La ecuación 3.9 de la dinámica del estado consiste simplemente en agregar $w_t = \varepsilon_{i,t+1}$, remover $\varepsilon_{i,t-L+1}$, y reemplazar u_{t-1} por u_t . Esto es,

$$x_{t+1} = f_t(x_t, u_t, w_t) = (\varepsilon_{i,t+1}^L, u_t).$$

El problema se formula como uno de horizonte móvil, es decir, se toma un horizonte de H días en el futuro y el objetivo es resolver un problema como aquel de la ecuación 3.19. Faltan dos ingredientes para que el problema se formule completamente, el primero es la función de costo final $g_H(x_{t+H})$. Si se quiere es posible tomar a esta función como cero, sin embargo en este caso resulta más apropiado tomar a esta como el costo de transacción para cerrar la posición, esto es

$$g_H(x_{t+H}) = -U(-\|\Phi_{i,t}\|_1 \xi |u_{t+H-1}|) = -U(-\|\Phi_{i,t}\|_1 \xi |\text{pr}_2(x_{t+H})|),$$

donde $\text{pr}_2(x_t) = u_{t-1}$ indica la proyección en la última coordenada del estado dada por el control del día anterior. Esta elección se debe a que queremos que la estrategia abra posiciones que resulten beneficiosas en el corto plazo, más específicamente dentro del horizonte móvil que definimos. Se toma $H = 10$ días, ya que de acuerdo a estudios empíricos en Avellaneda y Lee (2010), la mediana del tiempo de reversión es 7.5 días, y el 75 % de los portafolios de arbitraje revierte en menos de 11 días.

El último ingrediente para especificar el problema de forma completa consiste en el modelo probabilístico que sigue $w_t = \varepsilon_{i,t+1}$. Esto lo hacemos de acuerdo al modelo $AR(1)$ de Avellaneda y Lee (2010) sobre los residuos acumulados (X_1, \dots, X_L) definidos en la ecuación 3.28. De acuerdo al modelo definido en la ecuación 5.2 tenemos que

$$\varepsilon_{i,t+1} = X_{L+1} - X_L = \mu + \phi(X_L - \mu) + \nu_{t+1} - X_L = (1 - \phi)(\mu - X_L) + \nu_{t+1},$$

donde $\phi = e^{-\kappa}$ y $\nu_{t+1} \sim N(0, \sigma_\nu^2)$ con $\sigma_\nu^2 = \frac{\sigma^2(1-e^{-2\kappa})}{2\kappa}$. Con lo cual, queda especificada la distribución de $w_t = \varepsilon_{i,t+1}$ condicional al estado x_t y el control u_t , aunque es independiente de este último.

Esto nos permite formular el problema como

$$\begin{aligned} \min_{\mu_t, \dots, \mu_{t+H-1}} \mathbb{E} & \left[\sum_{s=t}^{t+H-1} -U \left(\mu_s(x_s) w_s - \|\Phi_{i,t}\|_1 (\xi |\mu_s(x_s) - \mu_{s-1}(x_{s-1})| + \frac{\delta}{2} |\mu_s(x_s)|) \right) \right. \\ & \left. -U (\|\Phi_{i,t}\|_1 \xi |\mu_{t+H-1}(x_{t+H-1})|) \right] \\ \text{sujeto a } \mu_s(x_s) & \in \{-1, 0, 1\}, x_{s+1} = (\varepsilon_{i,s+1}^L, \mu_s(x_s)), \end{aligned} \quad (5.5)$$

dado x_t .

5.2.2. La estrategia *naive rollout*

Una vez planteado el problema en 5.5, podemos aplicar a este los métodos introducidos en la subsección 3.2. En un primer enfoque se ataca al problema mediante el método del *rollout* o *one-step lookahead* introducido en la subsección 3.2.3, tomando como política base la estrategia de Avellaneda y Lee. Recordamos que lo que queremos es evaluar los controles $u_t \in \{-1, 0, 1\}$ para elegir el que nos dé un menor Q -factor $Q_t^\mu(x_t, u_t)$ cuando a partir del día siguiente aplicamos la política de Avellaneda y Lee basada en el *score*. Para evaluar estos Q -factores podemos hacer M simulaciones del problema empezando con cada control como en la ecuación 3.15. Una ventaja de este problema es que dado que la distribución de los w_s no depende de los controles utilizados (implícitamente estamos asumiendo que nuestra posición no mueve al mercado), se simulan M secuencias de $w_s^{(i)}$ y se utilizan para evaluar los tres controles. Se toma el número de simulaciones $M = 1000$, este debe ser grande ya que la convergencia del método de Monte Carlo es lenta (de orden $\approx 1/\sqrt{M}$). Esta política está garantizada a mejorar a la política base de acuerdo a la proposición 3.2.3, al menos cuando el problema siga realmente al modelo de OU, y cuando los parámetros del modelo son conocidos. Esta mejora puede evaluarse simulando procesos de OU discretizados.

Ahora bien, en la práctica los parámetros del modelo son desconocidos, estos son ϕ, σ_ν^2 del $AR(1)$ o alternativamente κ, σ^2 del proceso de OU. En un primer enfoque, lo que se hace es simplemente reemplazar los valores verdaderos por los estimados en la subsección 5.1.2, y realizar las simulaciones con estos parámetros. Este método es lo que llamamos la estrategia *rollout* Avellaneda y Lee (RAL) o *naive rollout*, que consiste en la aplicación *naive* o ingenua del *rollout* a la estrategia base de Avellaneda y Lee. Este método es reforzado por las mismas

técnicas de *screening* utilizadas para AL. Esto es, en un primer paso se evalúa el R^2 y el κ y se aplica un umbral según el cual se descarta el portafolio de arbitraje si no se cumplen estos requerimientos. Es decir, si $R^2 \leq R_{crit}^2$ o $\kappa \leq \kappa_{crit}$ entonces ni siquiera se realizan las simulaciones y se elige $u_t = 0$.

5.2.3. La estrategia *rollout with bootstrap*

Dado que los parámetros son desconocidos, un enfoque más apropiado a utilizar es el de plantear el problema como uno de información parcial de los estados introducido en la subsección 3.2.6. Ahora el problema es un POMDP con un parámetro desconocido $\theta = (\phi, \sigma_v^2)$ ó alternativamente reparametrizando se puede tomar (κ, σ^2) . Si bien θ es desconocido, se toma un enfoque bayesiano sobre este y se tiene una distribución de θ condicional a las observaciones, que en este caso será la serie de los residuos acumulados (X_1, \dots, X_L) . Para definir la distribución de θ condicional a las observaciones, utilizamos un método *bootstrap* para series de tiempo introducido en Efron y Tibshirani (1986) similar a los ya mencionados en la sección 3.1.3.

Dada una serie de tiempo $AR(1)$ con media cero:

$$Z_t = \phi Z_{t-1} + \varepsilon_t, \quad t = 1, \dots, T,$$

con $|\phi| < 1$ y $\varepsilon_t \sim N(0, \sigma^2)$ independientes, entonces primero se estiman $\hat{\phi}, \hat{\sigma}^2$ por OLS. Dado $\hat{\phi}$ se tienen residuos estimados,

$$\hat{\varepsilon}_t = Z_t - \hat{\phi} Z_{t-1} \tag{5.6}$$

Esto permite generar muestras *bootstrap* empezando con $Z_1^{(b)} = Z_1$ y generando los siguientes Z de acuerdo a

$$Z_t^{(b)} = \hat{\phi} Z_{t-1}^{(b)} + \hat{\varepsilon}_t^{(b)}, \quad t = 2, \dots, T, \tag{5.7}$$

donde los $\hat{\varepsilon}_t^{(b)}$ se obtienen haciendo muestreo con reposición de los residuos estimados en la ecuación 5.6, para $b = 1, \dots, B$. Luego de obtener las muestras *bootstrap* $\{(Z_1^{(b)}, \dots, Z_T^{(b)})\}_{b=1}^B$, se puede volver a estimar para cada una de ellas los parámetros $\theta^{(b)} = (\phi^{(b)}, (\sigma^2)^{(b)})$ por OLS. Esto define una distribución a posteriori de los parámetros $\theta = (\phi, \sigma^2)$ condicional a las observaciones, que se obtiene fácilmente y sin declarar una distribución a priori.

Una alternativa a este método de *bootstrap* es un método completamente paramétrico, según el cual se simulan los residuos de acuerdo a muestras independientes $\hat{\varepsilon}_t^{(b)} \sim N(0, \hat{\sigma}^2)$ y luego

se procede como en la ecuación 5.7.

Notamos que una simplificación que se está haciendo en este método es suponer que μ es conocida cuando en realidad lo que se hace en la práctica es reemplazarla por su estimador $\hat{\mu}$ dado por el promedio. Esto no es necesario realmente, podría re-estimarse también por *bootstrap*.

Con esta distribución simulada sobre los parámetros desconocidos podemos aplicar el método *rollout* introducido en la ecuación 3.22. Dadas las simulaciones de los parámetros $\theta^{(b)} = (\phi^{(b)}, (\sigma_\nu^2)^{(b)})$ obtenidas por *bootstrap* se simulan $X_{L+1}^{(b)}, \dots, X_{L+H}^{(b)}$ de acuerdo a

$$X_s^{(b)} = \hat{\mu} + \phi^{(b)}(X_{s-1}^{(b)} - \hat{\mu}) + \nu_s^{(b)}, \quad s = L + 1, \dots, L + H,$$

con $\nu_s^{(b)} \sim N(0, (\sigma_\nu^2)^{(b)})$. Esto define a los $w_s^{(b)}$ tomando diferencias. Con estas simulaciones se evalúan los controles estimando los Q -factores de acuerdo a 3.23 donde a partir del siguiente día se aplica la política de Avellaneda y Lee. Se toman $B = 1000$ muestras *bootstrap* y para cada una de ellas se hacen simulaciones, con lo cual el costo computacional es del mismo orden que el *naive rollout*. Este segundo método lo llamamos *rollout con bootstrap* de Avellaneda y Lee, y en realidad hay dos variantes, la versión no paramétrica y la versión paramétrica de acuerdo a cuál método *bootstrap* se utiliza (las notamos *rollout with non-parametric bootstrap, RNPB-AL* y *rollout with parametric bootstrap, RPB-AL* respectivamente).

Universidad de
San Andrés

6. Resultados

6.1. Resultados en un entorno de simulado

Para evaluar las estrategias propuestas generamos simulaciones de procesos $AR(1)$ y se evalúa las estrategias sobre estos como si fueran residuos operables. Tomamos los parámetros $\mu = 0, \phi = 0,9$ y $\sigma_v = 0,005$ que son similares a los que se obtienen al estimar los procesos $AR(1)$ sobre portafolios de arbitraje. Graficamos una simulación de estas características en la figura 2.

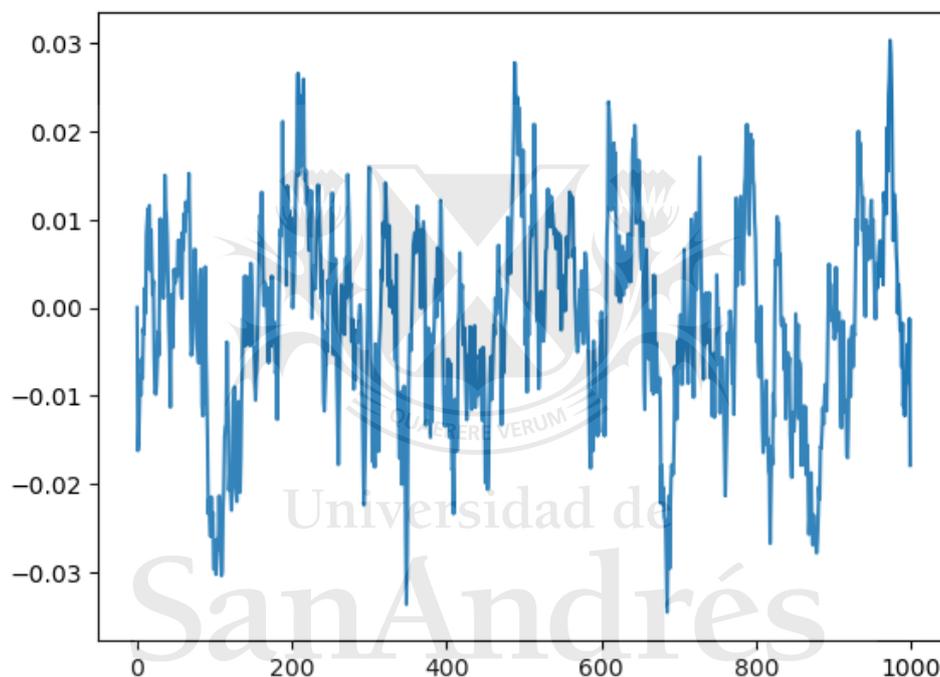


Figura 2: Gráfico de $(X_t)_t$, una serie de tiempo $AR(1)$ simulada con $\mu = 0, \phi = 0,9, \sigma_v = 0,005$, para 1000 días.

En la figura 3 observamos el retorno acumulado obtenido por las cuatro estrategias para un solo activo simulado durante 1000 días. Los algoritmos utilizan una ventana de estimación de $L = 60$ días. Se observa que las estrategias están bastante correlacionadas y usualmente toman la misma decisión. También se observa un rendimiento superior de las estrategias basadas en *rollout*.

Para evaluar de forma más robusta los algoritmos decidimos hacerlo en el horizonte de $H = 10$ días pero simulando un gran número de veces cada estrategia. Esto es, se simulan $N =$

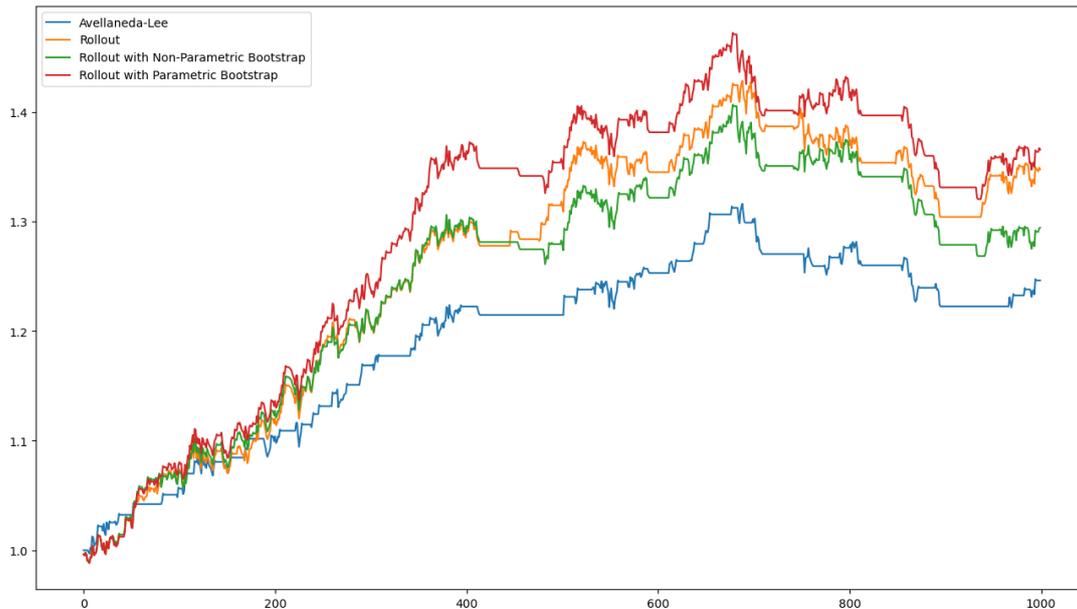


Figura 3: Retorno acumulado de las estrategias para un activo simulado $AR(1)$ con $\phi = 0,9$, $\sigma_v = 0,005$, exposición por activo de $\Lambda = 2$ dólares, costo de transacción $\xi = 0,0005$ y sin costos de tenencia (i.e. $\delta = 0$).

10000 “activos simulados” idénticos e independientes. Graficamos la distribución de retornos resultantes para los activos en los que el algoritmo opera en la ventana de 10 días, es decir al menos abre una posición. Estos son los retornos relevantes ya que el *equity* se reparte solo sobre los activos donde se abren posiciones. Se observa en la figura 4 que las estrategias de *rollout* (RAL) y *rollout with bootstrap* (RBALNP, RBALP) obtienen un retorno medio más bajo que la estrategia base de Avellaneda y Lee (AL), y una volatilidad similar. También notamos que la estrategia de AL es más conservadora ya que abre alrededor de la mitad de posiciones que las estrategias de *rollout*, ver cuadro 2. Notamos que las estrategias de *rollout* obtienen resultados muy similares. A primera vista uno podría concluir que es preferible entonces la estrategia base de Avellaneda-Lee, sin embargo hay que tener en cuenta que de acuerdo a esta simulación, en 10 AL solo abre 65 posiciones en un portafolio con 500 activos, mientras que las estrategias de *rollout* operan aproximadamente 105 activos. En el caso de activos independientes, al tener más del doble de activos esto logra que la varianza del retorno promedio se reduzca lo suficiente como para obtener una estrategia con mejor *Sharpe ratio*, como observamos en 2.

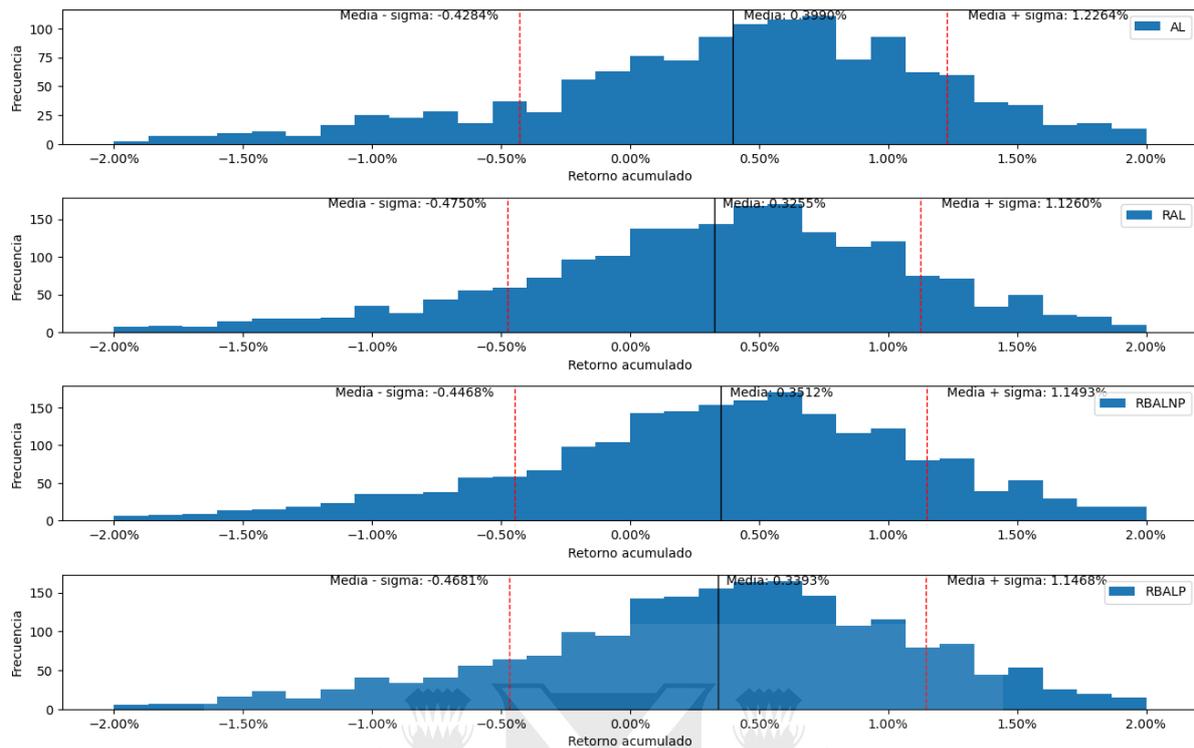


Figura 4: Distribución de retornos individuales acumulados obtenidos en una ventana de $H = 10$ días para $N = 10000$ activos simulados $AR(1)$ con $\mu = 0, \phi = 0,9, \sigma_v = 0,005$ desconocidos (es decir, estimados por los algoritmos), exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costos de tenencia $\delta = 0,0001$.

Estrategia	N° de operaciones	μ	σ	Sharpe ratio
AL	1335	0.40 %	0.83 %	2.96
RAL	2033	0.33 %	0.80 %	3.91
RBALNP	2102	0.35 %	0.80 %	4.42
RBALP	2103	0.34 %	0.81 %	4.31

Cuadro 2: Número de operaciones por estrategia en la simulación de la figura 4, retorno medio (μ) y volatilidad (σ) por activo, y *Sharpe ratio* aproximado de 10 días de las estrategias para 500 activos.

En una segunda simulación se generan $N = 1000$ activos “sintéticos” con parámetros $\mu = 0, \phi = 0,9, \sigma_v = 0,005$. Se simulan procesos $AR(1)$ independientes en un período de 120 días y se los opera a partir del día $L = 60$. Se forma un portafolio que asigna los pesos de acuerdo

a la metodología de la subsección 3.3.5. Se observa en la figura 5 que la estrategia base de AL obtiene retornos superiores en esta simulación que las basadas en *rollout*, que prácticamente no se diferencian, sin embargo es significativamente más volátil que estas, por lo que resulta en un menor *Sharpe ratio* como observamos en el cuadro 3.

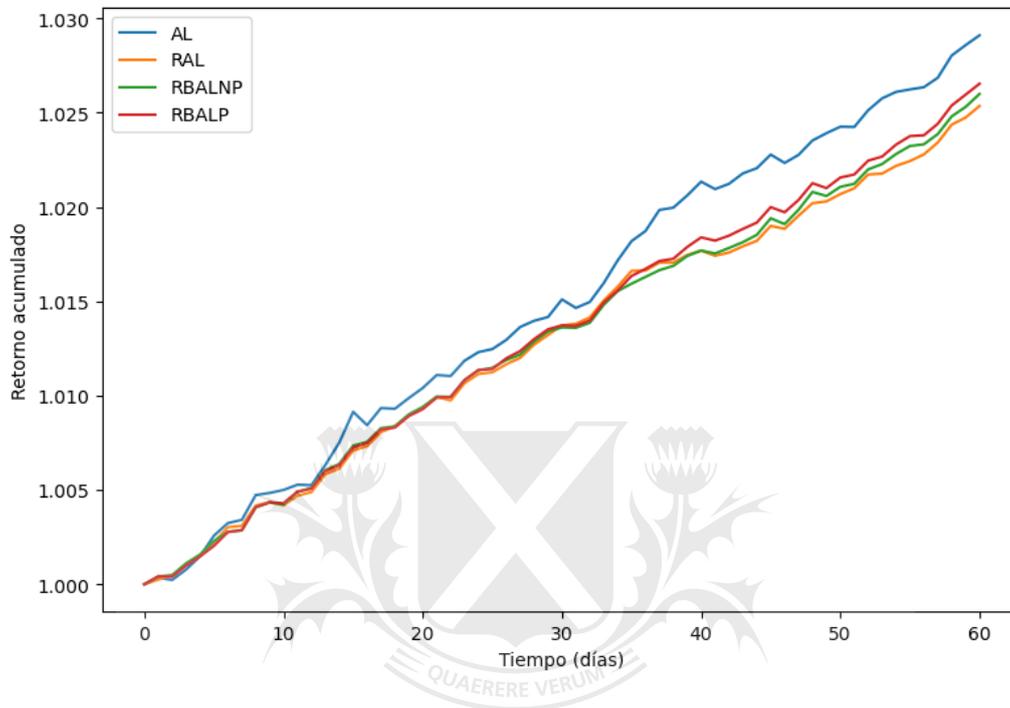


Figura 5: Retorno acumulado en 60 días para $N = 1000$ activos simulados $AR(1)$ $\phi = 0,9, \sigma_v = 0,005$, exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costo de tenencia $\delta = 0,0001$.

Estrategia	Retorno Acumulado	Sharpe ratio mensual
AL	2.91 %	4.80
RAL	2.54 %	6.16
RBALNP	2.60 %	6.19
RBALP	2.65 %	6.24

Cuadro 3: Retorno acumulado y *Sharpe ratio* mensual por estrategia en la simulación de la figura 5.

Para entender mejor como se diferencian las estrategias realizamos primero una regresión del retorno acumulado en función de X_0 estandarizado (i.e. el *score*). Para capturar la no-

linealidad de estas funciones se utiliza una regresión por núcleos (*Kernel Regression*). En la figura 6 se observan perfiles marcadamente distintos para las estrategias basadas en *rollout* respecto a Avellaneda y Lee. En general los retornos esperados son más altos y sorprendentemente resultan bastante asimétricos.

Luego, aplicamos un clasificador para predecir la probabilidad de elegir cada control en función de X_0 estandarizado. Se utiliza un clasificador basado en una red neuronal (con dos capas de 6 y 5 neuronas respectivamente) para capturar las no-linealidades de las funciones. Nuevamente se diferencia claramente en la figura 7 la estrategia de AL respecto a las de *rollout*. Estas estrategias en general resultan más “agresivas”, abriendo posiciones con mayor probabilidad. Notamos que si bien la estrategia de AL es una función *threshold* del *score*, esto ocurre solamente cuando se calcula el *score* con los parámetros verdaderos, pero aquí el modelo tiene que estimar μ, ϕ, σ_ν con lo cual en la práctica se observa un “*soft threshold*” en función del *score* teórico (i.e. con el parámetro verdadero).

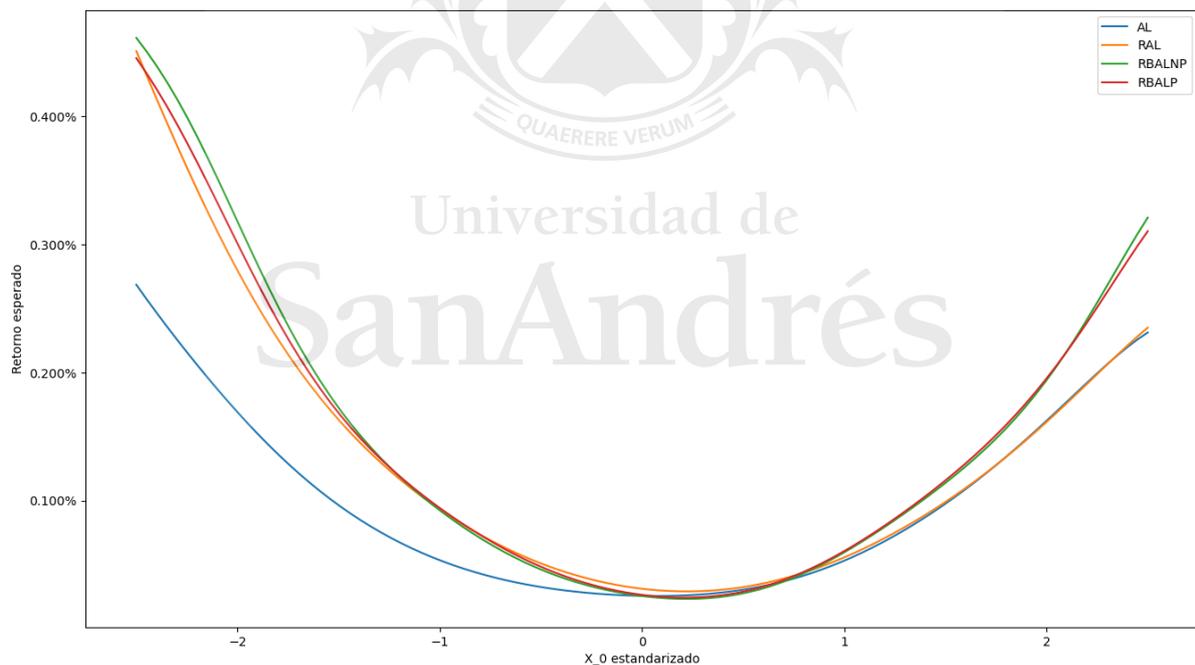


Figura 6: Regresión por núcleos del retorno acumulado en 10 días en función de X_0 estandarizado para cada estrategia. Para $N = 10000$ activos simulados $AR(1)$ $\phi = 0,9$, $\sigma_\nu = 0,005$, exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costo de tenencia $\delta = 0,0001$.

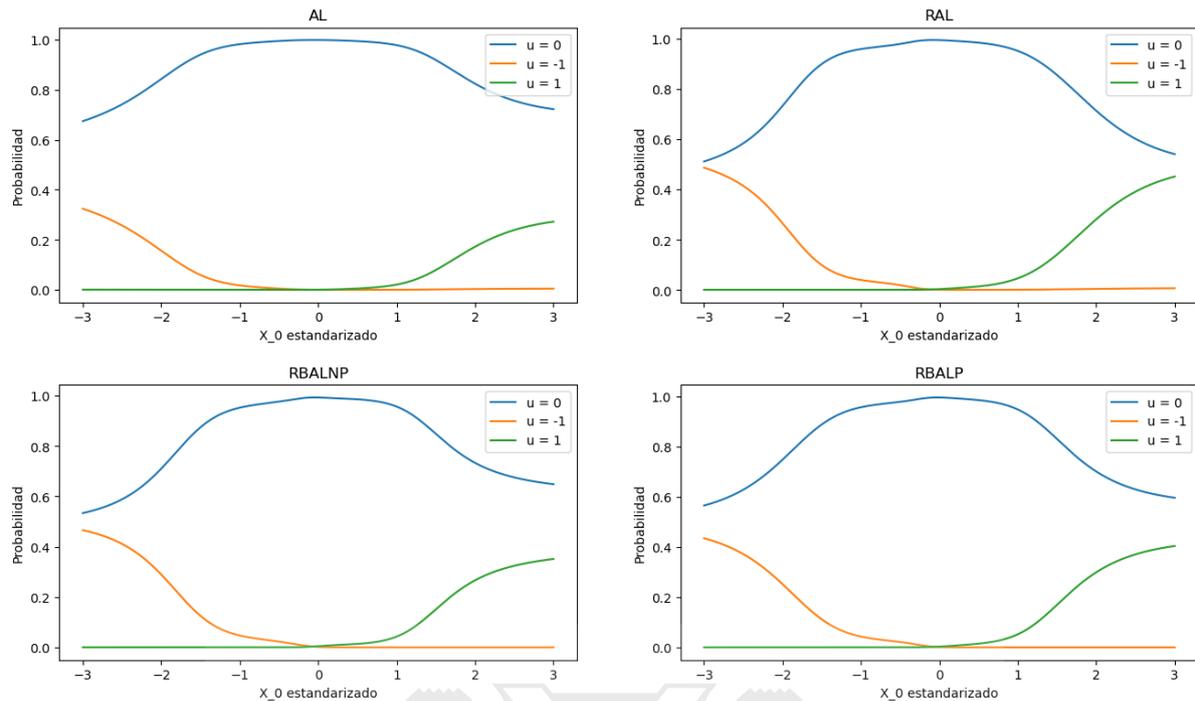


Figura 7: Clasificación por redes neuronales del control elegido en función de X_0 para cada estrategia. Para $N = 10000$ activos simulados $AR(1)$ $\phi = 0,9, \sigma_\nu = 0,005$, exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costo de tenencia $\delta = 0,0001$.

Por último variamos los parámetros ϕ y σ_ν para ver como cambia el retorno esperado incondicional (i.e. para X_0 aleatorio) en función de estos. En la figura 8 se observa que el retorno esperado crece en forma convexa a medida que ϕ se achica y en la figura 9 vemos que el retorno esperado aumenta linealmente con σ_ν . Esto implica que uno podría explotar estos *trade-offs* abriendo posiciones más grandes en los activos con mejores parámetros. El problema es que la estimación y el modelo deben ser buenos para que esto sea verdaderamente explotable.

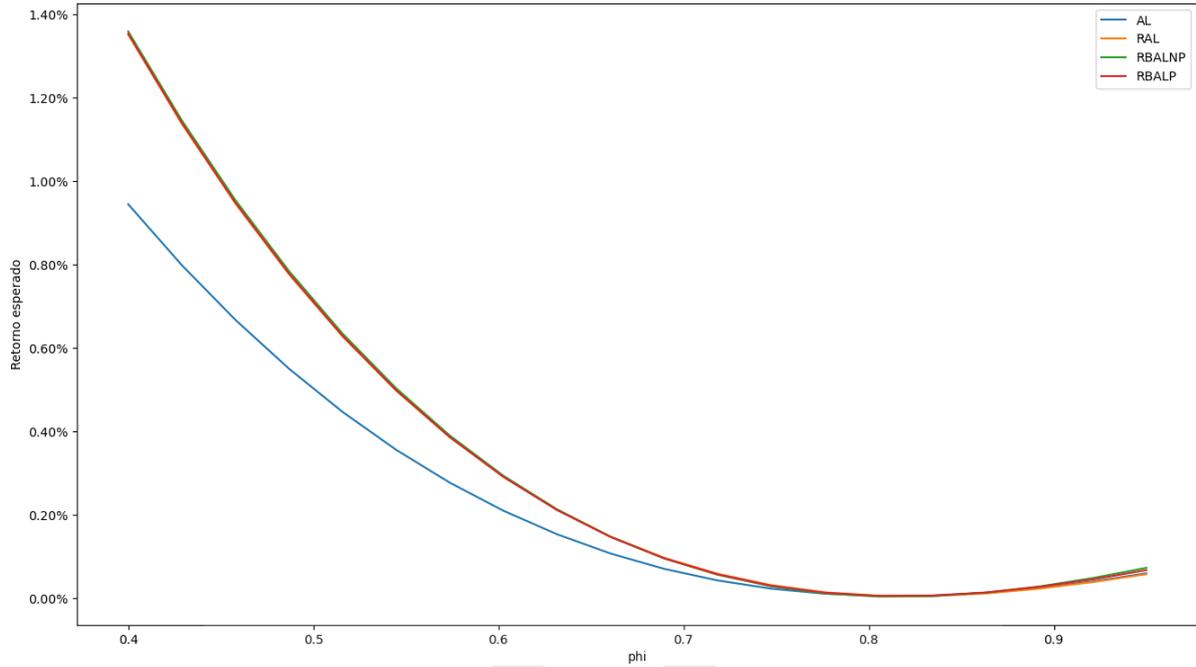


Figura 8: Regresión del retorno esperado en función de ϕ para cada estrategia. Para $N = 1000$ activos simulados $AR(1)$ para cada ϕ con $\sigma_\nu = 0,005$, exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costo de tenencia $\delta = 0,0001$.

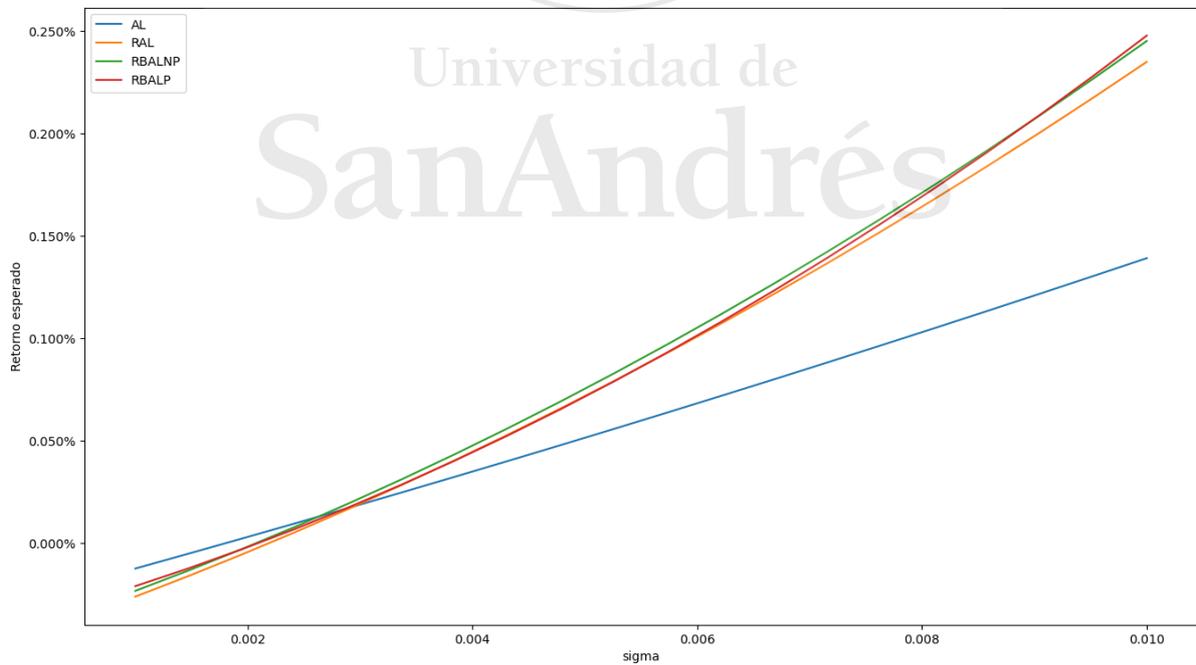


Figura 9: Regresión del retorno esperado en función de σ_ν para cada estrategia. Para $N = 1000$ activos simulados $AR(1)$ para cada σ_ν con $\phi = 0,9$, exposición por activo 2 dólares, costo de transacción $\xi = 0,0005$ y costo de tenencia $\delta = 0,0001$.

6.2. Resultados por *back-testing* en datos históricos

Para evaluar las estrategias obtenidas se utilizan los datos históricos obtenidos de acuerdo a la sección 4. Se tiene un poco más de 10 años de retornos de $N = 451$ componentes del índice S&P 500, desde el 1^{ro} de enero de 2013 al 18 de mayo de 2023.

Se utiliza una ventana de 252 días (i.e. aproximadamente un año de retornos) para obtener los factores de PCA, con lo cual nuestra estrategia comienza a operar recién el 1^{ro} de enero de 2014, para obtener los pesos $w_{t-1}^F \in \mathbb{R}^{N \times K}$. El número de factores es un hiperparámetro que tomamos en los valores $K = 5, 8, 10, 15$ para evaluar el rendimiento de las estrategias en función de K . Se utiliza una ventana de $L = 60$ días para estimar la matriz $\beta_{t-1} \in \mathbb{R}^{N \times K}$. También se utilizan $L = 60$ días para estimar la regresión de series de tiempo del $AR(1)$ de cada portafolio de arbitraje. Las estrategias basadas en *rollout* se evalúan con $N_{sim} = 1000$ simulaciones, y se usa un horizonte móvil de $H = 10$ días. Se utiliza $U(x) = \log(1 + x)$ para optimizar el retorno acumulado en 10 días en cada portafolio de arbitraje.

Se construye y se evalúa el portafolio siguiendo la metodología de la subsección 3.3.5, específicamente la ecuación 3.34 nos da el P&L del portafolio y el *equity* se distribuye de acuerdo a la ecuación 3.36 con $\Lambda_t = 1$ lo que nos da un *leverage bound* de uno para todo tiempo. Se evalúa las estrategias primero sin costos de transacción (i.e. $\xi = 0, \delta = 0$) como en Pelger et al. (2021), de acuerdo a su retorno total, *Sharpe ratio* (SR), retorno excedente medio (μ) y volatilidad (σ).

Los resultados del cuadro 4 muestran que las estrategias basadas en *rollout* obtienen resultados significativamente superiores a la estrategia base de Avellaneda y Lee, para cualquier número de factores de PCA (K). En términos de *Sharpe ratio*, la mejora en algunos casos más que duplica a la estrategia base. Se obtienen SR mayores a uno que son considerados altos, siendo el máximo conseguido con $K = 10$ factores en 1,15 para la estrategia RAL. También mejoran en las métricas de retorno excedente medio y volatilidad (más baja). Se observa que para $K = 15$ factores el rendimiento de las estrategias se empieza a deteriorar, lo que muestra que empiezan a incurrir en “*noise trading*”. Para $K = 5$ también los resultados son más pobres, lo que da indicios de que falta capturar factores de riesgo en el mercado.

En las figuras 10,11,12 y 13 se observan los retornos acumulados de las cuatro estrategias a lo largo del tiempo, para distintos números de factores de PCA (K). A simple vista se observa

que los retornos para $K = 10$ son menos volátiles sin perder retorno medio, lo que resulta en *Sharpe ratios* más altos en general. También se observa que cuanto mayor es la volatilidad del mercado, mayor es el retorno de las estrategias en general, hecho que se puede apreciar claramente en la crisis del Covid-19 de febrero a agosto de 2020.

K	Estrategia	Retorno Total	SR	\bar{R}_{II}	μ	σ
5	AL	13.04 %	0.29	1.31 %	0.31 %	1.08 %
	RAL	16.63 %	0.73	1.65 %	0.65 %	0.89 %
	RBALNP	14.85 %	0.47	1.48 %	0.48 %	1.03 %
	RBALP	18.61 %	0.81	1.83 %	0.83 %	1.02 %
8	AL	14.88 %	0.59	1.48 %	0.48 %	0.82 %
	RAL	17.61 %	1.10	1.73 %	0.73 %	0.67 %
	RBALNP	18.29 %	1.03	1.80 %	0.80 %	0.78 %
	RBALP	18.63 %	1.07	1.83 %	0.83 %	0.77 %
10	AL	15.63 %	0.76	1.55 %	0.55 %	0.73 %
	RAL	16.89 %	1.15	1.67 %	0.67 %	0.58 %
	RBALNP	17.93 %	1.12	1.76 %	0.76 %	0.68 %
	RBALP	16.25 %	0.90	1.61 %	0.61 %	0.68 %
15	AL	10.30 %	0.08	1.05 %	0.05 %	0.58 %
	RAL	13.01 %	0.66	1.31 %	0.31 %	0.46 %
	RBALNP	10.95 %	0.21	1.11 %	0.11 %	0.52 %
	RBALP	13.01 %	0.59	1.31 %	0.31 %	0.52 %

Cuadro 4: Resultados por número de factores de PCA (K), estrategia, *Sharpe ratio* (SR), retorno medio (\bar{R}_{II}), retorno excedente medio (μ), y volatilidad (σ) en términos anuales de los portafolios en un *back-test* desde el 2 de enero de 2014 al 18 de mayo de 2023. Sin costos de transacción, es decir $\xi = 0$, $\delta = 0$.

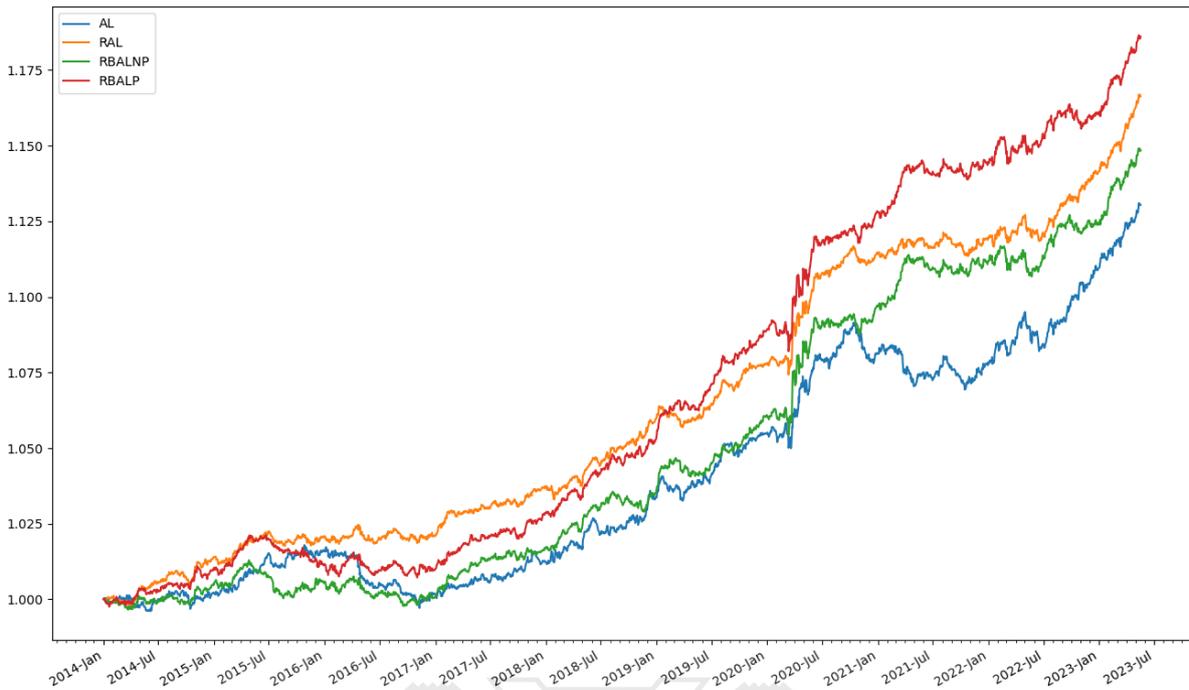


Figura 10: Retorno acumulado de los portafolios de cada estrategia con $K = 5$ factores de PCA, sin costos de transacción, para un *back-testing* desde el 2 de enero de 2014 al 18 de mayo de 2023.

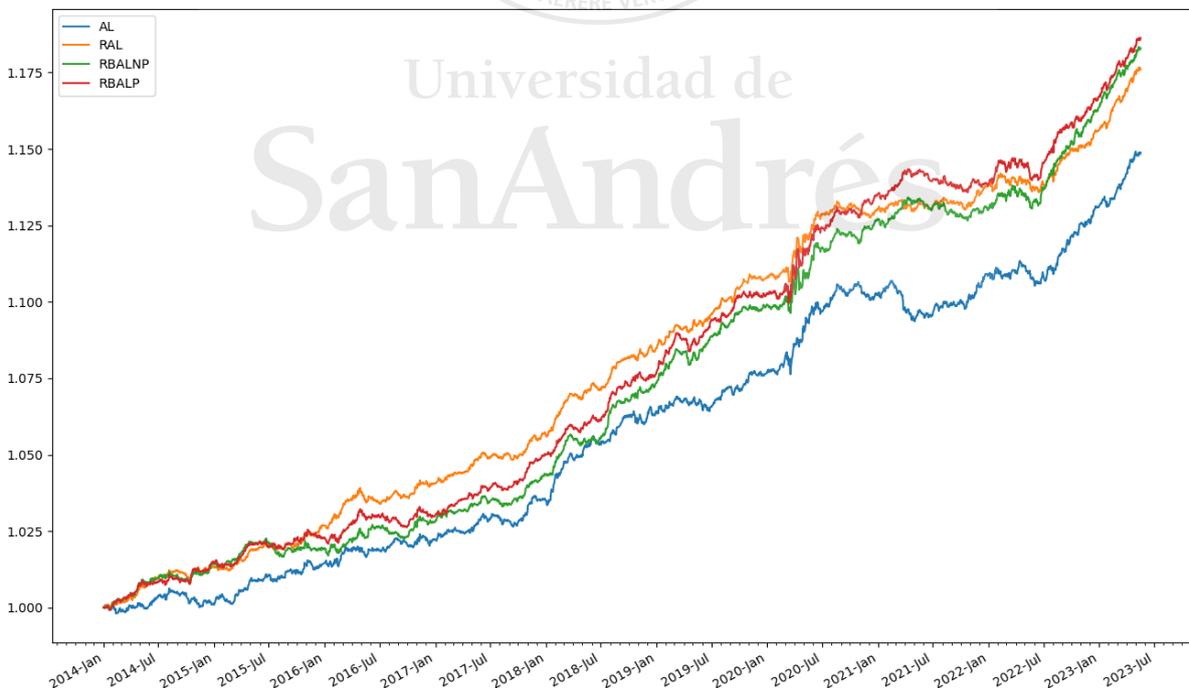


Figura 11: Retorno acumulado de los portafolios de cada estrategia con $K = 8$ factores de PCA, sin costos de transacción, para un *back-testing* desde el 2 de enero de 2014 al 18 de mayo de 2023.

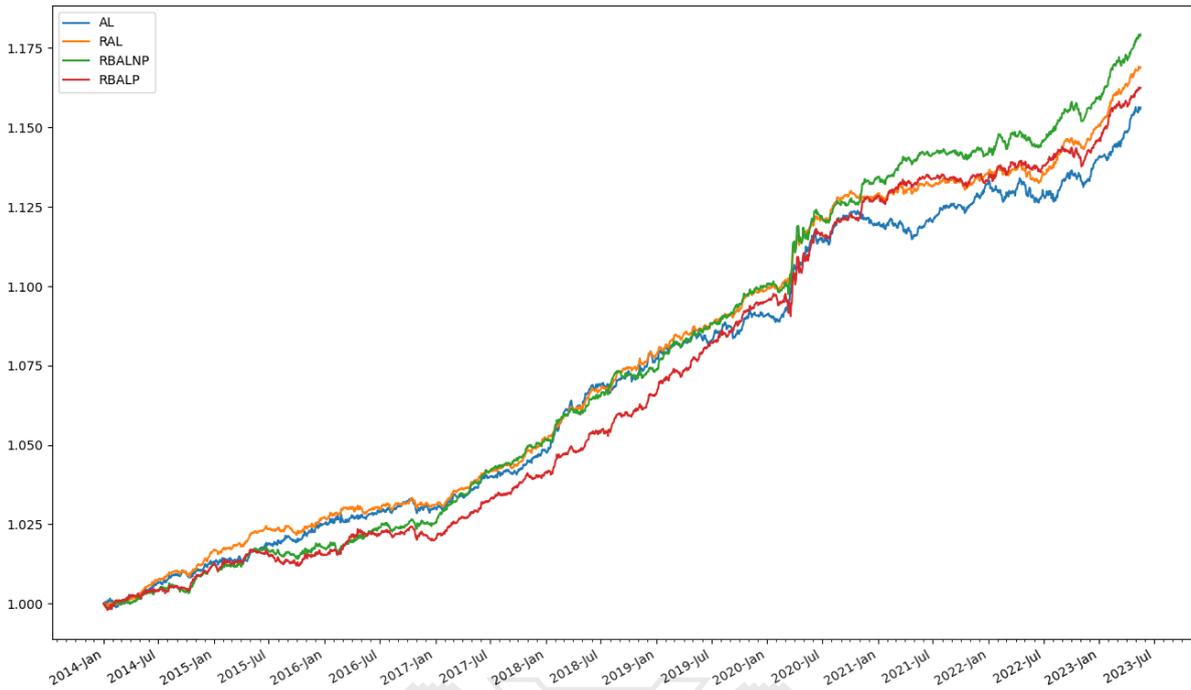


Figura 12: Retorno acumulado de los portafolios de cada estrategia con $K = 10$ factores de PCA, sin costos de transacción, para un *back-testing* desde el 2 de enero de 2014 al 18 de mayo de 2023.

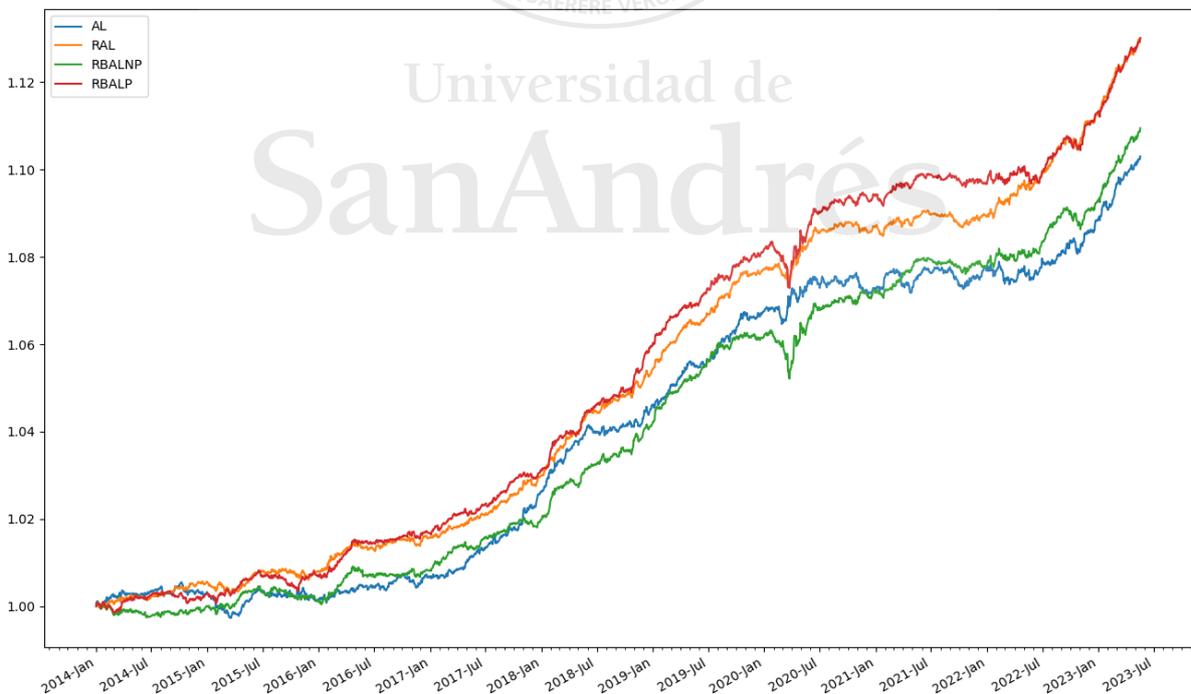


Figura 13: Retorno acumulado de los portafolios de cada estrategia con $K = 15$ factores de PCA, sin costos de transacción, para un *back-testing* desde el 2 de enero de 2014 al 18 de mayo de 2023.

Luego se evalúan las estrategias con costos de transacción. En principio se tomó $\xi = 0,0005$, es decir 5 *bps*, pero este costo resulta demasiado alto para ser representativo de los últimos años. De acuerdo a Hagströmer (2021), el *effective bid-ask spread* es de 2 *bps* para la mayoría de los componentes del S&P 500, lo que resulta en un costo base de 1 *bps* por transacción (es decir, la mitad ya que se reparte entre la compra y venta del activo). Decidimos tomar 2 *bps* como *proxy* de costo de transacción conservador, considerando componentes menos líquidas o costos del *broker*.

Por otra parte, como explicamos en la subsección 5.1.1, la re-estimación diaria de los factores de PCA y de los β por OLS resulta en altos costos de transacción por el rebalanceo propio de los factores en los portafolios de arbitraje. Los resultados para la estimación por OLS se muestran en el cuadro 5 y la figura 14. Se observa que las estrategias obtienen un SR negativo ya que sus retornos excedentes son negativos. Esto se debe a que gastan más en transaccionar que lo que ganan.

Es por esto que con OLSTF se le agrega un término de penalidad a los costos de transacción en la ecuación 5.1, lo que resulta en una reducción significativa de los costos de rebalanceo en los factores. En palabras, estamos agregando un *bias* a OLS para que los portafolios de arbitraje sean parecidos a los del período anterior. Elegimos el parámetro λ para que estos costos de rebalanceo resulten menores a medio *bps* por activo. Esto resulta en estrategias distintas aún en el caso de que no se consideren costos de transacción, ya que los betas de los factores son distintos. Se observa en el cuadro 6 que para $K = 8$ factores las estrategias obtienen un retorno más alto con OLSTF que cuando los betas se estiman por OLS. Sin embargo, la volatilidad de las mismas se incrementa prácticamente en igual proporción con lo cual los SR resultan similares a los del cuadro 4 para 8 factores. Notamos que RAL tiene un rendimiento bastante más pobre que el resto, lo que sorprende dado que en el cuadro 4 RAL era la estrategia con mejor *performance* en SR. Cuando agregamos los costos de transacción las estrategias se mantienen rentables. En términos de SR las estrategias de *rollout with bootstrap* obtienen resultados similares a Avellaneda y Lee, pero no logran mejorarlo.

Estrategia	Retorno Total	SR	\bar{R}_{II}	μ	σ
AL	4.41 %	-0.65	0.46 %	-0.54 %	0.82 %
RAL	6.96 %	-0.43	0.72 %	-0.28 %	0.66 %
RBALNP	1.72 %	-1.04	0.19 %	-0.82 %	0.78 %
RBALP	3.37 %	-0.83	0.19 %	-0.64 %	0.77 %

Cuadro 5: Resultados para $K = 8$ factores de PCA por estrategia para las estrategias con costo de transacción $\xi = 0,0002$ (i.e. 2 bps) y β calculado por OLS. *Sharpe ratio* (SR), retorno medio (\bar{R}_{II}), retorno excedente medio (μ), y volatilidad (σ) en términos anuales de los portafolios en un *back-test* desde el 2 de enero de 2014 al 18 de mayo de 2023.

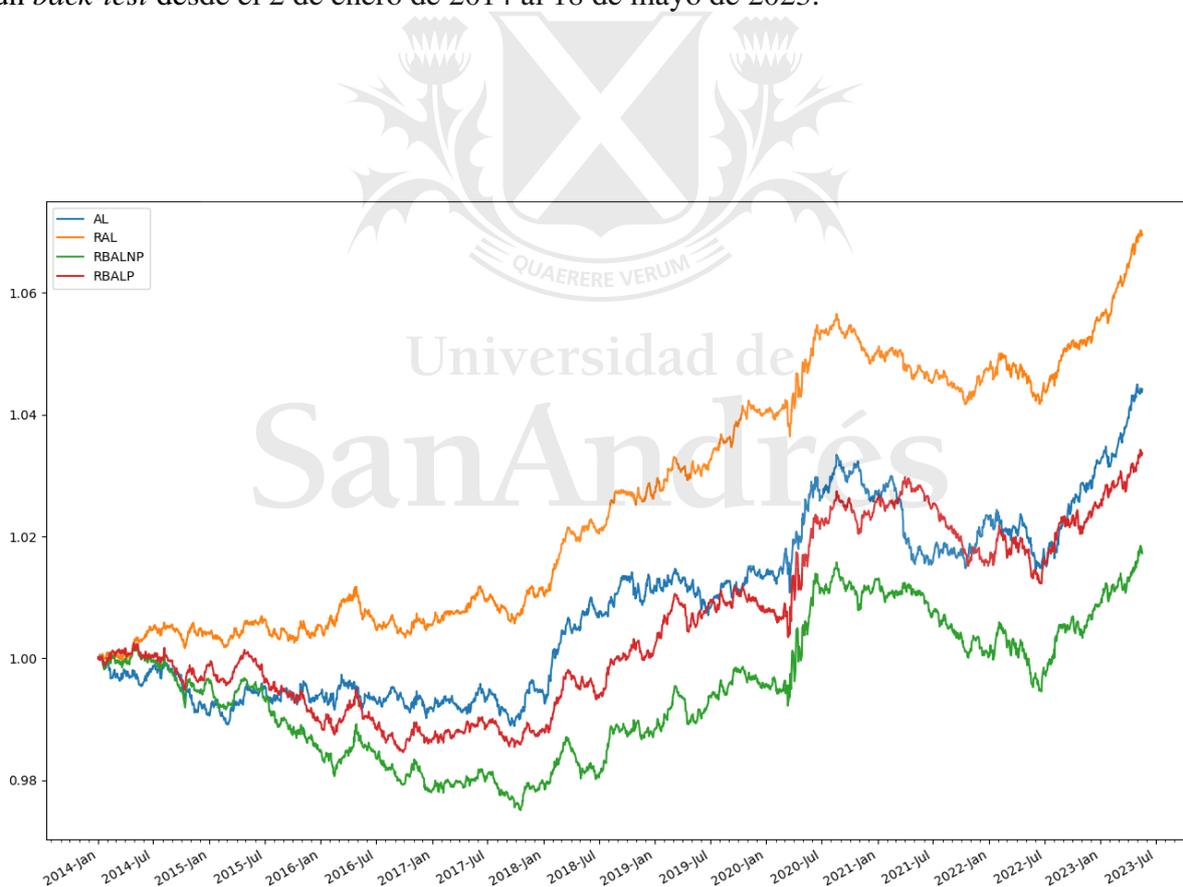


Figura 14: Retorno acumulado de los portafolios de cada estrategia con $K = 8$ factores de PCA, con costo de transacción $\xi = 0,0002$ (i.e. 2 bps) y β calculado por OLS, en un *back-test* desde el 2 de enero de 2014 al 18 de mayo de 2023.

ξ	Estrategia	Retorno Total	SR	\bar{R}_{Π}	μ	σ
0.0000	AL	25.80 %	0.82	2.47 %	1.47 %	1.78 %
	RAL	18.97 %	0.69	1.86 %	0.86 %	1.25 %
	RBALNP	26.89 %	1.10	2.55 %	1.55 %	1.42 %
	RBALP	27.43 %	1.13	2.60 %	1.60 %	1.41 %
0.0002	AL	14.20 %	0.24	1.43 %	0.43 %	1.78 %
	RAL	10.10 %	0.03	1.04 %	0.03 %	1.39 %
	RBALNP	12.59 %	0.20	1.28 %	0.27 %	1.39 %
	RBALP	12.88 %	0.22	1.30 %	0.30 %	1.37 %

Cuadro 6: Resultados por estrategia para $K = 8$ factores de PCA para costos de transacción (ξ) de 0 bps y 2 bps, y β estimados con OLSTF. *Sharpe ratio* (SR), retorno medio (\bar{R}_{Π}), retorno excedente medio (μ), y volatilidad (σ) en términos anuales de los portafolios para un *back-testing* desde el 2 de enero de 2014 al 18 de mayo de 2023.

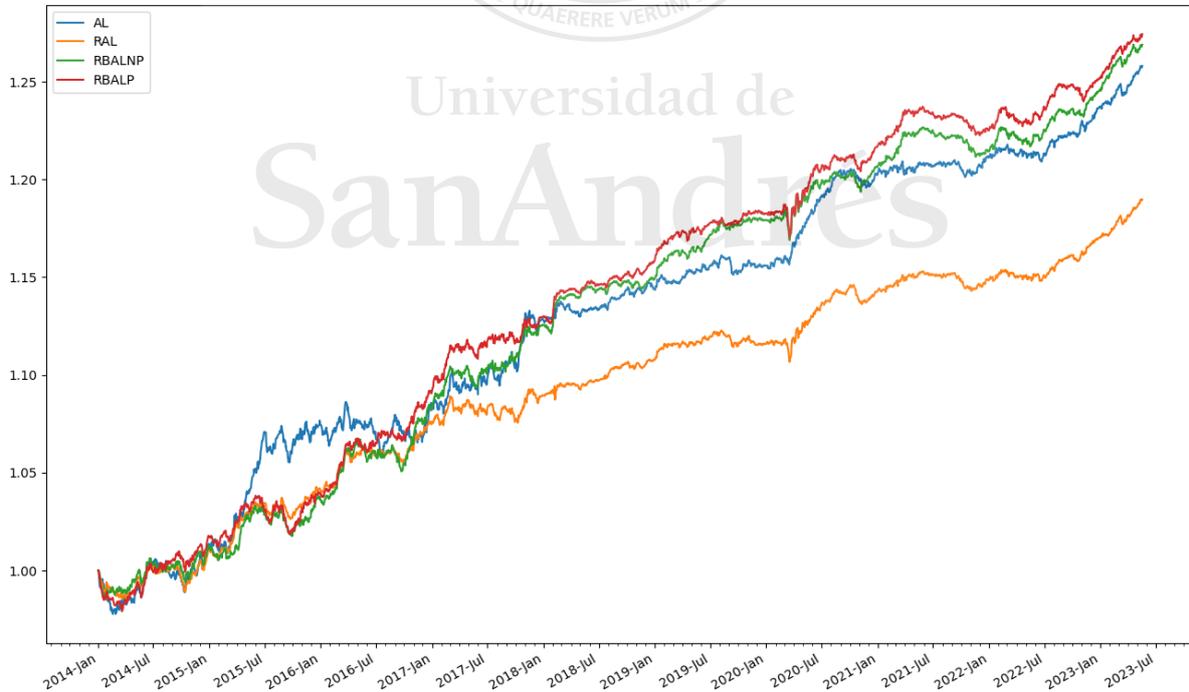


Figura 15: Retorno acumulado de los portafolios de cada estrategia con $K = 8$ factores de PCA, sin costo de transacción ($\xi = 0$) y β estimados con OLSTF, en un *back-test* desde el 2 de enero de 2014 al 18 de mayo de 2023.



Figura 16: Retorno acumulado de los portafolios de cada estrategia con $K = 8$ factores de PCA, con costo de transacción de 2 bps y β estimados con OLSTF, para un *back-testing* desde el 2 de enero de 2014 al 18 de mayo de 2023.

Otra alternativa que evaluamos fue la de utilizar métodos robustos para la regresión de series de tiempo, dado la conocida presencia de colas pesadas en los retornos diarios de las acciones (ver Fama 1965). Para esto se utiliza una regresión con la función de pérdida de Huber (*Huber loss*) de la ecuación 5.3 en lugar de la función de pérdida cuadrática que resulta muy sensible a la presencia de *outliers*. Los resultados obtenidos para $K = 8$ factores se reportan en el cuadro 7, donde observamos que el rendimiento de las estrategias es peor en términos de SR que con la regresión de series de tiempo por OLS. Más aún, las mejoras significativas en el SR observadas en el cuadro 4 respecto a AL no se obtienen en este caso, solo un incremento marginal en el caso de RBALP.

Por último en el cuadro 8 y la figura 18 se observan los resultados para un *back-test* con $N = 329$ componentes del S&P 500 desde el 2 de enero 2003 hasta el 30 de diciembre de 2011, para $K = 8$ factores de PCA. Se obtienen SR menores a los del *back-test* de 2014 a 2023, sin embargo esto puede deberse al menos parcialmente al menor número de activos en nuestra muestra. A su vez, no es exactamente el mismo período evaluado por Avellaneda y Lee (2010) ('97-'07) y Pelger et al. (2021) ('02-'16), ni tampoco la metodología es exactamente la misma.

Estrategia	Retorno Total	SR	\bar{R}_{II}	μ	σ
AL	15.30 %	0.67	1.52 %	0.52 %	0.77 %
RAL	14.01 %	0.64	1.40 %	0.40 %	0.63 %
RBALNP	13.16 %	0.45	1.32 %	0.32 %	0.71 %
RBALP	14.97 %	0.70	1.49 %	0.49 %	0.70 %

Cuadro 7: Resultados para $K = 8$ factores de PCA por estrategia para los modelos con regresión robusta de series de tiempo (*Huber loss*). *Sharpe ratio* (SR), retorno medio (\bar{R}_{II}), retorno excedente medio (μ), y volatilidad (σ) en términos anuales de los portafolios en un *back-test* desde el 2 de enero de 2014 al 18 de mayo de 2023. Sin costos de transacción, es decir $\xi = 0$, $\delta = 0$, y β estimados por OLS.

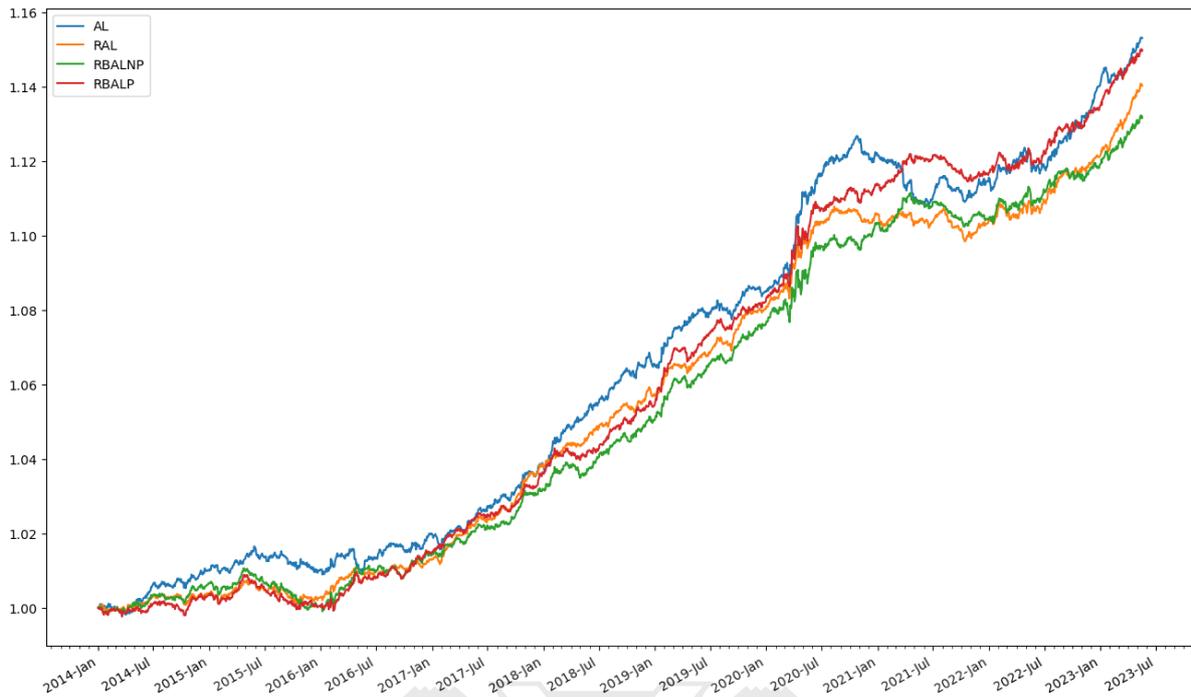


Figura 17: Retorno acumulado de los portafolios de cada estrategia con $K = 8$ factores de PCA, sin costo de transacción ($\xi = 0$), con regresión robusta de series de tiempo (*Huber loss*) y β estimados con OLS, para un *back-testing* desde el 2 de enero de 2014 al 18 de mayo de 2023.

Estrategia	Retorno Total	SR	\bar{R}_{II}	μ	σ
AL	20.91 %	0.13	2.12 %	0.15 %	1.12 %
RAL	27.05 %	0.87	2.67 %	0.70 %	0.80 %
RBALNP	25.51 %	0.62	2.53 %	0.56 %	0.92 %
RBALP	23.10 %	0.38	2.32 %	0.35 %	0.92 %

Cuadro 8: Resultados para $K = 8$ factores de PCA por estrategia. *Sharpe ratio* (SR), retorno medio (\bar{R}_{II}), retorno excedente medio (μ), y volatilidad (σ) en términos anuales de los portafolios para un *back-testing* desde el 2 de enero de 2003 al 30 de diciembre de 2011. Sin costos de transacción, es decir $\xi = 0$, $\delta = 0$, y β estimados por OLS.

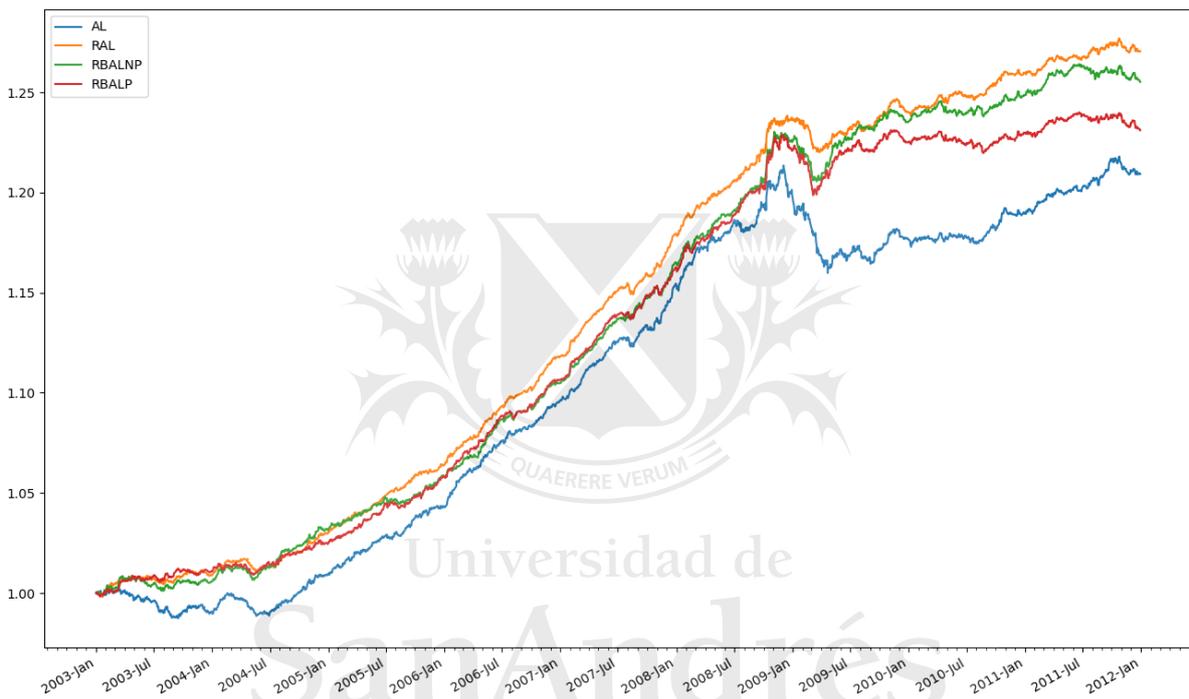


Figura 18: Retorno acumulado de los portafolios de cada estrategia en un *back-test* desde el 2 de enero de 2003 al 30 de diciembre de 2011. Se utilizan $K = 8$ factores de PCA, sin costo de transacción de γ y β estimados por OLS.

7. Conclusiones y nuevas líneas de investigación

Los resultados obtenidos en la sección 6, particularmente los cuadros 3 y 4, muestran que las nuevas estrategias introducidas basadas en *rollout* consiguen mejorar significativamente a la estrategia base de Avellaneda y Lee (2010), tanto en un entorno de simulación como en un *back-testing* con datos históricos. Esto muestra que esta técnica fundamental del aprendizaje reforzado puede ser utilizada para mejorar estrategias del arbitraje estadístico. Esto confirma la robustez del método *rollout* en la práctica como se afirma en Bertsekas (2022a). Más aún, se abre la puerta a investigar nuevas metodologías partiendo de distintos modelos del arbitraje estadístico como las de Pelger et al. (2021), u otros métodos más sofisticados del aprendizaje reforzado (e.g. *Multi-step Lookahead*, *Multi-agent Rollout*, *Approximation in Value Space*, *Deep Q-learning*, etc.).

En el caso de la evaluación con costos de transacción en el cuadro 6, no podemos concluir que las estrategias presentadas representen una mejora significativa respecto a la estrategia base de Avellaneda y Lee (con OLSTF), pero si obtenemos resultados similares. Observamos que aquí resultan fundamentales los costos de transacción para la *performance* de las estrategias. Sin embargo nuestro modelado de estos es bastante simplificado en este aspecto, lo que debe tenerse en cuenta a la hora de interpretar los resultados. Una posible mejora en esta dirección sería utilizando el método del filtro de Kalman como en Feng y Palomar (2016) en lugar del método de la ventana móvil, o los métodos recursivos de Tsagaris (2010) y Montana et al. (2009). También como se observa en Pelger et al. (2021) modelos de factores más sofisticados con menores errores de *pricing* pueden mejorar a estas estrategias de forma significativa, por ejemplo los modelos de Kelly et al. (2019), Kozak et al. (2020) o Pelger et al. (2023) en lugar de PCA (Jolliffe 2005).

El cuadro 7 nos muestra que un modelo robusto para la regresión de series de tiempo no mejora al desempeño de las estrategias. Al menos en la modificación básica con *Huber regression* los resultados son peores. Es posible sin embargo que los métodos robustos sean útiles en otra parte, por ejemplo en la regresión de PCA, donde podrían utilizarse los métodos de Maronna (1976) o Tyler (1987), o también otros métodos en la regresión de series de tiempo.

Por último, los resultados del cuadro 8 no nos permiten concluir que el rendimiento de estrategia de Avellaneda y Lee (2010) se haya deteriorado significativamente en los últimos

años. Diferencias en los activos utilizados, en las metodologías y en los períodos hacen que los resultados no sean directamente comparables a los de Avellaneda y Lee (2010) y Pelger et al. (2021).



Universidad de
San Andrés

Referencias

- [Ave13] Marco Avellaneda. «Hedge Funds 360». RiskMathics Workshop, 2013. URL: <https://math.nyu.edu/~avellane/HedgeFunds360.pdf>.
- [AL10] Marco Avellaneda y Jeong-Hyun Lee. «Statistical arbitrage in the US equities market». En: *Quantitative Finance* 10.7 (2010), págs. 761-782.
- [Ber12] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*. Vol. 1. Athena scientific, 2012.
- [Ber19] Dimitri Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.
- [Ber22a] Dimitri Bertsekas. *Lessons from AlphaZero for optimal, model predictive, and adaptive control*. Athena Scientific, 2022.
- [Ber22b] Dimitri Bertsekas. «Rollout Algorithms and Approximate Dynamic Programming for Bayesian Optimization and Sequential Estimation». En: *arXiv:2212.07998* (2022).
- [BT96] Dimitri Bertsekas y John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.
- [Bis06] Christopher M Bishop. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.
- [Boy+13] Stephen Boyd et al. «Performance bounds and suboptimal policies for multi-period investment». En: *Foundations and Trends® in Optimization* 1.1 (2013), págs. 1-72.
- [Boy+17] Stephen Boyd et al. «Multi-period trading via convex optimization». En: *Foundations and Trends® in Optimization* 3.1 (2017), págs. 1-76.
- [Coc09] John Cochrane. *Asset pricing: Revised edition*. Princeton university press, 2009.
- [DB16] Steven Diamond y Stephen Boyd. «CVXPY: A Python-embedded modeling language for convex optimization». En: *The Journal of Machine Learning Research* 17.1 (2016), págs. 2909-2913.
- [DCB13] Alexander Domahidi, Eric Chu y Stephen Boyd. «ECOS: An SOCP solver for embedded systems». En: *2013 European control conference (ECC)*. IEEE. 2013, págs. 3071-3076.

- [ET86] Bradley Efron y Robert Tibshirani. «Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy». En: *Statistical science* (1986), págs. 54-75.
- [ET94] Bradley Efron y Robert Tibshirani. *An introduction to the bootstrap*. Vol. 57. Monographs on Statistics and Applied Probability. Springer Science+Business Media, 1994.
- [Fam65] Eugene F Fama. «The behavior of stock-market prices». En: *The journal of Business* 38.1 (1965), págs. 34-105.
- [FF93] Eugene F Fama y Kenneth R French. «Common risk factors in the returns on stocks and bonds». En: *Journal of financial economics* 33.1 (1993), págs. 3-56.
- [FF15] Eugene F Fama y Kenneth R French. «A five-factor asset pricing model». En: *Journal of financial economics* 116.1 (2015), págs. 1-22.
- [FP16] Yiyong Feng y Daniel P Palomar. «A signal processing perspective on financial engineering». En: *Foundations and Trends® in Signal Processing* 9.1–2 (2016), págs. 1-231.
- [Hag21] Björn Hagströmer. «Bias in the effective bid-ask spread». En: *Journal of Financial Economics* 142.1 (2021), págs. 314-337.
- [HTF09] Trevor Hastie, Robert Tibshirani y Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.
- [Jol05] Ian Jolliffe. «Principal component analysis». En: *Encyclopedia of statistics in behavioral science* (2005).
- [KPS19] Bryan T Kelly, Seth Pruitt y Yinan Su. «Characteristics are covariances: A unified model of risk and return». En: *Journal of Financial Economics* 134.3 (2019), págs. 501-524.
- [KNS20] Serhiy Kozak, Stefan Nagel y Shrihari Santosh. «Shrinking the cross-section». En: *Journal of Financial Economics* 135.2 (2020), págs. 271-292.
- [Mar76] Ricardo Antonio Maronna. «Robust M-estimators of multivariate location and scatter». En: *The annals of statistics* (1976), págs. 51-67.

- [MTT09] Giovanni Montana, Kostas Triantafyllopoulos y Theodoros Tsagaris. «Flexible least squares for temporal data mining and statistical arbitrage». En: *Expert Systems with Applications* 36.2 (2009), págs. 2819-2830.
- [PCZ23] Markus Pelger, Luyang Chen y Jason Zhu. «Deep learning in asset pricing». En: *Management Science* (2023).
- [PGZ21] Markus Pelger, Jorge Guijarro Ordonez y Greg Zanotti. «Deep learning statistical arbitrage». En: *arXiv preprint arXiv:2106.04028* (2021).
- [PX22] Markus Pelger y Ruoxuan Xiong. «Interpretable sparse proximate factors for large dimensions». En: *Journal of Business & Economic Statistics* 40.4 (2022), págs. 1642-1664.
- [Shr04] Steven E Shreve. *Stochastic calculus for finance II: Continuous-time models*. Vol. 11. Springer, 2004.
- [SB18] Richard S Sutton y Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [Tsa10] Theodoros Tsagaris. «Adaptive regression methods with application to streaming financial data». Tesis doct. Imperial College London, 2010.
- [Ty187] David E Tyler. «A distribution-free M-estimator of multivariate scatter». En: *The annals of Statistics* (1987), págs. 234-251.
- [YP17] Joongyeub Yeo y George Papanicolaou. «Risk control of mean-reversion time in statistical arbitrage». En: *Risk and Decision Analysis* 6.4 (2017), págs. 263-290.
- [ZHT06] Hui Zou, Trevor Hastie y Robert Tibshirani. «Sparse principal component analysis». En: *Journal of computational and graphical statistics* 15.2 (2006), págs. 265-286.

A. Demostraciones

Demostración de la proposición 3.2.3. Probamos esto por inducción, para $k = N$ es trivial pues $J_N^{\tilde{\mu}} = J_N^{\mu} = g_N$. Supongamos que vale para $k + 1$, entonces

$$J_k^{\tilde{\mu}}(x_k) = \mathbb{E} \left[g(x_k, \tilde{\mu}_k(x_k), w_k) + J_{k+1}^{\tilde{\mu}}(x_{k+1}) \right] \leq \mathbb{E} \left[g(x_k, \tilde{\mu}_k(x_k), w_k) + J_{k+1}^{\mu}(x_{k+1}) \right] = \\ \min_{u_k \in U_k(x_k)} \mathbb{E} \left[g(x_k, u_k, w_k) + J_{k+1}^{\mu}(x_{k+1}) \right] \leq \mathbb{E} \left[g(x_k, \mu_k(x_k), w_k) + J_{k+1}^{\mu}(x_{k+1}) \right] = J_k^{\mu}(x_k),$$

donde la primera igualdad es la ecuación de DP para el *rollout*, luego usamos la hipótesis inductiva, posteriormente usamos la definición del *rollout*, y en la última desigualdad el hecho de que $\mu_k(x_k)$ es un control en $U_k(x_k)$. \square

Demostración de la proposición 3.3.1. Se define $f(t, x) = e^{\kappa t} x$ y se calcula $df(t, X(t))$ utilizando el Lema de Itô (ver Shreve 2004, Teorema 4.4.1):

$$df(t, X(t)) = f_t(t, X(t)) dt + f_x(t, X(t)) dX(t) + \frac{1}{2} f_{xx}(t, X(t)) dX(t) dX(t) = \\ \kappa e^{\kappa t} X(t) dt + e^{\kappa t} (\kappa(\mu - X(t)) dt + \sigma dW(t)) = e^{\kappa t} \kappa \mu dt + e^{\kappa t} dW(t).$$

Integrando esta ecuación en t llegamos a que:

$$e^{\kappa t} X(t) - X(0) = \int_0^t e^{\kappa s} \kappa \mu ds + \sigma \int_0^t e^{\kappa s} dW(s) = (e^{\kappa t} - 1)\mu + \sigma \int_0^t e^{\kappa s} dW(s).$$

Finalmente, reordenando términos obtenemos la ecuación 3.30:

$$X(t) = X(0)e^{-\kappa t} + \mu(1 - e^{-\kappa t}) + \sigma e^{-\kappa t} \int_0^t e^{\kappa s} dW(s).$$

Para la última afirmación consideramos $I(t) = \int_0^t e^{\kappa s} dW(s)$. Se sabe que $I(t)$ tiene distribución normal ya que $e^{\kappa t}$ es determinística y tiene media cero dado que es una martingala. Más aún, su varianza se puede calcular por la Isometría de Itô (ver Shreve 2004, Teorema 4.3.1):

$$\text{var}[I(t)] = \mathbb{E}[I^2(t)] = \int_0^t e^{2\kappa s} ds = \frac{1}{2\kappa} (e^{2\kappa t} - 1).$$

Por lo tanto $\text{var}[X(t)] = \frac{\sigma^2(1 - e^{-2\kappa t})}{2\kappa}$. \square