



Introducción al aprendizaje de máquinas

BCRP – CEFA 2025

Objetivos

OBJETIVO: Introducir el enfoque del 'aprendizaje de máquinas' y ejemplificar aplicaciones de utilidad en la gestión de portafolios.

OBJETIVOS ESPECÍFICOS:

1. Introducir el enfoque de '*Statistical Learning*' para analizar algoritmos de aprendizaje de máquinas.
2. Mostrar ejemplos de algoritmos de aprendizaje de máquinas supervisados en finanzas.
3. Mostrar ejemplos de algoritmos de aprendizaje de máquinas no supervisados en finanzas.

Machine Learning

El campo del aprendizaje de máquinas (ML) es una rama de la inteligencia artificial (AI) enfocada en posibilitar que los computadores y las máquinas aprendan a resolver algún problema sin la necesidad de una instrucción directa.

Ejemplo: Un sistema de clasificación de correos que diferencie entre correos spam y correos no spam.

Más específicamente, para el trabajo que se realizará durante el curso nos centraremos en el campo del 'Statistical Machine Learning' (Trebor, Witten, Hastie, Tibshirani, Taylor, 2023).

Statistical Learning

Particularmente contamos con:

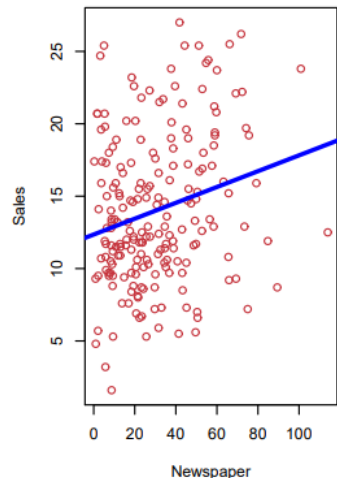
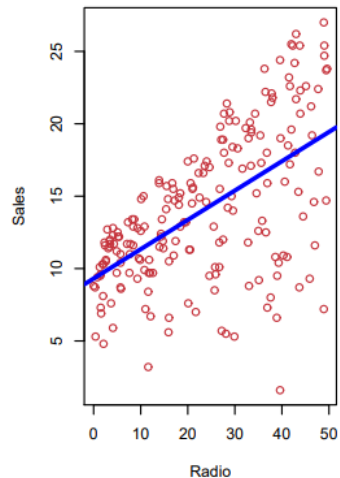
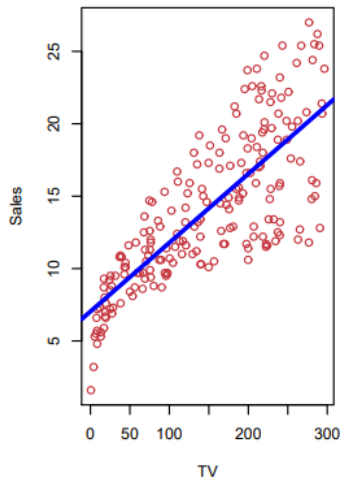
- A) Un set de variables de entrada al cual le podemos denotar con X
- B) Una variable de respuesta Y .

La relación entre ambas variables puede entonces enunciarse de manera general como:

$$Y = f(X) + \epsilon$$

Donde ϵ es un término aleatorio asociada a variables no observadas. Y f es la información que sistemáticamente aporta X sobre Y .

'Statistical Learning' se refiere entonces a un conjunto de aproximaciones para la estimación de la función f .



Statistical Learning

Siendo que f no es posible conocerla directamente, podemos estimar una función \hat{f} para aproximarnos a la f real.

$$\hat{Y} = \hat{f}(X)$$

Definimos entonces el error esperado del modelo aplicando la siguiente expectativa:

$$E \left(Y - \hat{f}(X) \right)^2$$

Este error total lo podríamos descomponer en: el error reducible y el error irreducible. El error reducible está asociado al hecho de que potencialmente podríamos hacer a \hat{f} lo más parecido a f .

El error irreducible lo podemos asociar al término aleatorio ϵ que está asociado al carácter aleatorio de los datos.



Error de Predicción

Podemos entonces descomponer el error total de nuestro modelo de la siguiente manera, por practicidad denotemos $\hat{f} = \hat{f}(x)$

$$\begin{aligned} E(Y - \hat{f})^2 &= E[Y - f + f - \hat{f}]^2 \\ &= E(Y - \hat{f})^2 + E(f - \hat{f})^2 + 2E[(Y - f)(f - \hat{f})] \\ &= E(Y - \hat{f})^2 + E(f - \hat{f})^2 + 2E[fY - f^2 - \hat{f}Y + \hat{f}f] \end{aligned}$$

Tomando en cuenta que $E[f] = f$ y $E[Y] = f$

$$\begin{aligned} E[\epsilon]^2 + E[f - \hat{f}]^2 + 2(f^2 - f^2 - \hat{f}E[\hat{f}] + \hat{f}E[\hat{f}]) \\ = \sigma_{\epsilon} + E[(f^2 - \hat{f}^2 - \hat{f}E[\hat{f}] + \hat{f}E[\hat{f}])] \\ = \sigma_{\epsilon} + E[(f - \hat{f})^2] \end{aligned}$$

Podemos entonces asociar un término reducible y otro irreducible:

$$\begin{aligned} &= E[(f - \hat{f})^2] + \sigma_{\epsilon}^2 \\ &= \text{error reducible} + \text{error irreducible} \end{aligned}$$

En general, si el problema es de predicción habríamos de considerar el algoritmo como una "caja negra", buscando minimizar el error reducible y consecuentemente el error total.

Modelos de Predicción

Un aspecto importante para entender la diferencia entre *error reducible* y *error irreducible* es que es que la predicción de \hat{Y} es una variable aleatoria. Esto debido a que la función \hat{f} que estamos encontrando la hallamos a partir de una función (o de forma más general de un algoritmo) ejecutado con variables aleatorias. Pensemos en el ejemplo de la regresión lineal, nosotros queremos estimar la función $Y = f(X) + \epsilon = X\beta + \epsilon$ a partir de las variables aleatorias. El estimador de β va a ser:

$$\hat{\beta} = (X'X)^{-1}(X'Y)$$

El cuál va a depender de una serie de variables aleatorias que usemos para la estimación, si cambiamos el conjunto de datos

$\{X_n, Y_n\}$, $\{X_m, Y_m\}$, $\{X_k, Y_k\}$ por un set distinto en cada oportunidad obtendremos una función de f distinta en cada oportunidad

Conjunto de datos	Función estimada
$\{X_n, Y_n\}$	$\hat{\beta}_n = (X_n'X_n)^{-1}(X_n'Y_n)$
$\{X_m, Y_m\}$	$\hat{\beta}_m = (X_m'X_m)^{-1}(X_m'Y_m)$
$\{X_k, Y_k\}$	$\hat{\beta}_k = (X_k'X_k)^{-1}(X_k'Y_k)$

Sesgo - Varianza

Podemos adicionalmente descomponer este término de error reducible entre sesgo y varianza.

$$E(Y - \hat{f})^2 = E(f - \hat{f}) + \sigma_\epsilon^2$$

$$E(f - \hat{f})^2 = E(f + E[\hat{f}] - E[\hat{f}] + \hat{f}) + \sigma_\epsilon^2$$

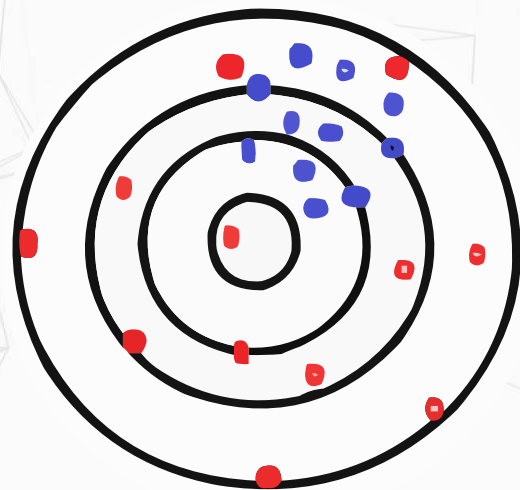
$$E(f - \hat{f})^2 = E(f + E[\hat{f}])^2 + E(f + E[\hat{f}])^2 + \sigma_\epsilon^2$$

De manera que obtenemos:

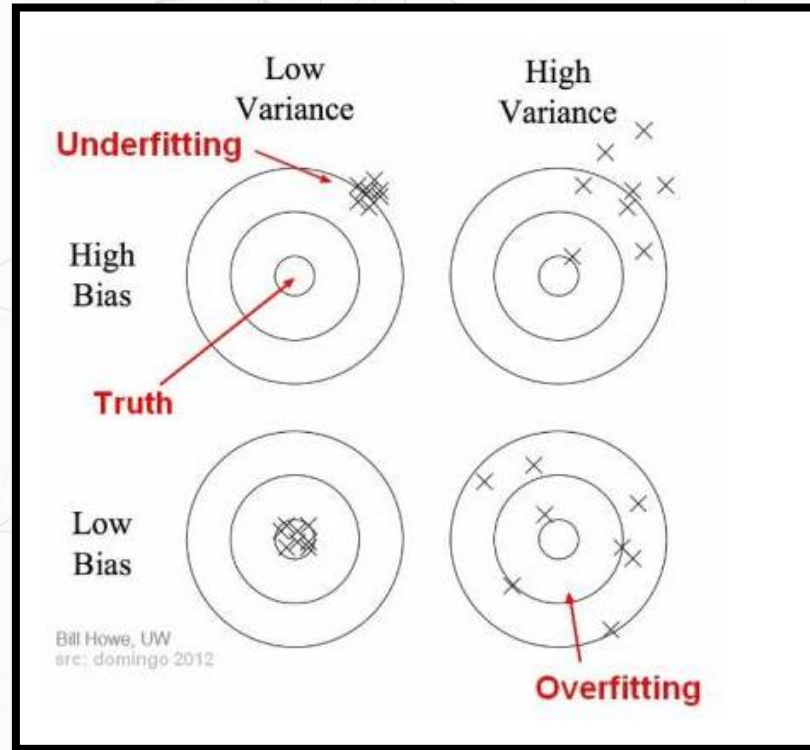
$$E(f - \hat{f})^2 = Bias^2(\hat{f}) + Var(\hat{f}) + \sigma_\epsilon^2$$

La varianza entonces va a ser que tanta dispersión tiene f respecto a su valor esperado. En otras palabras, que tan sensible es el algoritmo al set de variables de entrada.

Y el sesgo es que tanta diferencia existe entre la función correcta y el valor esperado de f .

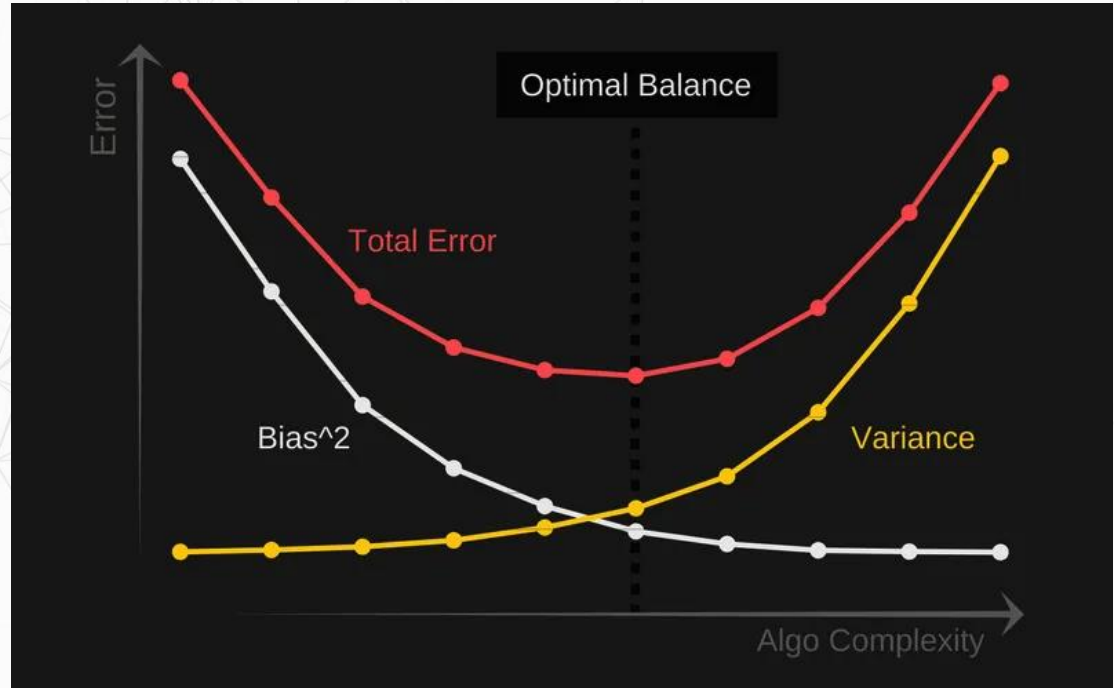


Sesgo - Varianza



Sesgo - Varianza

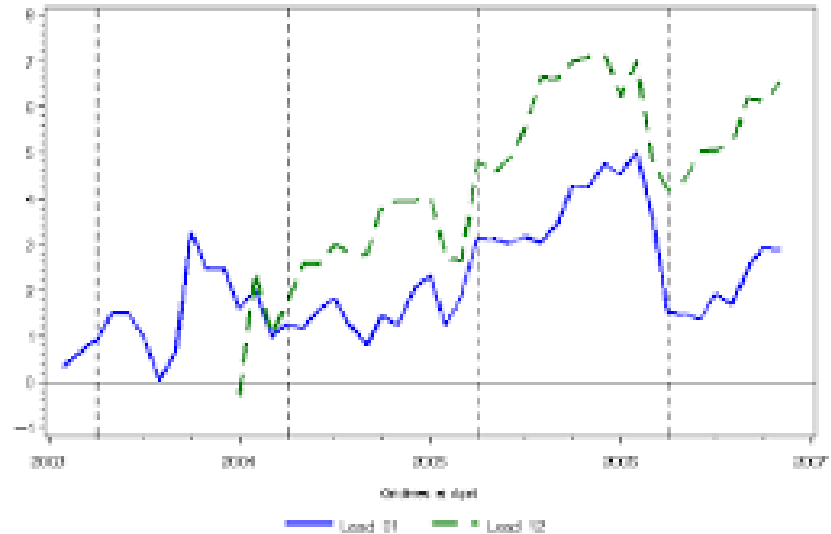
En general, si el problema es de predicción habríamos de considerar el algoritmo como una “caja negra”, buscando minimizar el error reducible y el error total.



Objetivo modelos de ML

Son pocos los casos donde es posible construir de manera apropiada las curvas de sesgos contra varianza. Típicamente las curvas que se encuentran en los libros se realizan con datos simulados. Por lo que lo usual es querer “encontrar” el error cuadrático medio out-of-sample, a manera de error cuadrático medio con el cuál nos toparíamos en la práctica.

Differences of the Sum of Squared Forecast Errors
MRTR Sales (p0b4801) Weekday-Weekend - MRTR Sales (p0b4801) No Error



Métodos de validación

Podemos entender los “métodos de validación” como aquellos métodos que nos deben permitir “estimar” (encontrar) el error cuadrático medio fuera de la muestra. En líneas generales existen las siguientes tres métodos de validación:

1. Conjunto de validación.
2. Validación cruzada de un elemento.
3. Validación cruzada.

Métodos de validación

1. Conjunto de validación. Esta metodología es bastante usual encontrarla en 'papers' donde se examina el desempeño de una estrategia, especialmente por la importancia de la continuidad de los datos en una serie de tiempo. Consiste en seccionar los datos entre datos de entrenamiento, también conocidos como *in - sample* y datos de prueba *out - of - sample*. Por lo general se usan proporciones de 60-40 o 50-50 para este propósito.

Suponiendo que m es el número de datos fuera de la muestra calcularíamos la métrica con la siguiente ecuación.

$$CV_{(m)} = \frac{1}{m} \sum_{i=1}^m MSE_i$$

Métodos de validación

- 2. *Validación cruzada de un elemento (LOOCV)*. Resuelve el problema n particiones del set de entrenamiento, donde en cada oportunidad se deja por fuera un dato, de esta manera acabamos contando con n conjuntos.

Para cada subconjunto usamos una predicción de Y que nos sirva para encontrar el MSE error como:

$$MSE_i = (y_i - \hat{y}_i)^2$$

Suponiendo que n es el número de datos fuera de la muestra calcularíamos la métrica con la siguiente ecuación.

$$CV_{(n)} = \frac{1}{N} \sum_{i=1}^N MSE_i$$

Métodos de validación

3. Validación cruzada. Consiste básicamente en los siguientes pasos.
 1. Dividir las observaciones en k grupos u observaciones (por ejemplo $k=5$) de aproximadamente el mismo tamaño.
 2. Usar los datos de pertenecientes a los $k-1$ datos para entrenar el algoritmo, testear el algoritmo en el k -esimo dato. $MSE_i = \sum_{j=1}^{m_k} MSE_j$
 3. Para cada subconjunto computar el “estimador” del error cuadrático medio con la siguiente función

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^k MSE_i$$

Problemas de la ciencia de datos

Tipo de Algoritmo de Aprendizaje Automático (ML)

Variables	Supervisado (Variable Objetivo)	No Supervisado (Sin variable Objetivo)
Continuas	Regresión <ul style="list-style-type: none">• Lineal; Regresión Penalizada/LASSO• Logística• Árbol de Clasificación y Regresión (CART)• Bosque Aleatorio (Random Forest)	Reducción de Dimensionalidad <ul style="list-style-type: none">• Análisis de Componentes Principales (PCA) Agrupamiento (Clustering) <ul style="list-style-type: none">• K-Medias (K-Means)• Jerárquico
Categorías	Clasificación <ul style="list-style-type: none">• Logística• Máquinas de Vectores de Soporte (SVM)• K- Vecinos más Cercanos (K-Nearest Neighbor, KNN)• Árbol de Clasificación y Regresión (CART)	Reducción de Dimensionalidad <ul style="list-style-type: none">• Análisis de Componentes Principales (PCA) Agrupamiento (Clustering) <ul style="list-style-type: none">• K-Medias (K-Means)• Jerárquico
Continuas o Categoricals	<ul style="list-style-type: none">• Redes Neuronales• Aprendizaje Profundo (Deep Learning)• Aprendizaje por Refuerzo (Reinforcement Learning)	<ul style="list-style-type: none">• Redes Neuronales• Aprendizaje Profundo (Deep Learning)• Aprendizaje por Refuerzo (Reinforcement Learning)

Referencias: CFA Institute. (2024). *Quantitative methods*. In *CFA Program Curriculum 2024: Level II (Vol. 1)*. CFA Institute.



Aprendizaje Supervisado: Modelo de Nelson y Siegel

Nelson & Siegel

Nelson & Siegel propusieron descomponer la curva de rendimiento en tres factores:

- nivel (movimiento paralelo)
- pendiente (diferencia de tasas entre plazos)
- curvatura (cambio de inflexión)

Utilizando estos tres factores se puede explicar de manera precisa la forma de la curva. De esta manera en vez de proyectar cada nodo de la curva de rendimiento se proyectarían solamente tres factores y luego se reconstruiría la curva a partir de estos valores.



Descomposición por factores de carga de Nelson & Siegel

La función que describe la curva de rendimientos es la siguiente:

$$Y(t, \tau_i) = \beta_0 + \beta_1 \left(1 - \frac{\exp(-\lambda \tau_i)}{\lambda \tau_i} \right) + \beta_2 \left(\frac{1 - \exp(-\lambda \tau_i)}{\lambda \tau_i} - \exp(-\lambda \tau_i) \right)$$

$Y(t, \tau_i)$ es el rendimiento (o tasa) del nodo τ_i de la curva en la fecha t , los β_i (betas) son los coeficientes de los factores de carga y λ es la tasa de decaimiento.

Dependiendo de los valores de los β_i (betas) la curva puede adoptar formas monótonas, jorobadas o, incluso, con puntos de inflexión (en forma de "S").

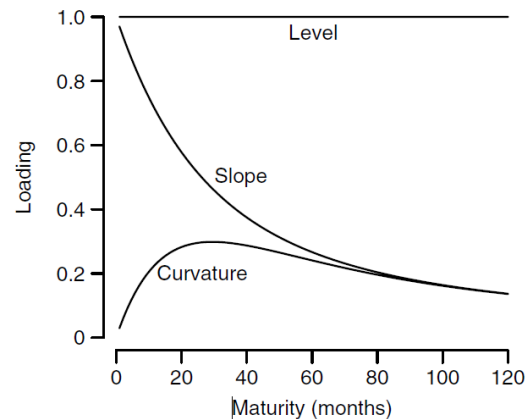


Figure 2.1. DNS Factor Loadings. We plot DNS factor loadings as a function of maturity, for $\lambda = 0.0609$.

Estimación de modelos de Nelson-Siegel

Hay dos metodologías para la estimación del modelo de Nelson y Siegel.

Metodología 1

- Definir una función de perdida como lo puede ser el error cuadrático medio (ECM).
- Usar algún tipo de optimizador para asegurar que se esté encontrando el set de parámetros $(\beta_0, \beta_1, \beta_2, \lambda)$ que minimiza el ECM. Particularmente vamos a usar el downhill algorithm .

Estimación de modelos de Nelson-Siegel

Metodología 2

- Definir una grilla para valores de λ
- Para cada grilla de valores de λ , encontrar los parámetros de $\beta_0, \beta_1, \beta_2$ utilizando mínimos cuadrados ordinarios.
- Seleccionar el valor de λ que tuvo el menor error cuadrático medio.



Aprendizaje Supervisado: Análisis Técnico

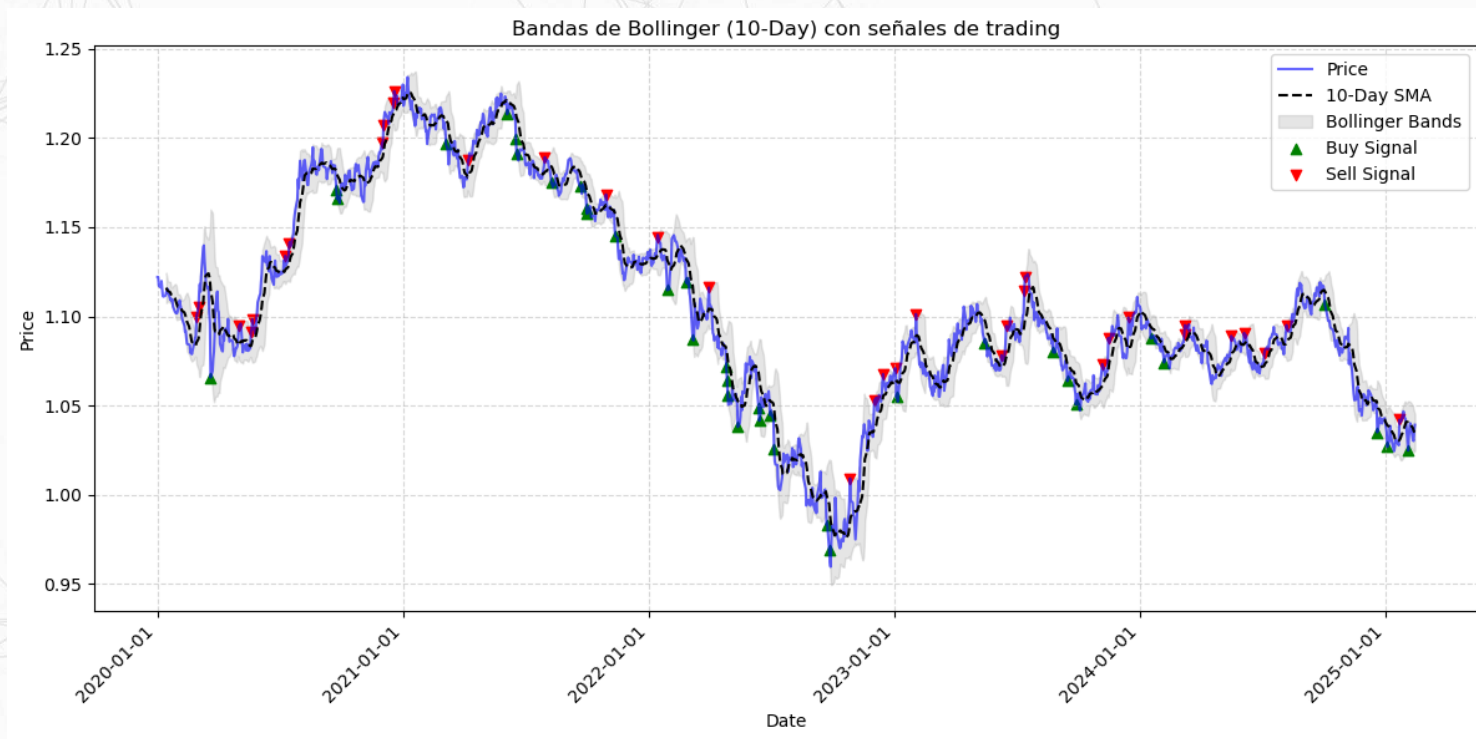
Implicaciones de la hipótesis del mercado eficiente

Sobre el análisis técnico

El análisis técnico esencialmente se trata de la búsqueda de patrones recurrentes y predecibles en el precio de las acciones. Aunque sus usuarios reconocen el valor de la información sobre los prospectos económicos, ellos creen que no es necesaria para desarrollar una estrategia exitosa.



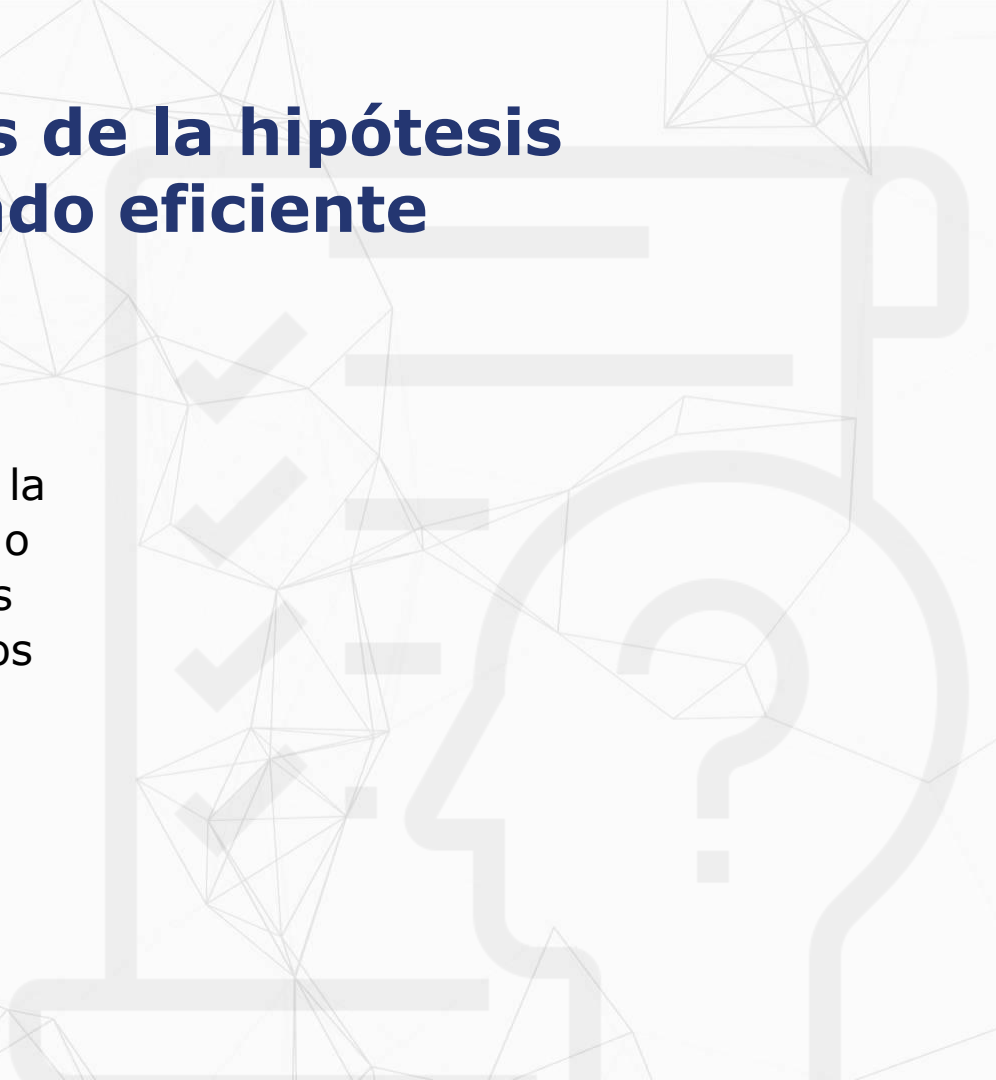
Bandas de Bollinger



Implicaciones de la hipótesis del mercado eficiente

Sobre el análisis técnico

Un ejercicio mental permite debatir la noción de los niveles de resistencia o los niveles de soporte. Estos valores se supone que son precios de activos financieros que la acción no podría exceder (en caso de que sean soportes superiores) o estar por debajo (en caso de ser soportes inferiores).



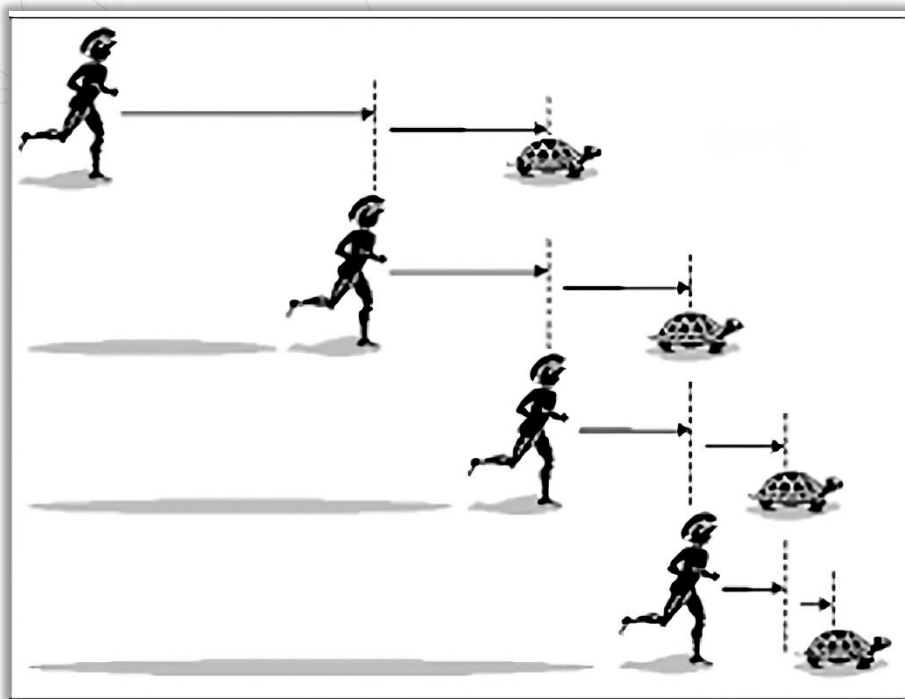
Análisis Técnico: Consideraciones

El ejercicio hipotético que se podría trabajar es el siguiente:

Suponga que se sabe que el precio de una acción XYZ efectivamente tiene un precio de resistencia en USD 72.

En 71.50 nadie estaría dispuesto a comprarlo en 71.50, porque entonces 71.50 pasaría a ser el nuevo nivel de resistencia.

En 71 nadie estaría dispuesto a comprarlo a 71, por qué no habría ningún espacio posible a que suba el precio entonces 71 se convertiría en el nuevo nivel de resistencia.



Hipótesis del mercado eficiente

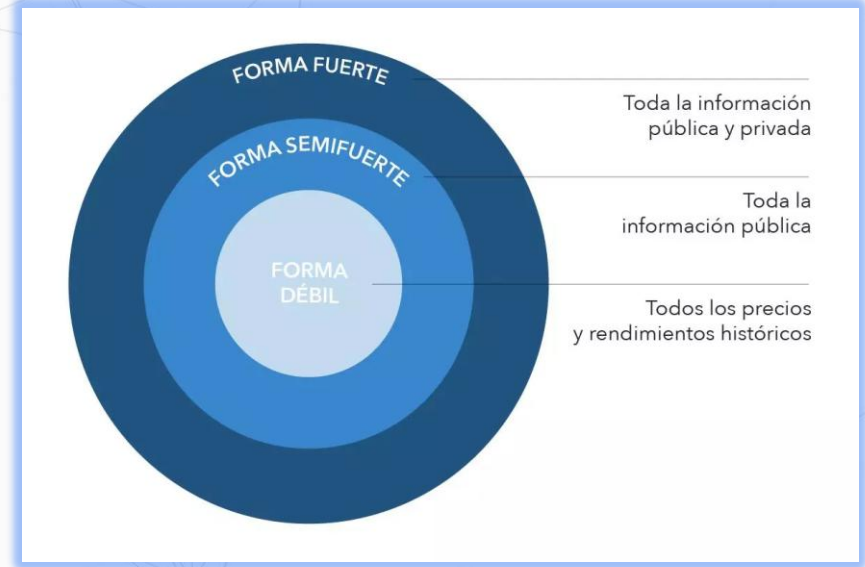
De forma más general Eugene Fama formula la hipótesis del mercado eficiente, la cuál se puede resumir en los siguientes dos postulados:

1. Cualquier información que potencialmente puede ser usada para predecir el rendimiento del activo debería estar ya reflejada en el precio.
2. El cambio en el precio de los activos (el retorno de los activos) solo responde a la nueva información. Esta nueva información, por definición, debe ser impredecible puesto que de otra manera ya sería parte de la información de hoy. Esto implica que el cambio en el precio de los activos es entonces **impredecible**.

Formas de Eficiencia del Mercado

La hipótesis del mercado eficiente considera los siguientes tres escenarios:

1. **Forma Débil.** Se refleja toda la información histórica sobre la firma o el activo. Bajo esta forma de eficiencia del mercado, no tiene validez el análisis técnico.
2. **Forma Semifuerte.** Toda la información pública de los prospectos de una firma debería verse reflejado en el precio del activo. Bajo esta forma de eficiencia del mercado, no tiene validez el análisis fundamental.

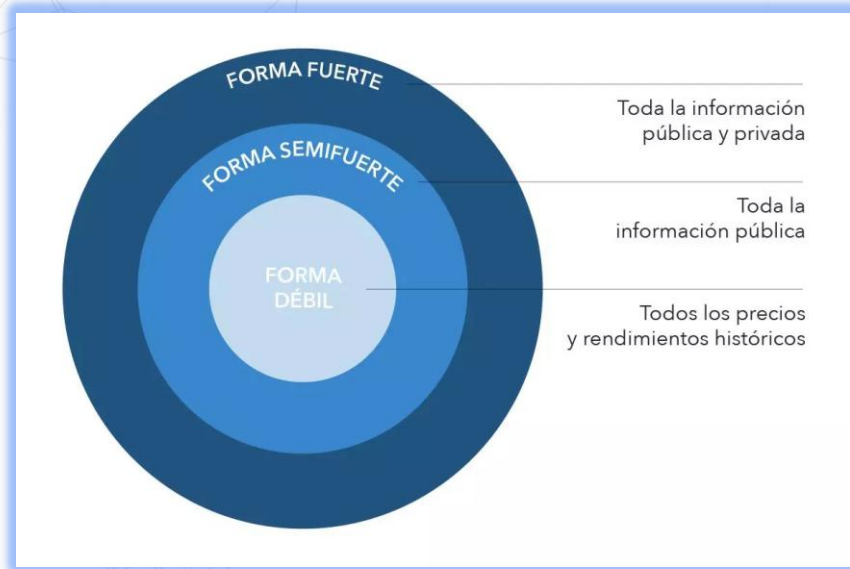


Formas de Eficiencia del Mercado

La hipótesis del mercado eficiente considera los siguientes tres escenarios:

3. Forma Fuerte. La hipótesis del mercado eficiente en su forma fuerte establece que el precio de los activos refleja toda la información relevante sobre la firma, incluyendo la información que esta solamente disponible para los 'insiders'.

Esta versión de la hipótesis del mercado eficiente es un tanto 'extrema'. No obstante, buena parte de la misión de la U.S. Security Exchange Commission se dedica exclusivamente a cumplir este objetivo.

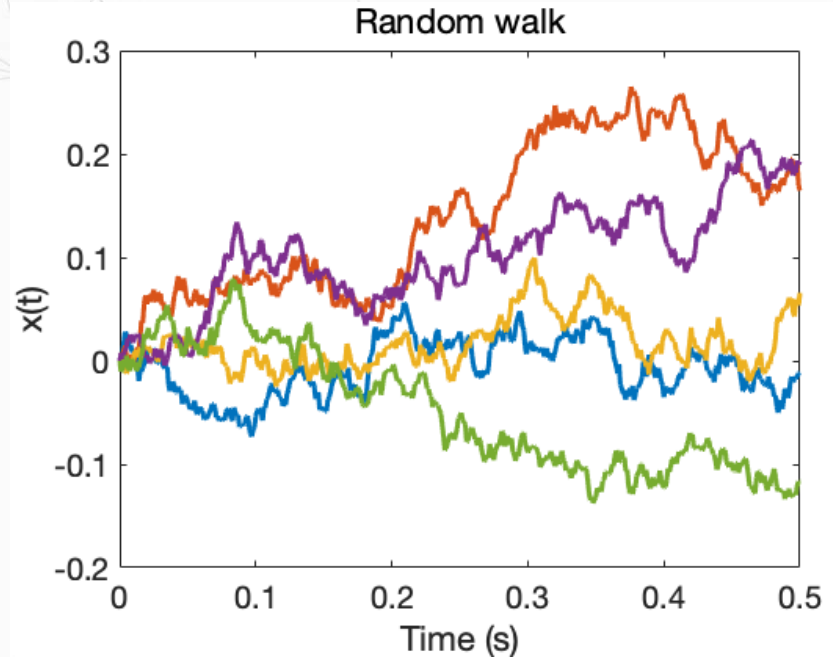


La hipótesis del mercado eficiente

Estas dos conclusiones son la esencia del argumento de que el precio de los activos sigue una **caminata aleatoria**, o lo que es lo mismo, el cambio en el precio de los activos debe ser aleatorio e impredecible.

$$X_t = X_{t-1} + \epsilon$$

Como podemos ver el valor del precio del activo solo depende de su estado actual.





Aprendizaje no Supervisado: Hierarchical Risk Parity

Aprendizaje no supervisado: HRP

Uno de los principales ejemplos de un caso de un algoritmo de clasificación en finanzas es el del 'Hierarchical Risk Parity'.

A continuación, expondremos el funcionamiento de este algoritmo.



Aprendizaje no supervisado: Clasificación

¿Qué problema estaba queriendo resolver el algoritmo de 'Hierarchical Risk Parity'?

La técnica de Risk Parity parte de ubicar defectos del enfoque de media-varianza, los cuáles son principalmente dos:

(1) Los portafolios óptimos obtenidos al resolver el problema de Media-Varianza llegan a ser muy inestables en el tiempo

(2) La estimación de retornos esperados es un ejercicio difícil y se puede demostrar que tiene un error estándar alto.

También se puede comentar que tiende a ser muy propenso a soluciones de esquina.



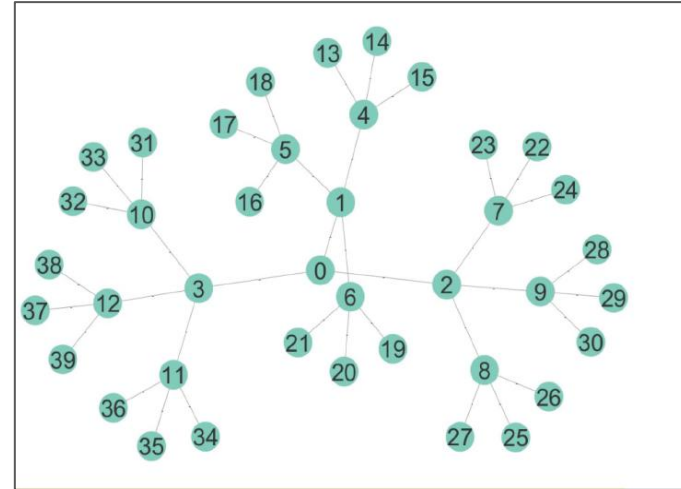
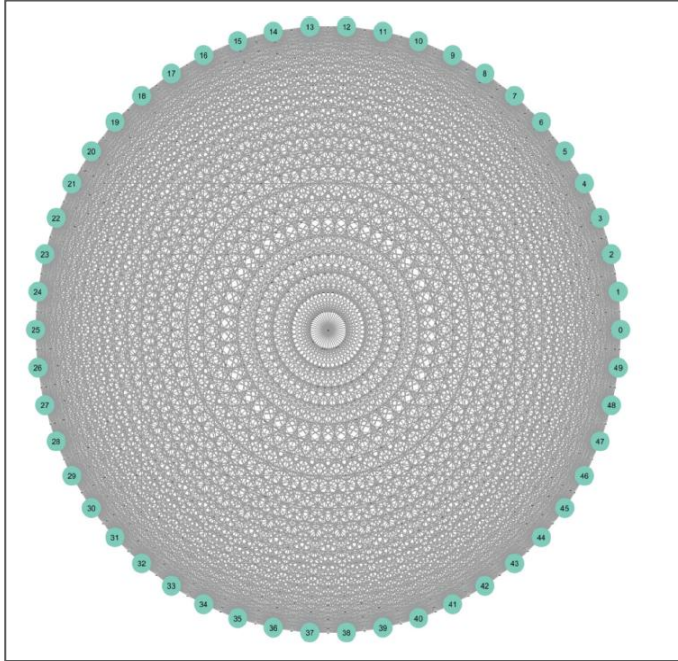
Markowitz curse

Markowitz curse: Entre más correlacionados los activos en el universo de inversión, la mayor necesidad de diversificación, y aun así la mayor probabilidad de recibir soluciones inestables. Los beneficios de la diversificación estarían más que superados por el error de especificación puesto que se requiere estimar más términos de covarianzas.

Una manera un poco más directa asimilar este fenómeno es calcular el determinante de una matriz de activos financieros nos damos cuenta de que es muy cercano a cero.

La solución analítica del portafolio de mínima varianza tiene la siguiente forma: $w = \sigma_p^2 \Sigma^{-1} \mathbf{1}$.
Depende de Σ^{-1} por ende si el determinante de sigma es inestable también lo son las soluciones al problema.

Aprendizaje no supervisado: Hierarchical Risk Parity



Paso 1: Tree Clustering

Paso 1: Tree Clustering: Forma un dendograma para los activos de la optimización.

Paso 2: Quasi-Diagonalización. Reordenar la matriz de covarianzas.

Paso 3: Bisección Recursiva. Optimizar la matriz iterativamente.



Paso 1: Tree Clustering

El primer paso consiste en calcular la matriz de distancias, para ello primero se computa:

$$d_{ij} = d[X_i, X_j] = \sqrt{\frac{1}{2}(1 - \rho_{ij})}$$

Posteriormente se computa la siguiente métrica:

$$\tilde{d}_{ij} = d[D_i, D_j] = \sqrt{\sum_{n=1}^N (d_{n,i} - d_{n,j})^2}$$

Esta métrica de distancia ya considera toda la matriz al iterar sobre los n activos.



Paso 1: Tree Clustering

En este caso los dos nodos con menor distancia entre ellos (17) son a y b, por lo que pasaran a formar el primer cluster.

	a	b	c	d	e
a	0	17	21	31	23
b	17	0	30	34	21
c	21	30	0	28	39
d	31	34	28	0	43
e	23	21	39	43	0

Paso 1: Tree Clustering

Cuál sería entonces la manera de computar la distancia entre el 'Cluster' formado y el resto de los nodos para ello hay varias opciones en las librerías de 'machine learning', entre ellas por ejemplo el 'nearest point algorithm':

$$\bar{d}_{A,B} = \min[\{\widetilde{d}_{ij}\}_{i \in A, j}]$$

	(a,b)	c	d	e
(a,b)	0	21	31	21
c	21	0	28	39
d	31	28	0	43
e	21	39	43	0

Paso 1: Tree Clustering

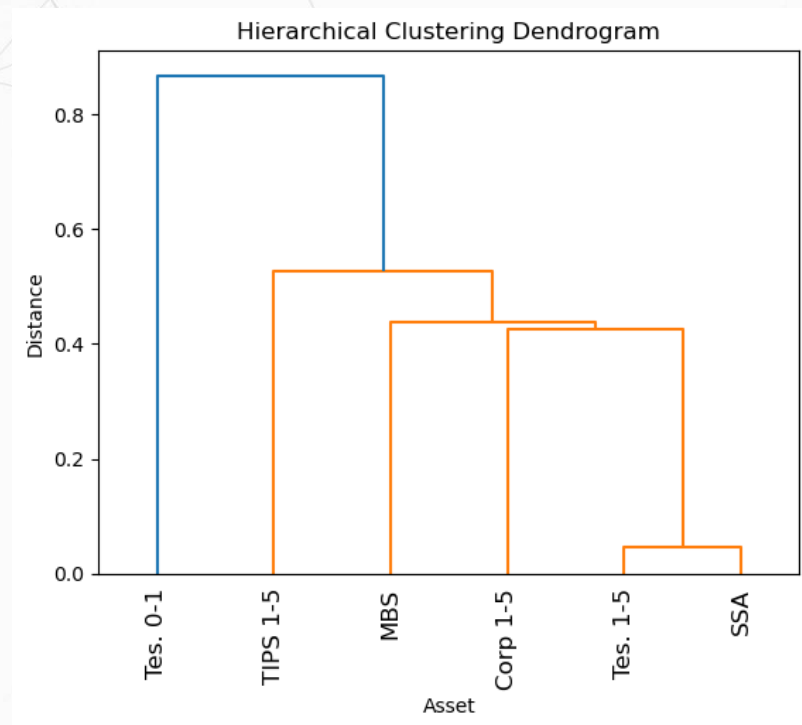
Al final los resultados del algoritmo serían los siguientes:

	((a,b),c,e)	d
((a,b),c,e)	0	28
d	28	0

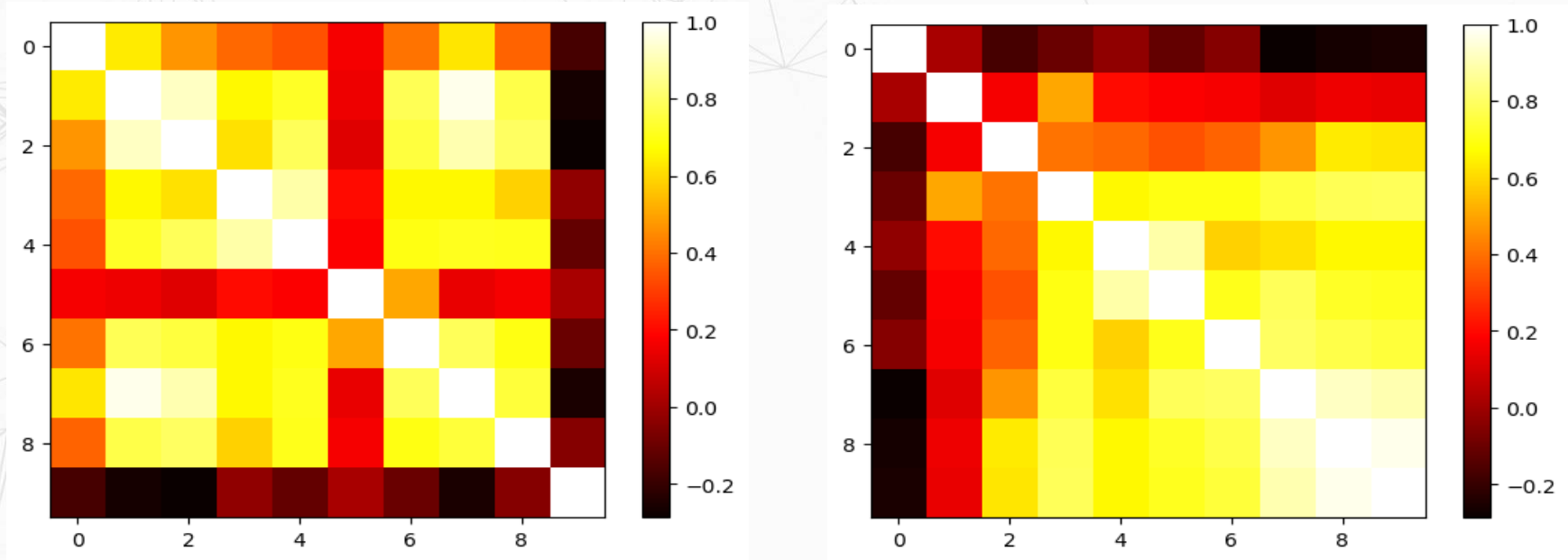
De esta manera la el conjunto final lo podríamos denotar de la siguiente manera: $\{(a,b),c,e\},d\}$.

Paso 1: Tree Clustering

	Orden Inicial	Nuevo Orden
0	Tes. 0-1	Tes. 0-1
1	Tes. 1-5	TIPS 1-5
2	TIPS 1-5	MBS
3	Corp 1-5	Corp 1-5
4	SSA	Tes. 1-5
5	MBS	SSA



Paso 2: Quasi-Diagonalización



Consiste en ir sustituyendo en la matriz final de clusters, las columnas y filas de la matriz de covarianza.

Paso 3: Bisección Recursiva

```
def get_cluster_var(cov, c_items):  
    cov_ = cov.iloc[c_items, c_items] # matrix slice  
    # calculate the inverse-variance portfolio  
    ivp = 1./np.diag(cov_)  
    ivp/=ivp.sum()  
    w_ = ivp.reshape(-1,1)  
    c_var = np.dot(np.dot(w_.T, cov_), w_)[0,0]  
    return c_var
```

```
def get_rec_bipart(cov, sort_ix):  
    # compute HRP allocation  
    # initialize weights of 1  
    w = pd.Series(1, index=sort_ix)  
  
    # initialize all items in one cluster  
    c_items = [sort_ix]  
    while len(c_items) > 0:  
        # bisection  
        """"  
        [[ 0, 3, 4], [6, 2, 1, 5]]  
        [[0], [3, 4], [6, 2], [1, 5]]  
        [[3], [4], [6], [2], [1], [5]]  
        []  
        """"  
  
        print([])  
        c_items = [i[int(j):int(k)] for i in c_items for j,k in  
                    ((0,len(i)/2),(len(i)/2,len(i)))] if len(i)>1]  
  
        print(c_items)  
        # now it has 2  
        for i in range(0, len(c_items), 2):  
  
            c_items0 = c_items[i] # cluster 1  
            c_items1 = c_items[i+1] # cluster 2  
  
            c_var0 = get_cluster_var(cov, c_items0)  
            c_var1 = get_cluster_var(cov, c_items1)  
  
            alpha = 1 - c_var0/(c_var0+c_var1)  
  
            print(c_var0, c_var1)  
  
            w[c_items0] *= alpha  
            w[c_items1] *=1-alpha  
  
        print(w)  
    return w
```


Paso 3: Bisección Recursiva

El paso de la bisección recursiva consta a su vez de los siguientes tres pasos:

1. Inicialización
2. Verificar Detención
3. Bisección
4. Repetir en *loop* desde el paso 2.



4. Iteración:

1. Repite el proceso desde el paso 2

Paso 3: Bisección Recursiva

1. Inicialización:

1. Configura un conjunto de conjuntos llamado L , donde inicialmente L estará conformando por solo un conjunto L_0 que contiene todos los activos desde el 1 hasta el N $\{1, 2, \dots, N\}$.
2. Asuma un peso unitario para cada activo:
 $w_n = 1 \forall n \in N$

2. Verificar Detención:

1. Si cada subconjunto de L $L_i \in L$, tiene solo un elemento ($|L_i| = 1 \forall L_i \in L$, entonces detener las iteraciones del algoritmo.



4. Iteración:

1. Repite el proceso desde el paso 2

Paso 3: Bisección Recursiva

3. Bisección:

1. Para cada subconjunto de $L_i \in L$ que tiene más de un elemento proceder con los siguientes pasos.

a. Realice la bisección de L_i en dos conjuntos: $L_i^1 \cup L_i^2 = L_i$, donde $|L_i^1| = \text{int} \left\lfloor \frac{1}{2} |L_i| \right\rfloor$, y el orden se preserva.

b. Definimos la varianza del subconjunto L_i^j , $j = 1, 2$ con la siguiente formula cuadrática:

$$\tilde{V}_i^j = (\tilde{w}_i^j)^T V_{[i,i]} (\tilde{w}_i^j) \quad \forall j \in \{1, 2\}$$

Donde los pesos \tilde{w} de la fórmula se obtienen solucionando el problema de 'Inverse Variance Portfolio':



4. Iteración:

1. Repite el proceso desde el paso 2

Paso 3: Bisección Recursiva

Problema de *Inverse Variance Portfolio*.

Consiste en encontrar pesos en proporciones inversas a la volatilidad del portafolio

$$w_i = \frac{\left(\frac{1}{\sigma_i^2}\right)}{\sum_{i=1}^n \left(\frac{1}{\sigma_i^2}\right)}$$

En el paper de López de Prado, se prefiere entonces utilizar la notación matricial para formular el problema:

$$\tilde{w}_i^j = \frac{\text{diag}(v_i^j)^{-1}}{\text{tr}(\text{diag}(v_i^j)^{-1})}$$



4. Iteración:

1. Repite el proceso desde el paso 2

Paso 3: Bisección Recursiva

3. Bisección

1. Para cada subconjunto de $L_i \in L$ que tiene más de un elemento proceder con los siguientes pasos.
 - a. Realice la bisección de L_i en dos conjuntos: $L_i^1 \cup L_i^2 = L_i$, donde $|L_i^1| = \text{int} \left\lfloor \frac{1}{2} |L_i| \right\rfloor$, y el orden se preserva.
 - b. Definimos la varianza del subconjunto L_i^j , con la formula cuadrática $\tilde{V}_i^j = (\tilde{w}_i^j)^T V_{[i,i]} (\tilde{w}_i^j) \forall j \in \{1,2\}$ y los pesos
 - c. Computamos el factor de reponderación para los pesos del portafolio con la siguiente ecuación: $\alpha_i = 1 - \frac{\tilde{V}_1^{(1)}}{\tilde{V}_1^{(1)} + \tilde{V}_1^{(2)}}$
 - d. Re-escalamos entonces los pesos w_n por los factores de reponderación $\alpha_i \forall n \in L_i^{(1)}$
 - e. Re-escalamos entonces los pesos w_n por el factor de reponderación $1 - \alpha_i \forall n \in L_i^{(2)}$

4. Repetir en 'loop' el paso 2.

¿Cuál es el gran beneficio del HRP?

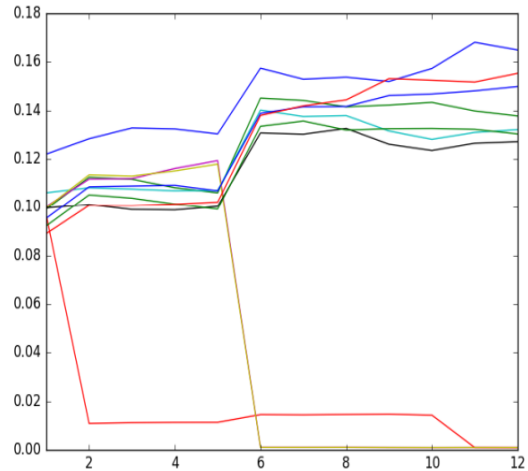


Exhibit 8(a) – Time series of allocations for IVP

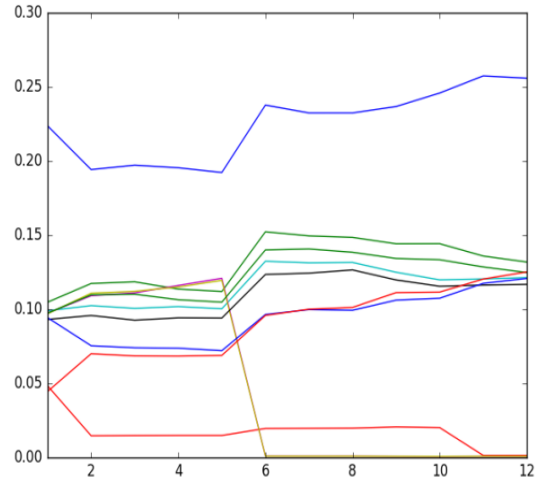


Exhibit 8(b) – Time series of allocations for HRP

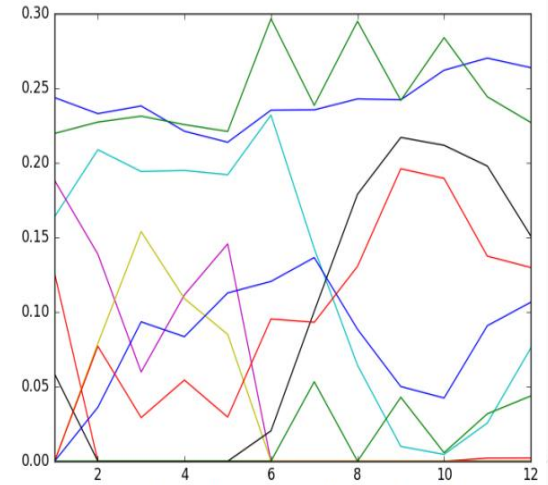


Exhibit 8(c) – Time series of allocations for CLA



Aprendizaje No Supervisado: Factor Risk Parity.

Aprendizaje no supervisado: Factores de Riesgo

En líneas generales se pueden clasificar tres tipos de modelos de factores de riesgo:

- (1) Modelos macroeconómicos de factores.
- (2) Modelos fundamentales de factores.
- (3) Modelos estadísticos de factores de riesgo.

Particularmente, en nuestra exposición nos vamos a centrar en los Modelos Estadísticos y sus aplicaciones en la construcción de portafolios.

Factores de Riesgo: Análisis de Componentes Principales

El análisis de componentes principales (PCA) es una técnica estándar para extraer factores no correlacionados como combinaciones lineales de los retornos de los activos. Generalmente los factores obtenidos de esta manera son ordenados en proporción al peso que explica cada variable.

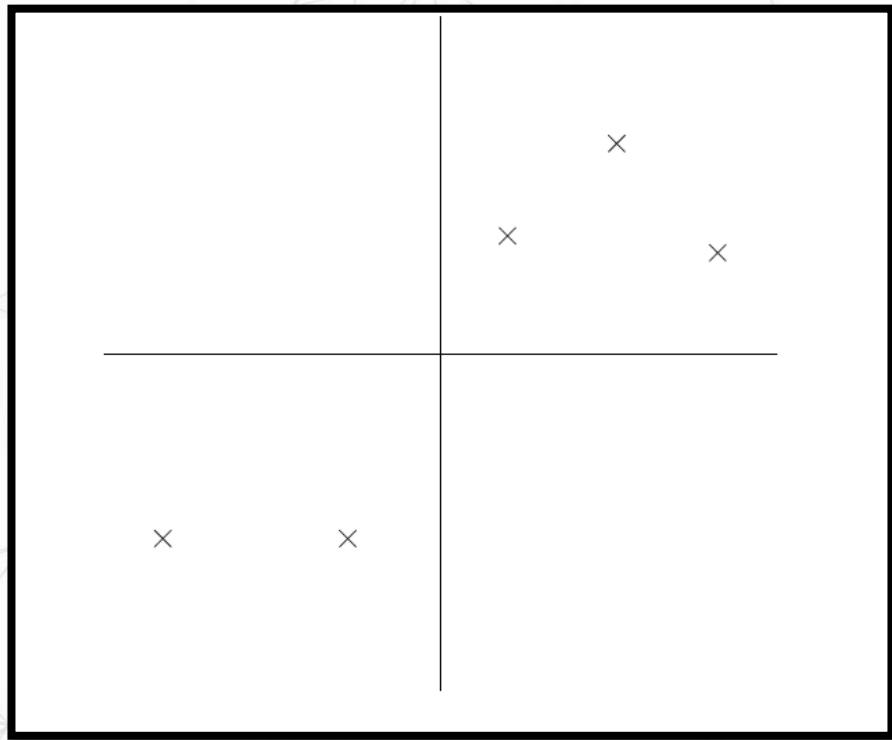
Que características deseables deberían tener estos factores

1. Que preserven la mayor parte de la variabilidad de los datos. Esto quiere decir que no se pierda información que pueda resultar importante.

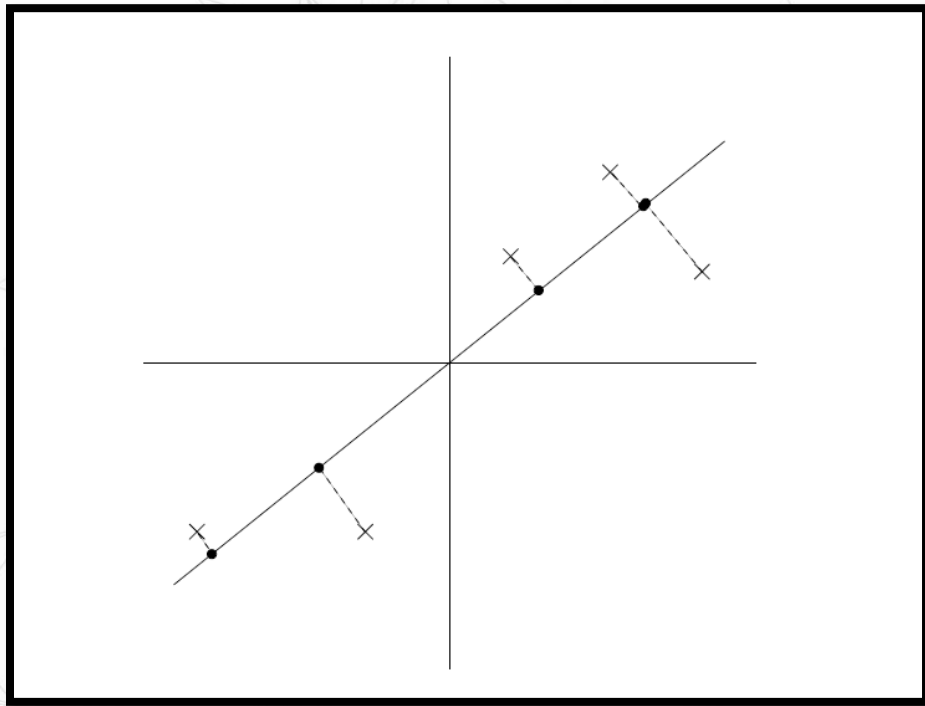
¿Por qué se aplica generalmente esta técnica en el contexto de 'feature engineering'?

1. En el contexto de la ciencia de datos, se le conoce como una técnica de 'feature engineering' para la reducción de la dimensionalidad.
2. En finanzas y otros contextos puede querer usarse para captar aquellos factores no observados que son difíciles de captar o de medir. Por ejemplo, en pruebas estandarizadas la capacidad innata del individuo o en finanzas el sentimiento de mercado (Hastie, Tibishirani y Friedman, 2023)

Análisis de Componentes Principales: Intuición



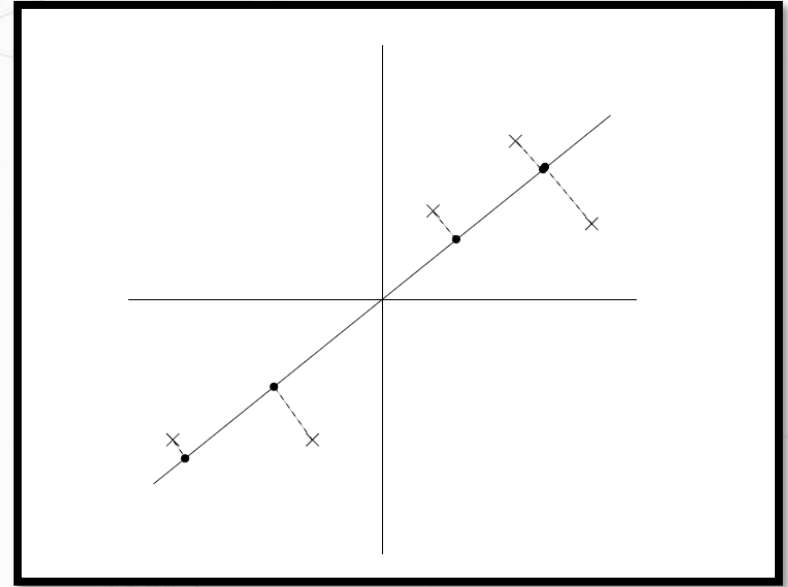
Análisis de Componentes Principales: Intuición



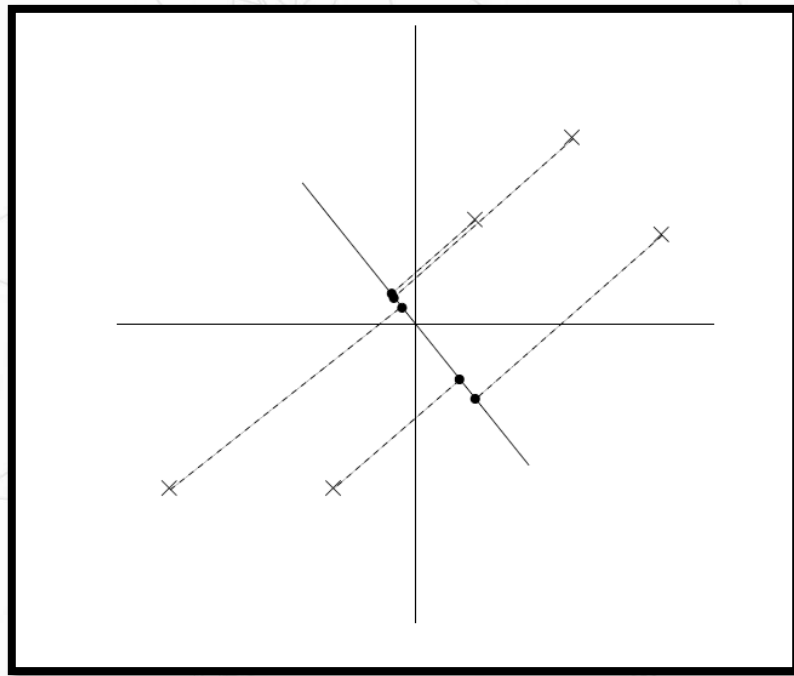
Análisis de Componentes Principales: Intuición

Queremos hallar una nueva variable como una combinación lineal de las variables explicativas ya existentes.

Esta variable debería de capturar la mayor parte de la varianza posible de nuestros datos.



Análisis de Componentes Principales: Intuición



Factores de Riesgo: Análisis de Componentes Principales

Siendo que la proyección de alguna observación $x^{(i)}$ sobre esta recta definida por el vector u , está dada por $x^T u$

La idea de maximizar la varianza de las proyecciones estaría determinada por la siguiente ecuación:

$$\max \frac{1}{n} \sum_{i=1}^n (d_i)^2$$
$$\frac{1}{n} \sum_{i=1}^n (d_i)^2 = \frac{1}{n} \sum_{i=1}^n (x^{(i)T} u)^2 = \frac{1}{n} \sum_{i=1}^n u^T (x^{(i)} x^{(i)T}) u$$

Al reagrupar

$$\sum_{i=1}^n u^T \left(\frac{1}{n} x^{(i)} x^{(i)T} \right) u = u^T (\Sigma) u$$

Este problema no es otro que el de obtener los valores propios de la matriz Σ . Particularmente si se normalizan los vectores propios, se puede encontrar que entre mayor sea la magnitud del valor propio mayor la distancia al centro y por ende mayor la variabilidad capturada de los datos.

Factores de Riesgo: Análisis de Componentes Principales

¿Cuántos factores retener?

El número de factores principales a retener generalmente es especificado por el analista. No hay una regla que funcione como norma general, pero una norma justificada desde la teoría estadística es la propuesta por Laloux et al. (2000) según la cual se debe preservar los eigenvalues que cumplan:

$$\lambda > \lambda_{max} = \left[1 + \sqrt{\frac{N}{T}} \right]^2$$

Factores de Riesgo: Minimum Linear Torsion

Para sobreponerse a los anteriores problemas mencionados Meucci, Santangelo, and Deguest (2015) proponen un método alternativo llamado "Minimum Linear Torsion". Esta técnica también permite encontrar una serie de factores no correlacionados, lo más parecidos a los activos originalmente disponible.

La idea es entonces encontrar una canasta de pesos que forme unos factores no correlacionados y que la suma del tracking error de los factores con respecto a los activos sea tan baja como sea posible.

De esta manera se gana en robustez y en interpretabilidad.

Factores de Riesgo: Minimum Linear Torsion

¿Qué busca hacer el Minimum Linear Torsion?

Supóngase que se define el vector R , y también se cuenta con una matriz A no-singular que permite hacer el cambio de base de factores a activos. Esto se puede expresar matemáticamente de la siguiente forma:

$$F = A'R$$

El retorno de un portafolio se podría escribir como:

$$R_p = w'R = w' A^{-1}AF = (A^{-1}w)'F$$

De tal manera que las exposiciones del portafolio se pueden calcular con la siguiente ecuación:

$$\beta_p = A^{-1}w$$

Factores de Riesgo: Minimum Linear Torsion

El problema de optimización es entonces encontrar la matriz A que minimiza la suma del cuadrado del Tracking Error entre el factor y el activo, mientras, que se busca que estos factores tengan las mismas varianzas de los activos:

$$\min_A \sum_{i=1}^N (A_{:,i} - e_i)' \Sigma (A_{:,i} - e_i)$$

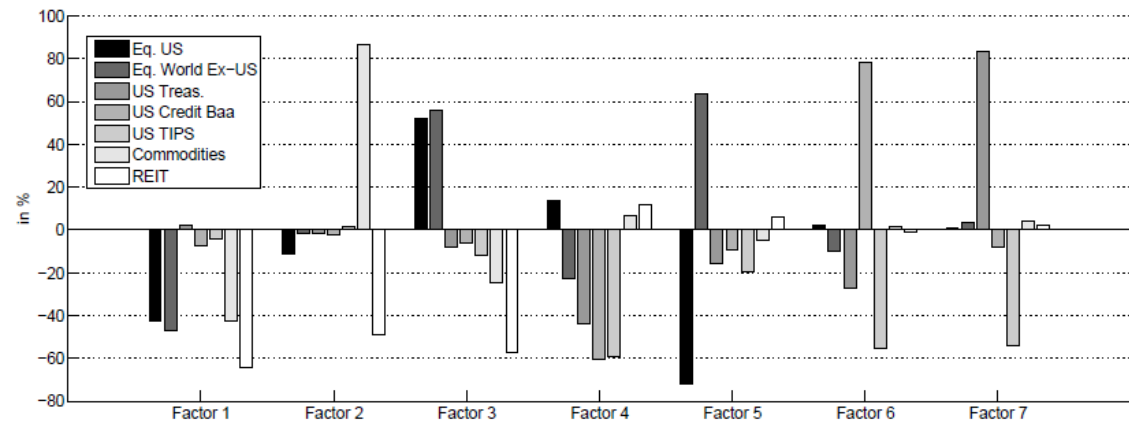
$$\min_A \sum_{i=1}^N (A_{:,i} - e_i)' \Sigma (A_{:,i} - e_i)$$

s. a.

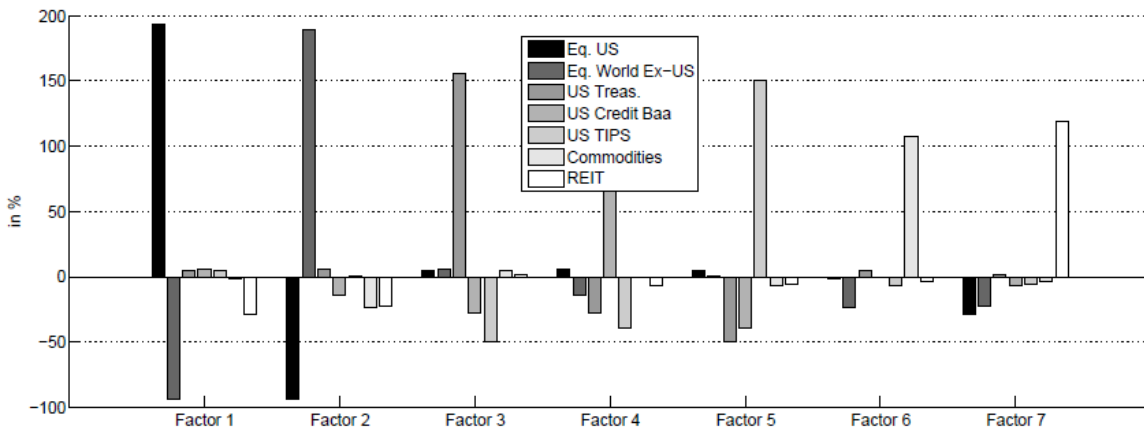
$$F = A' R$$

$$A' \Sigma A = \text{diag}(\Sigma)$$

(a) Principal component analysis.



(b) Minimum linear torsion.



Factores de Riesgo: Minimum Linear Torsion

Factor Risk Parity

El enfoque de Factor Risk Parity parte de hacer una atribución al riesgo con factores ortogonales. Esto tiene la gran ventaja de no tener que recurrir a la norma de segmentar la covarianza de los activos en dos.

Se dice que el retorno de un portafolio será igual al retorno de cada clase de activo contra el factor:

$$\begin{aligned} R_p &= \sum_{i=1}^N w_i R_i = \sum_{i=1}^N w_i \left(c_i + \sum_{k=1}^K \beta_{ik} F_k \right) \\ R_p &= \sum_{i=1}^N w_i c_i + \sum_{i=1}^N \sum_{k=1}^K w_i \beta_{ik} F_k \\ R_p &= \sum_{i=1}^N w_i c_i + \sum_{k=1}^K \sum_{i=1}^N w_i \beta_{ik} F_k \end{aligned}$$

Factor Risk Parity

$$R_p = \sum_{i=1}^N w_i c_i + \sum_{k=1}^N \left(\sum_{i=1}^N w_i \beta_{ik} \right) F_k$$

El término entre paréntesis es la exposición del portafolio a cada factor.

$$R_p = \sum_{i=1}^N w_i c_i + \sum_{k=1}^N \beta_{pk} F_k$$

Este término se podría encontrar con una regresión lineal, no obstante, al estar usando factores ortogonales también se puede hallar como:

$$\beta_{pk} = \frac{\text{cov}(R_p, F_k)}{\text{Var}(F_k)}$$

Podemos entonces ahora si expresar la varianza como una función de la varianza de los factores, al ser ortogonales:

$$\sigma_p^2 = \left(\sum_{i=1}^N \beta_{pk}^2 \sigma_{F,k}^2 \right)$$

Factor Risk Parity

La contribución al riesgo de cada factor sería:

$$c_{F,k} = \frac{\beta_{p,k}^2 \sigma_{F,k}^2}{\sigma_p}$$

De esta manera se puede formar una métrica de Risk Parity:

$$c_{F,1} = \dots = c_{F,k}$$

Bajo la cuál cada factor concentre el mismo porcentaje del riesgo del portafolio.



Referencias

CFA Institute. (2024). *Quantitative methods*. In *CFA Program Curriculum 2024: Level II (Vol. 1)*. CFA Institute.

De Prado, M. L. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, 42(4), 59-69. [Building Diversified Portfolios that Outperform Out-of-Sample by Marcos Lopez de Prado :: SSRN](#) *Hierarchical Risk Parity*

EDHEC – Risk Institute (2018). Chapter 1.3 – Technical Supplement: Factor Risk Parity Portfolio. 1 – 11. *Factor Risk Parity*

EDHEC – Risk Institute (2018). Chapter 1.3 – Technical Supplement: Factor Risk Parity Portfolios. 1 – 9. *Factor Risk Parity*



Referencias

Hudson & Thames. (n.d.). *The Hierarchical Risk Parity Algorithm: An Introduction*. Retrieved February 12, 2025, from <https://hudsonthames.org/an-introduction-to-the-hierarchical-risk-parity-algorithm/> *Hierarchical Risk Parity*

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer. *Machine Learning*

Kinlaw, W., Kritzman, M. P., & Turkington, D. (2021). *Asset Allocation: From Theory to Practice and Beyond*. John Wiley & Sons. *Machine Learning*



Referencias

Ng, A. (2020). *CS229 Lecture Notes - Principal Component Analysis*.

Retrieved February 12, 2025, from <https://cs229.stanford.edu/notes2020spring/cs229-notes10.pdf>.

*Aprendizaje No Supervisado.
Reducción de dimensionalidad*

Sekine, A. (2022). Estimating the Yield Curve Using the Nelson-Siegel Model: Evidence from Daily Yield Data. *Public Policy Review*, 18(1), 1-14. *Aprendizaje No Supervisado.
Regresión*



Referencias

Shalev-Shwartz, S., & Ben-David, S.
(2014). *Understanding machine
learning: From theory to algorithms*.
Cambridge university press.

Machine Learning

