Trabajo Práctico 1: Autovalores y Cuadrados Mínimos

1. Velocidad del viento

El archivo `wind.data` presenta la velocidad diario promedio del viento desde 1961 hasta 1978 en 12 estaciones meteorológicas de Irlanda (vea la descripción más detallada de los datos en el archivo `wind.desc`.

1. Los datos están dados en *nudos* (*knots* en inglés). Conviértalos a m/s.

2. Calcule el valor medio de la velocidad del viento en cada estación meteorológica (esos son 12 valores).

3. Para cada una de las estaciones, ajuste los datos a una fórmula del siguiente tipo:

   $$v(t) = A_0 + A_1 \cos(2\pi f_1 t) + B_1 \sin(2\pi f_1 t),$$

   donde v es la velocidad del viento, t es el tiempo en días y $f_1 = \frac{1}{365{,}25}\,\text{día}^{-1}$.

   a) Calcule el error cuadrático medio del ajuste.
   b) ¿Por qué 365.25 días de período?
   c) ¿Cómo se relaciona A_0 con el valor medio calculado más arriba?
   d) Grafique un histograma del error entre los valores y el ajuste.

   Resuelva este problema usando uno de los siguientes métodos (pregunte al profesor encargado de los Trabajos Prácticos cuál le corresponde usted):

   - Ecuaciones normales y eliminación Gaussiana.
   - Cholesky.
   - QR.

   Implemente su propio algoritmo y compare los resultados con los obtenidos, por ej., mediante Matlab u Octave.

2. Mercado Financiero

El archivo `spdc2693.txt` contiene datos del valor de cierre del índice Standard and Poor's 500 entre los años 1926 y 1993 (una descripción más completa de los datos la puede encontrar en el mismo archivo).

Nota: Frente a cualquier conjunto de datos, una buena práctica suele ser estandarizarlos de alguna manera de forma tal de evitar números demasiado grandes o demasiado pequeños en los cálculos. Una forma común de estandarizar datos es restándoles la media muestral y dividiéndolos por el desvío estándar muestral. Recomendamos fuertemente hacer esto con la escala temporal (primera columna de los datos) de este ejercicio.

1. Escribamos los datos como $[\vec{t}, \vec{s}]$, donde $\vec{t}$ es el vector columna correspondiente al tiempo y $\vec{s}$ el vector columna correspondiente al índice. Deseamos ajustar el índice Standard and Poor's $\vec{s}$ a la función:

   $$y_1(t) = \exp\left\{A_0 + A_1 t + A_2 t^2\right\},$$

   donde y_1 es el valor del índice y t es el tiempo. Para ello, se realizará el ajuste por cuadrados mínimos del logaritmo natural del índice a la función

   $$y_2(t) = A_0 + A_1 t + A_2 t^2.$$

   a) Calcule el error cuadrático del ajuste $\| \ln(\vec{s}) - y_2(\vec{t})\|$.

   b) Calcule el error cuadrático del ajuste $\|\vec{s} - y_1(\vec{t})\|$.

   c) ¿Existe alguna relación entre los errores calculados más arriba? Saque conclusiones.

   Resuelva este problema usando uno de los siguientes métodos (pregunte al profesor encargado de los Trabajos Prácticos cuál le corresponde usted):

   - Ecuaciones normales y eliminación Gaussiana.
   - Cholesky.
   - QR.

   Implemente su propio algoritmo y compare los resultados con los obtenidos, por ej., mediante Matlab u Octave.

3. Autovalores y compresión de imágenes

En este ejercicio vamos a trabajar con una imagen que se utiliza habitualmente como test que es la famosa "Lena" (`lena512.bmp`). En nuestro caso particular, se trata de una imagen de 512×512 pixels en escala de grises. Será de gran utilidad que use Matlab u Octave con el paquete de procesamiento de imágenes. Como ejemplo, usaremos el paquete `image` de Octave en lo que sigue de este ejercicio.

3.1. Compresión a lo bruto 1: la imagen media

Una forma muy simple de comprimir una imagen es la siguiente. Divida la imagen en bloques no solapados de, por ej., 16×16. En la imagen de Lena (de 512×512) hay 1024 bloques: dar la imagen es equivalente a dar 256 números por cada bloque. Una forma burda de comprimir la imagen es dar lo mismos 256 números para todos los bloques, comprimiendo la imagen (con pérdida) de 256 kbytes a 256 bytes. ¿Qué números conviene dar? Una opción es dar los valores correspondientes a un "bloque promedio". La siguiente secuencia de comandos en Octave permite hacer dicha operación:

```
a = imread('lena512.bmp');
imshow(a) % muestro la imagen original
b = im2col(a,[16,16],'distinct');
m = mean(double(b.'));
m = m.';
M = repmat(m,1,1024);
d = col2im(M,[16,16],[512,512],'distinct');
meanimg = uint8(round(d));
imshow(meanimg) % muestro la imagen ``media''
```

Ejercicios:

1. Realice la compresión propuesta sobre la imagen de Lena provista. En caso de usar el código de Octave propuesto, indique claramente qué hace cada función.
2. Muestre la imagen antes y después de la compresión. Comente acerca de los resultados.

3.2. Compresión a lo bruto 2: PCA

Desde el punto de vista del álgebra lineal, la imagen dividida en bloques de 16×16, como en el ejercicio anterior, se puede considerar simplemente como 1024 vectores de 256 elementos cada uno (la matriz `b` más arriba). Es decir, tenemos vectores en $\mathbb{R}^{256}$. Dada una base cualquiera de $\mathbb{R}^{256}$, que cuenta con 256 vectores linealmente independientes, podemos representar cada uno de los bloques de la imagen por su proyección sobre cada vector en la base. Es decir, si queremos darle la imagen a alguien, le podemos dar la base y los 256 números correspondientes a la proyeccción sobre cada vector de la misma.

¿Y si usamos $M < 256$ vectores linealmente independientes en la base? De esta manera, para comunicar la imagen sólo tenemos que dar M vectores en la base y M números por cada bloque. El problema es que usando sólo M vectores no se puede representar todo $\mathbb{R}^{256}$ y la imagen comunicada no sería exactamente igual a la original. ¿Cómo podemos elegir qué M vectores utilizar de manera de minimizar la pérdida de calidad de la imagen? La respuesta la da lo que se denomina *Principal Component Analysis* y, en nuestro caso, es más o menos como sigue.

Recordemos de nuestro curso de Probabilidad que la covarianza nos da una idea de qué tanto cambian juntas dos variables. Si tenemos, por ej., dos variables aleatorias Y, Z la covarianza está definida como

$$\operatorname{cov}(Y, Z) = \mathrm{E}\left[(Y - \mathrm{E}(Y))\,(Z - \mathrm{E}(Z))\right].$$

En nuestro caso, por ej., podemos considerar a cada uno de los 256 números que representan el nivel de gris de un pixel en un bloque como un variable aleatoria. Llamemos X_i, $i = 1, 2, \cdots, 256$, a dichas variables aleatorias. Luego, podemos definir una matriz $\mathbf{C} = (C_{ij})$ de la siguiente manera,

$$C_{ij} = \operatorname{cov}(X_i, X_j).$$

A $\mathbf{C}$ se la denomina matriz de covarianza. Esta matriz tiene algunas características interesantes:

- es simétrica;
- los elementos de la diagonal representan la varianza de cada variable aleatoria ($C_{ii} = \operatorname{var}(X_i)$) y, por tanto, son no-negativos.

Basados en lo anterior, se puede demostrar que $\mathbf{C}$ es semi-definida positiva (y, casi siempre, se puede tachar "semi"). Recordemos qe una matriz simétrica semi-definida positiva tiene todos sus autovalores mayores o iguales a cero.

Sin embargo, nosotros no tenemos distribuciones de probabilidad que nos permitan calcular la matriz covarianza. Sólo tenemos, en el ejemplo de la imagen de Lena, 1024 muestras de cada variable aleatoria: una por cada bloque. Esto nos permite calcular la matriz de covarianza muestral $\hat{\mathbf{C}} = (\hat{C}_{ij})$:

$$\hat{C}_{ij} = \frac{1}{1024 - 1} \sum_{k=1}^{1024} (x_{ik} - \hat{x}_i)\,(x_{jk} - \hat{x}_j)\,,$$

donde

- x_{ik} es la k-ésima muestra de la i-ésima variable aleatoria o, en el contexto de nuestra imagen, el nivel de gris del i-ésimo pixel en el k-ésimo bloque, y
- $\hat{x}_i$ es la media muestral de la i-ésima variable aleatoria o el nivel de gris del i-ésimo pixel del "bloque promedio",

$$\hat{x}_i = \frac{1}{1024} \sum_{k=1}^{1024} x_{ik}.$$

Volviendo al principio: ¿qué información nos da la matriz de covarianza muestral $\hat{\mathbf{C}}$? $\hat{C}_{ij}$ nos habla de qué tanto varían juntos los niveles de gris del i-ésimo y el j-ésimo pixel en los bloques de la imagen. Y ahora viene el punto clave: vamos a calcular los autovalores y autovectores. Cada autovector nos una dirección característica de cambio y el autovalor correspondiente (siempre ≥ 0) nos dice qué tan importante es el cambio en dicha dirección. En el contexto de la imagen de Lena, cada autovector es una "autoimagen" que nos habla de un esquema de cambio característico de los niveles de gris de los bloques y cada autovalor nos habla de la importancia de ese esquema frente a los demás.

¿Y cómo nos sirve todo esto para comprimir la imagen? Los autovectores conforman la base de $\mathbb{R}^{256}$ de la que hablábamos más arriba. Los autovalores nos dicen cómo elegir un subconjunto de esos autovectores: elegimos aquellos que tienen el mayor autovalor.

El código de Octave que sigue, muestra cómo realizar la compresión de la imagen usando estas ideas. En particular, se muestra el resultado obtenido de considerar una sola autoimagen.

```
ds = double(b)-M;
cc = cov(double(b'));
[V,D] = eig(cc);
% ordeno los autovalores de mayor a menor
D = diag(D);
[D,i] = sort(D,'descend');
D = diag(D);
V = V(:,i);
% muestro la primera ''autoimagen''
c = (V(:,1)-min(V(:,1)))*256/(max(V(:,1))-min(V(:,1)));
dv1 = col2im(c,[16,16],[16,16],'distinct');
dv1 = uint8(round(dv1));
imshow(dv1)
% proyecto la primer autoimagen
pv1 = V(:,1).'*ds.';
% si solamente considero la primer autoimagen...
d = M;
for k = 1:1024
   d(:,k) = d(:,k) + pv1(k)*V(:,1);
end
compimg = col2im(d,[16,16],[512,512],'distinct');
compimg = uint8(round(compimg));
imshow(compimg)
```

Ejercicios:

1. Realice la compresión propuesta sobre la imagen de Lena provista. En caso de usar el código de Octave propuesto, indique claramente qué hace cada función.

2. Muestre las primeras cinco autoimágenes.

3. Muestre la imagen antes y después de la compresión cuando usa 1, 2, 3, 4 y 5 autovectores. Comente acerca de los resultados.

4. Repita la compresión con un solo autovector, pero usando su propio código para calcular el autovalor y el autovector. Utilice uno de los siguientes algoritmos (de acuerdo a lo que le indique el encargado de Trabajos Prácticos):

   - Método de potencias inversas con traslación.
   - Método de Jacobi.
   - Método de QR con traslaciones.

5. El mecanismo de compresión hasta aquí propuesto implica que, para dar una imagen, hay que comunicar valores de punto flotante (los autovectores y autovalores).

   a) Dé una alternativa que no implique comunicar números en punto flotante e impleméntela. Asuma que utiliza los primeros cinco autovectores.

   b) Muestre cómo queda la imagen luego de la compresión.

   c) ¿Cuánto se ha comprimido la imagen?

4. Un sistema de comunicaciones

Un modelo muy simple (discreto en banda base) de un sistema de comunicaciones es el siguiente:

1. El transmisor envía un dato s_k cada T segundos, donde s_0 es enviado en $t = 0$, s_1 en $t = T$, s_2 en $t = 2T$, ...

2. Los datos son modificados por el canal, es decir, por el medio donde son transmitidos. Esa modificación está presentada por la así denominada respuesta al impulso del canal[1] $\{h_k\}_{k=0}^{L}$, donde L es la longitud de la respueta al impulso.

3. Además de ser modificados por el canal, los datos son afectados por ruido blanco Gaussiano aditivo $N_k \sim cN(0, \sigma)$.

Teniendo todo esto en cuenta, cada T segundos el receptor observa:

$$r_n = \sum_{k=0}^{L-1} h_k s_{n-k} + N_n. \tag{1}$$

También podemos expresar la Ec. 1 en forma matricial como

$$\vec{r} = \mathbf{H}\vec{s} + \vec{N}, \tag{2}$$

donde

$$\vec{r} = \begin{pmatrix} r_0 \\ r_1 \\ \vdots \\ r_M \end{pmatrix}, \ \vec{s} = \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{M-1} \end{pmatrix}, \ \vec{N} = \begin{pmatrix} N_0 \\ N_1 \\ \vdots \\ N_{M-1} \end{pmatrix}, \tag{3}$$

$$\mathbf{H} = \begin{pmatrix} h_0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ h_1 & h_0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & 0 & 0 & \cdots & 0 & 0 \\ h_{L-1} & h_{L-2} & \cdots & h_1 & h_0 & 0 & \cdots & 0 & 0 \\ 0 & h_{L-1} & \cdots & h_2 & h_1 & h_0 & \cdots & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & h_1 & h_0 \end{pmatrix} \in \mathbb{R}^{M \times M}, \tag{4}$$

y $M \geq L$.

Otra forma equivalente es la siguiente:

$$\vec{r} = \mathbf{S}\vec{h} + \vec{N}, \tag{5}$$

donde $\vec{r}$ y $\vec{N}$ son como antes y

$$\vec{h} = \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_L \end{pmatrix}, \tag{6}$$

[1] En la próxima guía vamos a explicar qué es una respuesta al impulso.

$$\mathbf{S} = \begin{pmatrix} s_0 & 0 & \cdots & & 0 \\ s_1 & s_0 & \cdots & & 0 \\ \vdots & & \vdots & & 0 \\ s_{L-2} & s_{L-3} & \cdots & s_0 & 0 \\ s_{L-1} & s_{L-2} & \cdots & s_1 & s_0 \\ s_L & s_{L-1} & \cdots & s_2 & s_1 \\ \vdots & s_{M-1} & s_{M-2} & \cdots & s_{M-L} \end{pmatrix} \in \mathbb{R}^{M\times L}. \tag{7}$$

Lo que nos interesa es recuperar correctamente la señal enviada $\vec{s}$ a partir de la señal recibida $\vec{r}$. Esto no sería tan difícil si se conociesen L y los $\{h_k\}$. En efecto, tomando $M = L$ en la Ec. 2, podemos buscar $\vec{s}$ que minimice:

$$\|\mathbf{H}\vec{s} - \vec{r}\|_2^2 . \tag{8}$$

El problema es, sin embargo, que en general ni L ni $\{h_k\}$ son conocidos: estimarlos corresponde al problema de estimación del canal.

Si L es conocido, una forma de estimar el canal es mediante la Ec. 5 y el método de cuadrados mínimos. En efecto, tómese $M > L$ y envíese una señal conocida por el receptor $\vec{s} \in \mathbb{R}^M$ (denominada secuencia de entrenamiento). Luego, se estima el canal planteando el siguiente problema de cuadrados mínimos: encontrar $\vec{h} \in \mathbb{R}^L$ que minimice

$$\left\|\mathbf{S}\vec{h} - \vec{r}\right\|_2^2 . \tag{9}$$

Este trabajo práctico consiste en la estimación de un canal mediante cuadrados mínimos. Más específicamente:

1. Genere una respuesta al impulso del canal aleatoria, con $L = 30$. En Octave, esto se puede hacer de la siguiente manera:

```
L = 30;
ganancia = 1/10;
h = ganancia*(1+randn(L,1));
```

2. Usando ruido con desvío estándar $\sigma = 1$, transmita la imagen de Lena en escala de grises que se le provee y muestre los resultados. La forma de transmitir la imagen de Lena de izquierda a derecha y de arriba hacia abajo. Para facilitar el proceso, transmitiremos una línea a la vez. En Octave, esto se puede hacer de la siguiente manera:

```
sigma = 1;
a = imread('lena512.bmp');
imshow(a) % muestro la imagen original
M = size(a,2);
```

```
P = size(a,1);
H = toeplitz([h.' zeros(1,M-L)],zeros(1,M));
r = zeros(M,P);
N = sigma*randn(M,1); % ruido
% se transmite una línea a la vez s = double(a(k,:)); % lo
que se envía r = H*s + N; % lo que se recibe
b = uint8(r.');
imshow(b)
```

---

3. Estime $\vec{h}$ usando una secuencia de entrenamiento conocida de longitud $E = 512$.

   a) Al estimar $\vec{h}$ suponga que $L = 1$.

   b) Al estimar $\vec{h}$ suponga que $L = 10$.

   c) Al estimar $\vec{h}$ suponga que $L = 30$.

   d) Al estimar $\vec{h}$ suponga que $L = 50$.

   ¿Cómo es la figura recuperada en cada caso? Comente sus resultados.

4. Repita el ejercicio 3 usando $E = 32$. Comente sus resultados.

5. Repita el ejercicio 3 usando $E = 1024$. Comente sus resultados.

6. Repita los ejercicios 2 y 3 usando $\sigma = 0$. Comente sus resultados.