Ejercicios busqueda estructura

1 Descomposición SVD

1. Mostrar que si una matriz G es positiva entonces existe una matriz H positiva tal que $G = H^2$.
2. Sea $A \in \mathcal{M}_{n,m}(\mathbb{C})$ una matriz a coeficientes en $\mathbb{C}$. Que sera la descomposición SVD en este caso ?
3. (Descomposición polar) Sea A una matriz cuadrada.
   (a) Mostrar que si una matriz $\Gamma \in \mathcal{M}_n(\mathbb{R})$ es positiva y $B \in \mathcal{M}_{n\times m}(\mathbb{R})$ entonces la matriz $B^T \Gamma B$ es positiva.
   (b) Usando la descomposición SVD mostrar que cada matriz cuadrada $A \in \mathcal{M}_n(\mathbb{R})$ tiene una descomposición polar :
   $$A = GQ$$
   donde G es positiva y Q es orthogonal.
   (c) Mostrar que si A es invertible, la descomposición polar es unica.
   (d) Si $A \in \mathcal{M}_n(\mathbb{C})$, mostrar que $\det(G) = |\det(A)| =: r$ y que $\det(Q) = e^{i\theta}$ para un cierto $\theta \in \mathbb{R}$ (desomposición polar del determinante).
4. (Unicidad de la pseudo-inversa) Digamos que A^- es la pseudo inversa de una matriz $A \in \mathcal{M}_{m,n}(\mathbb{R})$ si $AA^-A = A$ y $A^-AA^- = A^-$ y que las matrices AA^- y A^-A son simétricas.
   (a) Sea B una otra pseudo inversa de A. Consideramos $M = AB - AA^-$. Mostrar que M es simétrica y que $M^2 = 0$.
   (b) Deducir que $M = 0$.
   (c) Que pensar de la matriz $BA - A^-A$?
   (d) Usando lo anterior, mostrar la unicidad de la pseudo-inversa.

2 Factorización no-negativa

1. (ANLS) Queremos implementar la descomposición no-negativa de una matriz A en Python.
   (a) Implementar el algoritmo de mínimos cuadrados alternativos dado por
   $$U_{k+1} = \underset{U\geq 0}{\operatorname{argmin}} \|A - UV_k^T\|_F^2$$
   $$V_{k+1} = \underset{V\geq 0}{\operatorname{argmin}} \|A - U_kV^T\|_F^2$$
   sucesivamente. La inicialización se hace poniendo realizaciones de variables uniformes $U[0,1]$ en cada entrada de las matrices. Consideramos 30 iteraciones. (*Pista :* Usar la función `nnls` del paquete `scipy.optimize`)
   (b) Pruebe su implementación en el conjunto de datos sintéticos que se le proporcionó en `nmfdata.txt` (disponible en la pagina). Usa los mismos datos para responder todas las preguntas siguientes.
   (c) Implementar una función que trace la raiz cuadrada del error cuadrático medio (RMSE - root mean square error) del modelo como función del número de componentes, r. Para cada valor de r, ajuste el modelo a partir de múltiples inicializaciones y trazar el RMSE como un punto separado. Para el conjunto de datos proporcionado, ¿es sensible el RMSE a la inicialización ?
   (d) *(Comparación con SVD truncado)* Modifique su diagrama para incluir una línea que trace el RMSE de un modelo SVD truncado en función de r. Compare el rendimiento de NMF con el de SVD para el conjunto de datos `nmfdata.txt`. Cuando genera este gráfico a partir del conjunto de datos proporcionado, ¿el resultado es favorable o ¿desfavorable para NMF ?
2. Consideramos el problema siguiente de factorización regularizada
$$\min_{U,V} \left\{ \frac{1}{2}\|A - UV^T\|_F^2 + \frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2) \right\}$$

Mostrar que las actualizaciones para U y V cuando usamos descenso por gradiente son

$$U_{k+1} = (1-\gamma\lambda)U_k + \gamma E_k V_k$$
$$V_{k+1} = (1-\gamma\lambda)V_k + \gamma E_k^T U_k$$

donde $E_k = A - U_k V_k^T$.

3 Análisis en componentes principales (PCA)

1. En este ejercicio, queremos implementar el algoritmos PCA a mano. Usaremos dos bases de datos (en formato MATLAB) que son `data1.mat` y `faces.mat`.
   (a) Con el paquete `scipy.io` viene la función `loadmat`. Escribir un script para visualizar la nube de puntos de la base de datos `data1.mat`.
   (b) Escribir una función `feature_normalize` que normaliza cada atributo de la matriz de datos.
   (c) Escribir una función `pca` que usa la función `svd` de `numpy.linalg` para captar los valores propios de la matriz de varianza Σ y los vectores propios asociados.
   (d) En una misma grafica, superponer los datos y los dos vectores principales (con origen en la media m de los datos)
   (e) Escribir una función `project_data(X, U, K)` que toma la matriz de datos X, la matriz U de vectores propios y K el número de componentes a conservar y que responde la matriz Z con solamente los K atributos seleccionados.
   (f) Escribir una función `recover_data(Z, U, K)` que toma la matriz reducida Z, la matriz U de vectores propios y K el número de componentes a conservar y que responde la matriz X_rec de mismo tamaño que X tal que X_rec corresponde a la matriz Z relevada en $\mathbb{R}^2$.
   (g) Finalmente, en una última grafica, superponer los datos normalizados, los datos de X_rec y una linea "dashed" que relaciona cada punto de los datos con su proyección en X_rec.
2. Aplicamos el ejercicio anterior a la base de datos de imágenes de caras `faces.mat`.
   (a) Usar el script `display_data` dado en la pagina web para visualizar las 100 primeras caras del conjunto de datos.
   (b) En una misma grafica (pero dos subplots) poner las 100 caras normalizadas de un lado y las 100 proyecciones del otro lado.