UNIVERSIDAD ADOLFO IBAÑEZ
FACULTAD DE INGENIERÍA Y CIENCIAS

Tarea 2

ING560: Algebra Lineal y Optimización para Data Science

Profesor: Miguel Romero

Fecha de Entrega: 04 de Diciembre 2020

Indicaciones:

1. Se aceptarán tareas enviadas después de la fecha de entrega. Se descontará 1.0 punto sobre la nota final por cada día de retraso.

2. La tarea se puede hacer en grupos de a lo más 3 personas.

3. Debe entregar un **informe** con sus respuestas a los problemas. Junto con la tarea se adjunta un archivo `main.py` el cual deberá completar para implementar los algoritmos pedidos. Este archivo incluye también algunos datos de prueba. En las preguntas computacionales no basta con entregar el código, sino que **debe** explicar brevemente la idea de su implementación. Para la preparación del informe se recomienda el uso de LaTeX.

4. Puede bajar las últimas versiones de `Python`, `Numpy` y `Scipy` desde https://www.python.org/, https://numpy.org/ y https://www.scipy.org/.

Problema 1 (70%)

1. (2.5 pts) Utilizando las librerías `NumPy` y `SciPy`, implemente una función `PCA_dim`(A, k) que recibe como argumentos una matriz A y un número natural $k \geq 1$. La matriz A de $n \times m$ almacena m datos cada uno con n features (cada dato es una columna de A). La función `PCA_dim`(A, k) debe retornar la matriz P de $k \times m$ con los datos proyectados a k dimensiones, es decir, tal que la columna i-ésima de P corresponde a la columna i-ésima de A proyectada al subespacio generado por las k componentes principales. Recuerde que al aplicar PCA debe primero centrar los datos almacenados en A.

   Para implementar su función **debe** utilizar la función `svd` de la librería `linalg` de `SciPy`. Todas las matrices y vectores deben ser objetos `ndarray`. Puede ver el archivo `main.py` para algunos ejemplos de usos.

2. (2.5 pts) Implemente una función `PCA_var`(A, α) que recibe como argumentos una matriz A de $n \times m$ igual que la pregunta anterior y un número racional $\alpha \in (0, 1]$. La función `PCA_var`(A, α) debe retornar la matriz P de $k \times m$ con los datos proyectados a k dimensiones, donde k es la mínima cantidad de componentes principales de manera que la proporción de la varianza explicada con respecto a la varianza total es mayor o igual a α (recordar clase 17). Recuerde que al aplicar PCA debe primero centrar los datos almacenados en A.

   Igual que antes, **debe** utilizar la función `svd` de la librería `linalg` de `SciPy`. Todas las matrices y vectores deben ser objetos `ndarray`. Puede ver el archivo `main.py` para algunos ejemplos de usos.

3. (1.0 pts) Aplique su función `PCA_dim` al *Iris Flower dataset*[1] entregado en el archivo `iris.txt`. Este dataset consta de 150 datos acerca de flores cada uno con 4 features

[1] https://en.wikipedia.org/wiki/Iris_flower_data_set.

correspondientes al *largo y ancho del sépalo*, y al *largo y ancho del pétalo*. Adicionalmente las flores están clasificadas en 3 categorías: *setosa, virginica* y *versicolor*. Para cargar los datos a una matriz I, puede utilizar `I = np.loadtxt('iris.txt')`. Observe que en el archivo `iris.txt` cada fila es un dato, por lo cual la matriz que le debe pasar a `PCA_dim` es la transpuesta de I. Aplique `PCA_dim` con $k = 2$ y $k = 3$ y grafique sus resultados utilizando la función `DrawIrisDataSet`(P) dada en `main.py` que recibe los datos proyectados P. Observe que `DrawIrisDataSet` dibuja también las 3 categorías de los datos con distintos colores. Después de graficar sus datos, ¿Qué puede observar acerca de las 3 categorías?

Problema 2 (30%)

1. (4.0 pts) Utilizando las librerías `NumPy` y `SciPy`, implemente una función `LinearRegression`(X, $\mathbf{y}$) que recibe como argumentos una matriz X y un vector $\mathbf{y}$. La matriz X es de $n \times m$ y cada fila representa un dato[2], es decir, tenemos m variables independientes y n observaciones. El vector $\mathbf{y}$ es de dimensión n y contiene las observaciones para la variable a explicar (o dependiente). La función `LinearRegression`(X, $\mathbf{y}$) debe retornar el resultado de aplicar regresión lineal a X e $\mathbf{y}$, es decir, un vector $\mathbf{c}$ de dimensión $m + 1$ con los coeficientes óptimos, junto con un número *err* indicando el error obtenido, es decir, la suma de los errores cuadráticos.

   Para implementar su función **debe** utilizar la función `lstsq` de la librería `linalg` de `SciPy`. Todas las matrices y vectores deben ser objetos `ndarray`. Puede ver el archivo `main.py` para algunos ejemplos de usos.

2. (2.0 pts) Aplique `LinearRegression` al *Iris Flower dataset* del problema anterior utilizando las siguientes 4 combinaciones (X_1, X_2 son las variables independientes e Y la variable dependiente):

   (a) X_1 =largo sépalo, X_2 =ancho sépalo, Y =largo pétalo.

   (b) X_1 =largo sépalo, X_2 =ancho sépalo, Y =ancho pétalo.

   (c) X_1 =largo pétalo, X_2 =ancho pétalo, Y =largo sépalo.

   (d) X_1 =largo pétalo, X_2 =ancho pétalo, Y =ancho sépalo.

   ¿En qué combinación obtiene el menor error ?

[2]Notar que esto es diferente al Problema 1, en donde cada columna representa un dato.