Tarea 2 Reconocimiento de Patrones

A. Lectura y animaciones (no hay que entregar nada)

1. A aquellos que se sienten aun no muy familiarizados con análisis de datos, recomiendo leer la parte del libro `Applied Multivariate Analysis` sobre un análisis de PCA de los datos de (otro) heptatlon a partir de pag. 78 (pag. 92 en el pdf).
   Nota: en el libro se usa `prcomp` y no `princomp`. Ambos calculan PCA; la diferencia es más bien en el método númerico subyacente que se usa: `prcomp()` usa SVD y `princomp()` usa la matriz de covarianza. En general se considera que desde punto numérico `prcomp()` es mejor pero es más dificil sacar las proyecciones y scores. Si `objeto<- prcomp()`, entonces `objeto$rotation[,1]` es el equivalente a lo que da loadings[,1] con `princomp()`.

B. Preguntas cortas

1. Sea $\{x_i\}$ un conjunto de n vectores d dimensional. Definimos la matriz Kernel $[\mathbb{K}_{i,j}]$ con $\mathbb{K}_{i,j} = \langle x_i, x_j \rangle$ y $\mathbb{D}^2$ la matriz de distancias al cuadrada correspondiente.

   Verifica la identidad que usamos en clase:

   $$\mathbb{D}^2 = c1^t + 1c^t - 2\mathbb{X}\mathbb{X}^t,$$

   con 1 un vector de unos de longitud n y c el vector de longitud n con elementos $(\mathbb{K}_{i,i})_{i=1}^n$

2. En la página 18 del archivo `recpat6.pdf` de la clase del 9 de febrero, verifica cómo que se obtiene la expresión $\mathbb{K}_\Phi(x,y) = (1 + \langle x, y \rangle)^2$.

   De manera similar, supongamos que se define otro kernel K:

   $$K(x,y) = \langle x, y \rangle^3 \quad x, y \in \mathcal{R}^2$$

   Busca una función $\Phi()$ tal que:

   $$K(x,y) = \langle \Phi(x), \Phi(y) \rangle$$

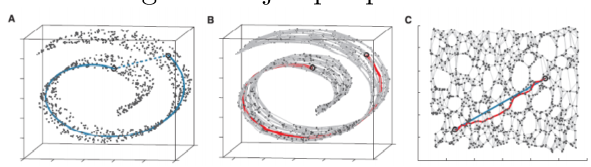
3. Sea S un conjunto finito. Definimos como medida de similitud entre dos subconjuntos A y B de S:

$$K(A,B) := \#(A \cap B)$$

Busca una función $\Phi()$ tal que:

$$K(A,B) = \langle \Phi(A), \Phi(B) \rangle$$

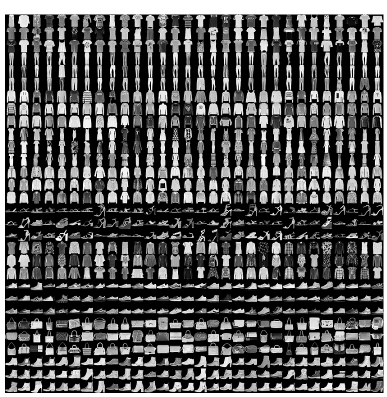
4. (hacer después de la clase miércoles)
Vimos el siguiente ejemplo para ilustrar ISOMAP



Decidimos que dos observaciones x_i, x_j son conectados por una arista en el grafo correspondiente ssi x_i está entre los k-vecinos más cercanos de x_j o x_j está entre los k-vecinos más cercanos de x_i. Muestra que la adición de una sola observación en este ejemplo puede destruir por completo el *desenrollamiento*. Márcala en el dibujo y explícalo.

C. Análisis de datos

1. Trabajamos con los de datos `fashion MNIST`.
Ver `https://www.kaggle.com/zalando-research/fashionmnist` Se trata de imágenes 28x28 de diez diferentes tipos de prendas.
Trabajaremos con `fashion-mnist_train.csv`



Código en R:

```
# definir primero  el working directory
train.data <- read.csv("fashion-mnist_test.csv")
# primera columna: indica el tipo de prenda
##0 T-shirt/top
##1 Trouser
##2 Pullover
##3 Dress
##4 Coat
##5 Sandal
##6 Shirt
##7 Sneaker
##8 Bag
##9 Ankle boot
# las siguientes 28*28 columnas: valores pixeles

#para mostrar una imagen particular:
rotate <- function(x) t(apply(x, 2, rev))
m<-matrix(t(train.data[600,1+(1:(28*28))]),ncol=28)
image(rotate(rotate(m)),col = grey(seq(0, 1, length = 256)))
```

Busca visualizaciones 2D y 3D basadas en PCA de las imágenes de T-shirts (clase "0"). ¿ Ves posible encontrar *interpretaciones* de los componentes como lo hicimos en clase con la base mnist (clásico) de dígitos?

(hacer después de la clase de miércoles) Compara el resultado con ISOMAP.

2. (este ejercicio puedes hacer en equipos de dos)
Trabajamos con datos de calificaciones de peliculas de Netflix por usuarios:
`https://grouplens.org/datasets/movielens/latest/`
Nos limitamos a la base chiquita.

Busca algunas visualizaciones informativas de estos datos y coméntalos (no se trata de hacer un análisis completo).

Aplica MDS para obtener una visualización de las películas, explora diferentes kernels (basándose en el vector de calificaciones de cada pelicula y/o los géneros a los cuales cada pelicula pertenece).

Hay muchísimas calificaciones faltantes. Limítate a un subconjunto chiquito que se puede trabajar facilmente.

En $\mathcal{R}$ una función útil para construir una matriz kernel es a partir `outer(v,w,f)` : se aplica la función f a las entradas de los vectores v y w, i.e. $f(v_i, w_i)$, y se regresa una matriz con los resultados. Para el equivalente en Python, ver por ejemplo:
`https://stackoverflow.com/questions/20061955/use-outer-function-with-fun-in-python`

Otras funciones útiles: `toString(), strsplit()`.