



TAREA 3

Entrega el 1 de abril

Tarea 3. Máquinas de soporte vectorial con *kernels*

- Implementar la máquina de soporte vectorial usando *kernels* no lineales detallada en la diapositiva 35. Se programarán las siguientes funciones:
 - `K = kernel(A,B,keropt,params)`, es la función que realiza la transformación no lineal de los datos entre las matrices `A` y `B` dado una opción de *kernel*, `keropt`, que puede ser `'lin'`, `'poly'`, y `'rbf'`, cuyos parámetros estarán en `params`, que define el orden para `poly` y `gamma` para `rbf`. La salida `K` es la matriz `kernel`.
 - `[sv,alpha_y,w0] = trainsvm(X,Y,params)`, es la función que entrena la SVM, donde `X` son los patrones de entrenamiento, `Y` la clase (-1 ó $+1$), `params` son los parámetros que indican el tipo de `kernel`, el parámetro de la función `kernel`, y la variable `C` de la SVM. Las salidas son los vectores de soporte `sv`, los valores $\alpha \cdot Y$ asociados a los vectores de soporte `sv`, y el valor del bias `w0`.
 - `[Yp,e] = classifysvm(X,Y,sv,alpha_y,w0,params)`, es la función que clasifica usando SVM donde `X` son los patrones de prueba, `Y` la clase verdadera (-1 ó $+1$). Las salidas son `Yp` que son las clases predichas por la SVM y `e` el porcentaje de error.
- Las funciones anteriores se ejecutarán desde un archivo principal para clasificar conjuntos de datos reales. Para encontrar los mejores parámetros de la SVM se hará una búsqueda en malla de ambos valores dentro de un rango (por ejemplo, `gamma=[0.01 0.1 0.5 1]` de la RBF y `C=[1, 10, 100, 1000]`). En cada paso de la búsqueda el conjunto original se dividirá en entrenamiento (70%) y prueba (30%) y se considerarán 31 ejecuciones, de modo que los parámetros más adecuados están definidos en la media del porcentaje de error más baja.
- Reporte de la práctica.