# Mixturas de gaussianas

## mixgaussian.m Ej 2.1

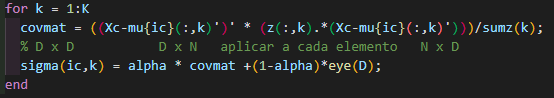
Se calcula la probabilidad a priori para cada índice de la clase y se normaliza respecto al número de clases. No se utiliza el operador ‘.’ porque octave aplica la normalización a toda la matriz.



La media para cada índice se calcula transponiendo la muestra de la clase c para que las dimensiones se ajusten a las requeridas para multiplicarlo por zk, y a cada elemento se le divide entre el sumatorio de zk para normalizar.

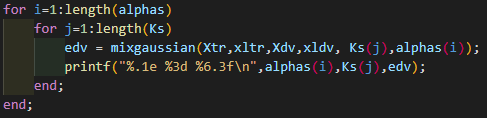


Se aplica la matriz de covarianzas a cada componente k. Finalmente se aplica un suavizado.



## mixgaussian-exp.m Ej 2.2

Se recorren tanto las alphas como las Ks y se guardan en edv. Para posteriormente imprimir la tasa de error.



## pca+mixgaussian-exp.m Ej 2.3

Aplicamos PCA sobre el conjunto de entrenamiento. Así obtenemos la matriz de proyección completa (W) y las medias (m).

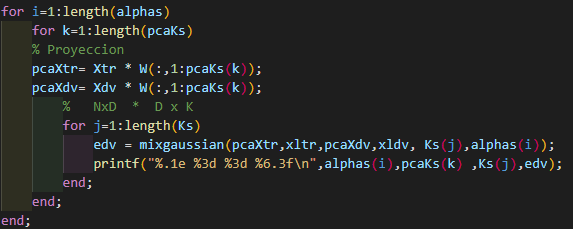


A cada dato se le resta su media (en ambos conjuntos).

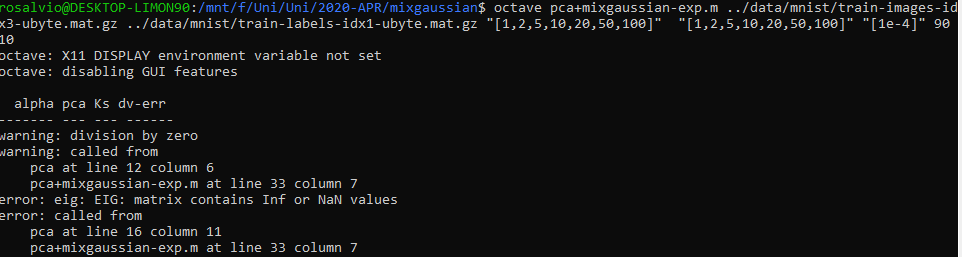


Se calculan los nuevos conjuntos en función de la proyección en PCA y luego se pasa como parámetro a mixgaussian.

A diferencia de mixgaussian-exp.m aquí se recorren también todas las dimensiones del PCA.



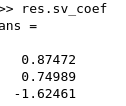
No he podido continuar con el experimento debido a que no me funciona el pca.



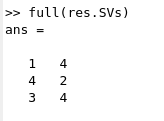
# Máquinas de vectores de soporte

## Ej 3.2

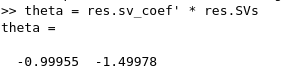
### Linealmente separable



Multiplicadores de Lagrange:



Vectores de soporte:

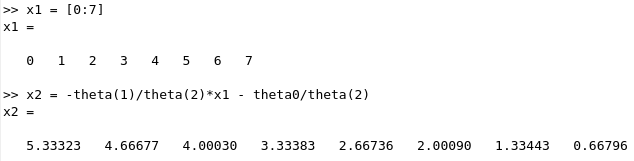


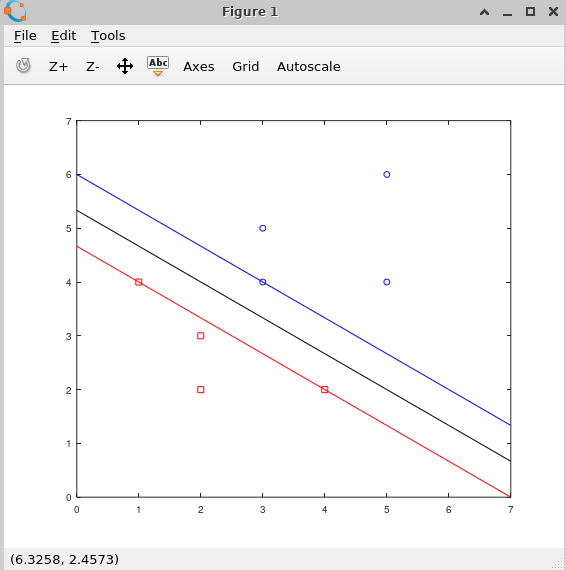
Vector de pesos:





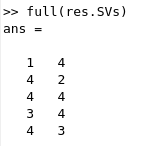
Margen:

Recta de separación: 

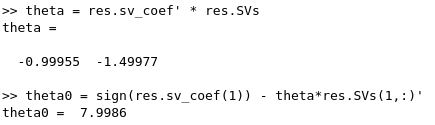


### No separable

Multiplicadores de Lagrange:

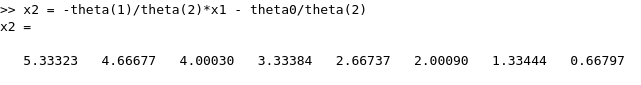


Vectores de soporte:

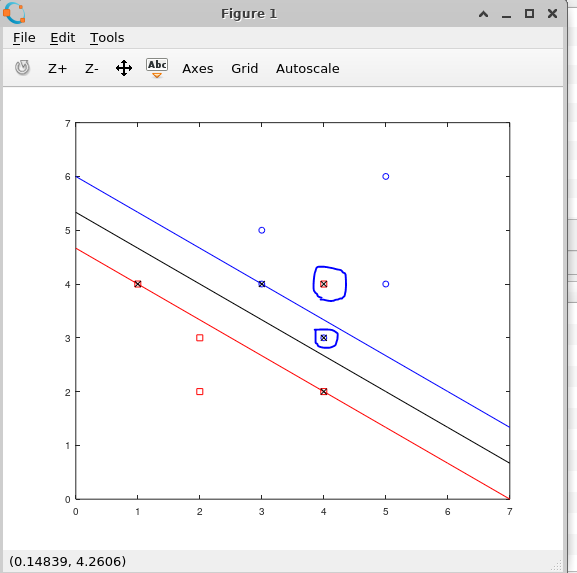


Vector de pesos:

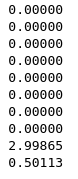
Margen:



Recta de separación:



Redondeados los vectores de soporte erróneos



Tolerancia de margen:

# Redes neuronales multicapa