

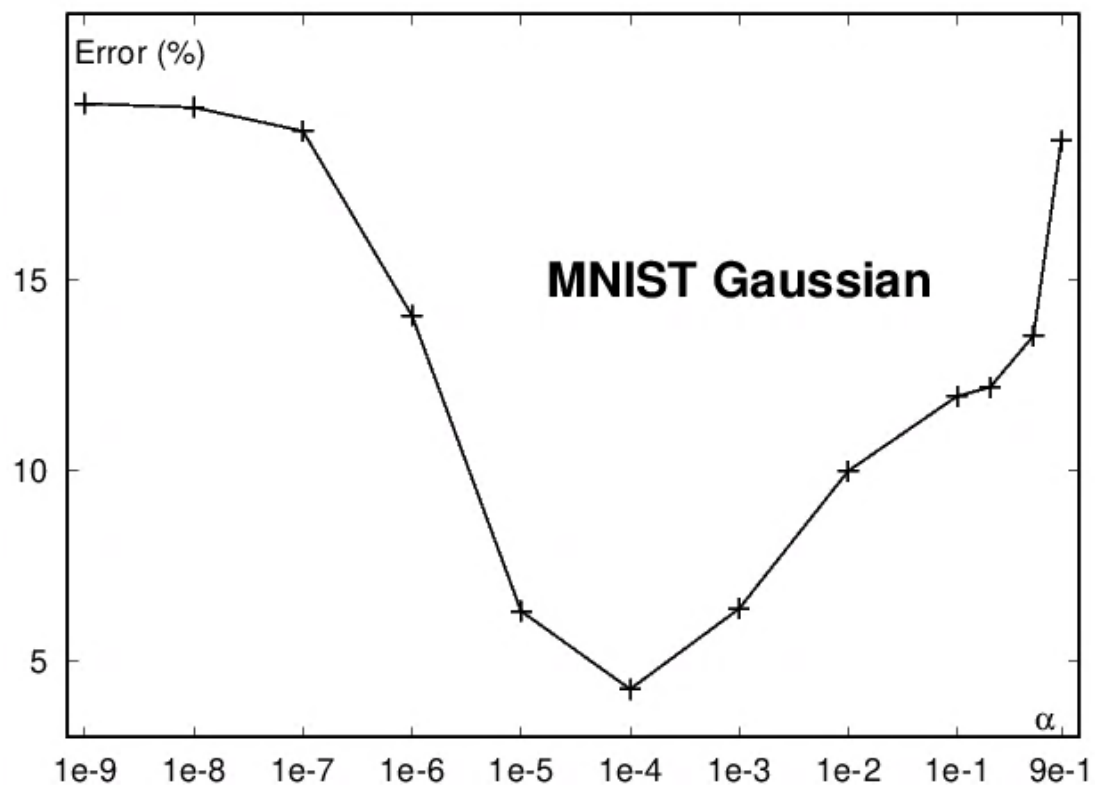
Entregable I: Mixturas de *Gaussianas*

Alumnos:

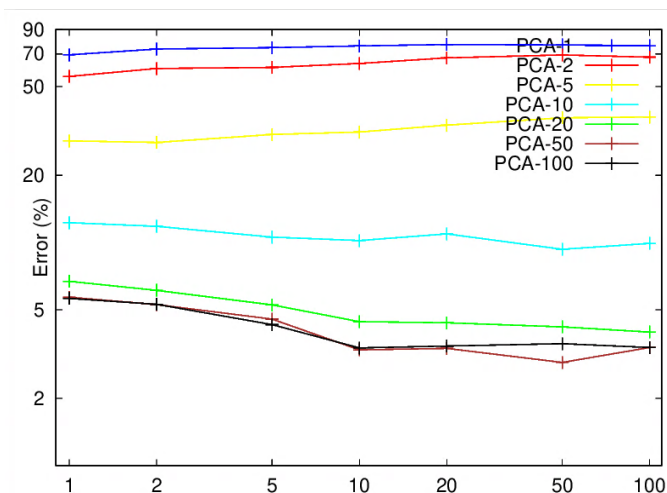
- **Luis Alberto Álvarez Zavaleta**
- **David Arnal García**

1.

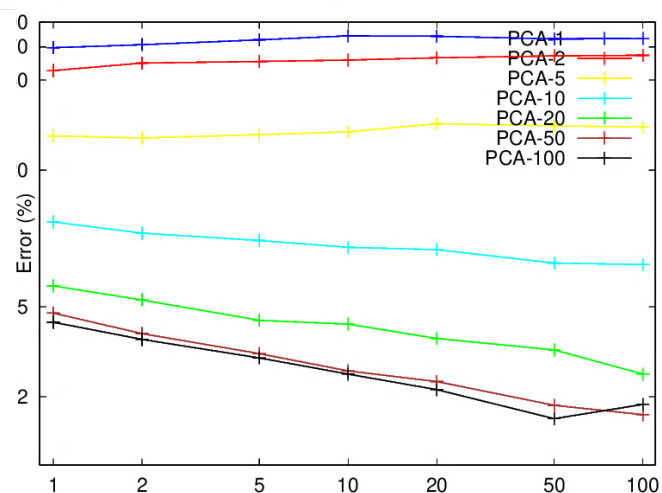
error	9e-1	5e-1	2e-1	1e-1	1e-2	1e-3	1e-4	1e-5	1e-6	1e-7	1e-8	1e-9
alpha	18.683	13.55	12.2	11.967	10	6.383	4.267	6.317	14.083	18.933	19.55	19.65



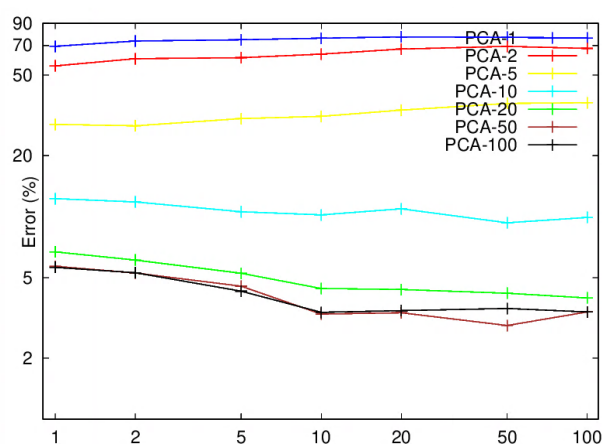
Como podemos observar gracias a la tabla y al gráfico, el valor por *flat smoothing* que mejor resultado da es el **1e-4**, y siendo sus valores más cercanos, el **1e-3** y el **1e-5** los que más se acercan al mejor valor. Los demás valores, ya muestran un error más pronunciado.



$\alpha = 1e-3$



$\alpha = 1e-4$



$\alpha = 1e-5$

Observando las gráficas, podemos concluir que la que mejor resultado obtiene es la de $\alpha = 1e-4$, PCA = 100 y K = 50 con un error de **1.6%**, siendo estos sus parámetros óptimos para el clasificador por mixtura de *gaussianas*. Aunque los valores con PCA = 50 y K = 100 obtienen **1.667%**, por lo que lo tendremos en cuenta para el siguiente apartado.

2.

Alpha	PCA	Ks	Error	Intervalo de confianza
1e-4	50	100	1.62	[1.618, 1.62247]
1e-4	100	50	1.92	[1.917, 1.92269]

Como podemos observar, el mejor resultado lo obtenemos con una reducción de dimensionalidad, $D = 50$, con 100 distribuciones gaussianas, $K_s = 100$, y con un valor de suavizado, $\alpha = 1e-4$.

Además, obtenemos que, con una confianza de 95%, podemos asegurar que su error se encontrará entre el intervalo [1.618, 1.62247].

Comparándolo con el clasificador gaussiano, y el clasificador *PCA + gaussiano* con 200 dimensiones, se puede ver como el resultado es mejor, ya que hemos mejorado nuestro clasificador reduciendo el error en un 2.49% al utilizar las mixturas de *Gaussianas*, y un 1.68% comparado con los resultados de *MNIST*.

Alpha	Error
1e-4 (sin PCA)	4.18
1e-4 ($D = 200$)	4.11

3.

Implementaremos la terminación temprana para el algoritmo EM en el que haremos que termine cuando el error de validación/test no disminuya de una iteración a la siguiente. Para ello, modificaremos *mixgaussian.m* de la siguiente manera:

```
it = 0;
lasterrY = inf;
printf(" It          oL          L  errX  errY\n");
printf(" ---  -----  -----  ----  ----\n");
do
    oL = L;
    L = 0;
    % For each class
    for c = classes'

% Classification of training and evaluation sets and error estimation
    [~, idX] = max(gX');
    errX = mean(classes(idX) != x1) * 100;
    [~, idY] = max(gY');
    errY = mean(classes(idY) != y1) * 100;
    it = it + 1;
    printf("%3d %11.5f %11.5f %5.2f %5.2f\n", it, oL, L, errX, errY);

    if (errY >= lasterrY)
        errY = lasterrY;
        break;
    endif

    lasterrY = errY;
until ((L - oL) / abs(oL) < epsilon)
end
```

Al evaluar esta nueva implementación en los conjuntos oficiales de *MNIST* para los valores óptimos, $PCA = 50$ y $K = 100$, obtenemos los siguientes resultados.

Alpha	PCA	Ks	Error	Intervalo de confianza
1e-4	50	100	1.6	[1.598, 1.6]