

Proyecto Percepción 2

En este proyecto se realizará un estudio de los resultados de error en los algoritmos multinomial y gaussiano para clasificar puntos en un espacio bidimensional, además del efecto del algoritmo de P.C.A. sobre el error del clasificador gaussiano.

Multinomial: es un clasificador que emplea el clasificador de Bayes sobre el que la f.d. sigue la distribución multinomial, y se trata de un clasificador lineal.

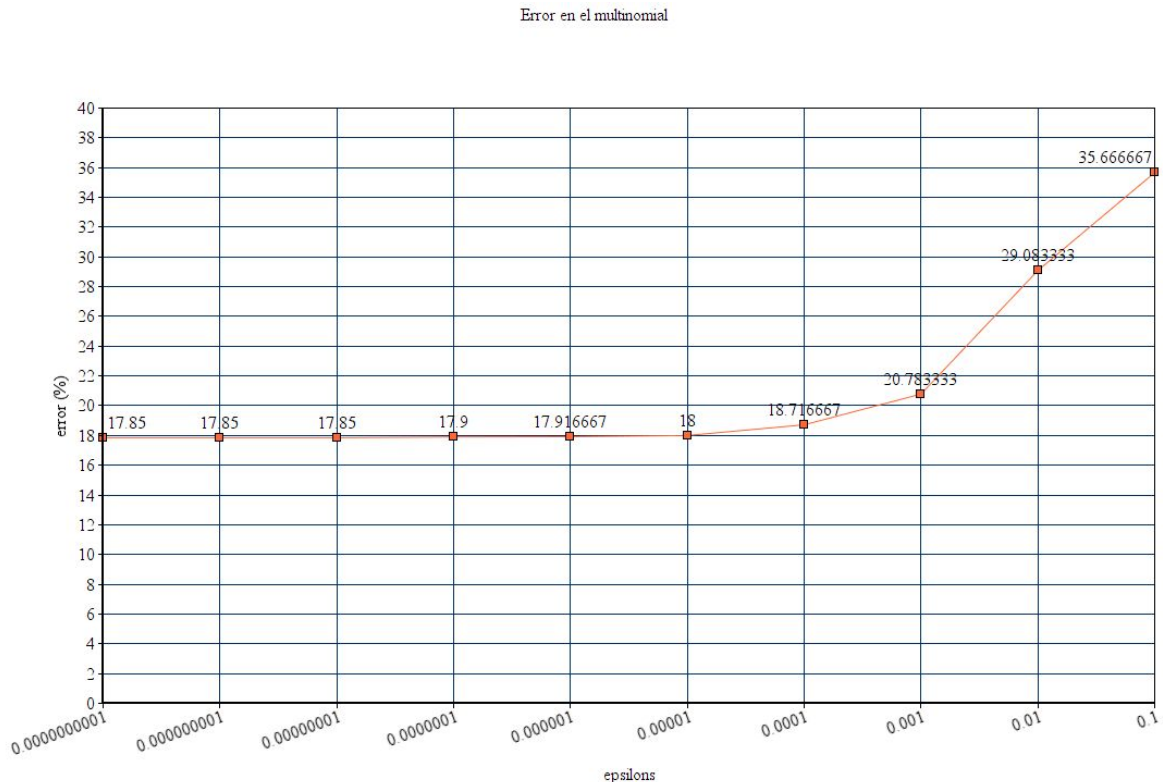
El código de la función que lo aplica está en el archivo **multinomial.m** adjunto con este documento.

Aplicaremos los cálculos sobre un conjunto con 60000 muestras de las cuales extraemos el 90% para entrenamiento y el 10% para test que están en los archivos **train-images-idx3-ubyte.mat.gz** y **train-labels-idx1-ubyte.mat.gz** adjunto con este documento.

Ahora probaremos a aplicar el multinomial para calcular el error variando el epsilon (un dato empleado en multinomial.m para evitar que las probabilidades se hagan 0) y representando el error de evaluación por cada epsilon.

El código que aplica esto está en el archivo **multinomial-exp.m** adjunto con este documento.

A continuación un gráfico con los resultados:



Dado que el error mínimo es el mismo tanto en el caso de epsilon $1e-9$, $1e-8$ y $1e-7$ usaremos este último por tener menos decimales (dado que dan el mismo valor no aportará mucho un perfilado más exacto).

Ahora vamos a evaluar este clasificador multinomial con el mejor epsilon ($1e-7$) sobre un conjunto de evaluación diferente para ver cómo se comporta, estas son los archivos **t10k-images-idx3-ubyte.mat.gz** y **t10k-labels-idx1-ubyte.mat.gz** adjuntos con este documento.

El código que aplica esto está en el archivo **multinomial-eva.m** adjunto con este documento.

Obtenemos así un error multinomial con el mejor epsilon (0.00000001) de 16.36% mucho más alto que los clasificadores knn, knn+pca o knn+pca+wilson vistos en la anterior entrega. Muy mayor a cualquier valor obtenido en MNIST, por lo que podemos concluir que el clasificador multinomial no es muy bueno para este tipo de clasificación.

Los datos de MNIST se encuentran en la página: <http://yann.lecun.com/exdb/mnist/>

Gaussiano: es un clasificador que emplea el clasificador de Bayes sobre el que la f.d. sigue la distribución gaussiana, y se trata de un clasificador cuadrático.

El código de la función que lo aplica está en el archivo **gaussiano.m** adjunto con este documento.

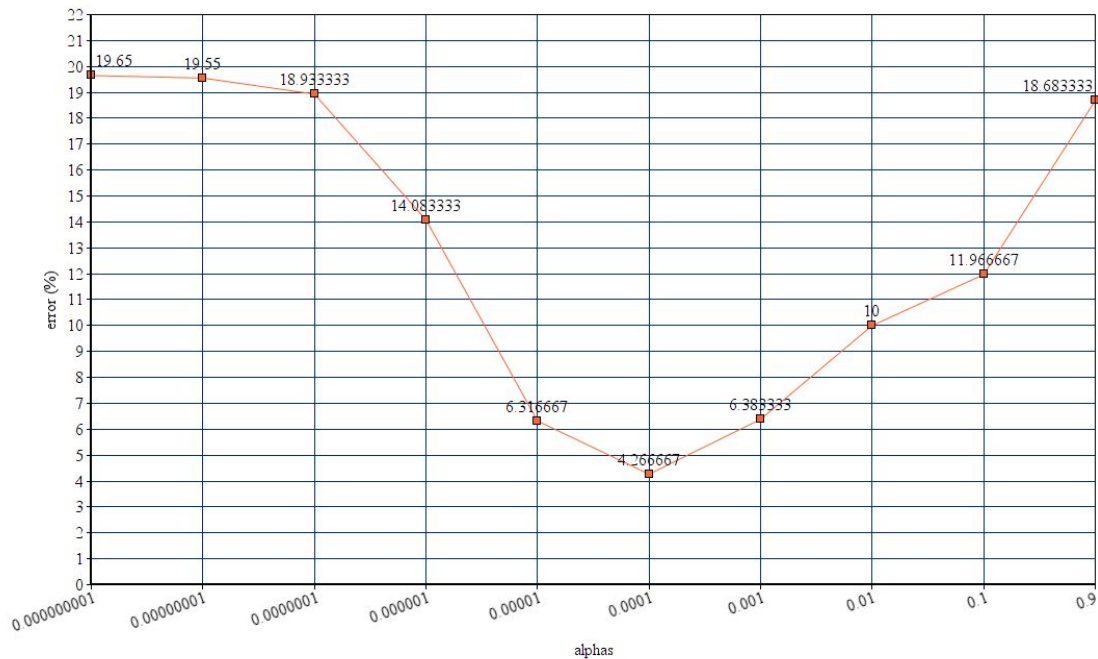
Aplicaremos los cálculos sobre un conjunto con 60000 muestras de las cuales extraemos el 90% para entrenamiento y el 10% para test que están en los archivos **train-images-idx3-ubyte.mat.gz** y **train-labels-idx1-ubyte.mat.gz** adjunto con este documento.

Ahora probaremos a aplicar el multinomial para calcular el error variando el alpha (un dato empleado en gaussiano.m para suavizar los datos en las matrices sigma de cada clase) y representando el error de evaluación por cada alpha.

El código que aplica esto está en el archivo **gaussiano-exp.m** adjunto con este documento.

A continuación un gráfico con los resultados:

Error gaussiano



El valor de alpha con error mínimo es 0.0001, tras un perfilado mejor se ha obtenido un alpha distinto pero tras estudiarlo en el conjunto de evaluación de t10k ha dado peor que 0.0001 por ello he decidido no incluirlo.

Ahora vamos a evaluar este clasificador gaussiano con el mejor alpha ($1e-4$) sobre un conjunto de evaluación diferente para ver cómo se comporta, estas son los archivos **t10k-images-idx3-ubyte.mat.gz** y **t10k-labels-idx1-ubyte.mat.gz** adjuntos con este documento.

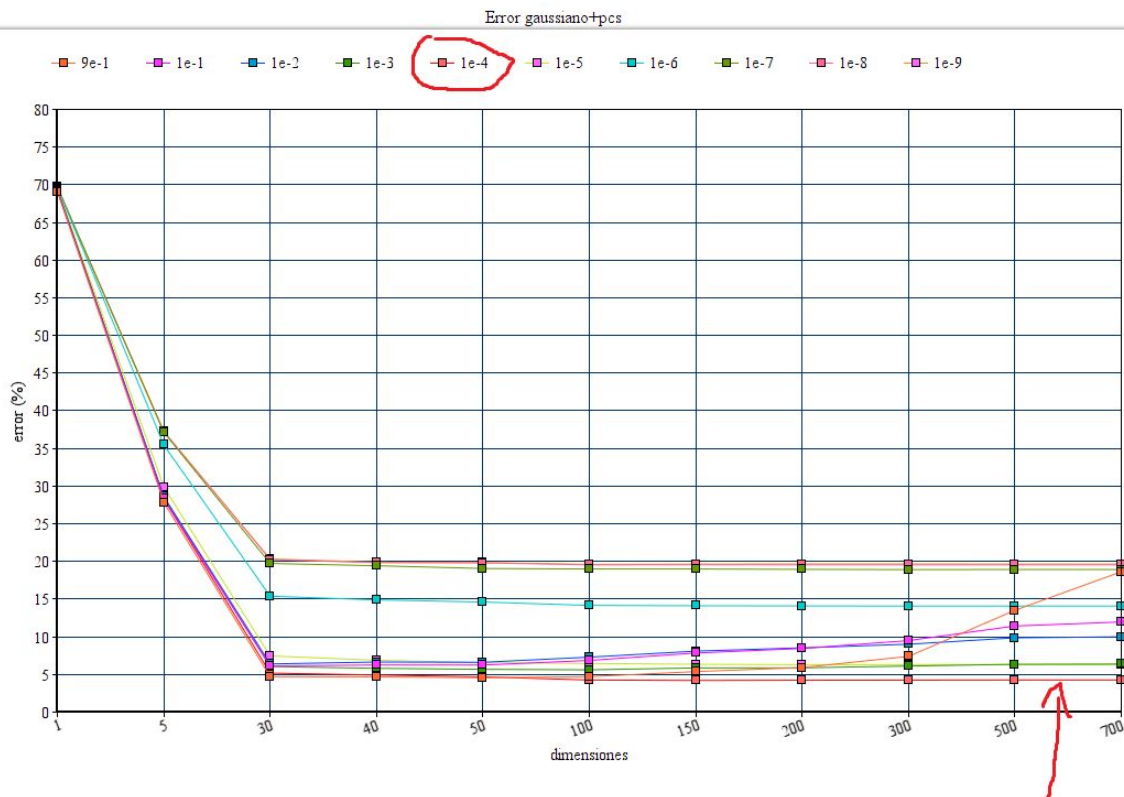
El código que aplica esto está en el archivo **gaussiano-eva.m** adjunto con este documento.

Obtenemos así un error gaussiano con el mejor epsilon (0.0001) de 4.18% aunque todavía es alto pero es similar al obtenido por MNIST con P.C.A. y un clasificador cuadrático con 3.3% de error obtenido por LeCun en 1998, teniendo en cuenta que aquí no se ha aplicado P.C.A., cosa que haremos a continuación.

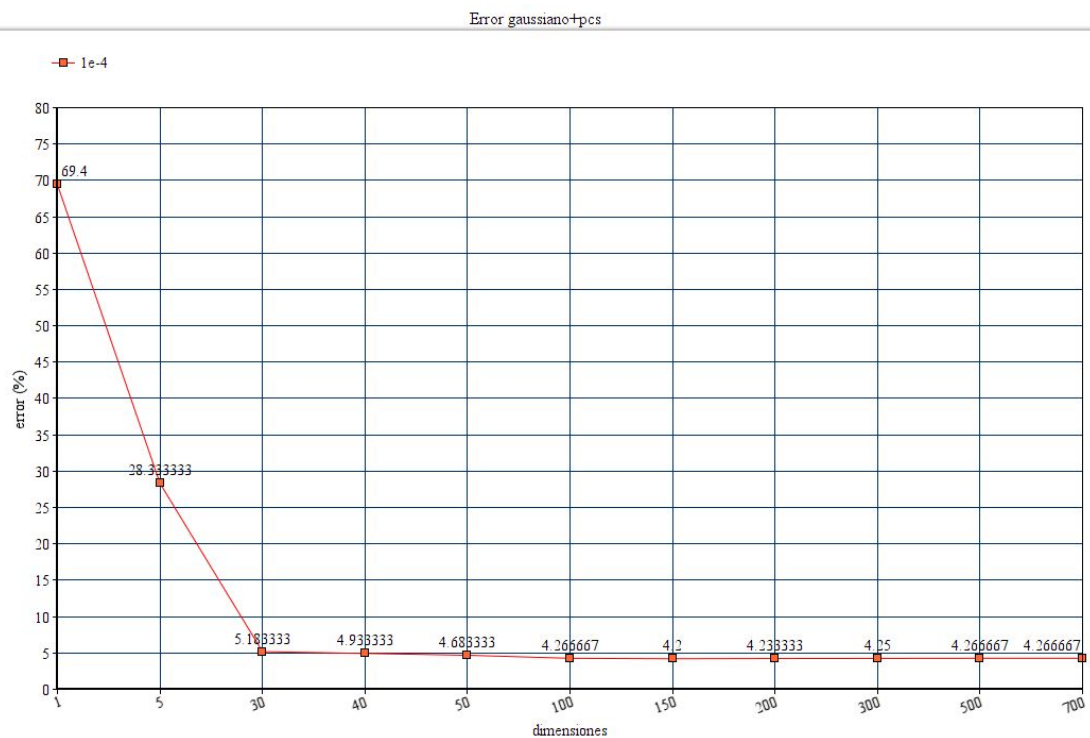
Aplicaremos los cálculos sobre un conjunto con 60000 muestras de las cuales extraemos el 90% para entrenamiento y el 10% para test que están en los archivos **train-images-idx3-ubyte.mat.gz** y **train-labels-idx1-ubyte.mat.gz** adjunto con este documento.

Vamos a estudiar como varía el error en función del alpha y de la dimensión de P.C.A.

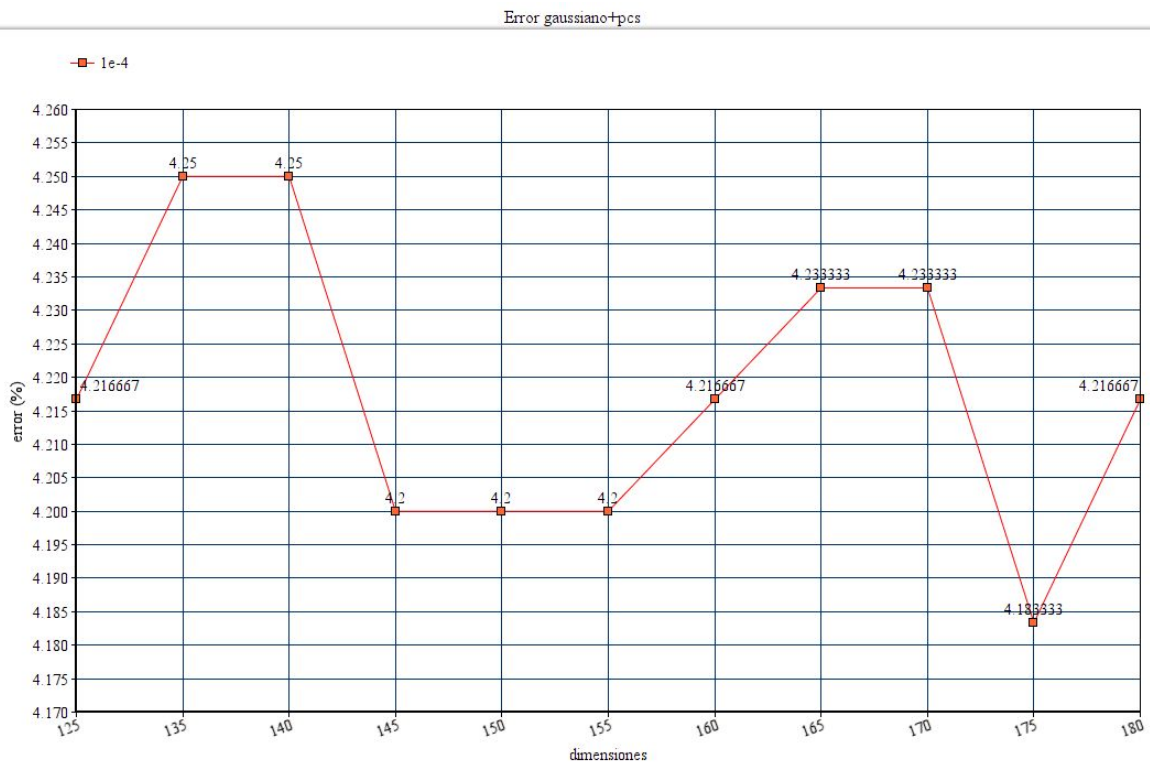
El código que aplica esto está en el archivo **gaussiano+pca-exp.m** adjunto con este documento.



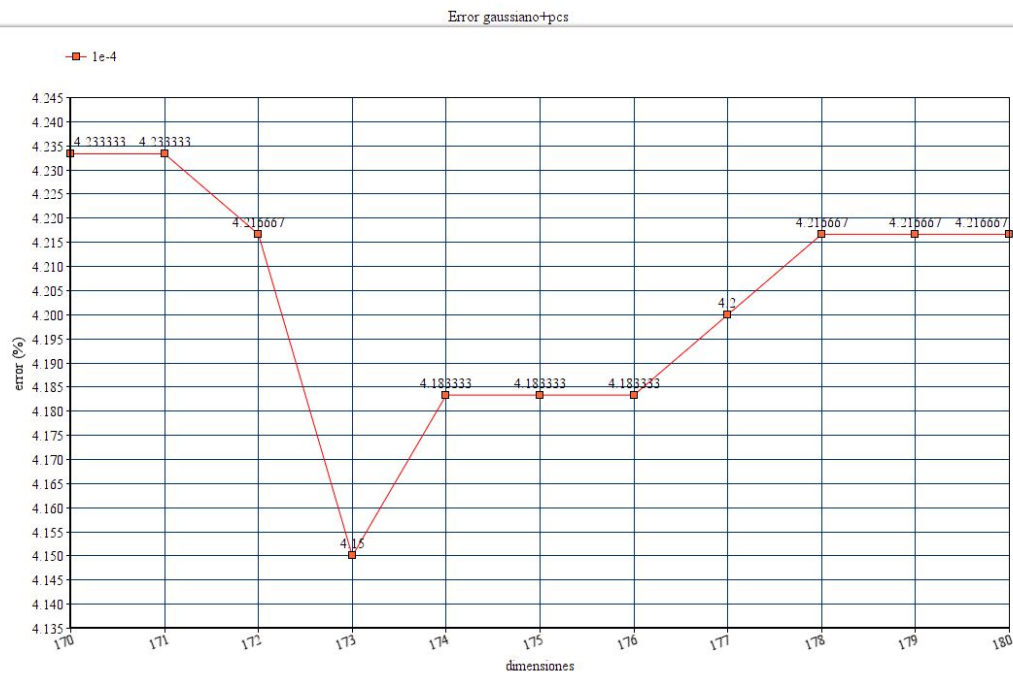
Esta es la gráfica con todas las alphas y dimensiones probadas está marcada la de error mínimo 1e-4 vamos a verla más de cerca.



El mejor valor de dimensiones es 150 vamos a estudiar los valores cercanos para terminar de perfilarlo



Vemos que el valor de menor error ha variado a 175, vamos a hacer un último para mayor exactitud.



Por último deducimos que el valor de menor error es de 173

Ahora vamos a evaluar este clasificador gaussiano con el mejor α ($1e-4$) y la mejor reducción de dimensionalidad 173 sobre un conjunto de evaluación diferente para ver cómo se comporta, estas son los archivos **t10k-images-idx3-ubyte.mat.gz** y **t10k-labels-idx1-ubyte.mat.gz** adjuntos con este documento.

El código que aplica esto está en el archivo **gaussiano+pca-eva.m** adjunto con este documento.

Error gaussiano con el mejor α (0.0001) y el mejor número de dimensiones según P.C.A.(173): 4.11% menor que sin aplicar pca (4.18%) pero nada significativo como para parecerse al 3.3% de MNIST

Gracias por su atención