Andrés Marín Galán

Proyecto Percepción

En este proyecto se realizará un estudio de los resultados de error en los algoritmos de P.C.A. y de k-nn para clasificar puntos en un espacio bidimensional, además del efecto del algoritmo de edición de wilson sobre el error sin aplicar P.C.A. y tras aplicarlo.

P.C.A.: Es una técnica de reducción de dimensionalidad no supervisada que preserva la mayor parte de la varianza de los datos, se asume que incluye la capacidad de discriminar entre clases, su objetivo es encontrar una matriz de proyección W que minimice el error de reconstrucción.

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

K-nn: Este algoritmo clasifica los puntos según el número de puntos más cercanos que le digamos según un número k, calculara los k puntos más cercanos y clasificará el punto en la clase que tenga mayor número de puntos en esos k.

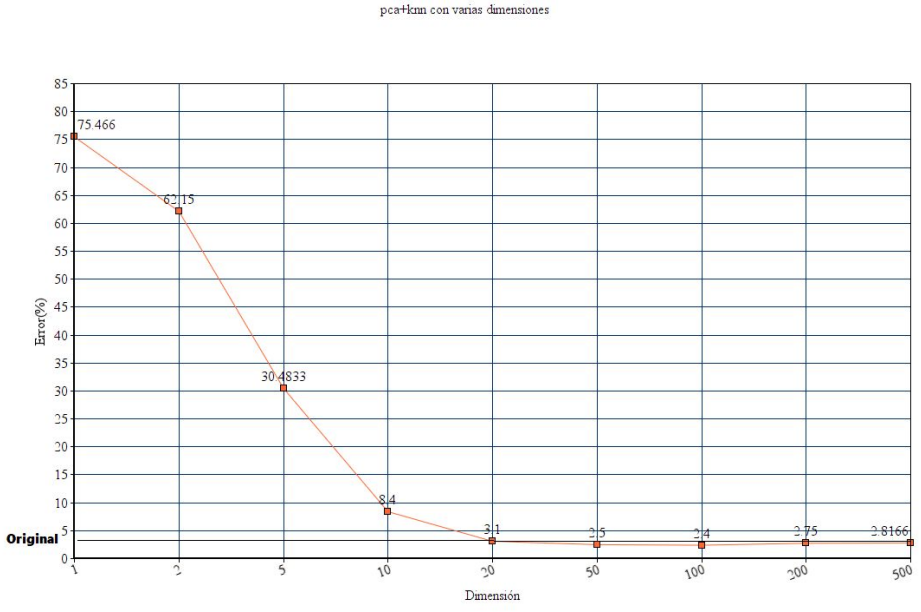
El código de la función que calcula el error dado unos puntos de entrenamiento y otros de evaluación está en el archivo **knn.m** adjunto con este documento y proporcionado por el profesor.

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.

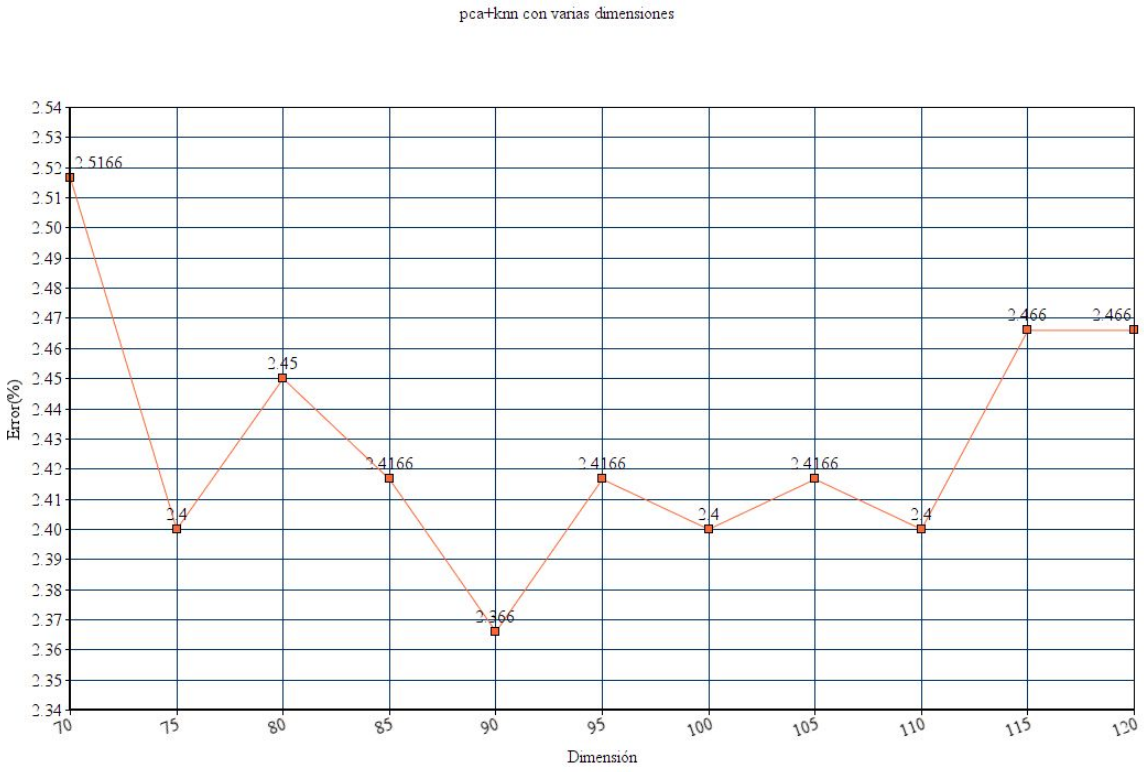
Ahora probaremos a aplicar P.C.A. y knn para calcular el error variando la reducción dimensionalidad de P.C.A. y empleando k-nn con k=1.

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

A continuación un gráfico con los resultados:



El mejor valor de este estudio es 100 vamos a estudiar los valores cercanos para terminar de perfilarlo:



De estos resultados obtenemos que el mejor valor para reducir la dimensionalidad de los datos con P.C.A. para un clasificador k-nn con k=1 es 90, pues es la de mínimo error (2.366%) muy similar a los datos obtenidos por "K-nearest-neighbors, Euclidean (L2)" aplicando alineación (2.4%) Obtenido por LeCunt en 1998.

Ahora vamos a evaluar este clasificador con pca y sin pca 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 **pca+knn-eva.m** adjunto con este documento.

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

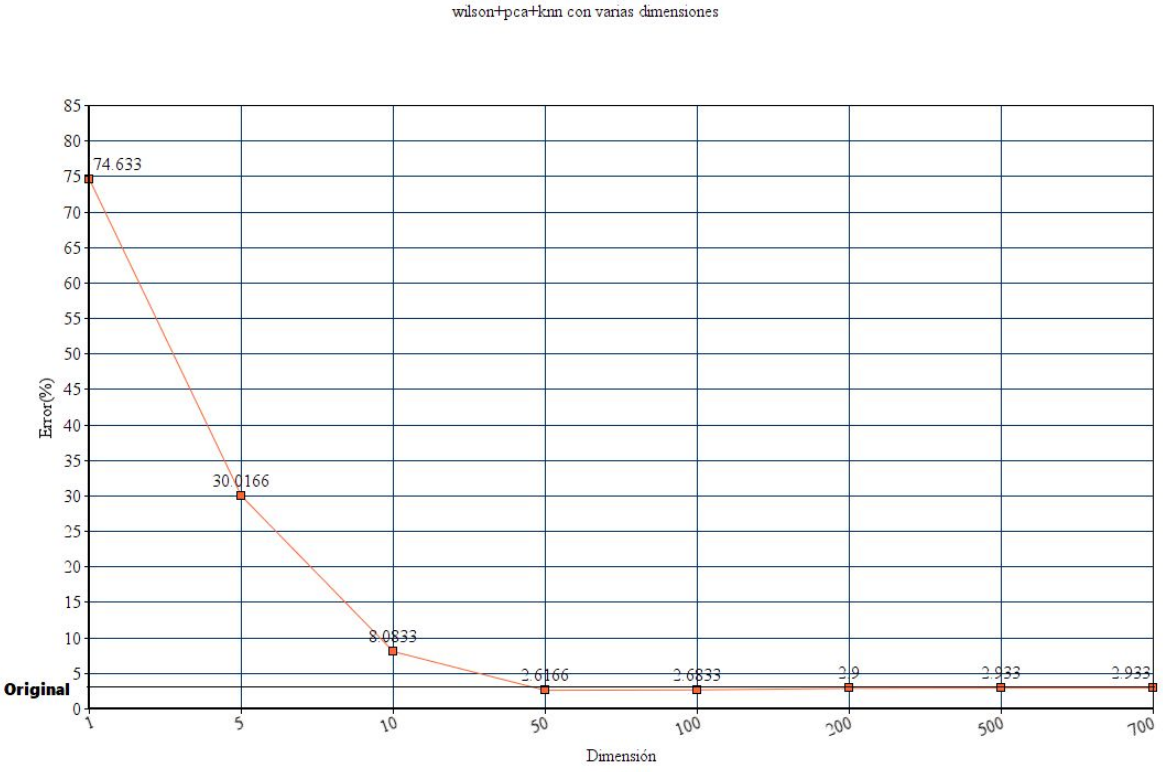
Obtenemos así:
Sin aplicar P.C.A. obtenemos un error de 3.09% vemos que el valor obtenido es idéntico al obtenido por MNIST por "K-nearest-neighbors, Euclidean (L2)" Obtenido por Kenneth Wilder de la universidad de Chicago.
Aplicando P.C.A. con reducción de 90 nos da un error de 2.7% similar a los datos obtenidos por "K-nearest-neighbors, Euclidean (L2)" aplicando alineación (2.4%) Obtenido por LeCunt en 1998. Vemos que P.C.A. ha reducido el error en un 0.39%

<u>Opcional</u>: Wilson
Ahora aplicaremos el algoritmo de wilson que elimina las muestras más cercanas a la frontera de decisión, lo aplicaremos igual para el 90% de entrenamiento y 10% de evaluación sobre los 60000 calculando el error con knn para k=1.

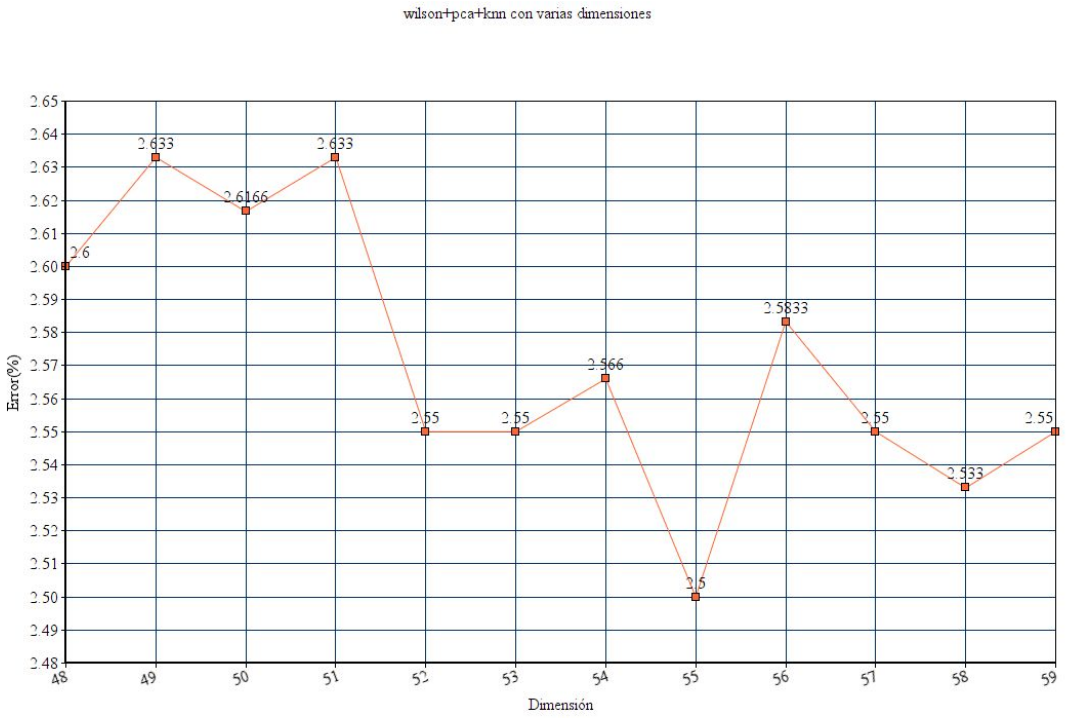
El código del algoritmo de wilson está en el archivo **wilson.m** y usa las funciones knn_class y knn_matrix que se encuentran en los archivos **knn_class.m** y **knn_matrix.m** respectivamente.

Y lo probamos con 90% entrenamiento 10% evaluación, sin aplicar pca únicamente calculamos el error con knn para k=1:
Vemos que elimina 1459 de 54000 (0.9*60000)
Y nos da un error de 2.933%

Por último aplicamos wilson para eliminar muestras, luego pca con varias dimensiones y el cálculo del error con knn para k=1.



El mejor valor para este estudio es 50 vamos a estudiar los valores cercanos para terminar de perfilarlo:



Vemos que la mejor dimensión de P.C.A. tras aplicar wilson es 55 en lugar de 90. Podemos concluir que con wilson el error es similar pero la dimensión de menor error es menor y lógicamente disponemos de menos puntos en el entrenamiento, quizá por esto el error es ligeramente mayor, para el mínimo entre un caso u otro.

Probamos con este valor para el conjunto de evaluación como hemos hecho con P.C.A.:
Wilson elimina 1576 puntos de 60000
Error con wilson, pca y knn: 2.73%
Error con wilson y knn: 3.21%

los archivos de evaluación son los mismos **t10k-images-idx3-ubyte.mat.gz** y **t10k-labels-idx1-ubyte.mat.gz** adjuntos con este documento.

Como hemos dicho con anterioridad para este caso aplicando wilson y knn(k=1) obtenemos un error de 3.21% y aplicando wilson pca con dimensión 55 y knn(k=1) obtenemos un error de 2.73% diferente a los datos obtenidos por “K-nearest-neighbors, Euclidean (L2)” aplicando alineación, eliminación de ruido y desenfoque (1.8%) Obtenido por Kenneth Wilder de la universidad de Chicago, la variación de 0.93% se puede deber a la aplicación de la técnica de desenfoque que no se ha aplicado aquí.

Gracias por su atención