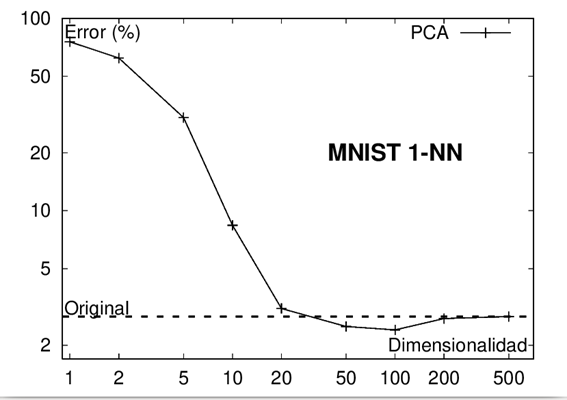
**Memoria PER**

**Ejercicio Obligatorio**

**Ejercicio *PCA*:**

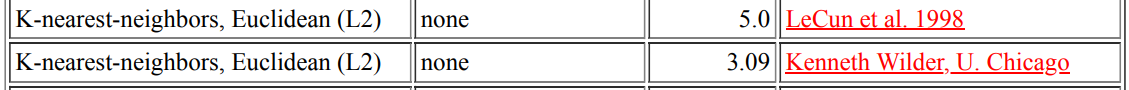
A continuación, tenemos la gráfica obtenida con *pca+knn-exp.m* utilizando, para ello, la distancia L2 implementada en el archivo *L2dist.m* del boletín y usando, a su vez, *knn.m* para calcular los vecinos más cercanos y *pca.m* para reducir su dimensionalidad. En dicho ejercicio, se nos planteó hacer un estudio sobre los distintos números de *k* a utilizar en nuestro k-medias con tal de encontrar el mejor valor para reducir los errores.



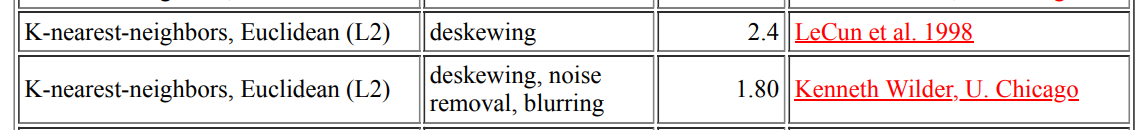
Resultados de *pca+knn-exp.m y pca+knn-eva.m:*

| ks | Error del clasificador |
| --- | --- |
| No *PCA* | 2.817 |
| 1 | 75.47 |
| 2 | 62.15 |
| 5 | 62.15 |
| 10 | 8.4 |
| 20 | 3.1 |
| 50 | 2.5 |
| 100 | 2.4 |
| 200 | 2.75 |
| 500 | 2.817 |
| Error de datos de evaluación con *PCA* | 2.84 |
| Error de datos de evaluación sin *PCA* | 3.09 |

Ahora compararemos los resultados que hemos obtenido con los disponibles en la página de *MNIST:*



Como podemos observar, al no aplicar ningún preproceso, nuestro clasificador nos da un error de 3.09%, lo mismo que el implementado por *Kenneth WIlder*, y proporciona 2% menos errores a la hora de clasificar en comparación con el de *LeCun.*



En cuanto al resultado aplicando *PCA*, nuestro clasificador da 2.84% de error, siendo 0.44% peor que el de *LeCun* y 1.04% peor que el de *Kennet Wilder.* En estos casos, dado que el porcentaje de error es muy cercano, habría que tener en cuenta, además del error obtenido, el coste computacional y temporal a la hora de ejecutar el algoritmo para decidir cuál de todos sería el mejor.