Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Introducción al Aprendizaje Automatizado

TRABAJO PRÁCTICO 4

Alumno: Rodríguez Jeremías

June 26, 2017

1 Ejercicio 1

Modifiqué nb-n.c para implementar el algoritmo k-nn (knn.c). El algoritmo recibe un parámetro Real extra, K, en el archivo de configuración .nb; y arroja el mismo output que el nb-n.c.

Las modificaciones, resumidamente, consistieron en reemplazar algunas matrices globales y usarlas para clasificar los nuevos puntos.

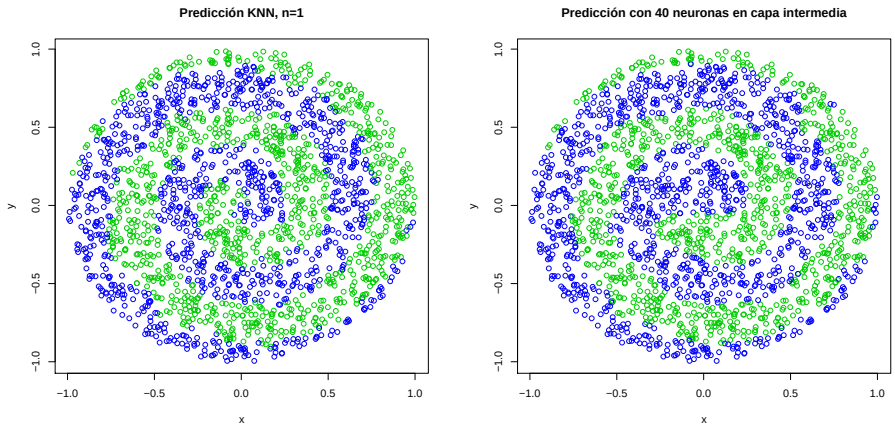
Para clasificar un punto cualquiera (sea del conjunto de training, test o validación) se procede a:

- Copiar todos los datos de training en una nueva matriz *distancias*
- Ordenar esta matriz, usando quicksort, en orden creciente respecto a la distancia euclideana al punto a clasificar.
- Elegir los k puntos a menor distancia (ignorar al punto mismo si se está clasificando en training) y hacer una votación de clases
- Clasificar al nuevo punto con la clase más votada

Esta implementación tiene una complejidad computacional considerable. Para implementar la búsqueda del k óptimo se utilizaron scripts de bash, corriendo knn.c para valores de k desde 1 hasta 50 y eligiendo el k cuya mediana de entre 7 ejecuciones presentara el mínimo error en validacón.

2 Ejercicio 2

Se resolvió el problema de las espirales anidadas usando knn con k=1 y la misma cantidad de puntos de training-test-validación que en el trabajo de redes neronales. Se grafica a continuación el conjunto de predicciones que generó la mediana en error de test para 21 ejecuciones; junto con la predicción en redes.



método	Error Test
Redes (40 neuronas)	8.450000
knn (n=1)	6.550000

Se puede ver que el método knn con n=1 funciona muy bien para este problema, incluso mejor que redes neuronales complejas. Los errores se cometen en puntos cercanos a la frontera real, donde el vecino más cercano puede ser de la clase equivocada. Esto podría intentar corregirse usando k más grande.

3 Ejercicio 3

En este ejercicio se utilizó knn para clasificar instancias del problema paralelo y diagonal con distintas dimensiones (2, 4, 8, 16, 32).

Para cada uno de los 20 problemas de una dada dimensión d, se ejecutó knn para valores de k variando entre 1 y 50. Para cada valor de k, se eligió la media del error de test de 7 ejecuciones; y luego el k óptimo según el error en validación.

La siguiente tabla muestra los resultados obtenidos en el prolema diagonal:

d	Test error % óptimo	Test error % k=1
2	10.690000 (k=8)	12.960000
4	11.930000 (k=3)	14.480000
8	11.440000 (k=33)	17.450000
16	11.740000 (k=25)	20.740000
32	13.430000 (k=43)	25.470000

La siguiente tabla muestra los mismos resultados pero del problema paralelo:

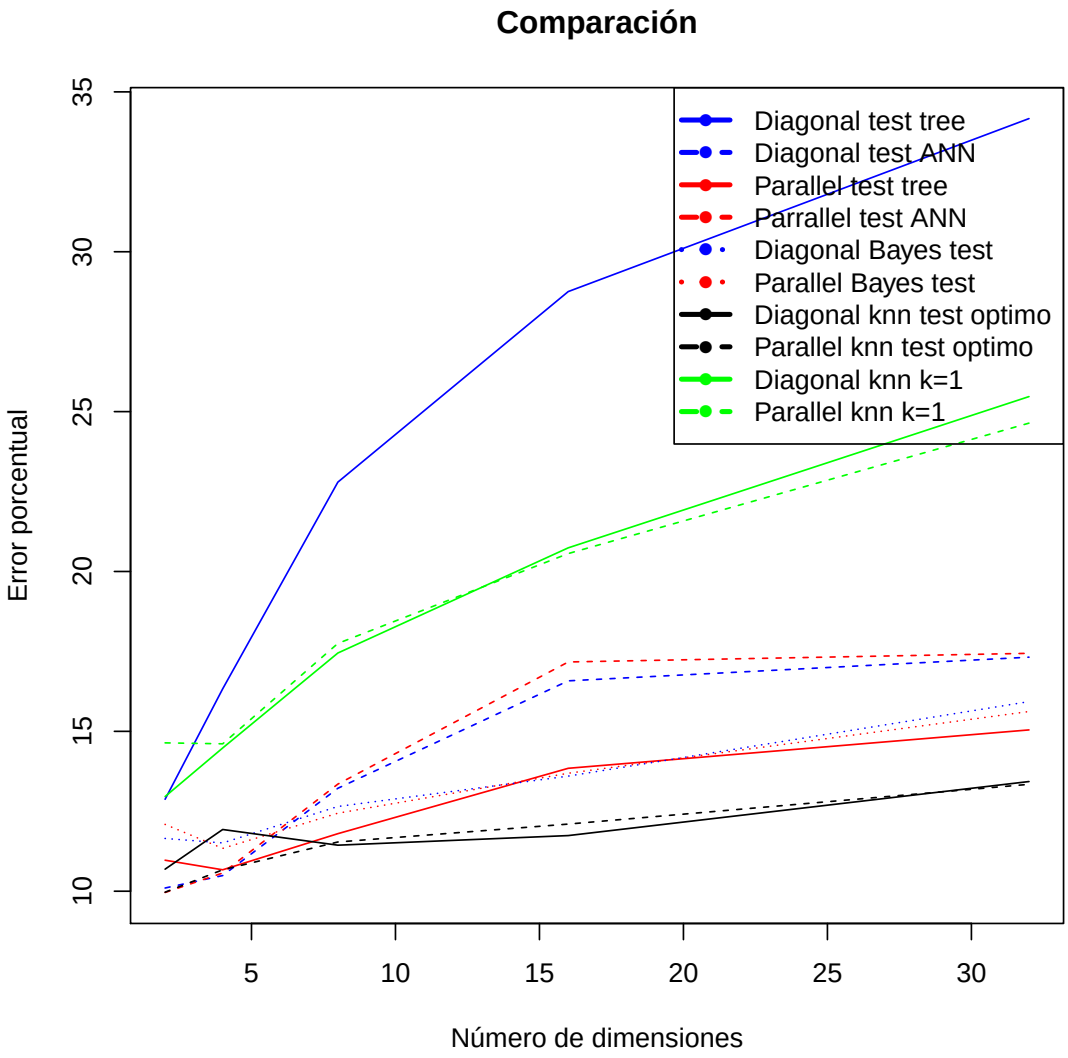
d	Test error % óptimo	Test error % k=1
2	9.970000 (k=16)	14.640000
4	10.670000 (k=13)	14.610000
8	11.540000 (k=22)	17.750000
16	12.100000 (k=18)	20.560000
32	13.430000 (k=41)	24.640000

A simple vista se puede ver que los errores no varían prácticamente entre el problema diagonal y paralelo. El aumento en la cantidad de dimensiones aumenta el error, aunque de forma suave si se elige el k adecuado.

Usar más de un vecino es mucho más eficiente que sólo uno, en especial cuantas más dimensiones hay. Esto sucede porque, aumentando el k, es más facil evitar errores producidor por puntos aislados; que son mucho más frecuentes en nuestros datasets con muchas dimensiones.

Concluyo que es fundamental elegir el k correcto.

Comparando con métodos anteriores:



Veo que este método es de los mejores para este tipo de problemas. Dada la complejidad computacional de mi algoritmo, las ejecuciones tardaban horas y puede que el orden de las curvas debiera ser ligeramente distinta.

4 Ejercicio 4

En este ejercicio se modificó knn.c de tal forma que, en vez de que voten los k vecinos más cercanos; voten los vecinos en un radio menor o igual a k.

El parámetro k es pasado de la misma forma que antes, y la modificación consistió en que, a la hora de clasificar un nuevo punto, símplemente se itera sobre todos los puntos de training y se los hace votar con su clase si la distancia euclideana es menor a K.

La verificación nuevamente se hace por fuera del algorimo, usando scripts de bash. Los valores candidatos de k probados son del orden de la distancia promedio entre los puntos de training.

Repitiendo el ejercicio 3 con este nuevo algoritmo, se generaron las siguientes dos tablas.

Diagonal:

d	Test error % óptimo
2	9.810000 (k=2.66)
4	10.410000 (k=4.89)
8	11.680000 (k=9.36)
16	12.320000 (k=18.08)
32	13.510000 (k=35.47)

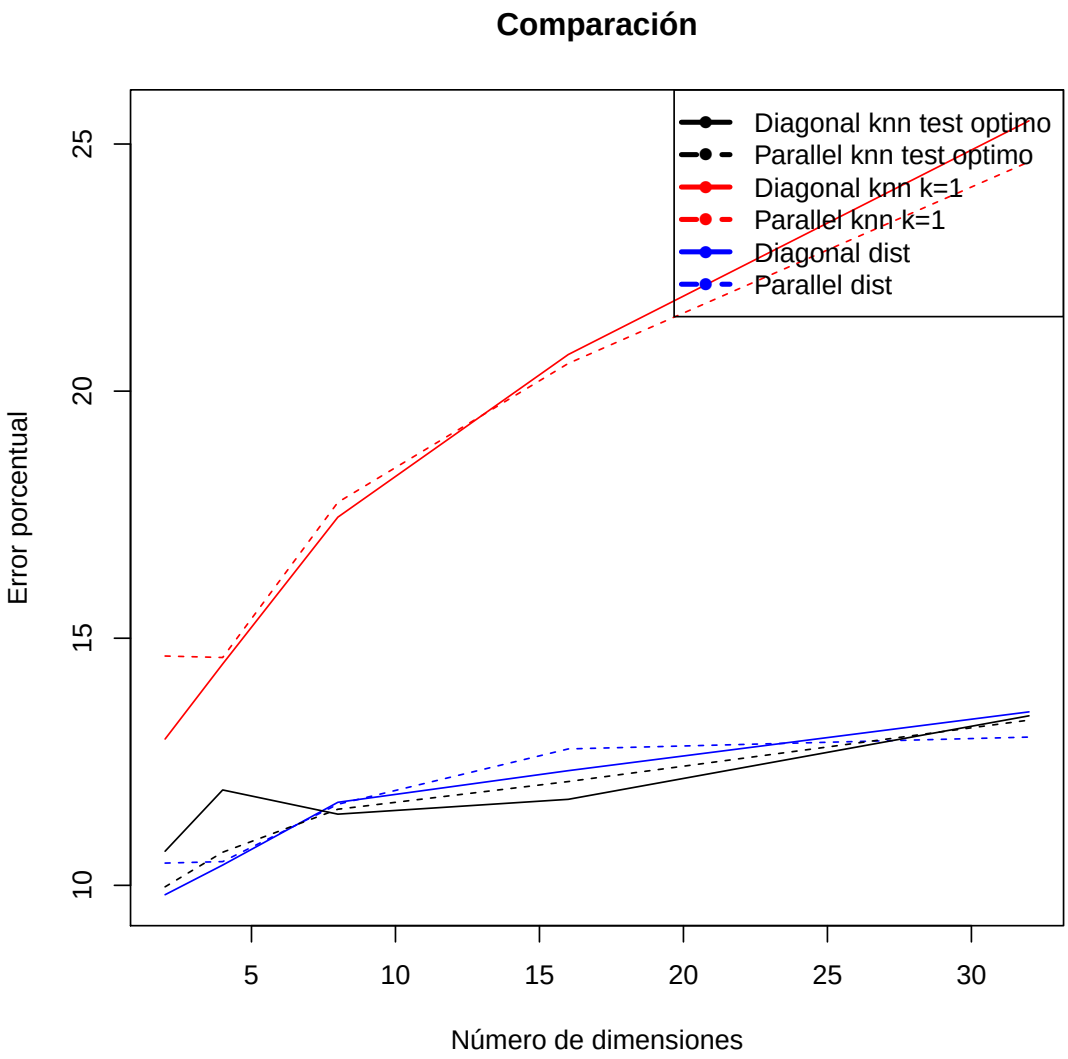
Paralelo:

d	Test error % óptimo
2	10.45
4	10.48
8	11.64
16	12.76
32	13.00

Nuevamente los números son muy parecidos entre el caso diagonal y paralelo. Se observa que mientras más dimensiones, más grande es el k necesario. Esto tiene sentido pues a más dimensiones, más separados estan los centro de las gaussianas y más separados están los puntos; por lo que es necesario considerar radios mayores para conseguir votantes.

No se observa que este método sea mejor que considerar k vecinos, al menos en estos dos problemas.

A continuación una gráfica de los errores del ejercicio anterior y en azul las de esta nueva implementación:



5 Ejercicio 5

Hay overfitting en k-nn?
Sí, si se usa un k pequeño. Por ejemplo en el caso de las espirales anidadas. Si queremos clasificar un punto cercano a la frontera, tal vez el punto más cercano sea de la clase incorrecta; y para k=1 por ejemplo lo clasificaríamos incorrectamente.

Si tenemos algún punto de clase 0 muy cercano a la frontera en el conjunto de training, los puntos de clase 1 del otro lado de la frontera serán clasificados como clase 0 con el vecino más cercano.
En este caso, si tenemos algún punto que es un poco anómalo o erróneo, entonces usar más de un vecino ayudará a equilibrar las clasificaciones. En general, podemos tener en un conjunto de training muy grande más de un punto anómalo juntos, por lo cual necesitaremos k un poco más grande para no aprender ruido. Elegir k demasiado grande también puede causar problemas (aunque no exáctamente por aprender demasiado), siendo el extremo el caso en que todos los puntos seran clasificados con la clase mayor.