

Practica 7

Clasificación supervisada y su Evaluación

Autor: Pablo García López
Profesor: Carlos Sáez Silvestre

Ejercicio 1. Clasificador Gaussiano lineal y cuadrático

1. Clasifica el conjunto de test de las clases 1 y 2 mediante un clasificador Gaussiano lineal entrenado con los datos de entrenamiento.

a) ¿Qué contiene la variable de salida POSTERIOR?

Las probabilidades a posteriori de las observaciones de entrenamiento. (Una matriz de 75x2, 2 clases 75 etiquetas).

b) Observa la salida coeff, ¿cuál es la frontera de decisión obtenida por el clasificador?

15x1

2. Clasifica el conjunto de test de las clases 1 y 2 mediante un clasificador Gaussiano cuadrático entrenado con los datos de entrenamiento. ¿Mejora los resultados del clasificador lineal?

Sí, mejora de un 0.1043 a un 0.0764.

3. Clasifica el conjunto de test de las clases 1, 2 y 3 con un clasificador lineal.

a) Calcula el error de test.

0.0827

b) Utiliza la función confus para conseguir la matriz de confusión del clasificador.

```
ans = 3x3
    43     6     3
     6    16     1
     1     2    13
```

4. Clasifica las 3 clases utilizando las 10 primeras componentes PCA. Para ello:

a) Tipifica el conjunto de entrenamiento y guarda las medias y varianzas para aplicarlas posteriormente para tipificar el test manualmente.

```
[XTtip, m, s] = tipificar([Xpittraining]);

l = size(Xpitest,1)
Xttip = (Xpitest - repmat(m,[l,1])) ./ repmat(s,[l,1])
```

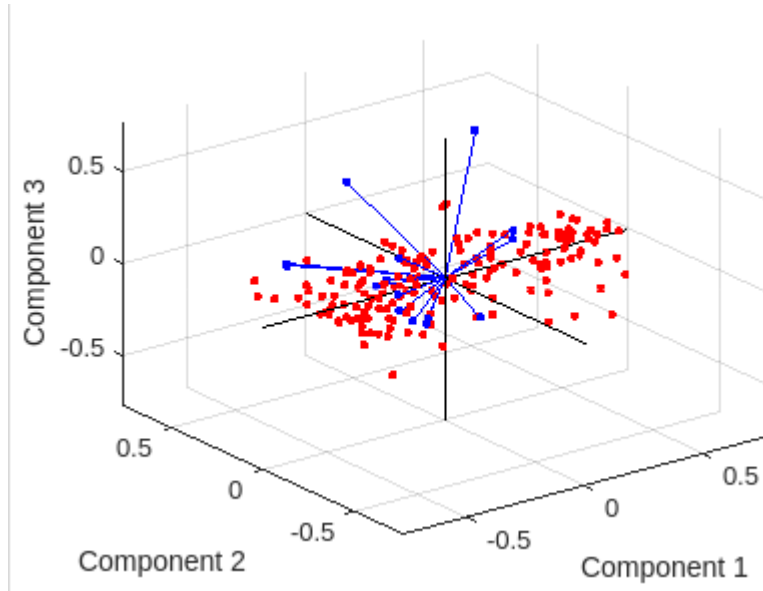
b) Estudia la normalidad de algunas de las variables mediante el test de KolmogorovSmirnov (kstest).

```
[h,p,k,c] = kstest(XTtip(:,floor(size(XTtip,2)/2)),[],0.05,0);
```

c) Obten mediante el método PCA (pca) las componentes principales (PC, loadings), y la proyección (scores) a partir de corpus de entrenamiento tipificado.

```
biplot(coeff(:,1:3), 'Scores', score(:,1:3))
```

d) Proyecta (obten scores) el conjunto de test tipificado mediante la matriz de loadings.



e) Clasifica el conjunto de test proyectado a las 10 primeras PC, mediante un clasificador lineal entrenado con los scores del conjunto de entrenamiento.

f) Calcula el error de test.

Ejercicio 2. Clasificador K Vecinos

1. Clasifica el conjunto de test de las clases 1, 2 y 3 con un k=1.

a) Calcula el error de test.

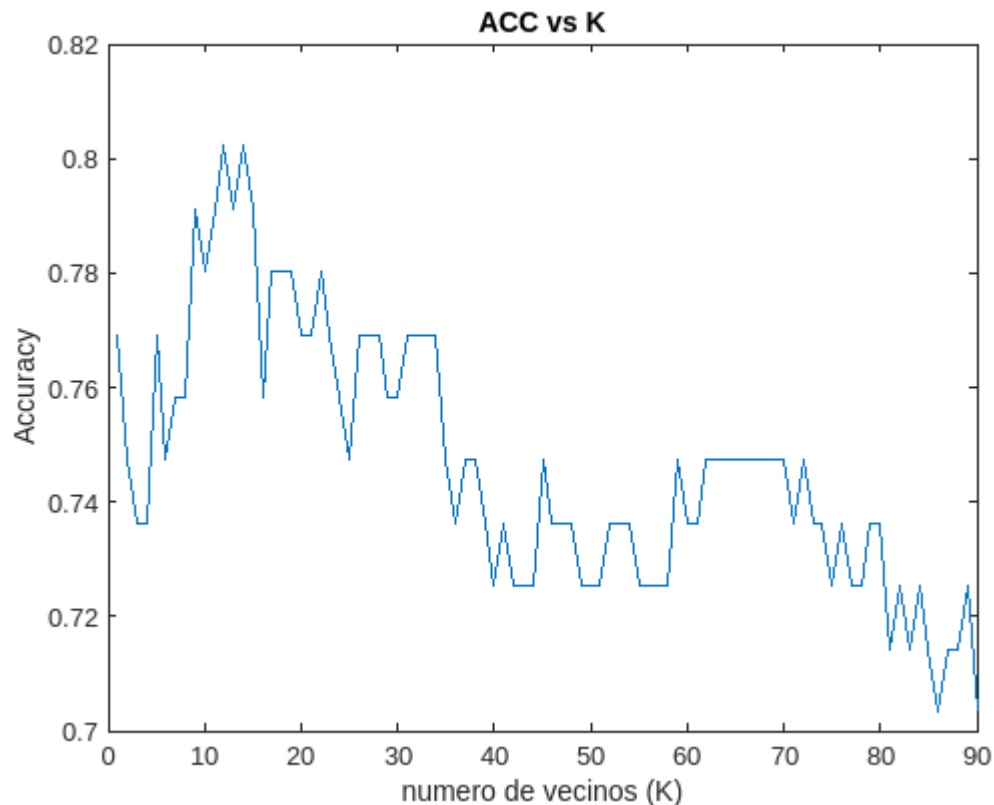
El error en test para este modelo es 0.2287.

b) Utiliza la función confus para conseguir la matriz de confusión del clasificador.

```
ans = 3x3
    45     3     4
     8    14     1
     1     4    11
```

2. Repite el proceso anterior para k desde 1 a 90 y muestra en un gráfico la dependencia del accuracy con el número de vecinos K . ¿Cómo ha evolucionado el error de test? ¿Por qué?

Primero el error mejora y llega un momento en el que vuelve a empeorar, esto es debido al coger de un vecino más lejano puede ir cogiendo con mayor facilidad de diferentes clases.



Ejercicio 3. Redes Neuronales

1. Con los parámetros existentes por defecto y para cada uno de los 4 conjuntos de datos, evalúa cuantas capas ocultas (hidden layers) son necesarias para poder generar una fronteras de decisión aceptable. Para este apartado utiliza por defecto 4 neuronas por capa.

- Circle: Con una capa oculta de 4 neuronas es más q suficiente.
- Exclusive or: Con una capa oculta de 4 neuronas es más q suficiente.
- Gaussian: No hacen falta capas ocultas.
- Spiral: Con los parametros por defecto y 4 neuronas por capa no se consigue generar una frontera aceptable.

2. Para un problema complejo como es el del conjunto de datos espiral y utilizando una arquitectura de dos capas, evalúa la influencia del número de neuronas por capa en la habilidad para generar fronteras complejas.

Hasta las 3 neuronas por capa no empieza a generar fronteras complejas, aunque no consiguen ser aceptables, mantiene un test loss de un 43% y un training loss de un 29%. Con 5 neuronas vemos una mejora a un test loss de un 34% y un training loss de 20%. Con 6 neuronas genera una frontera casi perfecta con un training loss de un 5% y un test loss de un 18%. Con 8

neuronas consigue bastante rapidamente una frontera con un test loss de 4.8% y training loss de 0.1%.

3. Para el segundo conjunto de datos Exclusive or, y con la arquitectura por defecto estudia la influencia del ratio de aprendizaje. ¿En qué influye? ¿Qué ventajas e inconveniente presenta el utilizar parámetros demasiado altos o demasiado bajos?

Influye en la velocidad con la que se actualizan los pesos. Un ratio alto puede hacer que converja más rapido, pero también puede hacer que el modelo salte a un mínimo global y no converja nada. Un ratio bajo puede hacer que el modelo converja lentamente, pero también puede hacer que el modelo se atasque en un mínimo local y no converja nada.

4. ¿Qué ocurre con el error de test, cuando aumentamos el ruido del problema y utilizamos una red con una arquitectura extremadamente sobredimensionada muchas capas y muchas neuronas por capa-? ¿Se puede solucionar aplicando algún método de regularización? Para este apartado debes dejar entrenar la red con un número de iteraciones alto y puedes probar con el conjunto de datos Exclusive or y un nivel de ruido de 50.

El error de test aumenta debido a un sobreajuste, el sobreajuste ocurre cuando el modelo se ajusta demasiado a los datos de entrenamiento y no generaliza bien a los datos nuevos. Se pueden aplicar tecnicas de regularización como la L1 o L2 que introducen un termino adicional para reducir el valor de los parámetros y evitar que se vuelvan demasiado grandes.