PRÁCTICA 1 - BIOMETRÍA

Medidas de calidad

Cristian Villarroya Sánchez

01-05-2021

Índice

1. Descripción de la tarea

Se disponen los scores tanto de clientes como de impostores de dos sistemas biométricos A y B. El fichero contiene en cada línea un id y un score, siendo este último el dato relevante para esta tarea.

Se ha creado un script en Python para resolver las tareas propuestas. Se ha utilizado la librería ArgumentParser para la recogida de los parámetros necesarios para su ejecución. La ayuda del script queda de la siguiente manera:

optional arguments:

-h, --help show this help message and exit

-fp X, --x X Valor de x para calcular FP(FN=x), FN(FP=x)

-c CLIENTS, --clients CLIENTS nombre del fichero con datos de clientes

-i IMPOSTORS, --impostors IMPOSTORS nombre del fichero con datos de impostores

No obstante, si no se especifica ninguno de estos parámetros, el script tiene unos parámetros por defecto, para que sea posible su ejecución. Por defecto son:

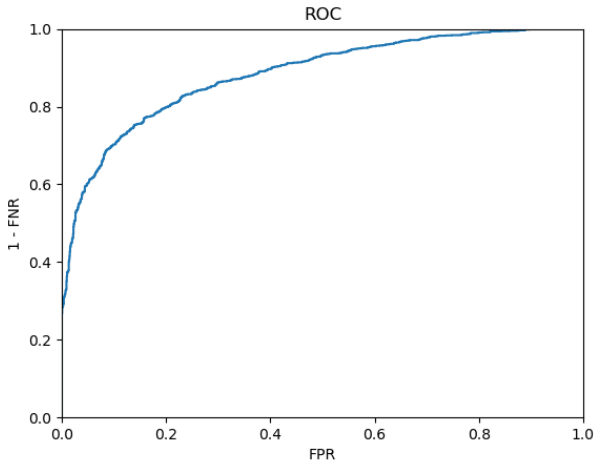
- -x = 0.05
- -c = scoresA_clientes
- -i = scoresA_impostores

Lo primero que hace el script es leer los ficheros de clientes e impostores y guardar sus scores en un array. En el array, se guarda una tupla, por un lado, el score del fichero y además un 1 si es cliente o un 0 si es impostor. Después, devuelve el array de scores ordenado de menor a mayor

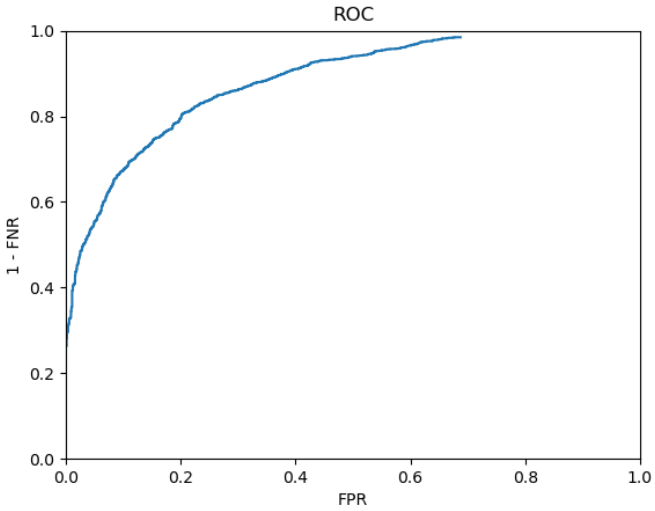
2. Dibujar curva ROC

La curva ROC es una representación gráfica de la sensibilidad (TPR) frente a la especificidad (FPR). Para dibujarla, se ha creado una función en la que, dados los scores de clientes e impostores (en un mismo array), calcula para todos los umbrales, que son todos aquellos valores existentes en el array de clientes e impostores, los falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos. Al calcular dichos valores para todos los umbrales, se crea una gráfica con la ratio FPR en el eje x y el TPR (1-FNR) en el eje y.

Para el sistema de scoresA, la curva ROC queda de la siguiente manera:



Y para el sistema de scoresB:



Esta función, además de devolver estas gráficas, devuelve todas las ratios de falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos, así como los umbrales que los han generado. Esta información será reutilizada en apartados posteriores.

3. Calcular FP cuando FN = x y umbral

En este caso, para obtener el caso en el que los falsos negativos son igual al valor “x”, se ha hecho uso de numpy. Es posible que no se encuentre un caso en el que los falsos negativos sean exactamente el valor “x”, por tanto, lo que se ha hecho es buscar el más cercano. Para encontrar ese valor, al array con las ratios de falsos negativos se le ha restado el valor “x” y se ha obtenido el índice de aquellos cuya diferencia con respecto a x es mínima, es decir, sean más parecidos al valor “x”.

Finalmente, como es posible obtener varios umbrales, se escogerá aquel que minimice la otra métrica, en este caso los falsos positivos. Para esto, de todos los índices obtenidos previamente, se obtiene su valor para los falsos positivos y se queda con el índice con menor valor en falsos positivos.

Una ejecución cuando x = 0.05 devuelve el siguiente resultado:

```
FP cuando FN =  0.05
FP =  0.5365384615384615
FN =  0.04965034965034965
Umbral =  0.004837
```

4. Calcular FN cuando FP = x y umbral

El procedimiento es el mismo que en el apartado anterior, solamente que aquí se realiza el proceso a partir de los falsos positivos, y se minimiza el valor de los falsos negativos.

Una ejecución cuando x = 0.05 devuelve el siguiente resultado:

```
FN cuando FP =  0.05
FP =  0.05
FN =  0.44545454545454544
Umbral =  0.144391
```

5. Calcular umbral en el que FP = FN

En este caso, se ha usado un proceso similar al de los apartados anteriores, solo que, en este caso, se calcula la diferencia entre las ratios de falsos positivos y falsos negativos. Para cada índice del numpy array se restan sus correspondientes valores de falsos positivos y falsos negativos, después, se coge el índice donde la diferencia sea mínima, es decir, donde sean más parecidos.

Finalmente se devuelve el valor de dichas ratios y el umbral que los produce. Una ejecución devuelve el siguiente resultado:

```
----------------------------
Umbral cuando FN = FP
FP =  0.2012820512820513
FN =  0.2006993006993007
Umbral =  0.044359
----------------------------
```

6. Área bajo la curva ROC

Para calcular el área bajo la curva ROC, se ha hecho uso de a prueba U de Mann-Whitney [1]. Esta prueba estadística, dice que, para unos valores “X” e “Y”, el correspondiente valor estadístico U, se calcula de la siguiente manera:

$$U = \sum_{i=1}^{n} \sum_{j=1}^{m} S(X_i, Y_j),$$

$$S(X,Y) = \begin{cases} 1, & \text{if } Y < X, \\ \frac{1}{2}, & \text{if } Y = X, \\ 0, & \text{if } Y > X. \end{cases}$$

Podemos considerar a “X” como los scores de clientes y a “Y” como los scores de impostores. Entonces, lo que hay que hacer es, mirar para cada valor de cliente, compararlo con todos y cada uno de los valores de impostores y sumar el valor correspondiente en función de si son iguales, mayor o menor.

La relación con el área bajo la curva roc, se basa en dividir el valor de U_1 por la multiplicación del número de muestras positivas y muestras negativas:

$$\mathrm{AUC}_1 = \frac{U_1}{n_1 n_2}$$

El valor U_1 se calcula de la siguiente manera:

$$U_1 = R_1 - \frac{n_1(n_1+1)}{2}$$

R_1 es la suma de los valores de los scores de clientes, n1 es la cantidad de scores de clientes y n2 es la cantidad de scores de impostores.

Con este método, se ha obtenido un área bajo la curva ROC de 0.8831 para el sistema A y un área de 0.8822 para el sistema B.

7. D-Prime

El valor d-prime o índice de sensibilidad es una estadística adimensional utilizada en la teoría de detección de señales [2]. Un índice más alto indica que la señal puede detectarse más fácilmente. Tiene en cuenta la separación de las distribuciones de clientes e impostores, así como el grado de dispersión y puede calcularse con la siguiente formula:

$$d' = \frac{\mu_{pos} - \mu_{neg}}{\sqrt{\sigma_{pos}^2 + \sigma_{neg}^2}}$$

Aplicando esa fórmula, se ha obtenido el valor d-prime de 0.7599 para el sistema A y de 0.8734 para el sistema B

8. Conclusiones

Se han estudiado e implementado diferentes métricas para evaluar un sistema biométrico. Con dichas implementaciones, se ha evaluado el rendimiento de dos sistemas A y B. Ambos tienen un área bajo la curva ROC similar, sin embargo, el valor d-prime es bastante más elevado en el sistema B, por lo que se podría decir que el sistema B es mejor que el A.

9. Bibliografía

[1] Wikipedia contributors, “Mann–whitney u test — Wikipedia, the free ency-clopedia,” 2021. [Online; accessed 27-May-2021].

[2] Wikipedia contributors. Sensitivity index — Wikipedia, the free encyclopedia, 2021. [Online; accessed 27-May-2021]