Clasificación utilizando el Discriminante Lineal de Fisher

Julián Bayardo[*], Christian Cuneo[**]

16 de agosto de 2017

1. Implementación de LDA para dos clases

Para resolver este ejercicio, utilizamos la solución obtenida por Cuadrados Mínimos del vector de pesos para el discriminante de Fisher[1]. Tenemos que la solución en este caso es tomar

$$W = S_W^{-1}(m_2 - m_1)$$
$$w_0 = -Wm$$

Con S_W la matriz de covarianza vista en clase, m_2 el promedio del dataset de los ejemplos con clase 2, m_1 el promedio del dataset de los ejemplos con clase 1, y m el promedio del dataset. Con esta definición, el clasificador queda definido como:

$$y(x) = \begin{cases} 1 & \text{if } Wx + w_0 \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

2. Implementación de LDA para K clases

Para este ejercicio, utilizamos la solución por resolución del problema de autovectores generalizado[2]. El problema se reduce a obtener todos los valores de w_i y λ_i que satisfacen la siguiente ecuación:

$$(S_B - S_W\lambda_i)w_i = 0$$

Donde S_B y S_W son las matrices de covarianza vistas en clase. Luego, normalizamos a los w_i, definimos a $\hat{W}$ como una matriz cuyas columnas son los w_i ordenados por el valor de su autovalor correspondiente (es decir, la primer columna es el autovector con mayor autovalor, y así sucesivamente). Luego tomamos

$$W = W(W^T m)$$

[*]Libreta universitaria 850/13, correo julian@bayardo.info
[**]Libreta universitaria 755/13, correo chriscuneo93@hotmail.com

$$w_0 = \frac{-1}{2}(Wm)I$$

Con m el vector de los promedios en cada uno de los features del conjunto de entrenamiento. Entonces, nuestro clasificador queda definido como

$$y(x) = argmax_{i \in [C]}(Wx + w_0)_i$$

Observemos que lo interesante de esta metodología es que nos permite tener una única función para clasificar, y no es una construcción a partir de múltiples clasificadores. Esto quiere decir que no sufre del problema de regiones que no son clasificables, como sí lo haría un clasificador por votación de la versión anterior para dos clases.

3. Experimentación con gausianas multidimensionales isotrópicas

Para toda la experimentación, utilizamos datos sampleados del proceso que motiva la formulación de Fisher: asumimos que hay k clases, cuya probabilidad de occurrencia $P(k = i)$ es $\pi_1, ..., \pi_k$ respectivamente, y además tenemos $P(X|k)$ $N(\mu_k, \Sigma_k)$. En los ejemplos que mostramos en el trabajo, asumimos siempre que $\pi_1 = ... = \pi_k = \frac{1}{k}$.

3.1. Un ejemplo con 3 clases

Para el primer caso, tomamos 3 distribuciones normales isotrópicas sobre un espacio bidimensional. En orden, tenemos las siguientes distribuciones: $N(\begin{bmatrix}5{,}0 & -5{,}0\end{bmatrix}, 5I)$, $N(\begin{bmatrix}-7{,}5 & -7{,}5\end{bmatrix}, 0{,}75I)$, $N(\begin{bmatrix}5 & 5\end{bmatrix}, I)$ para las clases 0, 1 y 2 respectivamente.

Podemos observar en la figura 1 las distribuciones particulares que utilizamos para el proceso, y el conjunto de datos de entrenamiento. En este caso utilizamos un conjunto de datos de entrenamiento de tamaño 100 (que es lo que se visualiza en la figura).

Una vez finalizado el entrenamiento, utilizamos 900 datos adicionales generados por el mismo proceso como conjunto de prueba, y podemos ver los resultados de clasificarlos en la figura 2. Podemos ver en la figura 2a que el clasificador parecería comportarse razonablemente ante nuevos datos generados por el mismo proceso: los puntos en las normales cercanas están efectivamente coloreados con el mismo color. Además, la figura 2b nos permite ver que efectivamente las decision boundaries corresponden con lo que intuitivamente esperaríamos que un clasificador lineal genere según vimos en clase y en la literatura.

3.2. Un ejemplo con 4 clases

En este caso, tomamos el ejemplo anterior y le agregamos una cuarta clase con distribución $N(\begin{bmatrix}5{,}0 & 5{,}0\end{bmatrix}, 0{,}5I)$. Replicamos los gráficos del ejemplo anterior sobre este caso particular en la figura 3.

Repitiendo también los gráficos después del entrenamiento en la figura 4, vemos el mismo comportamiento que en el caso anterior, pero con una nueva clase agregada.

4. Overlap entre clases y el error de clasificación

Decidimos realizar este ejercicio tomando una única feature (es decir, datos unidimensionales) para que fuera más fácil de visualizar el overlap y comprender la situación. Es decir, todas las situaciones que contemplaremos serán una transformación entre dos normales unidimensionales isotrópicas a un label 0 o 1.

Observemos que generalizar esto es relativamente simple en el sentido que tenemos que comenzar a tomar todas las posibles combinaciones para cada una de las componentes de la distribución correspondiente a cada uno de los labels. Sin embargo, el ejercicio en más dimensiones es considerablemente más complicado de visualizar, porque se generan muchos casos. Además, en el caso unidimensional sólo tenemos 2 posibles direcciones, pero en el caso bidimensional ya tenemos infinitas direcciones posibles; por lo que tendríamos que pensar muchas más proyecciones para darnos una buena idea de qué sucede.

Cabe destacar que, además, como estamos utilizando una funcón discriminante lineal y la dimensión de nuestro conjunto de datos es 1, estaremos generando un hiperplano de dimensión 0 como forma de separar las clases (es decir, tendremos una constante que separara a nuestras clases). En la notación que utilizamos para la sección 1, esta constante será $\frac{-w_0}{W}$.

4.1. Una contenida dentro de la otra

Si observamos la figura 5a, podemos ver que en este caso las distribuciones de los labels tienen mucho overlap: la clase azul es ligeramente más grande que la roja, y la roja está en su totalidad contenida dentro de la azul.

Como sabemos, el algoritmo de entrenamiento encontrará una constante para dividir el conjunto de datos en dos. Claramente, no es posible encontrar una única constante (aunque dos constantes funcionarían bien, si tomásemos las dos intersecciones entre las distribuciones, o bien utilizar un kernel radial). Esto es lo que nos está mostrando la figura 5b: el algoritmo encontró una constante entre 2 y 3 que utiliza para separar las dos clases. Podemos ver cómo se repite esta situación entre los datos de prueba de la figura 6.

La decisión tomada por el algoritmo se refleja de forma muy clara en la matriz de confusión que vemos en la figura 7: tenemos en este caso más false negatives que true positives y true negatives por separado.

4.2. Intersección parcial

En el caso de las figuras 8, tomamos clases con una pequeña intersección. Observemos que una vez más, nunca vamos a poder evitar errores en la clasificación de la intersección: no importa cómo elijamos la constante, vamos a estar clasificando erróneamente a los ejemplos de una clase o de la otra. La elección de la constante puede verse claramente en la figura 9: tenemos un claro cambio de color en los puntos cercanos al 3.

Como mencionamos anteriormente, el error en este caso es inevitable, y la matriz de confusión de la figura 10 muestra precisamente eso: si bien clasificamos fundamentalmente bien a las dos clases, tenemos algunos falsos positivos y algunos falsos negativos.

4.3. Totalmente separadas

Este último caso es también el más simple: como las clases están bien separadas, no tenemos el problema anterior que mencionamos sobre la intersección. Cabe destacar que como las normales tienen soporte infinito, efectivamente es posible que caiga alguno de los valores no mostrados en los gráficos de las figuras 11 o 12; sin embargo, el clasificador no deja de tener un punto divisorio aprendido a partir de los datos, que le permitirá discernir entre ambas clases.

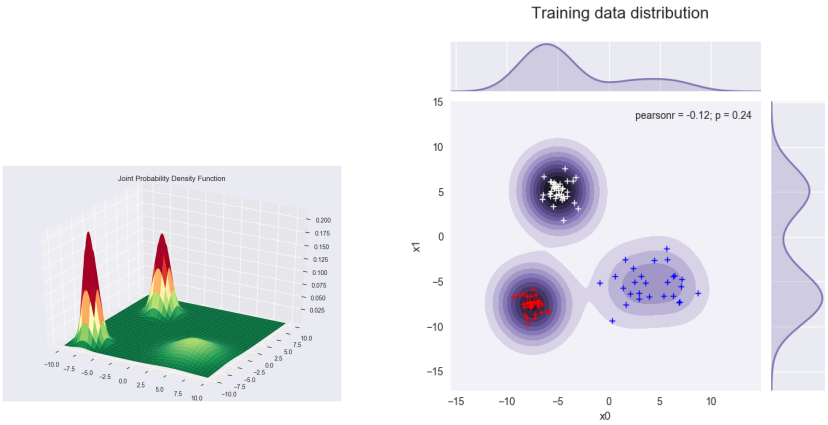
La matriz de confusión de la figura 13 efectivamente muestra que para el proceso que tomamos, los pesos tomados no cometen ningún error sobre el conjunto de test.

Referencias

[1] C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[2] R.O. Duda, P.E. Hart, , and D.G. Stork. *Pattern Classification*. Springer-Verlag New York, Inc., 2 edition, September 2007.

5. Figuras

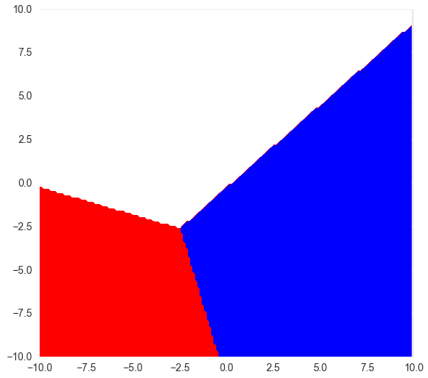


(a) Las distribuciones de probabilidad de las tres clases (por separado, no es la función de probabilidad continua).

(b) La distribución concreta de los datos de entrenamiento sampleados del proceso generador.

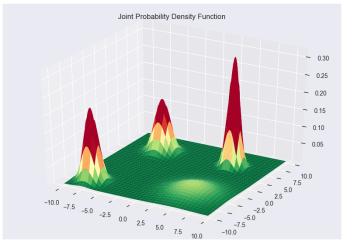
Figura 1

(a) El resultado de clasificar los datos generados para probar el clasificador.



(b) Decision boundaries de las clases

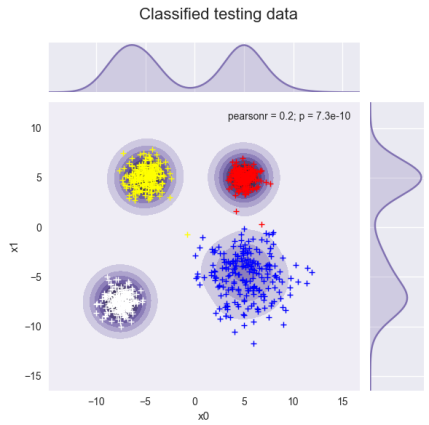
Figura 2



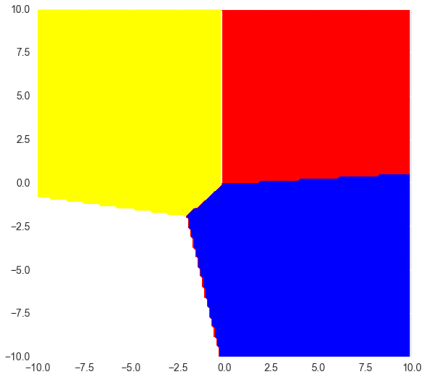
(a) Las distribuciones de probabilidad de las tres clases (por separado, no es la función de probabilidad continua).

(b) La distribución concreta de los datos de entrenamiento sampleados del proceso generador.

Figura 3

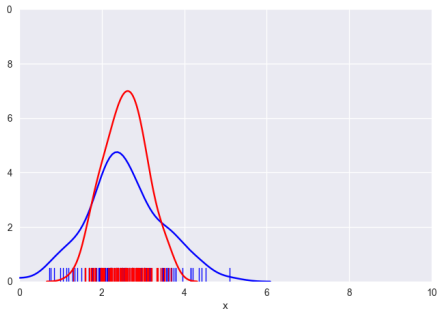


(a) El resultado de clasificar los datos generados para probar el clasificador.

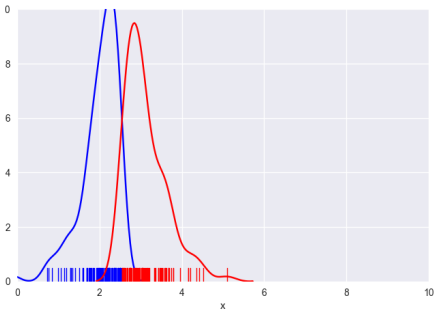


(b) Decision boundaries de las clases

Figura 4

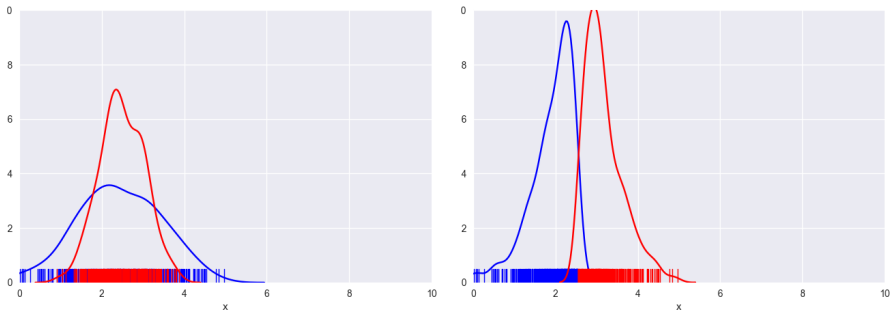


(a) Las distribuciones utilizadas para generar los datos.



(b) Distribuciones de los datos de prueba para cada clase. La intersección es error del método para plotear las distribuciones; el clasificador no çlasifica como ambos". El decision boundary es donde los datos dejan de "ser azulesz pasan a "ser rojos.[en] la parte inferior.

Figura 5: Datos de entrenamiento



(a) Distribución de los datos generados agrupado por label.

(b) Distribución de los datos generados agrupado por label predicho. La intersección es error del método para plotear las distribuciones; el clasificador no çlasifica como ambos". El decision boundary es donde los datos dejan de "ser azulesz pasan a "ser rojos.[en] la parte inferior.

Figura 6: Datos de prueba

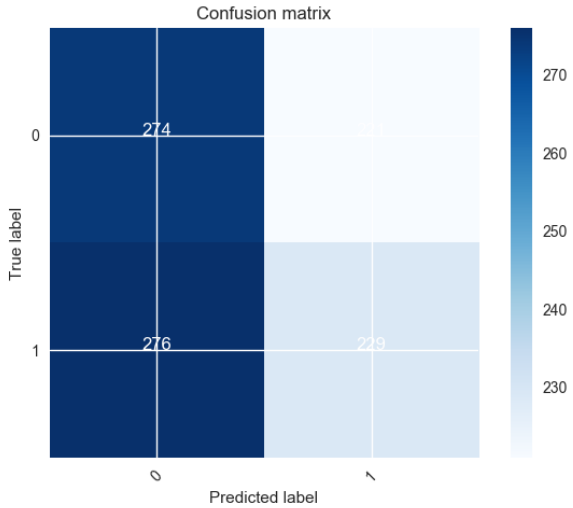
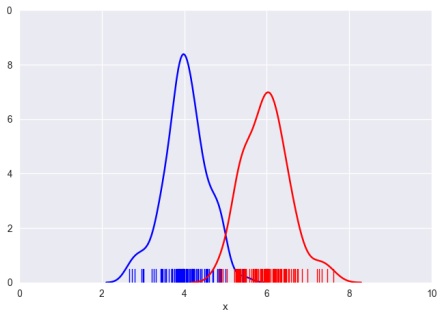
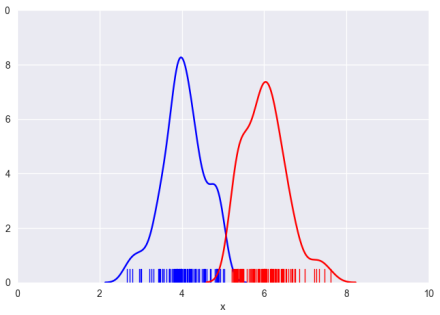


Figura 7: Matriz de confusión para los datos de prueba. El 0 corresponde con la clase azul, 1 con la clase roja.

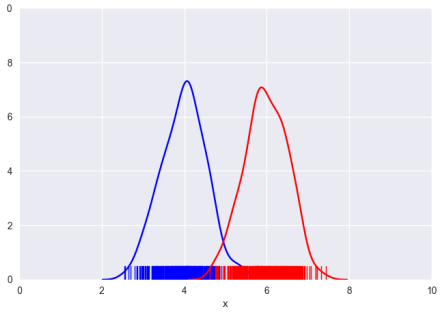


(a) Distribución de los datos generados agrupado por label.

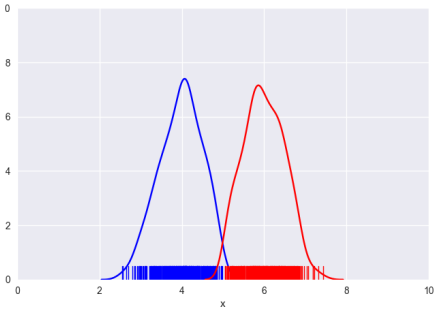


(b) Distribuciones de los datos de prueba para cada clase. La intersección es error del método para plotear las distribuciones; el clasificador no çlasifica como ambos". El decision boundary es donde los datos dejan de "ser azulesz pasan a "ser rojos.[en] la parte inferior.

Figura 8: Datos de entrenamiento



(a) Distribución de los datos generados agrupado por label.



(b) Distribución de los datos generados agrupado por label predicho. La intersección es error del método para plotear las distribuciones; el clasificador no çlasifica como ambos". El decision boundary es donde los datos dejan de "ser azulesz pasan a "ser rojos.[en] la parte inferior.

Figura 9: Datos de prueba

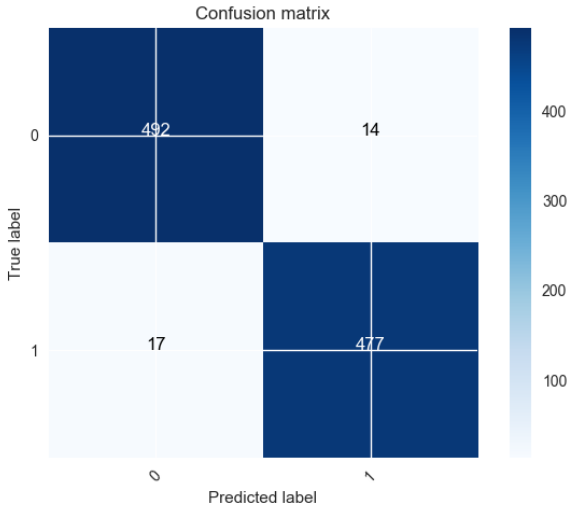
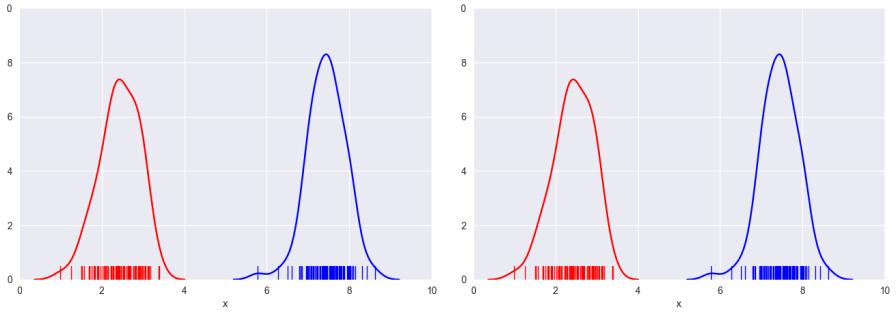


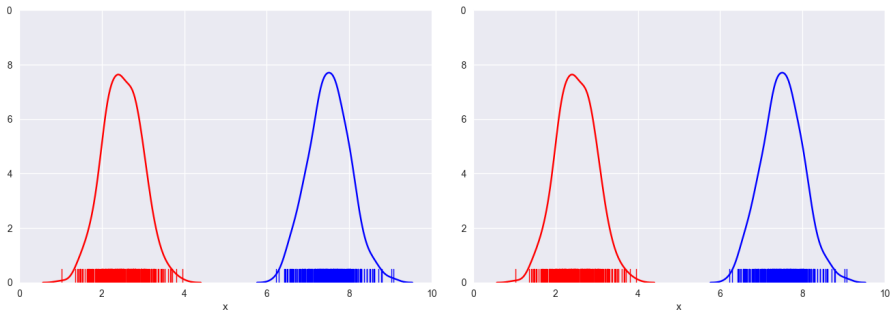
Figura 10: Matriz de confusión para los datos de prueba. El 0 corresponde con la clase azul, 1 con la clase roja.



(a) Distribución de los datos generados agrupado por label.

(b) Distribuciones de los datos de prueba para cada clase. La intersección es error del método para plotear las distribuciones; el clasificador no çlasifica como ambos". El decision boundary es donde los datos dejan de "ser azulesz pasan a "ser rojos.[en] la parte inferior.

Figura 11: Datos de entrenamiento



(a) Distribución de los datos generados agrupado por label.

(b) Distribución de los datos generados agrupado por label predicho. La intersección es error del método para plotear las distribuciones; el clasificador no çlasifica como ambos". El decision boundary se encuentra en algún lado (no mostrado) entre las dos normales.

Figura 12: Datos de prueba

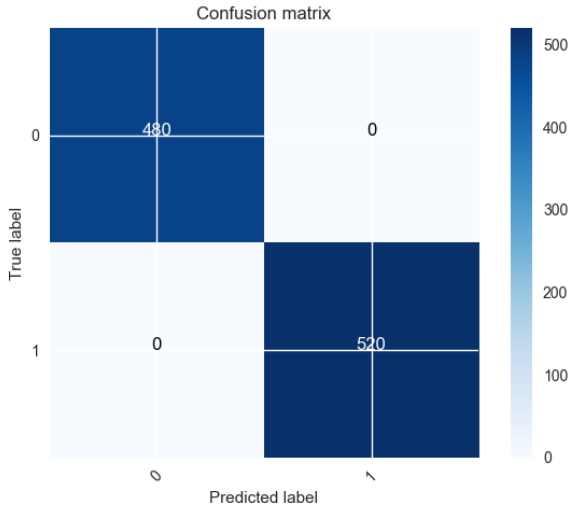


Figura 13: Matriz de confusión para los datos de prueba. El 0 corresponde con la clase azul, 1 con la clase roja.