Tarea 7: Discriminante Lineal de Fisher

1st Ericka Céspedes Moya
Tecnológico de Costa Rica
San José, Costa Rica
ericka.cespedes@gmail.com

2nd Esteban Alonso González Matamoros
Tecnológico de Costa Rica
San José, Costa Rica
esteb.gonza29@gmail.com

***Resumen*—El siguiente documento contiene el informe de la tarea 7 para el curso de Introducción al Reconocimiento de Patrones que trata sobre el discriminante lineal de Fisher. En este trabajo se encuentran los casos de uso y la teoría en la cual se basa el discriminante lineal de Fisher.**

***Index Terms*—discriminante, lineal, fisher, patrones**

I. Introducción

Análisis Discriminante Lineal (ADL, o LDA por sus siglas en inglés) es una generalización del discriminante lineal de Fisher, un método utilizado en estadística, reconocimiento de patrones y aprendizaje de máquinas. LDA está estrechamente relacionado con el análisis de varianza (ANOVA) y el análisis de regresión. Estos otros métodos son preferibles en aplicaciones donde no es razonable asumir que las variables independientes están normalmente distribuidas, lo cual es una suposición fundamental del método ADL. El caso del discriminante lineal de Fisher es particular y se le llama DLF (o FDL en inglés) [1].

II. Teoría

La idea básica del discriminante lineal de Fisher es elegir una dirección de proyección (transformación lineal, combinación lineal), reducir el problema de alta dimensión a un problema unidimensional para resolver, y que los datos unidimensionales transformados satisfagan las necesidades de cada tipo de muestras internas recopiladas tanto como sea posible. Al mismo tiempo, las muestras de diferentes tipos están lo más alejadas posible.

Con el discriminante lineal de Fisher se determina la dirección de proyección W y el umbral w_0 a través de los datos de entrenamiento dados. Es decir, se determina la función discriminante lineal y luego se prueban los datos de prueba de acuerdo con la función discriminante lineal.

El discriminante de Fisher tiene como objetivo lograr la mayor distancia entre clases y la menor distancia dentro de clases. La forma general de la función discriminante lineal se puede expresar como

$$g(X) = W^T X + w_0 \tag{1}$$

$$X = \begin{pmatrix} x_1 \\ \cdots \\ x_d \end{pmatrix} W = \begin{pmatrix} w_1 \\ w_2 \\ \cdots \\ w_d \end{pmatrix} \tag{2}$$

[2].

Fisher elige el principio de la dirección de proyección W, incluso si la proyección del vector de muestra original en esta dirección puede tener en cuenta la separación de la distribución entre clases tanto como sea posible, y el requisito de que las proyecciones de muestra dentro de la clase sean lo más densas posible.

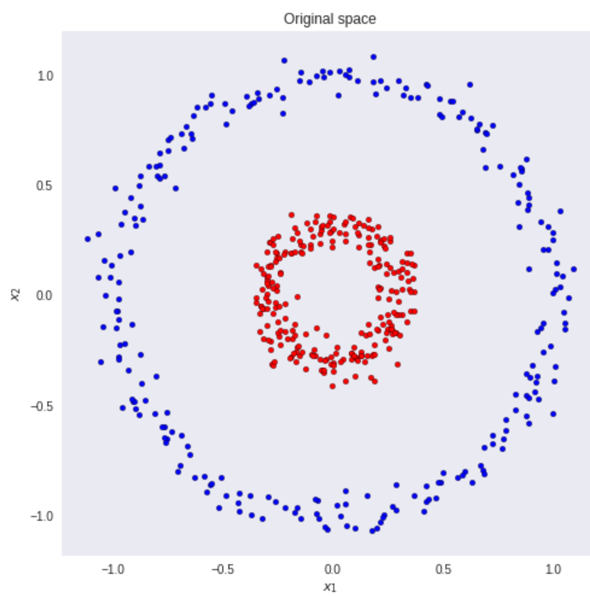

Figura 1. Dataset original

Supongamos que queremos clasificar correctamente los círculos rojo y azul. No existe una combinación lineal de las entradas y los pesos que asigne las entradas a sus clases correctas. Pero, ¿y si pudiéramos transformar los datos para poder dibujar una línea que separe las 2 clases?

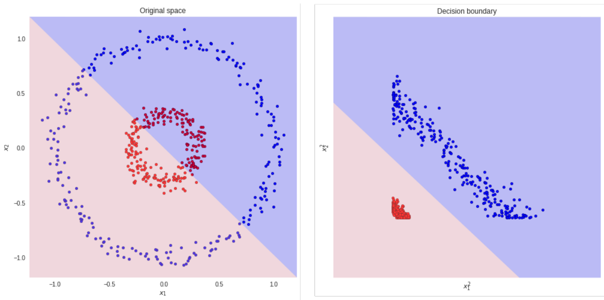

Figura 2. Separación de las dos clases

Una solución a este problema es aprender la transformación correcta. Esto se conoce como aprendizaje de representación (Deep Learning). No necesitamos "adivinar" qué tipo de transformación daría como resultado la mejor representación de los datos. El algoritmo lo resolverá.

Podemos ver los modelos de clasificación lineal en términos de reducción de dimensionalidad. Para empezar, considere el caso de un problema de clasificación de dos clases ($K = 2$). Puntos azules y rojos en R^2. En general, podemos tomar cualquier vector de entrada de dimensión D y proyectarlo a dimensiones D'). Aquí, D representa las dimensiones de entrada originales, mientras que D') son las dimensiones del espacio proyectado. Considere $D' < D$.

En el caso de proyectar a una dimensión, es decir, $D' = 1$, podemos elegir un umbral t para separar las clases en el nuevo espacio. Dado un vector de entrada x:

- si el valor predicho $y >= t$ entonces, x pertenece a la clase $C1$ (clase 1) - donde $y = W^T x$
- de lo contrario, se clasifica como $C2$ (clase 2).

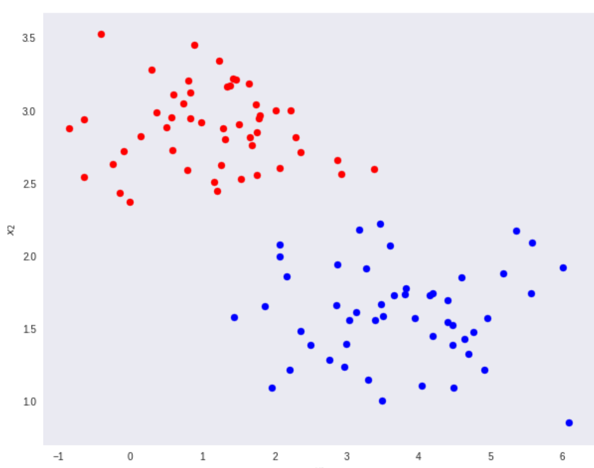

Figura 3. Vectores m_1 y m_2

De esta manera, tenemos los vectores m_1 y m_2 para las dos clases, donde N_1 y N_2 denotan el número de puntos en las clases $C1$ y $C2$ respectivamente. Ahora, usamos las medias de la clase como medida de separación. En otras palabras, queremos proyectar los datos en el vector W que une las medias de 2 clases.

$$m_1 = \frac{1}{N_1} \sum_{n \in C1} x_n \qquad m_2 = \frac{1}{N_2} \sum_{n \in C2} x_n \qquad (1)$$

Figura 4. Fórmulas de los vectores m_1 y m_2

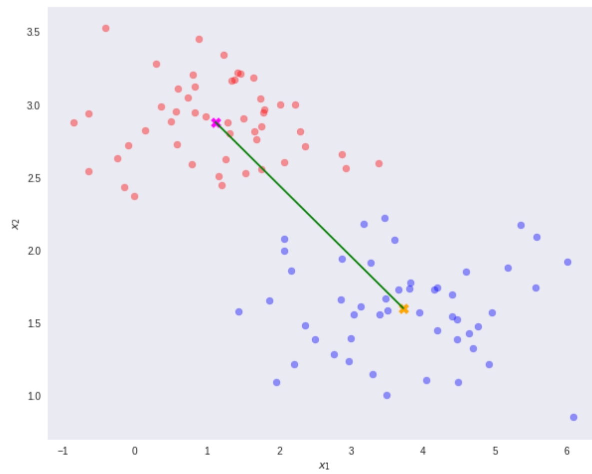

Figura 5. Proyección de las dos clases

Es importante señalar que cualquier tipo de proyección a una dimensión más pequeña puede implicar alguna pérdida de información. En este escenario, las dos clases son claramente separables (por una línea) en su espacio original.

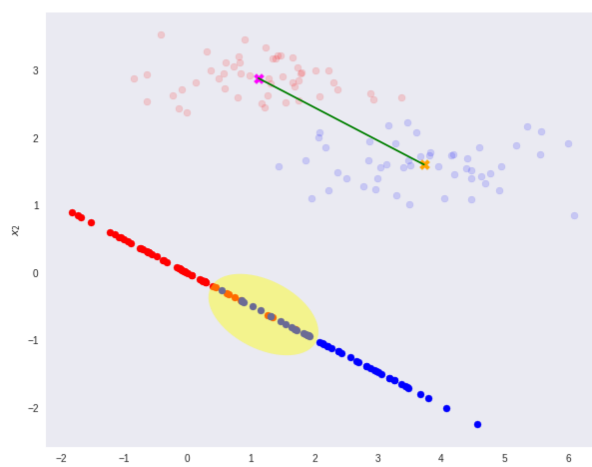

Figura 6. Reproyección de las dos clases

Sin embargo, después de la reproyección, los datos muestran algún tipo de superposición de clases, que se muestra mediante la elipse amarilla en el gráfico y el histograma a continuación.

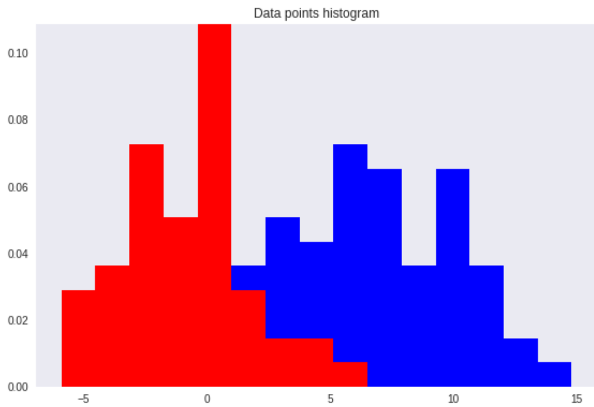

Figura 7. Histograma de las dos clases

Aquí es donde aplicamos el discriminante lineal de Fisher. La idea propuesta por Fisher es maximizar una función que proporcionará una gran separación entre las medias de clase proyectadas y al mismo tiempo una pequeña variación dentro de cada clase, minimizando así la superposición de clases. En otras palabras, el discriminante lineal de Fisher selecciona una proyección que maximiza la separación de clases.

Para hacer esto, maximiza la relación entre la variación entre clases y la variación dentro de la clase:

- Una gran variación entre las clases de conjuntos de datos.
- Una pequeña variación dentro de cada una de las clases de conjuntos de datos.

Así, una gran variación entre clases significa que los promedios de clase proyectados deben estar lo más separados posible. Por el contrario, una pequeña variación dentro de la clase tiene el efecto de mantener los puntos de datos proyectados más cerca unos de otros.

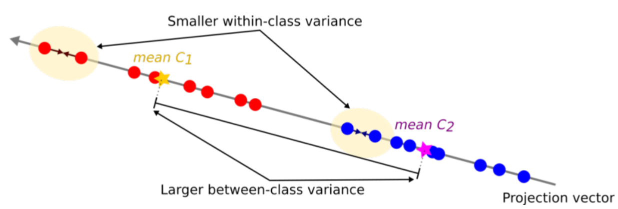

Figura 8. Vector de proyección de las medias y varianzas

Para encontrar la proyección con las siguientes propiedades, DLF tiene un vector W con el siguiente criterio:

$$J(\boldsymbol{W}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \qquad (1)$$

Between-class variance

Within-class variance

Figura 9. Vector W

Si sustituimos los vectores promedios m_1 y m_2 así como la varianza s dada por las ecuaciones (1) y (2) llegamos a la ecuación (3). Si tomamos la derivada de (3) (despúes de algunas simplificaciones) obtenemos la ecuación de aprendizaje para W (ecuación 4). Es decir, W (la transformación deseada) es directamente proporcional a la inversa de la matriz de covarianza dentro de la clase multiplicada por la diferencia de las medias de la clase.

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2 \quad y_n = \boldsymbol{W}^T x_n \qquad (2)$$

$$J(\boldsymbol{W}) = \frac{\boldsymbol{W}^T S_B \boldsymbol{W}}{\boldsymbol{W}^T S_W \boldsymbol{W}} \qquad (3)$$

$$\boldsymbol{W} \propto S_W^{-1}(m_2 - m_1) \qquad (4)$$

Per-class mean

Within-class variance

projection equation

Figura 10. Ecuaciones 2, 3 y 4 donde obtenemos W

Finalmente, el resultado permite una separación de clases perfecta con un umbral simple.

Figura 11. Proyección final de las dos clases separadas

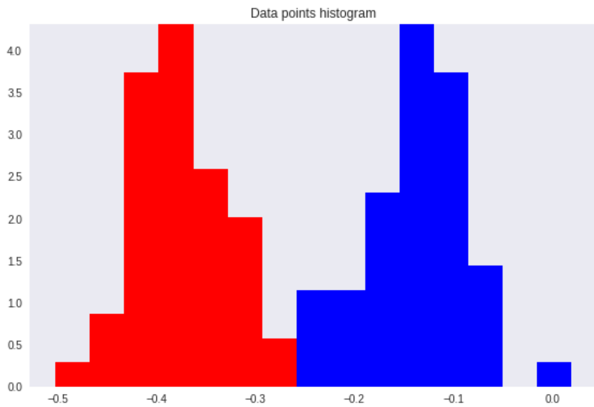

Figura 12. Histograma final de las dos clases separadas

Adicionalmente, agregamos el discriminante lineal de Fisher para clases múltiples. Podemos generalizar el discriminante para el caso de más de 2 clases ($K > 2$). Se tienen formas de generalización para las matrices de covarianza dentro de la clase y entre clases.

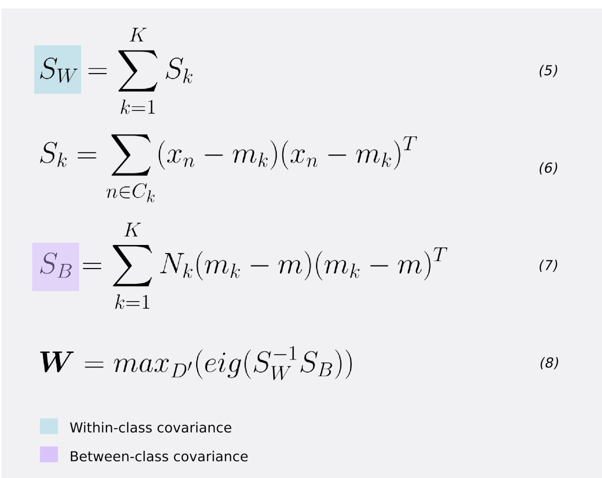

Figura 13. Fórmulas del discriminante lineal de Fisher para clases múltiples

Para la matriz de covarianza dentro de la clase S_W, para cada clase, se toma la suma de la multiplicación de la matriz entre los valores de entrada centralizados y su transposición (ecuaciones 5 y 6).

Para estimar la covarianza entre clases S_B, para cada clase $k = 1, 2, 3, \ldots, K$, se toma el producto externo de la media de clase local m_k y la media global m. Luego, se escala por el número de registros en la clase k (ecuación 7).

La maximización del criterio DLF se resuelve mediante una descomposición propia de la multiplicación de matrices entre la inversa de S_W y S_B. Por lo tanto, para encontrar el vector W, tomamos los vectores propios D' que corresponden a sus valores propios más grandes (ecuación 8).

En otras palabras, si queremos reducir nuestra dimensión de entrada de $D = 784$ a $D' = 2$, el vector W está compuesto por los 2 vectores propios que corresponden a $D' = 2$ valores propios más grandes. Esto da una forma final de $W = (N, D')$, donde N es el número de registros de entrada y D' las dimensiones reducidas de la característica [3], [4].

III. Casos de uso

En cuanto a casos de uso, encontramos 4 casos de usos principales, los cuales son: reconocimiento de caras, predicción de bancarrota, biomedicina y marketing.

III-A. *Reconocimiento de caras*

Primeramente se encuentra el uso más relacionado a este curso, el cual sería el reconocimiento de caras, dentro de este área de visión por computadora, se ha utilizado mucho el análisis de discriminante lineal de Fisher debido a que en base a una serie de datos ya identificados en distintas clases se pueden crear "plantillas"de las combinaciones lineales de valores de pixeles llamadas Fisherfaces o caras de Fisher [5], con ellas se pueden comparar imágenes e identificar a cuál grupo pertenece un nuevo espécimen [6]. Además el uso más común de este método es para disminuir el número de características antes de la clasificación de un dato.

III-B. *Predicción de bancarrota*

En segundo lugar se encuentra una aplicación más orientada al área de estadística y manejo de empresas, en este caso el ADL se utiliza para identificar si un empresa se encuentra en la bancarrota o está en un estado de supervivencia, para esto se utiliza un análisis de variables financieras como proporciones. Como dato interesante, el ADL fue el primer método estadístico aplicado sistemáticamente para explicar esta situación de una empresa.

III-C. *Biomedicina*

En este uso, el discriminante lineal se suele utilizar orientado al análisis biomédico, principalmente para distinguir el nivel de prioridad o severidad de un paciente en cuanto a cierta enfermedad o padecimiento, como por ejemplo el cáncer [7] o un daño en la retina. en este caso, se suele obtener un diagnostico que da la valoración del paciente según los distintos grupos que se tengan de la enfermedad específica. Por ejemplo, una clasificación muy común es la de ligera, moderada o severa.

III-D. *Marketing*

Por último, encontramos el uso del Marketing el cual se ha vuelto bastante popular en los últimos años debido a la gran importancia de esta área en la actualidad. Dentro de este ámbito el ADL se suele utilizar para distinguir si una persona pertenece a cierto grupo o a otro, por lo tanto, sirve para distinguir tipo de clientes, productos y/o servicios que sean analizados. Normalmente en el Marketing se tiene un proceso bastante simple en cuanto al uso del ADL, las etapas suelen ser de recopilación de datos cuantitativos u objetivos mediante encuestas o redes sociales, luego se da la estimación de los Coeficientes de la Función Discriminante y se determina el grado de importancia y validez, y por último, se trazan los resultados obtenidos del programa estadístico, usualmente en un mapa bidimensional, se definen las dimensiones y se interpretan los resultados, siendo por ejemplo, que entre más distancia exista mayor es la diferencia entre dos servicios. [8]

IV. Ejemplo programado

Utilizamos la clase *LinearDiscriminantAnalysis* predefinida que la biblioteca *scikit-learn* pone a nuestra disposición para llevar a cabo el ejercicio programado. [9]

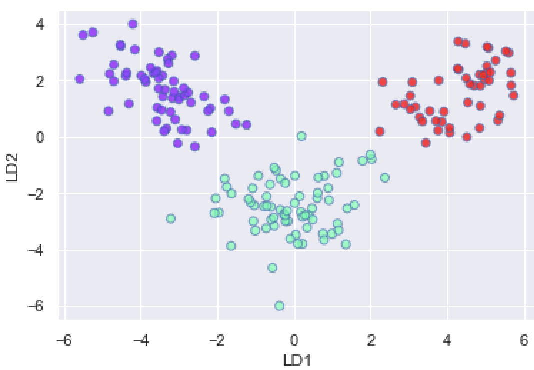

Figura 14. Ejercicio programado visual usando *scikit-learn*

Adicionalmente, se pueden revisar los enlaces de [2] que utiliza *scikit-learn* y [3] que utiliza *MNIST* como un conjunto de datos de prueba con espacios proyectados en $D' = 2$ y $D' = 3$.

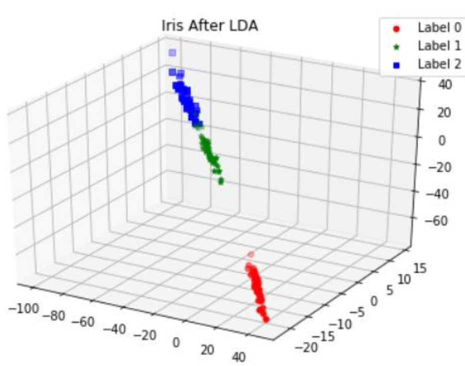

Figura 15. Otro ejercicio programado visual usando *scikit-learn*

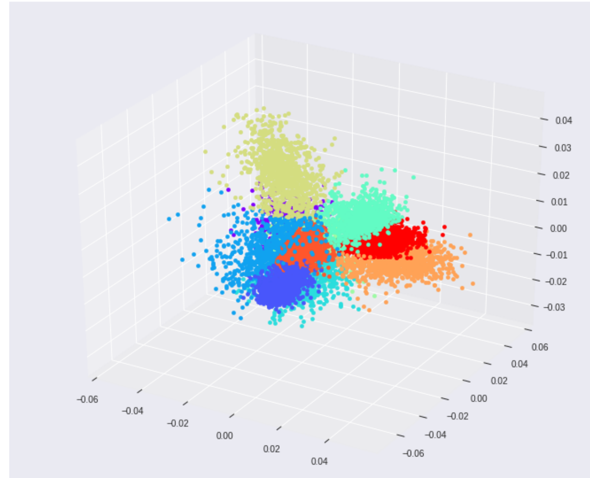

Figura 16. Otro ejercicio programado usando *MNIST*

Referencias

[1] J. Amat-Rodrigo, "Análisis discriminante lineal (lda) y análisis discriminante cuadrático (qda)," *Ciencia de Datos, Estadística, Machine Learning y Programación*, 2016. [Online]. Available: shorturl.at/fpxEX

[2] F. Tian, "Clasificador lineal-discriminación lineal de fisher," *Programador Clic*. [Online]. Available: https://programmerclick.com/article/63921659653/

[3] S. Thalles-Santos, "An illustrative introduction to fisher's linear discriminant," *Thalles' Blog*, 2019. [Online]. Available: https://sthalles.github.io/fisher-linear-discriminant/

[4] M. Welling, "Fisher linear discriminant analysis," *The Rachel and Selim Benin School of Computer Science and Engineering*, 2007. [Online]. Available: https://www.cs.huji.ac.il/ csip/Fisher-LDA.pdf

[5] E. Loayza and J. David, "Herramienta de reconocimiento facial con técnica de visión computacional 2d," *Universidad de Las Américas*, 2019. [Online]. Available: https://repositorioslatinoamericanos.uchile.cl/handle/2250/2794680

[6] J. A. Rodríguez-Hernández, "Reconocimiento de expresiones faciales para interacción con el computador." *Universidad de Guanajuato*, 2016. [Online]. Available: http://repositorio.ugto.mx/handle/20.500.12059/3199

[7] K. S. Opstad, C. Ladroue, B. A. Bell, J. R. Griffiths, and F. A. Howe, "Linear discriminant analysis of brain tumour ^{1}h mr spectra: a comparison of classification using whole spectra versus metabolite quantification," *Wiley Online Library*, 2007. [Online]. Available: https://doi.org/10.1002/nbm.1147

[8] F. M. Feinberg, "Discriminant analysis for marketing research applications. wiley international encyclopedia of marketing." *Wiley Online Library*, 2010. [Online]. Available: https://doi.org/10.1002/9781444316568.wiem02029

[9] C. Maklin, "Linear discriminant analysis in python," *Towards Data Science*, 2019. [Online]. Available: https://towardsdatascience.com/linear-discriminant-analysis-in-python-76b8b17817c2