Algoritmo Perceptrón

Joaquín Amat Rodrigo j.amatrodrigo@gmail.com

Septiembre, 2018

Tabla de contenidos

Versión PDF: Github

Más sobre ciencia de datos: joaquinamatrodrigo.github.io o cienciadedatos.net

Introducción

El algoritmo *Perceptron* fue publicado en 1957 por Frank Rosenblatt. El objetivo del *Perceptron* es encontrar un hiperplano capaz de separar correctamente un conjunto de datos que sean linealmente separables, una vez obtenido el hiperplano, este puede utilizarse para clasificaciones binarias. Aunque el *Perceptron* es un algoritmo de aprendizaje muy simple, entender su funcionamiento es clave para aprender otros métodos más complejos como las máquinas de vector soporte SVM o las *redes neuronales artificiales*.

Antes de describir en detalle el algoritmo, conviene conocer una serie de términos matemáticos: el producto escalar o *dot product*, el concepto de hiperplano y el concepto de linealmente separable.

Producto escalar o *dot product*

El *dot product* es una operación entre dos vectores (de la misma dimensión) que devuelve un único valor (escalar) con información sobre la relación entre ambos vectores. Existen dos formas de interpretar el *dot product*: geométrica y algebraica.

Interpretación geométrica

Geométricamente, el *dot product* se define como el producto entre la magnitud euclídea (módulo o segunda norma) de dos vectores y el coseno del ángulo que forman. Supóngase dos vectores **x** e **y** que forman un ángulo α entre ellos. Su *dot product* es:

$$\mathbf{x} \cdot \mathbf{y} = ||\mathbf{x}||\ ||\mathbf{y}||\ cos(\alpha)$$

El *dot product* está por lo tanto influenciado por el ángulo que forman los dos vectores:

- Si $\alpha = 0$, $cos(\alpha) = 1$ y $\mathbf{x} \cdot \mathbf{y} = ||\mathbf{x}||\ ||\mathbf{y}||$. Esto significa que, si los dos vectores tienen exactamente la misma dirección, *el dot product* equivale a la multiplicación de sus magnitudes.
- Si $\alpha = 90$, $cos(\alpha) = 0$ y $\mathbf{x} \cdot \mathbf{y} = 0$. Esto significa que, si los dos vectores son perpendiculares, *el dot product* es cero. Los dos vectores tienen cero relación.
- Si $\alpha = 180$, $cos(\alpha) = -1$ y $\mathbf{x} \cdot \mathbf{y} = -||\mathbf{x}||\ ||\mathbf{y}||$. Esto significa que, si los dos vectores tienen direcciones opuestas, *el dot product* equivale al valor negativo de la multiplicación de sus magnitudes.

```r
# Implementación del dot product (geométrico)
dot_product_gemometrico <- function(x, y, alpha){
  modulo_x <- sqrt(sum(x^2))
  modulo_y <- sqrt(sum(y^2))
  aplha_radianes <- (alpha * pi) / 180
  return(modulo_x * modulo_y * cos(aplha_radianes))
}

dot_product_gemometrico(x = c(3, 5), y = c(8, 2), alpha = 45)
```

```
## [1] 34
```

Interpretación algebraica

El *dot product* se define, desde el punto de vista del álgebra, como el sumatorio del producto de cada una de sus dimensiones.

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x}_1\mathbf{y}_1 + \mathbf{x}_2\mathbf{y}_2$$

o de forma genérica:

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} \mathbf{x}_i \, \mathbf{y}_i$$

Con esta definición, no es necesario conocer el ángulo que forman los dos vectores.

```r
# Implementación del dot product (algebra)
dot_product_algebra <- function(x, y) {
  resultado <- 0
  if (length(x) != length(y)) {
    stop("La longitud de los dos vectores debe ser la misma")
  }
  for (i in seq_along(x)) {
    resultado <- resultado + x[i] * y[i]
  }
  return(resultado)
}

dot_product_algebra(x = c(3, 5), y = c(8, 2))
```

```
## [1] 34
```

Linealmente separable

El concepto de separación lineal se entiende fácilmente cuando se estudia en espacios de 2 o 3 dimensiones. En 2 dimensiones, que dos grupos de observaciones sean linealmente separables significa que existe al menos una recta que permite separar perfectamente los dos grupos. En el caso de 3 dimensiones, los datos son linealmente separables si existe un plano tal, que es capaz de separar perfectamente los grupos. Este mismo concepto puede generalizarse para cualquier número de dimensiones, la condición sigue siendo la misma, que exista un elemento de una dimensión menor que separe los grupos. Ha este elemento se le conoce como hiperplano.

Hiperplano

En geometría, un hiperplano se define como un subespacio con una dimensión menos que el espacio que lo rodea. Esta definición es poco intuitiva, pero se entiende bien si se analiza un ejemplo en dos dimensiones.

En un espacio de 2 dimensiones, un hiperplano tiene *2-1* dimensiones, es decir, una recta. La ecuación que se emplea con más frecuencia en cálculo para definir una recta es:

$$y = ax + b$$

o de forma equivalente:

$$y - ax - b = 0$$

Otra forma de definir una recta es mediante el *dot product* de dos vectores. Supóngase los vectores $\mathbf{x} = (x, y)$ y $\mathbf{w} = (w_1, w_2)$, y la constante b, con ellos puede definirse una recta de la siguiente forma:

$$\mathbf{x} \cdot \mathbf{w} + b = 0$$

$$(x, y) \cdot (w_1, w_2) + b = 0$$

$$(xw_1 + yw_2) + b = 0$$

$$y = -\frac{w_1}{w_2}x - \frac{b}{w_2}$$

Si se define a y b como:

$$a = -\frac{w_1}{w_2} \quad y \quad c = -\frac{b}{w_2}$$

Se obtiene la ecuación de una recta:

$$y = ax + c$$

La ventaja de definir una recta empleando vectores es que puede generalizarse para cualquier número de dimensiones. De hecho, esta es la forma con la que se obtiene la ecuación de un hiperplano en cualquier dimensión.

Clasificación binaria mediante un hiperplano

Tal y como se ha definido previamente, un hiperplano de dimensión n divide un espacio de dimensión $n+1$ en dos partes. Esto significa que puede emplearse a modo de clasificador binario. Las observaciones que queden por encima del hiperplano pertenecen a una clase y las que quedan por debajo a la otra. En un espacio de dos dimensiones, cada observación i esta definida por un vector $\mathbf{x}_i$ y una variable respuesta y que puede tomar dos valores (*+1, -1*). Dado un hiperplano definido por el vector $\mathbf{w}$ y la constante b, para clasificar las observaciones, se busca una función h tal que:

$$h(\mathbf{x}_i) = \begin{cases} +1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b \geq 0 \\ -1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b < 0 \end{cases}$$

lo que es equivalente a:

$$h(\mathbf{x}_i) = sign(\mathbf{w} \cdot \mathbf{x} + b)$$

La ecuación de la función h puede simplificarse excluyendo el término b. Para ello, se añade, a todos los vectores $\mathbf{x}_i$, el valor 1 en la primera posición y, al vector del hiperplano $\mathbf{w}$ el valor de b. Aunque el resultado final es el mismo, suele ser más sencillo trabajar de esta forma a la hora de implementar los algoritmos. A los vectores modificados con este fin se les conoce como vectores aumentados.

Algoritmo perceptrón

Tal como se ha descrito hasta ahora, independientemente de la dimensión, se puede crear un clasificador binario (siempre que los datos sean linealmente separables) mediante un hiperplano, pero, ¿Cómo se encuentra dicho hiperplano? Es aquí donde entra en juego el algoritmo del *Perceptron*. Dado un set de datos con m observaciones n-dimensionales $(\mathbf{x}_i, y_i)$, el *Perceptron* trata de encontrar la función h capaz de predecir correctamente la clase y_i para cada $\mathbf{x}_i$. La función h empleada por el *Perceptron* es $h(\mathbf{x}_i) = sign(\mathbf{w} \cdot \mathbf{x})$, donde $\mathbf{w} \cdot \mathbf{x}$ no es más que la ecuación de un hiperplano definida por dos vectores aumentados, de los cuales, el único desconocido es $\mathbf{w}$. Por lo tanto, el objetivo del *Perceptron* es encontrar el vector $\mathbf{w}$ que permite separar perfectamente las observaciones.

La forma en que el algoritmo *Perceptron* encuentra el hiperplano óptimo es la siguiente:

---

1. Iniciar el proceso con un hiperplano aleatorio definido por el vector aleatorio $\mathbf{w}$.
2. Clasificar todas las observaciones acorde al hiperplano $\mathbf{w}$.
3. De entre las observaciones mal clasificadas, seleccionar una de forma aleatoria (i) y con ella actualizar el hiperplano:

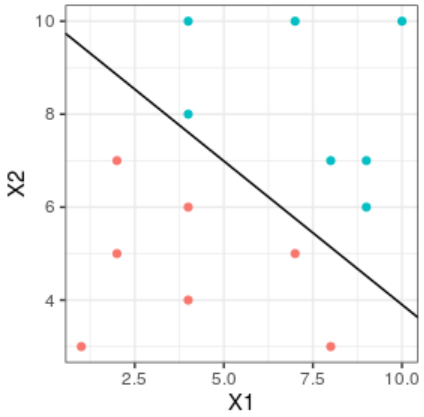
$$\mathbf{w} = \mathbf{w} + \mathbf{x}_i * y_i$$

4. Repetir los pasos 2 y 3 hasta que todas las observaciones estén bien clasificadas.

```r
perceptron <-  function(X, y, random_seed = 553){
  # Esta función implementa el algoritmo del perceptron para encontrar un
  # hiperplano que separe correctamente las dos clases.

  # Transformar X en vectores aumentados
  X <- cbind(1, X)
  # Inicialización aleatoria del hiperplano
  set.seed(random_seed)
  w <- c(runif(n = 3, min = 0, max = 1))
  # Clasificación
  clasificaciones <- predict_clase(X = X, w = w)
  # Índice de las observaciones mal clasificadas
  errores_clasificacion <- which((clasificaciones != y) == FALSE)

    while (length(errores_clasificacion) > 0) {
    # Se selecciona aleatoriamente una observación errónea
    i <- sample(x = errores_clasificacion, size = 1)
    # Actualización del hiperplano
    w <- w + X[i,] * y[i]
    clasificaciones <- predict_clase(X = X, w = w)
    errores_clasificacion <- which((clasificaciones == y) == FALSE)
  }
  return(w)
}

predict_clase <- function(X, w){
  # Esta función devuelve la clasificación de las observaciones
  # acorde al valor de sus predictores X y al hiperplano w
  clase_predicha <- apply(X = X, MARGIN = 1, FUN = function(x){crossprod(x,w)})
  clase_predicha <- sign(clase_predicha)
  return(clase_predicha)
}

# Ejemplo observaciones linealmente separables en 2 dimensiones
X <- matrix(c(8, 4, 9, 7, 9, 4, 10, 2, 8, 7, 4, 4, 1, 2, 7, 10, 7, 10, 6, 8, 10,
              7, 3, 5, 4, 6, 3, 5), ncol = 2, byrow = FALSE)
y <- c(1, 1, 1, 1, 1, 1, 1 , -1 , -1 , -1 , -1 , -1 , -1 , -1 )

hiperplano <- perceptron(X = X, y = y)
hiperplano
```

```
## [1] -28.452542   1.744047   2.822881
```

```r
library(ggplot2)
datos <- data.frame(X, y)
ggplot(data = datos, aes(x = X1, y = X2, color = as.factor(y))) +
  geom_point() +
  # La pendiente e intersección de la recta se obtienen siguiendo los pasos
  # descritos anteriormente para obtener una recta a partir dos vectores
  geom_abline(intercept = -(hiperplano[1]/hiperplano[3]),
              slope =     -(hiperplano[2]/hiperplano[3])) +
  theme_bw() +
  theme(legend.position = "none")
```



Es interesante tener en cuenta dos propiedades de este algoritmo:

- En cada actualización del hiperplano, se modifica el vector **w** intentando que clasifique bien una de las observaciones mal clasificadas (seleccionada aleatoriamente) pero sin tener en cuenta el resto. Esto significa que, una determinada actualización, puede hacer que se consiga clasificar bien la observación en cuestión pero que otra u otras que estaban bien clasificadas pasen a estar mal. Afortunadamente, los matemáticos han demostrado que el algoritmo del *Perceptron* siempre acaba encontrando un hiperplano de separación (siempre y cuando los datos sean linealmente separables).

- Como puede intuirse observando la imagen anterior, existen infinitos hiperplanos que consiguen separar las clases, cuando estas son linealmente separables. Dada la selección aleatoria de observaciones en el paso de actualización, el algoritmo genera diferentes hiperplanos cada vez que se ejecuta.

- Si bien el *Perceptron* siempre encuentra un hiperplano que separa perfectamente las clases (siempre y cuando sean linealmente separables), no tiene por qué ser el hiperplano óptimo, entendiendo por hiperplano óptimo aquel que separa correctamente las observaciones y que además se encuentra en el punto medio entre ambas clases.

Bibliografía

Introduction to Statistical Learning Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani

Support Vector Machines Succinctly by Alexandre Kowalczyk