Clase Bonus

Regresión y clasificación

Volvamos al principio del aprendizaje de máquina

Aprender:

A partir de una experiencia E, con respecto a una tarea T, y un rendimiento particular P.

Si su rendimiento en T, medido por P, se incrementa con la experiencia E se dice que se está llevando a cabo un proceso de aprendizaje.

Dos enfoques principales del aprendizaje de máquina que veremos en el curso.

Supervisado:

Hay un conocimiento del experto de que se espera que la máquina reproduzca.

$$f^*\left(\begin{bmatrix}\vec{X}_1\\ \vec{X}_2\\ \vdots\\ \vec{X}_k\end{bmatrix},\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_k\end{bmatrix}\right)=\begin{bmatrix}y_1^*\\ y_2^*\\ \vdots\\ y_k^*\end{bmatrix} \qquad \min\left(\begin{bmatrix}y_1^*\\ y_2^*\\ \vdots\\ y_k^*\end{bmatrix}-\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_k\end{bmatrix}\right)$$

Técnicas:

- Clasificación
- Regresión
- Selección de caracteristicas
- ...

No supervisado:

No se tiene un conocimiento a priori.

$$f\begin{bmatrix}\vec{X}_1\\ \vec{X}_2\\ \vdots\\ \vec{X}_k\end{bmatrix}=\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_k\end{bmatrix}$$

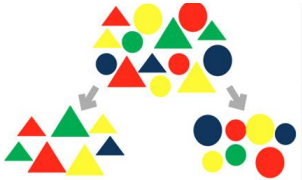
Técnicas:

- Agrupamiento
- Reducción de dimensión
- ...

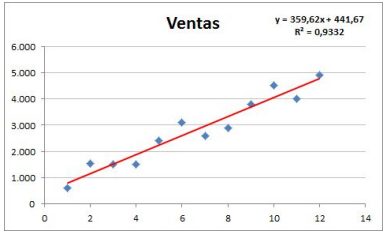
Empecemos por el lado supervisado:

Nos enfocaremos en dos técnicas: Regresión y Clasificación.

Clasificación



Regresión



Ambas tienen la misma finalidad, hacer que la máquina a partir de unos datos con etiquetas, poder predecir la etiqueta de un nuevo dato. La diferencia está que la regresión predice valores continuos mientras que la clasificación valores discretos (Clase 1, Clase 2, Clase 3, ...)

Regresión

Veremos dos enfoques para realizar la regresión de unos datos:

- Regresión lineal
- Regresión a partir del algoritmo KNN

Luego veremos que con redes neuronales también es posible realizar regresiones.

Algunas definiciones antes de comenzar:

Matriz de datos:

$$X_{m\times n} = \begin{vmatrix} X_{0,0} & X_{0,1} & \dots & X_{0,n} \\ X_{1,0} & X_{1,1} & \dots & X_{0,n} \\ \dots & \dots & \dots & \dots \\ X_{m,0} & X_{m,1} & \dots & X_{m,n} \end{vmatrix}$$

Vector de etiquetas:

$$y_{1\times n} = \begin{vmatrix} y_0 \\ y_1 \\ \dots \\ y_m \end{vmatrix}$$

Regresión lineal univariada

Parte del supuesto que la variable dependiente y, se puede predecir a partir de una combinación lineal de características de los datos. Primero resolveremos el caso para una sola variable dependiente, donde la hipótesis es:

$$h_\theta(x) = \theta_0 + \theta_1 * x$$

El objetivo de la regresión es minimizar la distancia de esta predicción a la variable y. Para esto se toma como medida el error cuadratico:

$$(h_\theta(x) - y)^2$$

Regresión lineal univariada

Si sumamos el error cuadrático de todas las muestras, tenemos:

$$\sum_{i=1}^{m} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right)^2$$

Se define entonces lo que comúnmente se llama función de costo, o mejor dicho nuestra función objetivo a minimizar:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right)^2$$

Regresión lineal univariada

Cómo solucionar este problema de optimización?

Una forma es recurrir al álgebra lineal:

$$\theta = (X^T X)^{-1} X^T y$$

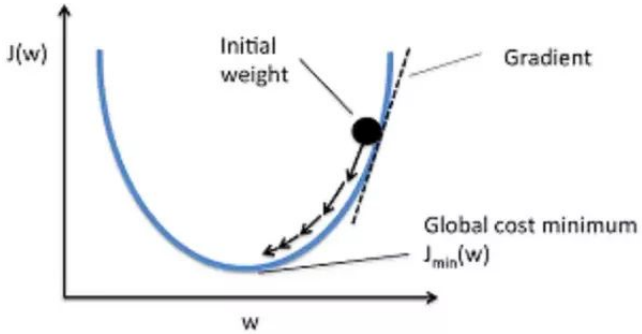
Lo cual da una solución exacta pero...

La expresión entre paréntesis no es siempre invertible.

Además cuando el número de características n es muy grande, se vuelve un cálculo demasiado lento.

Gradiente descendiente

Entonces hay otra solución (de hecho hay muchas), pero la más conocida es el concepto de gradiente descendiente.



$$w^{(i+1)} = w^{(i)} - \alpha * \frac{dJ(w)}{dw}$$

En el cual, a partir de la derivada de una función, se conoce hacia donde esta crece según su gradiente, es decir que decrece en sentido contrario al gradiente. A partir de pasos escalados por un factor alpha se va dirigiendo hacia el mínimo.

Regresión lineal univariada

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\frac{dJ}{d\theta_0} = \frac{1}{m} \sum_{i=1}^{m} (\theta_0 + \theta_1 * x^{(i)} - y^{(i)})$$

$$\theta_0^{(k+1)} = \theta_0^{(k)} - \alpha * \frac{dJ}{d\theta_0}$$

$$\frac{dJ}{d\theta_1} = \frac{1}{m} \sum_{i=1}^{m} (\theta_0 + \theta_1 * x^{(i)} - y^{(i)}) * x^{(i)}$$

$$\theta_1^{(k+1)} = \theta_1^{(k)} - \alpha * \frac{dJ}{d\theta_1}$$

Regresión lineal univariada

El proceso de aprendizaje es como sigue:

1. Se inicializan los pesos aleatoriamente.
2. Se calcula la función de costo.
3. Se actualizan los pesos de todas la variables según las derivadas, utilizando el gradiente descendiente (se utilizan todas las muestras de entrenamiento, lo que se conoce como “batch”)
4. Se devuelve al paso 2 y 3 si no se llega a un criterio de parada.

Regresión lineal multivariada

Ahora la hipótesis es predecir una variable dependiente y a partir de un conjunto de características.

$$h_\theta(x) = \theta_0 + \theta_1 * x_1 + \theta_2 * x_2 + \theta_3 * x_3 + \theta_4 * x_4 + \dots$$

$$\theta = \begin{vmatrix} \theta_0 \\ \theta_1 \\ \dots \\ \theta_n \end{vmatrix} \qquad x = \begin{vmatrix} 1 \\ x_1 \\ \dots \\ x_n \end{vmatrix}$$

$$h_\theta(x) = \theta^T * x$$

Regresión lineal multivariada

Se tiene la misma función de costo:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right)^2$$

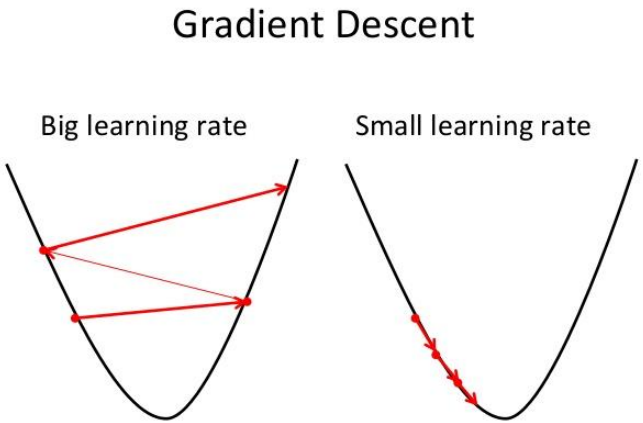
La derivada con respecto a una variable j es:

$$\frac{dJ(\theta)}{d\theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta \left(x^{(i)} \right) - y^{(i)} \right) * x_j^{(i)}$$

Se deben entonces actualizar todas las variables a la vez y luego volver a calcular la función de costo.

Aspectos que no hemos tenido en cuenta:

El parámetro alpha del gradiente descendiente es crucial para el entrenamiento, un valor muy bajo puede hacer que el entrenamiento sea muy lento. Y un valor muy alto puede hacer que el entrenamiento diverja y los parámetros oscilen en valores.



Aspectos que no hemos tenido en cuenta:

Hasta ahora solo hemos analizado la regresión lineal, es decir que las hipotesis solo generarán rectas en el caso univariado y hiperplanos en el caso multivariado. Es posible extender esta teoría a funciones cuadráticas, cúbicas and so on...

Cómo? Simplemente tratando las potencias de las variables como si fueran una variable más:

Caso cuadrático:

$$h_\theta(x) = \theta_0 + \theta_1 * x_1 + \theta_2 * x_1^2 = \theta_0 + \theta_1 * x_1 + \theta_2 * x_2$$

Caso cúbico;

$$h_\theta(x) = \theta_0 + \theta_1 * x_1 + \theta_2 * x_1^2 + \theta_3 * x_1^3$$

$$h_\theta(x) = \theta_0 + \theta_1 * x_1 + \theta_2 * x_2 + \theta_3 * x_3$$

Aspectos que no hemos tenido en cuenta:

Es posible crear nuevas variables combinando otras:

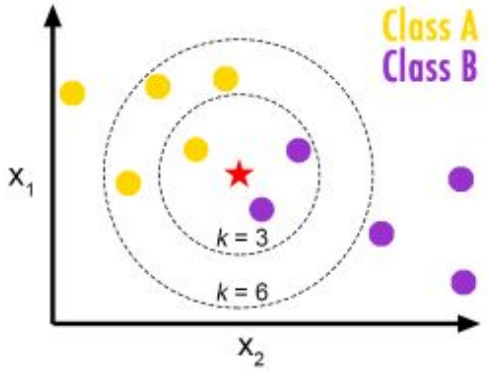
$$x_3 = \frac{x_2}{x_1}$$

$$x_3 = log(x_2)$$

Otra forma de regresión: variante del KNN

Primero, cómo funciona el KNN?

El algoritmo KNN es un algoritmo de clasificación que se basa en la hipótesis de que las muestras de una misma clase deben de estar cercanas entre sí. La lógica es muy sencilla, para cada nuevo punto se encuentran las k muestras más cercanas y el punto es asignado a la clase que más se repite en sus muestras más cercanas.



$$P(x \in \text{Class A}) = \frac{1}{3}$$

$$P(x \in \text{Class B}) = \frac{2}{3}$$

Es un algoritmo potente, pero con gran costo computacional.

Otra forma de regresión: variante del KNN

Ahora, como utilizar esta lógica en regresión?

Una vez encontradas las k muestras más cercanas, utilizamos las etiquetas de estas muestras y calculamos el promedio.

$$h(x) = \frac{1}{K} \sum_{i=1}^{K} y^V$$

donde y^V corresponde al valor de $y^{(i)}$ que acompaña al vector $x^{(i)}$ considerado vecino de x.

Método de ventana de Parzen o método Kernel

Este método construye una función de predicción dándole un peso a cada uno de los puntos en el conjunto de entrenamiento. Dicho peso será mayor cuanto más cercano esté el punto de entrenamiento al nuevo objeto x y se asigna a través de una función conocida como kernel.

$$k(u) = \frac{1}{2} * \exp(-\frac{1}{2} * u^2)$$

$$u = \frac{d(x,x_i)}{h}$$

h es la ventana de suavizado, aunque también puede verse como una desviación estandar.

Para predecir un valor de y dado una muestra x.

$$h(x) = \frac{\sum_{i=1}^{m} k\left(\frac{d(x,x^{(i)})}{h}\right) * y^{(i)}}{\sum_{i=1}^{m} k\left(\frac{d(x,x^{(i)})}{h}\right)}$$