IC-6200 Inteligencia Artificial
Apuntes clase 2024/02/20

Adrián Ramírez

I. Respuestas del Quiz #1

A. Preguntas

1) Si *u* y *v* son dos vectores colineales con magnitudes 5 y 6 respectivamente. Desarrolle: ¿cuál es el resultado del producto punto entre *u* y *v*? 40 pts.
2) Se sabe que el producto punto del vector *u* con él mismo es 25. Desarrolle: ¿cuál es su magnitud *L2*? 40 pts.
3) De acuerdo al pipeline de Machine Learning visto en clase, mencione la etapa donde se crean nuevos features a partir de features existentes. 5 pts.
4) Describa la diferencia entre un parámetro y un hiper parámetro. 15 pts.

B. Respuestas

1) Dado que dos vectores son colineales si tienen el mismo ángulo, esto significa que en la propiedad

$$u \cdot v = \|u\| \cdot \|v\| \cdot \cos\theta$$

Se cumple que la expresión $\cos\theta = 1$ y por lo tanto el resultado del producto punto es solo la multiplicación escalar de las magnitudes. Entonces $\|u\| \cdot \|v\| = 5 \cdot 6 = 30$

2) Aquí entran en juego dos propiedades que se deducen de las reglas del producto punto y la distancia euclideana (L2).

$$u \cdot u = \|u\| \cdot \|u\|$$
$$u \cdot u = \|u\|^2$$

Esta describe el producto punto de un vector sobre sí mismo, expresado como el cuadrado de su magnitud.

$$\sqrt{u \cdot u} = \sqrt{\|u\|^2}$$
$$\sqrt{u \cdot u} = \|u\|$$

Y esta propiedad describe la distancia euclideana en términos del producto punto. Teniendo eso presente, se deduce que la magnitud L2 del vector *u* es $\sqrt{25} = 5$

3) En el pipeline de Machine Learning existen las etapas de:
   a) Data Adquisition
   b) Data Preparation
   c) Feature Engineering
   d) ...

   Y en la etapa de Feature Engineering es donde se crean los nuevos features a partir de existentes o se seleccionan los más adecuados o relevantes. Esta etapa busca proporcionar features más informativos y relevantes para el problema a resolver.

4) Según el autor Burkov:

   Un hiper parámetro es una propiedad de un algoritmo de aprendizaje, usualmente con un valor numérico. Los hiper parámetros no son aprendidos por el modelo, sino que son elegidos por el analista de datos antes de correr el algoritmo. [1]

II. Ventajas v. Desventajas de k-NN

No siempre k-NN es un buen método de clasificación. Algunos problemas son sensibles al hiper parámetro, esto puede que se vea afectado por la cantidad de datos y la dimensionalidad de los atributos.

Ventajas de k-NN

Es un algoritmo simple de implementar. Es robusto y resistente al ruido (los datos "outlier" no alteran las clasificaciones).

Desventajas de k-NN

Tiene un alto costo computacional (mucho mayor a otros algoritmos). Exige mucha memoria porque hace muchas comparaciones. Necesita que se seleccione un k para operar.

III. Otro algoritmo de selección: Regresión Lineal

Este algoritmo trata de describir la pendiente de una línea (función) ficticia que se "ajusta" al comportamiento de un dataset. Con esta pendiente se puede predecir la etiqueta (label) de un nuevo dato x_n que se quiera clasificar.

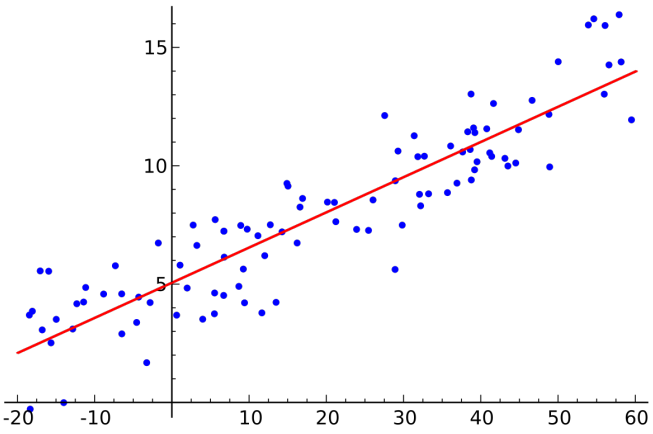

Fig. 1: Ejemplo de Regresión Lineal

En la regresión lineal si se tiene una colección de ejemplos etiquetados de la forma $\{(x_i, y_i)\}_{i=1}^{N}$ donde:

1) N es el tamaño de la colección
2) x_i es el vector D-dimensional, donde cada feature comprende la forma: $x_i^j, j = 1 \dots D$
3) y_i es la etiqueta y $y \in \mathbb{R}$
4) La pendiente de la recta está dada por la función $wx + b = \hat{y}$.

A. ¿Qué se quiere hacer?

La idea de utilizar RL es construir un modelo de la forma $f_{w,b}(x) = wx + b$. En este modelo w es el vector D-dimensional (mismo tamaño con la cantidad de features), b es un número real ($b \in \mathbb{R}$) y la predicción y es el la imagen de la función f.
En este modelo w describe los pesos de los features $[0.1, 0.09, \dots, 0.8]$ en las D-dimensiones. También se hace una combinación lineal de los features $w \cdot x$ lo que produce un valor escalar que sirve para convertir las D-dimensiones a una predicción $\hat{y}$
Este modelo de RL está parametrizado con w y b, por eso se debe encontrar los mejores valores de estos parámetros para que la función f produzca las predicciones más óptimas. (Óptimo $\neq$ Perfecto)
Se debe tener cuidado con el sobreajuste (overfitting) porque en ese extremo el modelo solo estaría aprendiendo el dataset y sería muy sensible al ruido.

B. Minimización del Error Cuadrado: MSE

Para medir la eficiencia del modelo se verán las dos funciones:

1) Función de Pérdida (Loss Function): Es una medida de penalidad del modelo. Lleva la forma:

$$(f_{w,b}(x_i) - y_i)^2$$

También se le puede entender a esta fórmula como el error cuadrático.

2) Función de Costo (Cost Function): Es el promedio de la función de pérdida se calcula de la siguiente forma:

$$L = \frac{1}{N}\Sigma((f_{w,b}(x_i) - y_i)^2)_{i=1\dots N}$$

Como dentro de la sumatoria está la definición de la función de pérdida, se le conoce como Error Cuadrático Promedio.

C. Pero, ¿qué se debe minimizar?

L es una herramienta para saber qué tanto el modelo está cometiendo errores en las predicciones de las nuevas etiquetas para datos nuevos x_{new}. Si L tiene un valor muy grande significa que el modelo no se ajusta bien al dataset. En cambio si L tiene un valor muy pequeño puede significar 2 cosas:

1) El modelo se ajusta bien a los datos.
2) $L = 0$ puede significar que hay demasiados buenos datos o hay overfitting.

D. Hallar valores de parámetros

Primero se debe formalizar la forma del modelo. A continuación unos conceptos preliminares a saber:

1) $f(x)$ tiene un mínimo local si $x = c, f(x) \geq f(c)$ para cada x que está en un intérvalo cerca de c.
2) Definimos un mínimo global como el mínimo local tomando como intervalo todo el dominio de f

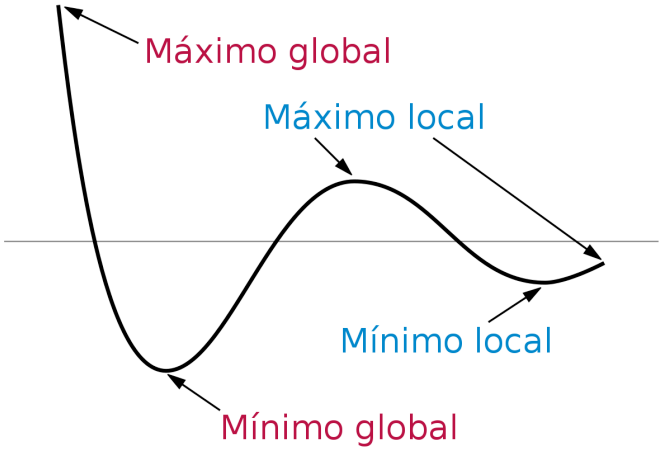

Fig. 2: Una función con los máximos y mínimos señalados

Se escoge la fórmula del MSE porque se buscan los mínimos para la función L. Como una función cuadrática describe una parábola, solo existe un mínimo para toda la gráfica de la función. Pero, ¿cómo se encuentra ese valor mínimo?

E. Derivadas para encontrar el mínimo en MSE

Las derivadas son la descripción de la pendiente de una función en un punto dado. Usando estas se puede encontrar la pendiente y la dirección para ajustar los valores y llegar más cerca del valor mínimo de L. Importante que la función de pérdida $(f_{w,b}(x_i) - y_i)^2$ también debe ser diferenciable.

Fig. 3: Al igual que un hombre bajando una pendiente, se debe buscar el recorrido abajo de la función

F. Derivadas Parciales

Son una generalización de la derivada en cálculo diferencial. Es la tasa de cambio de una función con respecto a sus variables independientes, manteniendo las demás variables como constantes.
Su notación es la siguiente, sea $f(x, y) = 2x + 3y$ se definen sus dos derivadas parciales:

$$\frac{\partial f}{\partial x} = 2$$
$$\frac{\partial f}{\partial y} = 3$$

G. Derivadas en Función Costo

Con las derivadas parciales se tiene que al desarrollar:

$$\begin{aligned} CostFunction/L &= \frac{1}{N}\Sigma((f_{w,b}(x_i) - y_i)^2)_{i=1\ldots N} \\ \frac{\partial L}{\partial w} &= \frac{1}{N}\Sigma((f_{w,b}(x_i) - y_i)^2)\prime \\ &= \ldots \\ &= \frac{1}{N}\Sigma(2((wx_i + b) - y_i) \cdot x_i) \end{aligned} \tag{1}$$

De manera similar el desarrollo de la derivada parcial $\frac{\partial L}{\partial b}$ da como resultado:

$$\frac{\partial L}{\partial b} = \frac{1}{N}\Sigma(2(wx_i + b) - y)$$

REFERENCES

[1] A. Burkov, The Hundred-Page Machine Learning Book. Place of publication unknown: Andriy Burkov, 2019.