Apuntes de Semana 5, Clase #2

Mauricio Campos Cerdas
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
maucampos@estudiantec.cr

***Abstract*—This document presents class notes on handling outliers, the concepts of bias and variance, and an introduction to logistic regression as a classification algorithm. Techniques for identifying and addressing outlying values are discussed, along with methods for splitting datasets and common scenarios encountered during training and validation. As well as the sigmoid function and parameter optimization in logistic regression are introduced, including the derivation of the sigmoid function.**

***Index Terms*—Outliers, bias, variance, logistic regression, classification, sigmoid function, parameter optimization, training and validation, Overfitting, Underfitting**

I. Noticias de la semana

A. Evento IEEE

IEEE está organizando un evento donde se tocarán temas muy interesantes, incluyendo la inteligencia artificial. Vendrán personas de gran renombre a dar charlas, habrá comida y demás. Se pide registrarse para calcular la alimentación para el día del evento.

B. Problema con las referencias y la IA

Se está produciendo un fenómeno en el que cada vez más artículos, notas y sitios web son generados con inteligencia artificial y se referencian entre sí. Esto puede llevar a que la propia IA se cite a sí misma, provocando un aumento de referencias generadas artificialmente.

C. Modelo Nano Banana

Google lanzó un nuevo modelo llamado Nano Banana. Su atractivo se encuentra que a diferencia de otros modelos, este agarra la imagen que está como input y la modifica sin tener que generarla otra vez. Se dió un ejemplo de un experimento donde una IA tenía que modificar una foto varias veces y se llegó a evidenciar que hubo un sesgo de generar la imagen de la persona cada vez con rasgos más latinos.

II. Potenciales problemas de la Regresión Lineal

- **No linealidad:** En regresión lineal se asume que existe una relación lineal entre las variables predictoras y la variable respuesta. Sin embargo, esto no siempre se cumple, lo que provoca que el modelo no capture adecuadamente la relación y que los residuos presenten patrones sistemáticos (por ejemplo, con forma parabólica) en lugar de distribuirse aleatoriamente (ver Fig. 1). Una estrategia para enfrentar este problema es aplicar *feature engineering*. Un ejemplo es incorporar términos polinómicos adicionales a las variables, lo que permite aproximar mejor relaciones no lineales.

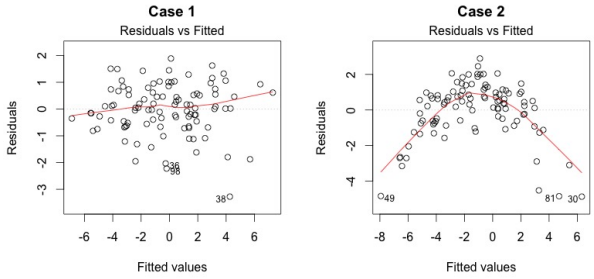

Fig. 1. Residual plots

- **Datos sobresalientes:** Siempre existirán outliers, ya sea por ruido o error humano. Lo que pasa es que nos afecta a nuestro modelo, siempre habrá cierta sensibilidad hay que tratarlos para evitar que nos afecte en gran medida nuestro modelo.

A. Métodos para tratar outliers

- **Standardized Residuals:** Tenemos el cálculo de los residuos y calculamos la desviación estándar, para asegurarnos de que nuestros datos siguen una distribución normal. A partir de que los tenemos estandarizados, calculamos a cuántas desviaciones estándar se encuentra ese dato. Lo que nos dirá es el límite de hasta dónde se consideran datos sobresalientes.
  - $|z| > 2$: posible outlier.
  - $|z| > 3$: outlier muy probable, se recomienda excluir.
- **Regla del rango intercuartílico (IQR):** Definido como $IQR = Q3 - Q1$. Los datos que se encuentran fuera del intervalo $[Q1 - 1.5 \cdot IQR, Q3 + 1.5 \cdot IQR]$ se consideran outliers.
- **Winsorización:** Técnica que consiste en reemplazar los valores extremos por los percentiles límite (por ejemplo, 5% y 95%).

III. Sesgo y Varianza

El dataset suele dividirse en train y test (80/20)

A. Training set

Se utiliza para ajustar el modelo. Nos puede pasar que entrenemos el modelo mucho tiempo, lleguemos al final y

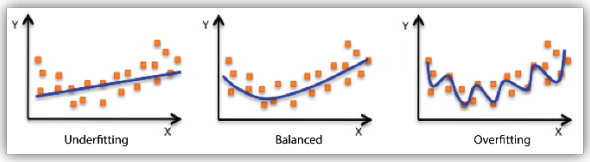

Fig. 2. Underfitting, Ideal, Overfitting plots

nos damos cuenta de que fallamos el examen. Si dedicamos mucho al entrenamiento pero nada a generalizar, se llama overfitting. Por eso queremos hacer tests pequeños durante el entrenamiento, con el validation set.

B. Validation set

Nos dice si los hiperparámetros son adecuados o no, para no continuar si no lo son y así no desperdiciar recursos.

C. Técnicas de subdividir el dataset

- **Random Sampling:** Se usa siempre que tengamos clases balanceadas. Si los datos no están balanceados, pueden quedar mal distribuidos, con más datos de una clase que de la otra.
- **Stratified Sampling:** Usado para datos imbalanceados, asegura una representación de todas las clases por separado.
- **K-Fold Cross-Validation:** División en k partes, en cada iteración se usan $k-1$ para entrenamiento y 1 para validación.

D. Escenarios posibles

- **Escenario ideal:** El modelo presenta bajo error tanto en training como en testing. Puede evitar el ruido de los datos y generalizar correctamente. Por cada época de entrenamiento el error debería ir disminuyendo, tendiendo siempre a la baja.
- **Overfitting:** Ocurre cuando el error en el validation set empieza a crecer o se estanca. Esto indica que el modelo era bueno hasta cierta época de entrenamiento, pero luego empieza a sobreajustarse a los datos de entrenamiento, produciendo overfitting. A esta técnica de detener el entrenamiento antes de que esto suceda se le llama early stopping.
- **Underfitting:** Se da cuando el error es alto tanto en training como en testing. Esto se conoce como underfitting, que ocurre cuando el modelo no logra ajustarse correctamente a los datos. Es lo opuesto al overfitting y se caracteriza por un alto sesgo. Para ver gráficamente estos escenarios, ver Fig. 2).
- **Bias-Variance Tradeoff:** Validación con buen resultado, pero entrenamiento con alto error. Es raro que suceda y tal vez hay errores de cálculo.

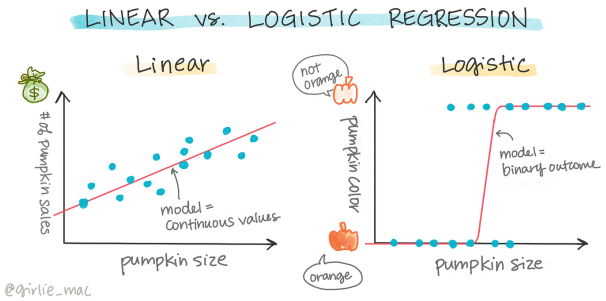

Fig. 3. Linear vs Logistic Regression

E. Alto Bias

Cuando el modelo comete muchos errores en el training set, se produce underfitting. Esto ocurre porque el modelo asume demasiado del training set, no utiliza todas los features disponibles y es demasiado simple para capturar la complejidad de los datos. Para evitar un alto sesgo, se puede utilizar un modelo más complejo. Además, es importante revisar que los features del training set sean adecuadas para la naturaleza del problema, ya que si no tienen la capacidad de capturar la información relevante, el modelo no podrá hacer predicciones correctas.

F. Alta Varianza

Ocurre cuando el modelo se ajusta demasiado a los datos de entrenamiento y no es capaz de generalizar correctamente. Esto suele suceder cuando los datos son de alta dimensionalidad y hay pocos ejemplos disponibles. Para evitar la alta varianza, se pueden usar modelos más simples, reducir la dimensionalidad de los datos, obtener más ejemplos y aplicar técnicas de regularización.

IV. Regresión Logística

Aunque su nombre contenga la palabra regresión, en realidad la regresión logística es un algoritmo de clasificación binaria. Distingue entre dos clases (0 y 1), estimando probabilidades. Fig. 3).

A. Distribución de Bernoulli

Utilizamos una distribución de Bernoulli para la ocurrencia de un evento binario.

$$P(Y=k) = p^k(1-p)^{1-k}, \quad k \in \{0,1\}$$

B. Función Sigmoide

Es una función que no se comporta linealmente. Tiene un Codominio de [0, 1]

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Nota: x puede ser cualquier número, hasta el resultado de otra función. Ver Fig. 4).

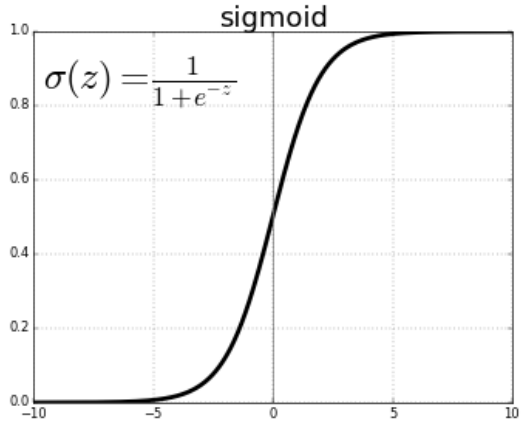

Fig. 4. Sigmoid plot

C. Clasificador

- Si $y < 0.5$, se clasifica como 0.
- Si $y \geq 0.5$, se clasifica como 1.

El umbral puede ajustarse según el problema.

D. Modelo combinado

Al aplicar la sigmoide a una función lineal $f_{w,b}(x) = wx + b$, obtenemos:

$$f_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}}$$

La relación de los features y pesos se da por regresión lineal. Lo que nos da es la probabilidad de que un evento suceda.

E. Optimización

En la regresión logística necesitamos optimizar los pesos w y el sesgo b. Para actualizar estos pesos, es necesario contar con una función de pérdida L que sea adecuada para probabilidades, ya que el MSE ya no es lo apropiado en este caso.

F. Derivada de la sigmoide

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Usando la regla del cociente:

$$\sigma'(x) = \frac{1' \cdot (1 + e^{-x}) - (1 \cdot (1 + e^{-x})')}{(1 + e^{-x})^2}$$

$$\sigma'(x) = \frac{0 - 1 \cdot (1' + (e^{-x})')}{(1 + e^{-x})^2}$$

$$\sigma'(x) = \frac{-(0 - (e^{-x}))}{(1 + e^{-x})^2}$$

$$\sigma'(x) = \frac{e^{-x}}{(1 + e^{-x})^2}$$

$$\sigma'(x) = \frac{e^{-x} + 1 - 1}{(1 + e^{-x})^2}$$

$$\sigma'(x) = \frac{e^{-x} + 1}{(1 + e^{-x})^2} - \frac{1}{(1 + e^{-x})^2}$$

De la fracción izquierda, puedo cancelar

$$\sigma'(x) = \frac{1}{(1 + e^{-x})} - \frac{1}{(1 + e^{-x})^2}$$

Aplicamos factor común

$$\sigma'(x) = \frac{1}{(1 + e^{-x})} \cdot (1 - \frac{1}{(1 + e^{-x})})$$

Como $\frac{1}{(1+e^{-x})} = \sigma(x)$, decimos que:

$$\sigma'(x) = \sigma(x)\big(1 - \sigma(x)\big)$$

REFERENCES

[1] Amazon Web Services, "Model Fit: Underfitting vs. Overfitting,". Available: https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html.

[2] University of Virginia Library, "Understanding Diagnostic Plots for Linear Regression Analysis,". Available: https://library.virginia.edu/data/articles/diagnostic-plots.

[3] ML4A, "Neural networks,". Available: https://ml4a.github.io/ml4a/es/neural_networks/.