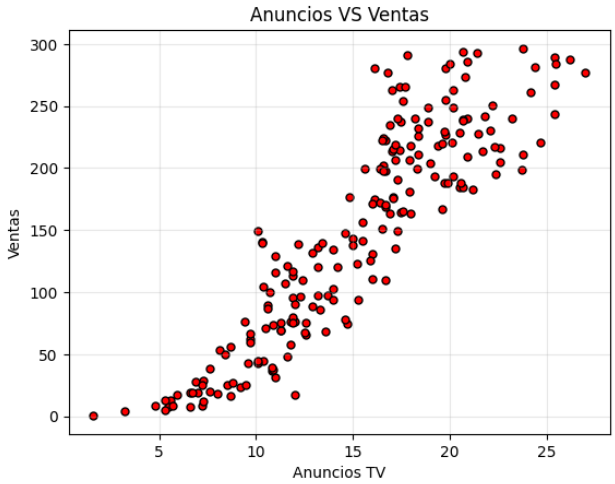
Semillero de Redes Neuronales

Universidad Nacional de Colombia – Sede Medellín
Facultad de Ciencias
Departamento de Matemáticas
Joan Sebastián Salazar Montoya

Regresión lineal

En esta sesión vamos a explorar más a fondo el modelo de regresión lineal. Hagamos un recorderis. Consideremos el siguiente conjunto de datos. En el eje X encontramos anuncios de televisión, y el eje Y corresponde a las ventas.



Parece que los datos obedecen una cierta relación "lineal". Lo que esto quiere decir es que las ventas de un producto son proporcionales a la cantidad de presupuesto que se invierte en anuncios y propagandas para ese producto. Matemáticamente, lo que esto quiere decir es que si y representa las ventas en un mes, y x representa el presupuesto invertido en anuncios en ese mes, tenemos que

$$y = \beta_1 x$$

donde β_1 es la constante de proporcionalidad entre las ventas y el presupuesto invertido en anuncios en un mes dado.

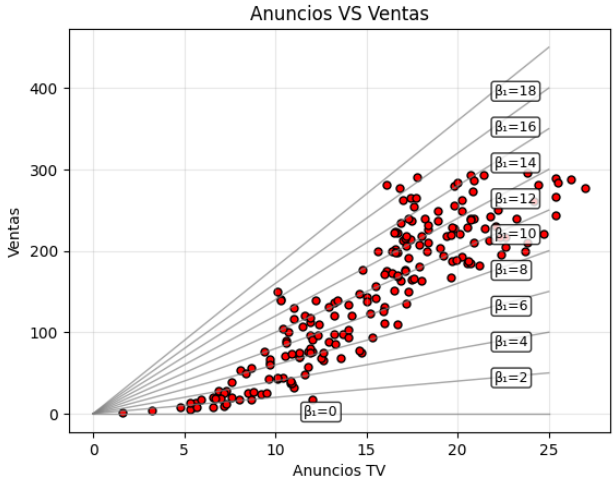
Digamos que queremos saber cuántas ventas podríamos obtener si invertimos 20 millones de pesos en anuncios. Sea $\hat{y}$ el valor de tal predicción. Entonces

$$\hat{y} = \beta_1 \cdot 25$$

corresponde al número de ventas que obtendríamos si invirtiéramos 25 millones de pesos en anuncios. El problema es que de momento *no conocemos* la constante de proporcionalidad β_1. ¿Qué podemos hacer para hallarla? Notemos que la ecuación

$$y = \beta_1 x$$

corresponde a la *ecuación de una recta* que *pasa por el origen*. Intuitivamente, queremos la recta que mejor se "adecue" al conjunto de datos. En la siguiente gráfica podemos ver varios candidatos de rectas, a partir de varios valores de β_1.



Parece que los valores de $\beta_1 = 10, 12$ son buenos candidatos, ya que las líneas rectas que tienen esas pendientes parecen adecuarse bien al conjunto de datos. ¿Cuál deberíamos escoger? Necesitamos una medida *numérica* que nos indique qué tan bien una recta se adecua al conjunto de datos. Hay mucha formas de hacerlo, pero en este caso vamos a elegir el *error cuadrático medio* (MSE por sus siglas en inglés) para medir el error. Para medir qué tan bien se adecua la recta al conjunto de datos, hacemos lo siguiente:

1. Evaluamos el modelo en cada ejemplo x de X. Llamemos $\hat{y}_i = \beta_1 x_i$, es decir, $\hat{y}_i$ es la predicción del modelo en el ejemplo i-ésimo.

2. Calculamos la diferencia entre predicción del modelo con el valor real y_i del dato. Esto es, calculamos $\hat{y}_i - y_i$. Esta diferencia corresponde al *error* entre la predicción del modelo y el valor verdadero.

3. No queremos que el error sea negativo. Por lo tanto, la evaluamos al cuadrado y calculamos $(\hat{y}_i - y_i)^2$. Además de hacer que el error no sea negativo, elevar al cuadrado "magnifica" los errores grandes, y "perdona" los errores pequeños.

4. Por último, sumamos todos estos términos, y dividimos entre el número de datos para obtener un promedio:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

Esta es la razón por la que se llama *error cuadrático medio*: es el promedio de los errores al cuadrado. Con esto, ya tenemos una medida para evaluar qué tan bien una recta se adecua al conjunto de datos: la mejor será aquella que tenga el menor MSE.

Para el ejemplo anterior, la recta que devuelve el menor MSE es aquella con pendiente $\beta_1 = 1$, y su MSE es de 1994.58

Digresión de álgebra lineal

Digamos que queremos crear un vector $\hat{y}$ tal que en la i-ésima componente contenga la i-ésima predicción $\hat{y}$. Esto es, queremos que $\hat{y}_i = \beta_1 x_i$. Recordemos que la matriz de diseño X es un vector $n \times 1$. Por lo tanto lo podemos multiplicar a derecha por una matriz 1×1 para obtener un vector $n \times 1$. Llamemos β al vector $\beta = [\beta_0]$ (en este caso, solo tenemos un parámetro). Luego, podemos expresar el vector $\hat{y}$ como

$$\hat{y} = X\beta.$$

El lector puede estar pensando que simplemente hemos reescrito la ecuación de una forma más complicada de entender, y que no nos da nada nuevo. Le ruego al lector que compruebe que la ecuación anterior tiene sentido, y le prometo que su esfuerzo se verá recompensando en muy poco tiempo.

Recta con intercepto

Volvamos al problema: encontrar la recta que mejor se adecue a los datos. Hemo encontrado que la recta que se adecua mejor a los datos de *entre las candidatas propuestas* es aquella con pendiente $\beta_1 = []$. Sin embargo, ¿es esta la mejor recta de *todas las posibles rectas*?

Hay un problema con nuestro modelo, y es que solo permite considerar rectas *que pasan por el origen.* Nos gustaría que el modelo tuviera más flexibilidad, y permitiera predecir a partir de rectas que no necesariamente pasen por el origen. Formalmente, queremos que la predicción $\hat{y}_i$ del modelo en el ejemplo x_i tenga la forma

$$\hat{y}_i = \beta_0 + \beta_1 x_i.$$

La ecuación anterior corresponde a una recta con pendiente β_1 e intercepto β_0 con el eje y. Ahora tenemos que encontrar tanto el parámetro β_0 como el parámetro β_1 que minimizan la solución *al mismo tiempo.* Podríamos volver a hacerlo a "ojo", calculando varias rectas, con varios interceptos y pendientes, elegir unos candidatos, y elegir aquel que tenga el menor error cuadrático medio. Sin embargo, hay una manera mucho más eficaz de encontrar los parámetros, la cual nos permitirá encontrar los parámetros *perfectos*, es decir, la mejor pareja β_0 y β_1 que hace que la línea recta con intercepto β_0 y pendiente β_1 se adecue al conjunto de datos lo mejor posible.

Argumento geométrico de la solución

Presentamos un argumento geométrico de la solución. Primero, ¿cómo podemos expresar el vector $\hat{y}$ de predicciones con una fórmula matricial? Ahora es cuando recogemos los frutos de la digresión anterior de álgebra lineal. Primero, notemos que podemos expresar cada $\hat{y}_i$ como un producto punto de la forma

$$\hat{y}_i = \begin{bmatrix} 1 \\ x_i \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \beta_0 + \beta_1 x_i$$

Viendo al primer vector como una matriz, podemos reescribir la expresión anterior como una multiplicación de matrices:

$$\hat{y}_i = \begin{bmatrix} 1 & x_i \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \beta_0 + \beta_1 x_i$$

Recordemos que dadas dos matrices A, B solo podemos multiplicarlas si la dimensión de A es $n \times m$, y la dimensión de B es $m \times k$. El producto AB será una matriz de tamaño $n \times k$, las dimensiones de los "extremos". Por lo tanto, si la matriz de diseño X tiene la forma

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$$

entonces tiene dimensión $n \times 2$. Por lo tanto, podemos multiplicar por cualquier matriz de tamaño $2 \times k$. De esto se sigue que si expresamos

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

entonces el producto $X\beta$ tiene sentido. Se invita al lector a que compruebe que

$$X\beta = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 x_1 \\ \beta_0 + \beta_1 x_2 \\ \vdots \\ \beta_0 + \beta_1 x_n \end{bmatrix}$$

y por lo tanto $X\beta = \hat{y}$, por lo que hemos logrado expresar el vector de predicciones como un producto matricial.

Ahora, veamos más cuidadosamente la fórmula del error cuadrático medio:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

Veamos la sumatoria anterior de una forma más inocente:

$$\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 = \frac{1}{n} [(\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + ... + (\hat{y}_n - y_n)^2].$$

Recordemos que si $v, u \in \mathbb{R}^n$, entonces su distancia euclídea está dada por

$$dist(u, v) = ||u - v|| = \sqrt{(u_1 - v_1)^2 + ... + (u_n - v_n)^2}.$$

Por lo tanto, tiene sentido interpretar la sumatoria del MSE como la distancia entre dos vectores (quitando la raíz cuadrada, por supuesto). Llamemos y al vector cuya componente y_i es la etiqueta real del ejemplo x_i, y llamemos $\hat{y}$ al vector que en la i-ésima componente tiene la predicción $\hat{y}_i$ del ejemplo x_i. Entonces la distancia al cuadrado entre ambos vectores es precisamente

$$dist(\hat{y}, y)^2 = ||\hat{y} - y||^2 = (\hat{y}_1 - y_1)^2 + ... + (\hat{y}_n - y_n)^2 = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

De esto se sigue que

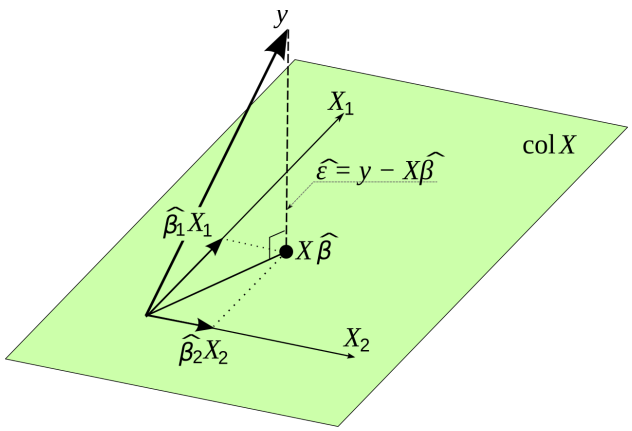
$$MSE = \frac{1}{n}||\hat{y} - y||^2.$$

Pero recordemos por lo discutido al inicio que $\hat{y} = X\beta$. En consecuencia

$$MSE = \frac{1}{n}||X\beta - y||^2.$$

Por lo tanto, encontrar el mejor vector de parámetros β, significa encontrar el β que minimice la distancia entre $X\beta$ y y. Hemos trasladado el problema de optimización al mundo de la geometría, que podemos visualizar.

Ahora, recordemos que como β es un vector y X es una matriz, el producto $X\beta$ es una combinación lineal de las columnas de X con coeficientes las componentes de β. Por lo tanto el vector $X\beta$ está en el subespacio vectorial generado por las columnas de X. Es decir, $X\beta$ está en el *espacio columna* de X. Como X tiene dos columnas, el espacio columna de X corresponde a un plano. Si el vector $X\beta$ en el espacio columna estuviera a la distancia mínima de y, entonces el vector $X\beta - y$ debe ser ortogonal al espacio columna de X, es decir, al plano generado por las dos columnas de X. La siguiente imagen puede aclarar la confusión de la discusión anterior:



Para que un vector v sea ortogonal al espacio columna de una matriz A, v debe ser ortogonal a todas las columnas de A. Esto es,

$$A^T v = 0,$$

donde A^T corresponde a la transpuesta de A. Se invita al lector que compruebe con un ejemplo que la fórmula anterior significa exactamente que v es ortogonal a cada columna de A. Con esto en mente, la condición de que el vector $X\beta - y$ sea ortogonal al espacio columna de X se escribe como

$$X^T(X\beta - y) = 0.$$

Operando la ecuación anterior obtenemos

$$X^T X\beta - X^T y = 0$$
$$X^T X\beta = X^T y,$$

suponiendo que $X^T X$ es invertible, podemos multiplicar por $(X^T X)^{-1}$ a ambos lados de la ecuación y obtener

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

y con esto hemos conseguido el vector de parámetros β correspondiente a la recta que mejor se adecua a los datos con una *fórmula cerrada*.

Notemos que la derivación anterior dependió crucialmente de que la matriz $X^T X$ fuera invertible. ¿Cómo puede fallar esta condición? Si X tuviera filas o columnas linealmente dependientes, entonces el producto $X^T X$ no es invertible. Si tenemos en cuenta que X es una matriz cuyas filas corresponden

a ejemplos y cuyas columnas corresponden a características, esto significa que

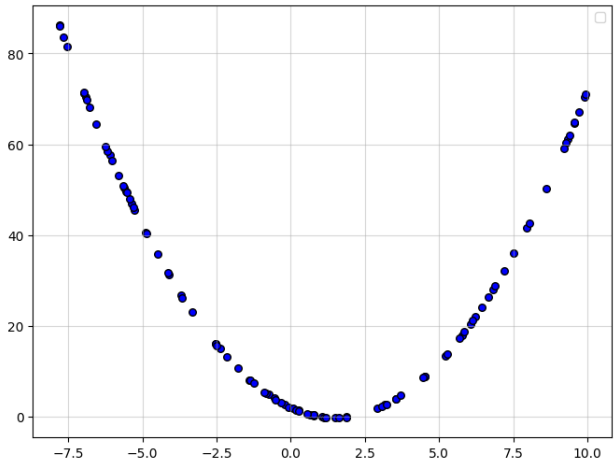
1. No puede haber ejemplos linealmente dependientes en el conjunto de datos. En particular, no pueden haber *ejemplos repetidos*.

2. No puede haber características linealmente dependientes en el conjunto de datos. En particular, no pueden haber *características repetidas*.

En otras palabras, necesitamos que no hayan "redundancias" en el conjunto de datos.

Ingeniería de características

Consideremos el siguiente conjunto de datos:



Es claro que el conjunto de datos no presenta una relación lineal, sino más bien una relación "polinómica". En particular, parece que el conjunto de datos sería bien ajustado por un polinomio de la forma $\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2$. El problema consiste en encontrar los parámetros β_0, β_1 y β_2 del modelo. ¿Será que la regresión lineal puede ayudarnos? Podríamos apresurarnos a responder que no, y argumentar que la regresión *lineal* solo permite crear modelos *lineales*, es decir, líneas rectas, y que estamos desamparados en el caso de predecir algo como los coeficientes de un polinomio. Sin embargo, no

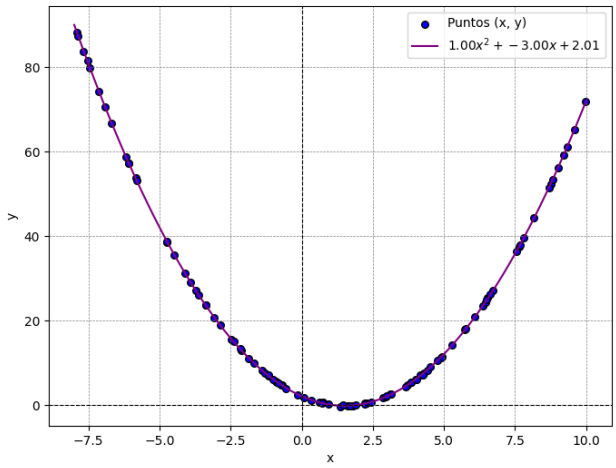
estamos tan embalados como parece. Primero, notemos que para el ejemplo i-ésimo podemos interpretar

$$\hat{y}_i = \begin{bmatrix} 1 \\ x_i \\ x_i^2 \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$$

Con esto, podemos pensar en hacer algo similar a cuando añadimos la columna de 1's a la matriz de diseño para poder considerar rectas que no pasaran por el origen. En este caso, a cada ejemplo le estamos añadiendo la característica x_i^2, de tal manera que ahora la matriz de diseño tiene la forma

$$X = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \\ 1 & x_n & x_n^2 \end{bmatrix}$$

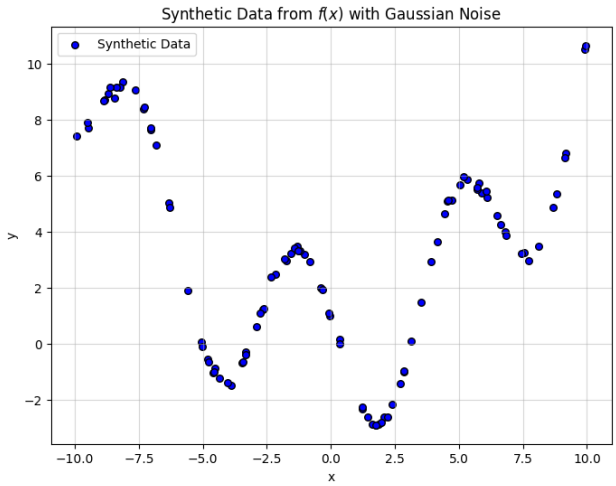
Estamos añadiendo una nueva *característica* al conjunto de datos. Notemos que como estamos añadiendo x_i^2, esta no es una combinación lineal de las columnas ya presentes anteriormente, por lo que no hay problemas con la dependencia lineal de las columnas. Es decir, estamos añadiendo una nueva característica que no depende linealmente de las otras. Si calculamos el vector de parámetros $\beta = [\beta_0 \quad \beta_1 \quad \beta_2]$ mediante la fórmula cerrada de la solución, y graficamos el polinomio, obtenemos



Es decir, el ¡modelo de regresión lineal permite encontrar curvas de predicción no lineales! La observación crucial es que para lograr esto, teníamos que saber

de antemano cual era *exactamente* la característica que teníamos que añadir al conjunto de datos. En general, si conocemos *exactamente* cual es el la curva "subyacente" a los datos, el problema de cierta forma ya está resuelto: simplemente tenemos que añadir esta información al conjunto de datos. Por ejemplo, el siguiente conjunto de datos fue generado añadiendo ruido a la función

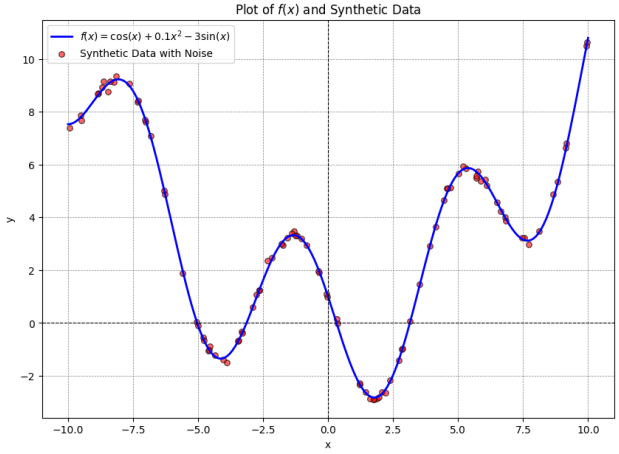
$$f(x) = \cos(x) + 0.1x^2 - 3\sin(x)$$



Si conocemos *de antemano* que el proceso generador de datos viene de una combinación lineal de las funciones $\cos(x), \sin(x), x^2$, entonces basta añadir estas características a la matriz de diseño

$$X = \begin{bmatrix} \cos(x_1) & \sin(x_1) & x_1^2 \\ \cos(x_2) & \sin(x_2) & x_2^2 \\ \vdots & \vdots & \vdots \\ \cos(x_n) & \sin(x_n) & x_n^2 \end{bmatrix}$$

y simplemente calcular el vector de parámetros β a partir de la solución cerrada que ya conocemos. Para este caso, obtenemos una curva que se ajusta muy bien a los datos



El detalle es que para los conjuntos de datos que solemos encontrar en lo salvaje *no sabemos cual es el proceso generador de datos*. Este es precisamente el problema a la hora de crear un modelo, ¿cuales son las características importantes que debería tener el modelo sobre los datos? De aquí viene la importancia de tener conjuntos de datos de calidad, con características pertinentes al problema que se quiere resolver.

El proceso de construir manualmente nuevas características que permitan crear modelos más flexibles y que realicen mejores predicciones se conoce como *ingeniería de características*. Construir características para un conjunto de datos requiere tener un cierto dominio y experticia sobre el conjunto de datos. Por ejemplo, si tenemos un conjunto de datos que contiene las características $[peso \quad altura]$, y queremos predecir si una persona sufre de problemas de salud, sería pertinente *construir* una nueva característica llamada IMC (índice de masa corporal)

$$IMC = \frac{peso}{estatura^2},$$

y sería razonable pensar que al añadir esta característica al conjunto de datos, el modelo que estemos entrenando tendrá una pieza extra de información sobre la cual trabajar.

El problema radica en que para algunos problemas, no es nada claro cuales son las características más relevantes para el problema. Por ejemplo, para el caso del reconocimiento de rostros en una imagen, ¿cómo creamos

nuevas características a partir de una imagen? O si tenemos un audio, ¿cómo construimos características que nos permitan obtener el texto de lo que se dice en el audio? Nos gustaría que hubiera una forma de que no tuviéramos que crear las características a mano, sino que el algoritmo de aprendizaje las encontrara por sí solo, de alguna forma. Este es exactamente el problema que solucionan las redes neuronales.