Laboratorio de Datos
Entrenamiento y testeo

Primer Cuatrimestre 2024
Turnos tarde y noche

Facultad de Ciencias Exactas y Naturales, UBA

¿Cómo elegir entre distintos modelos?

Toda tarea de «aprendizaje automático», «machine learning» o «inteligencia artificial», consiste en:

1. Tomar un problema relevante del mundo material.
2. Elegir un modelo matemático que lo represente. El modelo en general va a depender de ciertos parámetros $\beta_1, \ldots, \beta_s$. Por ejemplo en un modelo de regresión lineal, estos parámatros son los coeficientes de cada variable.
3. Definir una forma de medir qué tan bueno es un modelo, en relación a la realidad (vía los datos disponibles). Esto suele hacerse mediante una *función de pérdida*.
4. ...

¿Cómo elegir entre distintos modelos?

Función de pérdida: Si fijamos los parámetros de nuestro modelo $\beta_1, \ldots, \beta_s$ y aplicamos el modelo resultante a un conjunto de datos X, la función de pérdida mide cuánto "perdemos" utilizando el modelo en lugar de los datos reales. En regresión lineal, la función de pérdida más común es el *error cuadrático medio (MSE)*.

¿Cómo elegir entre distintos modelos?

Toda tarea de «aprendizaje automático», «machine learning» o «inteligencia artificial», consiste en:

1. Tomar un problema relevante del mundo material.
2. Elegir un modelo matemático que lo represente. El modelo en general va a depender de ciertos parámetros $\beta_1, \ldots, \beta_s$. Por ejemplo en un modelo de regresión lineal, estos parámatros son los coeficientes de cada variable.
3. Definir una una *función de pérdida*.
4. "Aprender" los coeficientes $\beta_1, \ldots, \beta_s$. Es decir, encontrar valores $\beta_1, \ldots, \beta_s$ que minimicen la pérdida.

Planteamos el sistema lineal

Llamamos c_1, c_2 y c_3 a la cantidad de acciones de cada tipo. Para calcular los valores, tenemos que resolver el siguiente sistema de ecuaciones:

$$
\begin{array}{lrcccc}
 & & YPF & Santander & Nvidia \\
 & & \downarrow & \downarrow & \downarrow \\
\text{Día 1} \rightarrow & 170262.00 &= 20935c_1 &+ 20100c_2 &+ 37100.0c_3 \\
\text{Día 2} \rightarrow & 169929.50 &= 21030c_1 &+ 20500c_2 &+ 36255.0c_3 \\
\text{Día 3} \rightarrow & 171064.00 &= 20770c_1 &+ 21700c_2 &+ 36000.0c_3 \\
\text{Día 4} \rightarrow & 169637.35 &= 20950c_1 &+ 21000c_2 &+ 35645.5c_3 \\
\text{Día 5} \rightarrow & 164625.45 &= 20750c_1 &+ 20316c_2 &+ 33878.5c_3
\end{array}
$$

Resolvemos el sistema

Como tenemos 3 incógnitas, nos alcanza con 3 ecuaciones:

$$\begin{aligned} 170262.00 &= 20935c_1 + 20100c_2 + 37100.0c_3 \\ 169929.50 &= 21030c_1 + 20500c_2 + 36255.0c_3 \\ 171064.00 &= 20770c_1 + 21700c_2 + 36000.0c_3 \end{aligned}$$

Para resolver el sistema, construimos la matriz ampliada

$$\left(\begin{array}{ccc|c} 20935.0 & 20100.0 & 37100.0 & 170262.00 \\ 21030.0 & 20500.0 & 36255.0 & 169929.50 \\ 20770.0 & 21700.0 & 36000.0 & 171064.00 \end{array}\right)$$

¿Qué hay en las primeras 3 columnas de la matriz? ¿Qué hay en la última columna?

Solución del sistema

Triangulando la matriz y despejando, obtenemos los valores

$$c_1 = 3.2, \qquad c_2 = 2.0, \qquad c_3 = 1.7.$$

Estas son las cantidades de cada acción que tiene el fondo de inversión.

Notación matricial

Podemos escribir el sistema de ecuaciones en forma compacta usando notación matricial:

$$\begin{pmatrix} 20935.0 & 20100.0 & 37100.0 \\ 21030.0 & 20500.0 & 36255.0 \\ 20770.0 & 21700.0 & 36000.0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 170262.00 \\ 169929.50 \\ 171064.00 \end{pmatrix}$$

Obtenemos un sistema de la forma

$$Xc = y$$

¿Qué hay en las columnas de X? ¿Que hay en la matriz y?

Verificación del "modelo"

En base a los resultados que obtuvimos, ¿podemos confirmar que las acciones del fondo son las 3 acciones que usamos?

Si no estamos seguros si eran acciones de Santander o Galicia, ¿cómo podríamos asegurarnos?

Cambiamos los valores de Santander por los valores de Galicia:

$$
\begin{aligned}
170262.00 &= 20935c_1 + \mathbf{20100}c_2 + 37100.0c_3 \\
169929.50 &= 21030c_1 + \mathbf{19400}c_2 + 36255.0c_3 \\
171064.00 &= 20770c_1 + \mathbf{21900}c_2 + 36000.0c_3
\end{aligned}
$$

Sobreajuste

Resolvemos el sistema y obtenemos estos valores:

$$c_1 = 6.69507872, \qquad c_2 = 1.16828332, \qquad c_3 = 0.17838362$$

Los números no se ven menos redondos, pero eso no alcanza para decidir cuál es el modelo correcto.

Un sistema de 3 ecuaciones y 3 incógnitas en general **siempre** tiene solución.

Sobreajuste (overfitting). Cuando tenemos igual cantidad de parámetros que observaciones, el sistema (casi) siempre va a tener solución pero no nos da ninguna información sobre si el modelo es correcto, no podemos usarlo para estimar otros valores.

Selección de modelos

¿Qué estrategias se les ocurren para ver cuál modelo es mejor?

Selección de modelos

¿Qué estrategias se les ocurren para ver cuál modelo es mejor?

1. Verificar la fórmula en otros días.

Selección de modelos

¿Qué estrategias se les ocurren para ver cuál modelo es mejor?

1. Verificar la fórmula en otros días.
2. Utilizar más días al plantear el sistema de ecuaciones.

Selección de modelos

¿Qué estrategias se les ocurren para ver cuál modelo es mejor?

1. Verificar la fórmula en otros días.
2. Utilizar más días al plantear el sistema de ecuaciones.

Las dos estretegias son ideas centrales en la construcción de modelos:

1. Probar el modelo en datos distintos a los que usamos para construir el modelo.
2. Utilizar la mayor cantidad posible de datos para construir el modelo.

1. Conjuntos de entrenamiento y testeo

Separamos nuestro conjunto de datos en dos subconjuntos:

- **Conjunto de entrenamiento.** Lo utilizamos para construir el modelo. En un modelo lineal, lo usamos para calcular los coeficientes (c_1, c_2, c_3).
- **Conjunto de testeo.** Lo utilizamos para verificar si el modelo construido ajusta bien a los datos en este conjunto.

2. Más ecuaciones que variables - Ejemplo de juguete

Si consideramos el sistema original, tenemos 5 ecuaciones y 3 variables.

$$
\begin{array}{lcccc}
 & & YPF & Santander & Nvidia \\
 & & \downarrow & \downarrow & \downarrow \\
\text{Día 1} \rightarrow & 170262.00 & = 20935c_1 & + 20100c_2 & + 37100.0c_3 \\
\text{Día 2} \rightarrow & 169929.50 & = 21030c_1 & + 20500c_2 & + 36255.0c_3 \\
\text{Día 3} \rightarrow & 171064.00 & = 20770c_1 & + 21700c_2 & + 36000.0c_3 \\
\text{Día 4} \rightarrow & 169637.35 & = 20950c_1 & + 21000c_2 & + 35645.5c_3 \\
\text{Día 5} \rightarrow & 164625.45 & = 20750c_1 & + 20316c_2 & + 33878.5c_3
\end{array}
$$

En este ejemplo (de juguete) si utilizamos los datos de Santander, el sistema tiene solución. Si usamos los datos de Galicia el sistema no tiene solución.

2. Más ecuaciones que variables - La vida real

Cuando consideramos un sistema con más ecuaciones que variables, en general **NO** tiene solución.

Aunque teóricamente exista solución, en la práctica siempre aparecen errores numéricos y no podemos determinar si un sistema tiene solución (numéricamente es MUY difícil saber si un número es igual a 0 o no).

Solución: en vez de buscar una solución exacta del sistema de ecuaciones

$$Xc = y,$$

buscamos un vector c que minimice el error, es decir, que haga pequeñas las coordenadas del vector de errores

$$Xc - y.$$

El milagro de los mínimos cuadrados

Llegamos así al método de mínimos cuadrados. El vector c que minimiza la suma de los errores al cuadrado del sistema

$$Xc = y,$$

es solución del sistema lineal de ecuaciones

$$X^T X c = X^T y.$$

Es un sistema cuadrado y en general tiene solución única.

El problema DIFICIL de minimizar los errores se transforma en el problema FÁCIL de resolver un sistema lineal de ecuaciones. **Este es el milagro de los mínimos cuadrados.**

Resumen

Tenemos dos modelos posibles y queremos elegir el más apropiado:

$$A) total = c_1 \cdot YPF + c_2 \cdot Santander + c_3 \cdot Nvidia$$
$$B) total = c_1 \cdot YPF + c_2 \cdot Galicia + c_3 \cdot Nvidia$$

Seguimos los siguientes pasos:

1. Buscamos datos de la mayor cantidad posible de días.
2. Separamos el conjunto en dos: conjunto de entrenamiento ($80\,\%$ de los días) y conjunto de testeo ($20\,\%$ restante)
3. Calculamos c_1, c_2, c_3 para cada uno de los dos modelos utilizando mínimos cuadrados en el conjunto de entrenamiento.
4. Calculamos el error cuadrático medio de lás fórmulas resultantes aplicadas al conjunto de testeo.

Modelo lineal multivariado

El caso recién visto fue un ejemplo de juguete, donde existe una relación lineal entre las variables que tenemos que estimar.

En las aplicaciones reales, esa formula puede no existir (por ejemplo, si queremos estimar los gastos en tarjeta de crédito de una persona en función del sueldo y la cantidad de hijos), pero el modelo lineal puede darnos una buena estimación.

En el modelo lineal suponemos que existe una relación del tipo

$$Xc = y.$$

Como esa relación en general no existe, buscamos un vector c que haga que Xc se parezca lo más posible a y.

Ejercicio: modelos lineales

Dadas una variable a predecir y y variables explicativas $x_1, x_2, \ldots$, ¿cuáles de los siguientes modelos son lineales? ¿Cuál es la matriz X en cada caso?

1. $y = c_0 + c_1 x_1 + c_2 x_2$
2. $y = c_0 + c_1 x_1 + c_2 x_1^2$
3. $y = c_0 + c_1 x_1 + x_1^{c_2}$
4. $y = c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1 x_2$
5. $y = c_0 + c_1 \sin(x_1) + c_2 \sin(x_2)$
6. $y = c_0 + c_1 \sin(c_2 + x_1)$
7. $y = c_0 + c_1 e^{x_1}$
8. $y = c_0 \cdot c_1^{x_1}$

Ejercicio: modelos lineales

Dadas una variable a predecir y y variables explicativas $x_1, x_2, \ldots$, ¿cuáles de los siguientes modelos son lineales? ¿Cuál es la matriz X en cada caso?

1. $y = c_0 + c_1 x_1 + c_2 x_2$
2. $y = c_0 + c_1 x_1 + c_2 x_1^2$
3. $y = c_0 + c_1 x_1 + x_1^{c_2}$
4. $y = c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1 x_2$
5. $y = c_0 + c_1 \sin(x_1) + c_2 \sin(x_2)$
6. $y = c_0 + c_1 \sin(c_2 + x_1)$
7. $y = c_0 + c_1 e^{x_1}$
8. $y = c_0 \cdot c_1^{x_1}$

Algunos de estos modelos se pueden linearizar, pero eso ya es otra historia...

Ejemplo real: cálculo de calorías

Vamos a repetir estos pasos en un caso real: calcular las calorías de un alimento en función de sus componentes, utilizando las herramientas de Python para la construcción de modelos.

1. Construimos las matrices X e y utilizando `Formulaic`.
2. Separamos las matrices en entrenamiento y testeo utilizando `train_test_split`
3. Ajustamos el modelo utilizando `linear_model.fit()`.
4. Calculamos el error en los datos de testeo utilizando `linear_model.predict()`.