



BOURBAKI

COLEGIO DE MATEMÁTICAS

Índice

01. Introducción _____ pág. 03

- 01. ¿Por qué es difícil el forecast en presencia de outliers? _____ pág. 04
- 02. Nociones matemáticas _____ pág. 04
- 03. Preludio sobre la regla de tres _____ pág. 06

02. Instalación de Python y un entorno de trabajo para Machine Learning _____ pág. 09

- 01. Instalación de Miniconda [3] _____ pág. 09
- 02. Creación de un entorno virtual para análisis de datos _____ pág. 11
- 03. Jupyter Notebooks para ML en la Nube _____ pág. 14

03. Regresiones _____ pág. 15

- 01. Regresiones lineales: caso inyectivo y gaussiano _____ pág. 15

02. Cualidades estadísticas del caso inyectoro	pág. 18
03. Regresiones robustas y de Huber	pág. 19
04. Predecir el tiempo de llegada de los bomberos en París	pág. 21
05. Complementos de forecasting y outliers	
pág. 22	
01. Algoritmos de entrenamiento	_____ pág. 22
02. Hipótesis Gaussianas sobre el error y tests estadísticos	_____ pág. 23
03. Regresiones polinomiales	_____ pág. 24
04. Evaluación de las regresiones lineales	
pág. 25	
06. Referencias	_____ pág. 28

01 Introducción

Las siguientes notas son la bitácora de un curso de 8 horas que impartimos junto a Ana Isabel Ascencio. Además de este documento los invitamos a consultar el Github del curso [en este link](#).

El curso es una invitación al forecasting en presencia de outliers así como sus aplicaciones, nuestro ejemplo principal es la regresión de Huber sin embargo hablaremos de algunos otros, el curso está dividido en clases de la siguiente manera:

1. ¿Qué modelan las regresiones lineales y qué son los outliers? (una hora).
2. Un vistazo a las distribuciones de Laplace (una hora).
3. Descripción formal de las regresiones robustas y de Huber (dos horas).
4. Implementación para la predicción de tiempos de respuesta de los vehículos de emergencia en París (dos horas).
5. Dudas y complementos (dos horas).

¿Por qué es difícil el forecast en presencia de outliers?

Uno de los algoritmos más útiles para hacer una predicción es la regresión lineal, fue introducida por Legendre y Gauss a principios del siglo XIX. En las últimas décadas ha ganado un lugar importante gracias al uso de machine learning en diversas áreas. Es importante mencionar que a pesar de ser conocida hace tanto tiempo, recientemente ha ganado relevancia debido al poder computacional que existe para implementarla, incluso en presencia de numerosas dimensiones y enormes cantidades de datos. En este curso estudiaremos cómo es posible utilizar una ligera modificación de las regresiones lineales que más adecuadas en la presencia de outliers.

Nociones matemáticas

- Un subconjunto importante de los números reales es \mathbb{N} llamado el conjunto de los números naturales y consiste en los siguientes números $\mathbb{N} = \{1, 2, 3, \dots\}$.
- Si Ω es un conjunto y $A, B \subseteq \Omega$ son dos subconjuntos, denotaremos por:
 1. $A \cup B$ a la unión entre A y B .

- 2. $A \cap B$ a la intersección entre A y B .
- 3. A^c al complemento de A .
- Fijemos un conjunto finito Ω . Una distribución de probabilidad es una asignación numérica \mathbb{P} a cada elemento $A \subseteq \Omega$ tal que si A, B son dos subconjuntos:

 1. $0 \leq \mathbb{P}(A) \leq 1 = \mathbb{P}(\Omega)$
 2. $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$
 3. $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$, si $A \cap B = \emptyset$.

A la pareja (Ω, \mathbb{P}) se le llamará un espacio de probabilidad.

- (Distribución uniforme) supongamos que Ω es un conjunto de tamaño m . Definamos la siguiente ley de probabilidad $\mathbb{P}_{unif}(\omega_1, \dots, \omega_i) = \frac{i}{m}$. Para fijar ideas se puede pensar en este ejemplo cuando $m = 6$, esto corresponde a la probabilidad de obtener algún resultado cuando lanzamos un dado.
- (Distribución de Bernoulli) Supongamos que Ω es un conjunto de tamaño 2 i.e. $\Omega = \{\omega_1, \omega_2\}$. Sea $0 \leq p \leq 1$ un número arbitrario. Definimos $\mathbb{P}_{Bernoulli}(\omega_1) = 1 - p$ y $\mathbb{P}_{Bernoulli}(\omega_2) = p$. Para fijar notación supongamos que tenemos una moneda cargada a sol (digamos ω_1) tal que de cada 10 lanzamientos, 8 son sol, en ese caso $p = \frac{2}{10}$.
- La ley de probabilidad Gaussiana o normal con parámetros (μ, σ^2) se define para los intervalos $(-\infty, x]$ de la siguiente manera:

$$\mathbb{P}_{Gauss,\mu,\sigma^2}(-\infty, x] = \frac{1}{\sigma \cdot \sqrt{2\pi}} \int_{-\infty}^x \left(\exp \left(-\frac{(t-\mu)^2}{2 \cdot \sigma^2} \right) \right) dt$$

- (Distribución de Laplace) La distribución de Laplace con media μ y varianza $2b^2$ se define como

$$\mathbb{P}_{Laplace}(-\infty, x) = \frac{1}{2b} \int_{-\infty}^x e^{-\frac{|t-\mu|}{b}} dt$$

- Sea (Ω, \mathbb{P}) un espacio de probabilidad. Sean $A, B \subseteq \Omega$ son dos eventos aleatorios tales que $\mathbb{P}(B) > 0$. Definimos una nueva ley de probabilidad $\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$ lo cual se lee como la probabilidad del evento A una vez que haya ocurrido el evento B .

Preludio sobre la regla de tres

La regla de tres es el modelo matemático más simple para explicar un fenómeno, las regresiones lineales son una generalización de estos modelos. Hemos incluido esta sección con el objetivo de motivar el significado de los modelos lineales.

03.1 Proporción directa

El siguiente problema es fácil de resolver desde la escuela primaria:

Example 03.1. Si 2 kilos de fruta cuestan 70 pesos,

- ¿Cuánto cuestan 8 kilos de fruta?
- ¿Cuánto cuestan 700 gramos de fruta?

Demostración. Notemos que si uno de ellos crece el otro también lo hace y si uno decrece el otro también.

En ese caso es necesario utilizar una regla de tres directa, a saber:

- Digamos que compramos 8 kilos de fruta, entonces $y_8 = \frac{70}{2} \cdot 8$
- Si compramos 700 kilos de fruta entonces $y_{700} = \frac{70}{2} \cdot 700$

La cualidad fundamental de este modelo es que existe una relación lineal entre el precio de la fruta y la cantidad de kilos que se compran. □

Exercise 03.2. Plantear un problema concreto similar al anterior tal que el modelo correcto sea $y_x = \frac{A}{B} \cdot x + b$, es decir la ecuación clásica de una recta.

03.2 Proporción inversa

La regla de tres usual no es capaz de resolver todos los problemas similares al anterior por ejemplo:

Exercise 03.3. Si 8 trabajadores se tardan 15 horas en construir un muro,

- ¿Cuánto tardarán 5 trabajadores en construir un muro?

El problema es que a diferencia del caso de las frutas, en este caso si uno crece el otro decrece y viceversa.

Para resolver este problema es necesario utilizar la siguiente ecuación:

$$8 \cdot 15 = 5 \cdot y_5$$

Notemos que la ecuación $y_x \cdot x = A \cdot B$ no es la ecuación de una línea en la variable x , por eso no es posible utilizar una línea para modelar este fenómeno.

03.3 Conclusiones

Tal como lo hemos visto con los ejemplos anteriores, no es posible modelar cualquier fenómeno utilizando modelos lineales (reglas de tres inversas), inclusive en ese caso es posible utilizar técnicas muy similares a las que se usan en las reglas de tres para modelar algún fenómeno, a saber componiendo con la función de kernel apropiada (en este caso es $\phi(x) = x^{-1}$). En este curso estudiaremos algunos fenómenos tales que inclusive después de encontrar el kernel adecuado, no es posible utilizar el mismo algoritmo que se utilizaría para encontrar el modelo ideal.

02 Instalación de Python y un entorno de trabajo para Machine Learning

La forma más rápida y eficiente de instalar Python, manejar sus librerías y evitar problemas de dependencia(por actualizaciones y versiones) es mediante el uso de Miniconda que además de la instalación de Python, instala conda[1] un sistema gestor de paquetes y de entornos virtuales¹ [2].

Una forma alternativa es utilizar Anaconda, que al igual que Miniconda, es soportada por la misma compañía Anaconda. Ambas alternativas instalan la misma versión de Python y de conda. La razón por la que se recomienda Miniconda es porque permite instalar solamente lo que se requiere para este curso, con la opción de descargar otros paquetes cuando se requieran. Anaconda por el contrario instala por defecto, una cantidad excesiva de paquetes que raramente son utilizados, lo cual requiere demasiado espacio y tiempo.

Instalación de Miniconda [3]

¹Los entornos virtuales permiten controlar las versiones de software (en nuestro caso python y sus librerías) usado para análisis o aplicaciones

Descarga aquí el archivo correspondiente a tu sistema operativo, ya sea que sea que MacOS, Windows y Linux, eligiendo la última versión de Python.

Instalación para Windows

1. Da doble clic en el archivo descargado.
2. Sigue las instrucciones que se muestren aceptando las opciones que se proponen por defecto (si es necesario se pueden cambiar después).
3. Desde el menú de Inicio de Windows abre el programa Anaconda Prompt.

Instalación para MacOS

1. En la Terminal de tu Mac, navega hasta el directorio en donde descargaste el instalador y corre la siguiente línea: bash Miniconda3-latest-MacOSX-x86_64.sh
2. Sigue las instrucciones, aceptando las opciones por defecto (si es necesario se pueden cambiar después)
3. Cierra la Terminal y vuélvela a abrir, para que los cambios sean actualizados.

Instalación para Linux

1. En la Terminal de Linux, navega hasta el directorio en donde descargaste el instalador y corre la siguiente línea:

```
bash Miniconda3-latest-Linux-x86_64.sh
```

2. Sigue las instrucciones, aceptando las opciones por defecto (si es necesario se pueden cambiar después)
3. Cierra la Terminal y vuélvela a abrir, para que los cambios sean actualizados.

Para todos los sistemas operativos:

4. Para ver todos los paquetes que instalados por defecto en la distribución de Miniconda, corre la siguiente línea:

```
conda list
```

Creación de un entorno virtual para análisis de datos

Un entorno virtual es un ambiente de trabajo aislado, lo que permite instalar determinadas librerías o versiones de librerías sin que afecte al resto del sistema principal o de otros ambientes.

Por defecto, Miniconda crea un entorno llamado base que estará activo cuando abramos la terminal. Este entorno contiene todos los programas enlistados cuando se utilizó el comando conda list. Aunque podríamos usar este entorno, una mejor práctica es crear un entorno nuevo. Para ello es conveniente cambiar la configuración de conda para que no abra automáticamente el entorno base.

Desactivar el entorno base

- En Mac y Linux, se debe correr la linea que se muestra a continuación y enseguida cerrar y volver a abrir la terminal:

```
conda config --set auto_activate_base false
```

- Los usuarios de Windows deberán desactivar manualmente el entorno base utilizando:

```
conda deactivate
```

Creación de un entorno específico para análisis de datos mediante conda-forge

El entorno que vamos a crear lo titularemos MachineLearning, en éste instalaremos las librerías que necesitaremos para nuestros análisis.

1. Crea el ambiente de trabajo mediante la siguiente línea:

```
conda create -n MachineLearning
```

2. Actívalo con la siguiente instrucción:

```
conda activate MachineLearning
```

3. Agrega el canal ²

```
conda config --env --add channels conda-forge
```

Se recomienda utilizar el canal conda-forge, ya que es resultado de un esfuerzo colectivo con una gran variedad de librerías y paquetes que son cuidadosamente actualizados y donde se asegura tener versiones compatibles para macOS, Linux y Windows[4].

Puedes ver los canales instalados mediante la siguiente línea:

```
conda config --show channels
```

Cada canal tiene al menos una dirección URL asociada donde se localizan los repositorios de paquetes y librerías. Puedes ver estas direcciones utilizando:

```
conda info
```

4. Instala las librerías Pandas³, Scikit-learn⁴, Seaborn, Matplotlib⁵,

²Los repositorios desde donde podemos descargar las librerías de Python se llaman canales, Anconda, Inc provee por defecto el canal llamado default, sin embargo permite a cualquier usuario crear su propio canal si necesita otros paquetes que no están incluidos en el canal default, por esta razón existe una variedad de canales disponibles en la web desde donde se pueden descargar librerías de Python.

³Pandas es la principal librería de python utilizada Analisis de datos. Al instalar Pandas, por defecto se instala Numpy, sobre la cual funciona Pandas. Numpy es ampliamente usada en Ciencia de Datos porque permite el fácil manejo de matrices y sus operaciones.

⁴Scikit-learn es una importante librería para machine learning

⁵Matplotlib es las principales librerías de Python usadas para graficar y visualizar información

Plotly, Yellowbrick y Jupyter Notebooks⁶ copiando el siguiente comando (es importante copiarlo en una sola línea, separando mediante un espacio, cada librería a ser instalada):

```
conda install pandas matplotlib notebook jupyter_contrib_nbextensions  
scikit-learn yellowbrick plotly plotly=4.12.0
```

Jupyter Notebooks para ML en la Nube

Una alternativa altamente recomendable es utilizar el entorno que ofrece Google Colab ya que permite ejecutar código de Python utilizando notebooks de Jupyter a cualquier persona con una cuenta de Google, sin la necesidad de tener que instalar nada en la computadora del estudiante, y ejecutar el código directamente en los servidores alojados en la nube de Google. Google Colab es altamente compatible con el repositorio del curso que se encuentra en Github

⁶Jupyter notebook es una aplicación que nos ayudará a hacer nuestros análisis paso a paso, crear visualizaciones e incluir comentarios, como si se tratara de nuestro cuaderno de apuntes.

03 Regresiones

Este capítulo incluye los detalles técnicos y las definiciones principales tanto de las regresiones clásicas como de las regresiones robustas.

Regresiones lineales: caso inyectivo y gaussiano

Remark 01.1. Recordemos que una matriz real $X \in \mathbb{R}^{d \cdot N}$ induce una función $X : \mathbb{R}^d \rightarrow \mathbb{R}^N$, decimos que esta matriz es inyectiva cuando la función que induce es inyectiva i.e. si $X(v) = 0 \in \mathbb{R}^N$ entonces $v = 0 \in \mathbb{R}^d$. La propiedad principal que utilizaremos sobre estas matrices es que si X es inyectiva entonces $X^T \cdot X$ es invertible i.e. existe una matriz $(X^T \cdot X)^{-1}$ tal que $(X^T \cdot X)(X^T \cdot X)^{-1} = Id$.

Definition 01.1. Fijemos dos enteros positivos $d, N \in \mathbb{N}$. Definimos una base de datos lineal e inyectiva como un conjunto $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ tal que $(x, y) \in \mathbb{R}^d \times \mathbb{R}$, $x_{i,d} = 1$ para todo i y existe algún $\beta^* = (\beta_1^*, \beta_2^*, \dots, \beta_d^*)$ que satisface $y_i = \beta^* x_i + \epsilon_i$ donde ϵ_i son variables aleatorias que satisfacen:

- $\mathbb{E}(\epsilon) = 0$,
- Las variables aleatorias ϵ_i son independientes e identicamente distribuidas,
- Las variables aleatorias siguen una distribución gaussiana $\epsilon_i \sim N(0, \sigma)$

- La matriz de $N \times d$, $X = (x_i)_{i \leq N}$ es inyectiva.

Remark 01.2. Notemos que en la suposición del modelo lineal inyectivo y determinista la aleatoriedad solo depende de ϵ y por tanto ambas matrices X, Y están determinadas.

Con las suposiciones anteriores si denotamos por Y al vector $(y_i)_{i \leq N}$ y ϵ al vector $(\epsilon_i)_{i \leq N}$, una base de datos lineal e inyectiva S se puede escribir como $Y = Xm^* + \epsilon$.

Definition 01.2. Dada una base de datos lineal e inyectiva S y un vector $\beta \in \mathbb{R}^d$, definimos el error de β como la variable aleatoria

$$err_S(\beta) = \frac{1}{N} \cdot \sum_{i \leq N} (\langle x_i, \beta \rangle - y_i)^2$$

El estimador óptimo de mínimos cuadrados se define de la siguiente manera:

$$\beta_{LSE} = \underset{\beta \in \mathbb{R}^d}{\operatorname{argmin}} (err_S(\beta))$$

Theorem 01.3. Sea S una base de datos lineal e inyectiva, entonces el estimador óptimo de mínimos cuadrados existe y es único.

Demostración. Vamos a demostrar que $\beta_{LSE} = (X^T X)^{-1} X^T Y$ satisface la condición pero comenzamos explicando cómo encontrarlo:

Si consideramos a $err_S(\beta)$ como una función de β , sus mínimos (de existir) estarán en aquellos puntos donde el gradiente es cero.

Si calculamos el gradiente obtenemos lo siguiente:

$$\frac{2}{N} \sum_{i \leq N} (\langle x_i, \beta \rangle - y_i) x_i$$

Es fácil deducir de lo anterior que β_{LSE} debe de ser la solución a la igualdad con cero.

Para asegurarnos que es en verdad un mínimo es necesario calcular el Hesiano de X y verificar que es definido positivo.

□

01.1 Interpretación geométrica

Sea S una base de datos lineal e inyectiva como en la sección anterior.

Definition 01.3. Definimos $Im(X)$ la imagen de X como el conjunto de vectores $y \in \mathbb{R}^N$ tales que existe un $\beta \in \mathbb{R}^d$ con $X \cdot \beta = y$.

Remark 01.4. Al ser X inyectiva, necesariamente $N \geq d$, de hecho se puede demostrar que $\dim(Im(X)) = d$. Por definición del estimador óptimo, β_{LSE} es un vector tal que $\|Y - X \cdot \beta\|_2^2$ es mínimo, el cuál es por definición la proyección ortogonal de Y en $Im(X)$.

Exercise 01.5. Hacer el dibujo cuando $d = 1$.

Cualidades estadísticas del caso inyectivo

Una de las principales cualidades de las regresiones lineales es lo bien que se comportan respecto a hipótesis estadísticas tanto de los errores como de los datos, en esta sección enlistamos algunas de estas propiedades.

Proposition 02.1. *Si S es una base de datos como en la definición 01.1 entonces*

$$\mathbb{E}(\beta_{LSE}) = \beta^* \text{ y } \text{Var}(\beta_{LSE}) = \sigma^2 (X X^T)^{-1}.$$

Corollary 02.2. *Para todo $x \in \mathbb{R}^d$, si definimos la variable aleatoria $y_{LSE} := \beta_{LSE} \cdot x$, entonces:*

$$\mathbb{E}[y_{LSE}] = \beta^* \cdot x, \text{Var}(y_{LSE}) = \sigma^2 x^T (X X^T)^{-1} x$$

La proposición anterior solo es útil cuando conocemos la varianza de ϵ , sin embargo en general eso no es posible. Una manera de calcular esa varianza una vez hecha la predicción es la siguiente:

Definition 02.1. Fija una base de datos lineal e inyectiva definimos la varianza natural del estimador óptimo de mínimos cuadrados m^* de la siguiente forma:

$$\sigma_{nat}^2 = \frac{1}{N - d - 1} \cdot \sum_{i \leq N} (\langle x_i, \beta_{LSE} \rangle - y_i)^2$$

Proposition 02.3. $\mathbb{E}(\sigma_{nat}^2) = \sigma^2$.

Regresiones robustas y de Huber

En la sección anterior hemos estudiado la predicción de una regresión lineal, estas suponen en general que los errores siguen una distribución gaussiana. Las regresiones robustas logran que los outliers tengan menos influencia en la solución de una regresión lineal, eso se logra mediante el cambio de la función de error:

Definition 03.1. (Regresión robusta) Bajo las hipótesis de la definición 01.1, definimos el error de acuerdo a la regresión robusta de la siguiente manera:

$$err_{S,Rob}(\beta) = \frac{1}{N} \left(\sum_{i \leq N} |y_i - x_i \beta_i| \right)$$

Remark 03.1. *Otra manera de entender a la regresión robusta es cambiar la suposición de normalidad por la siguiente suposición:*

$$\epsilon_i \sim Lap(0, \sigma^2)$$

Desafortunadamente la función err_{Rob} no es una función fácil de optimizar y por tanto el entrenamiento vía los métodos clásicos del gradiente no funcionan, por tanto se debe de considerar una solución distinta a la del método del gradiente que se utiliza en regresiones lineales clásicas.

Otra manera de solucionar el problema de la presencia de outliers al hacer una predicción es considerando la función de error Huber:

Definition 03.2. (Regresiones de Huber) Bajo las hipótesis de la definición

01.1, definimos al error de Huber para (x_i, y_i) , con parámetro δ de la siguiente manera, si $|y_i - \beta x_i| \leq \delta$ entonces:

$$err_{i,Hub}(\beta) = (y_i - \beta \cdot x_i)^2$$

y en el caso contrario:

$$err_{i,Hub}(\beta) = \left(\delta |y_i - \beta \cdot x_i| - \frac{\delta^2}{2} \right)$$

Definimos a $err_{S,Hub}(\beta)$ como la suma de los errores $err_{i,Hub}(\beta)$ sobre $i \leq N$.

El estimador óptimo de mínimos cuadrados se define de la siguiente manera:

$$\beta_{Huber} = \underset{\beta \in \mathbb{R}^d}{\operatorname{argmin}} (err_{S,Hub}(\beta))$$

Remark 03.2. *Afortunadamente esta función sí es diferenciable y es posible utilizar el método del gradiente para su entrenamiento.*

04 Predecir el tiempo de llegada de los bomberos en París

A continuación incluimos una breve descripción del data set que utilizaremos para ejemplificar la implementación de las regresiones lineales y las regresiones polinomiales. Los datos fueron obtenidos de un Challenge en el que se pretende predecir el tiempo de respuesta y el tiempo de llegada de los bomberos de acuerdo a distintas características del incidente como por ejemplo la distancia, el tipo de vehículo utilizado, la altura del incidente (en caso de ser un edificio por ejemplo), etc. Pueden consultar los detalles en el [siguiente link](#). La hipótesis sobre estos datos que nosotros planteamos es la siguiente: predecir el tiempo de respuesta a un incidente puede ser predicho utilizando una regresión (quizás polinomial sin embargo con fines pedagógicos nos concentraremos en el sencillo caso de las regresiones lineales). Esta predicción puede verse afectada gravemente por la presencia de outliers en nuestros datos, por ejemplo aquellos causados por huelgas, atentados o en general accidentes que dificulten la movilidad de los vehículos de emergencia. Por ello la segunda parte de nuestra hipótesis es utilizar regresiones robustas para hacer estas predicciones de manera más acertada.

El conjunto de datos utilizados está disponible [siguiente link](#)

05 Complementos de forecasting y outliers

Esta sección incluye un compendio de temas que buscan complementar el conocimiento de los estudiantes sobre el forecasting mediante regresiones y el estudio de los outliers.

Algoritmos de entrenamiento

Vale la pena notar que en este curso no hemos hablado sobre ninguno de los algoritmos de entrenamiento que predicen ni β_{LSE} , β_{Rob} ni β_{Huber} . Dichos detalles exceden los objetivos de este curso pues requieren un nivel de sofisticación matemática un poco más elevada. De cualquier manera hemos decidido incluir algunas palabras con el objetivo de facilitar la inmersión del estudiante a este tipo de técnicas. Al algoritmo principalmente utilizado para el proceso de entrenamiento de las regresiones es el método del gradiente, por su eficacia en realidad es comúnmente utilizado el método del gradiente estocástico el cuál es una aleatorización del anterior. Estos métodos consisten en encontrar el mínimo de las ecuaciones que definen a β_{LSE} , β_{Huber} mediante un proceso diferencial iterativo basado en el siguiente principio: los puntos máximos y mínimos se encuentran donde el gradiente se anula. Es distinto el caso de la regresión

robusta la cuál requiere un método distinto ya que la función $err_{Rob}(\beta)$ no es diferenciable al incluir el valor absoluto, en ese caso se recomienda una solución vía la programación lineal.

Hipótesis Gaussianas sobre el error y tests estadísticos

Hasta ahora no hemos utilizado la hipótesis sobre la distribución de nuestros errores ϵ_i , en esta breve sección hablaremos de algunas consecuencias:

Theorem 02.1. *Supongamos que $\epsilon_i \sim N(0, \sigma^2)$, entonces:*

- $\beta_{LSE} \sim N\left(\beta^*, \sigma^2 (XX^T)^{-1}\right)$
- $(N - d - 1) \sigma_{nat}^2 = \sigma^2 \chi_{N-p-1}^2$
- σ_{nat}^2 y β_{LSS} son variables aleatorias independientes.
- Para todo $x \in \mathbb{R}^d$, si definimos la variable aleatoria $y := \beta_{LSE} \cdot x$, entonces:

$$\mathbb{P}_y = N\left(\beta^* x, \sigma_{nat}^2 x (XX^T)^{-1} x\right)$$

Gracias a los resultados anteriores es posible formular algunos tests estadísticos útiles en la práctica:

Corollary 02.2. *(Intervalos de confianza) Sea t_{N-d} la ley t de probabilidad del*

estudiante con $N-d$ grados de libertad, para cualquier $\epsilon > 0$ y cualquier $x \in \mathbb{R}^d$

$$\mathbb{P}((\beta^* \cdot x) \in [g, g']) = 1 - \epsilon$$

donde

$$g = \beta_{LSE} \cdot x - \sqrt{\sigma_{nat}^2 x (X X^T)^{-1} x} - q_{1-\frac{\epsilon}{2}}(t_{N-d})$$

$$g' = \beta_{LSE} \cdot x + \sqrt{\sigma_{nat}^2 x (X X^T)^{-1} x} - q_{1-\frac{\epsilon}{2}}(t_{N-d})$$

Es decir, $[g, g']$ es un intervalo de confianza ϵ .

Regresiones polinomiales

Hasta ahora hemos hablado de las regresiones lineales, es decir algoritmos que arrojan modelos lineales, sin embargo las regresiones en general son algoritmos más generales buscan modelar la siguiente función:

$$g(x) = \mathbb{E}[Y|X=x]$$

Donde $x \in \mathbb{R}^d$, Y es una variable aleatoria y X es un vector aleatorio en \mathbb{R}^d .

Las regresiones polinomiales por ejemplo buscan algún polinomio de grado superior a uno g para hacer una predicción (o quizás otro tipo de funciones más complicadas como en la sección 03.2), afortunadamente este no es un

problema radicalmente distinto por lo siguiente:

Sea $p(x) = a_0 + a_1x + \dots + a_nx^n$ un polinomio en una sola variable, notemos que si $\phi(x) = (1, x, x^2, \dots, x^n)$ entonces $p(x) = \langle \phi(x), (a_0, a_1, \dots, a_n) \rangle$.

Definition 03.1. Sea $S = \{(x_1, y_1), \dots, (x_N, y_N)\} \subseteq \mathbb{R}^2$, definimos el resultado de la regresión polinomial como m^* , el resultado de minimizar

$$\frac{1}{N} \sum_{i \leq N} (y_i - \langle \phi(x_i), (a_0, \dots, a_n) \rangle)^2$$

para (a_0, \dots, a_n) .

Es posible considerar regresiones polinomiales en más de una variable, de la misma forma podríamos considerar vectores con entradas no lineales distintos a los anteriores, algunos ejemplos usuales de estos vectores incluyen funciones trigonométricas o tanto logaritmos y exponenciales.

Evaluación de las regresiones lineales

Cuando utilizamos una regresión lineal es importante analizar si el modelo está funcionando correctamente. Existen diversas maneras de realizar este análisis y en esta sección nos concentraremos en el coeficiente de determinación también conocido como R^2 .

Definition 04.1. Supongamos que $S = \{(x_i, y_i)\}_{i \leq N}$ es una base de datos como en la definición 01.1, β_{LSE} el resultado de una regresión lineal en los datos

de S e $\bar{y} = \frac{1}{N} \sum_{i \leq N} y_i$ es el promedio empírico de las variables dependientes. Definimos el coeficiente de determinación (relativo a la variable S y β_{LSE}) de la siguiente manera:

$$R_S^2 = \frac{\sum_{i \leq N} (\beta_{LSE} \cdot x_i - \bar{y})^2}{\sum_{i \leq N} (y_i - \bar{y})^2}$$

Lemma 04.1. *Bajo las hipótesis de la definición anterior, la cantidad R_S^2 está acotada entre 0 y 1.*

Demostración. Re-escribamos la diferencia entre nuestros datos observados y_i y la media \bar{y} de la siguiente manera:

$$y_i - \bar{y} = (x_i \cdot \beta_{LSE} - \bar{y}) + (y_i - x_i \cdot \beta_{LSE})$$

Ahora elevemos al cuadrado y sumemos sobre todos los N ejemplos:

$$\sum_{i \leq N} (y_i - \bar{y})^2 = \sum_{i \leq N} (x_i \cdot \beta_{LSE} - \bar{y})^2 + \sum_{i \leq N} (y_i - x_i \cdot \beta_{LSE})^2$$

Al dividir sobre la cantidad de la izquierda obtenemos lo siguiente:

$$1 = R^2 + \frac{\sum_{i \leq N} (y_i - x_i \cdot \beta_{LSE})^2}{\sum_{i \leq N} (y_i - \bar{y})^2}$$

Debido a que las cantidades del lado derecho de la ecuación son todas no negativas podemos concluir la demostración.

□

Antes de hacer algunas observaciones sobre la interpretación correcta de R^2 vamos a definir el sobre-ajuste:

Definition 04.2. Sea S una base de datos como en 01.1, diremos que un modelo M_β sobre-ajusta cuando el modelo obedece excesivamente a ϵ_i .

Remark 04.2. La manera como debemos interpretar el resultado anterior es de la siguiente manera:

- *R^2 es la razón entre la varianza de la predicción que hace la regresión lineal y la varianza de nuestras observaciones hechas en S .*
- *Un valor cercano a cero de R^2 implica que el modelo no está funcionando correctamente pues una predicción constante \bar{y} es muy parecida.*
- *Un valor cercano a uno de R^2 significa que la varianza real de nuestros datos observados es muy parecida a la varianza de las predicciones que está haciendo el modelo, por tanto se podría interpretar como un buen ajuste del modelo a los datos.*
- *Notemos que un R^2 no siempre es una buena noticia, algunas veces podríamos incurrir en sobre-ajuste y tener un alto valor de R^2 .*

06 Referencias

- [1] «Conda — conda 4.8.3.post5+125413ca documentation». <https://docs.conda.io/pr> (accedido mar. 26, 2020).
- [2] Anaconda is Bloated - Set up a Lean, Robust Data Science Environment with Miniconda and Conda-Forge.
- [3] «Miniconda — Conda documentation». <https://docs.conda.io/en/latest/miniconda> (accedido mar. 26, 2020).
- [4] «A brief introduction — conda-forge 2019.01 documentation». <https://conda-forge.org/docs/user/introduction.html> (accedido mar. 26, 2020).



escuela-bourbaki.com