Regresiones logísticas

Escuela de Matemáticas Bourbaki

Junio 2020

Contents

1 Introducción

Las siguientes notas son la bitácora de un curso de 9 horas que impartimos en una alianza con Data Science Institute. Además de este documento los invitamos a consultar el Github del curso en este link.

El curso es una invitación al uso de los árboles de decisión para problemas de clasificación binaria, el curso está dividido en clases de la siguiente manera:

1. ¿Qué es machine learning y las regresiones logísticas? (una hora)
2. Un vistazo a R (una hora).
3. Descripción formal de las regresiones logísticas (dos horas).
4. Implementación de las regresiones logísticas para (dos horas).
5. Aspectos avanzados de las regresiones logísticas (tres horas).

1.1 ¿Qué es machine learning y las regresiones logísticas?

Machine learning es un conjunto de técnicas que nos permiten hacer predicciones utilizando información del pasado. Sin embargo algunas veces hacer predicciones correctas no es suficiente pues nos gustaría tener información de por qué cierta predicción se ha hecho para enriquecer nuestro entendimiento del problema, en machine learning aquellas técnicas que nos proporcionen dicha información se dice que son algoritmos transparentes.

En este curso hablaremos con detalle de las regresiones logísticas y sus aplicaciones a los problemas de clasificación. Estudiaremos este tipo de algoritmos desde un punto de vista discriminativo y generativo. La gran ventaja de este tipo de algoritmos frente a otros utilizados en machine learning es su enorme capacidad de detalles y tests estadísticos en sus soluciones. Analizaremos también el problema del sobre-ajuste para este algoritmo.

1.2 Nociones matemáticas y elementos de probabilidad

- En estas notas denotaremos por $\mathbb{R}$ al conjunto de los números reales es decir todos los números negativos o positivos que conocoemos (incluyenndo algunos como π etc.).
- Un subconjunto importante de los números reales es $\mathbb{N}$ llamada el conjunto de los números naturales y consiste en los siguientes números $\mathbb{N} = \{1, 2, 3, \ldots\}$.
- Si $d \in \mathbb{N}$ es un número natural entonces denotaremos por $\mathbb{R}^d$ al conjunto de vectores $(x_1, x_2, \ldots, x_d)$ de tamaño d con entradas en $\mathbb{R}$. Por ejemplo $\mathbb{R}^2$ es el plano cartesiano.
- Si $x = (x_1, \ldots, x_d), y = (y_1, \ldots, y_d) \in \mathbb{R}^d$ entonces el producto punto de x, y se denota por $\langle x, y \rangle$ y es el número real
$$\langle x, y \rangle = x_1 y_1 + \ldots x_d y_d$$
- El logaritmo es la función $log : (0, \infty) \to \mathbb{R}$ definida por $log(x) = \int_1^x \frac{1}{t} dt$
- La función exponencial $exp : \mathbb{R} \to \mathbb{R}$ es la función inversa del logaritmo i.e. $exp(log(x)) = x$ y $log(exp(x)) = x$. Además satisface $exp(x) = e^x$ donde $e = 2.7182\ldots$
- Para cualesquiera α, γ donde esté bien definida la función logaritmo, se tienen las siguientes igualdades:
  1. $log(\alpha^\gamma) = \gamma log(\alpha)$
  2. $log(\alpha\gamma) = log(\alpha) + log(\gamma)$

1.3 Elementos de probabilidad

Definition 1.1. 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.

Definition 1.2. 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 $(\Omega, \mathbb{P})$ se le llamará un espacio de probabilidad.

Example 1.1. *(Distribución uniforme) supongamos que Ω es un conjunto de tamano m. Definamos la siguiente ley de probabilidad $\mathbb{P}_{unif}(\omega_1, \ldots, \omega_i) = \frac{i}{m}$. 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.*

Example 1.2. *(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 $p = \frac{2}{10}$.*

Example 1.3. *(Distribusión gaussiana o normal) Definimos la ley de probabilidad con parámetros $\left(\mu, \sigma^2\right)$ se define para los intervalos $(-\infty, x]$ de la siguiente manera:*

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

Denotaremos esta ley de probabilidad por $\mathbf{N}\left(\mu, \sigma^2\right)$.

Definition 1.3. Sea $(\Omega, \mathbb{P})$ un espacio de probabilidad. Sean $A, B \subseteq \Sigma$ son dos eventos aleatorios, decimos que A y B son aleatorios cuando

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$$

Definition 1.4. Sea $(\Omega, \mathbb{P})$ un espacio de probabilidad. Sean $A, B \subseteq \Sigma$ son dos eventos aleatorios tales que $\mathbb{P}(B) > 0$. Definimos una nueva ley de probabilidad $\mathbb{P}(A|B) = \mathbb{P}\frac{(A \cap B)}{\mathbb{P}(B)}$ lo cual se lee como la probabilidad del evento A una vez que haya ocurrido el evento B.

Exercise 1.4. *Si dos eventos aleatorios A, B son independientes entonces $\mathbb{P}(A|B) = \mathbb{P}(A)$.*

Definition 1.5. Sea $(\Omega, \mathbb{P})$ una ley de probabilidad. Sea $\mathbb{R}$ el conjunto de todos los números. Una variable aleatoria (real) es una función $X : \Omega \to \mathbb{R}$.

Exercise 1.5. *Si Ω es el espacio de la experiencia aleatoria de lanzar dos dados justos, definimos la siguiente variable aleatoria: $X(i, j) = i + j$. Demostrar que $\mathbb{P}_X(2) = \mathbb{P}((1,1))$, $\mathbb{P}_X(3) = \mathbb{P}(\{(1,2),(2,1)\})$ y $\mathbb{P}_X(4) = \mathbb{P}(\{(1,3),(3,1),(2,2)\})$.*

Definition 1.6.

- Si $X : \Omega \to \mathbb{R}$ es una variable aleatoria y $p_i = \mathbb{P}(\omega_i)$, definimos la esperanza de X de la siguiente forma:

$$\mathbb{E}[X] = \sum_{\sigma_i \in \Sigma} p_i X(\sigma_i)$$

- Definimos la varianza de X como

$$Var(X) = \sum_{x \in Im(X)} (x - \mathbb{E}(X))^2 \cdot p_x$$

- Definimos la desviación estándar de una variable aleatoria como la raíz positiva de $Var(X)$, se denotará por ρ_X.

- Definimos la covarianza entre X y Y como

$$Cov(X, Y) = \mathbb{E}(X \cdot Y) - \mathbb{E}(X) \cdot \mathbb{E}(Y)$$

- El coeficiente de correlación entre X, Y se define como

$$Corr\left(X,Y\right)=\frac{Cov\left(X,Y\right)}{\sqrt{\rho_X\rho_Y}}$$

- Si $X, Z:\Omega\to\mathbb{R}$ dos variables aleatorias, definimos la variable aleatoria $W=(X,Z)$ como la función $W\left(x,z\right)=\left(X\left(x\right),Z\left(z\right)\right)$.
- Sean $X, Z:\Omega\to\mathbb{R}$ dos variables aleatorias, decimos que ellas son independientes si para todo $A\subseteq Im\left(X\right), B\subseteq Im\left(Z\right)$, $\mathbb{P}_W\left(A\times B\right)=\mathbb{P}_X\left(A\right)\cdot\mathbb{P}_Z\left(B\right)$.

Proposition 1.6. *Sean X, Z dos variables aleatorias, la correlación entre ellas satisface lo siguiente:*

- *Si X, Z son independientes entonces $Corr\left(X,Z\right)=0$*
- $-1\leq Corr\left(X,Z\right)\leq 1$.

2 Instalación de R, R Studio y Paquetes requeridos para Regresión Ridge

Para este curso utilizaremos el lenguaje de programación R a través del entorno R Studio, así como un conjunto de paquetes que nos permitirán realizar el manejo, preprocesamiento y modelado de los datos.

2.1 Instalación de R y Rstudio

2.1.1 para Windows

1. Descargar R desde el repositorio CRAN[1], haciendo click aquí. Cuando termine la descarga ejecutar el archivo R-4.0.0-win.exe, aceptar todas las opciones por defecto hasta que se complete la instalación.
2. Descargar el archivo .exe desde Rstudio dando click aquí. Igualmente cuando termine la descarga, ejecutar el archivo .exe y aceptar las opciones por defecto hasta completar la instalación

2.1.2 Para Mac

1. Descargar R desde el repositorio CRAN, haciendo click aquí. Cuando termine la descarga ejecutar el archivo R-4.0.0.pkg, aceptar todas las opciones por defecto hasta que se complete la instalación.
2. Descargar el archivo .dmg desde Rstudio, dando click aquí. Igualmente cuando termine la descarga, ejecutar el archivo .dmg y aceptar las opciones por defecto hasta completar la instalación

2.1.3 Para Ubuntu

1. Previo a la instalación, en la terminal, actualiza los paquetes instalados mediante: sudo apt update sudo apt -y update
2. Enseguida instala R mediante sudo apt -y install r-base
3. Descargar el archivo .deb desde Rstudio , correspondiente a la versión de Ubuntu.

[1] Administrado por R fundation. CRAN es el más grande repositorio de paquetes de R, sus siglas provienen del inglés: the comprehensive R archive network.

4. Cuando termine la descarga, desde la terminal ubicate en la carpeta de tus descargas y corre la siguiente línea: sudo dpkg -i rstudio-1.2.5033-amd64.deb

5. Si encuentras problemas de dependencias corre la siguiente linea y después vuelve al paso anterior sudo apt -f install

2.1.4 Para otras versiones de Linux

1. Descargar R desde el repositorio CRAN, siguiendo las instrucciones de acuerdo a la distribución que de Lunux que se tenga.

2. Descargar el archivo correspondientes según la distribución de Linux desde Rstudio

2.2 Instalación de Paquetes de R

Para en análisis que nos ocupa instaremos tidiverse [2], here[3], NPL y tm[4], RcolorBrewer y wordcloud[5], lattice y carret[6], y glmn[7]. En todos los sistemas operativos, abrir Rstudio y ejecutar en la consola la siguiente línea:

install.packages(c("tidiverse","here","NLP","tm","RcolorBrewer","wordcloud","lattice","caret" ,"glmnet"))

3 Regresión logística

3.1 Máxima verosimilitud

Un acercamiento probabilista a machine learning es vía la máxima verosimilutud, en esta sección ejemplificaremos este punto de vista con un ejemplo.

Supongamos que una compañía quiere modelar la satisfacción de un usuario utilizando la siguiente asignación, si el usuario está satisfecho se le asignará el valor uno, de otro modo se le asignará el valor cero.

Notemos que es posible modelar la satisfacción promedio de todos los usuarios mediante un solo valor $\beta^* \in [0,1]$ (igual a la esperanza de cierta variable aleatoria), sin embargo algunas veces podría ser complicado conocer la información de todos los usuarios y por tanto se considerará solo una muestra de tamaño N, digamos $S = \{x_1, \ldots, x_N\}$ de tal forma que esta muestra sea independiente e idénticamente distribuida. En ese caso es posible definir la estimación empírica $\frac{1}{N}\sum_{i\leq N} x_i$. Gracias a la ley de los grandes números $\mathbb{E}\left[\frac{1}{N}\sum_{i\leq N} x_i\right] = \beta^*$.

En la última línea no es claro por qué es necesario calcular la esperanza de la estimación $\frac{1}{N}\sum_{i\leq N} x_i$. Esta última cantidad es enrealidad una variable aleatoria. Ahora vamos a explicar qué significa esta cantidad en términos de verosimilitud. Notemos que si queremos calcular la probabilidad de obtener el muestreo S tenemos la siguiente fórmula:

$$\mathbb{P}(S) = \prod_{i\leq N} (\beta^*)^{x_i} (1-\beta^*)^{1-x_i} = (\beta^*)^{\sum_{i\leq N} x_i} (1-\beta^*)^{\sum_{i\leq N}(1-x_i)}$$

Eso implica que $log(\mathbb{P}(S)) = \left(\sum_{i\leq N} x_i\right) log(\beta^*) + \left(\sum_{i\leq N}(1-x_i)\right) log(1-\beta^*)$

[2] Colección de 8 paquetes para el manejo y visualización de datos: ggplot2, dplyr, tidyr, readr, purrr, tibble, string, forcast.

[3] Here, permite referenciar fácilmente archivos

[4] Permiten el procesamiento de Text

[5] Para visualizar nubes de palabras

[6] Con la finalidad de dividir los datos en conjunto de prueba y entrenamiento

[7] Para entrenear modelos lineales generalizados (GLM) con penalización

Como *log* es una función creciente, definimos el maximizador de la verosimilitud de la siguiente manera:

Definition 3.1.

- Dado un parámetro $\beta \in [0,1]$, definimos la cantidad $L(S,\beta) = \left(\sum_{i\leq N} x_i\right) log(\beta) + \left(\sum_{i\leq N}(1-x_i)\right) log(1-\beta)$ como la verosimilitud logarítmica del parámetro β dada la base de datos S.
- Dada una base de datos S, definimos el parámetro que maximiza la verosimilitud de la siguiente manera:

$$\beta_{MLE} = \max_{\beta\in[0,1]} L(S,\beta)$$

Proposition 3.1. *Bajo las hipótesis de esta sección se tiene que* $\beta_{MLE} = \frac{1}{N}\sum_{i\leq N} x_i$

3.2 Clasificación binaria y regresión logística

A lo largo del curso supondremos que estamos en un problema clásico de aprendizaje supervisado tipo clasificación binaria en $\mathbb{R}^d$, donde buscamos entrenar a nuestro modelo con un conjunto de ejemplos

$$S = \{(x_1, z_1), \ldots, (x_N, z_N)\}$$

cuando $x_i \in \mathbb{R}^d, z_i \in \{-1, +1\}$.

En el curso relacionado con el perceptrón hicimos una hipótesis de separabilidad lineal. En algunos casos no es posible suponer que existe un hiperplano que clasifica correctamente todos los puntos en la base de datos, en ese caso será necesario introducir un error de clasificación que corresponde a una variable aleatoria. La regresión logística es un modelo que permite lidiar satisfactoriamente con este tipo de situaciones.

En general se define la función $Hinge : \mathbb{R} \to \mathbb{R}$ de la siguiente manera: $Hinge(x) = max\{0, x\}$.

Además, para cada $\beta^* \in \mathbb{R}^d$, definimos la función $Hinge_{\beta^*} : \mathbb{R}^d \to \{0,1\}$ tal que $x \mapsto 1$ si $\langle x, \beta^*\rangle > 0$ y $x \mapsto 0$ si $\langle x, \beta^*\rangle < 0$. Utilizando la notación de la sección **??** notemos que como funciones,

$$Hinge_{\beta^*} = Hinge(sign_{\beta^*})$$

Además será necesario hacer la siguiente suposición estadística:

- Supondremos que nuestros ejemplos (x_i, z_i) son muestreos de N variables aleatorias identicamente distribuidas e independientes sobre $\mathbb{R}^d \times \{-1. + 1\}$.

Para hacer compatible nuestra base de datos $S = \{(x_i, z_i)\}_{i\leq N}$ con las funciones $Hinge$ definidas anteriormente es necesario cambiar los $z_i = -1$ por 0's. Así que a partide ahora re-definiremos la base de datos $S = \{(x_i, Hinge(z_i))\}_{i\leq N}$ y cuando escribimos z_i en realidad queremos decir $Hinge(z_i)$. Definimos el error logístico de la siguiente manera:

Definition 3.2. Sea $\beta \in \mathbb{R}^d$ $err_{Log,\beta} : \mathbb{R}^d \times \{0,1\} \to \mathbb{R}$
$err_{Log,\beta}(x,z) = z \cdot log\left(1 + e^{-x\cdot\beta}\right) + (1-z)\, log\left(1 + e^{x\cdot\beta}\right)$

Estamos interesados en minimizar $err_{Log,\beta}(x_i, z_i)$ para todo $i \leq N$.

Definition 3.3. Definimos el estimador logístico de la siguiente manera:

$$\beta_{Log} = \min_{\beta\in\mathbb{R}^d} \frac{1}{N}\sum_{i\leq N}(err_{Log}(\beta\cdot x_i, z_i))$$

3.3 Interpretación geométrica y probabilista

La función de pérdida logística admite la siguiente interpretación probabilista, sean X, Z dos variables aleatorias, X con imagen todos los reales, y Z con varlores $\{0, 1\}$.

Por la fórmula de Bayes:

$$\mathbb{P}(Z = 1|X) = \frac{\mathbb{P}(X|Z = 1) \cdot \mathbb{P}(Z = 1)}{\mathbb{P}(X|Z = 1)\mathbb{P}(Z = 1) + \mathbb{P}(X|Z = 0)\mathbb{P}(Z = 0)}$$

Entonces

$$\mathbb{P}(Z = 1|X) = \frac{1}{1 + \frac{\mathbb{P}(X|Z=0)\mathbb{P}(Z=0)}{\mathbb{P}(X|Z=1)\mathbb{P}(Z=1)}}$$

Eso implica que si $\sigma(x) = \frac{1}{1+e^{-x}}$ y además definimos la función f de la variable aleatoria X, $f(X) = -log\left(\frac{\mathbb{P}(X|Z=0)\mathbb{P}(Z=0)}{\mathbb{P}(X|Z=1)\mathbb{P}(Z=1)}\right) = log\left(\frac{\mathbb{P}(X|Z=1)\mathbb{P}(Z=1)}{\mathbb{P}(X|Z=0)\mathbb{P}(Z=0)}\right) = -log\left(\frac{\mathbb{P}(X|Z=0)}{\mathbb{P}(X|Z=1)}\right) - log\left(\frac{\mathbb{P}(Z=0)}{\mathbb{P}(Z=1)}\right)$, entonces

$$\mathbb{P}(Z = 1|X) = \sigma(f(X))$$

Definition 3.4. Diremos que una base de datos S satisface separabilidad lineal y logística cando sus elementos sean un muestreo i.i.d. y además exista algún $\beta^* \in \mathbb{R}^d$ tal que

$$log\left(\frac{\mathbb{P}(X|Z = 1)\mathbb{P}(Z = 1)}{\mathbb{P}(X|Z = 0)\mathbb{P}(Z = 0)}\right) = \beta^* \cdot X$$

Equivalentemente $\mathbb{P}(Z = 1|X) = \sigma(\beta^* \cdot X)$

Remark 3.2. *Regresando a la notación de un elemento en la base de datos y no de una variable aleatoria, observemos que*

- *Cuando $|\beta \cdot x_i|$ es suficientemente grande entonces $\sigma(x_i \cdot \beta) = \frac{1}{1+e^{-x_i \cdot \beta}}$ y por tanto $\mathbb{P}(Z = 1|X)$ es cercano a 1.*
- *Por otro lado cuando $|\beta \cdot x_i|$ es muy pequeño, $\sigma(x_i \cdot \beta) = \frac{1}{1+e^{-x_i \cdot \beta}}$ y por tanto $\mathbb{P}(Z = 1|X)$ es cercano a $\frac{1}{2}$.*

Proposition 3.3. *Si suponemos que $S = \{(x_1, z_1), \ldots, (x_N, z_N)\}$ satisface satisface separabilidad lineal y logística. Entonces β_{Log} también maximiza a $\mathbb{P}(Z = 1|X)$ cuando se supone que esta ley de probabilidad es gaussiana de la forma $\sigma(\beta \cdot X)^Z (1 - \sigma(\beta \cdot X))^{1-Z}$.*

4 Classificador de reseñas favorables

Se utiliza regresión logistica con penalización ridge para realizar una clasificación binaria utilizando un corpus de 1600 comentarios a hoteles de Chicago. Este corpus consiste en críticas falsas y autenticas de 20 hoteles de Chicago. Los datos fueron originalmente descritos en dos documentos de acuerdo con el sentimiento de la revisión:

[1] M. Ott, Y. Choi, C. Cardie, and J.T. Hancock. 2011. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.

[2] M. Ott, C. Cardie, and J.T. Hancock. 2013. Negative Deceptive Opinion Spam. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Este corpus contiene:

800 comentarios auténticos, de los cuáles 400 están clasificados como positivos y 400 como negativos 800 comentarios falsos, igualmente dividio en 400 comentarios positvas y 400 negativos.

La vectorización de los textos se realiza utilizando la técnica Bag of Words (BoW) o Bolsa de Palabras de manera que se obtiene una matriz de 1600 documentos por 3,528 palabras.

El conjunto de datos original puede ser descargado aquí. Este mismo ha sido también publicado por Kaggle en un archivo csv, que ha sido copiado en el presente repositorio.

5 Temas selectos en regresiones lineales

Existen diversos métodos para mejorar la capacidad estadística o computacional de las regresiones logísticas, en esta sección estudiaremos algunas de estas ideas.

5.1 Método del gradiente

Desafortunadamente incluso cuando X es una matriz inyectiva no es posible resolver de manera exacta las ecuaciones que desprenden de igualar el gradiente a cero, será necesario utilizar el método del descenso del gradiente o incluso el método de Newton.

5.2 Regularización tipo ridge

Una de las principales razones por las que un algoritmo de machine learning podría no funcionar es el llamado sobre-ajuste.

Definition 5.1. Si hemos fijado un tiempo t y una base de datos S_t, supongamos que utilizaremos un algoritmo A para predecir un modelo M, se dice que el el modelo M ha incurrido en sobre-ajuste relativo a la información anterior cuando M incluye sistemáticamente el ruido estocástico (aleatorio) de la base S_t.

En el caso de las regresiones logísticas el sobre-ajuste podría ocurrir cuando las variables aleatorias que generan a nuestros ejemplos de S están correlacionadas lo cuál contradice la hipótesis de ser ejemplos independientes e idénticamente distribuidos. Un caso particular es cuando la matriz $X = (x_{i,j})_{i\leq N, j\leq d}$ no es inyectiva (i.e. existe un vector $0 \neq \alpha \in \mathbb{R}^d$ tales que $X\alpha = 0$, un caso particular es cuando $d > N$). Eso significa que $X\beta = X\beta + X\alpha = X(\beta + \alpha)$ con $\alpha + \beta \neq \beta$ lo cuál significa que no existe un único parámetro β que maximice la verosimilitud. En ese caso definiremos el error de ridge de la siguiente manera:

Definition 5.2. Sean $\beta \in \mathbb{R}^d$ y $\lambda > 0$ $err_{Log,\beta,\lambda} : \mathbb{R}^d \times \{0,1\} \to \mathbb{R}$
$err_{Log,\beta,\lambda}(x,z) = \left(z \cdot log\left(1 + e^{-x\cdot\beta}\right) + (1-z)\, log\left(1 + e^{x\cdot\beta}\right)\right) + \lambda(\beta\cdot\beta)^2$

5.3 Maximum a posteriori

En la sección anterior estudiamos una manera de prevenir sobre-ajuste utilizando la penalización de ridge, en esta sección hablaremos de otra forma de prevenir tal sobre-ajuste, esta vez desde un enfoque bayesiano.

La idea principal consiste en utilizar la siguiente observación: los parámetros β que hacen posible nuestro modelo no son cualesquiera pues estamos observando la base de datos S. Así que en lugar de maximizar la probabilidad de dado β, obtener la base de datos S, maximizaremos la probabilidad $\mathbb{P}(\beta|S)$.

Utilizando el célebre teorema de bayes obtenemos

$$\mathbb{P}(\beta|S) = \frac{\mathbb{P}(S|\beta)\,\mathbb{P}(\beta)}{\mathbb{P}(S)}$$

Notemos que:

- $\mathbb{P}(S|\beta)$ es la verosimilitud usual,
- $\mathbb{P}(\beta)$ es generalmente simple pues nosotros decidimos arbitrariamente una región dónde creemos que ocurrirá nuestro parámetro, es aquí donde podríamos prevenir el sobre-ajuste.
- $\mathbb{P}(S)$ esta distribución también es complicada sin embargo gracias a la ecuación dada por el teorema de bayes, hemos simplificado un poco su búsqueda.

5.4 Intervalos de confianza

Definition 5.3. Sea $S = \{x_1, \ldots, x_N\} \subseteq \mathbb{R}^d$, definimos el promedio empírico de S como $\mu_S = \frac{1}{N}(x_1 + \ldots x_N)$.

Supongamos que $\beta \in \mathbb{R}$ y $\beta \sim \mathbf{N}(\mu, \sigma^2)$ donde variamos la esperanza μ y conocemos la varianza σ^2.

Para cada $\delta \in [0, 1]$ definimos el intervalo para β con confianza $1 - \delta$ como el invervalo en $\mathbb{R}$:

$$\left(\mu_S - \frac{\sigma^2 q^N_{1-\frac{\alpha}{2}}}{\sqrt{N}}, \mu_S + \frac{\sigma^2 q^N_{1-\frac{\alpha}{2}}}{\sqrt{N}}\right)$$