Elementos básicos de la regularización

¡Comencemos!

Competencias

- Conocer las normas L1.
- Utilizar los métodos de regularización para resolver problemas de dimensionalidad y mejora de desempeño predictivo.

Introducción

En esta sección conocerás las normas L1. Por otra parte, utilizaremos los métodos de regularización para resolver problemas de dimensionalidad y mejora de desempeño predictivo.

¡Vamos con todo!

Regularización Paramétrica y métodos de contracción (shrinkage methods)

Recordando nuestras bases estadísticas, cuando queremos estimar ciertos parámetros de alguna distribución o modelo utilizando ciertos datos disponibles, lo haremos mediante el Método de Máxima Verosimilitud, pues este nos entrega estimadores casi insesgados con varianza casi mínima. Sin embargo, para muchas aplicaciones y modelos modernos esta aproximación es ineficiente o inadecuada debido a la alta cantidad de atributos. Tener estimadores insesgados se ha convertido en un lujo que, por lo general, no podremos permitirnos.

La regresión lineal clásica ya vista se apoya en una versión del estimador MLE (Maximum Likelihood Method) para encontrar los parámetros del modelo. Recordemos la notación y forma de una regresión lineal multivariada, de forma matricial:

$$\mathbf{Y} = \mathbf{X} \cdot \vec{\beta}$$

O escrito de otra forma:

$$y_i = \sum_{j=1}^{p} x_{i,j}\beta_j + \beta_0$$

Donde:

$$\beta_j$$

j-ésimo coeficiente asociado al atributo/columna j.

$$x_{i,j}$$

i-ésimo registro de la columna j del dataset.

$$\beta_0$$

Intercepto de la regresión.

$$p$$

Es la dimensión del dataset, o dicho de otra forma, la cantidad de columnas/atributos que tenemos.

En nuestro modelo lineal, los parámetros que se deben encontrar al entrenar son los coeficientes

$$\beta_j$$

que definen la recta.

En este modelo usamos una función objetivo que minimiza la suma de los cuadrados de los errores (SSE):

$$\beta_{\text{OLS}} = \underset{\beta}{\text{argmin}} \sum_{i}^{n} (y_i - \hat{y_i})^2$$

Digresión: Complejidad computacional y la regresión lineal

Un aspecto importante a considerar a lo largo de todo este módulo es la complejidad computacional de entrenar nuestros modelos. No es lo mismo entrenar un modelo con 1 parámetro (como una regresión lineal simple univariada), que entrenar una red neuronal con una cantidad de parámetros del orden de los millones. No entraremos en detalles sobre cómo calcular la complejidad computacional de un algoritmo, pero si es bueno tener una buena intuición sobre este tema.

En general, un algoritmo (o modelo de entrenamiento en nuestro caso) es más complejo computacionalmente si realiza más operaciones elementales (como multiplicar dos números o sumarlos). Por lo tanto para un cierto modelo, la complejidad la podremos intuir considerando cuantas operaciones va a tener que realizar para antes de poder minimizar la función objetivo a un cierto valor aceptable (o mínimo).

En el caso de la regresión lineal, la complejidad lamentablemente no la podremos intuir sin incluir notación matricial engorrosa y tener que analizar casos de singularidades, sin embargo, de la regresión lineal podemos quedarnos con dos cosas:

1. Mientras más grande sea la matriz de atributos (filas y columnas del dataset) más operaciones tendrá que realizar, aumentando su tiempo de ejecución.
2. La regresión lineal, a diferencia de los métodos que veremos luego en el módulo, no es un método iterativo, es decir, encuentra el mínimo de la función objetivo sin necesidad de ir acercándose poco a poco a la solución. Esto es clave para analizar la complejidad del modelo pues en los modelos iterativos tendremos que decidir

cuántas iteraciones hacer, considerando que cada iteración añade un costo correspondiente en complejidad.

Intuición base sobre los métodos de regularización

Cuando nos enfrentamos a datos donde la cantidad de atributos es sustancialmente mayor a la cantidad de observaciones, podemos sufrir de sobre-ajuste en nuestra función candidata que explique de muy buena forma las observaciones en la muestra de entrenamiento y fallar en generalizar la función en la muestra.

La estrategia que aprenderemos a lo largo de esta lectura es implementar técnicas que impidan que el modelo sufra de sobre ajuste, conocidas como regularización en norma L1 y L2.

A grandes rasgos los métodos de regularización buscan penalizar parámetros βestimados con valores muy grandes. El objetivo es agregar un penalizador que sea proporcional a la magnitud de β.

Hay dos modos (también conocidas como normas) de generar regularización de parámetros:

- **Norma L1:** Llamada norma euclídea, sintetiza la distancia entre dos vectores mediante $\sqrt{x^2 + y^2}$
- **Norma L2:** Llamada norma taxicab, sintetiza la distancia entre dos vectores mediante $|\beta|$ lo que genera ángulos rectos entre los vectores.

Ambas normas tienen sus virtudes y defectos. Una imagen clásica que proviene de Hastie, Tibshirani y Friedman (2009) visualiza las normas (como las áreas celestes) y el efecto que tienen en los parámetros estimados (contornos rojos). El objetivo es encontrar un punto donde el contorno se junte con el área regularizada. Mientras que con la normalización L2 no hay soluciones únicas (dado que el área celeste es un círculo y hay un solo borde contínuo), en la normalización L1 existe la posibilidad que el contorno del parámetro estimado se junte con una de las esquinas del área, dando paso a situaciones donde algún parámetro sea igual a cero.

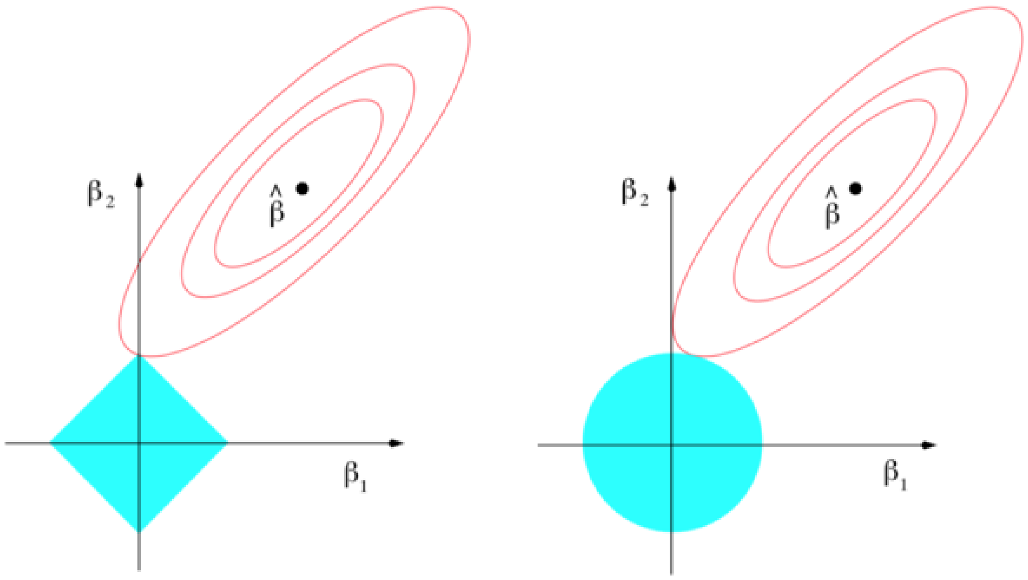

FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Imagen 3. Estimación de regresión.