Sesión 4: Modelos de regresión y clasificación

ML-IA: Módulo de Machine learning

Jacob Muñoz-Castro

Universidad de los Andes

12 de noviembre de 2020

Agenda

1. Regularización
2. Red elástica
3. Algoritmos de Clasificación
4. k-vecinos más cercanos (kNN)
5. Regresión logística
6. Análisis de discriminante lineal (LDA)

Regularización

- A la hora de seleccionar modelos: empezar con un modelo general con p variables y sacar variables no significativas? (profundización sesión 3).
- Una primera idea es:
  - Ajustar el modelo, obtener coeficientes y p-valores
  - Sacar los atributos con p-valor menor a un α
  - ¿Por qué es una mala idea? Multicolinealidad.
- Una mejor estrategia es penalizar complejidad.
- Lasso (regularización L1) es un algoritmo que logra esto.

Lasso

- Para $\lambda \geq 0$,

$$\min_{\beta} L(\beta) = \underbrace{\sum_{i=1}^{n}(y_i - x_i'\beta)^2}_{\textbf{Costo de ajuste}} + \underbrace{\lambda \sum_{s=2}^{p} |\beta_s|}_{\textbf{Penalidad}}$$

- El primer coeficiente es para la constante (baseline model).
- **Intuición:**
  - Si $\lambda = 0$, volvemos a MCO.
  - Si $\lambda \to \infty$, el modelo converge a su versión más ingenua porque todas las variables penalizan excesivamente.

Ridge

- Para $\lambda \geq 0$,

$$\min_{\beta} R(\beta) = \underbrace{\sum_{i=1}^{n}(y_i - x_i'\beta)^2}_{\textbf{Costo de ajuste}} + \underbrace{\lambda \sum_{s=2}^{p}(\beta_s)^2}_{\textbf{Penalidad}}$$

- Intuición es similar a Lasso pero el problema es completamente distinto.
- Este método vuelve diferenciable el problema y por lo tanto permite soluciones interiores.
- Esto último le agrega estabilidad al problema.

L1 y L2

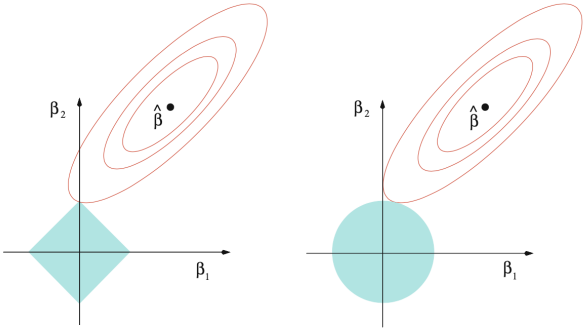


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are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

Más atributos que registros ($k > n$)

- Si el objetivo es precisión: minimizar el error de predicción con Ridge y Lasso.
- Si el objetivo es dimensionalidad: reducción del espacio de predictores.
- MCO falla con $k > n$ (invertibilidad).
- Esto ocurre porque regularización expande el sistema que estamos resolviendo con k puntos adicionales (data augmentation).
- Algo a tener en cuenta con Lasso:
  - Lasso elige solo un atributo cuando tenemos un grupo de variables correlacionadas (género y dummy está embarazada).
  - Esto hace inestable la predicción porque la selección de la variable (agrupamiento) es aleatorio.
  - Ridge por otro lado atenúa los coeficientes de las variables correlacionadas.

Elastic Net

- ¿Por qué no lo mejor de dos mundos? La mitad feliz.
- Red elástica (EN) nos transforma el problema a

$$\begin{aligned}\min_{\beta} EN(\beta) &= \sum_{i=1}^{n}(y_i - x_i'\beta)^2 + \lambda_1 \sum_{s=2}^{p}|\beta_s| + \lambda_2 \sum_{s=2}^{p}(\beta_s)^2 \\ &= \sum_{i=1}^{n}(y_i - x_i'\beta)^2 + \lambda \sum_{s=2}^{p}(\alpha|\beta_s| + (1-\alpha)\beta_s^2)\end{aligned}$$

- Lasso nos seleccina las variables
- La parte estrictamente convexa de la penalidad de Ridge nos soluciona el problema de inestabilidad por agrupación.
- Ahora pensemos en que esto empieza a agregar mucho sesgo. Para corregir esto, se sugiere corregir por $\frac{1}{\sqrt{1+\lambda_2}}$ (para más detalle Zou & Hastie (2005)).

Selección de λ

- Validación cruzada
- Elija una grilla de valores de λ y compute el error de CV para cada valor
- Elija el λ^* que minimiza el error de predicción

Clasificación

- Dar crédito basado en historial crediticio, variables demográficas..
- Clasificar e-mail: spam, personal, redes sociales, basado en el contenido.
- **Objetivo:** Clasificar y basado en X.
- y puede ser: cualitativa (spam, tipo de cliente, ...), no necesariamente ordenada, y no necesariamente dos categorías.

Clasificación

- Pero, empecemos por 2 categorías.
- Dos estados de la naturaleza $y \rightarrow i \in \{0, 1\}$
- Dos acciones $\hat{y} \rightarrow j \in \{0, 1\}$
- Tenemos entonces que los eventos ocurren con probabilidades $p = \Pr(Y = 1|X)$ y $1 - p = \Pr(Y = 0|X)$.
- Hasta ahora hemos medido nuestra función de pérdida con $d(y, \hat{y})$.
- Ahora, tenemos que nuestra función de pérdida es $L(i, j)$ que penaliza cuando la clasificación es impura.

Clasificación

- El riesgo de hacer clasificaciones va a estar dado por el valor esperado de la pérdida de tomar acción i.

$$E[L(i,j] = \sum_j p_j L(i,j)$$

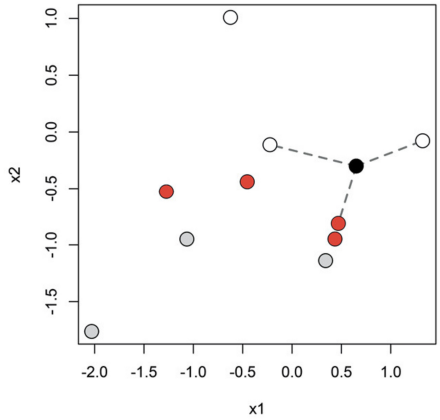
- La meta es la misma que antes: minimizar el riesgo (antes MSE).
- La expresión anterior la podemos reducir a

$$R(i) = (1-p)L(i,0) + pL(i,1)$$

- Esto se reduce entonces a encontrar $p = Pr(Y = 1|X)$

k-vecinos más cercanos (kNN)

- El algoritmo KNN predice la clase $\hat{y}$ a partir de x haciendo la pregunta: ¿cuál es la clase más común para las observaciones al rededor de x?



KNN

El algoritmo: dado un vector de entrada x_l al cual queremos asignar una clase

- Ecuentre los k vecinos más cercanos a este vector de nuestros datos etiquetados $\{(x_i, y_i)\}_{i=1}^{n}$. Definimos cercano por distancia euclidiana

$$d(x_i, x_l) = \sqrt{\sum_{j=1}^{p}(x_{ij} - x_{lj})^2}$$

- De esto resulta una colección de datos etiquetados $\{(x_i, y_i)\}_{i\in\Omega(k)}$ con $\Omega(k)$ es el conjunto de subíndices de los vecinos más cercanos.
- Usaremos (en la mayoría de algoritmos) regla de votación:

$$\hat{y}_l = moda(\{y_i\}_{i\in\Omega(k)})$$

KNN

Problemas mayores con implicaciones prácticas

- KNN es inestable como función de K
- Esta inestabilidad hace difícil elegir K y validación cruzada no funciona bien con KNN
- Este método es exhaustivo para un computador: cada predicción para un nuevo x requiere mucho conteo. Es un método muy caro si se quiere usar en datos grandes.
- Buena idea en teoría, pero es poco llamativo para usarlo en la práctica.

Regresión logística

- Recordamos que nuestro interés es encontrar $\Pr(Y = 1|X)$.
- Otro acercamiento es poner nuestros atributos en términos conocidos (regresión lineal) y unir esto con una función (link function):

$$\Pr(Y = 1|X) = f(X\beta)$$

- Una regresión logística cumple con esto:

$$\Pr(y = 1|X) = \frac{exp(X\beta)}{1 + exp(X\beta)}$$

Análisis de discriminante lineal (LDA)

- Recordamos del teorema de Bayes:

$$\Pr(Y = 1|X) = \frac{f(X|Y = 1)p(Y = 1)}{m(X)}$$

- $m(X)$ es la marginal de la distribución de X

$$m(x) = \sum f(X|Y)p(Y)$$

- **Intuición:** Nuestra predicción de la probabilidad va a depender de la distribución de X condicionado a cada clasificación (sale de la marginal) y la frecuencia de $Y = 1$ ($\hat{p}(Y = 1)$).

LDA

- El algoritmo encuentra el promedio de X para cada clase y encuentra una proyección direccional que maximiza la separación de los promedios.
- También tiene en cuenta la vairación intraclase para encontrar la proyección que minimiza traslapados de las distribuciones.