Ajuste de modelos para clasificación/regresión salida binarias (SÍ/NO): planteamiento de problema

Antonio Sala

DISA – AI^2 – Universitat Politècnica de València

Presentaciones en vídeo: http://personales.upv.es/asala/YT/V/**clasifintr1**.html, http://personales.upv.es/asala/YT/V/**clasifintr2**.html, http://personales.upv.es/asala/YT/V/**clasifNoLS**.html

Presentación

Motivación:

Dedicar más "tiempo de estudio" a una asignatura debería subir la "probabilidad de aprobar". La presencia de las palabras clave "Nigeria", "herencia" debería subir la "probabilidad de ser Spam" de un correo...
Quiero saber si una foto es o no de un perro.

Objetivos:

Comprender el planteamiento de problemas cuando lo que se espera de mi modelo es ajustar ciertos datos experimentales etiquetados "SÍ/NO".
Relación y diferencias con el ajuste por mínimos cuadrados.

Contenidos:

Planteamiento del problema: objetivos, ejemplos. Conclusiones.
Apéndice: ¿Por qué no seguir haciendo mínimos cuadrados?.

El problema de la "clasificación" (binaria) supervisada

Disponemos de serie de datos de variables X e Y, en pares (x_i, y_i) con:

- $x_i \in \mathbb{R}^n$ (o componentes "categóricos" {0,1}), en general $x_i \in \mathbb{X}$
- $y_i \in \{0, 1\}$ categórica. [conocida: aprendizaje **supervisado**]

El significado de "categórico" es que 0 o 1 son "etiquetas", no "números sobre los que hacer operaciones algebraicas", al menos en principio.
Suponemos "binario" por simplicidad, aunque podría ser multiclase {"Perro", "Gato", "Árbol"}...
*Siempre podremos trasladar a binario con $x \in \{0,1\}^3 =$ {"¿Es un perro?", "¿Es un gato?", "¿Es un Árbol?"}; la clase "Perro" llevaría asociada etiqueta $\{1, 0, 0\}$, "No perro" etiqueta $\{0, X, X\}$.

Ejemplos:

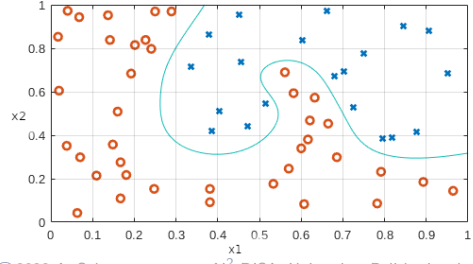
- Encuesta tiempo estudio + resultados examen: {María: (78h, aprueba); Pepe: (22h, suspende); Andrés: (72h, suspende), ...}
- {Foto 1: perro; Foto 2: NO perro; Foto 3: perro, ...}

Objetivos del problema (1)

[A] Clasificación **perfecta**, aprender TODOS los puntos (x_i, y_i).

- Puede no ser posible (aparte de fuerza bruta, memorización "rote learning" que no "generaliza") o no recomendable (predecir nota de examen en función de tiempo dedicado a estudio: 72h aprueba, 73h suspende, 74h aprueba)... O ser posible de muchas formas, y querer la "mejor" en cierto sentido.

Ejemplo: ["letras" a partir de "32x32 pixels"] buscamos idealmente ajuste perfecto.



$(x_1, x_2) \in \text{ROJO} \Leftrightarrow f(x_1, x_2, \theta^*) < 0$

Objetivos del problema (2)

[B] Clasificación **imperfecta**, aprender unos parámetros θ de $f(x, \theta)$ que:

- Con $f(x, \theta) : \mathbb{X} \mapsto \{0, 1\}$, minimizan cierta medida de "error", loss function $\mathcal{L}$, $\mathcal{L}(y_i, f(x_i, \theta))$, incluso con asimetrías pesando diferente los falsos positivos (diabetes diagnosticada a paciente sano) de los falsos negativos (diabetes no diagnosticada).

  $\mathcal{L}(1, 1) = \mathcal{L}(0, 0) = 0; \;\; \mathcal{L}(1, 0) = 7, \; \mathcal{L}(0, 1) = 2.$

[interpretación determinista]

Objetivos del problema (3)

[C] Clasificación **imperfecta**, aprender unos parámetros θ de $f(x, \theta)$ que:

- Con $f(x, \theta) : \mathbb{X} \mapsto [0, 1]$, maximizan probabilidad del conjunto de y_i dado x interpretando $f(x, \theta) \equiv p_\theta(y = 1|X = x)$ (maximum likelihood estimation, máxima verosimilitud)

[interpret. probabilística]

Solución trivial determinista/probabilista: $f(x_i) = 1$ si $y_i = 1$; $f(x_i) = 0$ si $y_i = 0$. [memorizar datos]

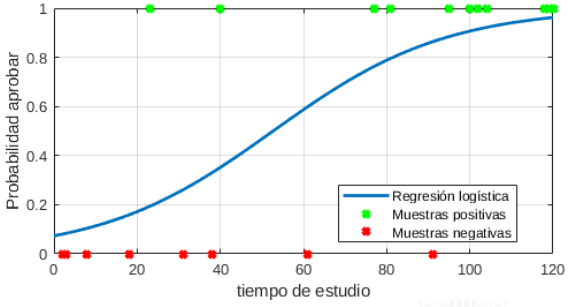
No nos vale... la "forma de f" debe estar justificada por alguna teoría/hipótesis de partida o, bueno, debe tener una cierta "suavidad" razonable... Eso es lo que se codifica en la parametrización θ.

Objetivos del problema (3b)

Existen distintas "formas de f" probabilista populares en literatura (logit, probit...)

Ejemplo: ["prob. aprobar" en función de "tiempo de estudio"] no queremos acertar los datos siempre.

$Prob(aprobar \mid x) \approx f(x, \theta^*)$



Conclusiones hasta ahora

Hemos visto cómo interpretar el problema "ajustar modelos $f_\theta(input) := f(input, \theta)$ a datos etiquetados SÍ/NO".

[A] Clasificación perfecta (letras, imágenes)... regla de decisión $f_\theta(input) > 0$.

[B] Si la clasif. perfecta no es posible, minimizar coste de falsos positivos y falsos negativos (no necesariamente iguales)

[C] A veces, se busca una interpretación probabilística... $p(y = true|x) = g_\theta(x)$.

- Las opciones están más cerca de lo que parece a primera vista:

  (1) El valor de f más o menos positivo podría indicar lo "seguro" que está el algoritmo de que hay un "perro"... A veces es imposible no equivocarse en algo si θ es "sencillo". Reconocer "perro" en función de la media del color verde de pixels es difícil.

  (2) A veces primero se optimiza un coste probabilístico (p.ej. regresión logística) y luego se decide un umbral para clasificar según la importancia de falsos positivos o falsos negativos.

¿Y por qué no seguimos haciendo mínimos cuadrados?

Los datos $\{0, 1\}$ podrían ajustarse minimizando $\mathcal{L}(\theta) = \sum_i (y_i - f(x_i, \theta))^2$ con una recta de regresión, red neuronal, polinomio o lo que sea. ¿Por qué no hacerlo así, y no complicarse con otras cosas? **Podría funcionar** y con f lineal sería **MUY eficiente computacionalmente** (con $f(x_i, \theta) = \Phi^T(x_i) \cdot \theta$), pero...

Clasificación "perfecta":

- En principio, en entorno "determinista", valdría: queremos aprender una función que devuelva '0' o '1' cuando toque... pero a lo mejor hay opciones que consiguen lo mismo con funciones más "sencillas"...

  Realmente sólo nos interesa que $f(x_i, \theta) > 0.5$ en muestras positivas, $f(x_i, \theta) < 0.5$ en muestras negativas. Puede que una función "sencilla" consiga eso (no hay problema en que $f(x_7, \theta) = -1241$) pero que no ajuste "bien" a los datos (forzar $f(x_7, \theta) \approx 0$ puede deformar f en otros sitios si no es suficientemente flexible)... pero añadir 'signos' pierde gradiente.

¿Y por qué no seguimos haciendo mínimos cuadrados?

Los datos $\{0, 1\}$ podrían ajustarse minimizando $\mathcal{L}(\theta) = \sum_i (y_i - f(x_i, \theta))^2$ con una recta de regresión, red neuronal, polinomio o lo que sea. ¿Por qué no hacerlo así, y no complicarse con otras cosas? **Podría funcionar** y con f lineal sería **MUY eficiente computacionalmente** (con $f(x_i, \theta) = \Phi^T(x_i) \cdot \theta$), pero...

Clasificación "perfecta":

- En principio, en entorno "determinista", valdría: queremos aprender una función que devuelva '0' o '1' cuando toque... pero a lo mejor hay opciones que consiguen lo mismo con funciones más "sencillas"...

  Realmente sólo nos interesa que $f(x_i, \theta) > 0.5$ en muestras positivas, $f(x_i, \theta) < 0.5$ en muestras negativas. Puede que una función "sencilla" consiga eso (no hay problema en que $f(x_7, \theta) = -1241$) pero que no ajuste "bien" a los datos (forzar $f(x_7, \theta) \approx 0$ puede deformar f en otros sitios si no es suficientemente flexible)... pero añadir 'signos' pierde 'gradiente'.

Más comentarios sobre el tema en https://notesonai.com/Least+squares+for+classification

¿Y por qué no seguimos haciendo mínimos cuadrados?

Los datos $\{0,1\}$ podrían ajustarse minimizando $\mathcal{L}(\theta) = \sum_i (y_i - f(x_i,\theta))^2$ con una recta de regresión, red neuronal, polinomio o lo que sea. ¿Por qué no hacerlo así, y no complicarse con otras cosas? **Podría funcionar** y con f lineal sería **MUY eficiente computacionalmente** (con $f(x_i,\theta) = \Phi^T(x_i)\cdot\theta$), pero...

Clasificación "imperfecta":

- Dar coste asimétrico a falsos $+$ y falsos $-$ requiere modificar $\mathcal{L}$.
- En entornos probabilísticos, el error cuadrático se interpreta como log-probability de $e^{-\epsilon^2/\sigma^2}$ (distribución normal), que "no cuadra" con salidas $\{0,1\}$ (Bernoulli).
- En bastantes casos sería más lógico la cuadrática "truncada":

$$\mathcal{L}(y_i, f_i) = \begin{cases} \mathbf{0} & y_i = 1 \,\&\, f_i \geq 1 \mid y_i = 0 \,\&\, f_i \leq 0 \\ (y_i - f_i)^2 & \text{resto de casos} \end{cases}$$

Ello daría más flexibilidad a "f" para poder resolver problemas con menos parámetros ajustables; claro, también podemos complicar más $\mathcal{L}$, ¿no?

Más comentarios sobre el tema en https://notesonai.com/Least+squares+for+classification