La tarea de clasificación en detalle (II)

Alfredo Cuesta Infante

E. T. S. Ingeniería Informática
Universidad Rey Juan Carlos

Master Univ. en Visión Artificial
Reconocimiento de Patrones

Contenidos

Contenidos

Clasificación no lineal

Transformación de características

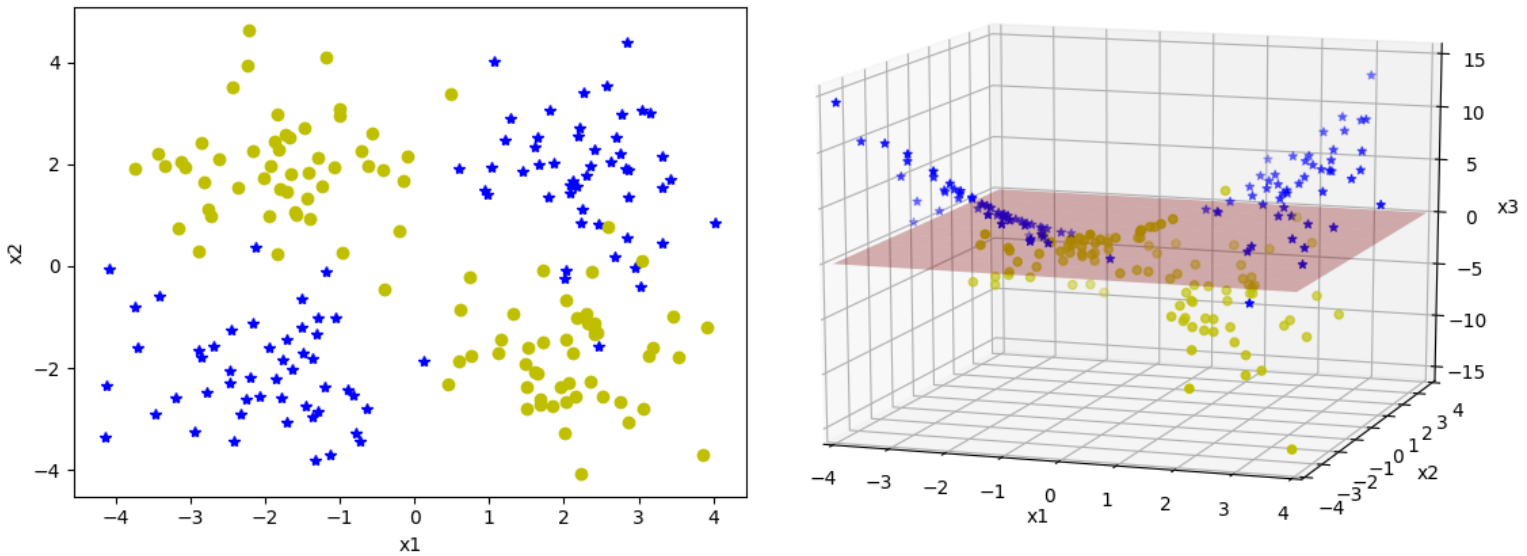


Figura: (Izq.) Conjunto de datos no separable linealmente. (Der.) Si a cada ejemplo le añadimos una nueva característica $x_3 = x_1 x_2$, en tres dimensiones sí lo podemos separar linealmente mediante el plano rojo semitransparente.
[Fuente: Original de A. Cuesta]

Una vez transformado se aplica un clasificador lineal

Clasificación no lineal

Tranformación polinómica

- Supongamos ejemplos bidimensionales $\mathbf{x} = (x_1, x_2)$,
  y aplicamos la transformación $\phi(\mathbf{x}) = \mathbf{z} = (z_1, z_2, z_2)$; donde
  $$z_1 = x_1 x_2, \; z_2 = x_1^2, \; z_3 = x_2^2$$
  $\Rightarrow$ Todas las características polinómicas de orden 2 (incluyendo x_1 y x_2)
- Supongamos ahora que la transformación es
  $\phi(\mathbf{x}) = \mathbf{z} = (z_1, z_2, z_2, z_3, z_4, z_5, z_6, z_7)$; donde
  $$z_1 = x_1 x_2, \; z_2 = x_1^2, \; z_3 = x_2^2,$$
  $$z_4 = x_1^2 x_2, \; z_5 = x_1 x_2^2, \; z_6 = x_1^3, \; z_7 = x_2^3.$$
  $\Rightarrow$ Todas las características polinómicas de orden 3 (incluyendo x_1 y x_2)
- En general para n variables, una tranformación polinómica de grado d produce
  $$\frac{(n+d)!}{d!n!}$$
- 2 peligros: Sobreajuste y dimensionalidad

Clasificación no lineal

Ejemplo

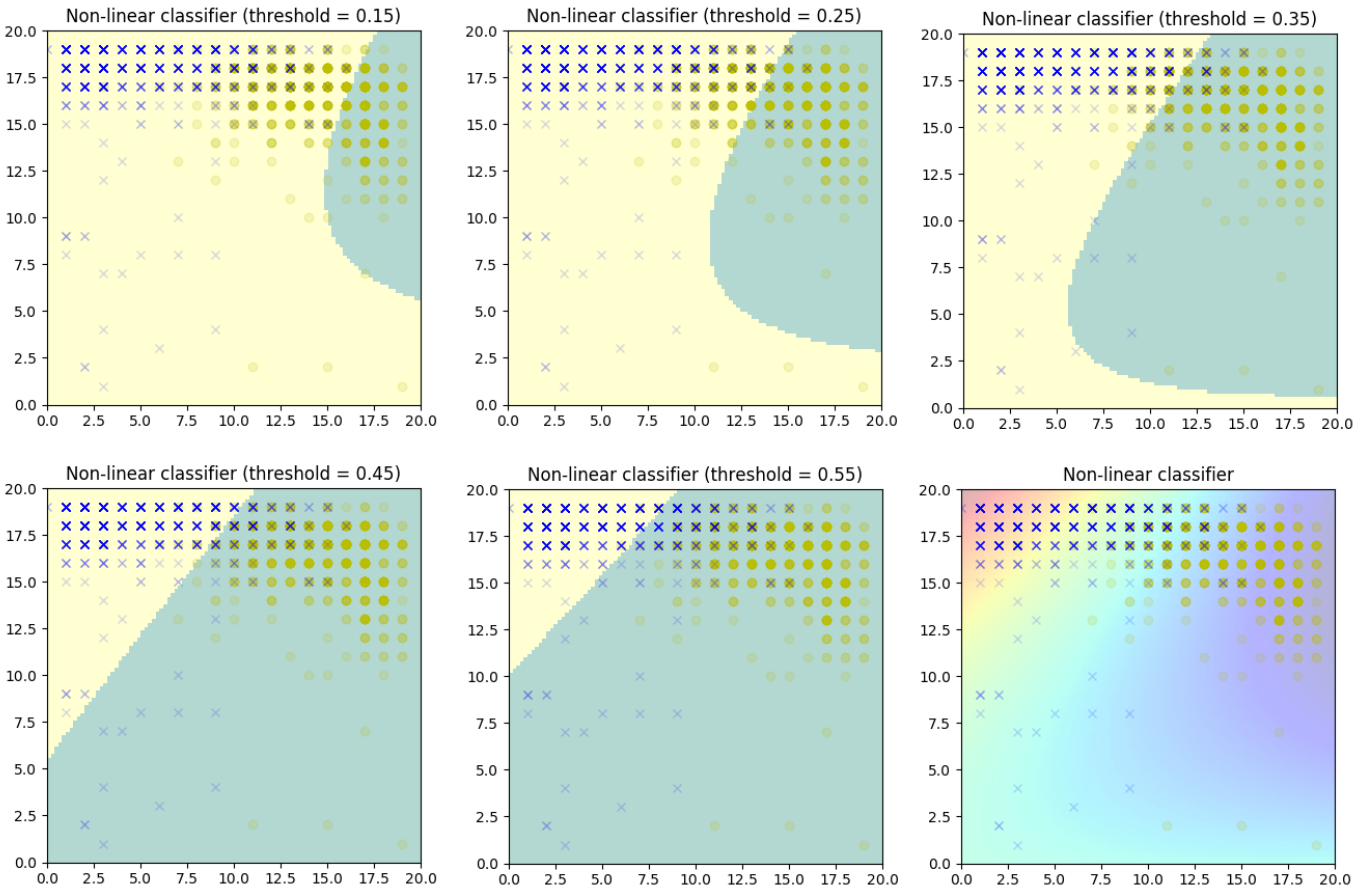
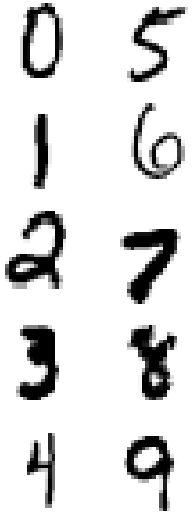


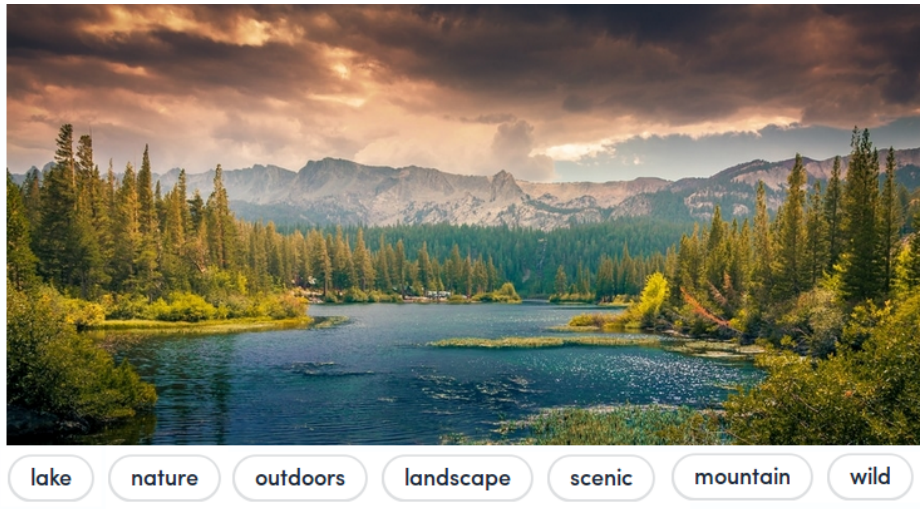
Figura: Clasificación no lineal con nuevas características polinómicas de grado 2 para umbrales $\{0.15, 0.25, 0.35, 0.45, 0.55\}$. En la última gráfica se muestra el valor de la función de decisión sobre el conjunto de datos. [Fuente: Original de A. Cuesta]

Contenidos

Problemas multiclase y multietiqueta



(a)



(b)

Figura: (a) Problema multiclase (b) Problema multietiqueta [Fuente: Original de A. Cuesta]

Representacion One-hot

	0	1	2	3	4	5	6	7	8	9
imagen MNIST de un 3				1						

(a)

	river	lake	sea	ocean	indoor	outdoor	landscape	skyline	mountain	city
Figura (b)		1				1	1		1	

(b)

(a) *one-hot* multiclase (b) *one-hot* multietiqueta. [Fuente: Original de A. Cuesta]

El problema multiclase

- Hay K clases distintas y el clasificador asigna una única etiqueta $\hat{t} \in \{t_1, t_2, \ldots, t_K\}$ a cada ejemplo.
- Estrategia **Uno contra todos** (*One vs All*, OvA)
  - Hay que entrenar K clasificadores
  - Las clases suelen estar desequilibradas
- Estrategia **Uno contra uno** (*One vs One* OvO)
  - Hay que entrenar $K(K-1)/2$ clasificadores
  - Pero los conjuntos de entrenamiento de cada uno son más pequeños
- Ambas pueden dar lugar a zonas de incertidumbre

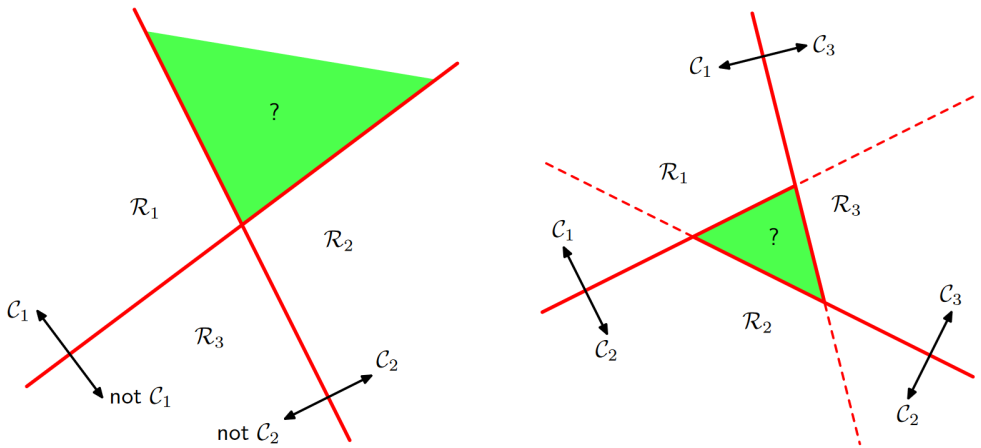


Figura: La región verde produce problemas de clasificación con estrategia OvA (Izq.), y con la estrategia OvO (Der.) [Fuente: Original de "Pattern Recognition and Machine Learning", C.M. Bishop, 2006]

El problema multiclase

Regresión Softmax

- Generalización de la Regresión Logística a K clases diferentes

  Recordar que consistía en hacer calcular los pesos del modelo lineal y aplicar la función logística al *score* obtenido para cada ejemplo.

- Ahora:
  - Tenemos K clases ..... en vez de 2
  - Aplicamos la función Softmax ..... en vez la función Logistica
  - La probabilidad estimada de que la asignación del ejemplo $\mathbf{x}$ a la clase k sea correcta es

$$\hat{p}_k = \frac{\exp\left(\mathbf{w}_k^T \mathbf{x}\right)}{\sum_{k=1}^{K} \exp\left(\mathbf{w}_k^T \mathbf{x}\right)} ,$$

  - ¡¡ Cada clase tiene su propio vector de pesos $\mathbf{w}_k$!!

    $\Rightarrow$ podemos agrupar todos los $\mathbf{w}_k$ en una matriz $\mathbf{W}$ de K columnas
  - La función de coste *log-loss* es la entropía cruzada

$$L(\mathbf{W}; \mathbf{X}, \mathbf{t}) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} t_k^{(i)} \log\left(\hat{p}_k^{(i)}\right) ,$$

  - La función discriminante es

$$\hat{t} = \arg\max_k (\hat{p}_k)$$

El problema multietiqueta

- Es sencillamente entrenar un clasificador para cada etiqueta

Análisis de errores

- Se construye la matriz de confusión, pero más grande.

Ej.: Para un problema con 3 clases

	GoldLabel_A	GoldLabel_B	GoldLabel_C	
Predicted_A	**30**	20	10	TotalPredicted_A=60
Predicted_B	50	**60**	10	TotalPredicted_B=120
Predicted_C	20	20	**80**	TotalPredicted_C=120
	TotalGoldLabel_A=100	TotalGoldLabel_B=100	TotalGoldLabel_C=100	

Contenidos

Variantes del descenso de gradiente

$$\mathbf{w}' = \mathbf{w} - \eta \nabla J(\mathbf{w}; \mathbf{X}, \mathbf{t})$$

Estrategias para variar el paso de aprendizaje η

- **Templado Simulado** *Simmulated Annealing*

  η comienza en un valor alto y va disminuyendo según una regla o *curva de enfriamiento*.
  - *Lineal*: Nuevo η es igual a la actual menos N $\eta' = \eta - N$.
  - *Fraccional*: Nuevo η es igual una fracción N del actual, $\eta' = \eta / N$.

Estrategias para implementar el descenso de gradiente

- **Usando todo el lote de ejemplos**
  - Si J =MSE entonces, $\nabla J = \frac{2}{m} \mathbf{X}^T (\mathbf{X}\mathbf{w} - \mathbf{t})$
  - Pero exige trasponer $\mathbf{X}$ y multiplicar después por $\mathbf{X}$ = Costoso
- **Descenso del gradiente estocástico**
  - Calcular el gradiente utilizando un sólo ejemplo, elegido aleatoriamente, en cada iteración
- **Descenso en mini-lotes**
  - Calcular el gradiente sobre un subconjunto (mini-lote) que tiene un tamaño mucho menor, en cada iteración.

Contenidos

Compromiso Bias-Variance

- **Supongamos** que hubiera un conjunto de parámetros 'absolutamente' óptimo, mejor que cualquier otro conjunto, representado por θ^*
- **Entonces**, para cualquier otro conjunto de parámetros óptimo, obtenido como resultado de un aprendizaje sobre un conjunto de datos, $\hat{\theta}$, habría una discrepancia $e = (\hat{\theta} - \theta^*)^2$.
- **Puesto que** el conjunto de datos de entrenamiento NO contiene todos los datos posibles,
- **habrá** una distribución $p(\hat{\theta})$ para la cual podemos calcular el valor esperado $\overline{\theta} = \mathbb{E}(\hat{\theta})$.

$$\begin{aligned}
\mathbb{E}\left[(\hat{\theta} - \theta^*)^2\right] &= \mathbb{E}\left[(\hat{\theta} - \overline{\theta} + \overline{\theta} - \theta^*)^2\right] \\
&= \mathbb{E}\left[(\hat{\theta} - \overline{\theta})^2\right] + 2(\overline{\theta} - \theta^*)\mathbb{E}\left[\hat{\theta} - \overline{\theta}\right] + (\overline{\theta} - \theta^*)^2 \\
&= \mathbb{E}\left[(\hat{\theta} - \overline{\theta})^2\right] + (\overline{\theta} - \theta^*)^2 \\
&= \text{variance}(\hat{\theta}) + \text{bias}^2(\hat{\theta})
\end{aligned}$$

Compromiso Bias-Variance

Bias

- Se debe a hipotesis de partida erroneas.
  Ej.: Asumir que el modelo es lineal o que los datos tienen una distribución normal.
- *Bias* alto $\simeq$ subajustar, es decir generaliza demasiado

Variance

- Se debe a un exceso de sensibilidad del modelo a pequeñas variaciones en el conjunto de entrenamiento.
  Ej.: Si estamos haciendo validación cruzada y clasificación no lineal con polinomios de orden muy alto
- *Variance* alto $\simeq$ sobreajustar, es decir memoriza demasiado

Irreducible

- Se debe al ruido propio de los datos.
- El único modo de reducir este error es 'limpiarlos' o mejorar la fuente de los datos.
- En cualquier caso no depende del modelo, y por eso no aparece en la expresión de arriba.