UNIVERSIDAD EAFIT
VIGILADA | MINEDUCACIÓN

Escuela de
Ciencias Aplicadas
e Ingeniería

Lecture 02

Modelos lineales para regresión y clasificación

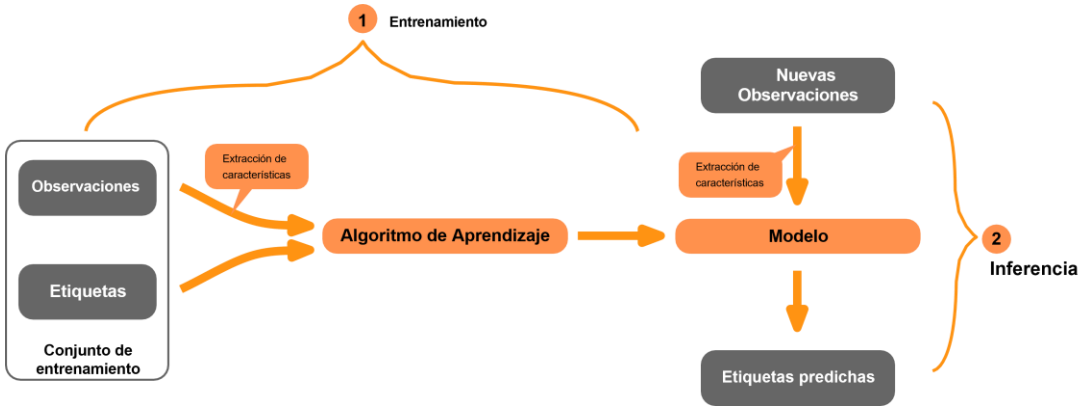
Temas de la Lección

UNIVERSIDAD EAFIT
VIGILADA | MINEDUCACIÓN
Escuela de Ciencias Aplicadas e Ingeniería

- Regresión lineal
- Sobre-asjuste, sub-ajuste
- Descomposición sesgo-varianza
- Regularización
- Ajuste de hiperparámetros

Aprendizaje supervisado

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería



Dado un conjunto de ejemplos:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$$

- Entrenamiento: Ajustar un modelo $f : \mathbf{x} \mapsto y$ a partir de los datos
- Prueba: Utilizar el modelo para predecir y dado $\mathbf{x}$, i.e,

$$\hat{y} = f(\mathbf{x})$$

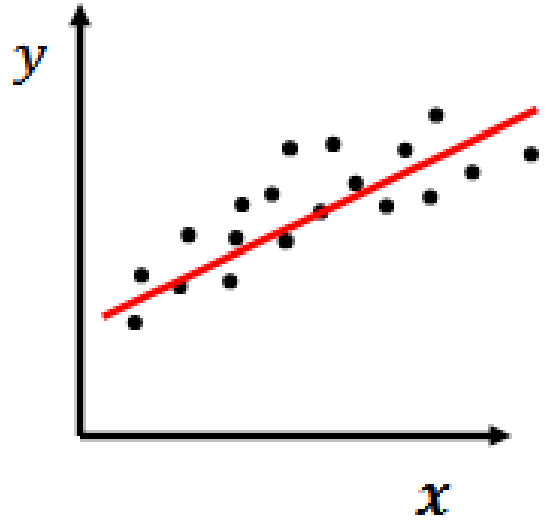
Modelo paramétrico

- Especificar forma explícita de $f(x;\theta)$ con parámetro(s) θ.
- Ejemplo:
  - Regresión lineal: $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$, $\theta = \{\mathbf{w}, b\}$.
  - Regresión logística
  - Clasificador Softmax
  - Redes profundas
- Entrenamiento: aprender θ a partir de las muestras de entrenamiento $\{(\mathbf{x}_i, y_i)\}$.
- Prueba: para una muestra de prueba $\mathbf{x}$, predecir $\hat{y}$ usando $f(\mathbf{x})$.

Regresión lineal

- Forma paramétrica: $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b.$
- Aprender $\mathbf{w}$ y b a partir de muestras $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ resolviendo:

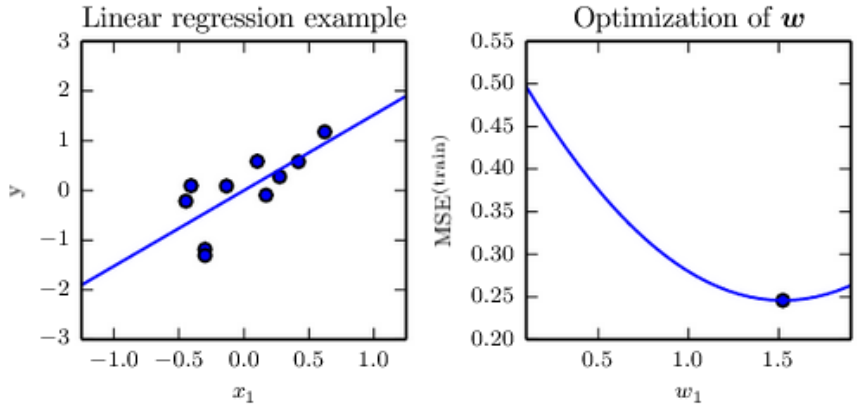
$$\min_{\mathbf{w},b} \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{w}^\top \mathbf{x}_i + b - y_i \right)^2 \equiv \frac{1}{n} \|\tilde{\mathbf{X}}\, \tilde{\mathbf{w}} - \mathbf{y}\|^2.$$



- Donde

$$\tilde{X} = \begin{bmatrix} \mathbf{x}_1^\top & 1 \\ \vdots & \vdots \\ \mathbf{x}_n^\top & 1 \end{bmatrix}, \qquad \tilde{\mathbf{w}} = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

- $\tilde{\mathbf{w}}^* = \tilde{\mathbf{X}}^\dagger \mathbf{y}$ (pseudoinversa de Moore–Penrose).



Ecuación normal

$$\Rightarrow \frac{1}{m}\nabla_{\boldsymbol{w}}||\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}||_2^2 = 0 \tag{5.8}$$

$$\Rightarrow \nabla_{\boldsymbol{w}}\left(\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}\right)^\top\left(\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}\right) = 0 \tag{5.9}$$

$$\Rightarrow \nabla_{\boldsymbol{w}}\left(\boldsymbol{w}^\top\boldsymbol{X}^{(\text{train})\top}\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - 2\boldsymbol{w}^\top\boldsymbol{X}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})} + \boldsymbol{y}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})}\right) = 0 \tag{5.10}$$

$$\Rightarrow 2\boldsymbol{X}^{(\text{train})\top}\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - 2\boldsymbol{X}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})} = 0 \tag{5.11}$$

$$\Rightarrow \boldsymbol{w} = \left(\boldsymbol{X}^{(\text{train})\top}\boldsymbol{X}^{(\text{train})}\right)^{-1}\boldsymbol{X}^{(\text{train})\top}\boldsymbol{y}^{(\text{train})} \tag{5.12}$$

Interpretación probabilística

UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

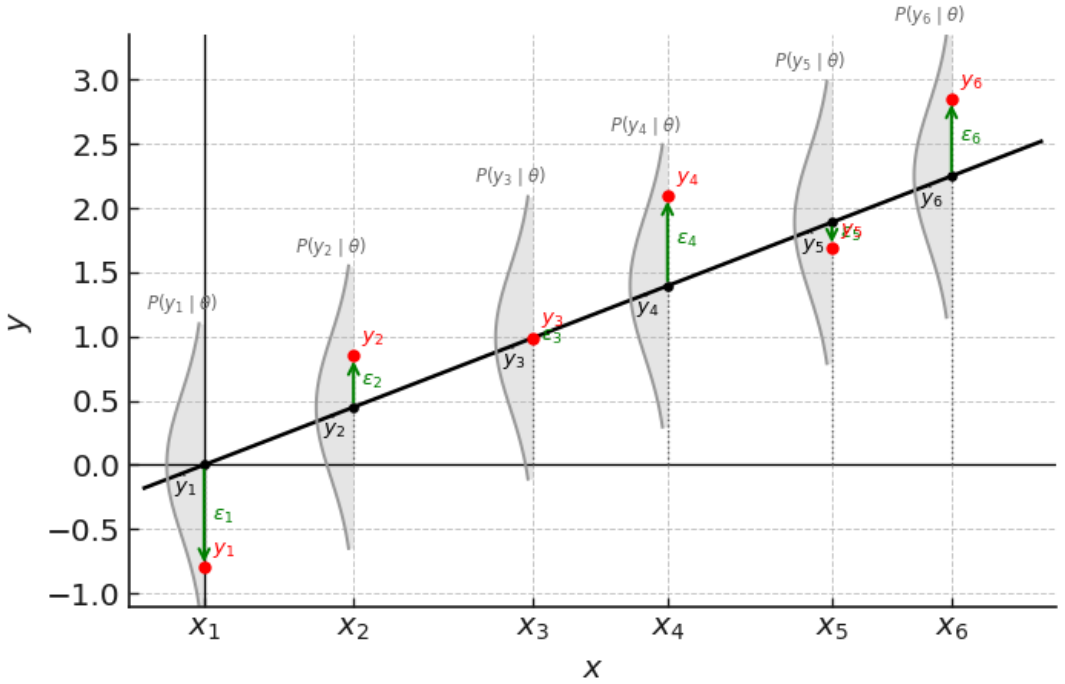
- Supongamos datos $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ i.i.d.
- Modelo generativo: $y_i = \mathbf{w}^\top \mathbf{x}_i + b + \varepsilon_i$, con $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ indep.
- Entonces

$$p(y_i \mid \mathbf{x}_i, \mathbf{w}, b, \sigma^2) = \mathcal{N}\big(y_i \mid \mathbf{w}^\top \mathbf{x}_i + b,\, \sigma^2\big).$$

- Verosimilitud total:

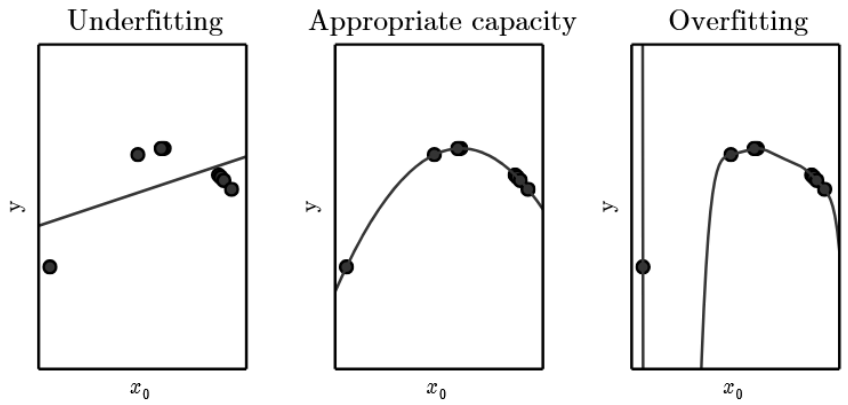
$$p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}, b, \sigma^2) = \prod_{i=1}^{n} \mathcal{N}\big(y_i \mid \mathbf{w}^\top \mathbf{x}_i + b, \sigma^2\big).$$

- Maximizar la log-verosimilitud $\iff$ minimizar $\sum_{i=1}^n (y_i - \mathbf{w}^\top \mathbf{x}_i - b)^2$. (¡Nuestro MSE!)

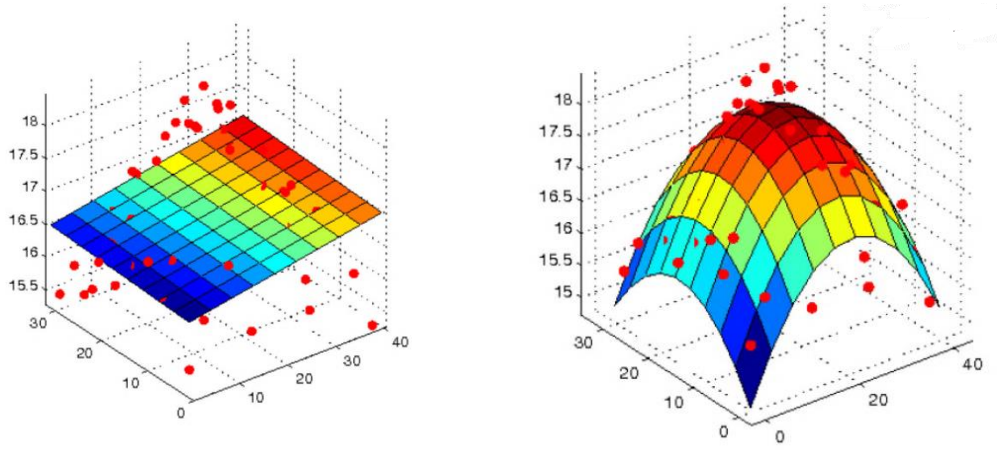


Sobre-ajuste, sub-ajuste

- Podemos crear *nuevas características* elevando potencias (expansión polinómica / cambio de base).
- 1D: $\phi(x) = [1, x, x^2, \ldots, x^d]^\top$.
- Modelo: $f(x) = \mathbf{w}^\top \phi(x)$.
  - No es lineal en x, pero sí es lineal en los parámetros $\mathbf{w}$.
- Grado $d \uparrow \Rightarrow$ mayor capacidad $\Rightarrow$ riesgo de ajustar ruido.
- Sobreajuste: error entrenamiento $\downarrow$, error prueba $\uparrow$.
- Motivación para **regularizar** (Ridge/L2, Lasso/L1) o limitar d.



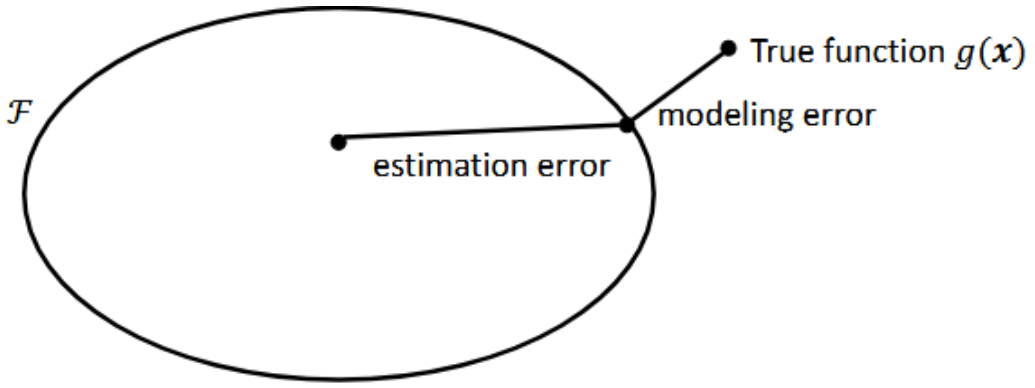
$$\phi(\mathbf{x}) = [1, x_1, x_2] \qquad \phi(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_2^2]$$



Generalización

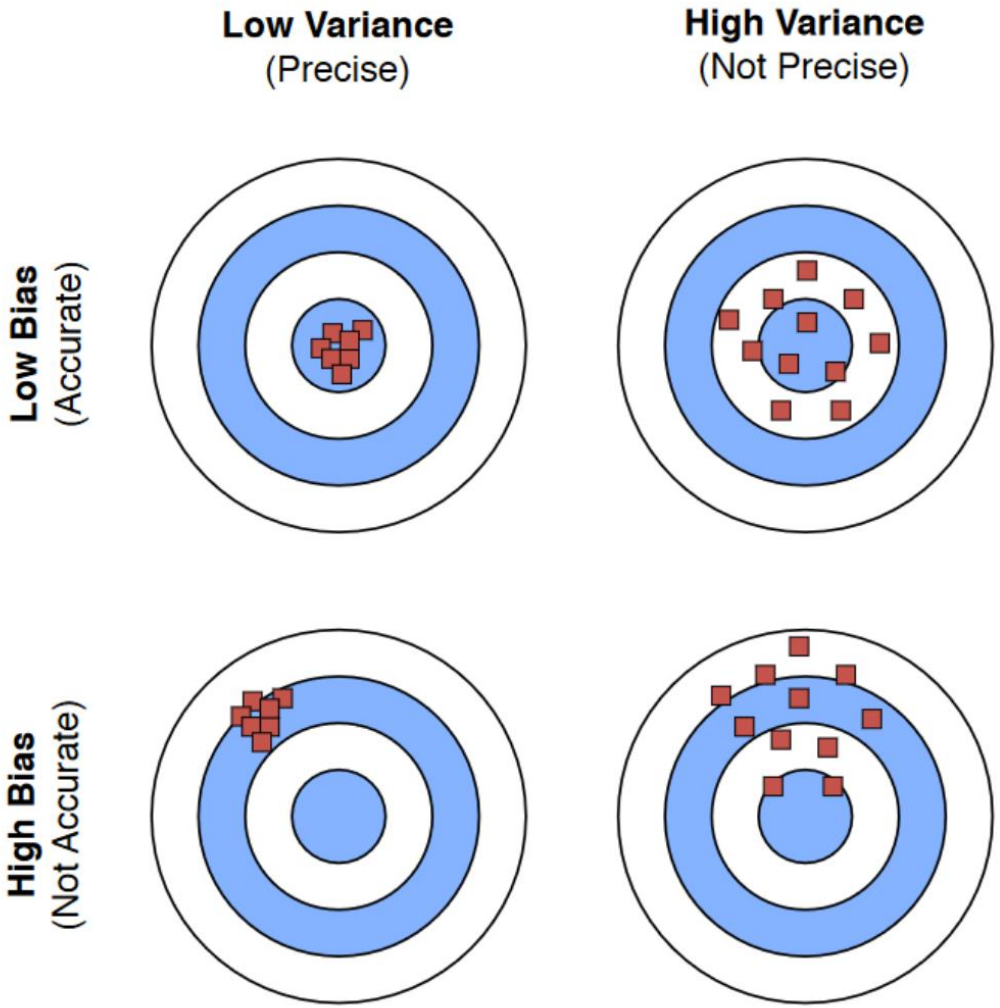
UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- $\mathcal{F}$: familia de modelos (hipótesis) que podemos representar.
- $f^\star = \arg\min_{f \in \mathcal{F}} R(f)$: mejor modelo dentro de la familia.
- $\hat{f}$: modelo aprendido con datos finitos (el que entrenamos).
- $g(x)$: función verdadera (puede estar fuera de $\mathcal{F}$).
- Error de estimación: $R(\hat{f}) - R(f^\star)$ (datos finitos / ruido / optimización).
- Error de modelado: $R(f^\star) - R(g)$ (limitación de la familia).
- Total (conceptual): $R(\hat{f}) - R(g) \approx$ modelado + estimación.



- Error de modelado = *bias*.
- Error de estimación = *varianza* respecto a los datos de entrenamiento.
- $\mathcal{F}$ más grande: error de modelado $\downarrow$ pero error de estimación $\uparrow$.

Bias-Variance trade-off



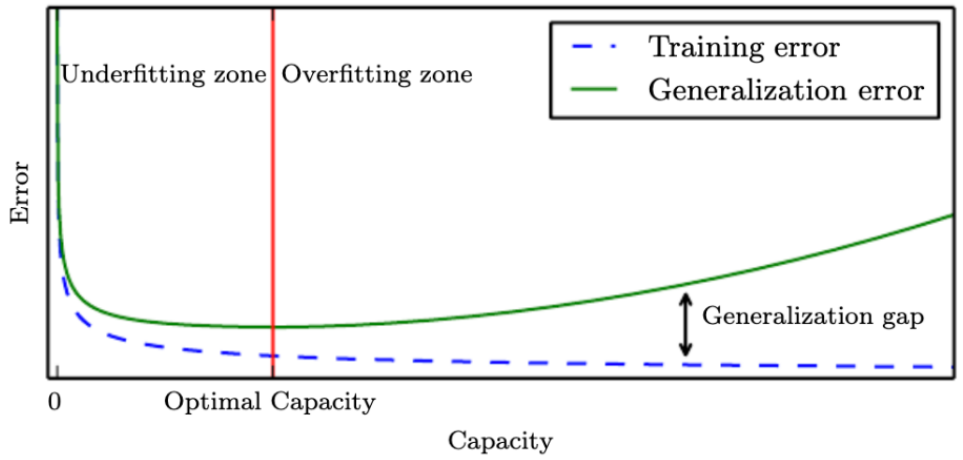
Bias-Variance trade-off

- $y = g(\mathbf{x}) + \varepsilon$, con $\mathbb{E}[\varepsilon] = 0$, $\mathrm{std}(\varepsilon) = \sigma$.
- Aprendemos un predictor $\hat{y} = f(\mathbf{x}; \mathcal{D})$ minimizando MSE en el conjunto de entrenamiento $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$.
- El error de generalización (riesgo cuadrático esperado) es:

$$
\begin{aligned}
\mathbb{E}_{\mathcal{D},\varepsilon}\big[(y - \hat{y})^2\big] &= \mathbb{E}_{\mathcal{D},\varepsilon}\big[(g(\mathbf{x}) - f(\mathbf{x};\mathcal{D}) + \varepsilon)^2\big] \\
&= \mathbb{E}_{\mathcal{D}}\big[(g(x) - f(\mathbf{x};\mathcal{D}))^2\big] + \mathbb{E}_{\varepsilon}[\varepsilon^2] \\
&= \mathbb{E}_{\mathcal{D}}\Big[\big(g(\mathbf{x}) - \mathbb{E}_{\mathcal{D}} f(\mathbf{x};\mathcal{D}) + \mathbb{E}_{\mathcal{D}} f(\mathbf{x};\mathcal{D}) - f(\mathbf{x};\mathcal{D})\big)^2\Big] + \sigma^2 \\
&= \underbrace{\big(g(\mathbf{x}) - \mathbb{E}_{\mathcal{D}} f(\mathbf{x};\mathcal{D})\big)^2}_{\text{bias}^2} + \underbrace{\mathbb{E}_{\mathcal{D}}\big[f(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}} f(\mathbf{x};\mathcal{D})\big]^2}_{\text{varianza}} + \underbrace{\sigma^2}_{\text{error irreducible}} .
\end{aligned}
$$

Bias-Variance trade-off

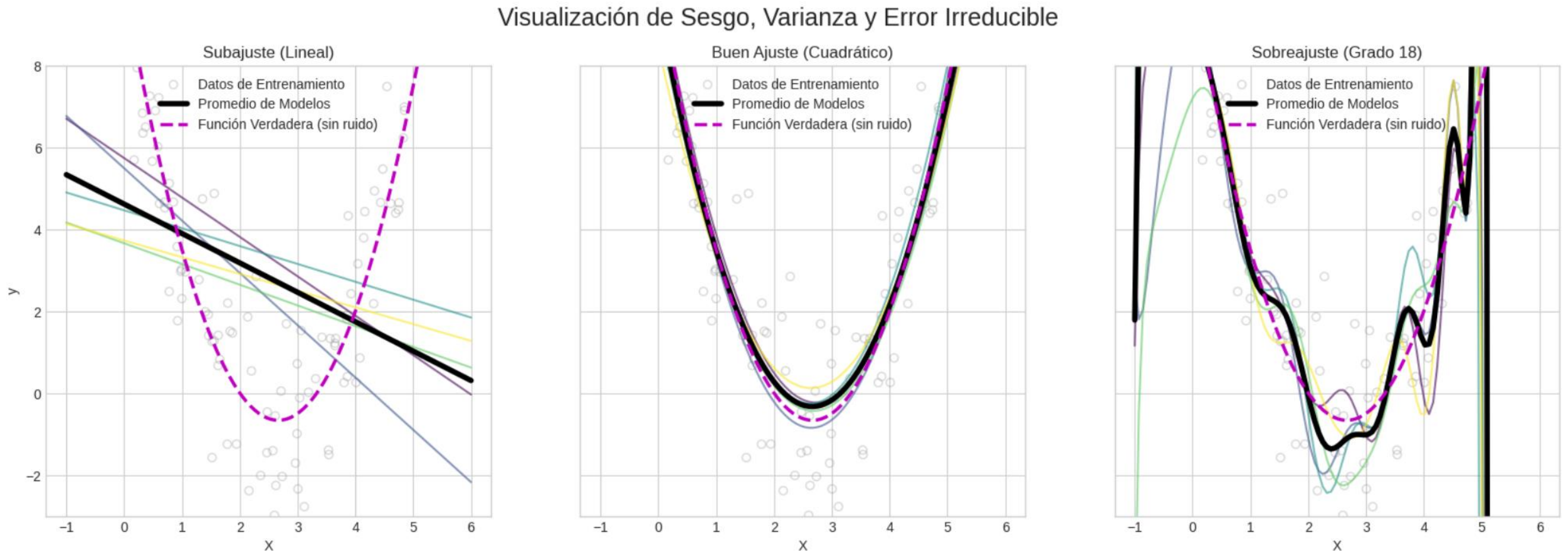
- ¿Cómo generalizar bien?
- Reducir el sesgo: pérdida de entrenamiento baja, un $\mathcal{F}$ rico.
- Reducir la varianza: brecha pequeña entre la pérdida de prueba y la de entrenamiento.
  - Conjunto de entrenamiento más grande.
  - Regularización: restringir a un subconjunto de $\mathcal{F}$.



Bias-Variance trade-off

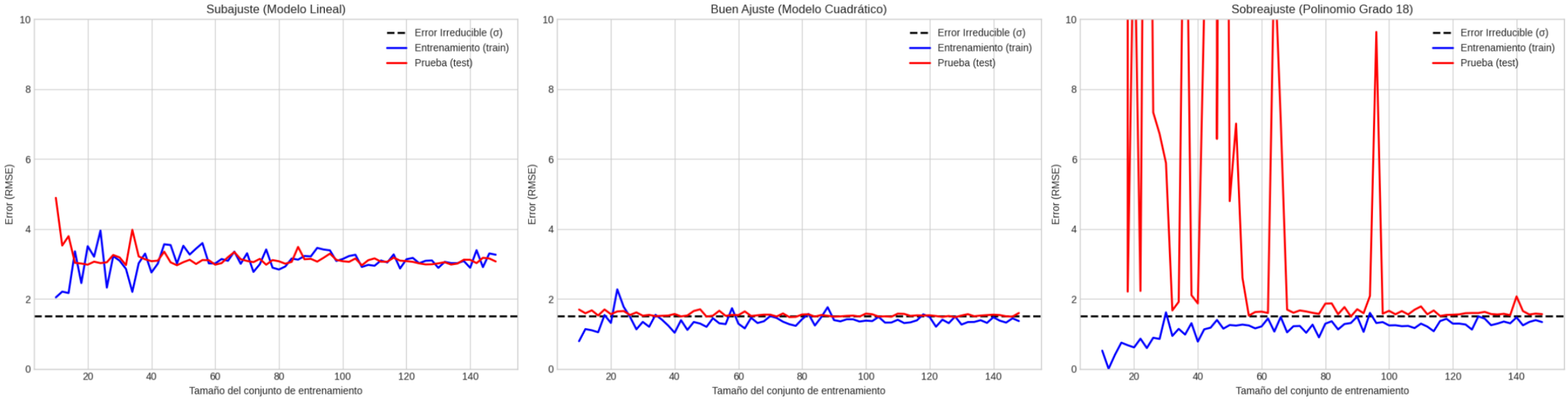
UNIVERSIDAD EAFIT
VIGILADA | MINEDUCACIÓN

Escuela de
Ciencias Aplicadas
e Ingeniería



Bias-Variance trade-off

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería



!Más datos mejoran el rendimiento del modelo si el modelo tiene la complejidad adecuada!

20 años de investigación en Learning Theory super-simplificados

Si tienes:

- Suficientes datos de entrenamiento $\mathcal{D}$ y
- $\mathcal{F}$ no es demasiado complejo

Entonces:

- probablemente podemos generalizar a datos de prueba no vistos.

Advertencias (Caveats):

- Varios resultados empíricos recientes (Zhang et. al) cuestionan nuestras intuiciones construidas a partir de esta clara separación.

Zhang et. al

Regularización

Principio: Combatir el sobreajuste (overfitting) penalizando la complejidad del modelo. La idea es que un modelo demasiado complejo tiene coeficientes ($\mathbf{w}$) con magnitudes muy grandes.

Formulación: Se añade un término de penalización ($R(\mathbf{w})$) a la función de error original ($E(\mathbf{w})$). El balance se controla con el hiperparámetro λ.

$$J(\mathbf{w}) = \underbrace{E(\mathbf{w})}_{\text{Error (Ajuste a datos)}} + \underbrace{\lambda}_{\text{Fuerza de Penalización}} \underbrace{R(\mathbf{w})}_{\text{Penalización por Complejidad}}$$

Regularización L2 (Weight Decay)

$$R(\theta) = ||\mathbf{w}||_2^2 = \sum_j w_j^2$$

Efecto: Fomenta pesos pequeños y distribuidos.

Regularización L1 (Lasso)

$$R(\theta) = ||\mathbf{w}||_1 = \sum_j |w_j|$$

Efecto: Fomenta la "escasez" (sparsity)

Regularización

1. Función de Costo

$$J(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^T\mathbf{w}$$

2. Cálculo de la Derivada

$$\begin{aligned}\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}}\Big(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} \\ &\qquad + \mathbf{y}^T\mathbf{y} + \lambda \mathbf{w}^T\mathbf{w}\Big) \\ &= 2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^T\mathbf{y} + 2\lambda\mathbf{w} \\ &= 2(\mathbf{X}^T\mathbf{X} + \lambda I)\mathbf{w} - 2\mathbf{X}^T\mathbf{y}\end{aligned}$$

3. Igualar a Cero y Despejar

Para encontrar el mínimo, igualamos la derivada a cero:

$$\begin{aligned}2(\mathbf{X}^T\mathbf{X} + \lambda I)\mathbf{w} - 2\mathbf{X}^T\mathbf{y} &= 0 \\ (\mathbf{X}^T\mathbf{X} + \lambda I)\mathbf{w} &= \mathbf{X}^T\mathbf{y}\end{aligned}$$

Solución Analítica Final:

$$\hat{\mathbf{w}}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{y}$$

Interpretación probabilísitca

El objetivo es encontrar los parámetros $\mathbf{w}$ más probables usando el Teorema de Bayes:

$$\underbrace{p(\mathbf{w}|\mathbf{y},\mathbf{X})}_{\text{Posterior}} \propto \underbrace{p(\mathbf{y}|\mathbf{X},\mathbf{w})}_{\text{Likelihood (Verosimilitud)}} \cdot \underbrace{p(\mathbf{w})}_{\text{Prior}}$$

- **Likelihood (Datos $\rightarrow$ MSE):**

$$\propto \exp\left(-\frac{(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})}{2\sigma_y^2}\right)$$

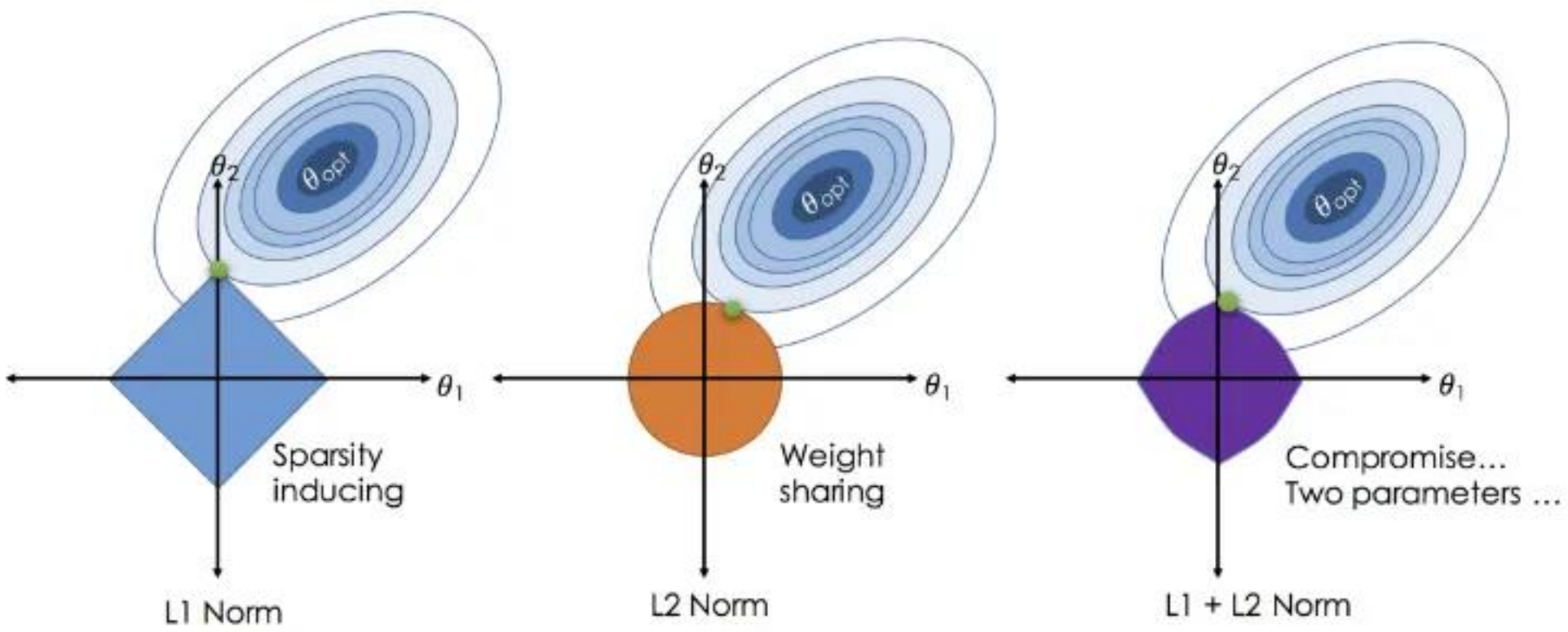
- **Prior (Creencia $\rightarrow$ L2):**

$$\propto \exp\left(-\frac{\mathbf{w}^T\mathbf{w}}{2\sigma_w^2}\right)$$

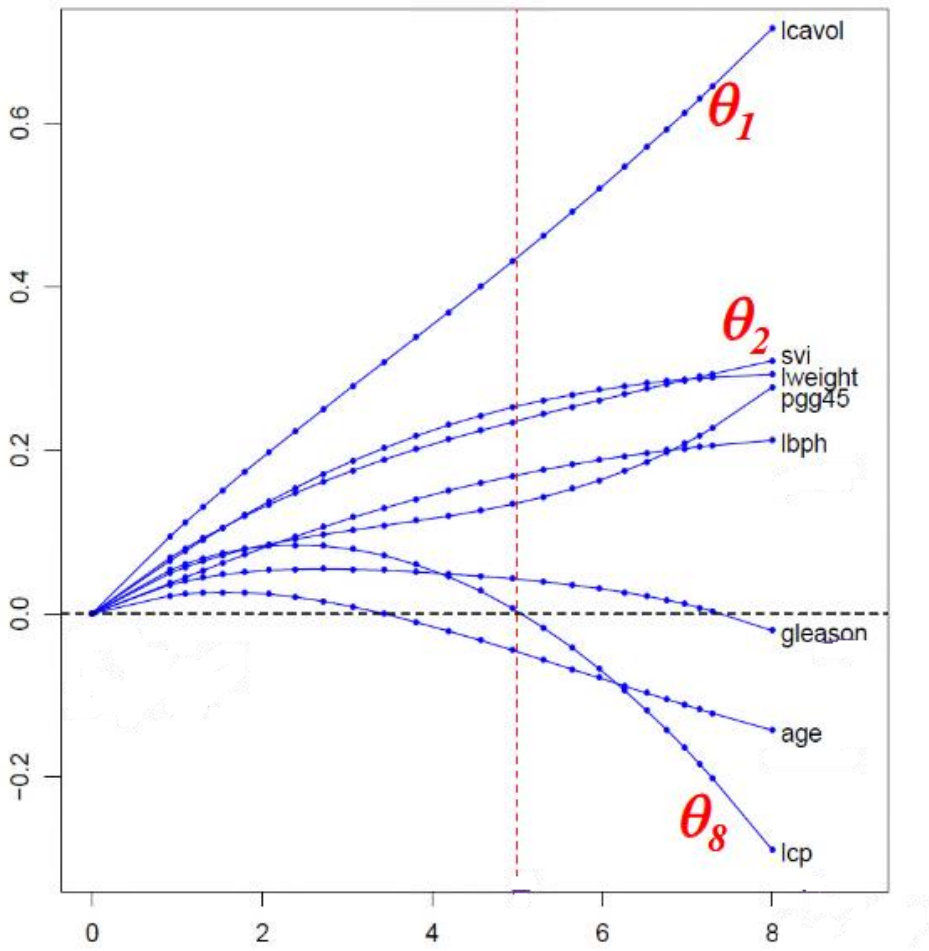
- **Resultado (MAP = Ridge):**

  Maximizar el posterior es equivalente a minimizar la suma del error (MSE) y la penalización L2.

Regularización



Regularización



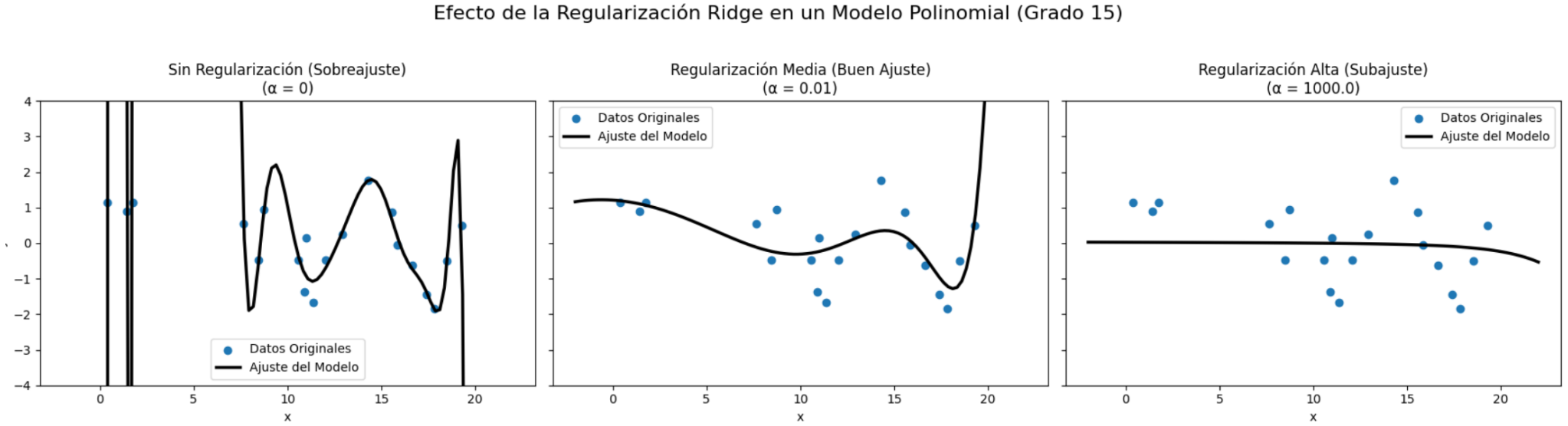
A medida que λ aumenta, para optimizer la función de costo los parámetros deben tender a 0

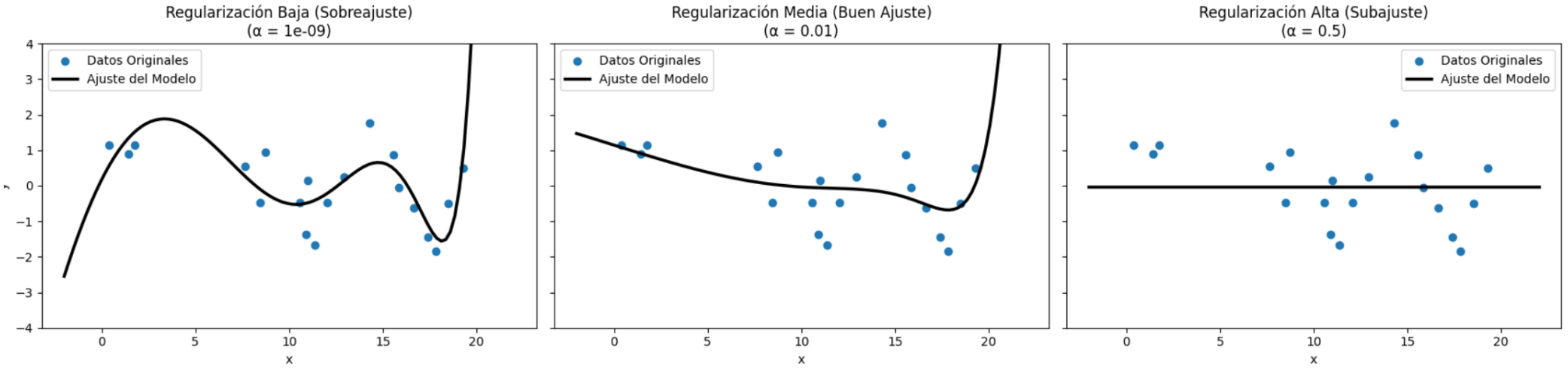
$$J(\mathbf{w}) = \underbrace{E(\mathbf{w})}_{\text{Error (Ajuste a datos)}} + \underbrace{\lambda}_{\text{Fuerza de Penalización}} \underbrace{R(\mathbf{w})}_{\text{Penalización por Complejidad}}$$

Regularización

UNIVERSIDAD EAFIT

Escuela de **Ciencias Aplicadas e Ingeniería**





Regresión con Kernels y RBFs

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- El vector de características $\phi(\mathbf{x})$ se construye evaluando la similitud de x con varios puntos centrales μ_j:

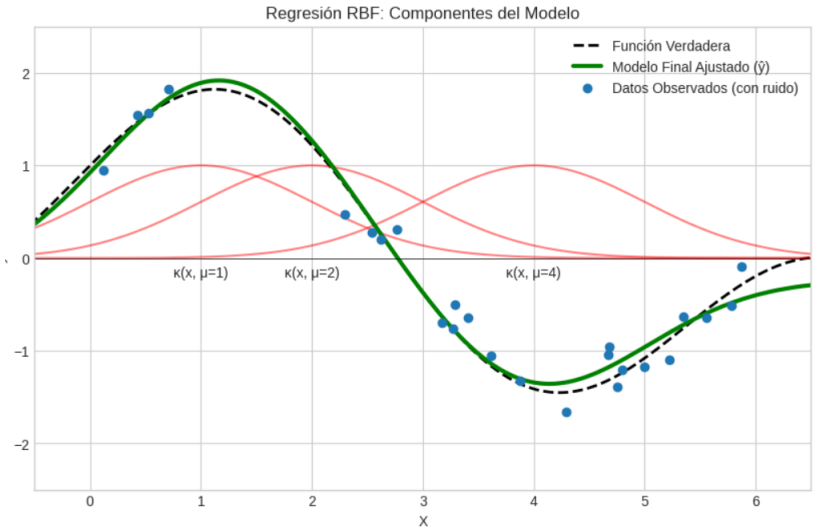
$$\phi(\mathbf{x}) = [\kappa(\mathbf{x}, \mu_1, \gamma), \ldots, \kappa(\mathbf{x}, \mu_d, \gamma)]$$

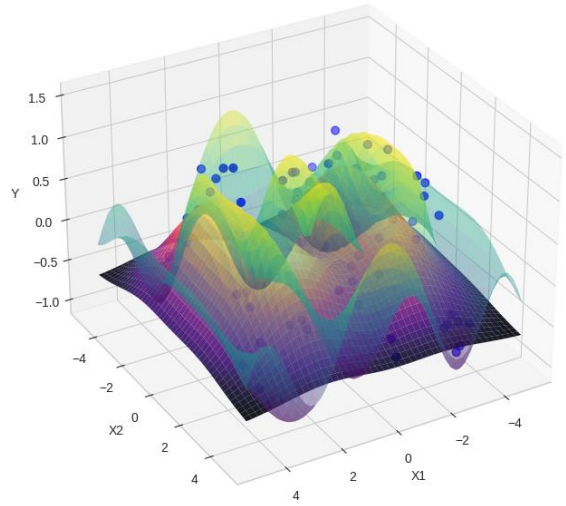
- Una RBF común es el kernel Gaussiano:

$$\kappa(\mathbf{x}, \mu_j, \gamma) = \exp\left(-\frac{1}{\gamma}||\mathbf{x} - \mu_j||^2\right)$$

- Construimos un **modelo lineal** sobre estas nuevas características:

$$\hat{y}(x) = \phi(\mathbf{x})^T \mathbf{w} = w_0 + \sum_{j=1}^{d} \kappa(\mathbf{x}, \mu_j, \gamma) w_j$$





Regresión con Kernels y RBFs

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- El vector de características $\phi(\mathbf{x})$ se construye evaluando la similitud de x con varios puntos centrales μ_j:

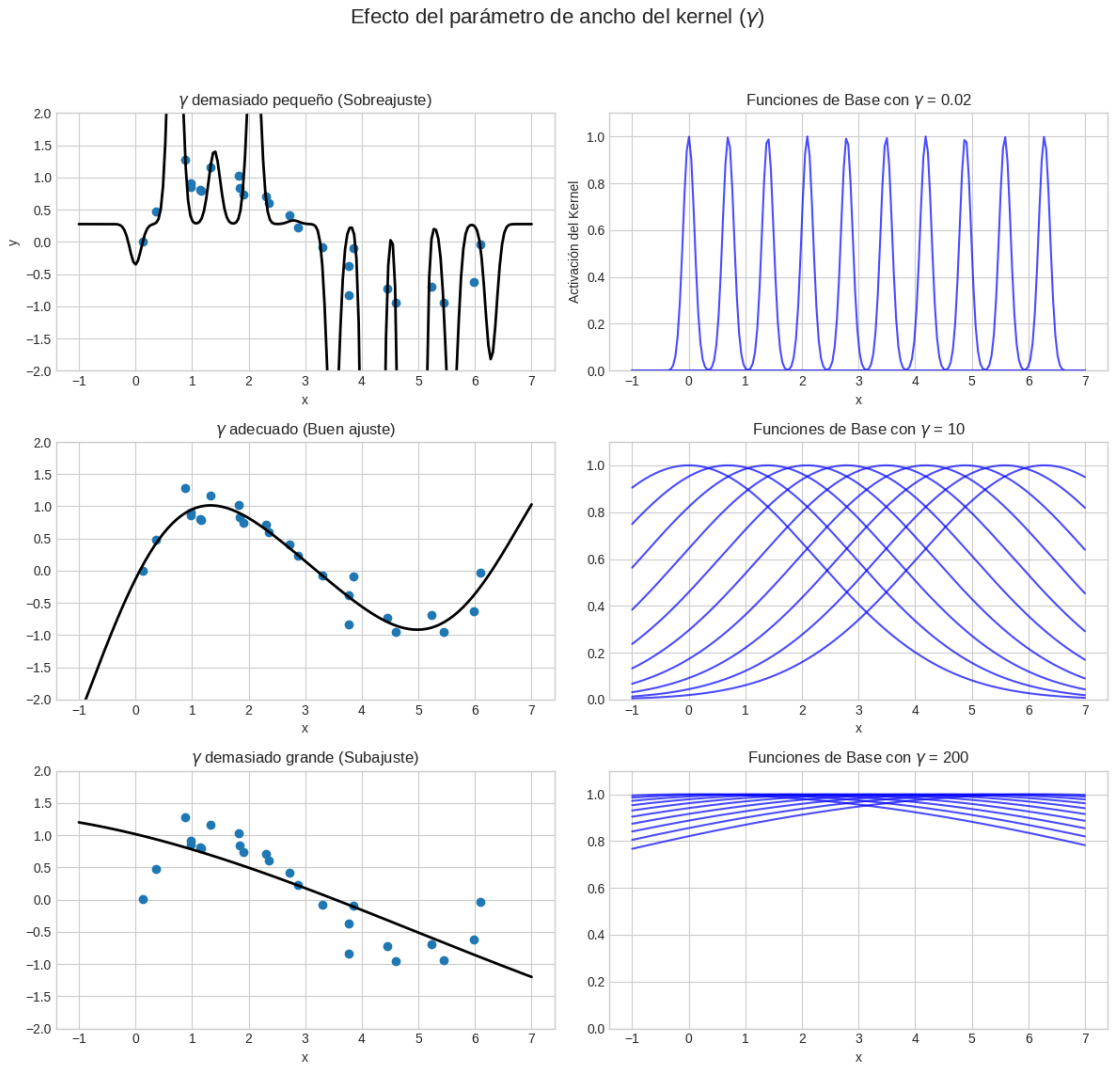
$$\phi(\mathbf{x}) = [\kappa(\mathbf{x}, \mu_1, \gamma), \ldots, \kappa(\mathbf{x}, \mu_d, \gamma)]$$

- Una RBF común es el kernel Gaussiano:

$$\kappa(\mathbf{x}, \mu_j, \gamma) = \exp\left(-\frac{1}{\gamma}||\mathbf{x} - \mu_j||^2\right)$$

- Construimos un **modelo lineal** sobre estas nuevas características:

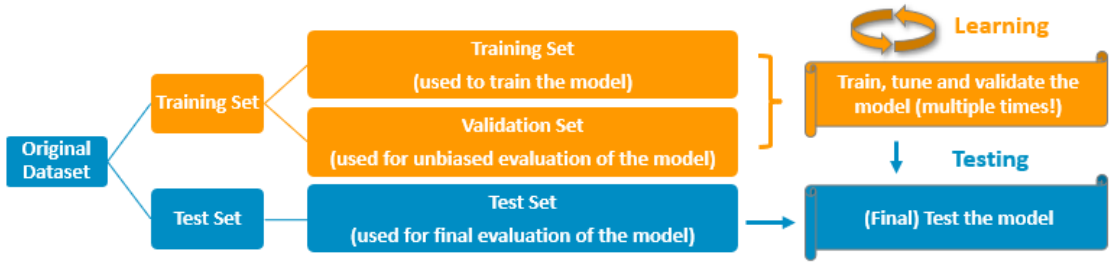
$$\hat{y}(x) = \phi(\mathbf{x})^T \mathbf{w} = w_0 + \sum_{j=1}^{d} \kappa(\mathbf{x}, \mu_j, \gamma) w_j$$



Sintonización de hiperparámetros

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

- ¿Cómo elegimos el coeficiente de regularización (λ), el ancho de los kernels (γ) o el grado del polinomio?



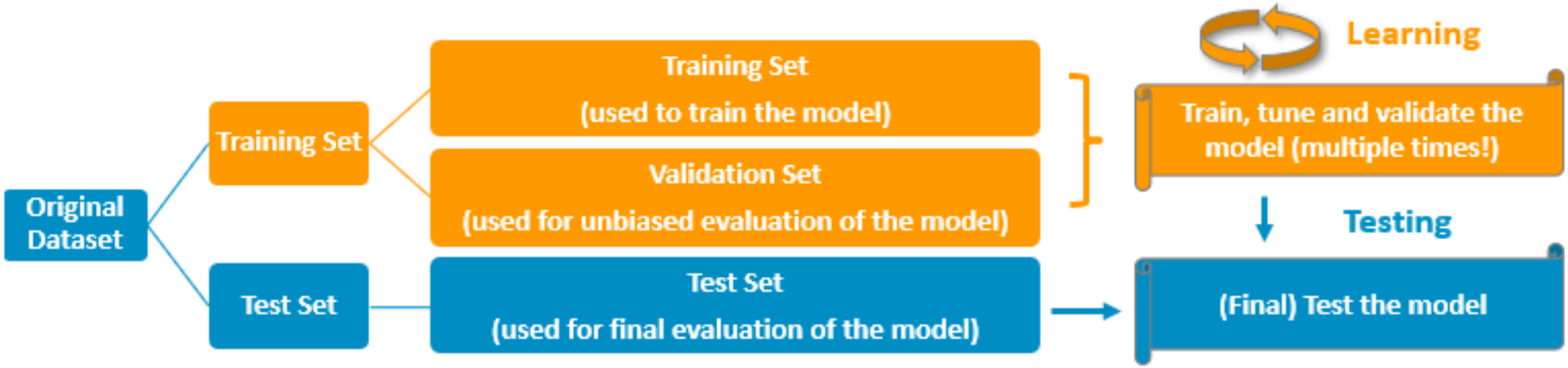
Hiperparámetros

- No se aprenden de los datos durante el entrenamiento.
- Son configuraciones del modelo que debemos elegir **antes** de entrenar.
- Controlan directamente el balance sesgo-varianza.

- **Entrenamiento (Train):** Para ajustar los parámetros del modelo ($\mathbf{w}$).
- **Validación (Validation):** Para evaluar y elegir los mejores hiperparámetros.
- **Prueba (Test):** Para medir el rendimiento final del modelo elegido (se usa **una sola vez**).

Sintonización de hiperparámetros

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

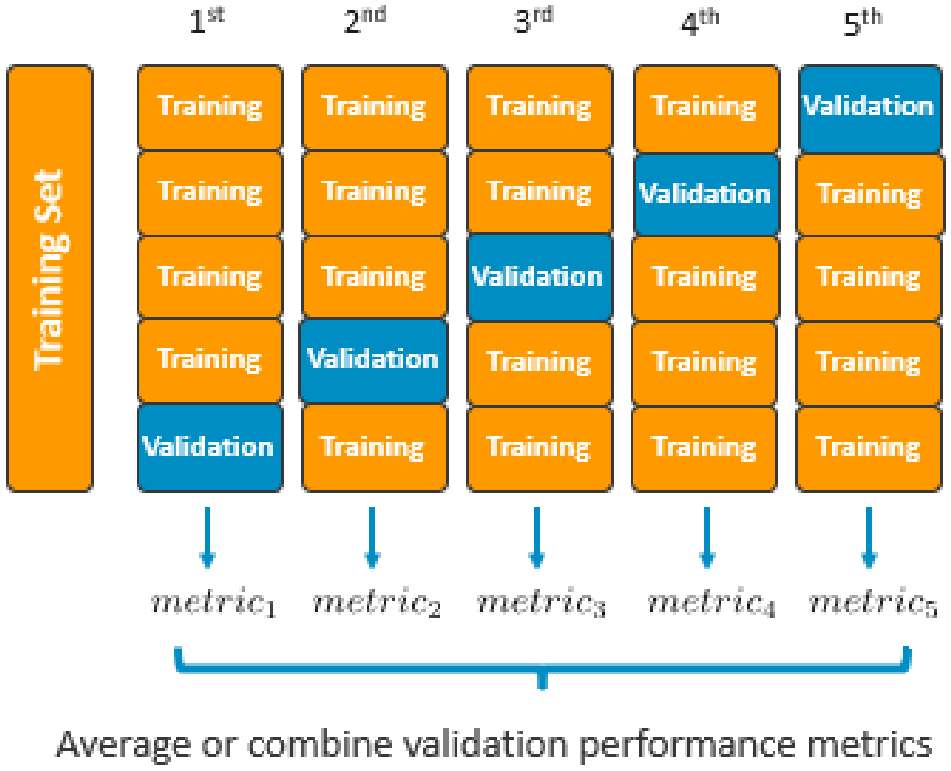


	bad_weather	is_rush_hour	mile_distance	urban_address	late
0	0.0	1.0	5.00	1.0	0.0
1	1.0	0.0	7.00	0.0	1.0
2	0.0	1.0	2.00	1.0	0.0
3	1.0	1.0	4.20	1.0	0.0
4	0.0	0.0	7.80	0.0	1.0
5	1.0	0.0	3.90	1.0	0.0
6	0.0	1.0	4.00	1.0	0.0
7	1.0	1.0	2.00	0.0	0.0
8	0.0	0.0	3.50	0.0	1.0
9	1.0	0.0	2.60	1.0	0.0
10	0.0	0.0	4.10	0.0	1.0

Generalmente se “barajan” los datos antes de hacer las particiones para evitar sesgos en los datos resultantes

Sintonización de hiperparámetros

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería



K-fold cross-validation: Es una técnica de validación para ver que tan bien generaliza un modelo a un conjunto de validación independiente.

Se utilizan **K muestras reservadas** para validar el modelo, cada vez entrenando con las muestras restantes:

- Dividir el conjunto de entrenamiento en K grupos (folds).
- Repetir K veces el siguiente procedimiento:
- Reservar el K^{th} fold para **validación**
- Entrenar el modelo en los folds restantes
- Calcular el desempeño en el fold de validación
- Combinar la métrica de rendimiento calculada