Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

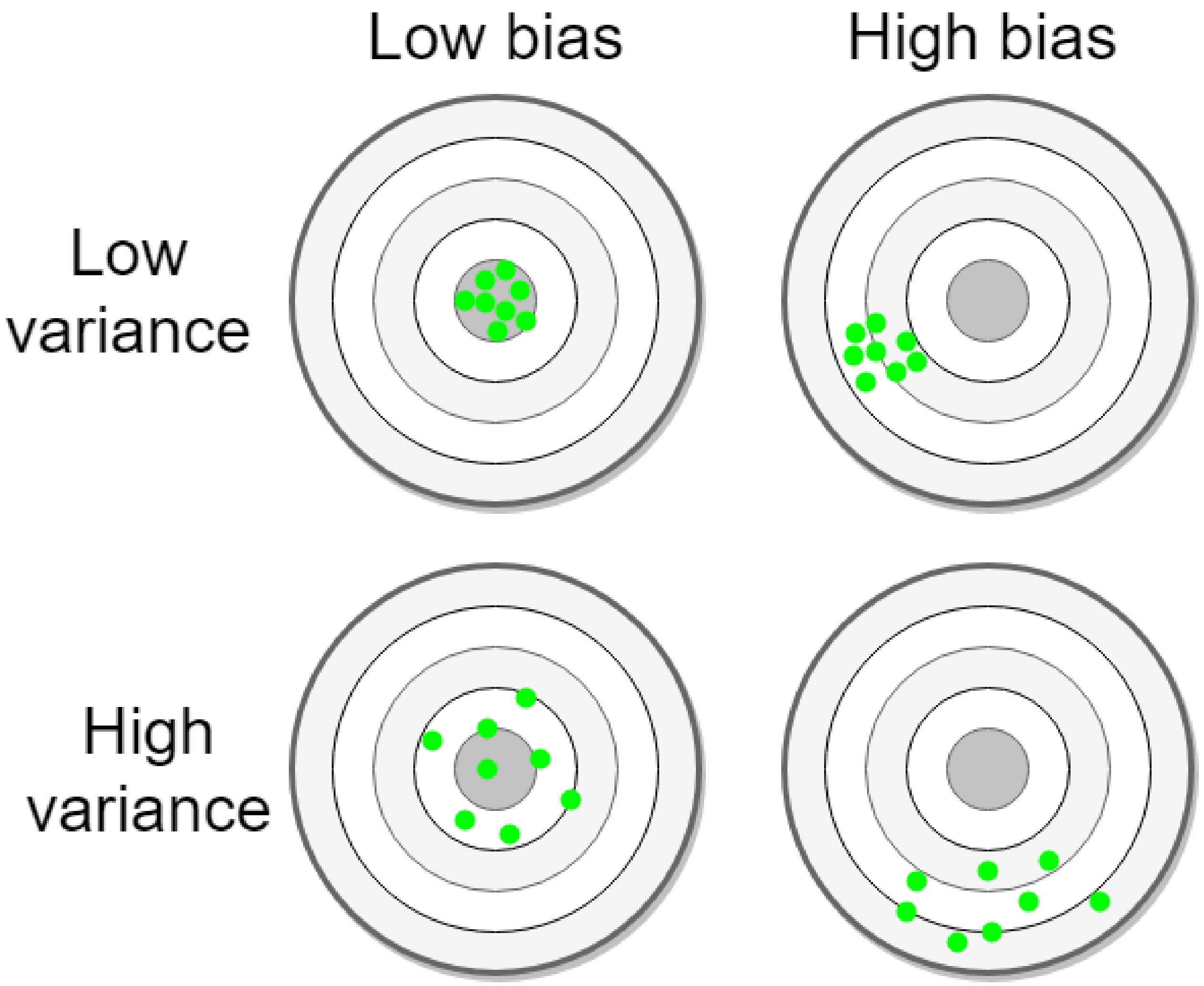
Sistemas Urbanos Inteligentes

Fundamentos de Machine Learning Parte 3

Hans Löbel
Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación

A pesar de ser clave, el set de entrenamiento no lo es todo

- En general, los algoritmos de aprendizaje viven y mueren por el set de entrenamiento.
- Lamentablemente, tener un buen set de entrenamiento, no asegura siempre tener buena generalización.
- Poder de representación del algoritmo de aprendizaje pasa a ser también un tema central.
- El porqué de esto está dado por un problema llamado Bias-Variance Tradeoff


Low bias
High bias
Low variance
High variance

BV tradeoff se da de forma natural en ML

$$Y = f(x) + \varepsilon, \varepsilon \sim N(0, \sigma^2)$$

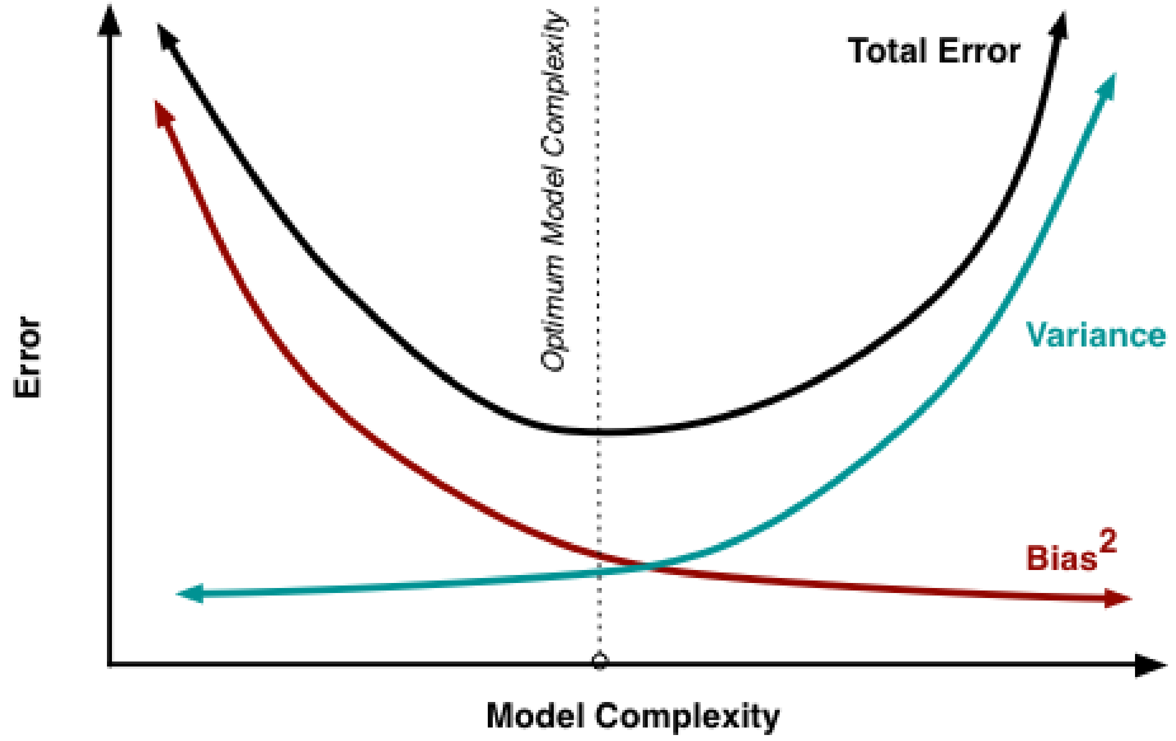
$$f(x) \approx \hat{f}(x)$$

$$Err(x) = E\left[\left(Y - \hat{f}(x)\right)^2\right]$$

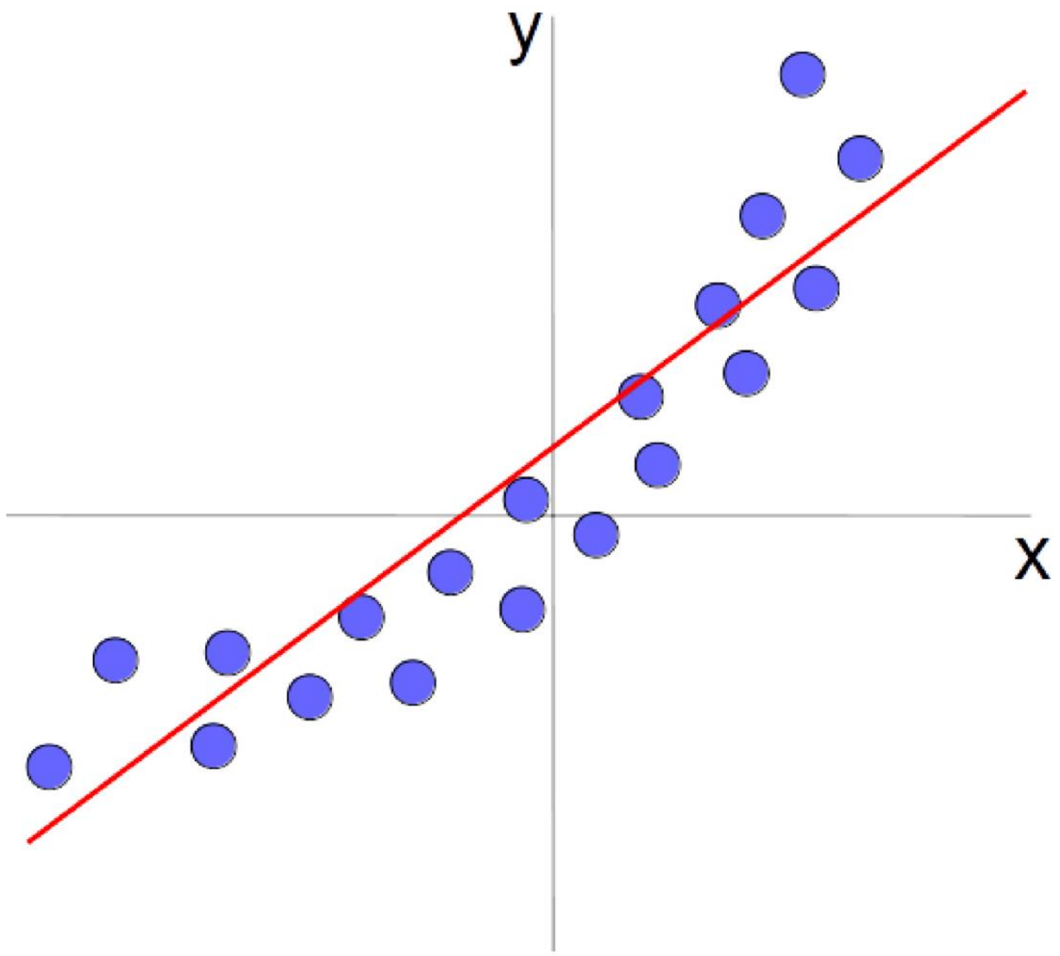
$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] + \sigma^2$$

$$Err(x) = Bias^2 + Varianza + Error\ irreducible$$

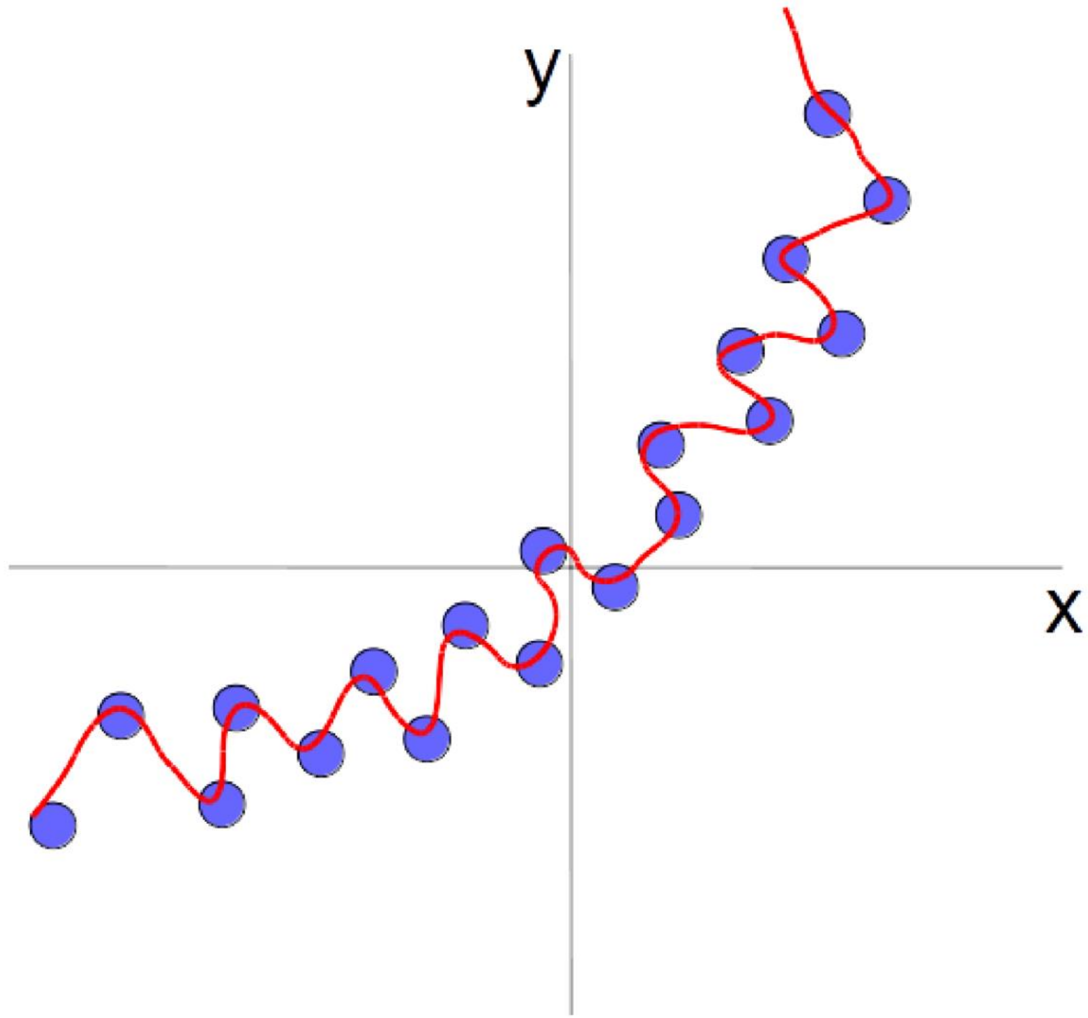
Complejidad del modelo es el parámetro que permite capturar el *BV tradeoff*



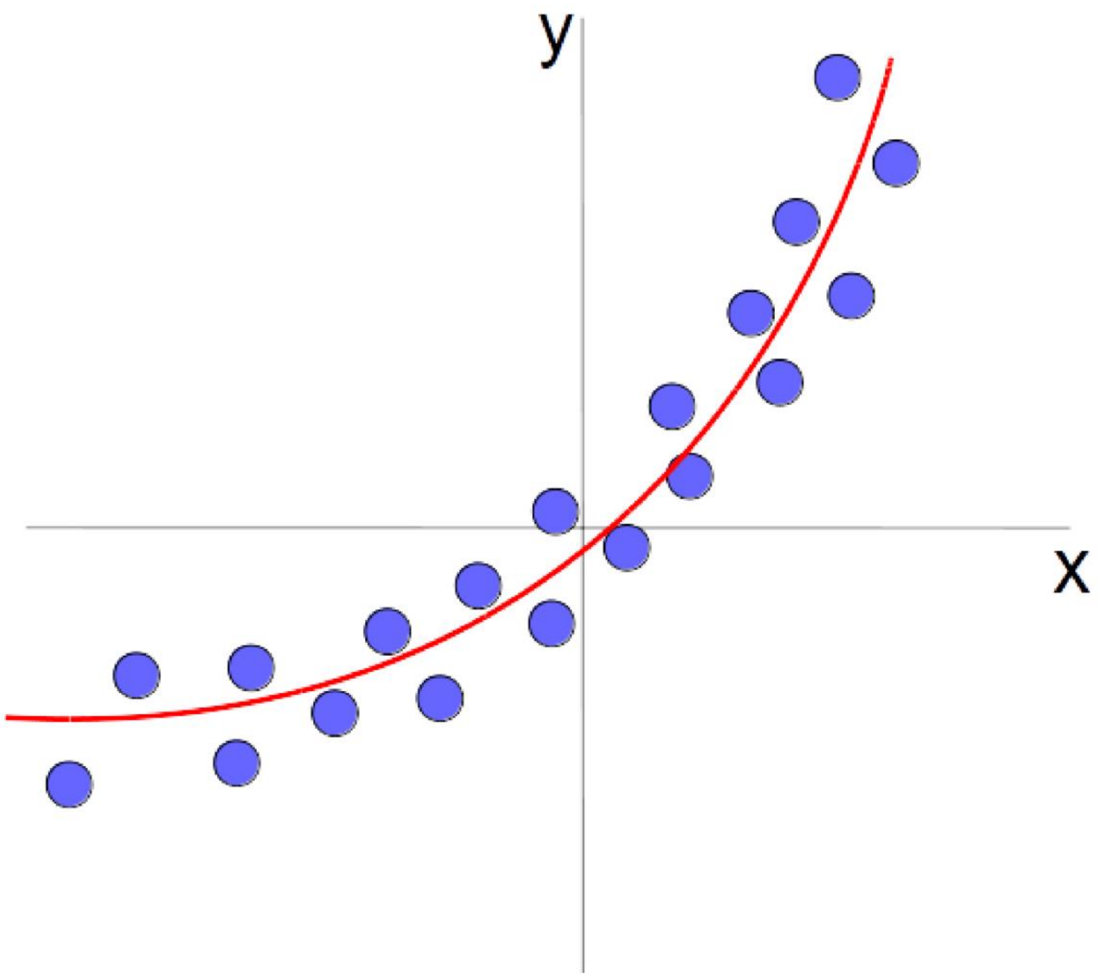
$$Err(x) = Bias^2 + Varianza + Error\ irreducible$$



Modelo es demasiado simple para capturar el comportamiento de los datos (*underfitting, alto sesgo*).



Modelo es muy complejo, y captura hasta el ruido presente en los ejemplos (*overfitting, alta varianza*).



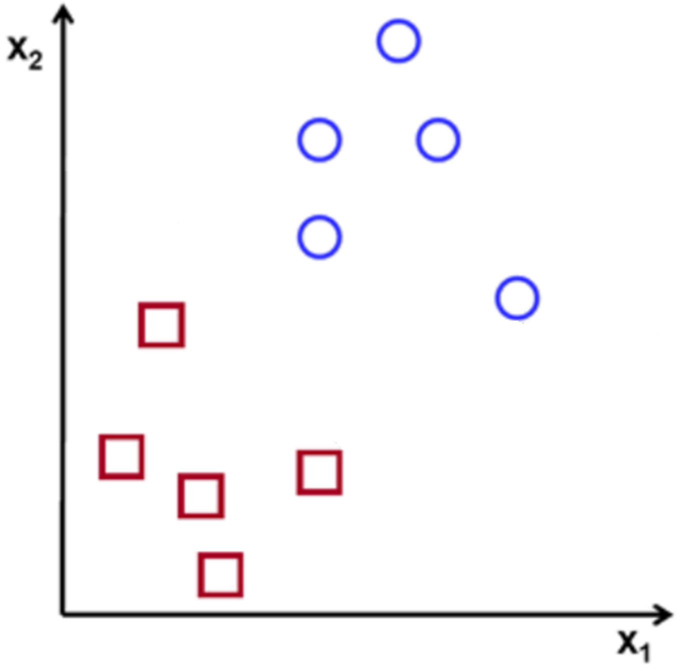
Modelo tiene la complejidad necesaria para capturar los patrones relevantes, controlando sesgo y varianza.

Vamos a Colab…

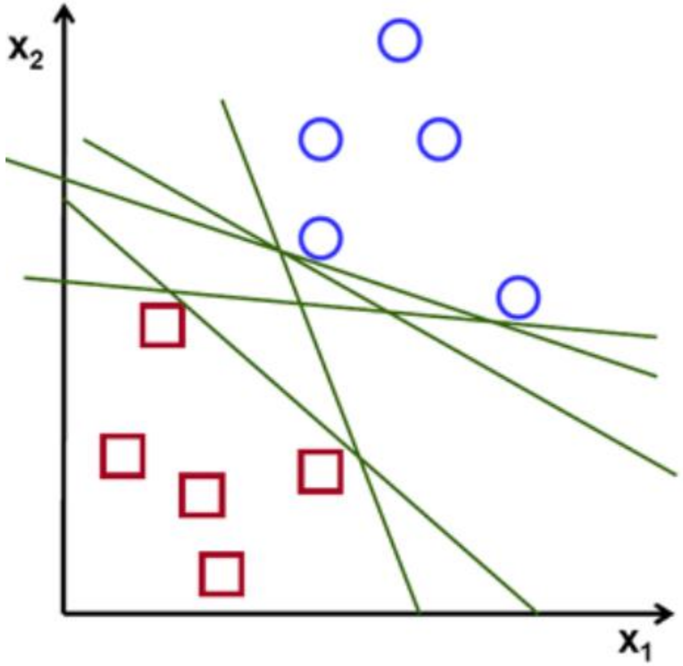
Para terminar, veremos un modelo más:

Support Vector Machine (SVM)

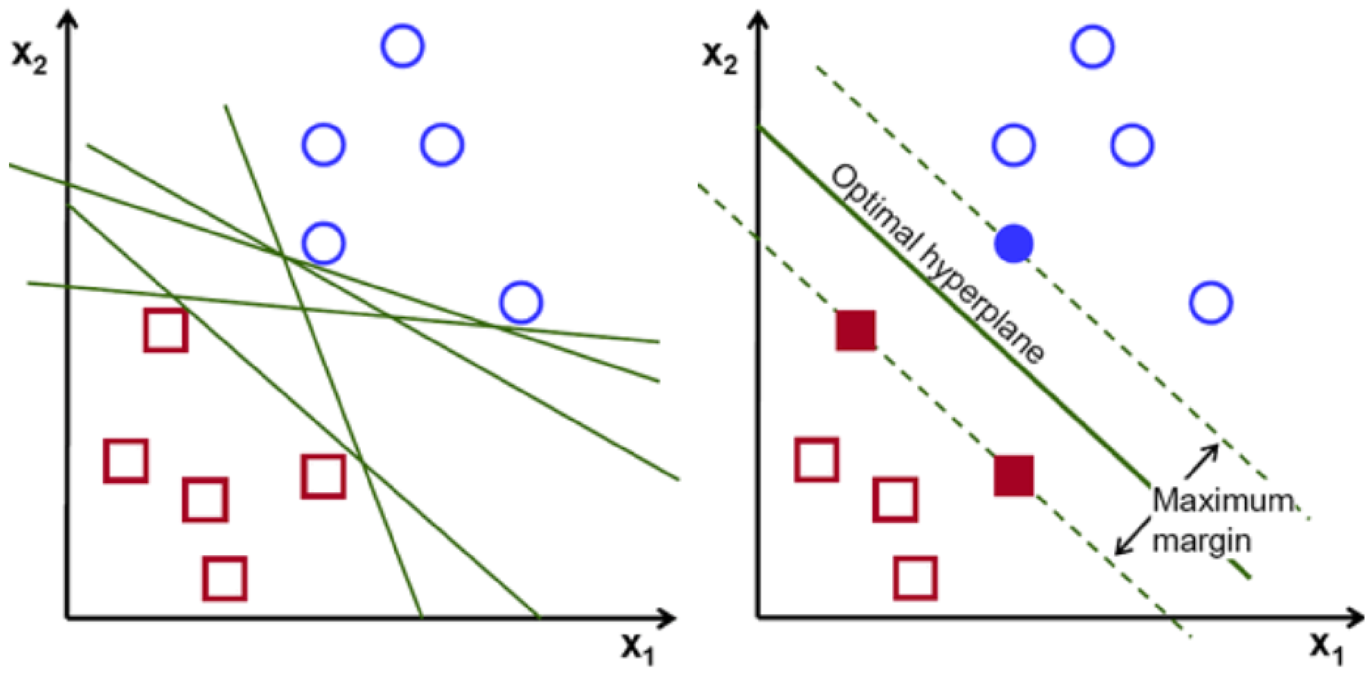
Visualicemos un problema binario de clasificación en 2D



¿Cuál es el (hiper)plano que mejor separa dos categorías?

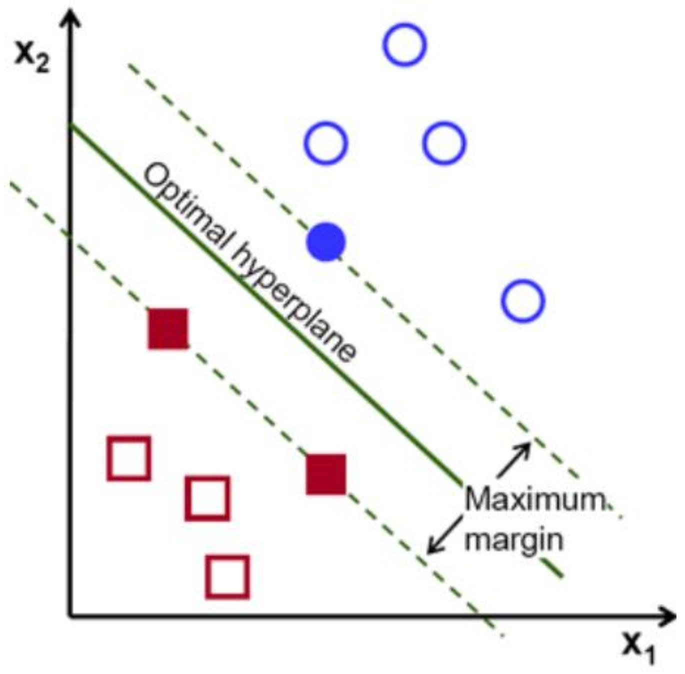


¿Cuál es el (hiper)plano que mejor separa dos categorías?

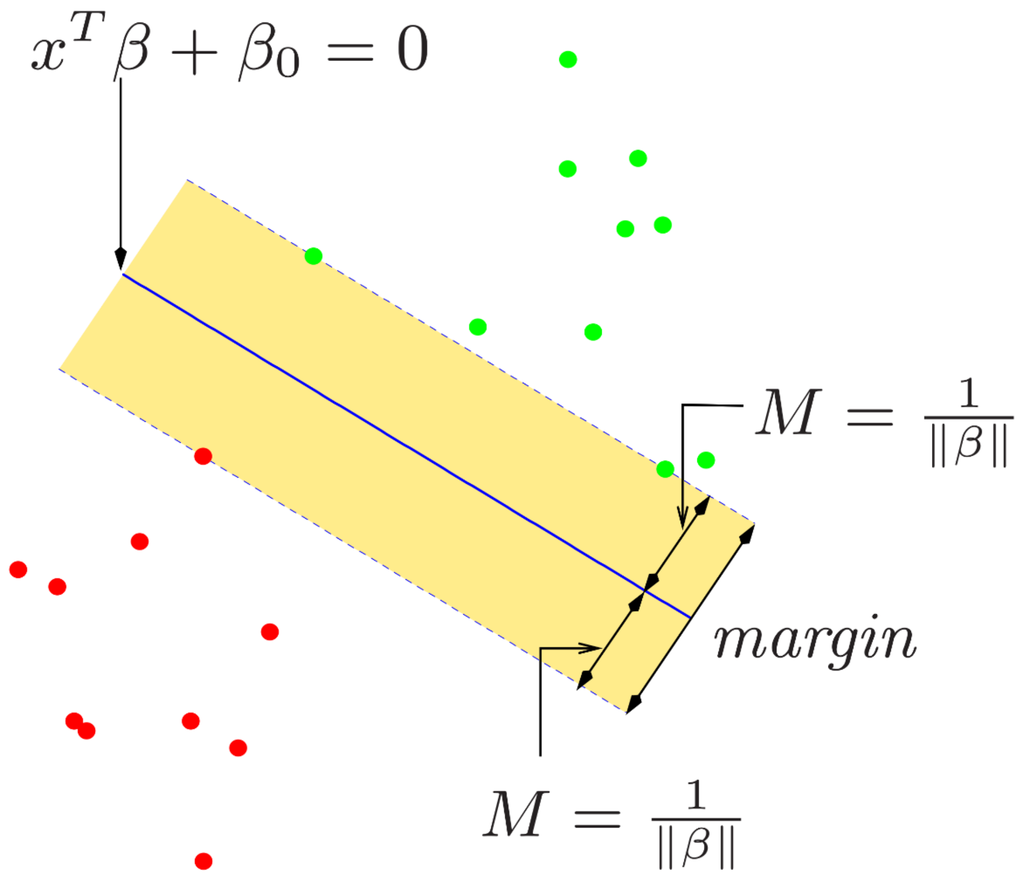


Para construir un SVM, se utiliza un enfoque basado en optimización

- Enfoque predominante en ML moderno.
- Busca plantear el aprendizaje como un problema de optimización, cuya solución óptima entrega los parámetros del clasificador/regresor.
- El caso que veremos a continuación, se formula como un problema de minimización cuadrático con restricciones lineales.
- En general, en el resto del curso nos centraremos en problemas no convexos sin restricciones.



¿Cómo podemos plantear esto como un problema de optimización?



$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2$$

$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1,\ i = 1, \ldots, N$$

Veamos como podemos resolver este problema

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2$$
$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1, \; i = 1, \ldots, N$$

Lagrangiano

$$L_P = \frac{1}{2}||\beta||^2 - \sum_{i=1}^{N} \alpha_i[y_i(x_i^T\beta + \beta_0) - 1]$$

Veamos como podemos resolver este problema

$$L_P = \frac{1}{2}||\beta||^2 - \sum_{i=1}^{N} \alpha_i [y_i(x_i^T \beta + \beta_0) - 1]$$

Derivando e igualando a cero, obtenemos:

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i \qquad\qquad 0 = \sum_{i=1}^{N} \alpha_i y_i$$

Sustituyendo todo esto en el lagrangiano, obtenemos el dual:

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k x_i^T x_k$$

$$\text{subject to } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0$$

Veamos como podemos resolver este problema

$$\max_{\alpha} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k x_i^T x_k$$

$$\text{subject to } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0$$

optimalidad KKT -> $\text{and } \alpha_i [y_i (x_i^T \beta + \beta_0) - 1] = 0 \; \forall i$

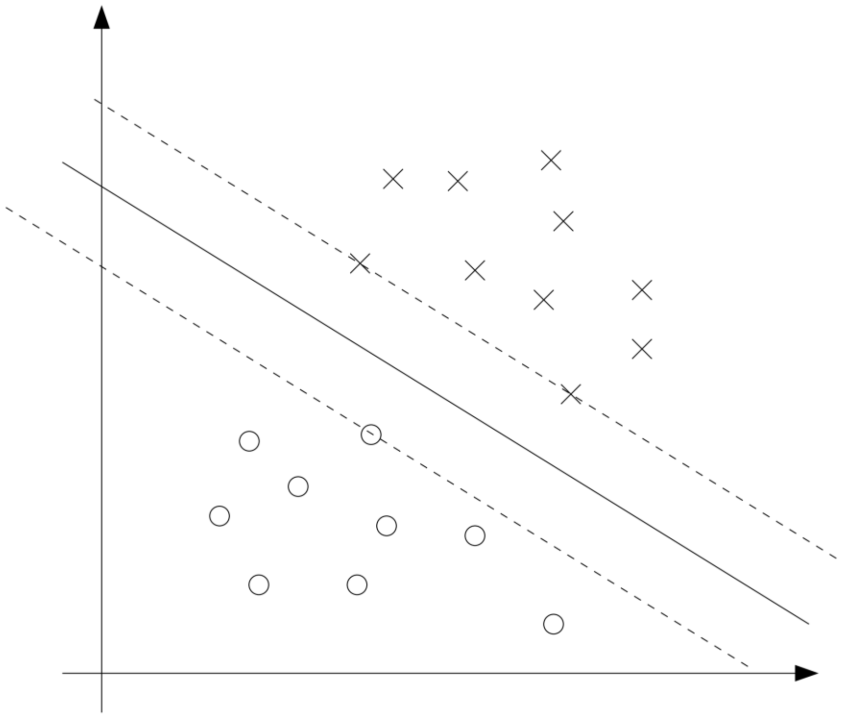
Esta última restricción (KKT) es fundamental para entender los SVM:

- Si $\alpha_i > 0$, $y_i (x_i^T \beta + \beta_0) = 1$ (el punto queda sobre el límite del margen)
- Si $y_i (x_i^T \beta + \beta_0) > 1$ (punto queda fuera del margen), $\alpha_i = 0$.

El problema dual, permite una interpretación más clara de los vectores de soporte.

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$\alpha_i [y_i (x_i^T \beta + \beta_0) - 1] = 0 \; \forall i$$



Más detalles sobre la formulación pueden encontrarse en “The Elements of Statistical Learning”, de Hastie, Tibshirani y Friedman, secciones 4.5.2, 12.2 y 12.3: https://web.stanford.edu/~hastie/ElemStatLearn/

Vamos a Colab…

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Sistemas Urbanos Inteligentes

Fundamentos de Machine Learning Parte 3

Hans Löbel
Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación