



UTEC Posgrado



MAESTRÍA

# Linear Regression

OLS, Bayesian regression. Learning theory.



# Modelado

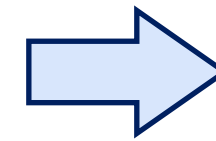
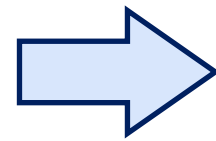
Modelo lineal



# Modeling



Mundo real



$$\vec{F} = \frac{\partial \vec{p}}{\partial t} = m \frac{\partial \vec{v}}{\partial t}$$

$$F = G \frac{m_1 m_2}{r^2}$$

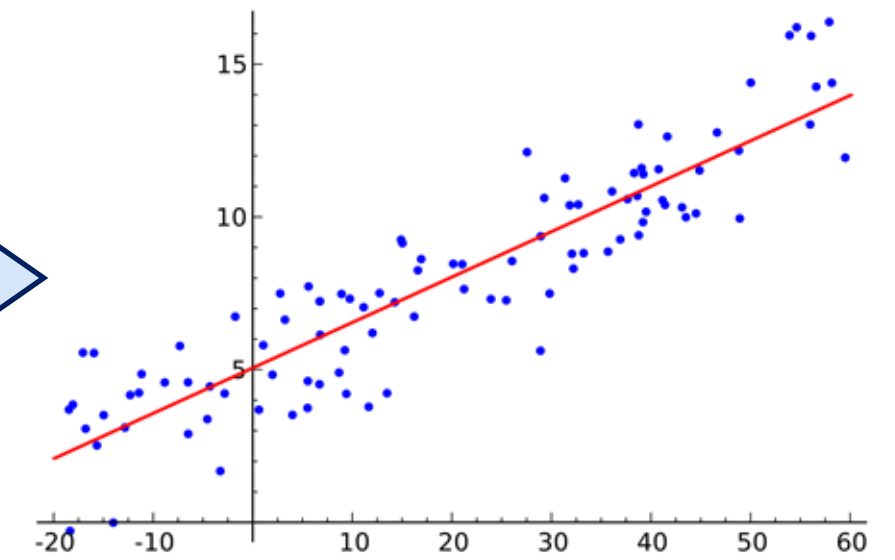
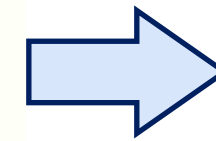
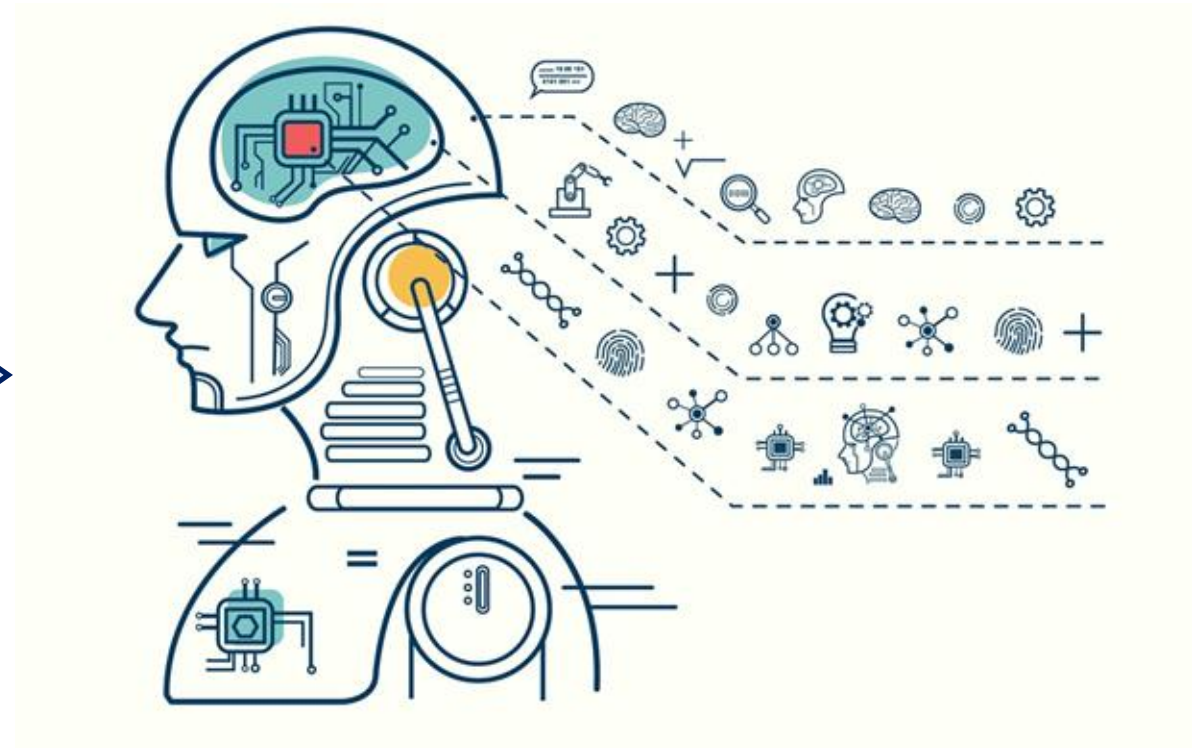
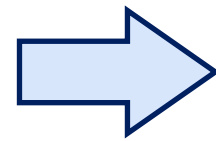
Leyes físicas



# Modeling



Mundo real



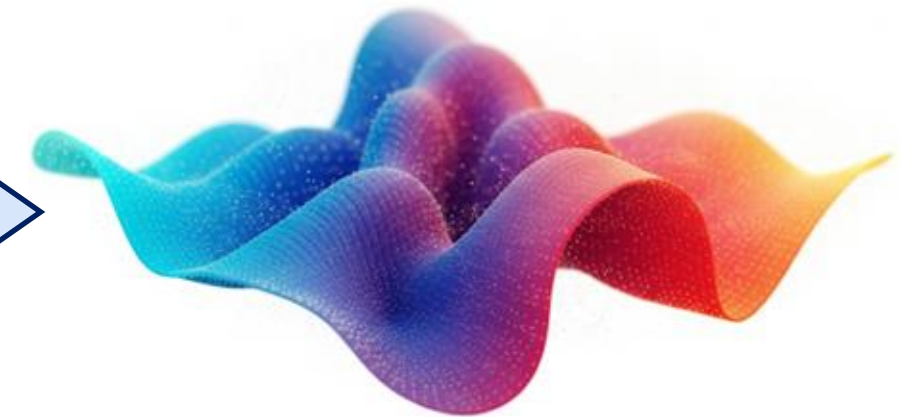
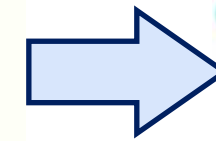
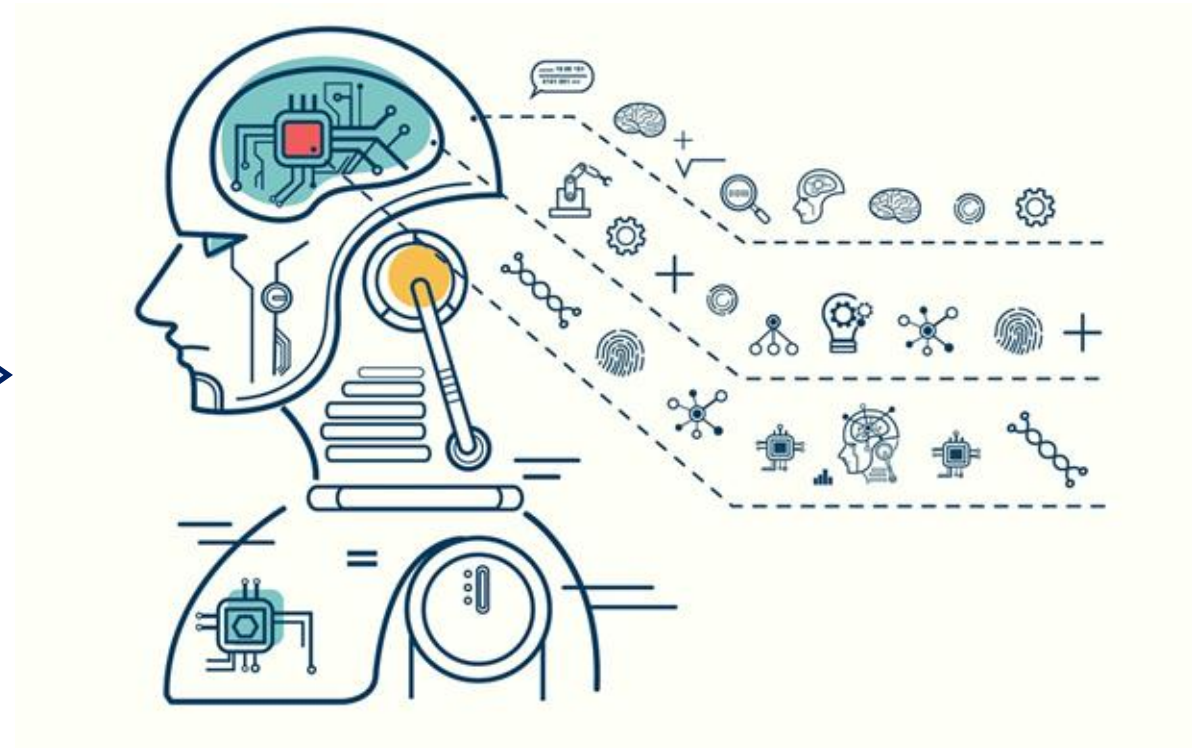
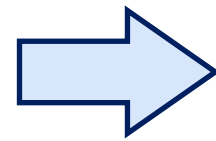
Predicciones



# Modeling



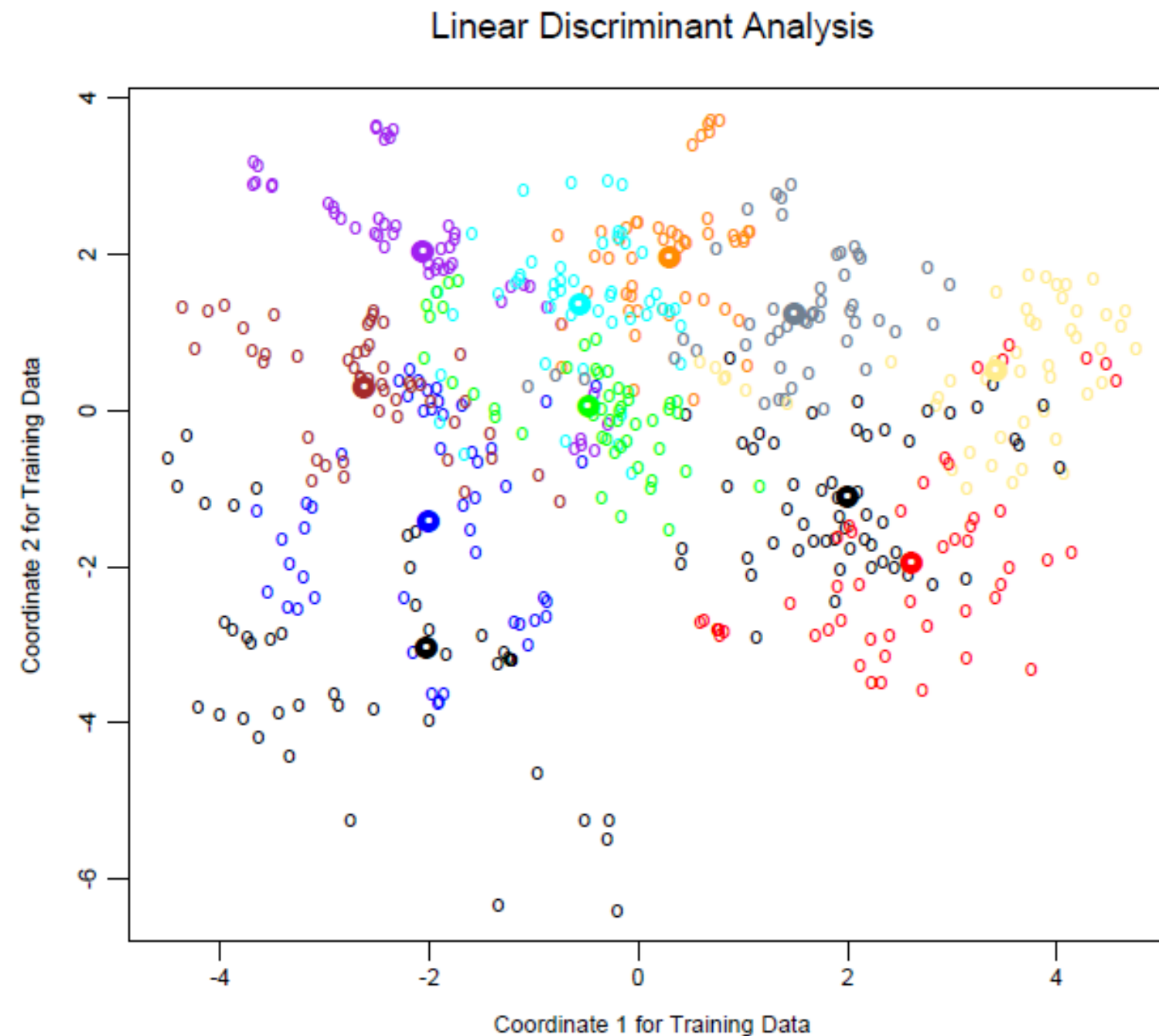
Mundo real



Representaciones



# Linear Discriminant Analysis



Suponiendo que  $f_k(x)$  es la clase de densidad condicional de  $X$  en la clase  $G = k$  y  $\pi_k$  la probabilidad prioritaria de la clase  $k$  con:

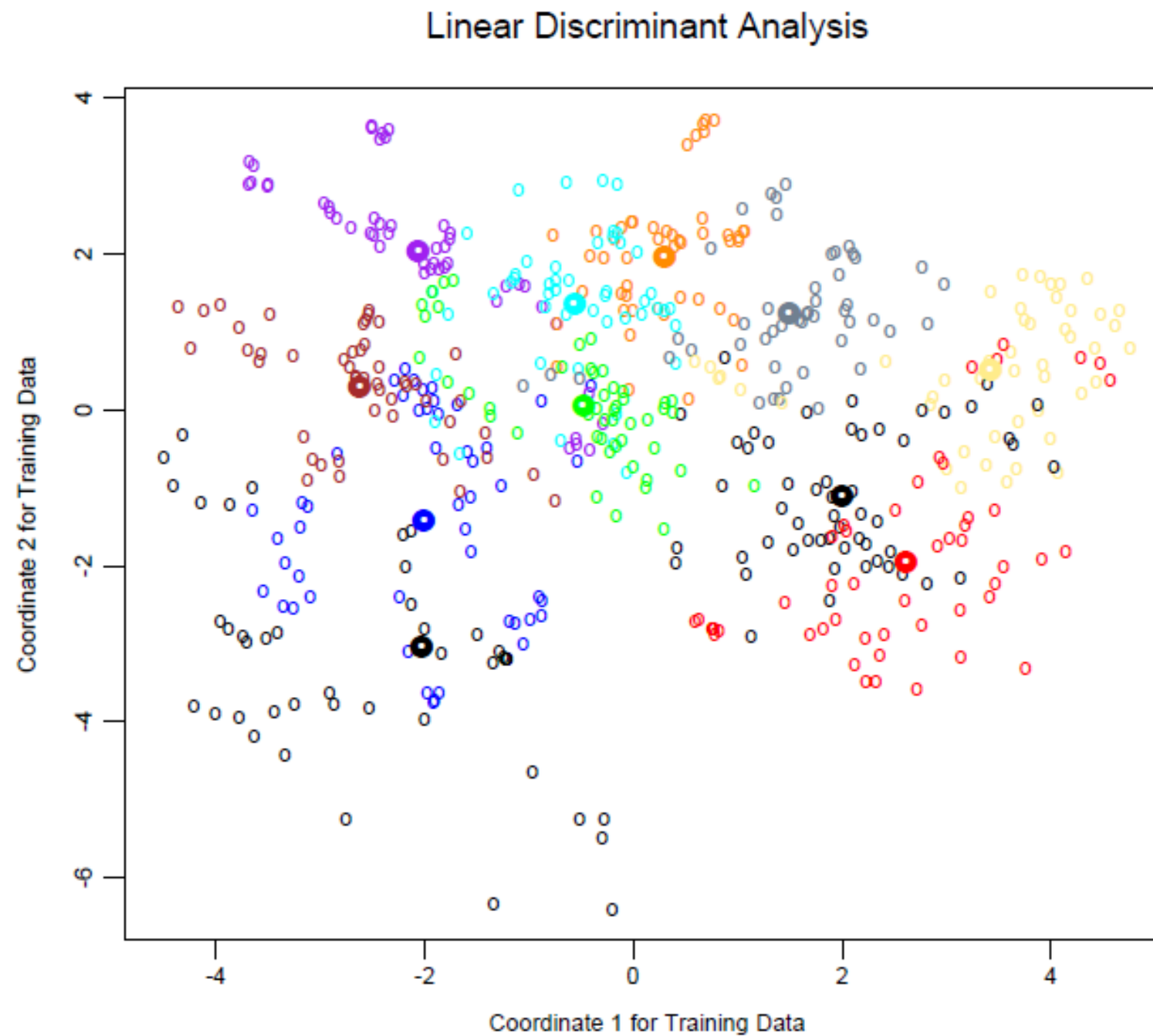
$$\sum_{k=1}^K \pi_k = 1.$$

Aplicando Bayes:

$$\Pr(G = k | X = x) = \frac{f_k(x) \pi_k}{\sum_{\ell=1}^K f_{\ell}(x) \pi_{\ell}}.$$



# Linear Discriminant Analysis

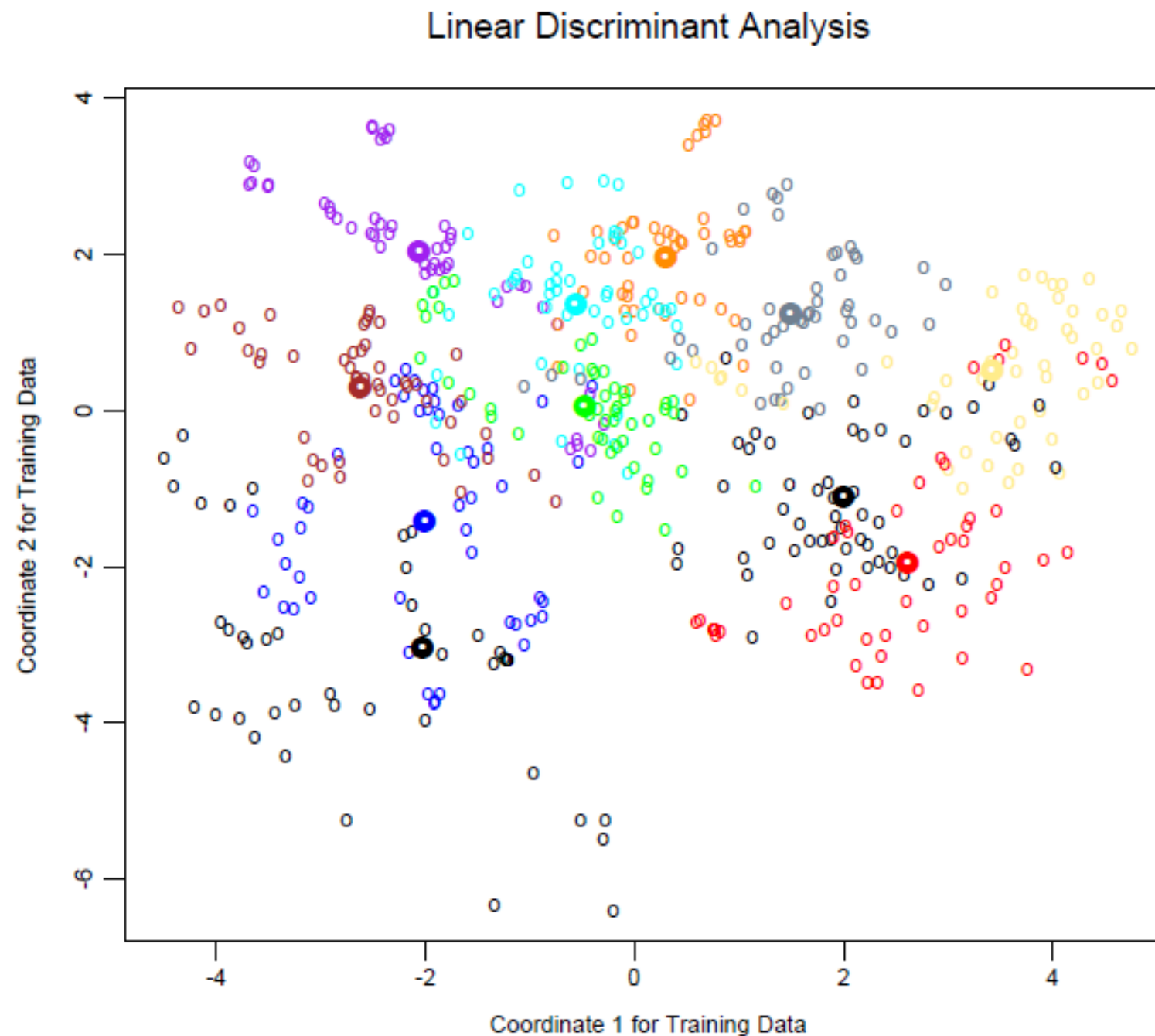


Suponiendo que modelamos cada densidad de clase como Gaussian multivariante:

$$f_k(x) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)}.$$



# Linear Discriminant Analysis



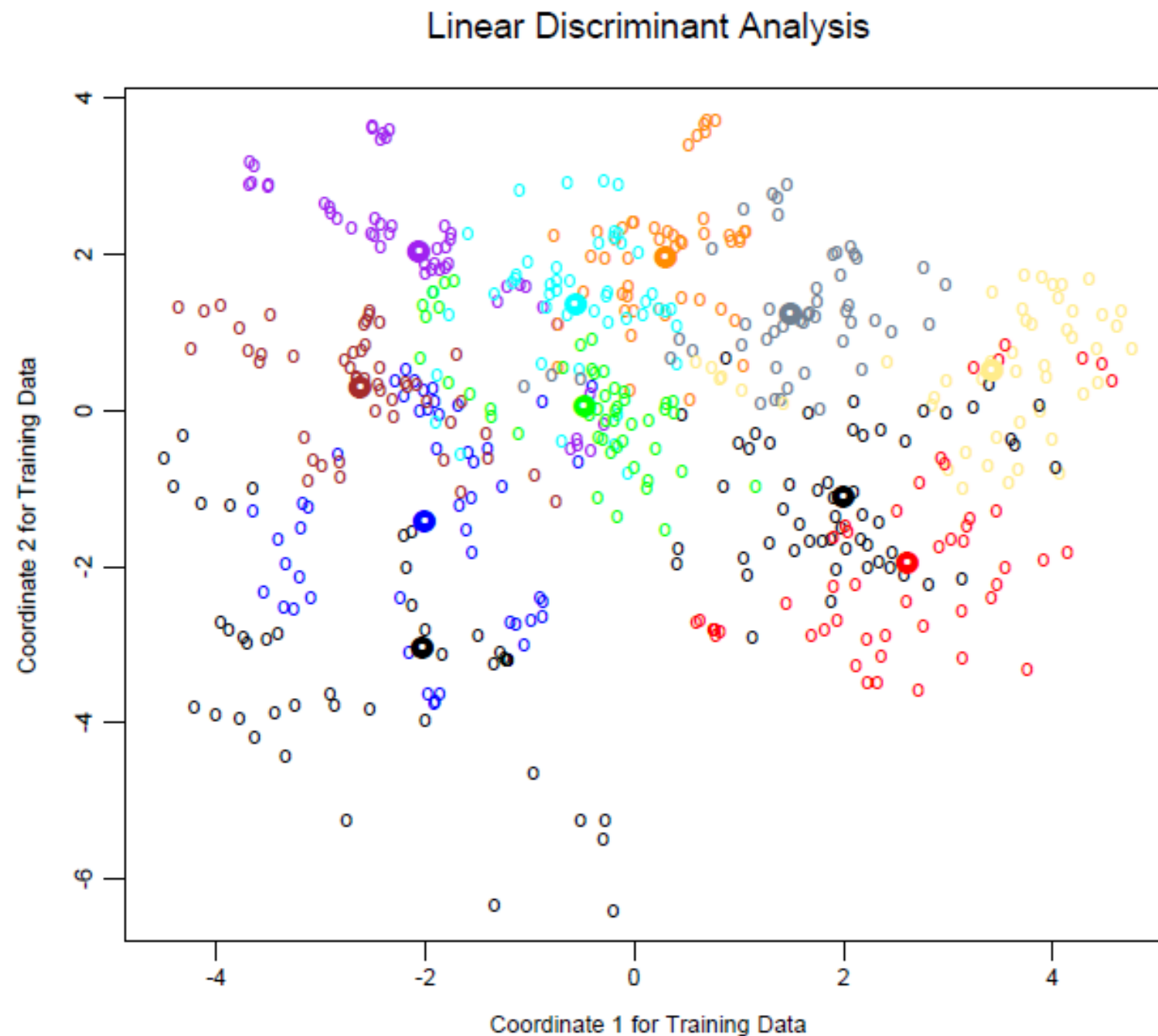
Suponiendo que modelamos cada densidad de clase como Gaussian multivariante:

$$f_k(x) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)}.$$

El análisis discriminante lineal (LDA) surge en el caso especial en el que suponemos que las clases tienen una matriz de covarianza común. Al comparar dos clases  $k$  y  $\ell$ , basta con observar la relación logarítmica, y vemos que:



# Linear Discriminant Analysis



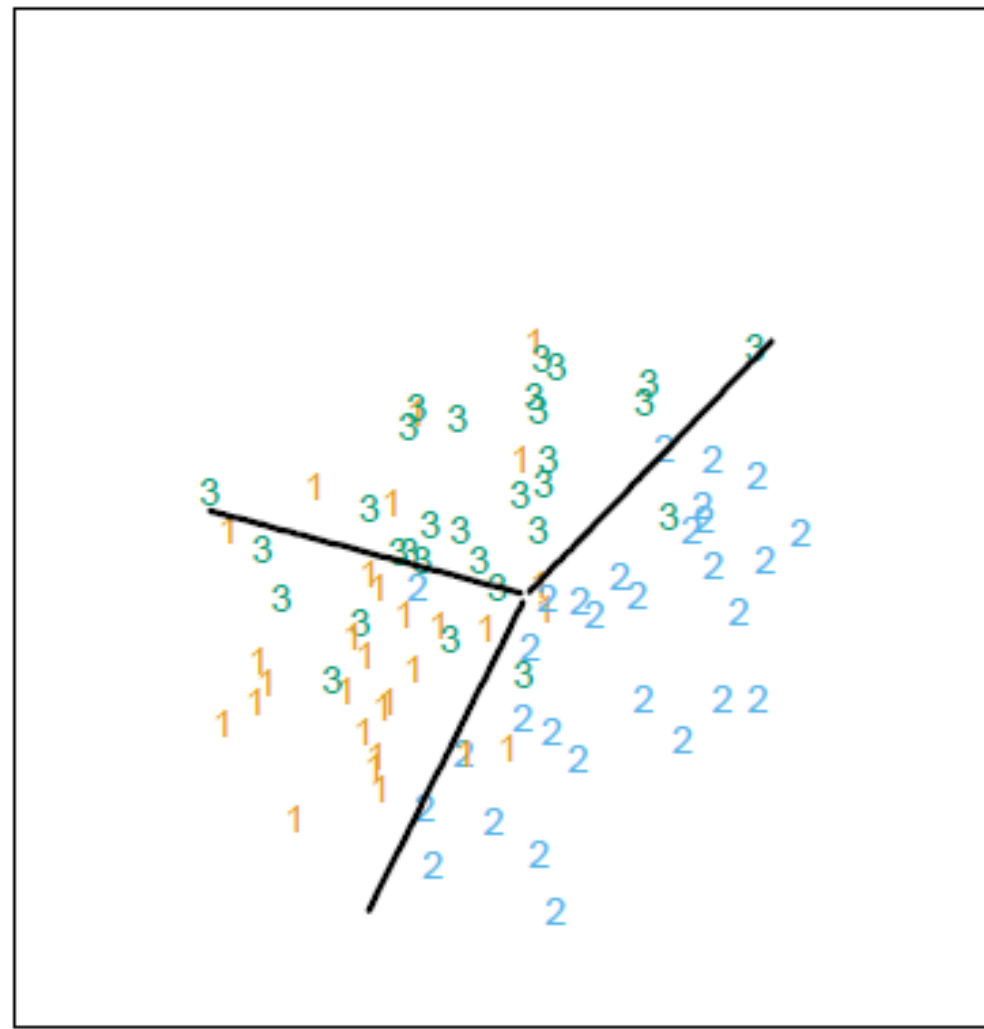
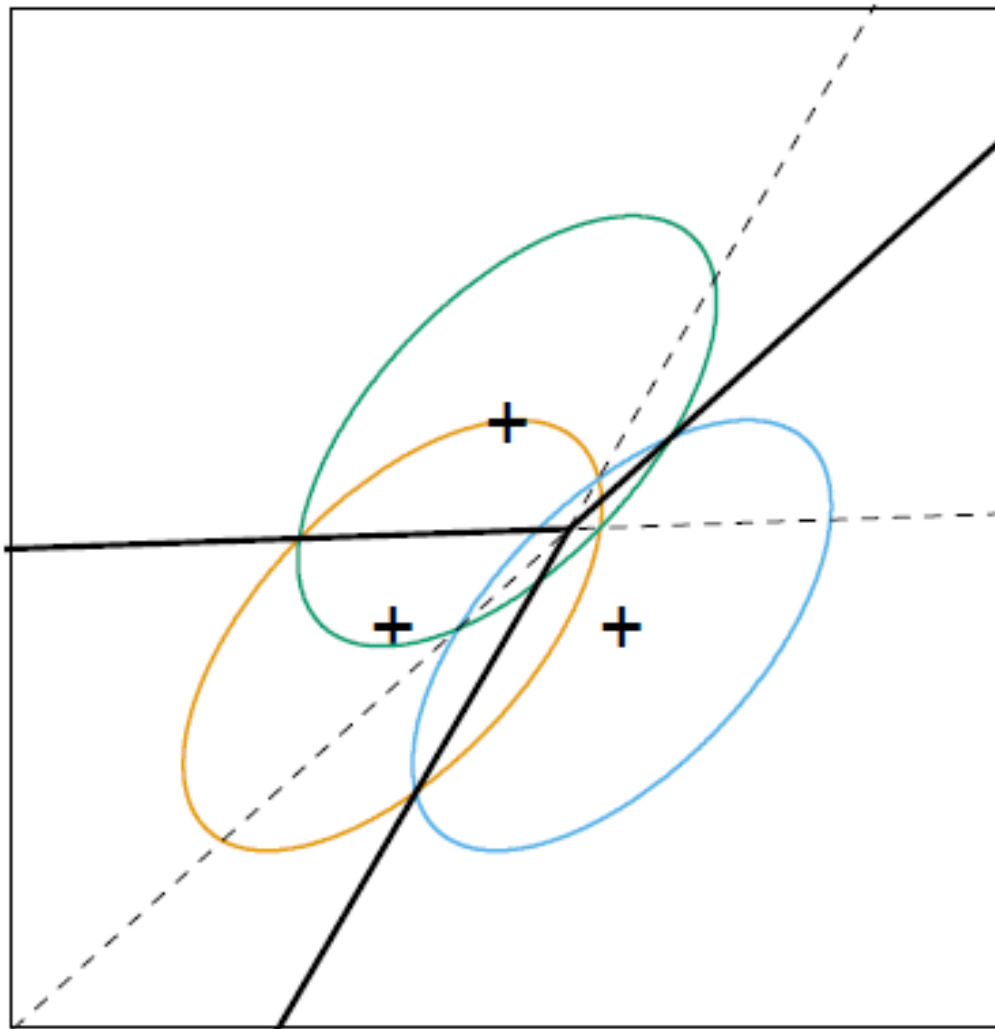
Suponiendo que modelamos cada densidad de clase como Gaussian multivariante:

$$f_k(x) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)}.$$

El análisis discriminante lineal (LDA) surge en el caso especial en el que suponemos que las clases tienen una matriz de covarianza común. Al comparar dos clases  $k$  y  $\ell$ , basta con observar la relación logarítmica, y vemos que:

$$\begin{aligned} \log \frac{\Pr(G = k|X = x)}{\Pr(G = \ell|X = x)} &= \log \frac{f_k(x)}{f_\ell(x)} + \log \frac{\pi_k}{\pi_\ell} \\ &= \log \frac{\pi_k}{\pi_\ell} - \frac{1}{2}(\mu_k + \mu_\ell)^T \Sigma^{-1} (\mu_k - \mu_\ell) \\ &\quad + x^T \Sigma^{-1} (\mu_k - \mu_\ell), \end{aligned}$$

# Linear Discriminant Analysis



Tres Distribuciones Gaussianas (izquierda) con la misma covarianza y diferentes medias (contorno de densidad 95% de probabilidad en cada caso), límites de decisión (líneas rectas continuas).

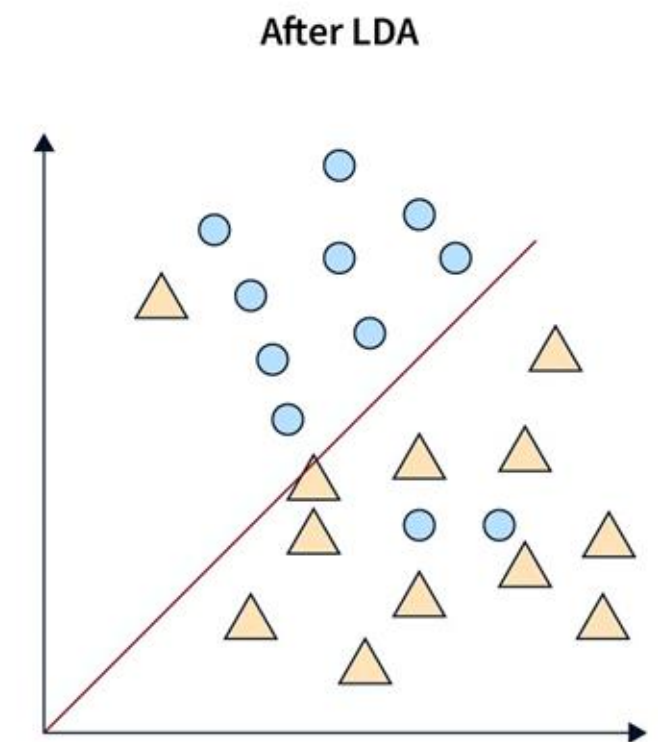
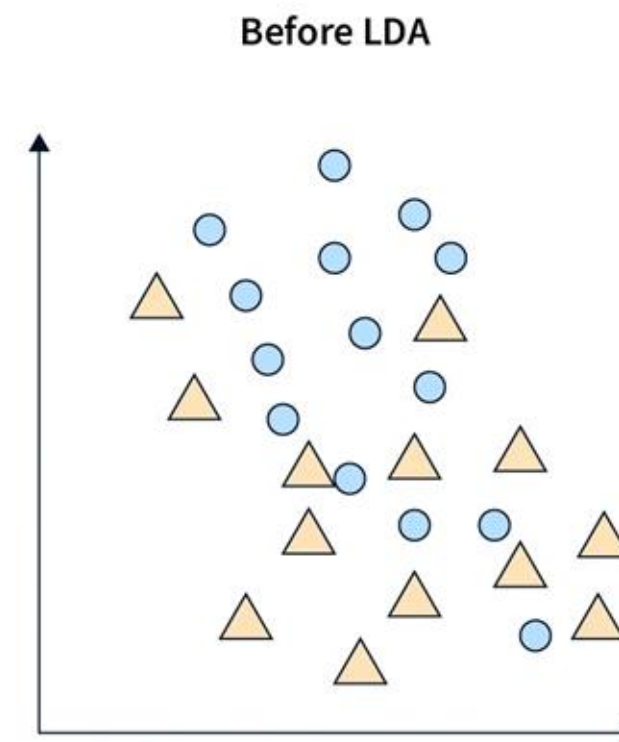
A la derecha, muestra de 30 puntos extraída de cada distribución gaussiana y los límites de decisión LDA ajustados.



# Linear Discriminant Analysis

La función lineal discriminante:

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$



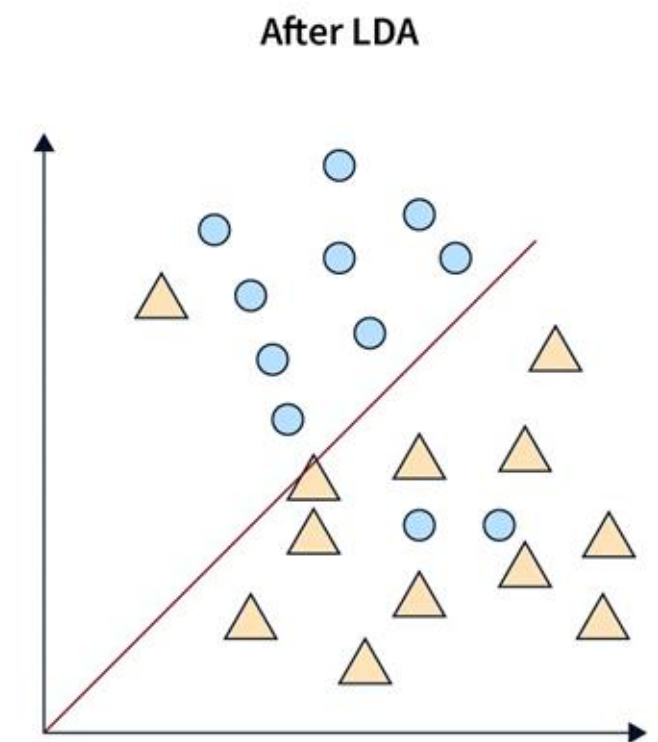
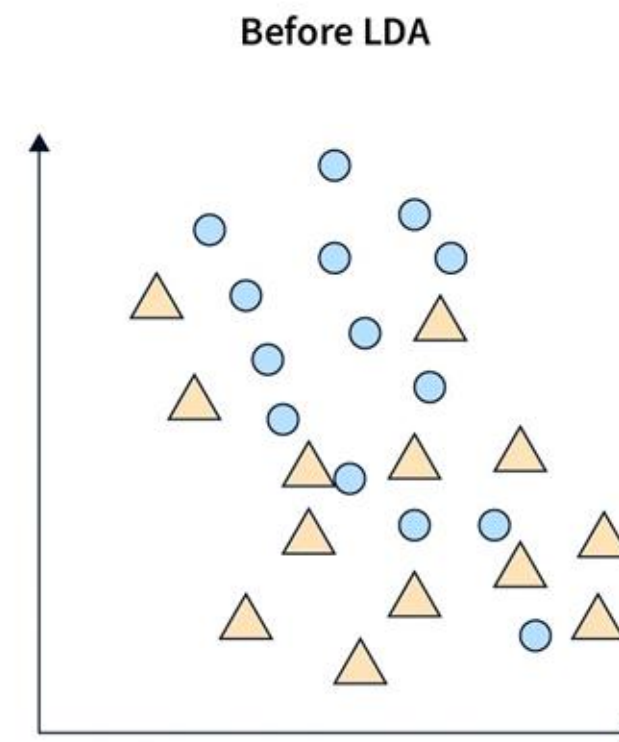
# Linear Discriminant Analysis

La función lineal discriminante:

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

Es una descripción equivalente de la regla de decisión, con:

$$G(x) = \operatorname{argmax}_k \delta_k(x)$$



# Linear Discriminant Analysis

La función lineal discriminante:

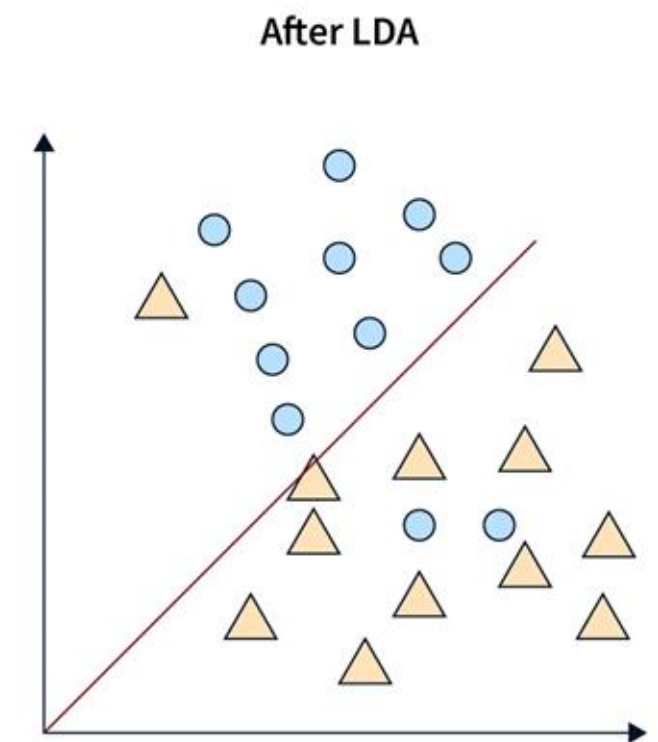
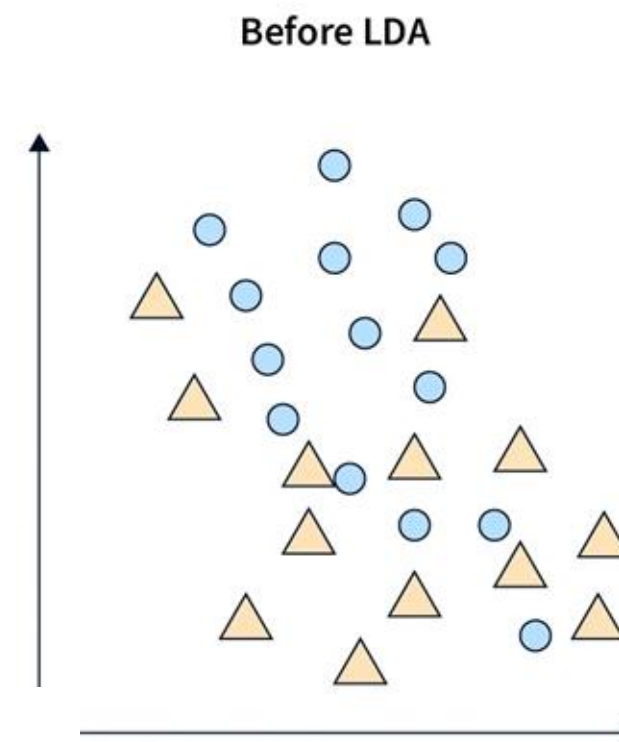
$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

Es una descripción equivalente de la regla de decisión, con:

$$G(x) = \operatorname{argmax}_k \delta_k(x)$$

No conocemos los parámetros de las distribuciones gaussianas, por lo que tendremos que estimarlos utilizando nuestros datos de entrenamiento:

- $\hat{\pi}_k = N_k/N$ , Donde  $N_k$  es el número de observaciones de la clase-k
- $\hat{\mu}_k = \sum_{g_i=k} x_i / N_k$ ;
- $\hat{\Sigma} = \sum_{k=1}^K \sum_{g_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T / (N - K)$ .



# Linear Discriminant Analysis

Con dos clases, existe una correspondencia simple entre el análisis discriminante lineal y la clasificación por regresión lineal. La regla LDA clasifica en la clase 2 si:

$$x^T \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) > \frac{1}{2} (\hat{\mu}_2 + \hat{\mu}_1)^T \hat{\Sigma}^{-1} (\hat{\mu}_2 - \hat{\mu}_1) - \log(N_2/N_1),$$

y en la clase 1 en caso contrario.

Supongamos que codificamos los objetivos de las dos clases como +1 y -1, respectivamente. Es fácil demostrar que el vector coeficiente de mínimos cuadrados es proporcional a la dirección LDA.



# Regresión Logística

Descripción

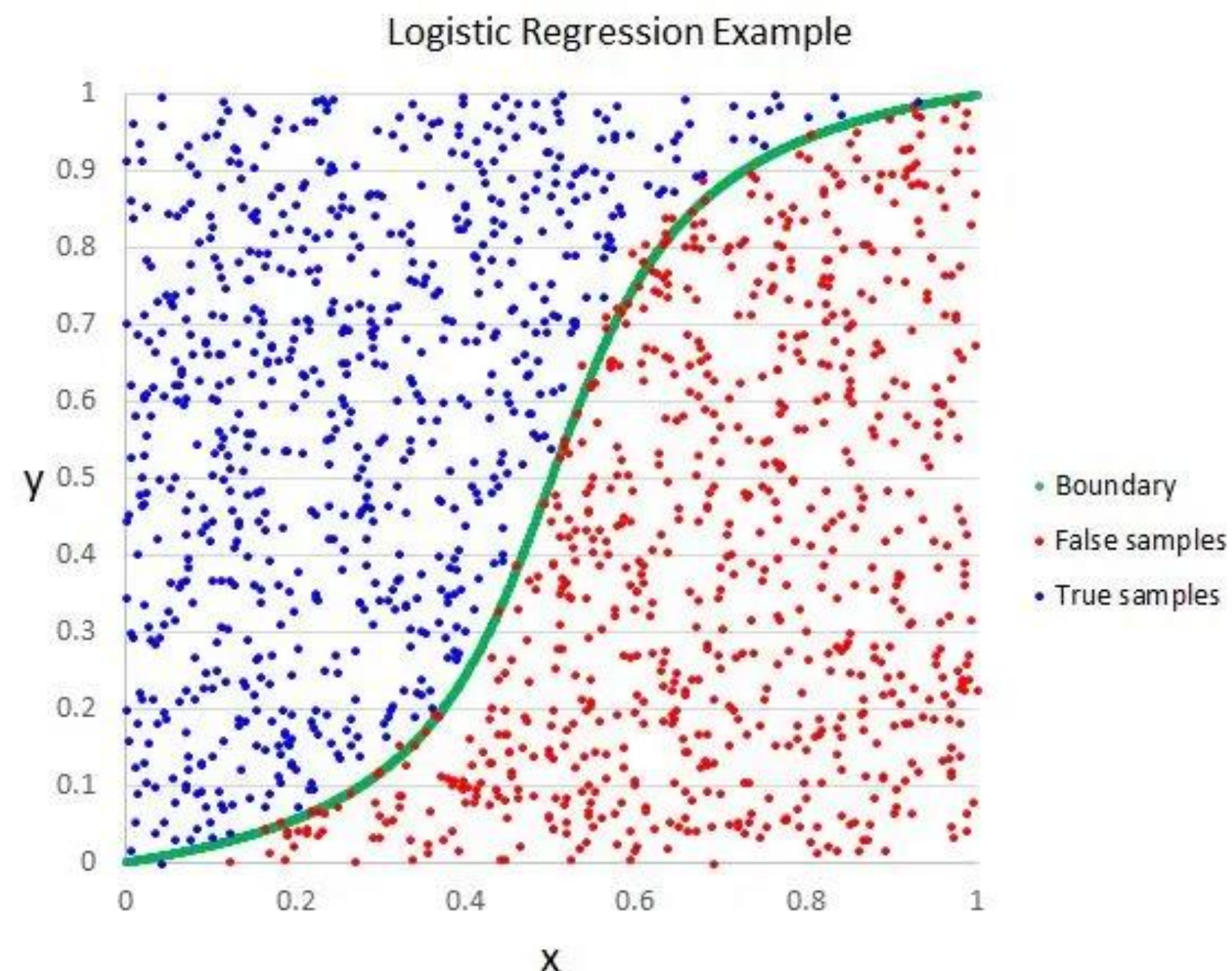


# Regresión Logística

Surge del deseo de modelar las probabilidades a posteriori de las clases  $K$  mediante funciones lineales en  $x$ , al tiempo que se garantiza que su suma sea igual a uno y que permanezcan en  $[0, 1]$ .

El modelo tiene la forma:

$$\begin{aligned} \log \frac{\Pr(G = 1|X = x)}{\Pr(G = K|X = x)} &= \beta_{10} + \beta_1^T x \\ \log \frac{\Pr(G = 2|X = x)}{\Pr(G = K|X = x)} &= \beta_{20} + \beta_2^T x \\ &\vdots \\ \log \frac{\Pr(G = K - 1|X = x)}{\Pr(G = K|X = x)} &= \beta_{(K-1)0} + \beta_{K-1}^T x. \end{aligned}$$

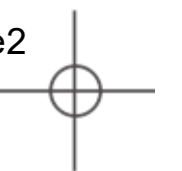
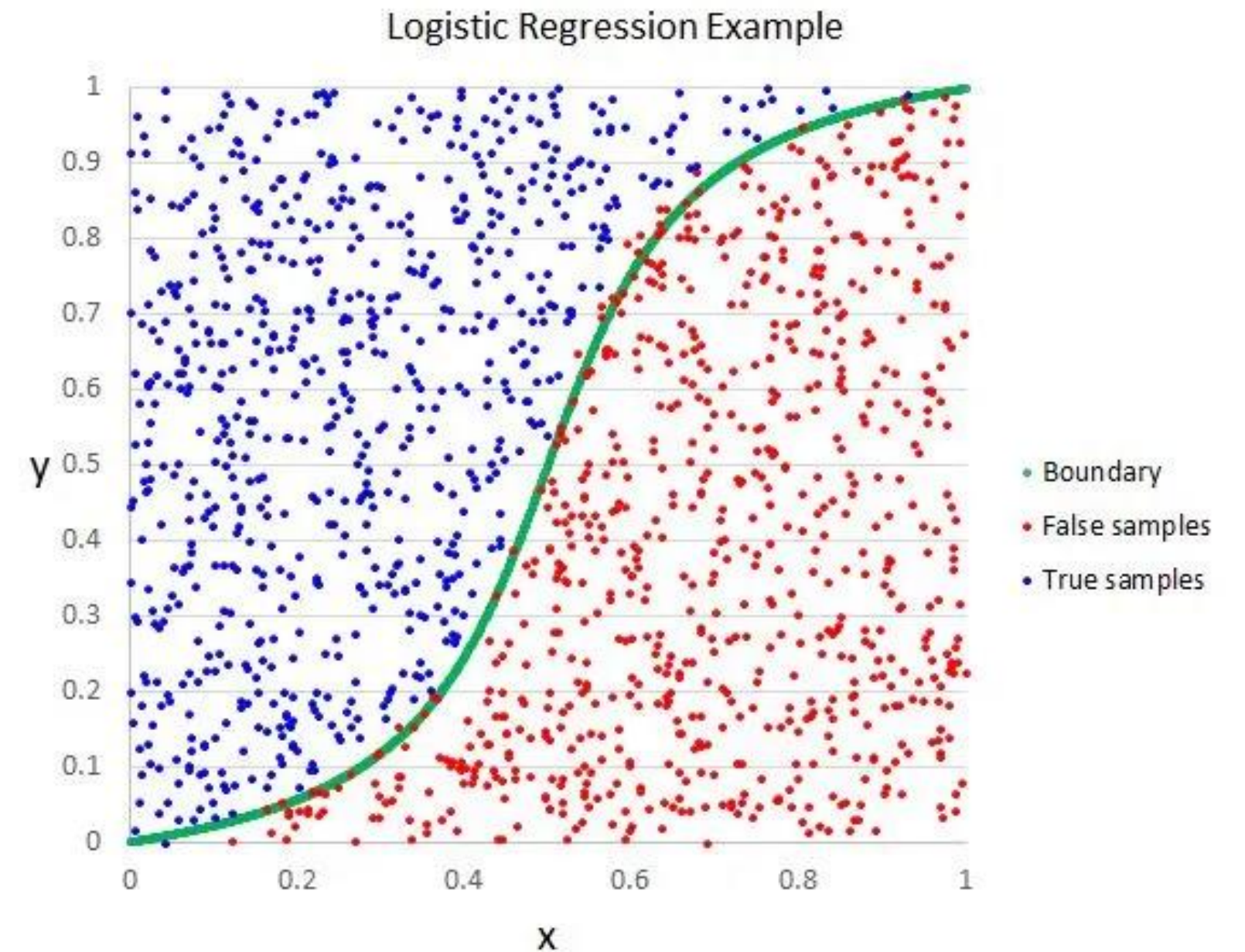


# Regresión Logística

Un simple cálculo muestra que:

$$\Pr(G = k|X = x) = \frac{\exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_{\ell}^T x)}, \quad k = 1, \dots, K - 1,$$

$$\Pr(G = K|X = x) = \frac{1}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_{\ell}^T x)},$$



# Regresión Logística

Un simple cálculo muestra que:

$$\Pr(G = k|X = x) = \frac{\exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_{\ell}^T x)}, \quad k = 1, \dots, K-1,$$

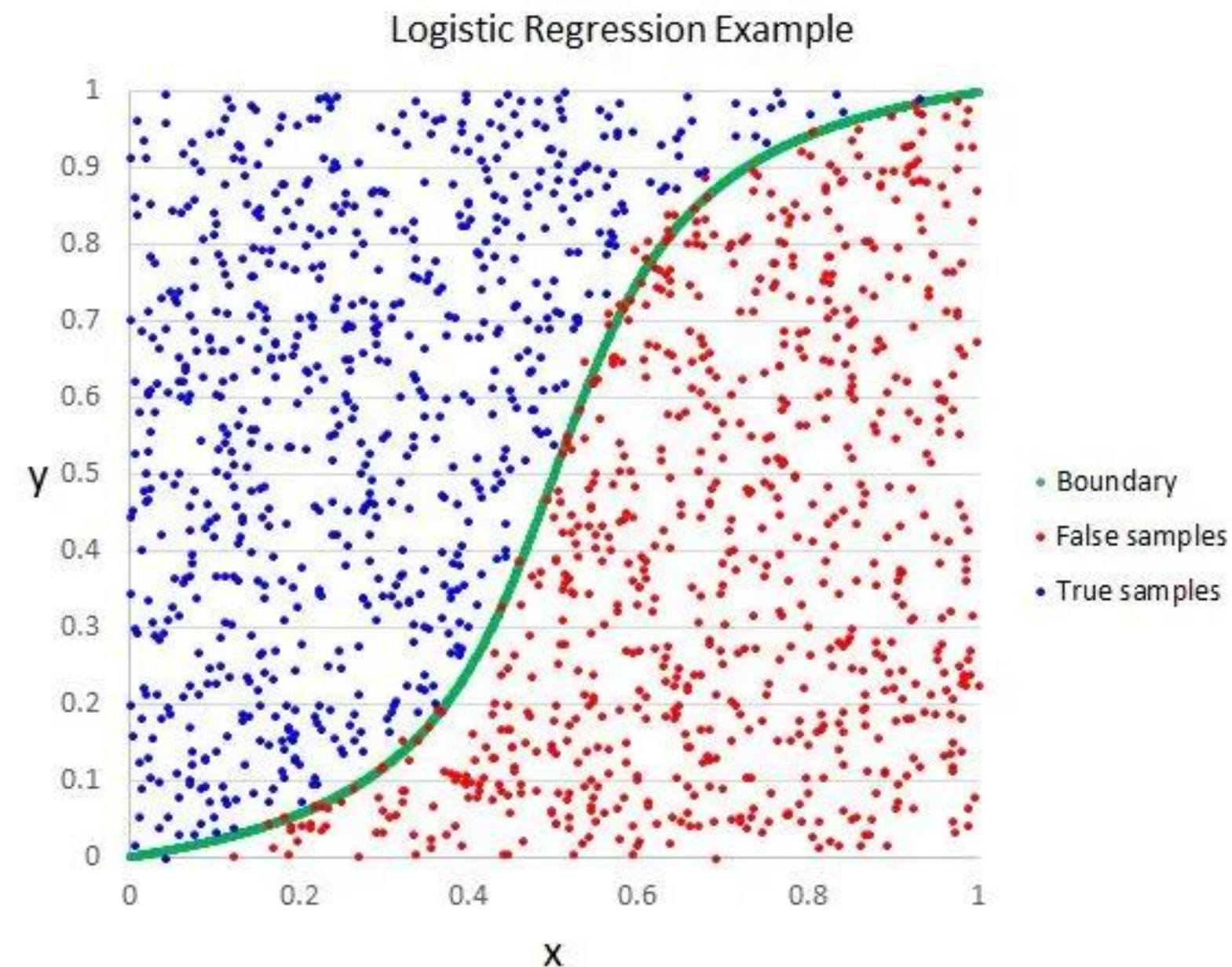
$$\Pr(G = K|X = x) = \frac{1}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_{\ell}^T x)},$$

Para enfatizar la dependencia de todo el conjunto de parámetros:

$$\theta = \{\beta_{10}, \beta_1^T, \dots, \beta_{(K-1)0}, \beta_{K-1}^T\}$$

denotamos las probabilidades:

$$\Pr(G = k|X = x) = p_k(x; \theta)$$



# Regresión Logística

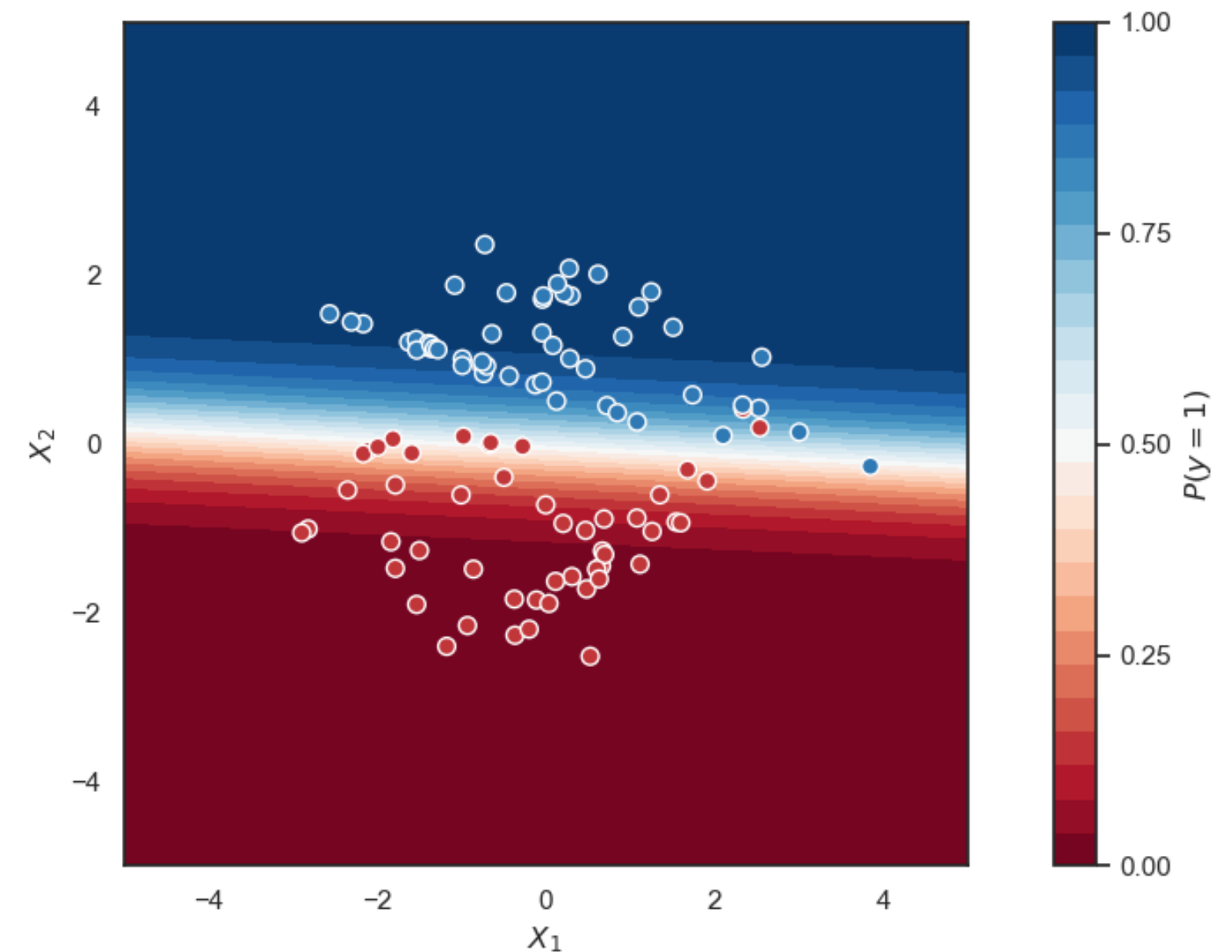
denotamos las probabilidades:

$$\Pr(G = k|X = x) = p_k(x; \theta)$$

Cuando  $K = 2$ , este modelo es especialmente sencillo, ya que solo hay una única función lineal.

Se utiliza ampliamente en aplicaciones bioestadísticas en las que las respuestas binarias (dos clases) son bastante frecuentes.

Por ejemplo, los pacientes sobreviven o mueren, padecen o no padecen una enfermedad cardíaca, o una afección está presente o ausente.



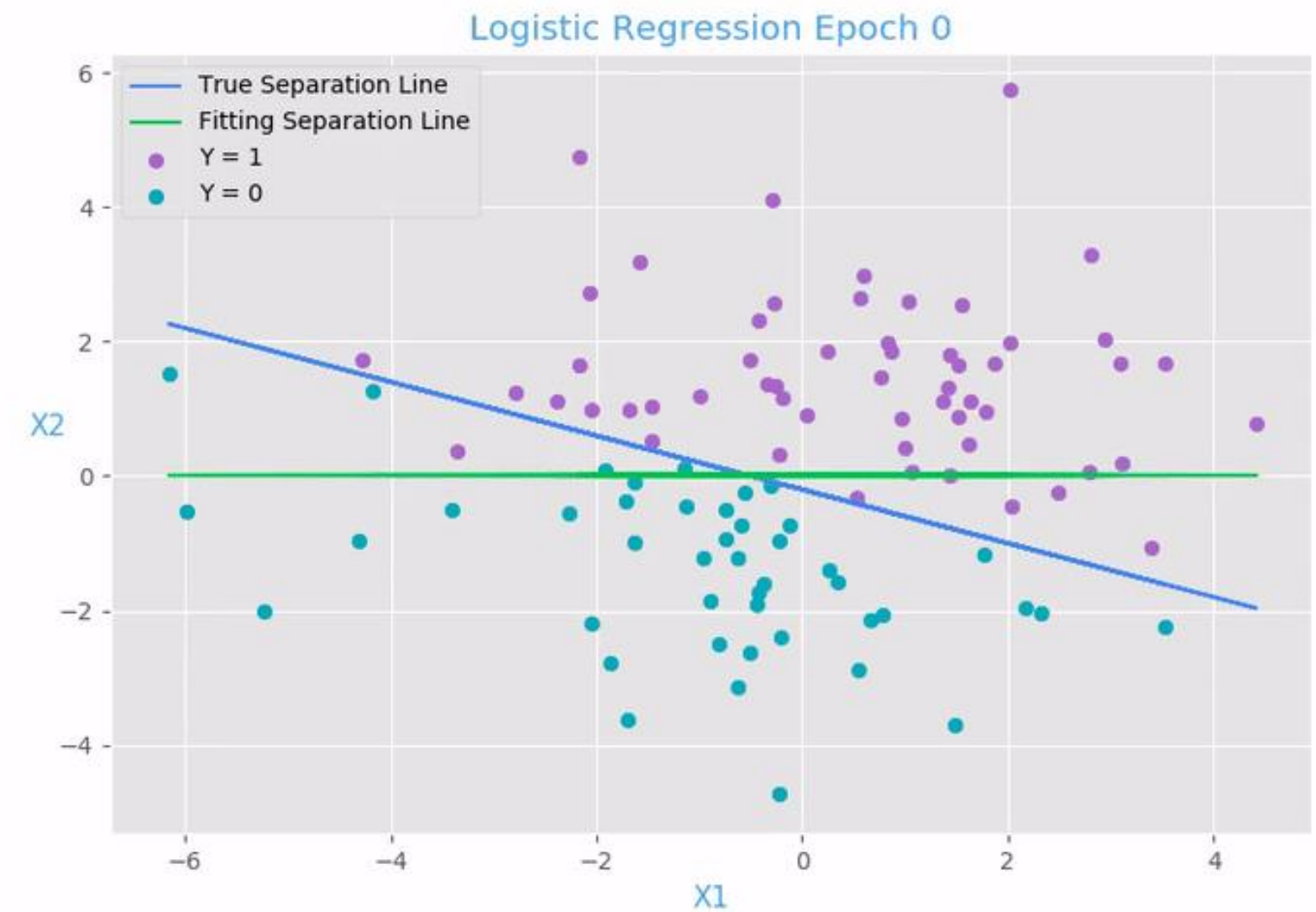
<https://ml-explained.com/blog/logistic-regression-explained>



# Fitting - Modelos de Regresión Logística

La verosimilitud logarítmica para N observaciones es

$$\ell(\theta) = \sum_{i=1}^N \log p_{g_i}(x_i; \theta),$$

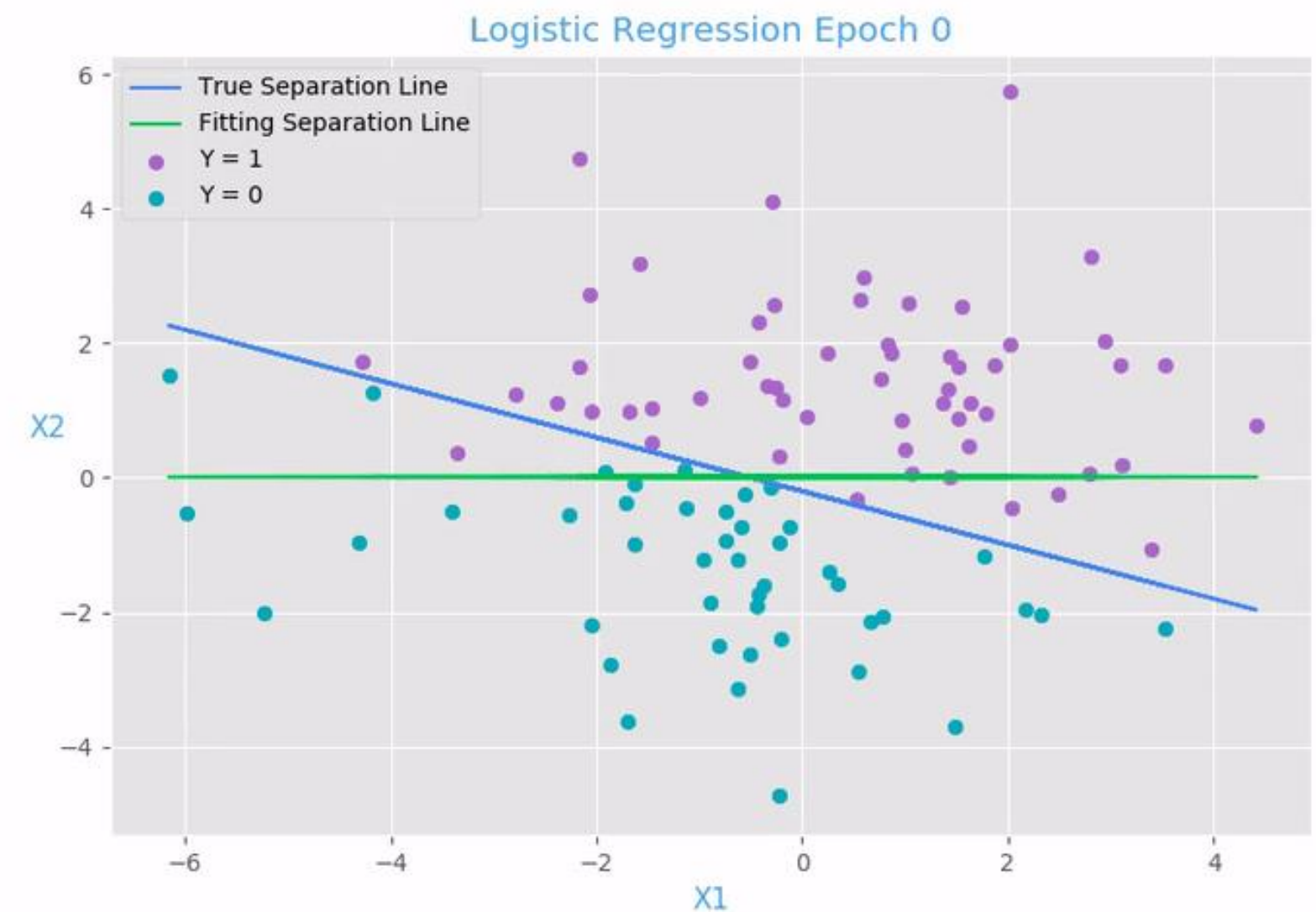


# Fitting - Modelos de Regresión Logística

La verosimilitud logarítmica para N observaciones es

$$\ell(\theta) = \sum_{i=1}^N \log p_{g_i}(x_i; \theta),$$

Donde:  $p_k(x_i; \theta) = \Pr(G = k | X = x_i; \theta)$



# Fitting - Modelos de Regresión Logística

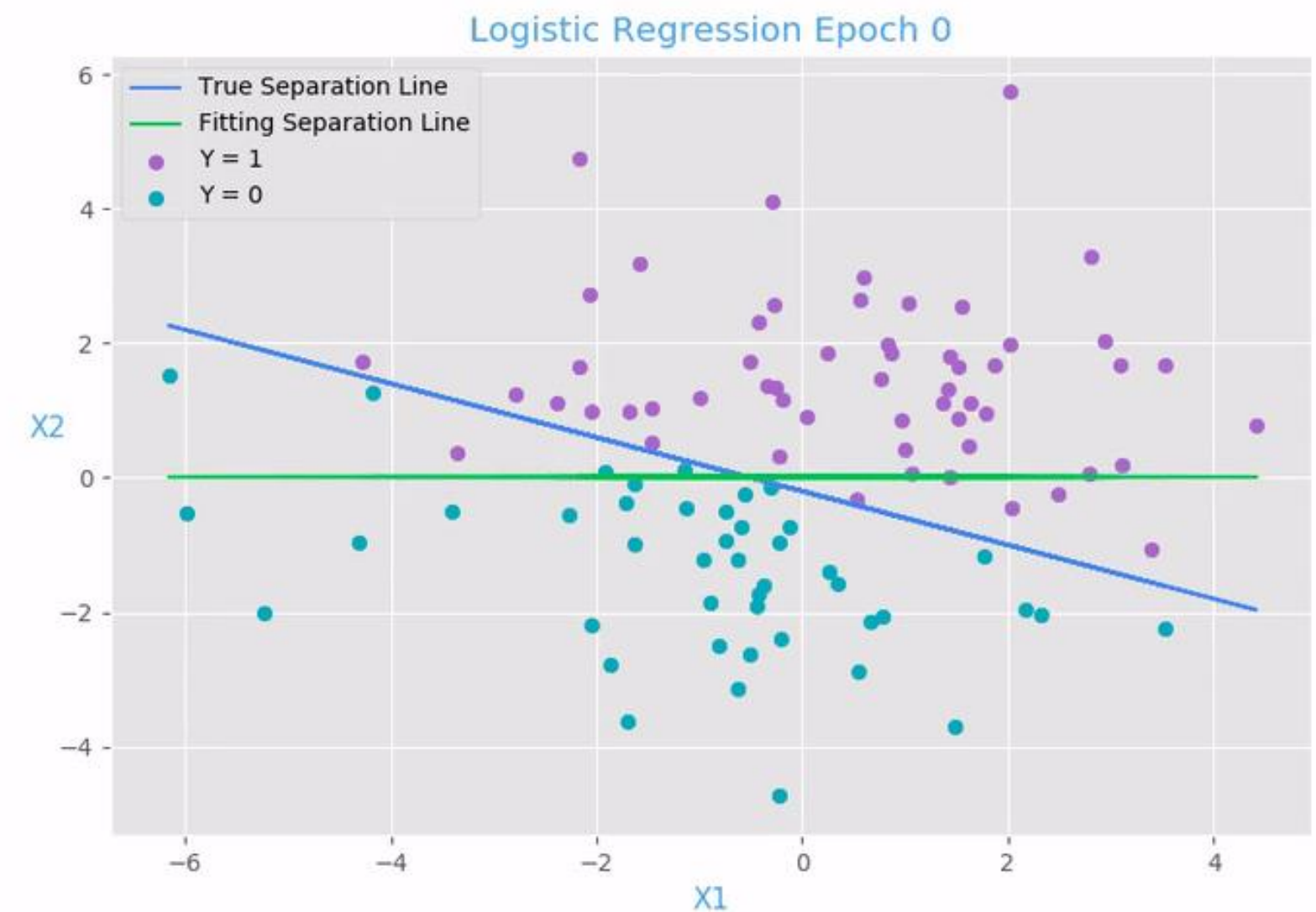
La verosimilitud logarítmica para N observaciones es

$$\ell(\theta) = \sum_{i=1}^N \log p_{g_i}(x_i; \theta),$$

Donde:  $p_k(x_i; \theta) = \Pr(G = k | X = x_i; \theta)$

La verosimilitud logarítmica se puede escribir como:

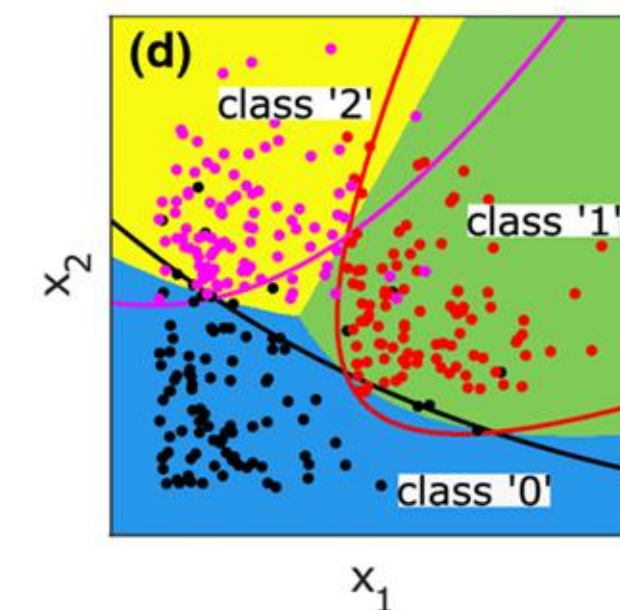
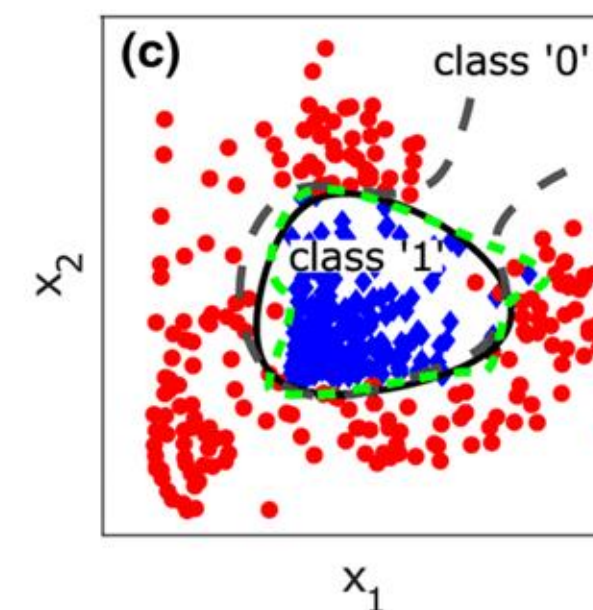
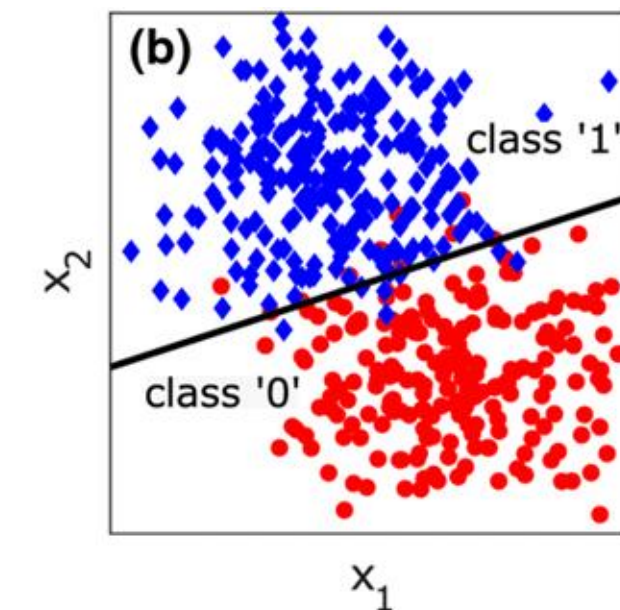
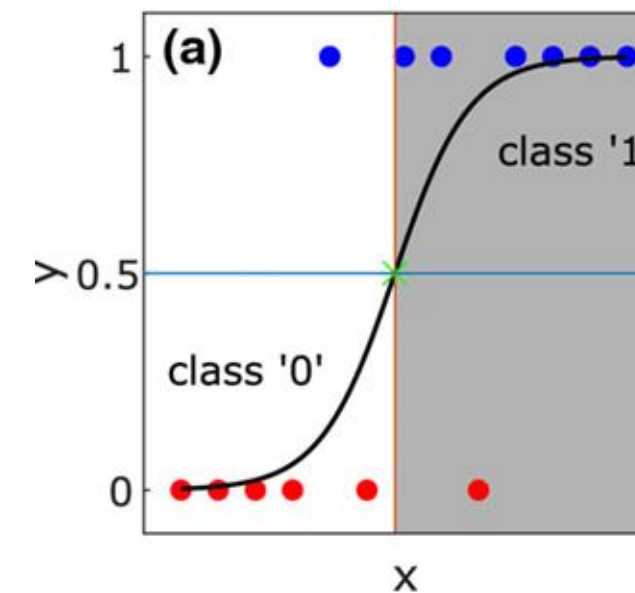
$$\begin{aligned} \ell(\beta) &= \sum_{i=1}^N \left\{ y_i \log p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta)) \right\} \\ &= \sum_{i=1}^N \left\{ y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}) \right\}. \end{aligned}$$



# Fitting - Modelos de Regresión Logística

Para maximizar la verosimilitud logarítmica, establecemos sus derivadas en cero. Estas ecuaciones de puntuación son

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^N x_i (y_i - p(x_i; \beta)) = 0,$$



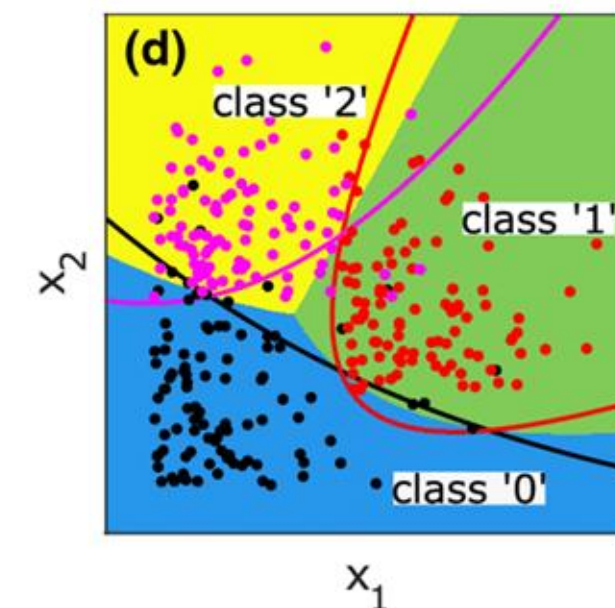
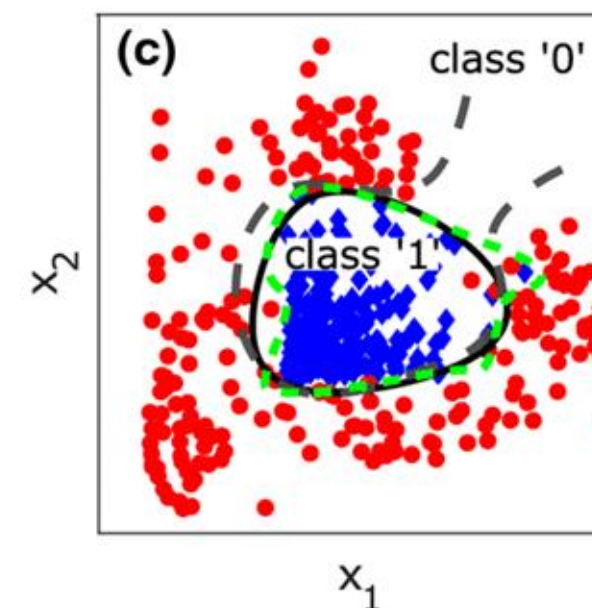
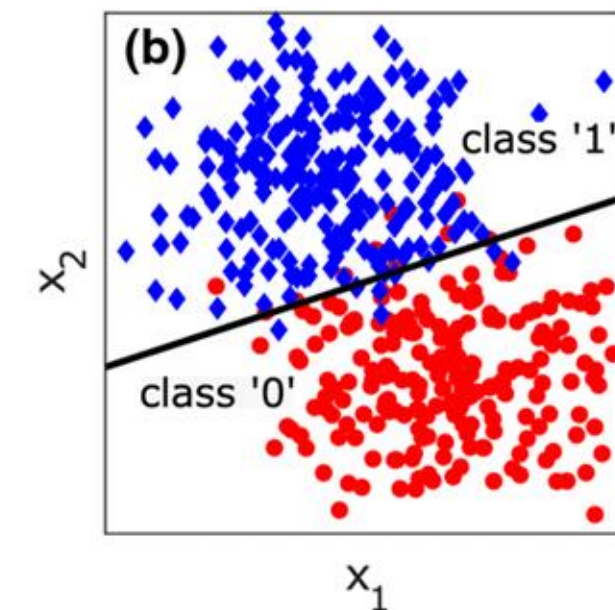
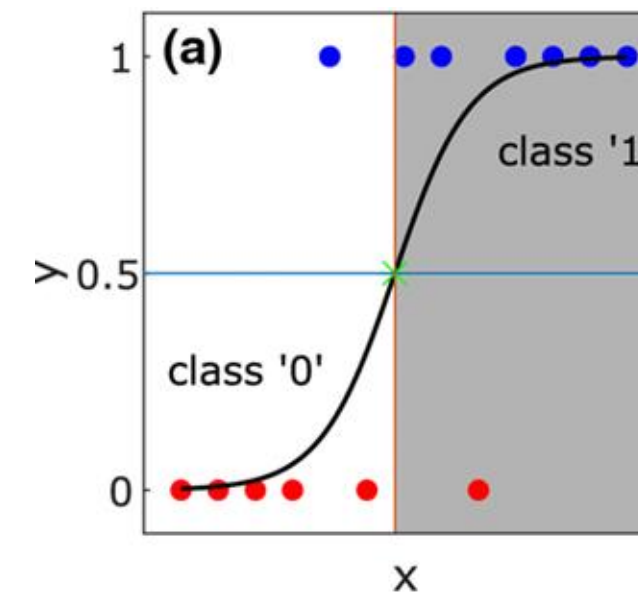
# Fitting - Modelos de Regresión Logística

Para maximizar la verosimilitud logarítmica, establecemos sus derivadas en cero. Estas ecuaciones de puntuación son

$$\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{i=1}^N x_i (y_i - p(x_i; \beta)) = 0,$$

Para resolver esta ecuación, utilizamos el algoritmo de Newton-Raphson, que requiere la derivada segunda o la matriz Hessian:

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = - \sum_{i=1}^N x_i x_i^T p(x_i; \beta) (1 - p(x_i; \beta)).$$

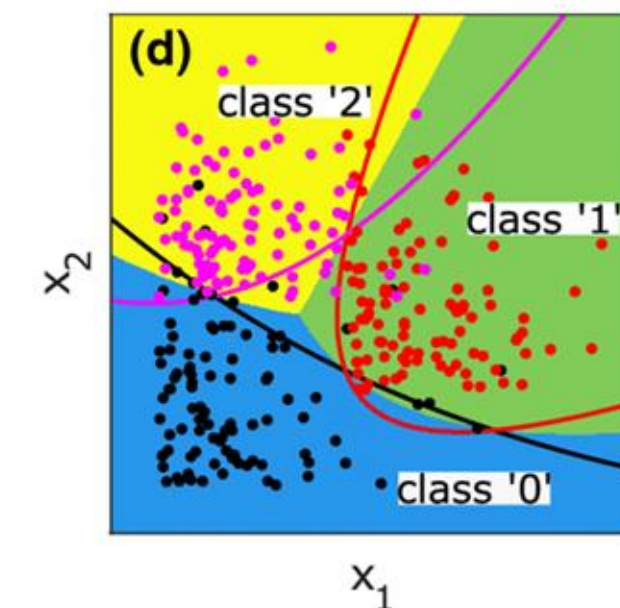
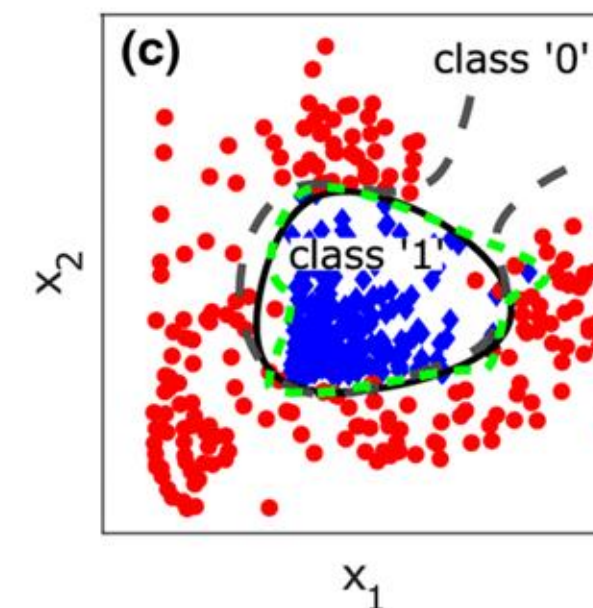
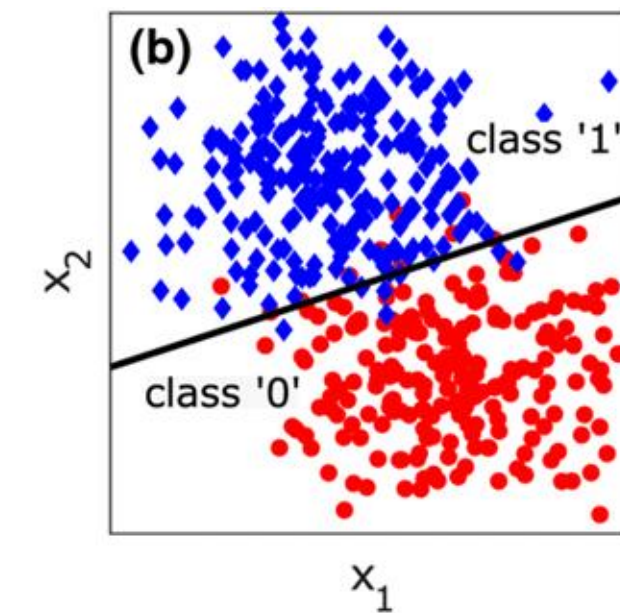
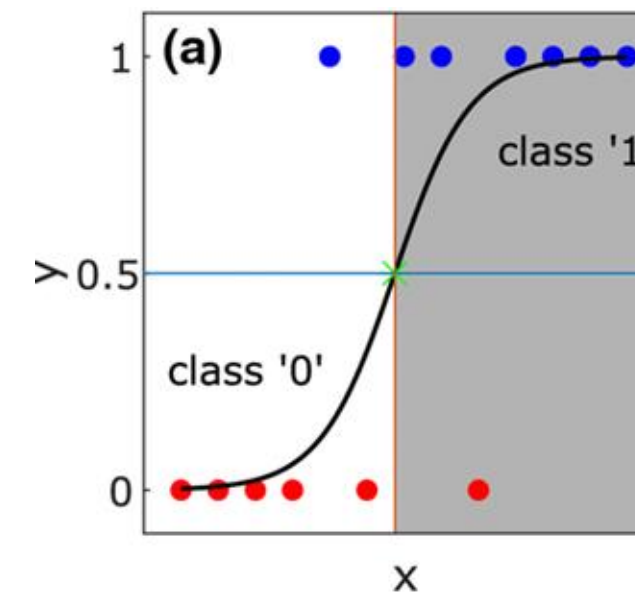


# Fitting - Modelos de Regresión Logística

Comenzando con  $\beta^{\text{old}}$ , una única actualización de Newton es:

$$\beta^{\text{new}} = \beta^{\text{old}} - \left( \frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial \ell(\beta)}{\partial \beta},$$

Donde las derivadas se evalúan en  $\beta^{\text{old}}$ .

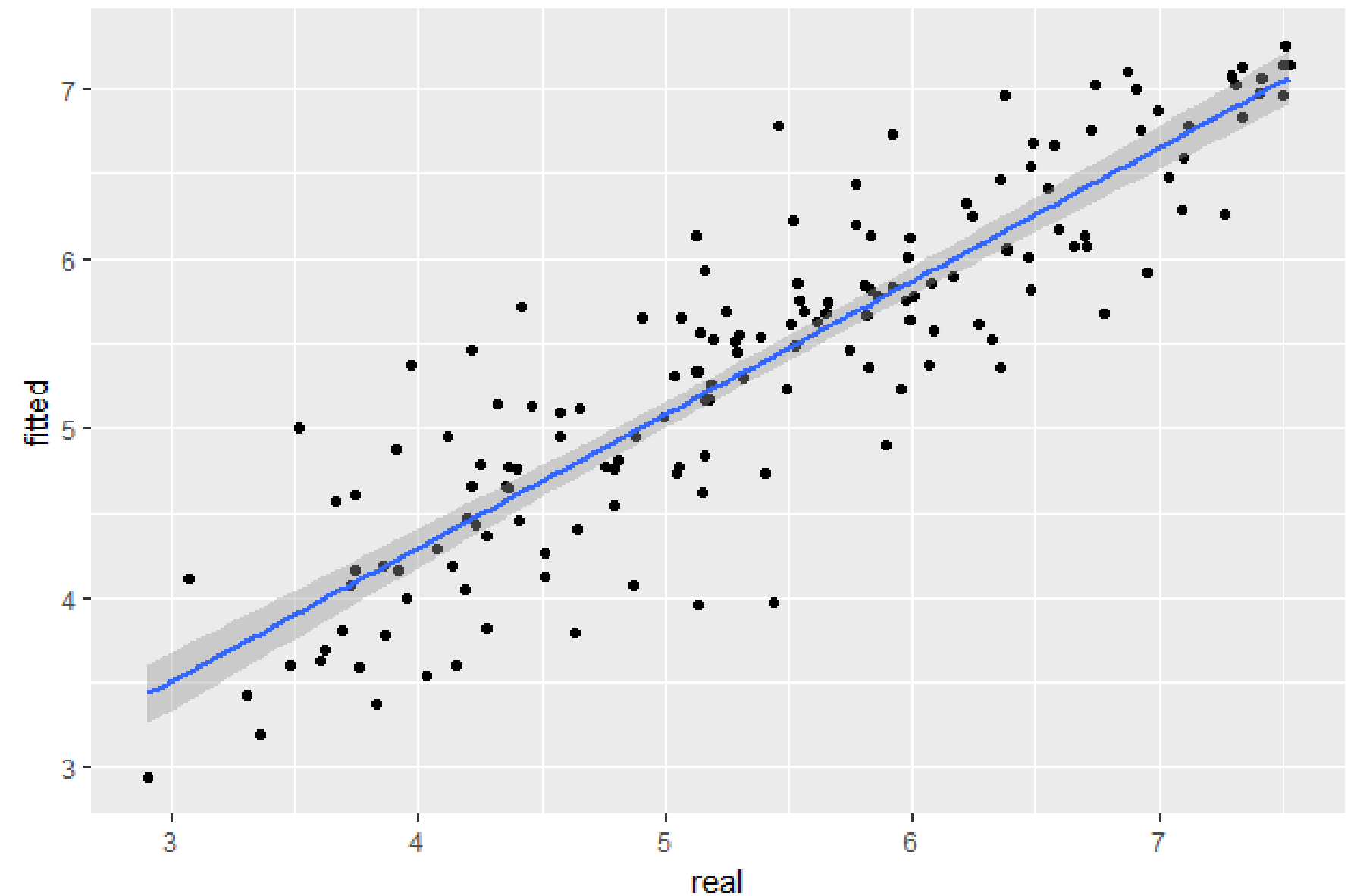


# Fitting - Modelos de Regresión Logística

Sea  $y$  el vector de valores  $y_i$ ,  $X$  la matriz  $N \times (p + 1)$  de valores  $x_i$ ,  $p$  el vector de probabilidades ajustadas con el elemento  $i$ -ésimo  $p(x_i; \beta_{old})$  y  $W$  una matriz diagonal  $N \times N$  de pesos con el elemento diagonal  $i$ -ésimo  $p(x_i; \beta_{old})(1 - p(x_i; \beta_{old}))$ . Tenemos:

$$\frac{\partial \ell(\beta)}{\partial \beta} = X^T (y - p)$$

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -X^T W X$$



# Fitting - Modelos de Regresión Logística

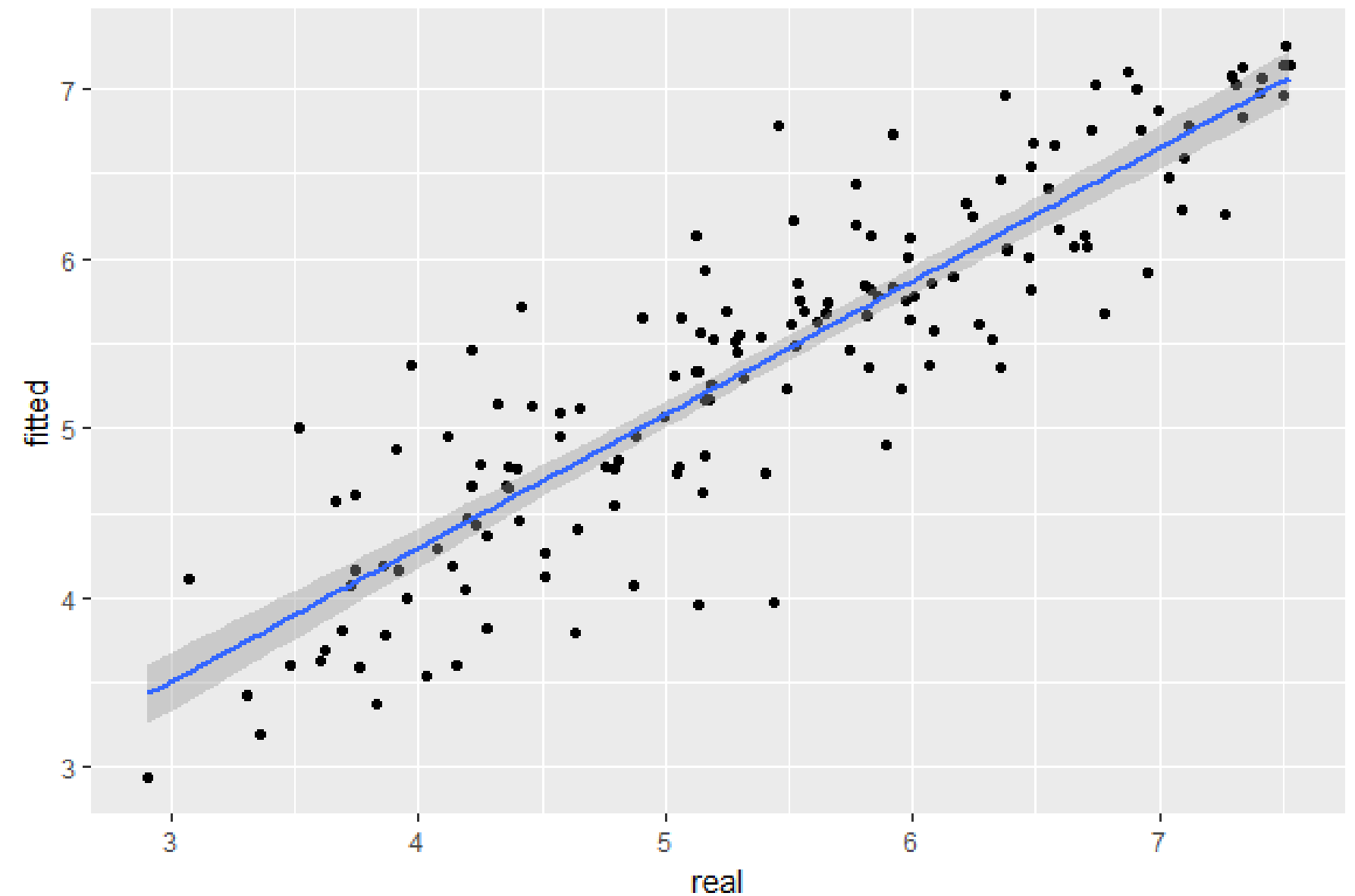
Sea  $y$  el vector de valores  $y_i$ ,  $X$  la matriz  $N \times (p + 1)$  de valores  $x_i$ ,  $p$  el vector de probabilidades ajustadas con el elemento  $i$ -ésimo  $p(x_i; \beta^{\text{old}})$  y  $W$  una matriz diagonal  $N \times N$  de pesos con el elemento diagonal  $i$ -ésimo  $p(x_i; \beta^{\text{old}})(1 - p(x_i; \beta^{\text{old}}))$ . Tenemos:

$$\frac{\partial \ell(\beta)}{\partial \beta} = X^T (y - p)$$

$$\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = -X^T W X$$

El paso de Newton es, por lo tanto,

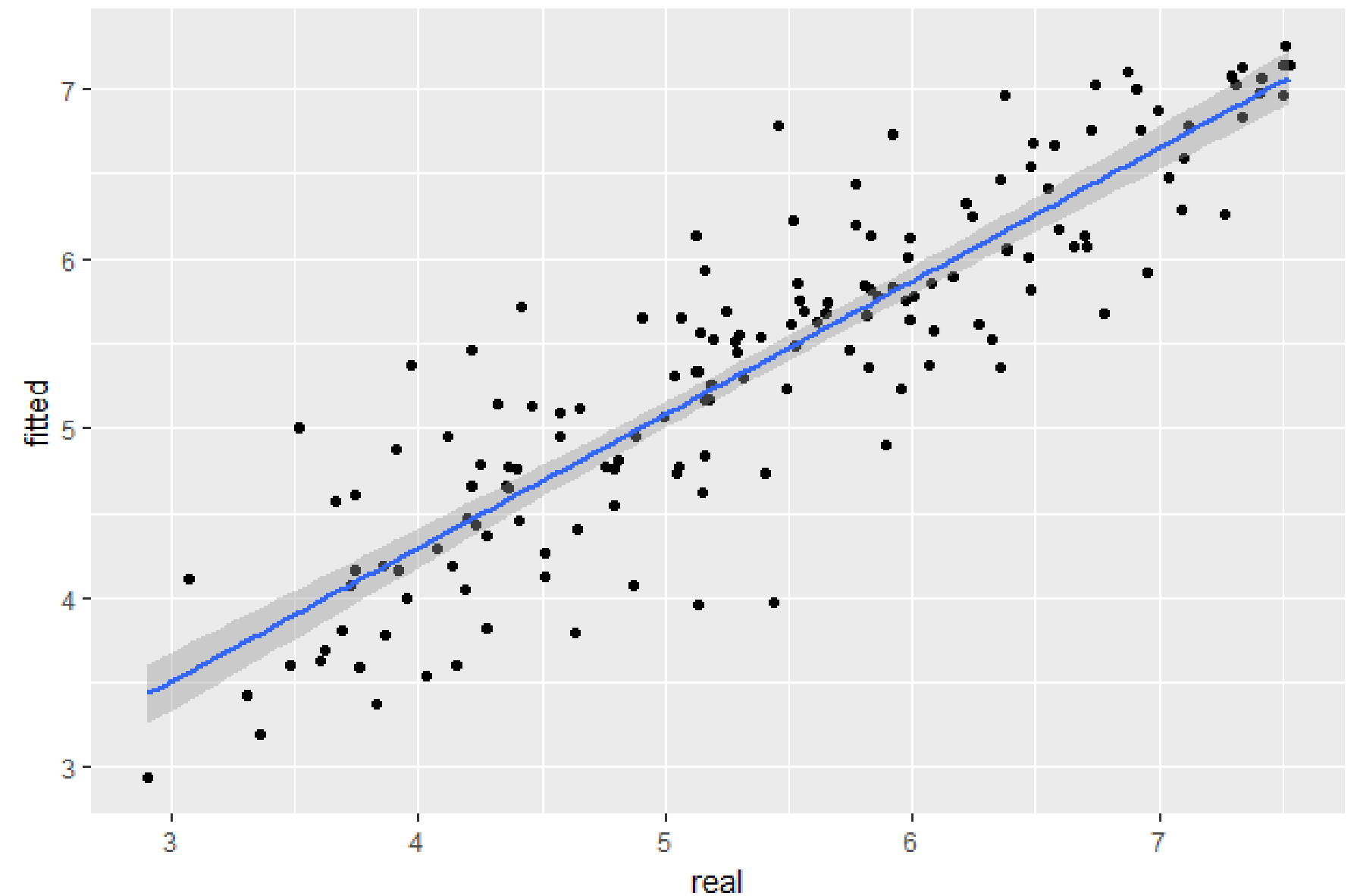
$$\begin{aligned} \beta^{\text{new}} &= \beta^{\text{old}} + (X^T W X)^{-1} X^T (y - p) \\ &= (X^T W X)^{-1} X^T W (X \beta^{\text{old}} + W^{-1} (y - p)) \\ &= (X^T W X)^{-1} X^T W z. \end{aligned}$$



# Fitting - Modelos de Regresión Logística

El paso de Newton es, por lo tanto,

$$\begin{aligned}\beta^{\text{new}} &= \beta^{\text{old}} + (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{y} - \mathbf{p}) \\ &= (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} (\mathbf{X} \beta^{\text{old}} + \mathbf{W}^{-1} (\mathbf{y} - \mathbf{p})) \\ &= (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{z}.\end{aligned}$$



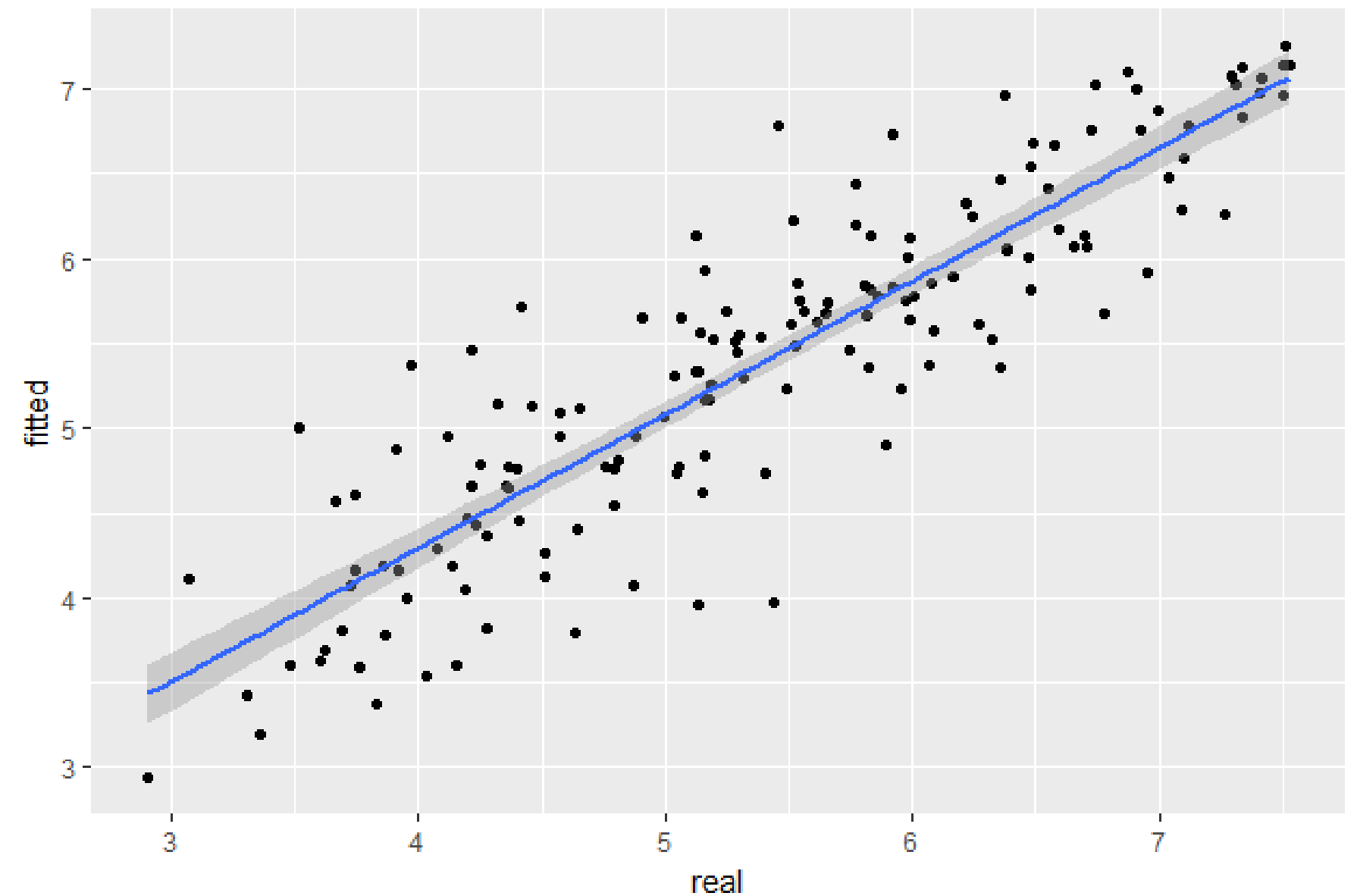
# Fitting - Modelos de Regresión Logística

El paso de Newton es, por lo tanto,

$$\begin{aligned}\beta^{\text{new}} &= \beta^{\text{old}} + (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{y} - \mathbf{p}) \\ &= (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} (\mathbf{X} \beta^{\text{old}} + \mathbf{W}^{-1} (\mathbf{y} - \mathbf{p})) \\ &= (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{z}.\end{aligned}$$

En la segunda y tercera línea hemos reexpresado el paso de Newton como un paso de mínimos cuadrados ponderados, con la respuesta

$$\mathbf{z} = \mathbf{X} \beta^{\text{old}} + \mathbf{W}^{-1} (\mathbf{y} - \mathbf{p}),$$



# Fitting - Modelos de Regresión Logística

El paso de Newton es, por lo tanto,

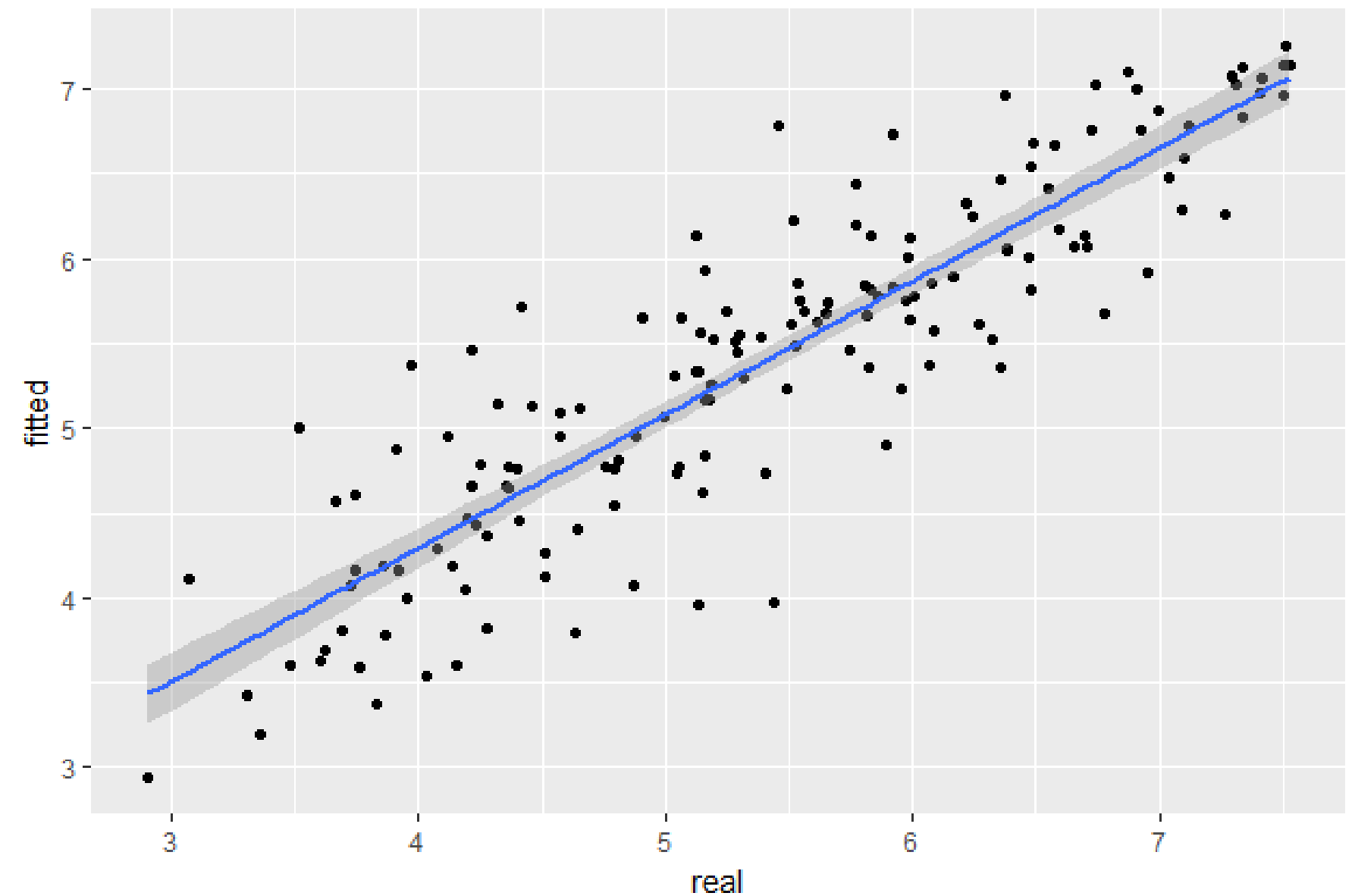
$$\begin{aligned}\beta^{\text{new}} &= \beta^{\text{old}} + (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{y} - \mathbf{p}) \\ &= (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} (\mathbf{X} \beta^{\text{old}} + \mathbf{W}^{-1} (\mathbf{y} - \mathbf{p})) \\ &= (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{z}.\end{aligned}$$

En la segunda y tercera línea hemos reexpresado el paso de Newton como un paso de mínimos cuadrados ponderados, con la respuesta

$$\mathbf{z} = \mathbf{X} \beta^{\text{old}} + \mathbf{W}^{-1} (\mathbf{y} - \mathbf{p}),$$

Se conoce como respuesta ajustada. Este algoritmo se denomina mínimos cuadrados iterativamente reponderados o IRLS, ya que cada iteración resuelve el problema de mínimos cuadrados ponderados:

$$\beta^{\text{new}} \leftarrow \arg \min_{\beta} (\mathbf{z} - \mathbf{X} \beta)^T \mathbf{W} (\mathbf{z} - \mathbf{X} \beta).$$



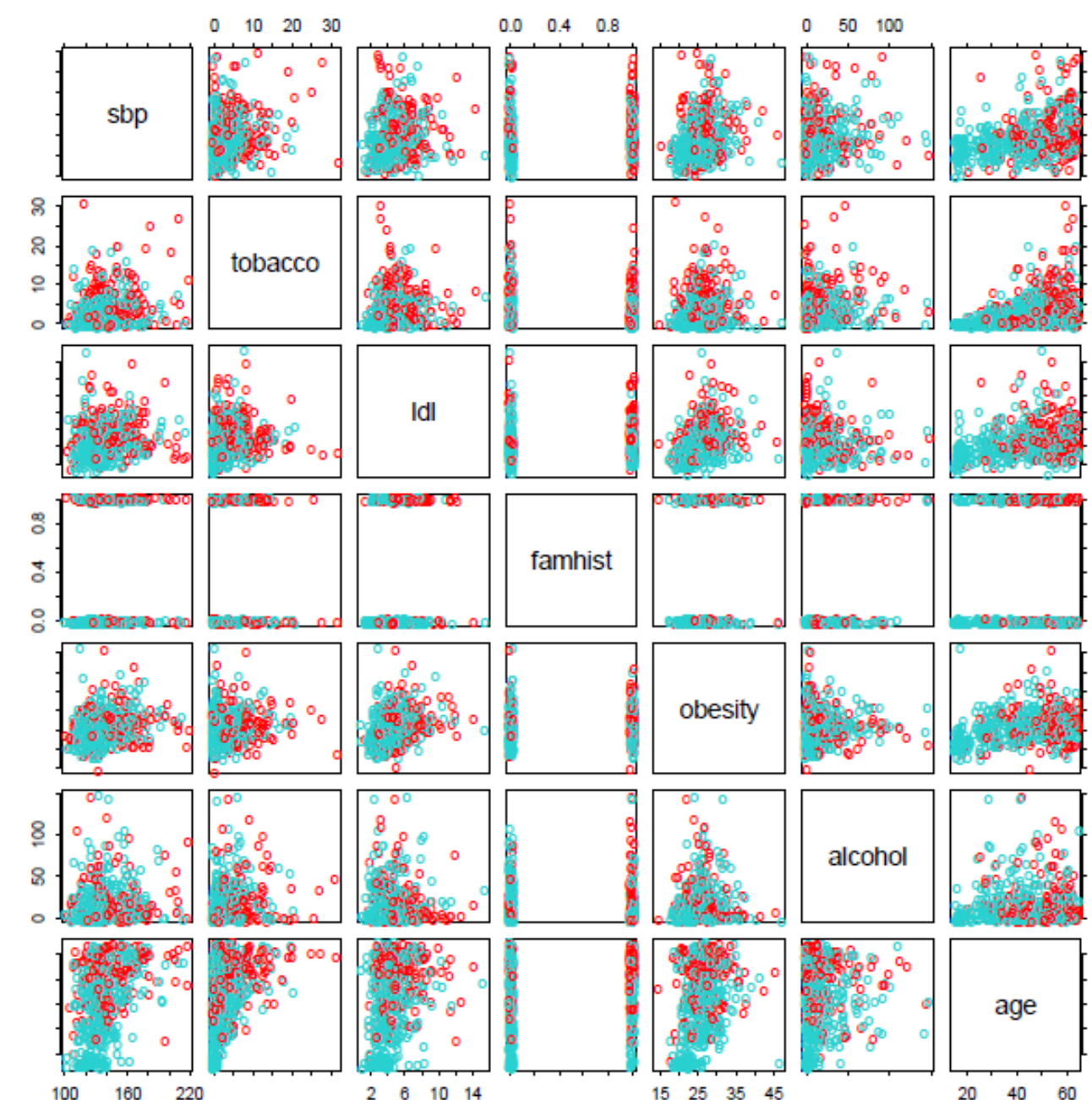
# Fitting - Modelos de Regresión Logística

Ajuste de regresión logística

	Coefficient	Std. Error	Z Score
(Intercept)	-4.130	0.964	-4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	-1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

Ajuste de regresión logística por pasos

	Coefficient	Std. Error	Z score
(Intercept)	-4.204	0.498	-8.45
tobacco	0.081	0.026	3.16
ldl	0.168	0.054	3.09
famhist	0.924	0.223	4.14
age	0.044	0.010	4.52



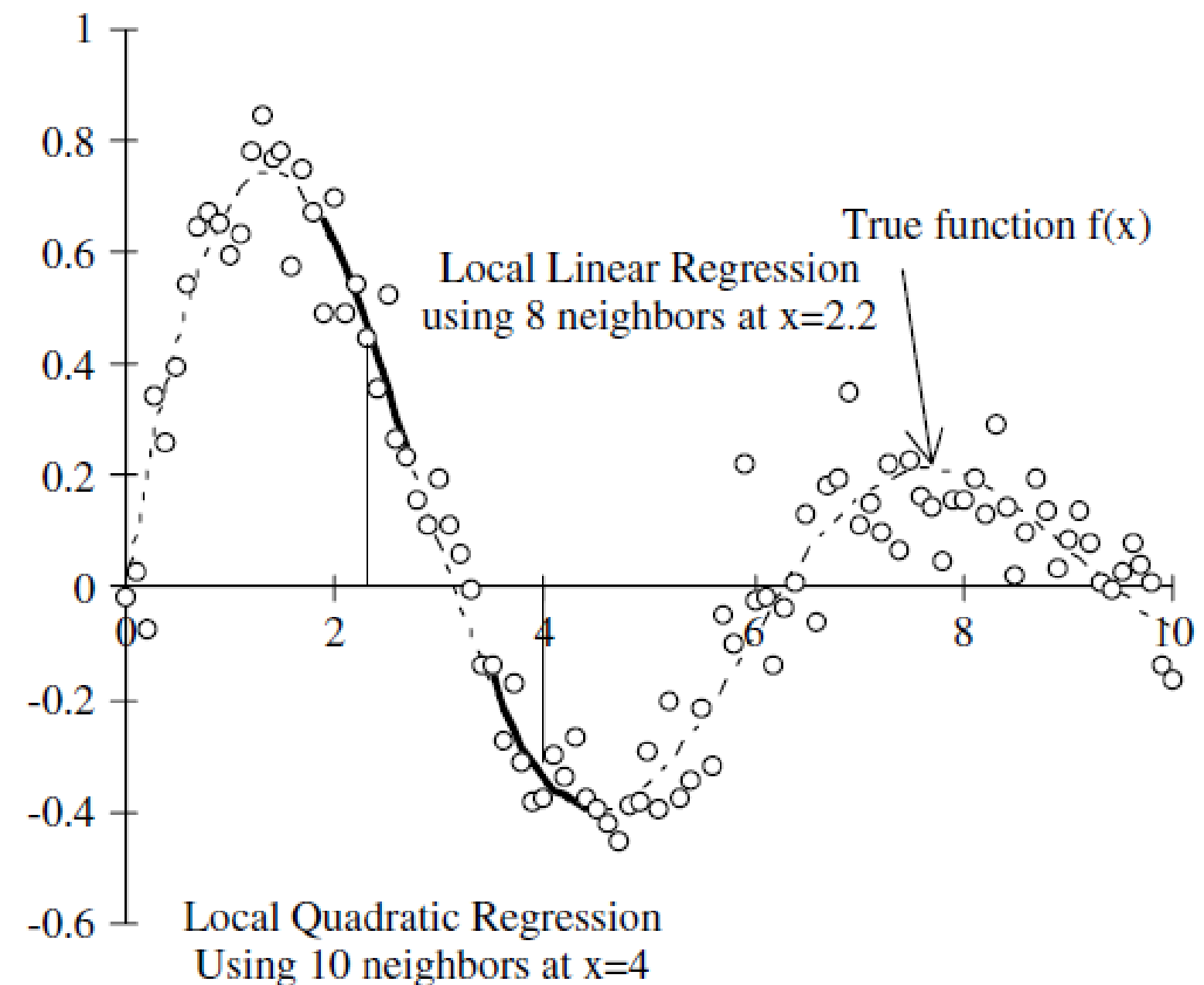
Resultados de un ajuste de regresión logística y scatterplots a los datos sobre un caso: enfermedades cardíacas en Sudáfrica.



# Aproximaciones cuadráticas e inferencia

Las estimaciones de los parámetros de máxima verosimilitud  $\hat{\beta}$  satisfacen una relación de autoconsistencia: son los coeficientes de un ajuste ponderado por mínimos cuadrados, donde las respuestas son:

$$z_i = x_i^T \hat{\beta} + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)},$$



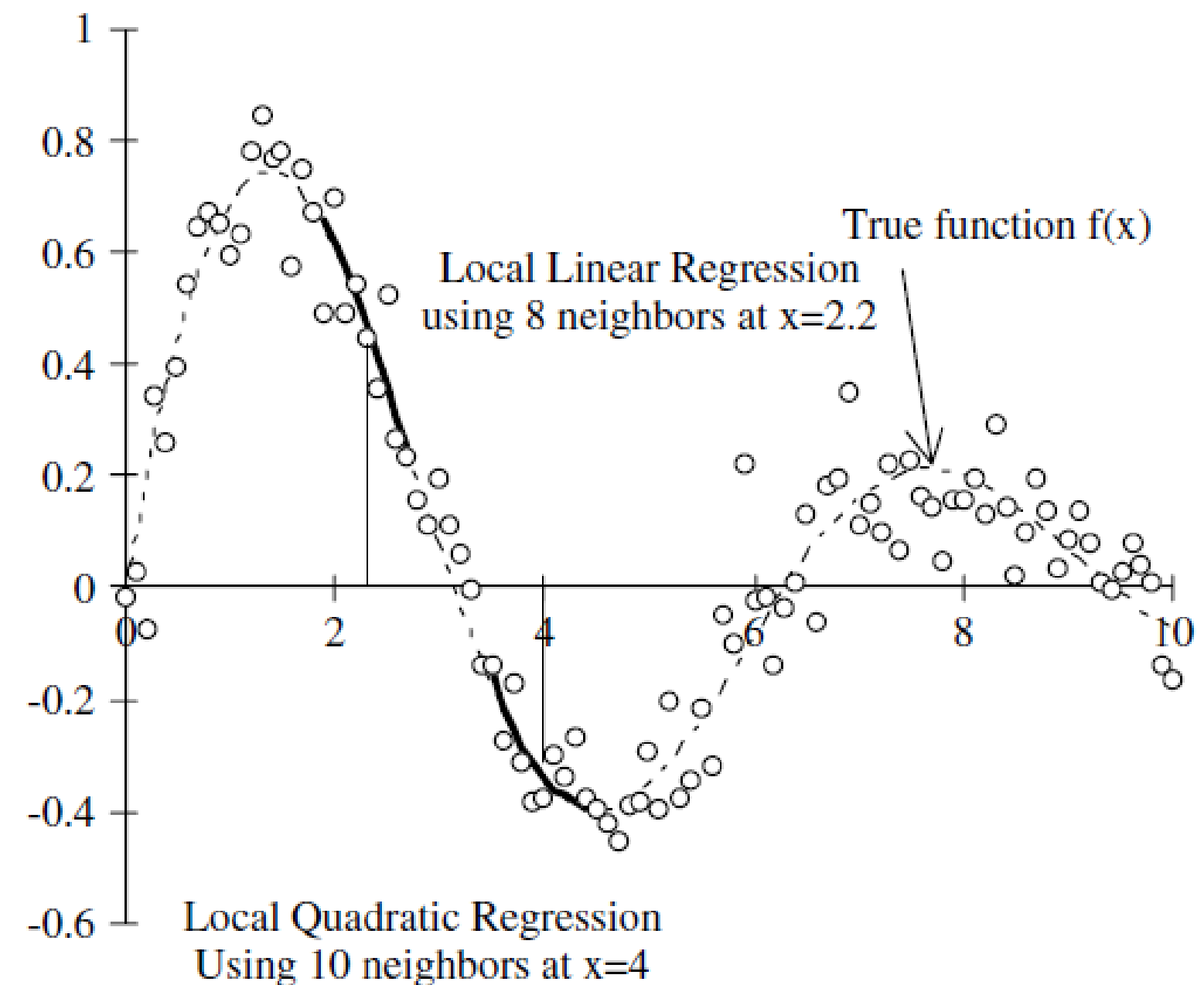
# Aproximaciones cuadráticas e inferencia

Las estimaciones de los parámetros de máxima verosimilitud  $\hat{\beta}$  satisfacen una relación de autoconsistencia: son los coeficientes de un ajuste ponderado por mínimos cuadrados, donde las respuestas son:

$$z_i = x_i^T \hat{\beta} + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)},$$

Los pesos son:

$$w_i = \hat{p}_i(1 - \hat{p}_i)$$



# Aproximaciones cuadráticas e inferencia

Las estimaciones de los parámetros de máxima verosimilitud  $\hat{\beta}$  satisfacen una relación de autoconsistencia: son los coeficientes de un ajuste ponderado por mínimos cuadrados, donde las respuestas son:

$$z_i = x_i^T \hat{\beta} + \frac{(y_i - \hat{p}_i)}{\hat{p}_i(1 - \hat{p}_i)},$$

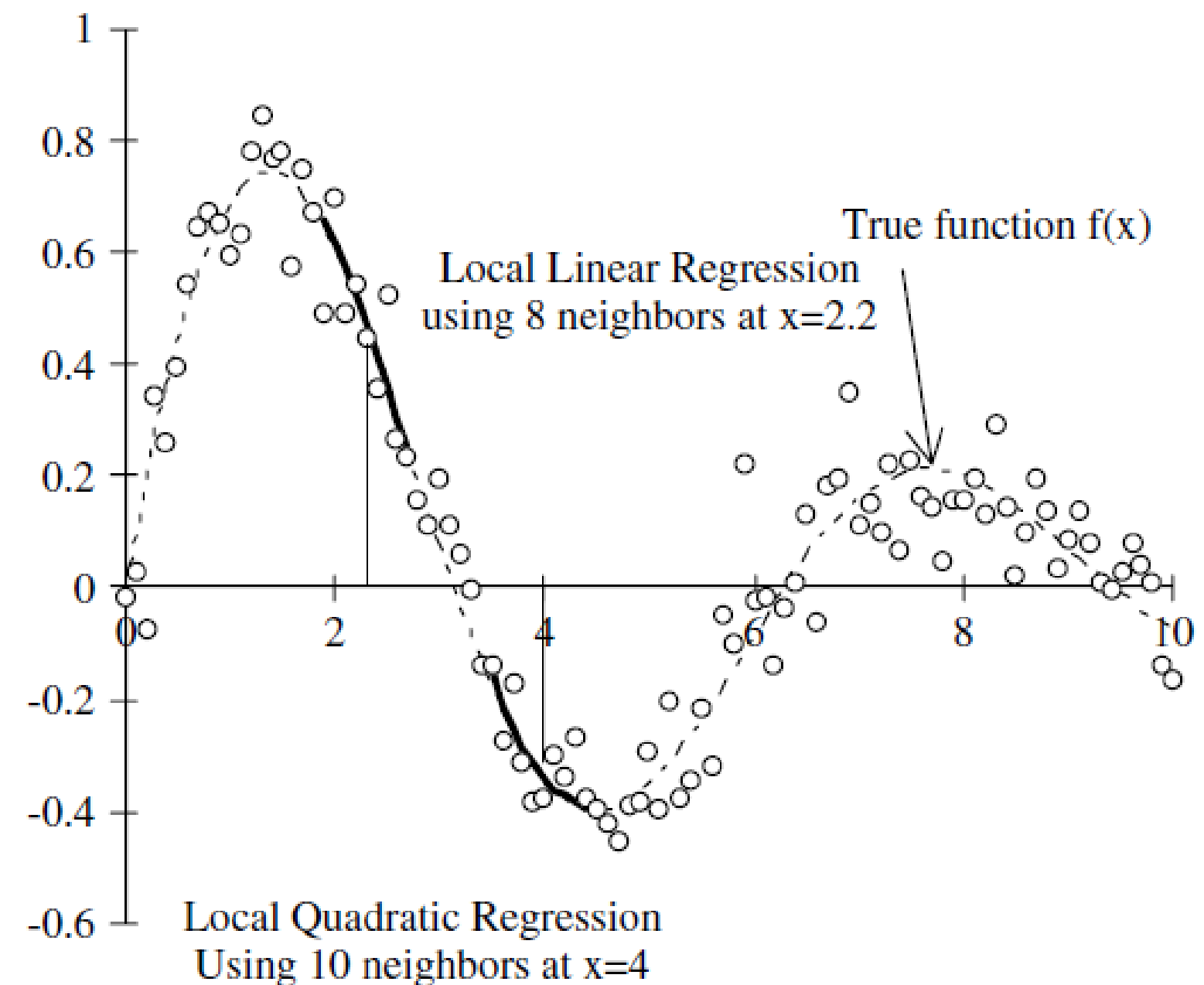
Los pesos son:

$$w_i = \hat{p}_i(1 - \hat{p}_i)$$

La suma de cuadrados residual ponderada es la conocida estadística chi-cuadrado de Pearson.

$$\sum_{i=1}^N \frac{(y_i - \hat{p}_i)^2}{\hat{p}_i(1 - \hat{p}_i)},$$

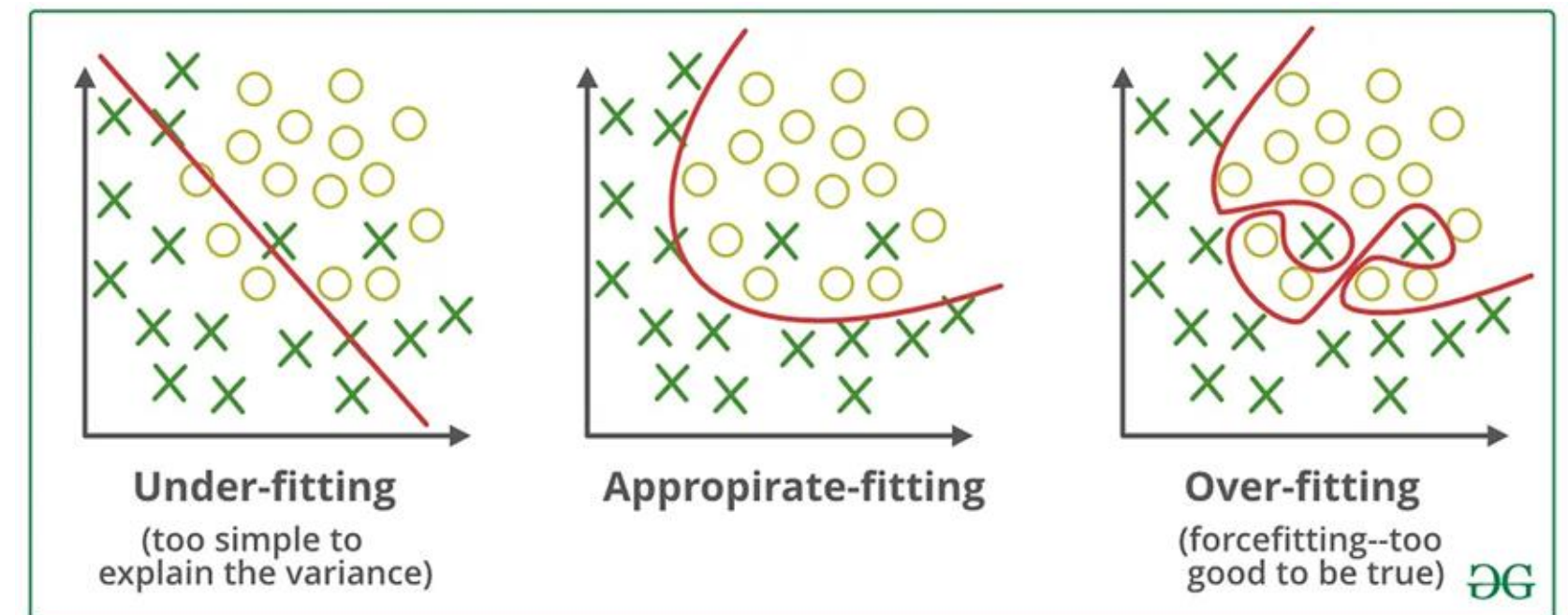
una aproximación cuadrática a la desviación. Si el modelo es correcto,  $\hat{\beta}$  es consistente.



# Regresión logística regularizada $L_1$

La verosimilitud logarítmica:

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^N \left\{ y_i \log p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta)) \right\} \\ &= \sum_{i=1}^N \left\{ y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}) \right\}.\end{aligned}$$



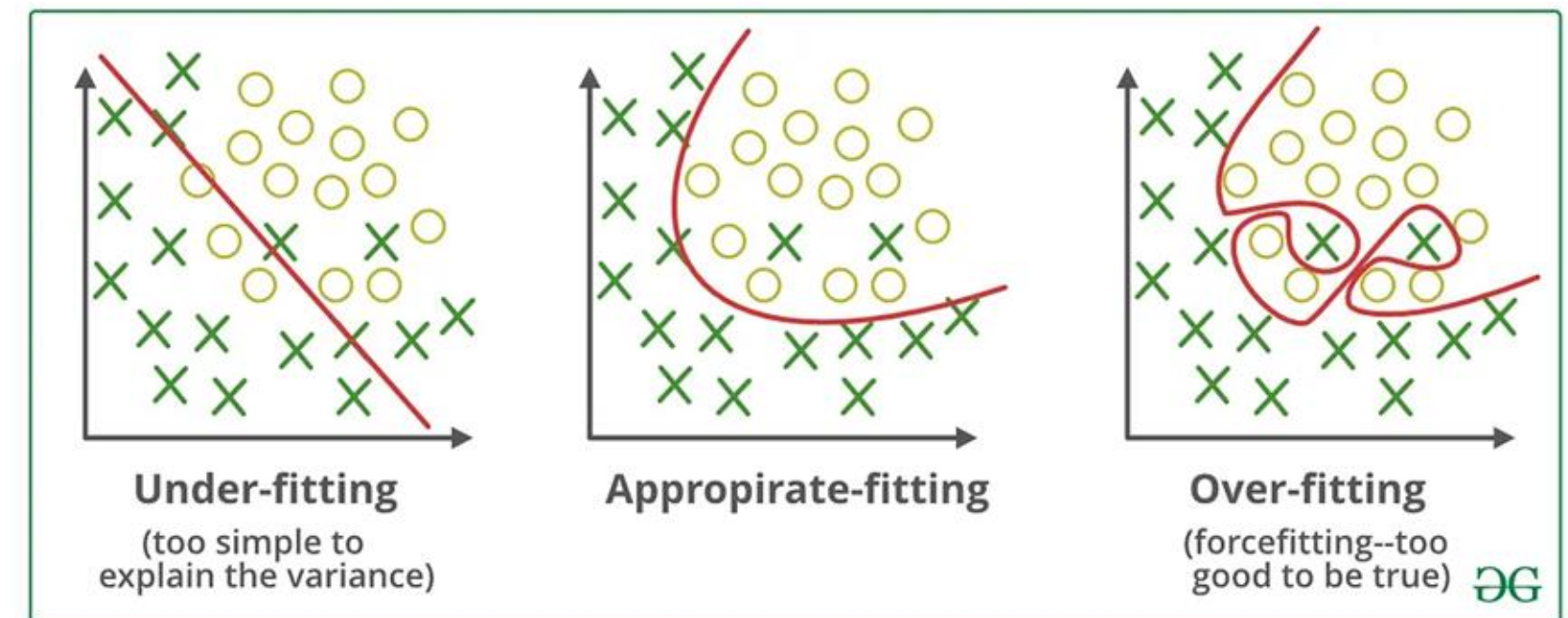
# Regresión logística regularizada $L_1$

La verosimilitud logarítmica:

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^N \left\{ y_i \log p(x_i; \beta) + (1 - y_i) \log(1 - p(x_i; \beta)) \right\} \\ &= \sum_{i=1}^N \left\{ y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}) \right\}.\end{aligned}$$

La penalización  $L_1$  utilizada en el método Lasso. puede utilizarse para la selección de variables y la reducción con cualquier modelo de regresión lineal. Para la regresión logística, maximizaríamos una versión penalizada de verosimilitud logística:

$$\max_{\beta_0, \beta} \left\{ \sum_{i=1}^N \left[ y_i (\beta_0 + \beta^T x_i) - \log(1 + e^{\beta_0 + \beta^T x_i}) \right] - \lambda \sum_{j=1}^p |\beta_j| \right\}.$$



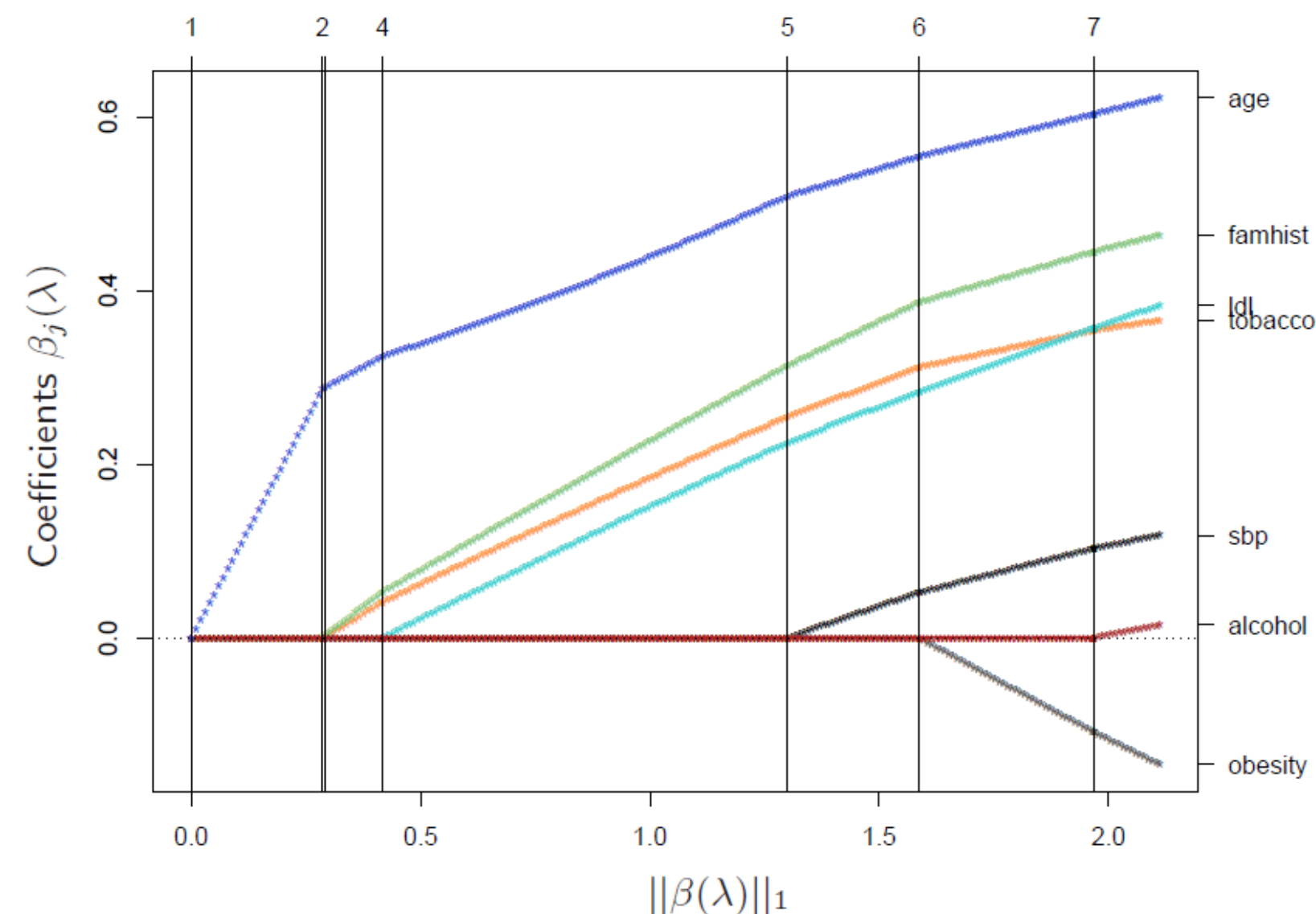
# Regresión logística regularizada $L_1$

Los algoritmos de ruta como LAR para lasso son más difíciles, porque los perfiles de los coeficientes son suaves por partes en lugar de lineales. No obstante, se pueden lograr avances utilizando aproximaciones cuadráticas.

Coeficientes de regresión logística regularizados  $L_1$  para el caso anterior, los datos sobre enfermedades cardíacas en Sudáfrica, representados gráficamente en función de la norma  $L_1$ .

Todas las variables se estandarizaron para tener una varianza unitaria.

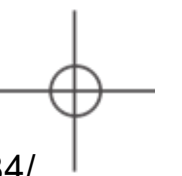
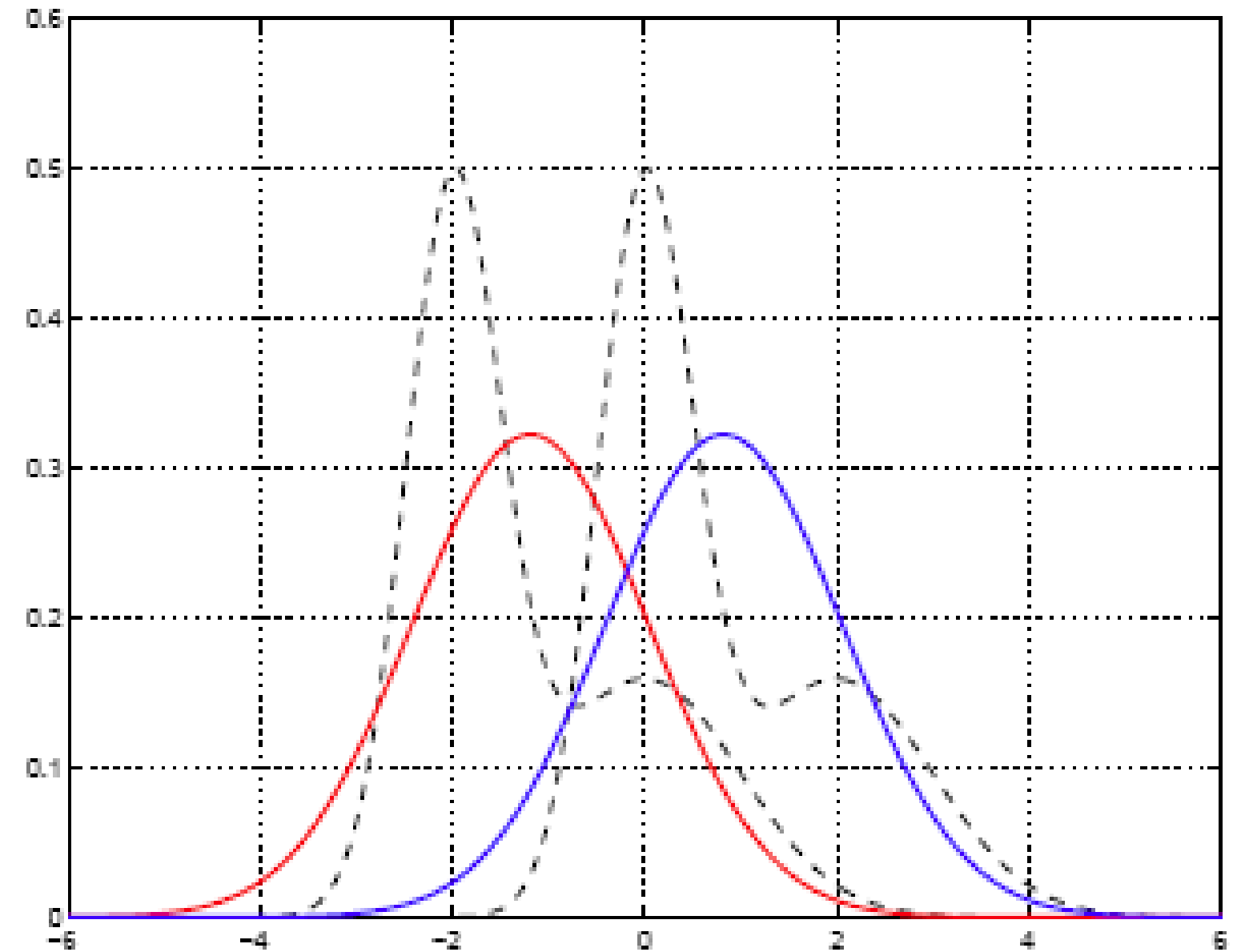
Los perfiles se calculan exactamente en cada uno de los puntos representados gráficamente.



# ¿Regresión logística o LDA?

Encontramos que las probabilidades log-posteriores entre la clase  $k$  y  $K$  son funciones lineales de  $x$

$$\begin{aligned} \log \frac{\Pr(G = k|X = x)}{\Pr(G = K|X = x)} &= \log \frac{\pi_k}{\pi_K} - \frac{1}{2}(\mu_k + \mu_K)^T \Sigma^{-1}(\mu_k - \mu_K) \\ &\quad + x^T \Sigma^{-1}(\mu_k - \mu_K) \\ &= \alpha_{k0} + \alpha_k^T x. \end{aligned}$$



# ¿Regresión logística o LDA?

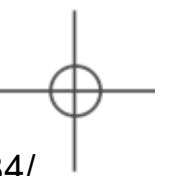
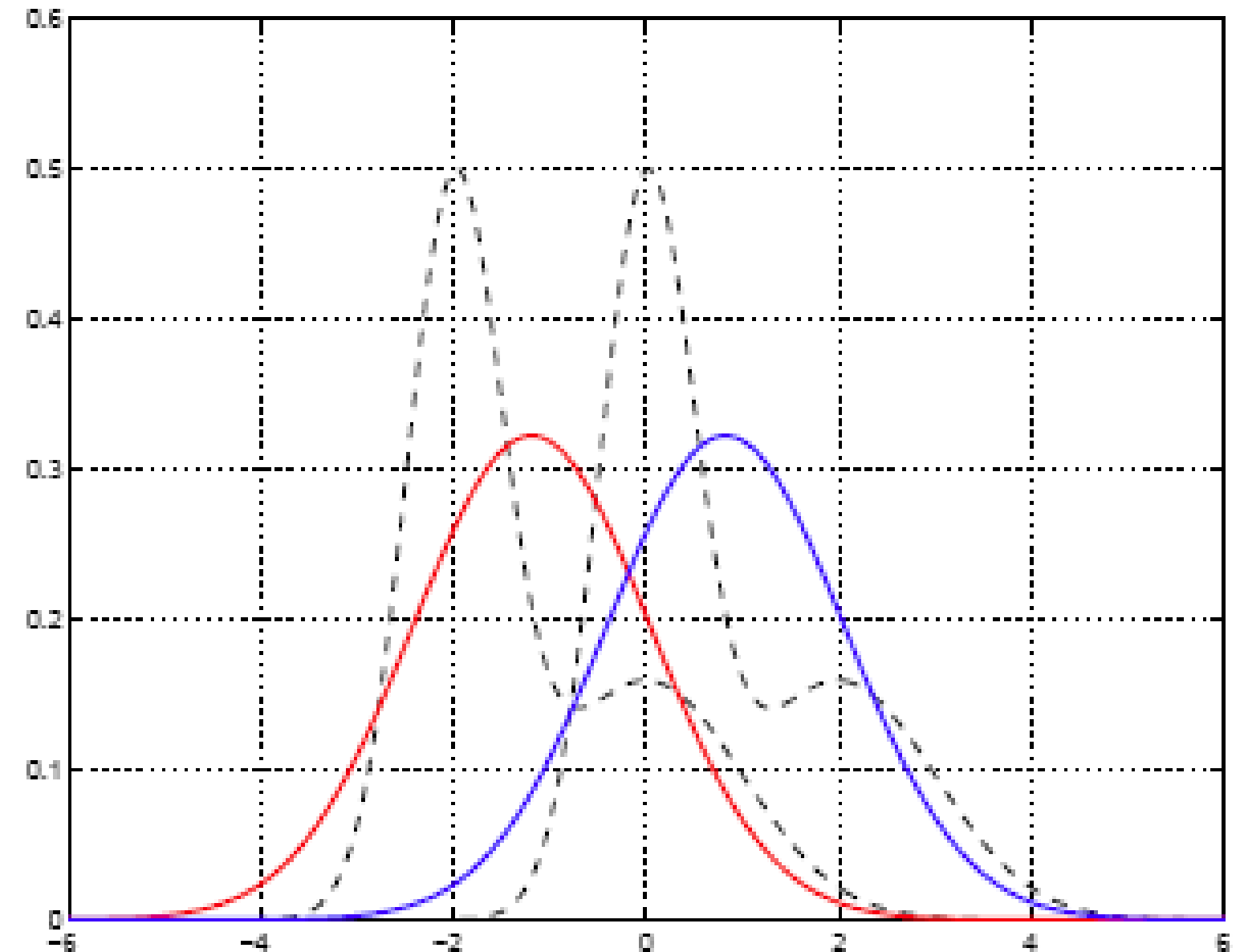
Encontramos que las probabilidades log-posteriores entre la clase  $k$  y  $K$  son funciones lineales de  $x$

$$\begin{aligned} \log \frac{\Pr(G = k|X = x)}{\Pr(G = K|X = x)} &= \log \frac{\pi_k}{\pi_K} - \frac{1}{2}(\mu_k + \mu_K)^T \Sigma^{-1}(\mu_k - \mu_K) \\ &\quad + x^T \Sigma^{-1}(\mu_k - \mu_K) \\ &= \alpha_{k0} + \alpha_k^T x. \end{aligned}$$

El modelo logístico lineal por construcción tiene logits lineales:

$$\log \frac{\Pr(G = k|X = x)}{\Pr(G = K|X = x)} = \beta_{k0} + \beta_k^T x.$$

Parece que los modelos son iguales, la diferencia radica en la forma en que se estiman los coeficientes lineales. El modelo de regresión logística es más general, ya que hace menos suposiciones

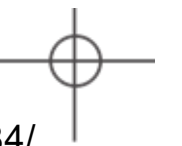
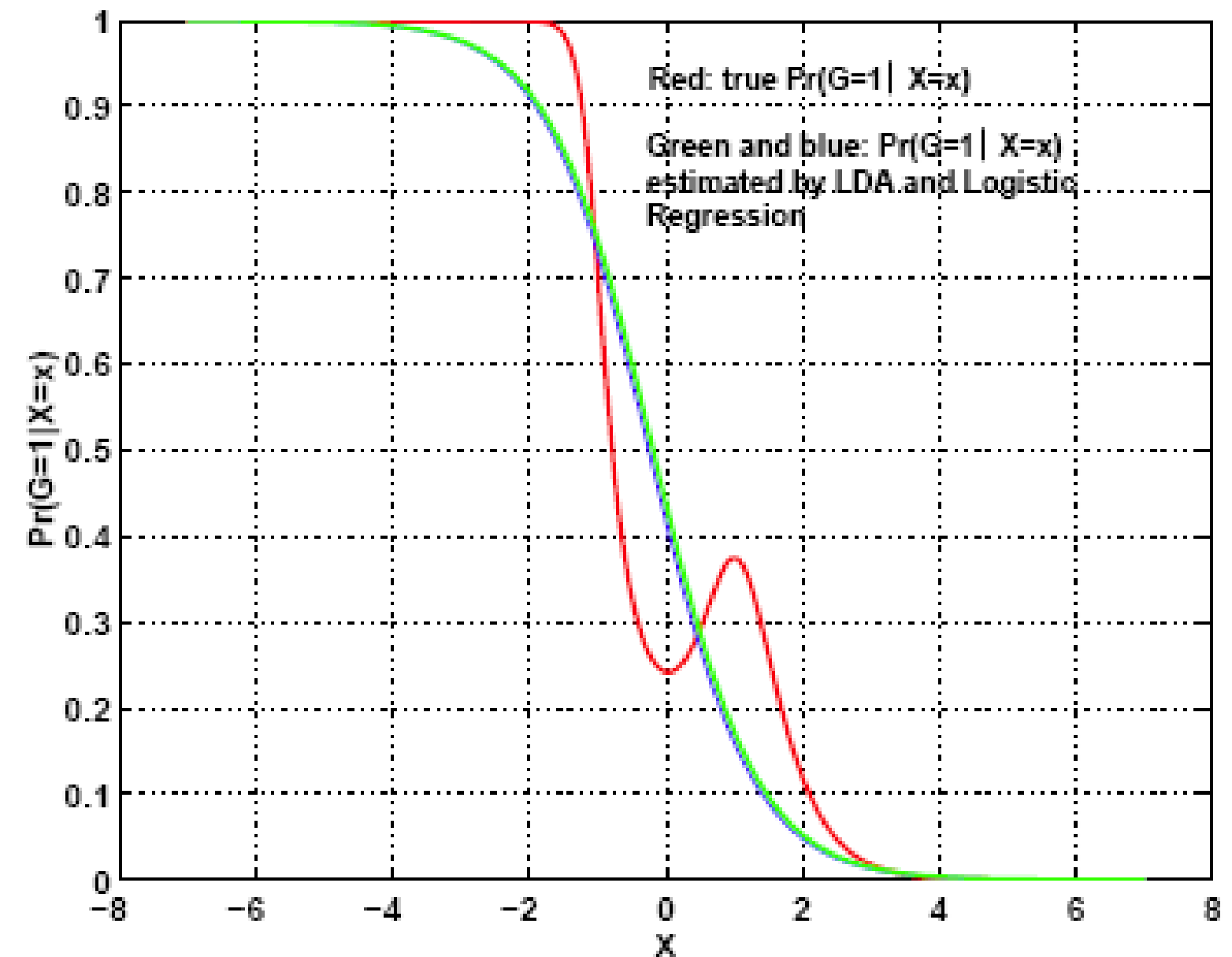


# ¿Regresión logística o LDA?

Podemos escribir la densidad conjunta de  $X$  y  $G$  como

$$\Pr(X, G = k) = \Pr(X)\Pr(G = k|X),$$

$\Pr(X)$  denota la densidad marginal de las entradas  $X$ .



# ¿Regresión logística o LDA?

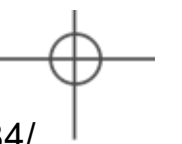
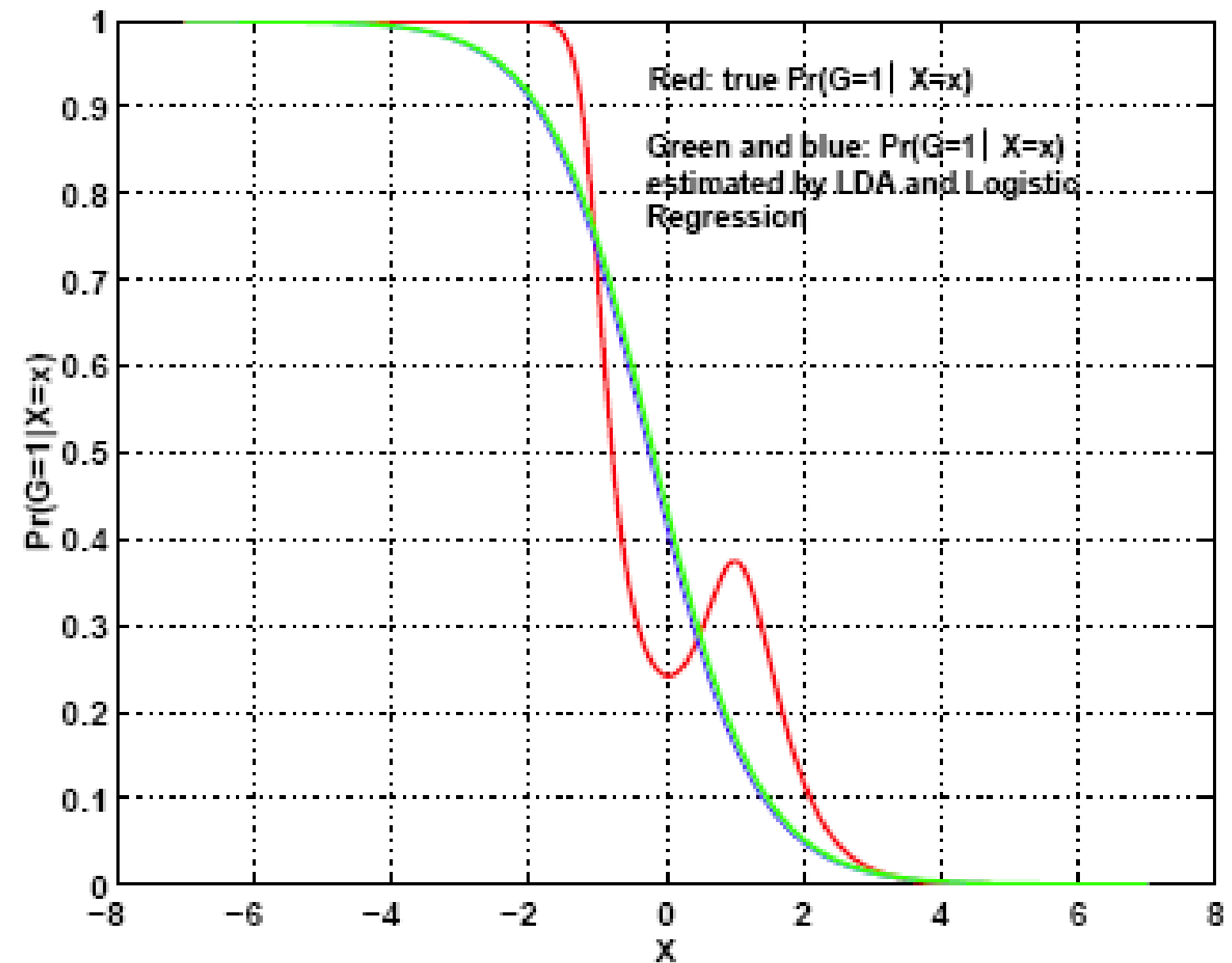
Podemos escribir la densidad conjunta de  $X$  y  $G$  como

$$\Pr(X, G = k) = \Pr(X)\Pr(G = k|X),$$

$\Pr(X)$  denota la densidad marginal de las entradas  $X$ .

Tanto para LDA como para la regresión logística, el segundo término a la derecha tiene la forma logit-lineal

$$\Pr(G = k|X = x) = \frac{e^{\beta_{k0} + \beta_k^T x}}{1 + \sum_{\ell=1}^{K-1} e^{\beta_{\ell 0} + \beta_{\ell}^T x}},$$



# ¿Regresión logística o LDA?

Podemos escribir la densidad conjunta de  $X$  y  $G$  como

$$\Pr(X, G = k) = \Pr(X)\Pr(G = k|X),$$

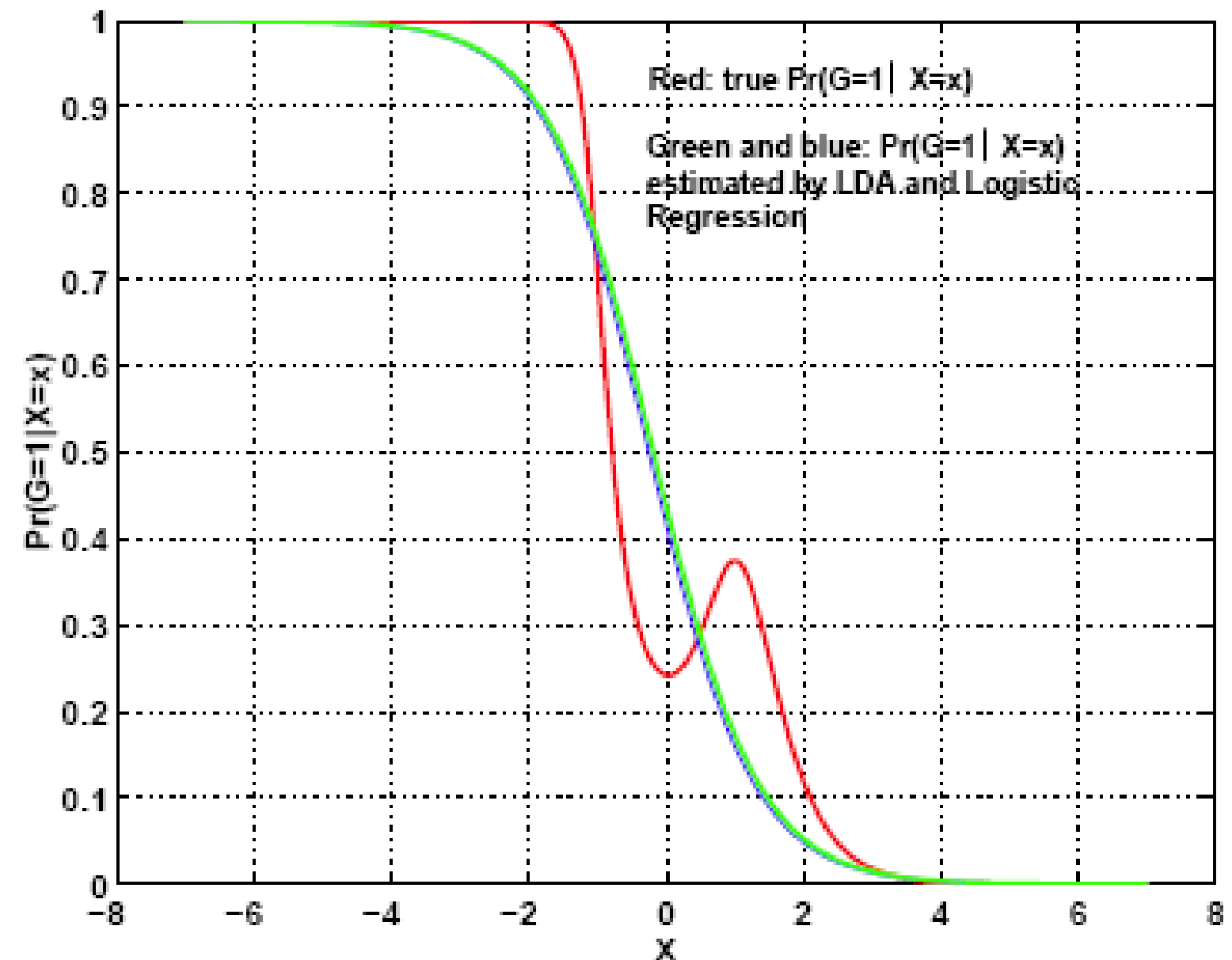
$\Pr(X)$  denota la densidad marginal de las entradas  $X$ .

Tanto para LDA como para la regresión logística, el segundo término a la derecha tiene la forma logit-lineal

$$\Pr(G = k|X = x) = \frac{e^{\beta_{k0} + \beta_k^T x}}{1 + \sum_{\ell=1}^{K-1} e^{\beta_{\ell 0} + \beta_{\ell}^T x}},$$

Con LDA ajustamos los parámetros maximizando la log-verosimilitud completa, basándonos en la densidad conjunta.

$$\Pr(X, G = k) = \phi(X; \mu_k, \Sigma) \pi_k,$$



# ¿Regresión logística o LDA?

Podemos escribir la densidad conjunta de  $X$  y  $G$  como

$$\Pr(X, G = k) = \Pr(X)\Pr(G = k|X),$$

$\Pr(X)$  denota la densidad marginal de las entradas  $X$ .

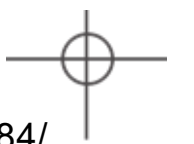
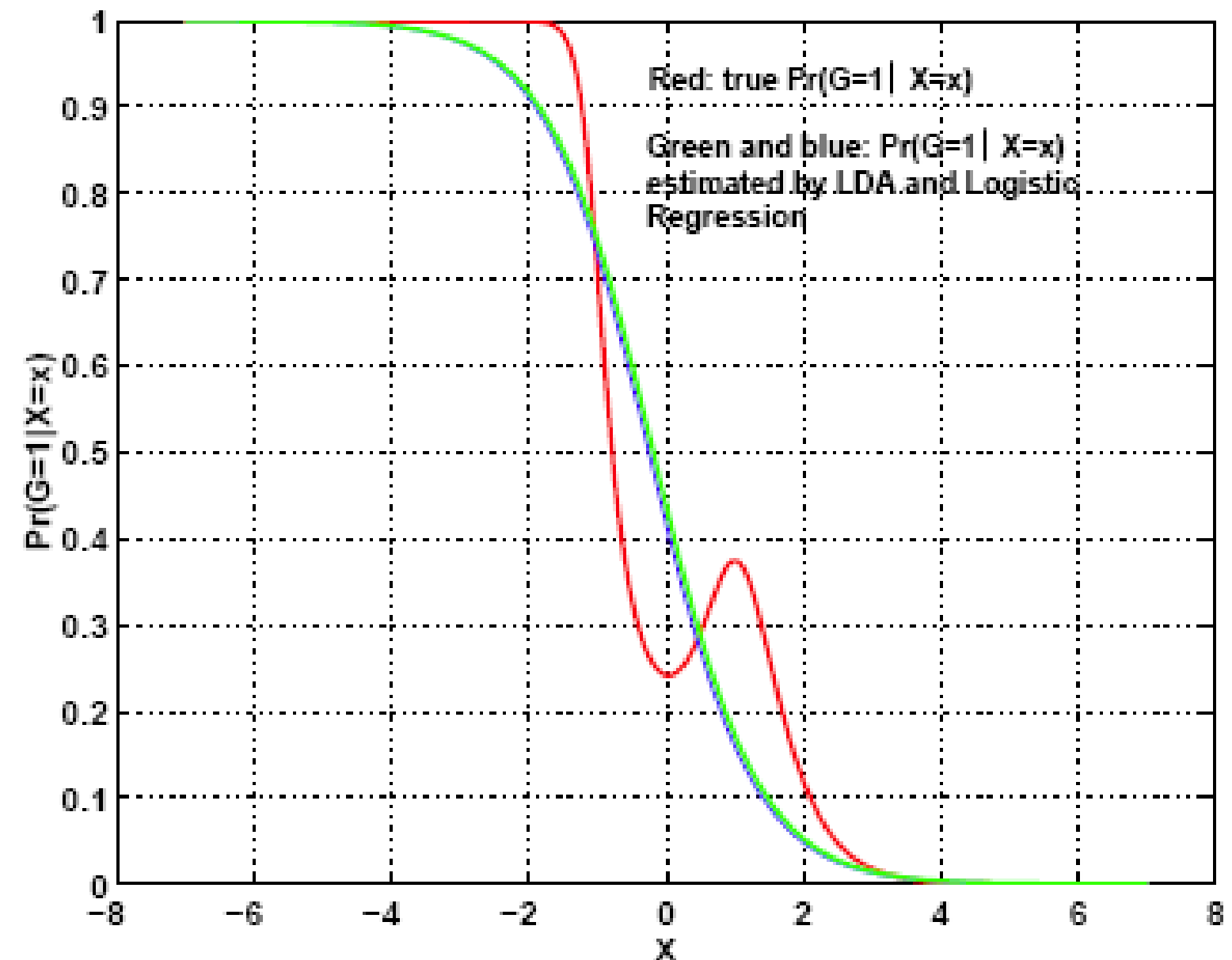
Tanto para LDA como para la regresión logística, el segundo término a la derecha tiene la forma logit-lineal

$$\Pr(G = k|X = x) = \frac{e^{\beta_{k0} + \beta_k^T x}}{1 + \sum_{\ell=1}^{K-1} e^{\beta_{\ell 0} + \beta_{\ell}^T x}},$$

Con LDA ajustamos los parámetros maximizando la log-verosimilitud completa, basándonos en la densidad conjunta.

$$\Pr(X, G = k) = \phi(X; \mu_k, \Sigma) \pi_k,$$

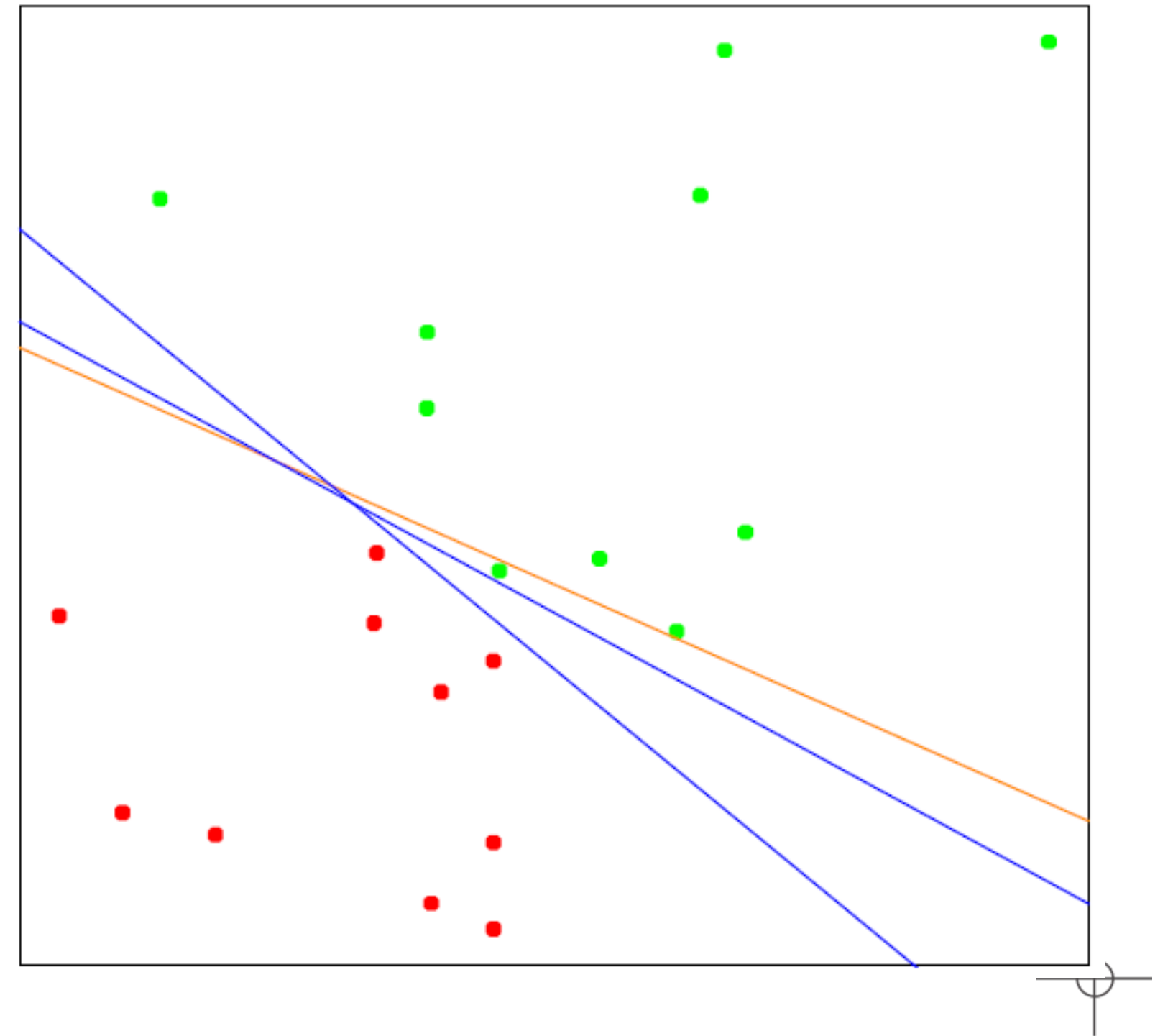
donde  $\phi$  es la función de densidad gaussiana



# ¿Regresión logística o LDA?

La densidad marginal  $\Pr(X)$  es una densidad mixta

$$\Pr(X) = \sum_{k=1}^K \pi_k \phi(X; \mu_k, \Sigma),$$



# ¿Regresión logística o LDA?

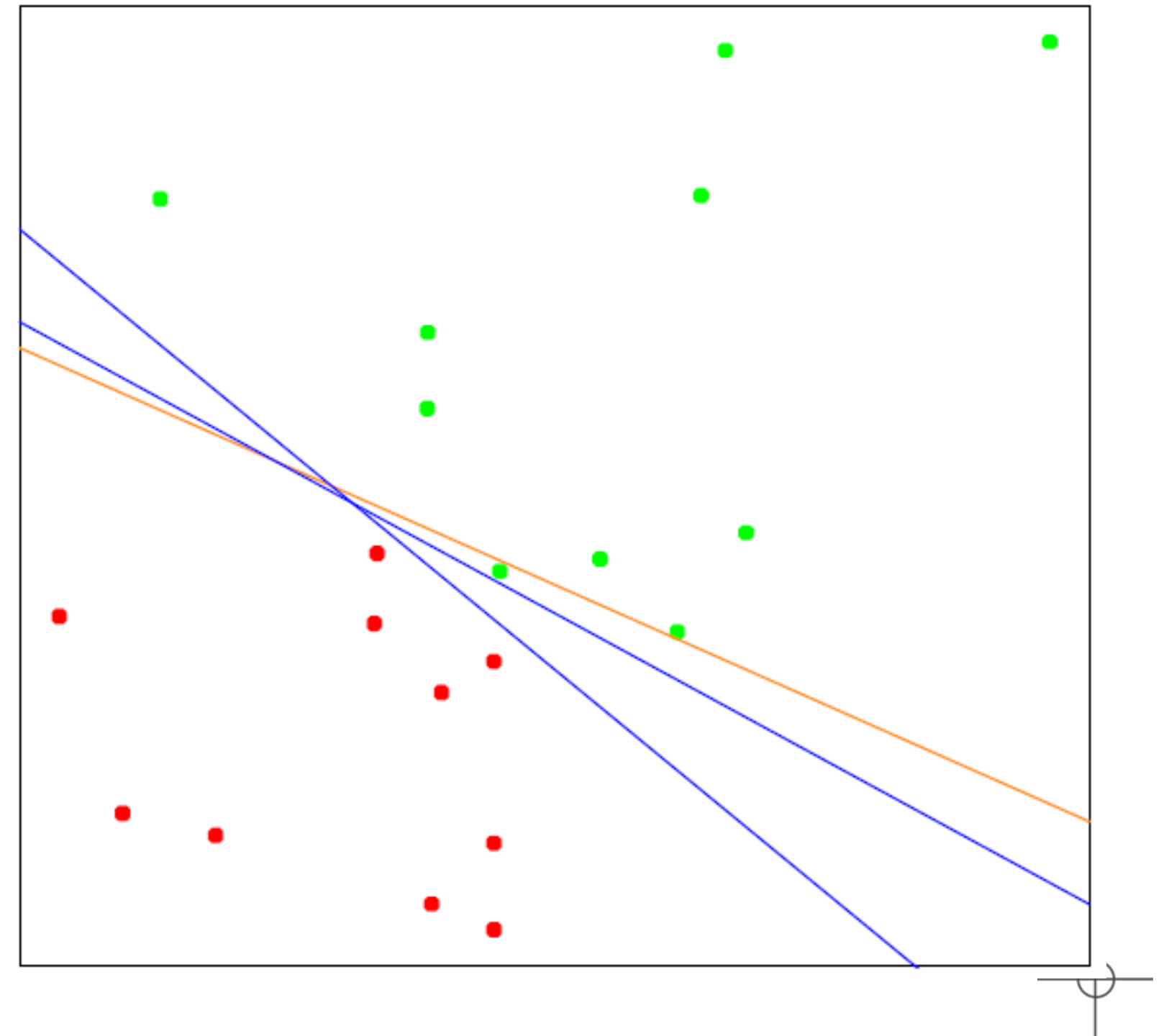
La densidad marginal  $\Pr(X)$  es una densidad mixta

$$\Pr(X) = \sum_{k=1}^K \pi_k \phi(X; \mu_k, \Sigma),$$

En el gráfico se muestra un ejemplo sencillo con dos clases separables por un hiperplano.

La línea naranja es la solución de mínimos cuadrados, que clasifica erróneamente uno de los puntos de entrenamiento.

También se muestran dos hiperplanos separadores azules encontrados por el algoritmo de aprendizaje perceptrón con diferentes inicios aleatorios.

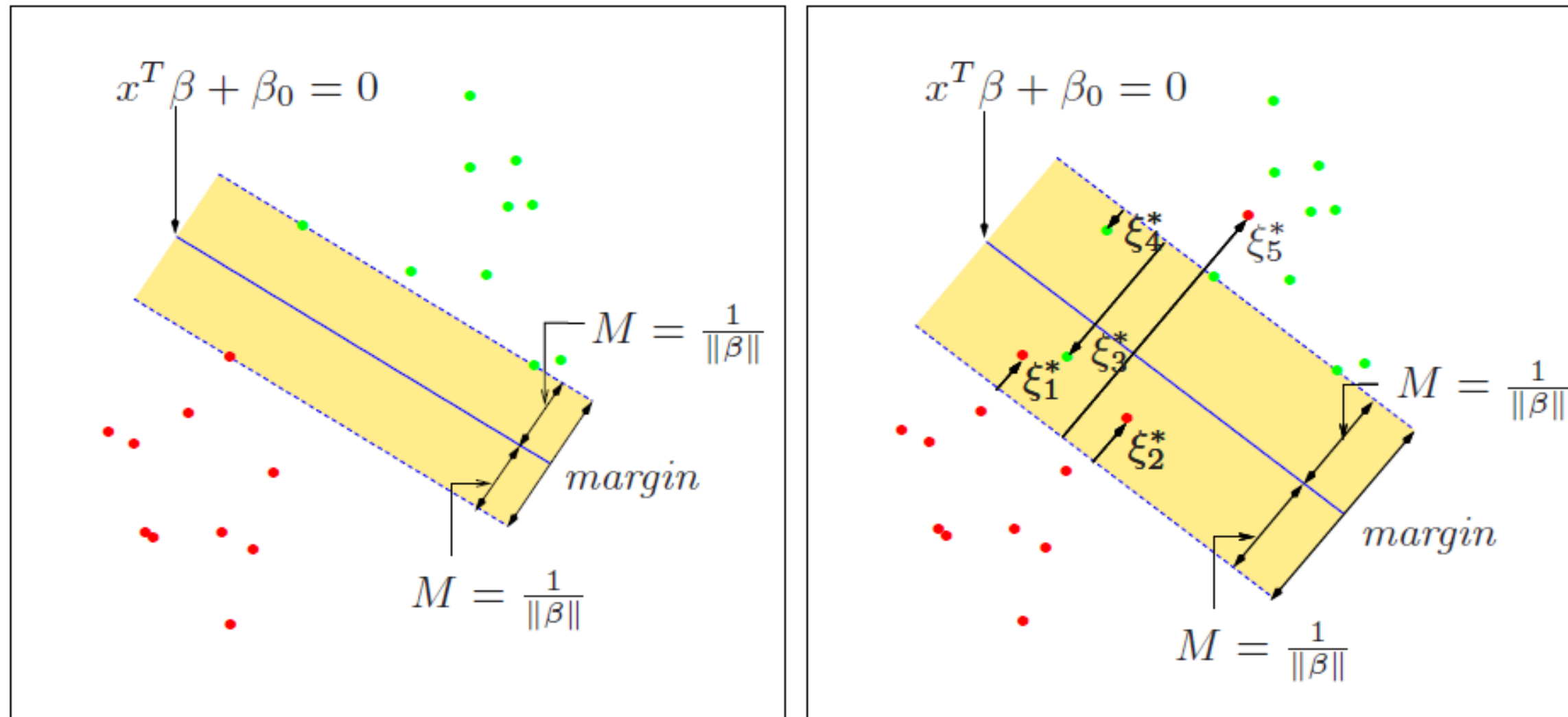


# Support Vector Machines & Flexible Discriminants

Descripción



# The Support Vector Classifier



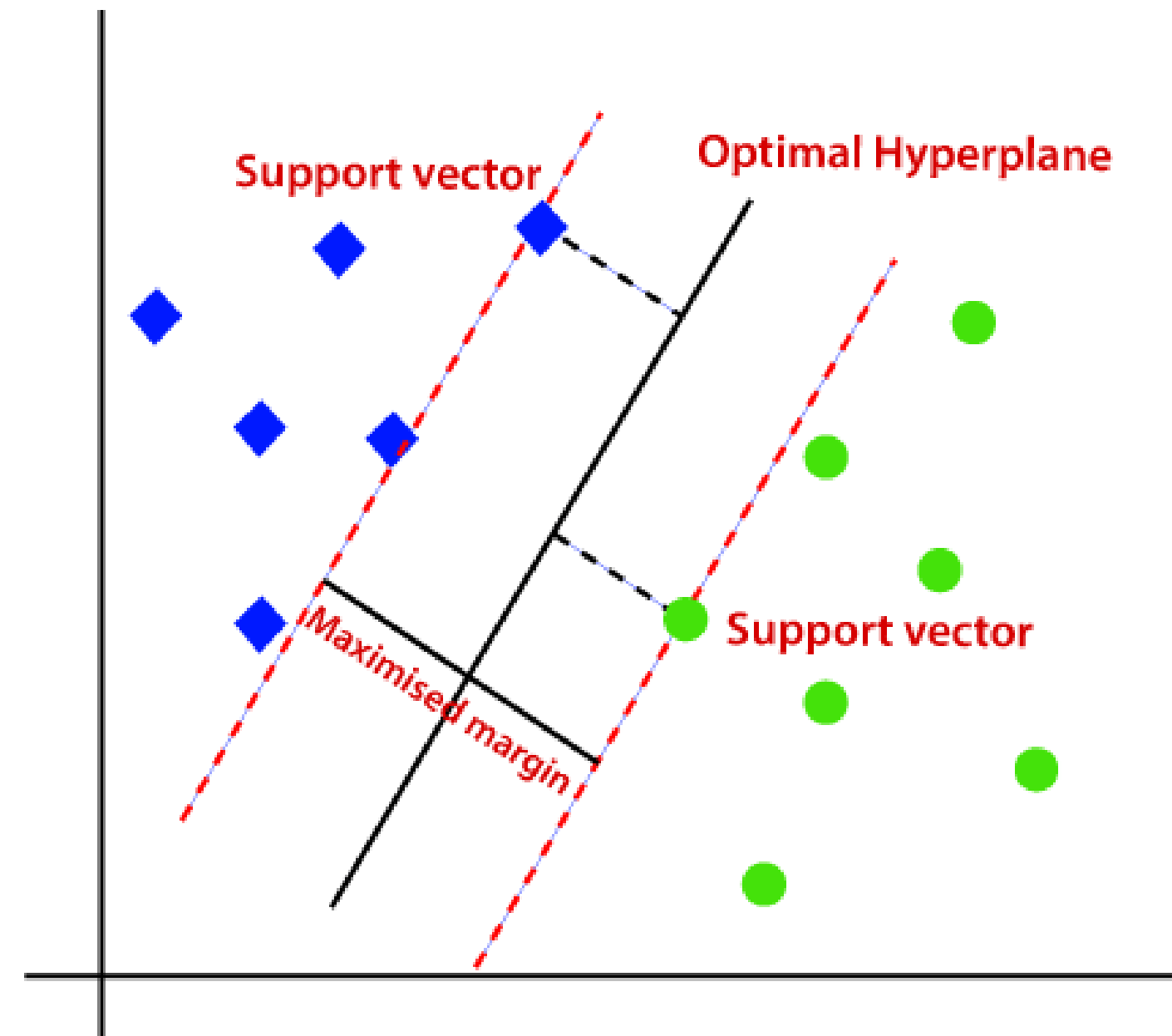
Clasificadores de vectores de soporte. El panel izquierdo muestra el caso separable. El límite de decisión es la línea continua, mientras que las líneas discontinuas delimitan el margen máximo sombreado de anchura  $2M = 2/\|\beta\|$ . El panel derecho muestra el caso no separable (superposición). Los puntos etiquetados como  $\xi_j^*$  se encuentran en el lado incorrecto de su margen en una cantidad  $\xi_j^* = M\xi_j$ .



# The Support Vector Classifier

Nuestros datos de entrenamiento consisten en  $N$  pares  $(x_1, y_1)$ ,  $(x_2, y_2), \dots, (x_N, y_N)$ , con  $x_i \in \mathbb{R}^p$  y  $y_i \in \{-1, 1\}$ . Define un hiperplano mediante:

$$\{x : f(x) = x^T \beta + \beta_0 = 0\},$$

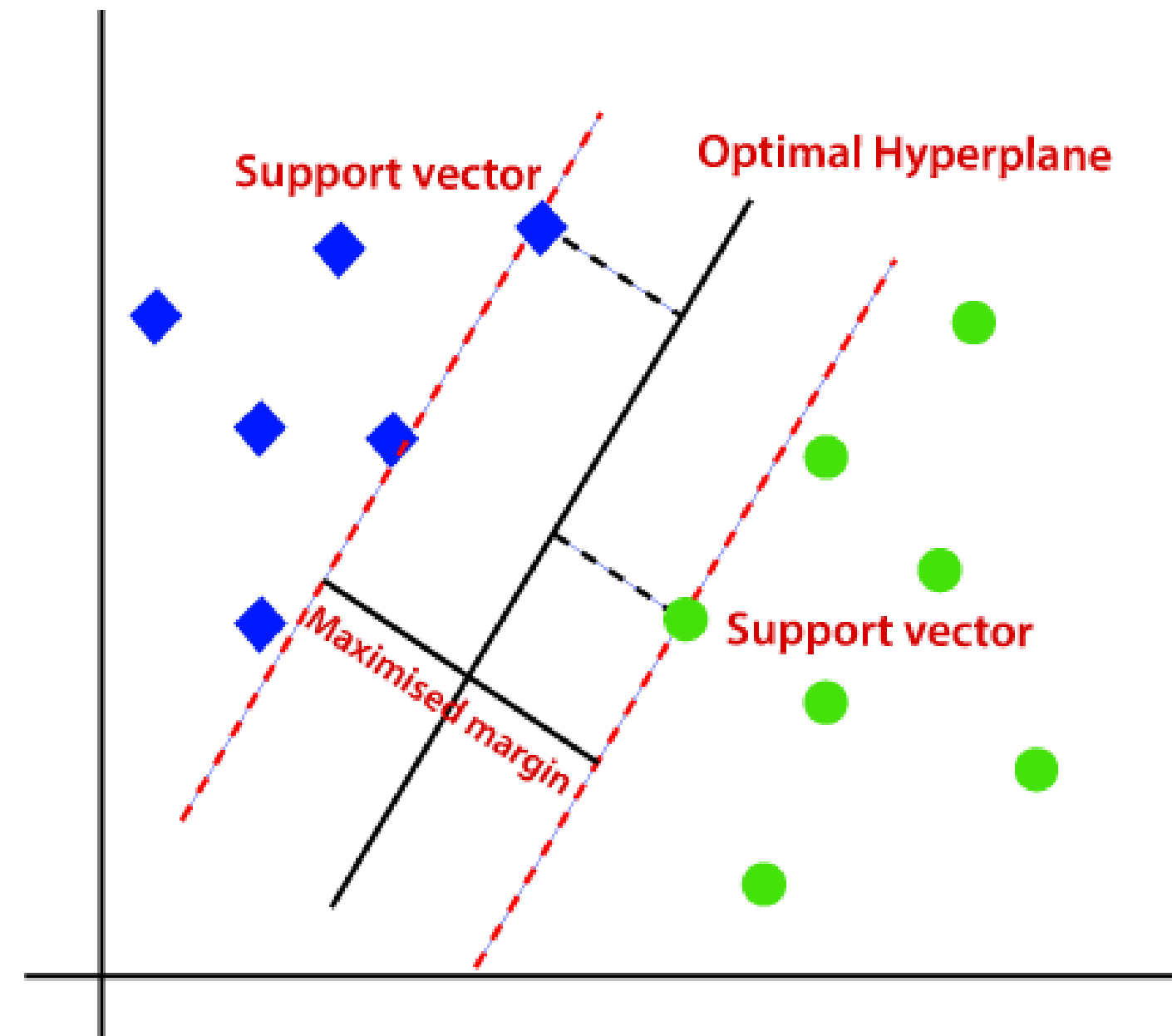


# The Support Vector Classifier

Nuestros datos de entrenamiento consisten en  $N$  pares  $(x_1, y_1)$ ,  $(x_2, y_2), \dots, (x_N, y_N)$ , con  $x_i \in \mathbb{R}^p$  y  $y_i \in \{-1, 1\}$ . Define un hiperplano mediante:

$$\{x : f(x) = x^T \beta + \beta_0 = 0\},$$

donde  $\beta$  es un vector unitario:  $\|\beta\| = 1$ .



# The Support Vector Classifier

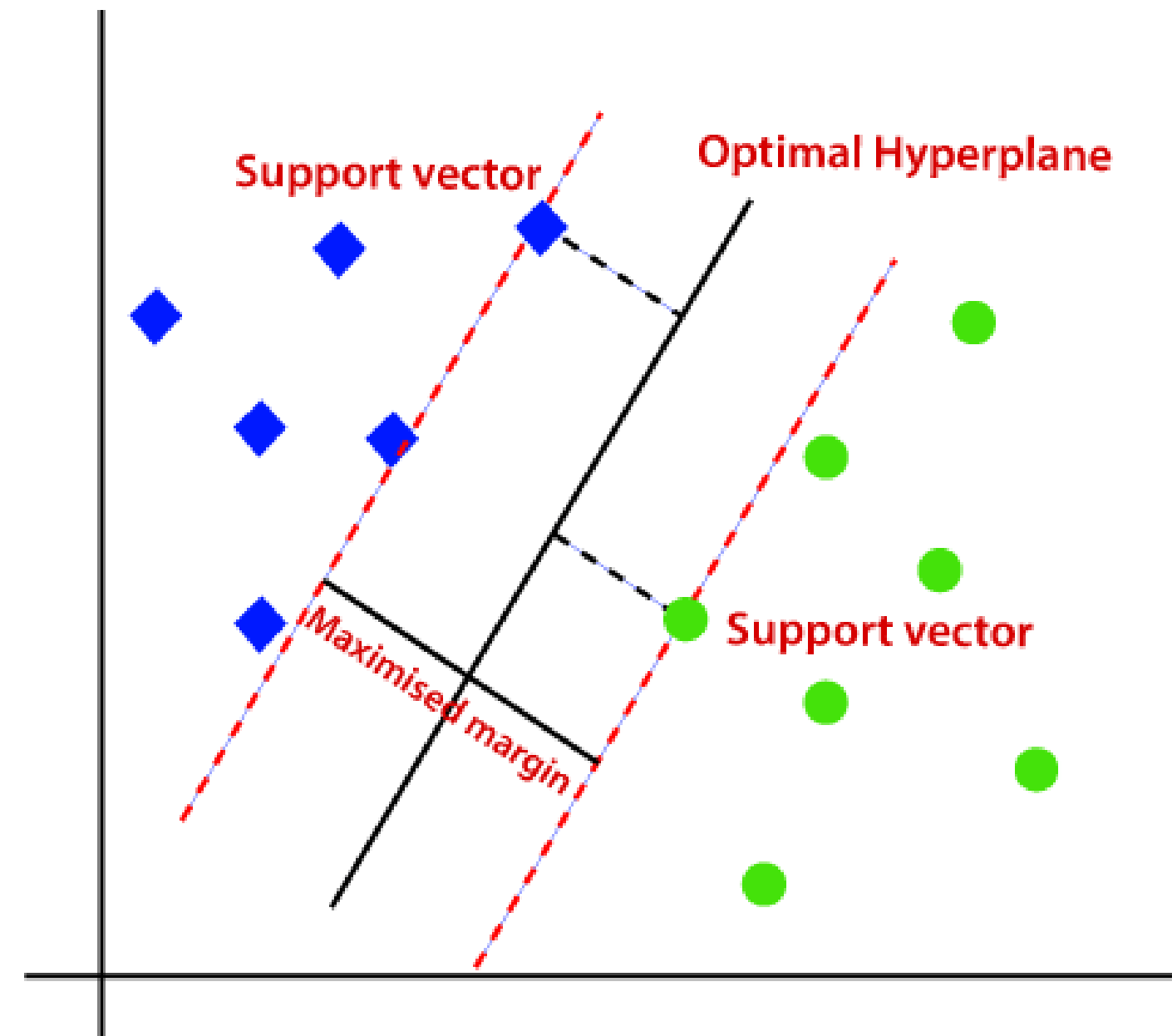
Nuestros datos de entrenamiento consisten en  $N$  pares  $(x_1, y_1)$ ,  $(x_2, y_2), \dots, (x_N, y_N)$ , con  $x_i \in \mathbb{R}^p$  y  $y_i \in \{-1, 1\}$ . Define un hiperplano mediante:

$$\{x : f(x) = x^T \beta + \beta_0 = 0\},$$

donde  $\beta$  es un vector unitario:  $\|\beta\| = 1$ .

Una regla de clasificación inducida por  $f(x)$  es

$$G(x) = \text{sign}[x^T \beta + \beta_0].$$



# The Support Vector Classifier

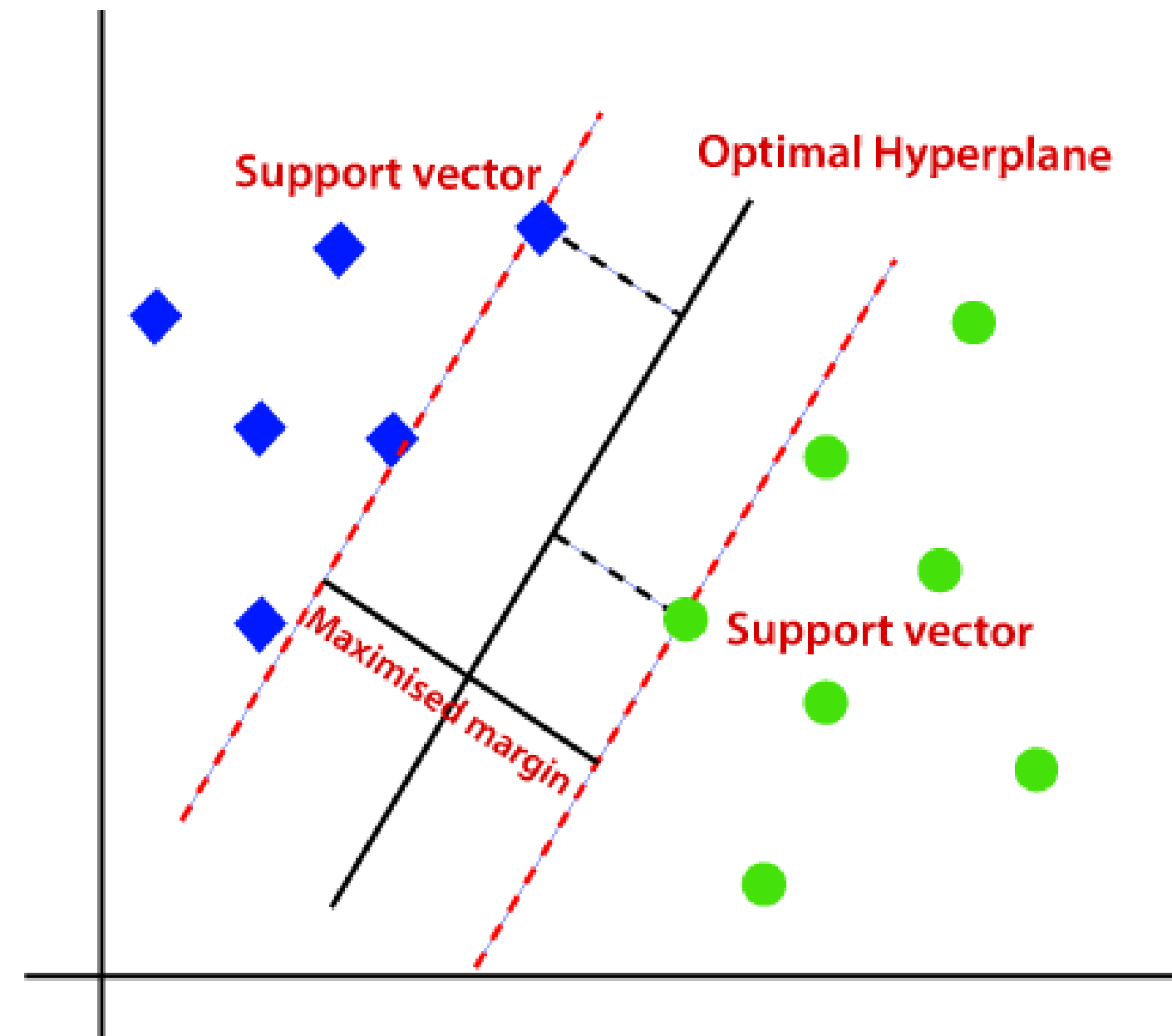
Dado que las clases son separables, podemos encontrar una función:

$$f(x) = x^T \beta + \beta_0 \text{ with } y_i f(x_i) > 0 \quad \forall i.$$

Por lo tanto, podemos encontrar el hiperplano que crea el mayor margen entre los puntos de entrenamiento para la clase 1 y -1.

El problema de optimización captura este concepto.

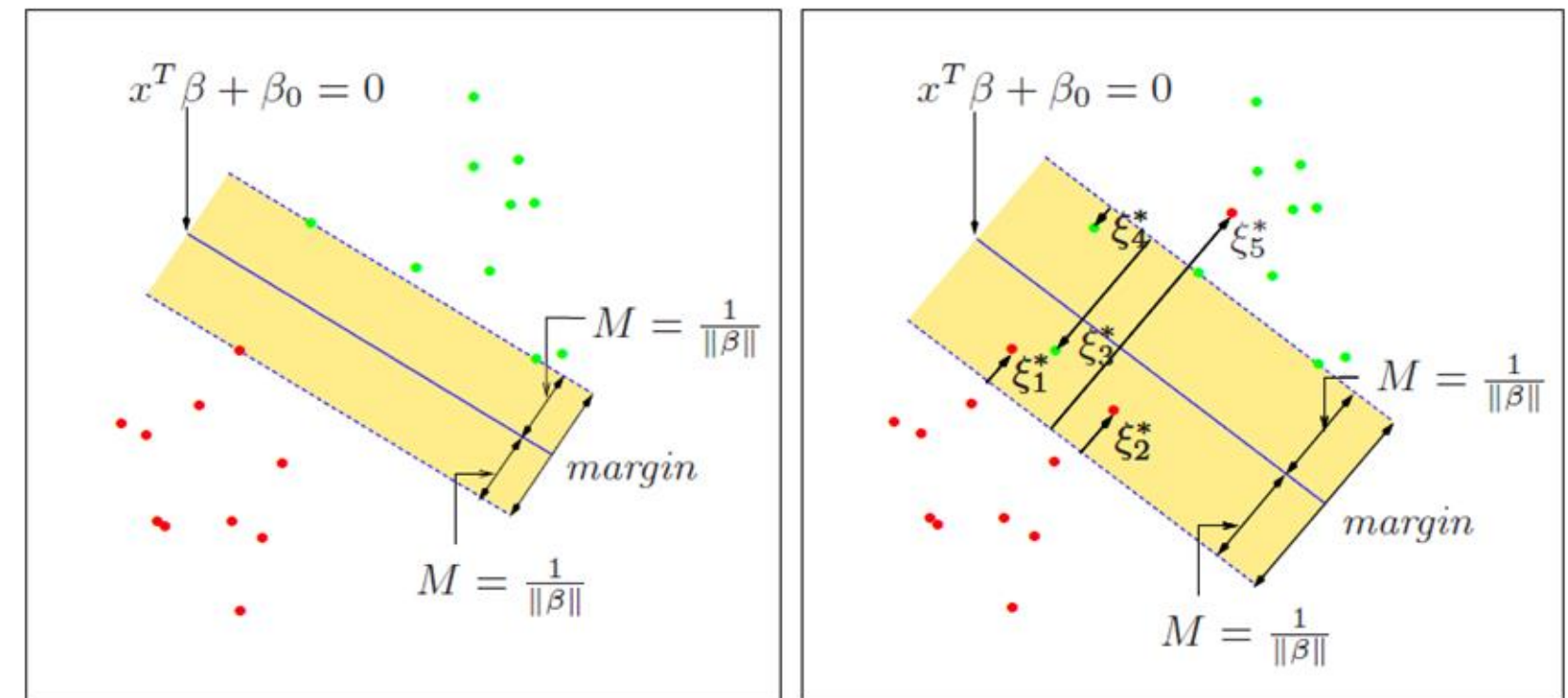
$$\begin{aligned} & \max_{\beta, \beta_0, \|\beta\|=1} M \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq M, \quad i = 1, \dots, N, \end{aligned}$$



# The Support Vector Classifier

La banda de la figura está a  $M$  unidades de distancia del hiperplano a ambos lados y, por lo tanto, tiene  $2M$  unidades de ancho.

Se denomina margen. Hemos demostrado que este problema se puede reformular de manera más conveniente como



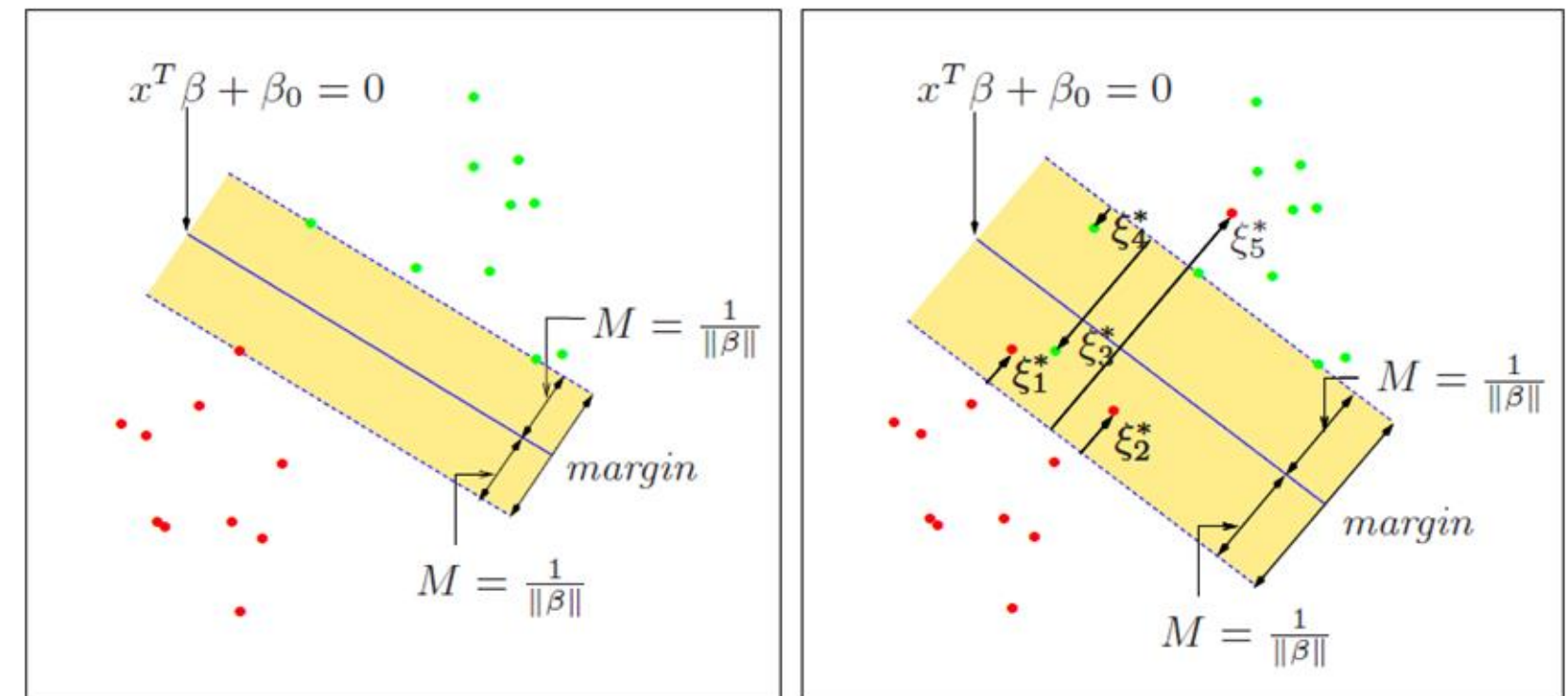
# The Support Vector Classifier

La banda de la figura está a  $M$  unidades de distancia del hiperplano a ambos lados y, por lo tanto, tiene  $2M$  unidades de ancho.

Se denomina *margen*. Hemos demostrado que este problema se puede reformular de manera más conveniente como

$$\begin{aligned} & \min_{\beta, \beta_0} \|\beta\| \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq 1, \quad i = 1, \dots, N, \end{aligned}$$

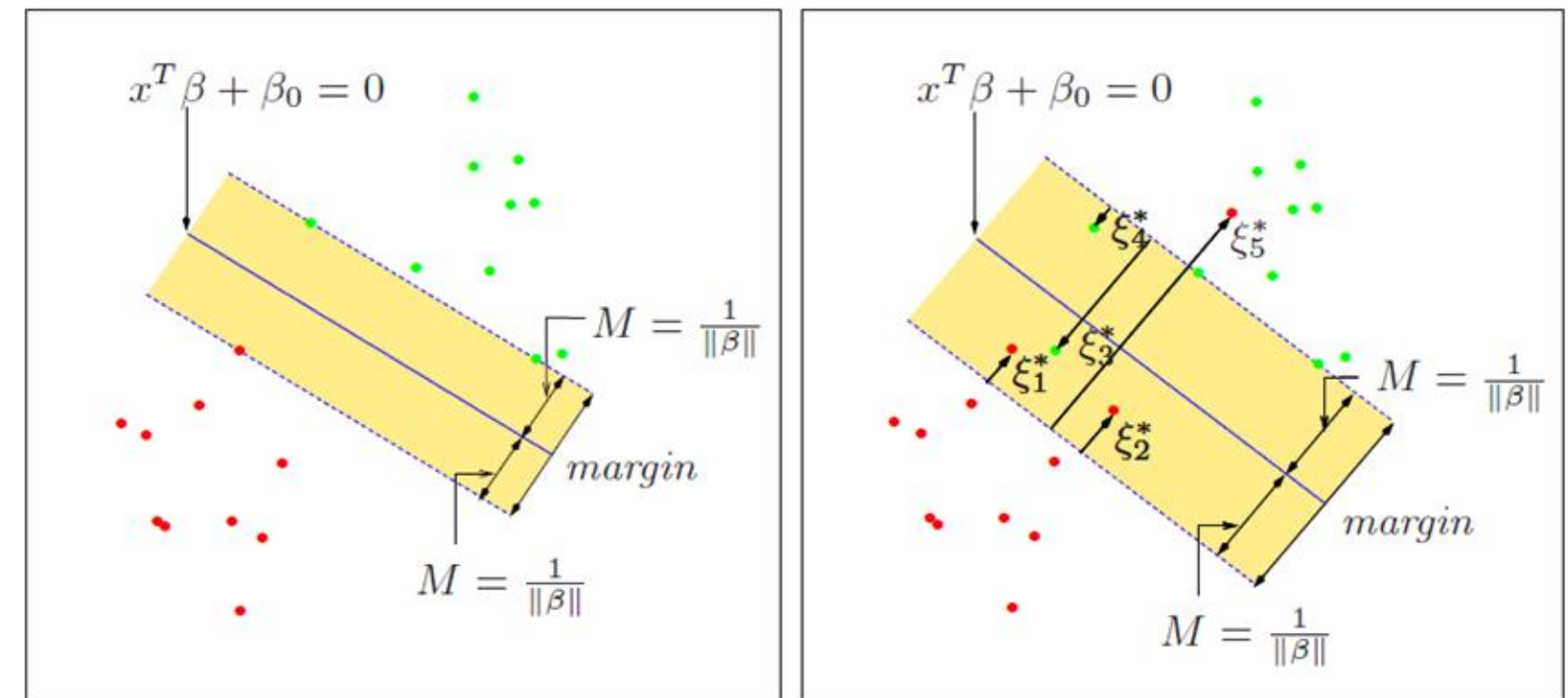
Se trata de un problema de optimización convexa (criterio cuadrático, restricciones de desigualdad lineal).



# The Support Vector Classifier

Supongamos ahora que las clases se superponen en el espacio de características.

Una forma de abordar la superposición es seguir maximizando  $M$ , pero permitiendo que algunos puntos se encuentren en el lado incorrecto del margen. Defina las variables de holgura  $\xi = (\xi_1, \xi_2, \dots, \xi_N)$ .



# The Support Vector Classifier

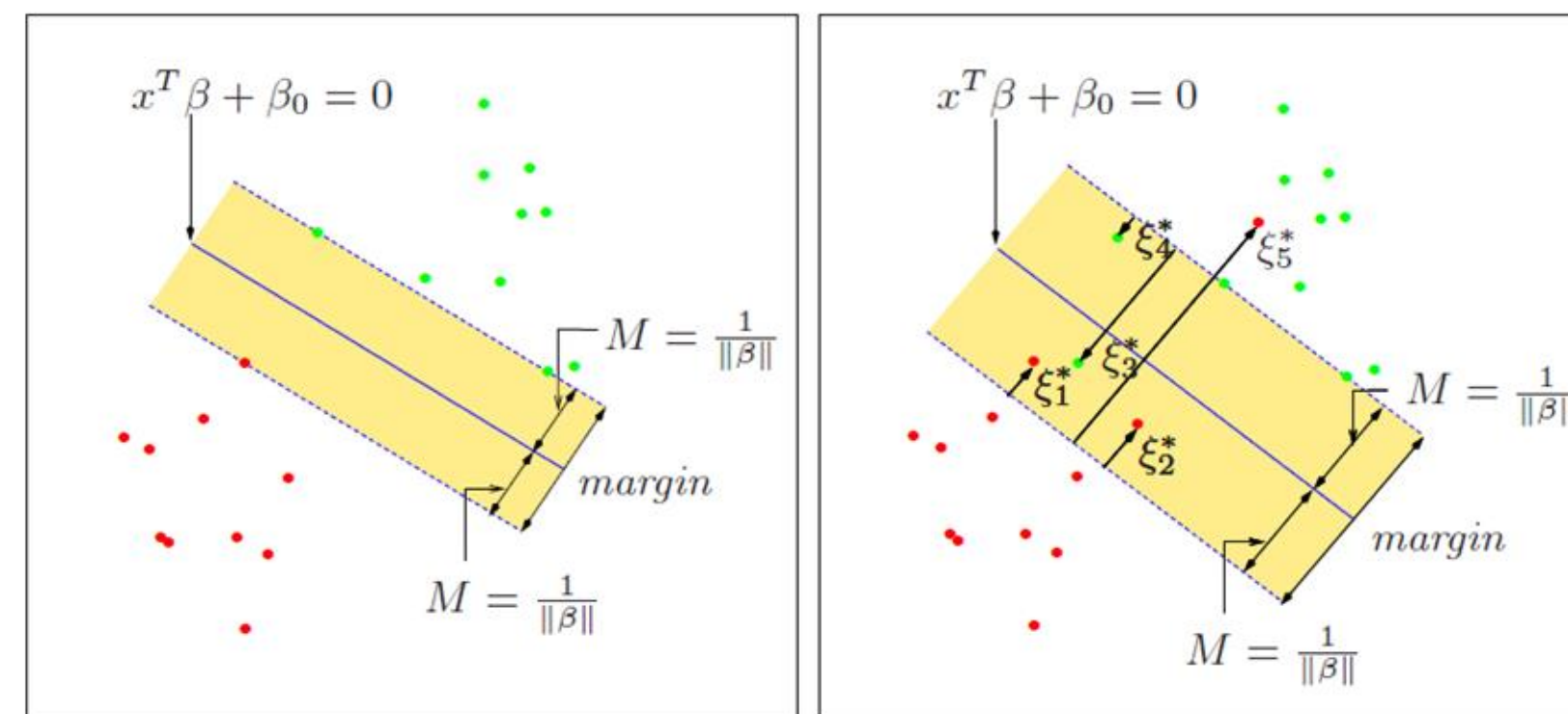
Supongamos ahora que las clases se superponen en el espacio de características.

Una forma de abordar la superposición es seguir maximizando  $M$ , pero permitiendo que algunos puntos se encuentren en el lado incorrecto del margen. Defina las variables de holgura  $\xi = (\xi_1, \xi_2, \dots, \xi_N)$ .

$$y_i(x_i^T \beta + \beta_0) \geq M - \xi_i,$$

or

$$y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i),$$



# The Support Vector Classifier

Supongamos ahora que las clases se superponen en el espacio de características.

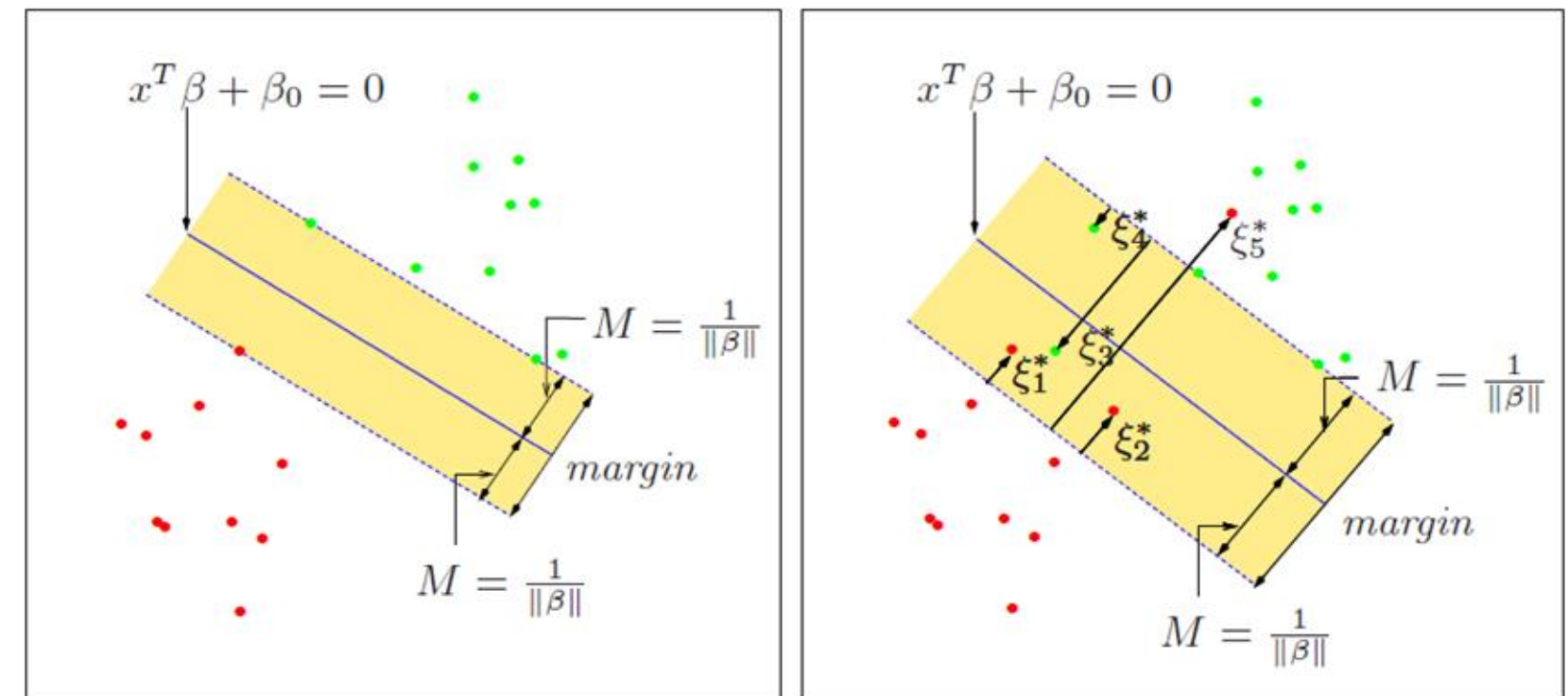
Una forma de abordar la superposición es seguir maximizando  $M$ , pero permitiendo que algunos puntos se encuentren en el lado incorrecto del margen. Defina las variables de holgura  $\xi = (\xi_1, \xi_2, \dots, \xi_N)$ .

$$y_i(x_i^T \beta + \beta_0) \geq M - \xi_i,$$

or

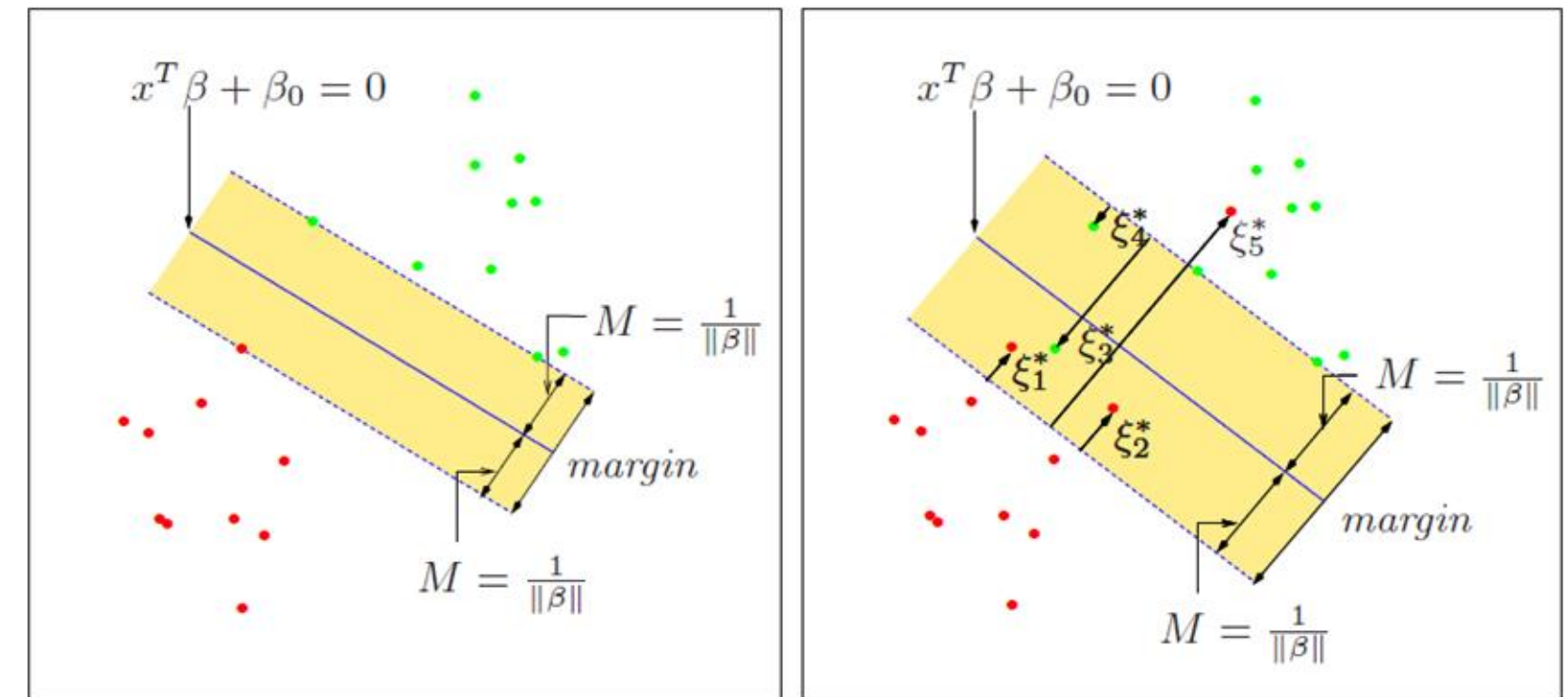
$$y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i),$$

$$\forall i, \xi_i \geq 0, \sum_{i=1}^N \xi_i \leq \text{constant}.$$



# The Support Vector Classifier

El valor  $\xi_i$  en la restricción  $y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i)$  es la cantidad proporcional por la que la predicción

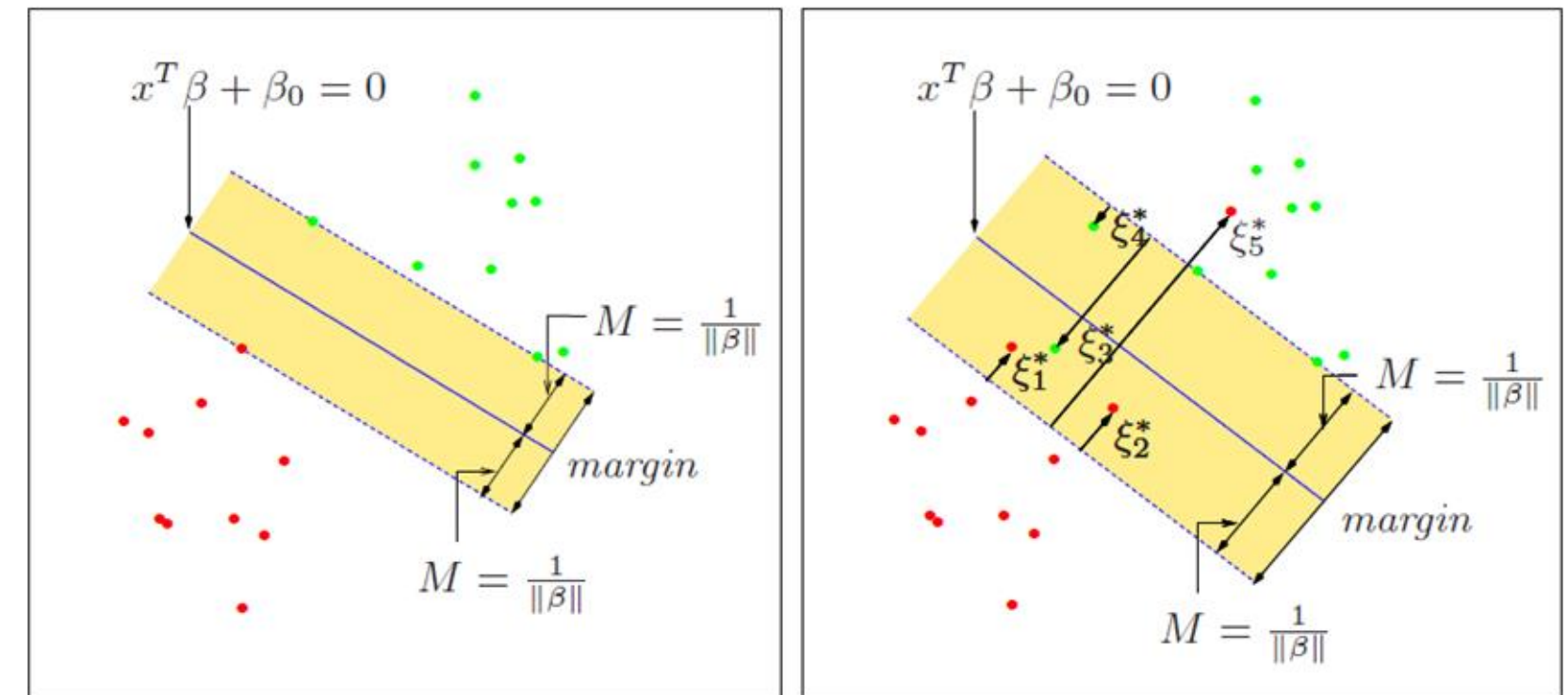


# The Support Vector Classifier

El valor  $\xi_i$  en la restricción  $y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i)$  es la cantidad proporcional por la que la predicción

$$f(x_i) = x_i^T \beta + \beta_0$$

está en el lado equivocado de su margen



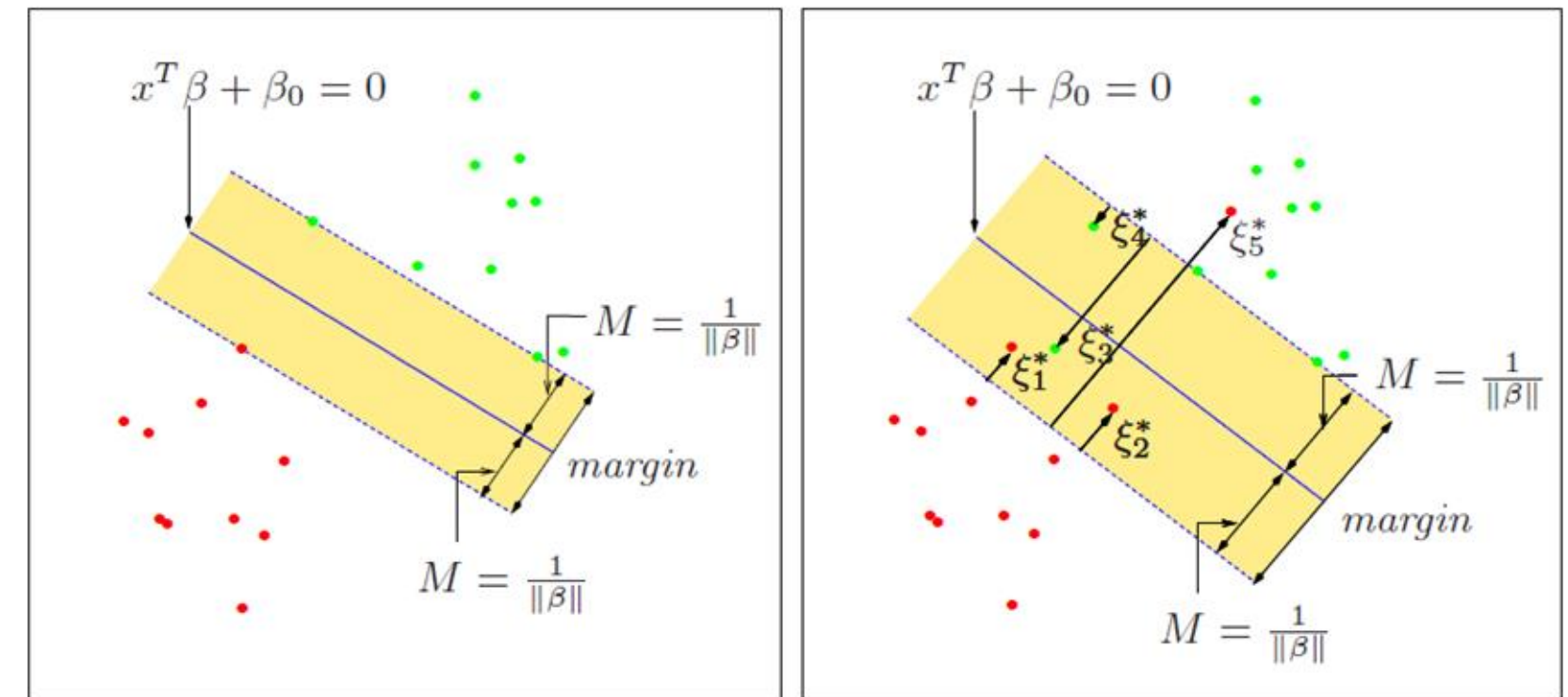
# The Support Vector Classifier

El valor  $\xi_i$  en la restricción  $y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i)$  es la cantidad proporcional por la que la predicción

$$f(x_i) = x_i^T \beta + \beta_0$$

está en el lado equivocado de su margen

Podemos eliminar la restricción normativa sobre  $\beta$ , definir  $M = 1/\|\beta\|$  y escribir la optimización convexa en la forma equivalente



# The Support Vector Classifier

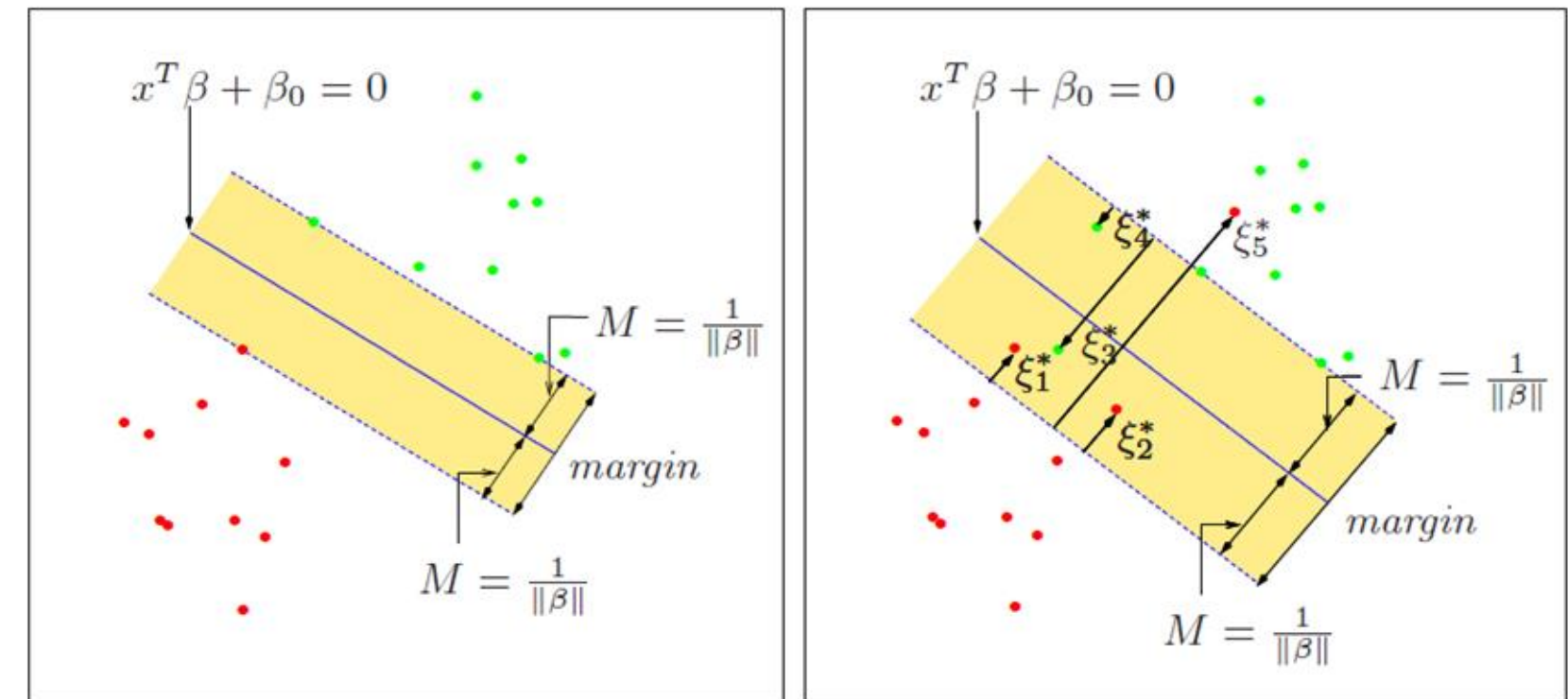
El valor  $\xi_i$  en la restricción  $y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i)$  es la cantidad proporcional por la que la predicción

$$f(x_i) = x_i^T \beta + \beta_0$$

está en el lado equivocado de su margen

Podemos eliminar la restricción normativa sobre  $\beta$ , definir  $M = 1/\|\beta\|$  y escribir la optimización convexa en la forma equivalente

$$\min \|\beta\| \quad \text{subject to} \quad \begin{cases} y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \quad \forall i, \\ \xi_i \geq 0, \quad \sum \xi_i \leq \text{constant}. \end{cases}$$



# The Support Vector Classifier

El valor  $\xi_i$  en la restricción  $y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i)$  es la cantidad proporcional por la que la predicción

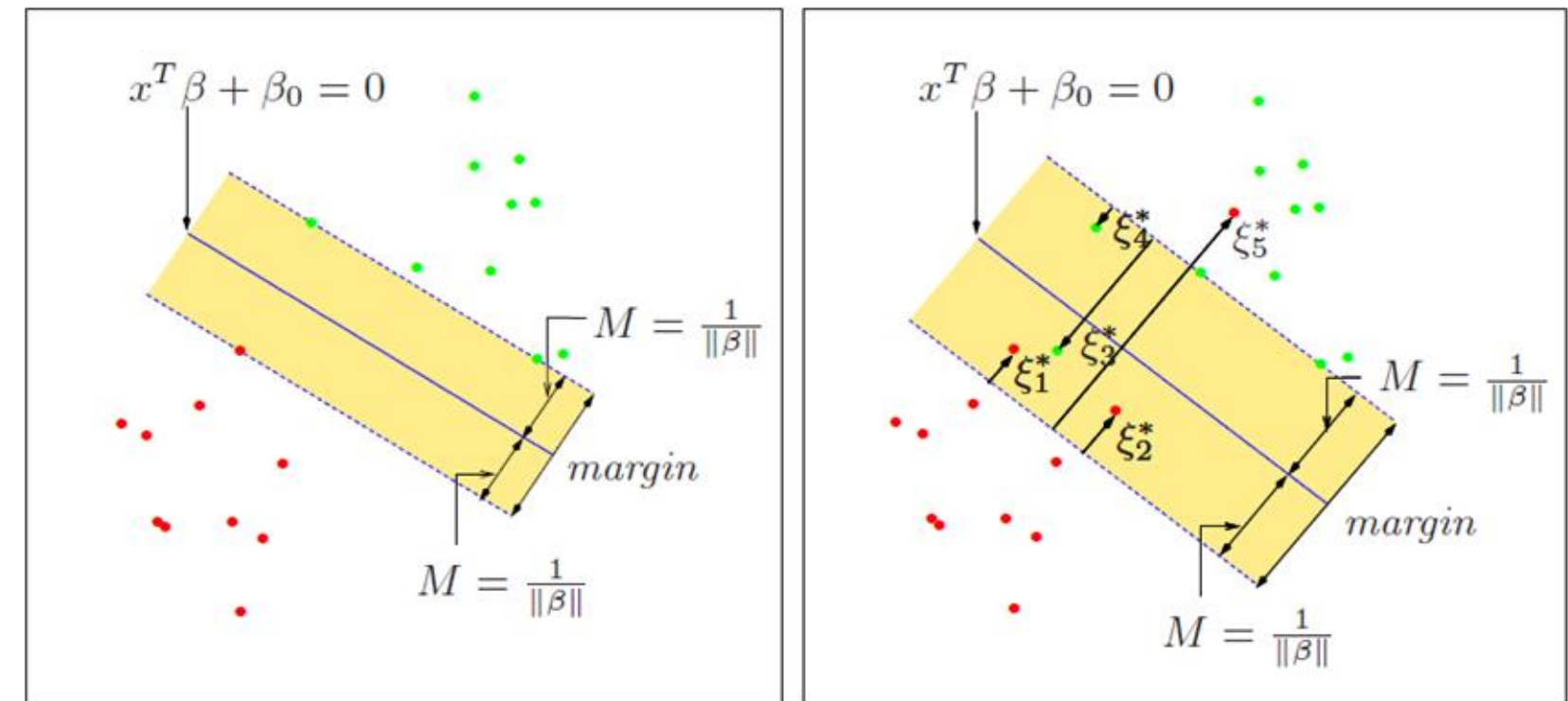
$$f(x_i) = x_i^T \beta + \beta_0$$

está en el lado equivocado de su margen

Podemos eliminar la restricción normativa sobre  $\beta$ , definir  $M = 1/\|\beta\|$  y escribir la optimización convexa en la forma equivalente

$$\min \|\beta\| \quad \text{subject to} \quad \begin{cases} y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \quad \forall i, \\ \xi_i \geq 0, \sum \xi_i \leq \text{constant}. \end{cases}$$

Esta es la forma habitual en que se define el clasificador de vectores de soporte para el caso no separable.



# Computing the Support Vector Classifier

Describimos una solución de programación cuadrática utilizando multiplicadores de Lagrange.

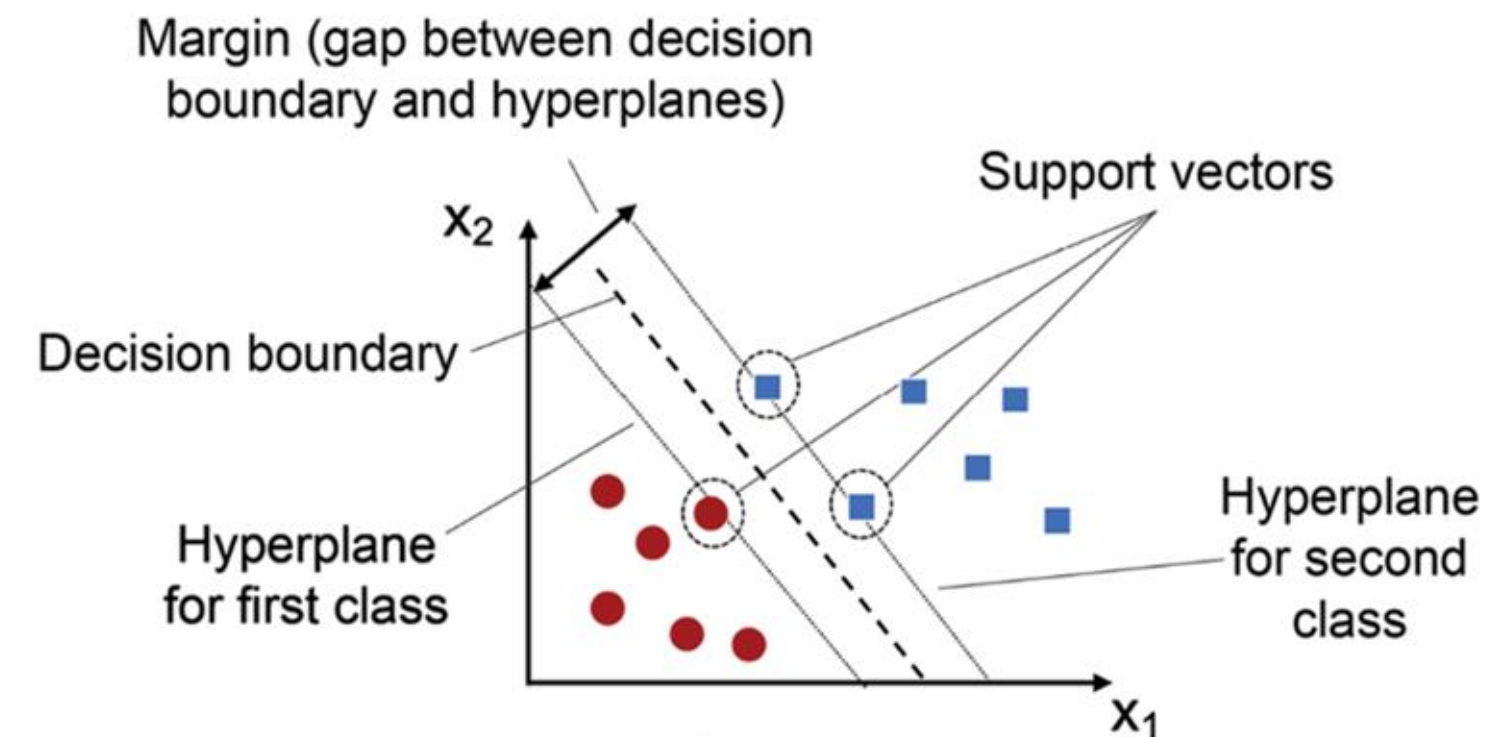
$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i$$

$$\text{subject to } \xi_i \geq 0, y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \quad \forall i,$$

el caso separable corresponde a  $C = \infty$ . La función de Lagrange (primaria) es

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

por el cual minimizamos w.r.t  $\beta$ ,  $\beta_0$  y  $\xi_i$ .



# Computing the Support Vector Classifier

Al establecer las derivadas respectivas en cero, obtenemos:

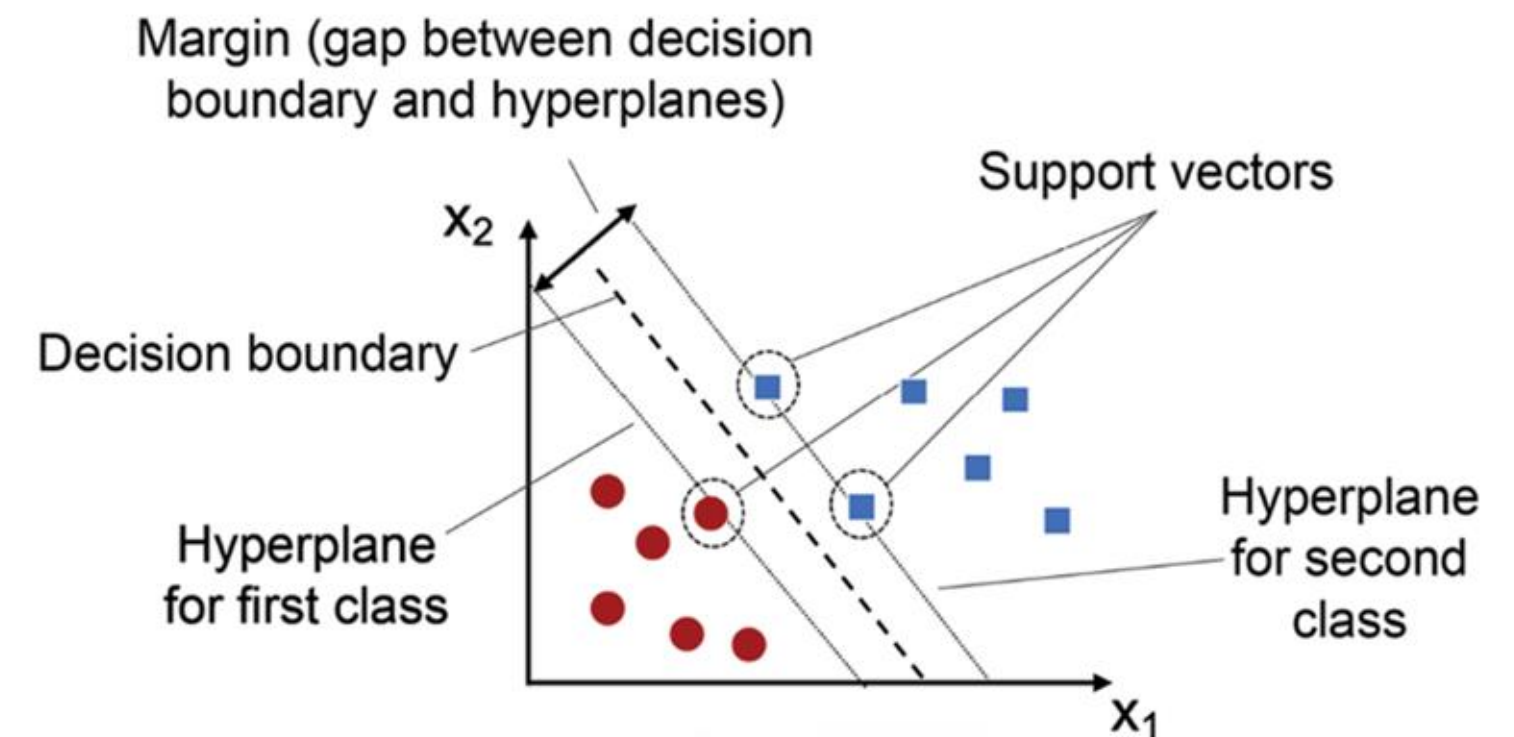
$$\beta = \sum_{i=1}^N \alpha_i y_i x_i,$$

$$0 = \sum_{i=1}^N \alpha_i y_i,$$

$$\alpha_i = C - \mu_i, \forall i,$$

obtenemos la función objetivo dual de Lagrange (Wolfe)

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{i'=1}^N \alpha_i \alpha_{i'} y_i y_{i'} x_i^T x_{i'},$$

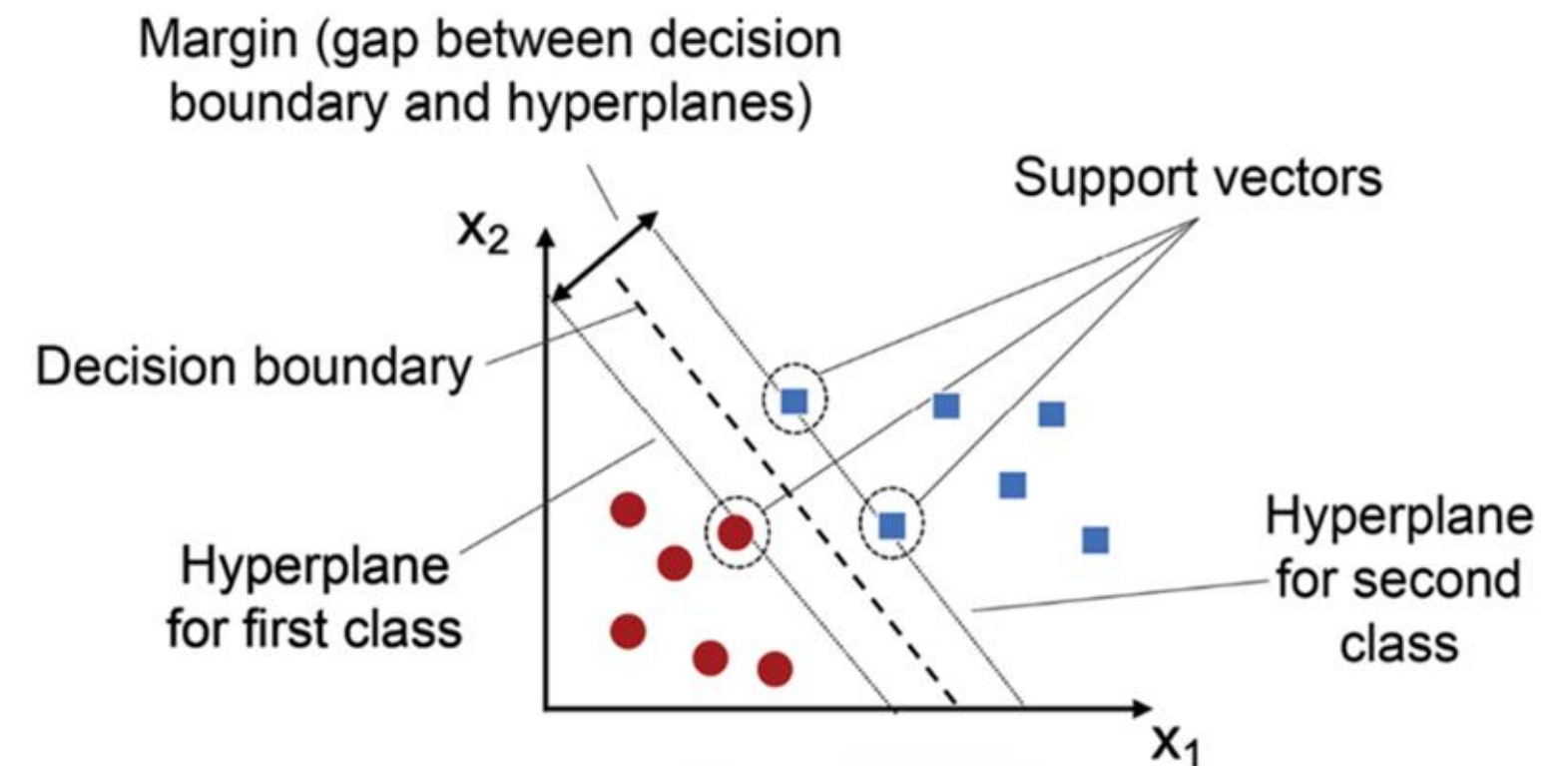


# Computing the Support Vector Classifier

Las condiciones de Karush-Kuhn-Tucker incluyen las restricciones

$$\begin{aligned}\alpha_i[y_i(x_i^T \beta + \beta_0) - (1 - \xi_i)] &= 0, \\ \mu_i \xi_i &= 0, \\ y_i(x_i^T \beta + \beta_0) - (1 - \xi_i) &\geq 0,\end{aligned}$$

Para  $i = 1, \dots, N$ .



# Computing the Support Vector Classifier

Las condiciones de Karush-Kuhn-Tucker incluyen las restricciones

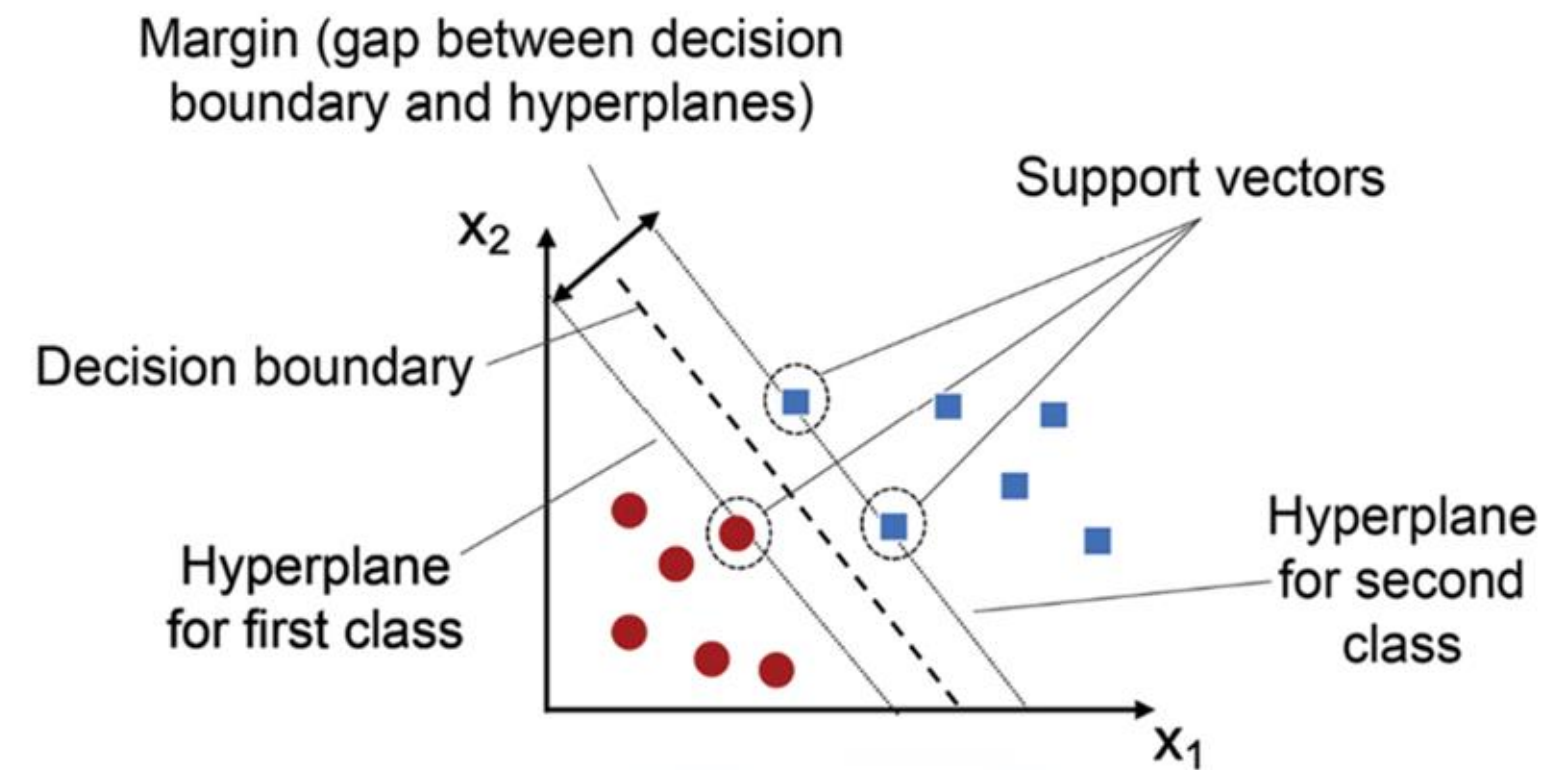
$$\begin{aligned}\alpha_i[y_i(x_i^T \beta + \beta_0) - (1 - \xi_i)] &= 0, \\ \mu_i \xi_i &= 0, \\ y_i(x_i^T \beta + \beta_0) - (1 - \xi_i) &\geq 0,\end{aligned}$$

Para  $i = 1, \dots, N$ .

vemos que la solución para  $\beta$  tiene la forma

$$\hat{\beta} = \sum_{i=1}^N \hat{\alpha}_i y_i x_i,$$

Estas observaciones se denominan vectores de soporte, ya que  $\hat{\beta}$  se representa únicamente en términos de ellos.

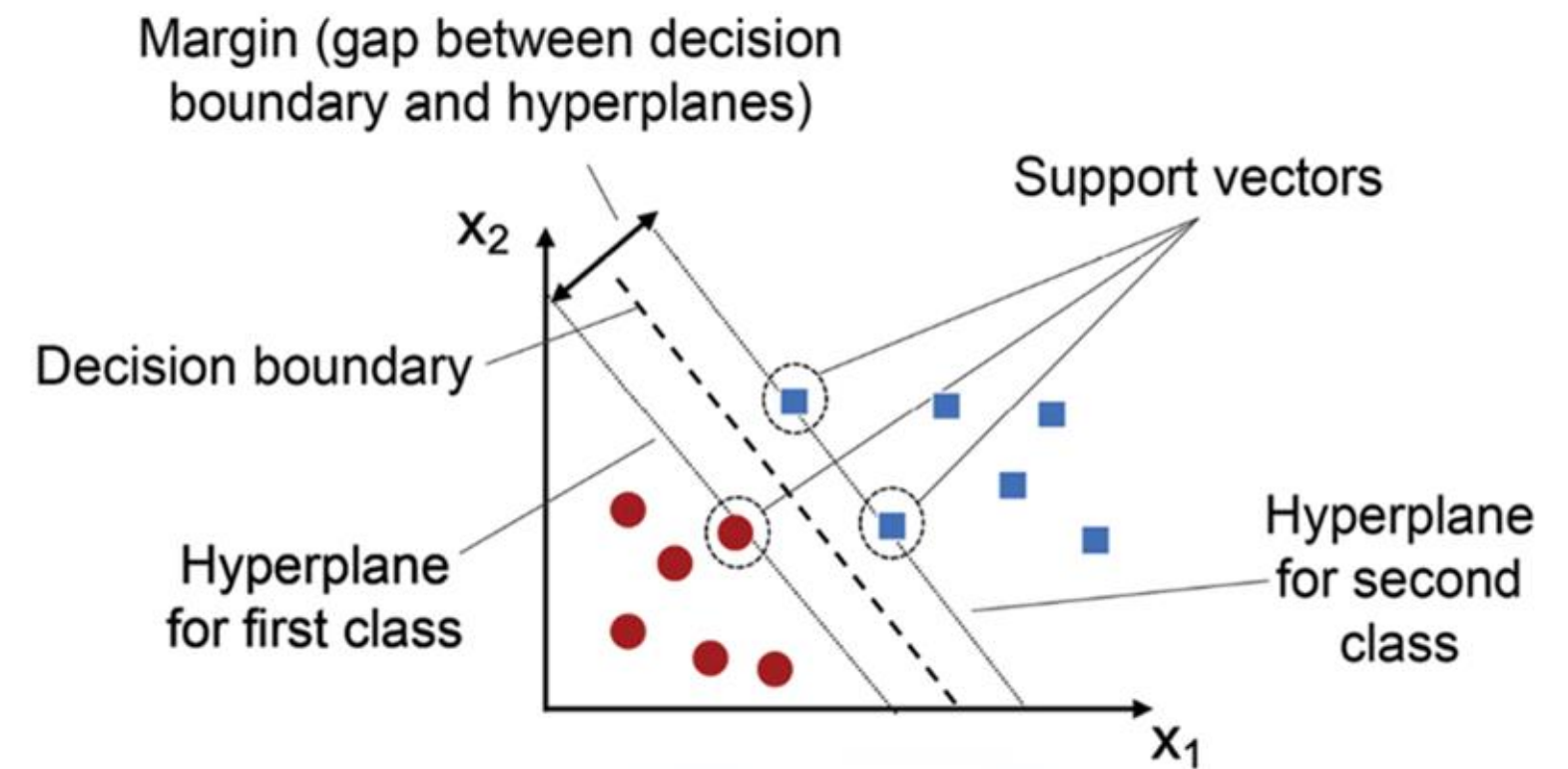


# Computing the Support Vector Classifier

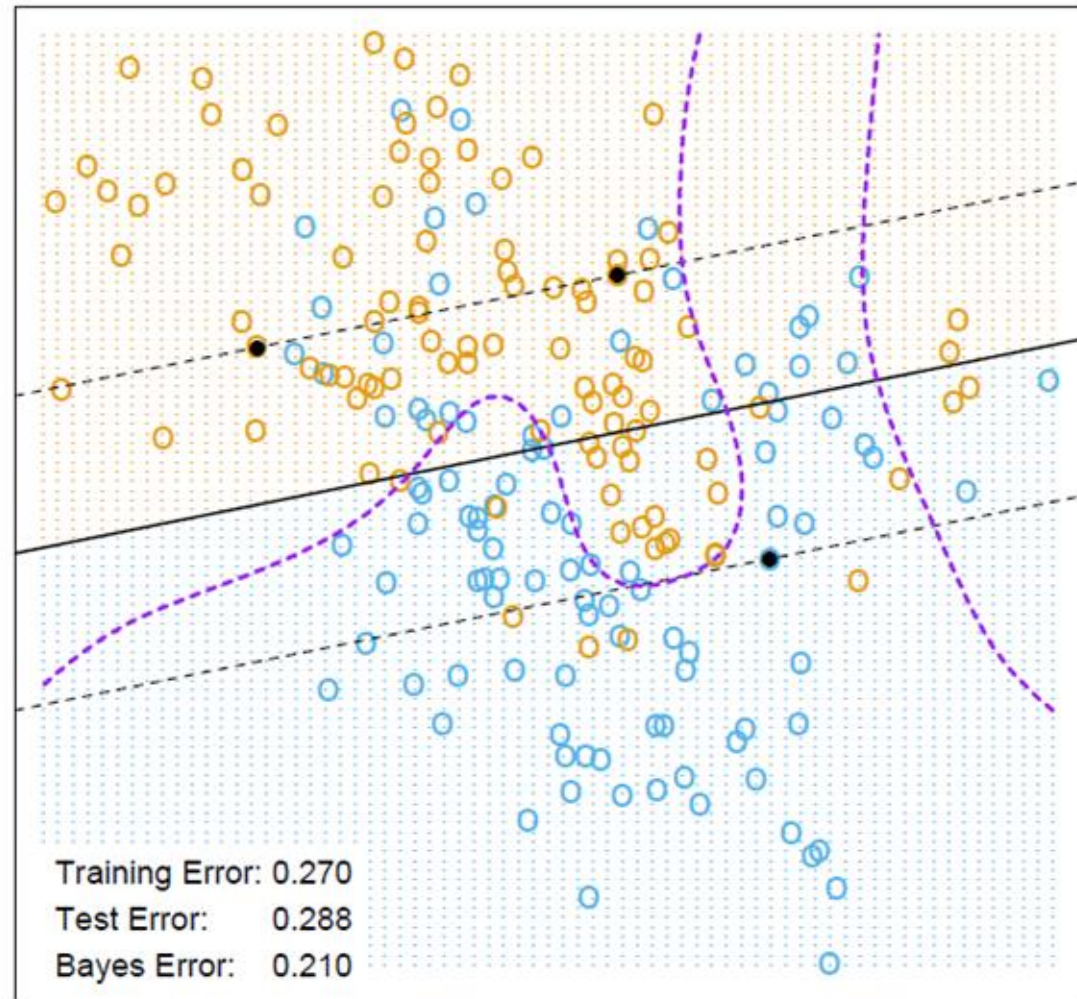
Dadas las soluciones  $\hat{\beta}_0$  y  $\hat{\beta}$ , la función de decisión se puede escribir como

$$\begin{aligned}\hat{G}(x) &= \text{sign}[\hat{f}(x)] \\ &= \text{sign}[x^T \hat{\beta} + \hat{\beta}_0].\end{aligned}$$

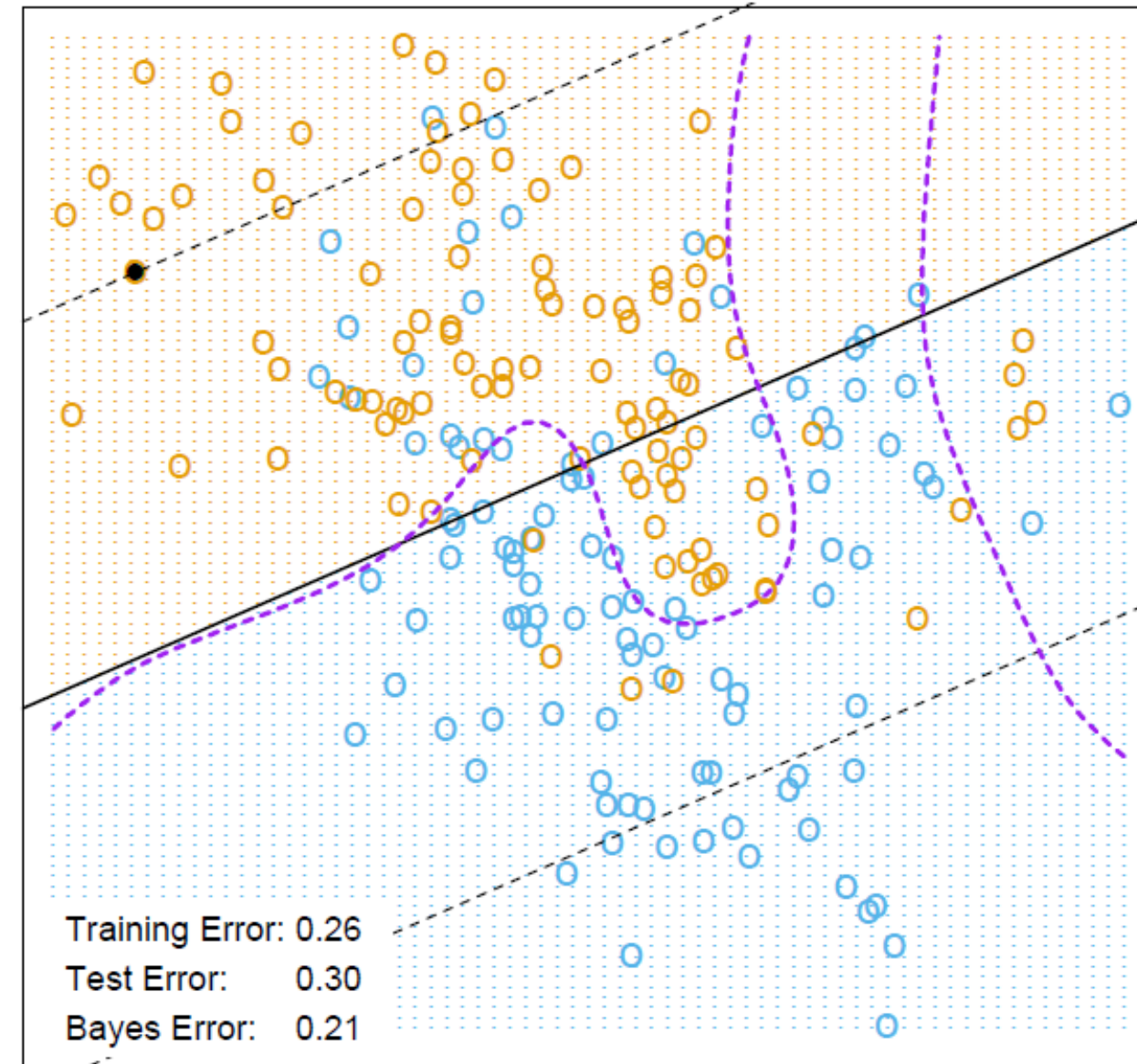
El parámetro de ajuste de este procedimiento es el parámetro de coste  $C$ .



# Mixture Example



$C = 10000$



$C = 0.01$

El límite lineal del vector de soporte para el ejemplo de datos mixtos con dos clases superpuestas, para dos valores diferentes de  $C$ . Las líneas discontinuas indican los márgenes, donde  $f(x) = \pm 1$ . En el panel superior, el 62 % de las observaciones son puntos de soporte, mientras que en el panel inferior lo son el 85 %. La curva púrpura discontinua del fondo es el límite de decisión de Bayes.



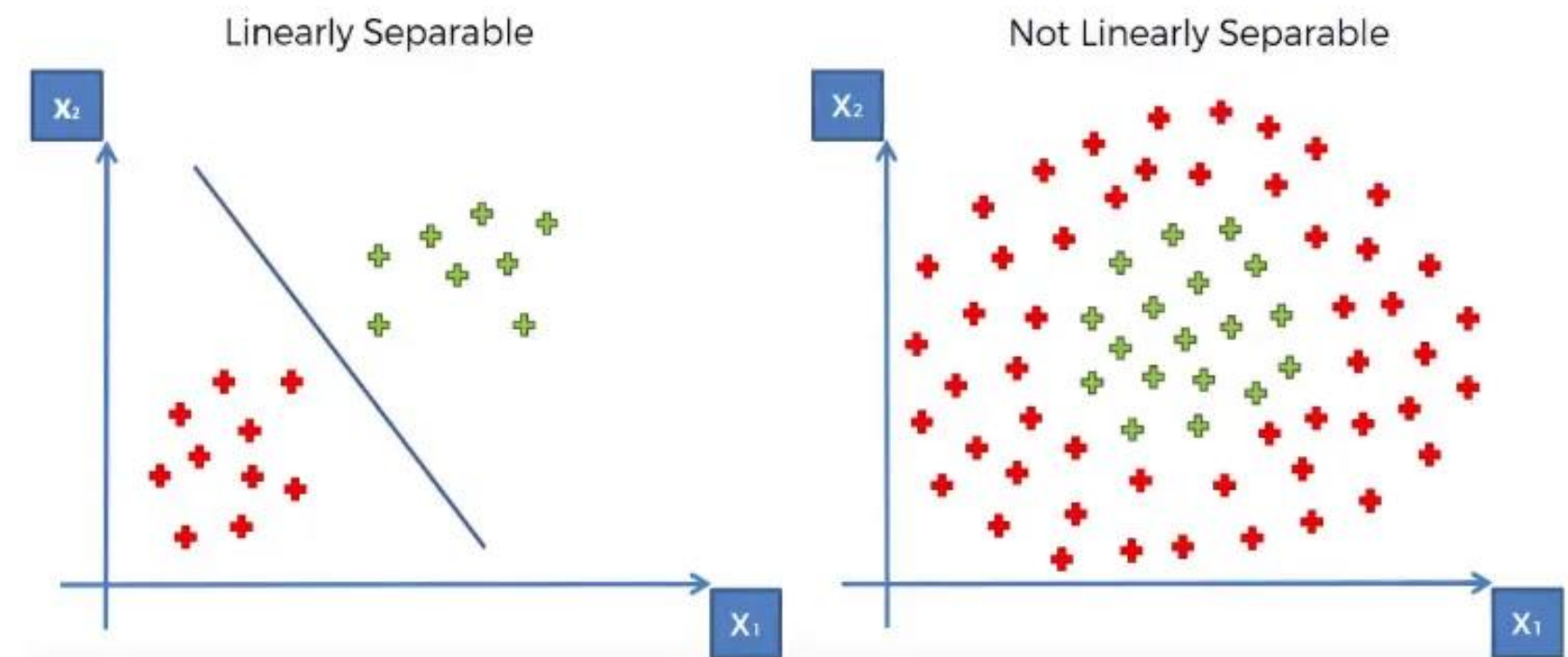
# Support Vector Machines and Kernels

Ajustamos el clasificador SV utilizando las características de entrada  $h(x_i) = (h_1(x_i), h_2(x_i), \dots, h_M(x_i))$ ,  $i = 1, \dots, N$ , y producimos la función (no lineal)  $\hat{f}(x) = h(x)^T \hat{\beta} + \hat{\beta}_0$ . El clasificador es  $\hat{G}(x) = \text{sign}(\hat{f}(x))$ .

El clasificador de máquinas de vectores de soporte es una extensión de esta idea, en la que se permite que la dimensión del espacio ampliado sea muy grande, infinita en algunos casos.

Podría parecer que los cálculos se volverían prohibitivos.

El clasificador SVM resuelve un problema de ajuste de funciones utilizando un criterio y una forma de regularización concretos, y forma parte de una clase de problemas mucho más amplia que incluye las splines de suavizado

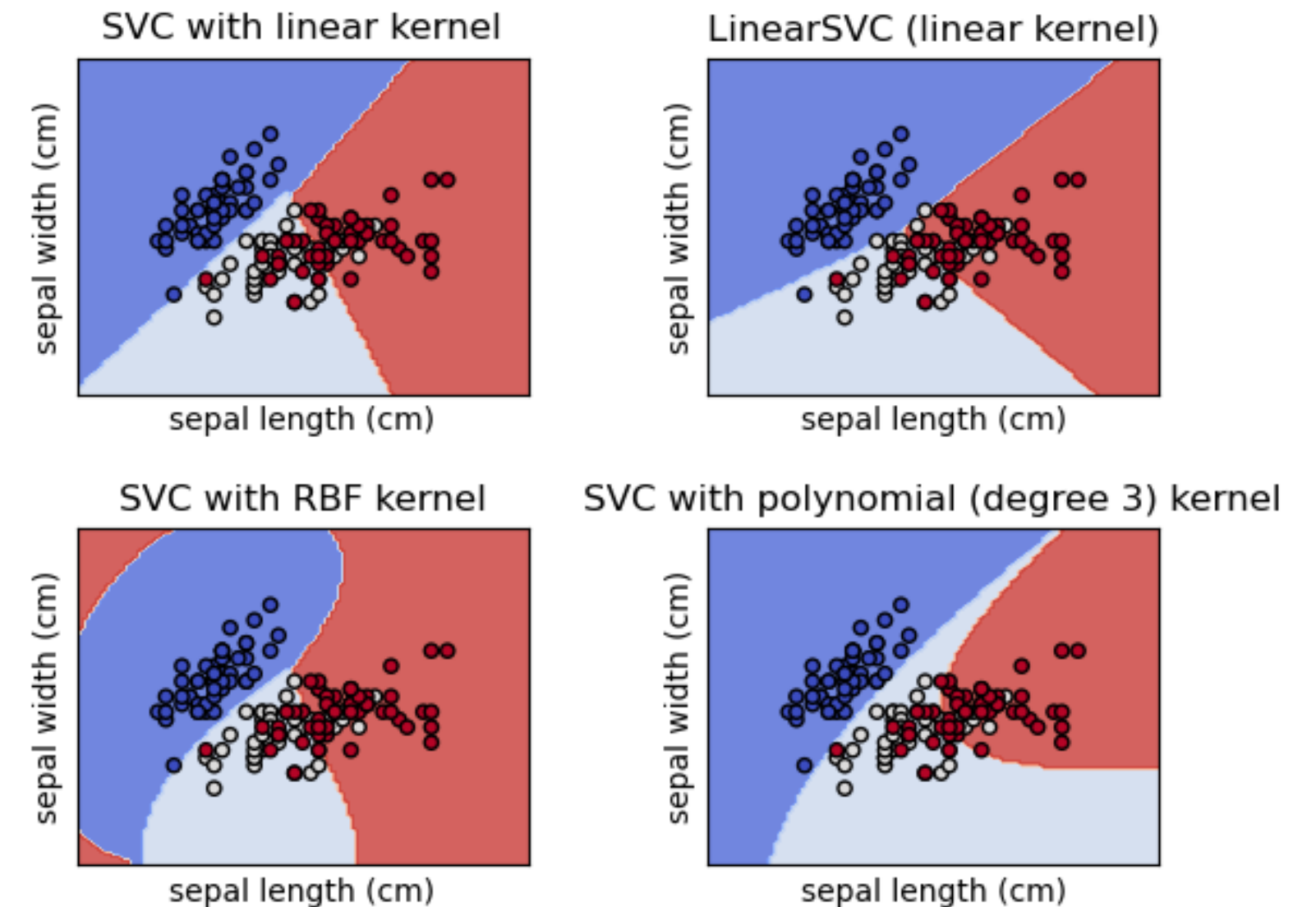


# Computing the SVM for Classification

La función dual de Lagrange tiene la forma

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{i'=1}^N \alpha_i \alpha_{i'} y_i y_{i'} \langle h(x_i), h(x_{i'}) \rangle.$$

vemos que la función solución  $f(x)$  se puede escribir



# Computing the SVM for Classification

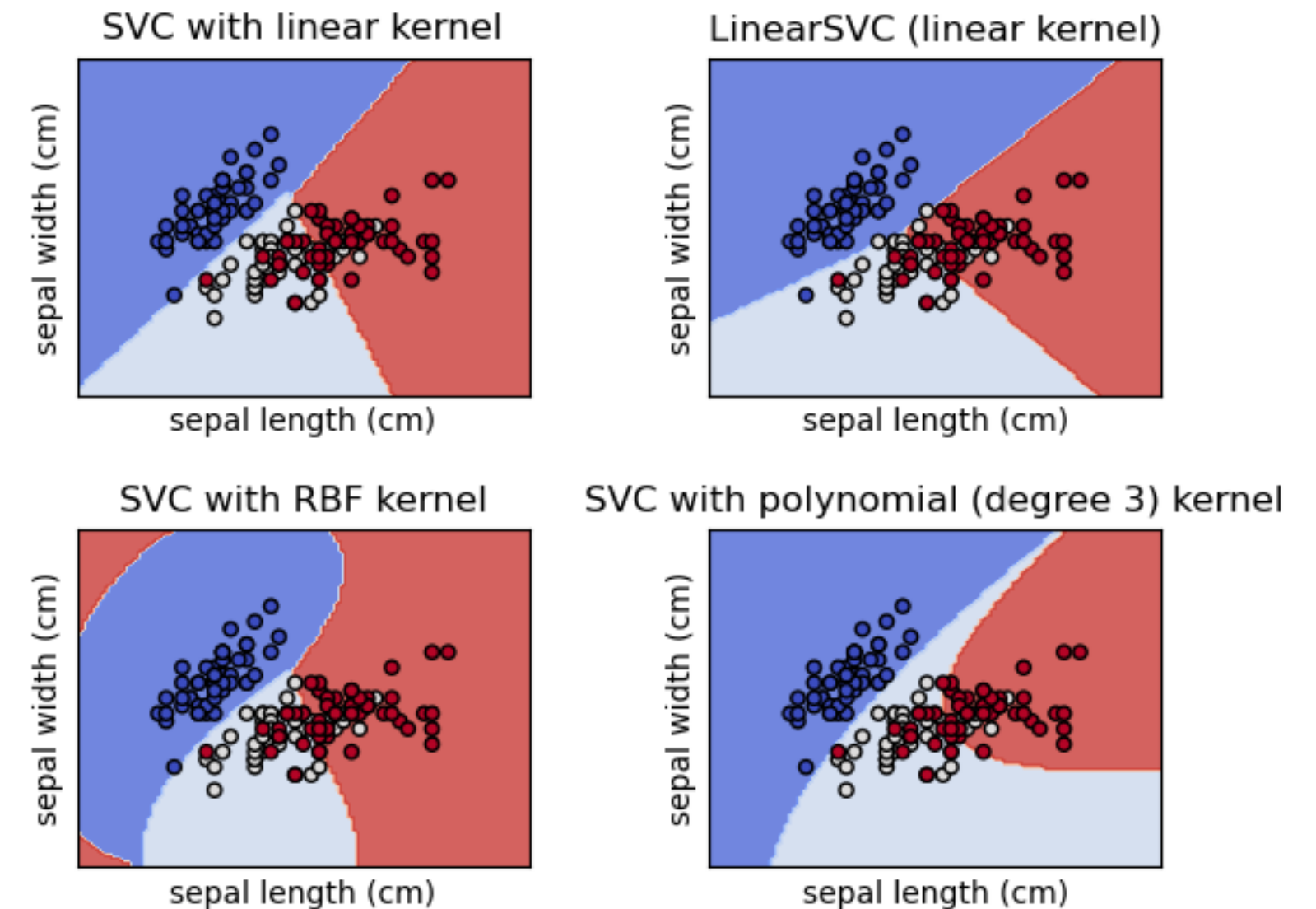
La función dual de Lagrange tiene la forma

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{i'=1}^N \alpha_i \alpha_{i'} y_i y_{i'} \langle h(x_i), h(x_{i'}) \rangle.$$

vemos que la función solución  $f(x)$  se puede escribir

$$\begin{aligned} f(x) &= h(x)^T \beta + \beta_0 \\ &= \sum_{i=1}^N \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0. \end{aligned}$$

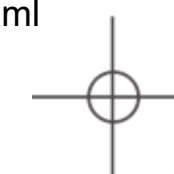
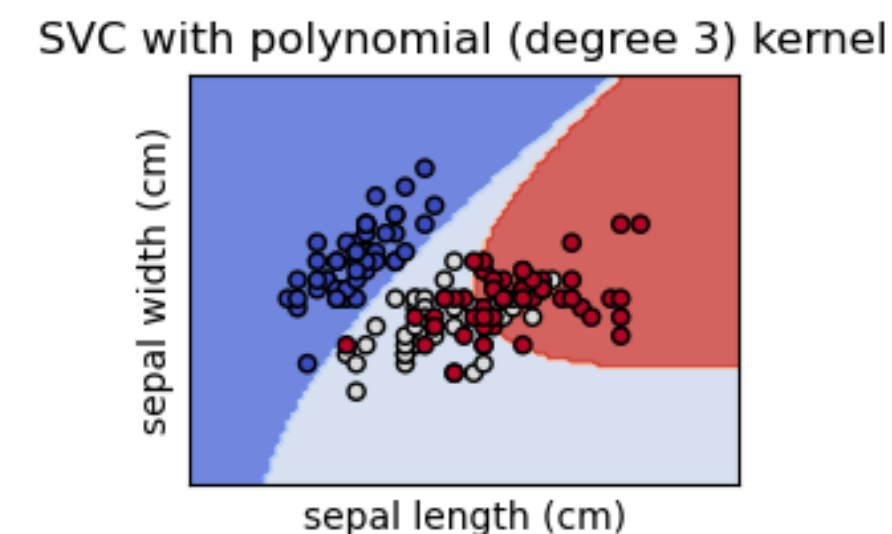
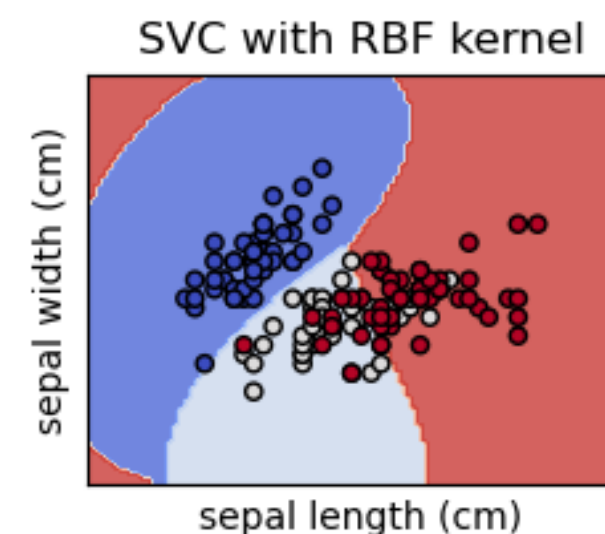
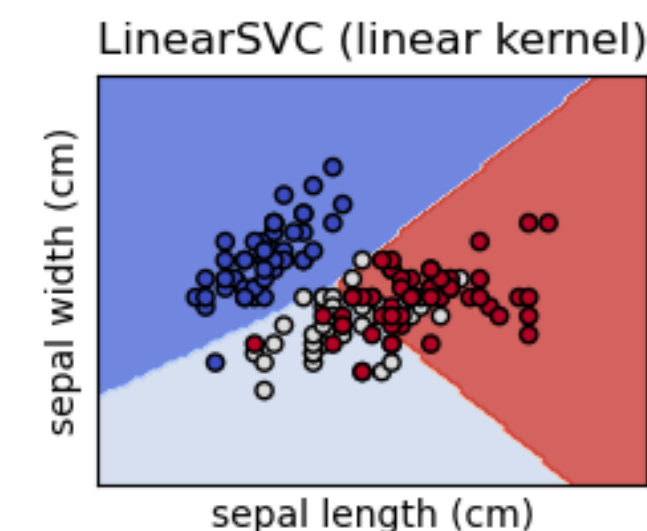
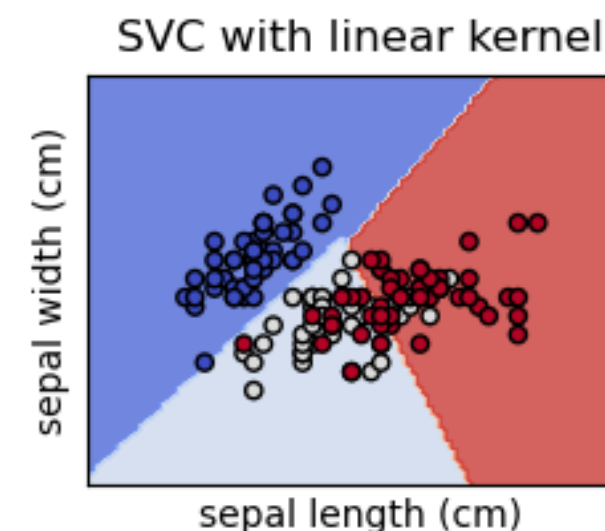
Dado  $\alpha_i$ ,  $\beta_0$  puede determinarse resolviendo  $y_i f(x_i) = 1$  para cualquier (o todos)  $x_i$  para los que  $0 < \alpha_i < C$ .



# Computing the SVM for Classification

Implican  $h(x)$  solo a través de productos internos. De hecho, no necesitamos especificar la transformación  $h(x)$  en absoluto, sino que solo requerimos el conocimiento de la función del núcleo.

$$K(x, x') = \langle h(x), h(x') \rangle$$



# Computing the SVM for Classification

Implican  $h(x)$  solo a través de productos internos. De hecho, no necesitamos especificar la transformación  $h(x)$  en absoluto, sino que solo requerimos el conocimiento de la función del núcleo.

$$K(x, x') = \langle h(x), h(x') \rangle$$

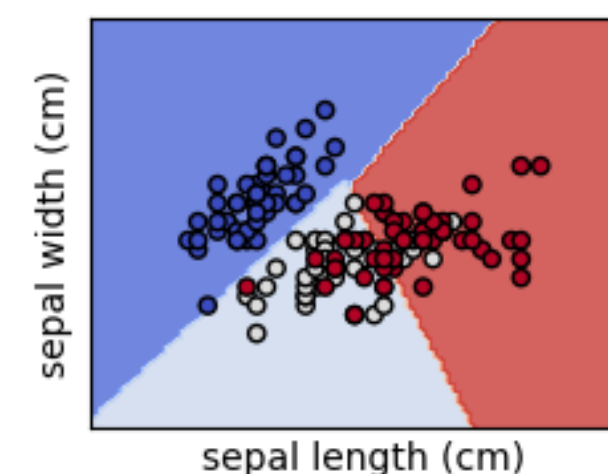
que calcula productos internos en el espacio transformado.  $K$  debe ser una función simétrica positiva (semi)definida. Tres opciones populares para  $K$  en la literatura sobre SVM son:

*d*th-Degree polynomial:  $K(x, x') = (1 + \langle x, x' \rangle)^d$ ,

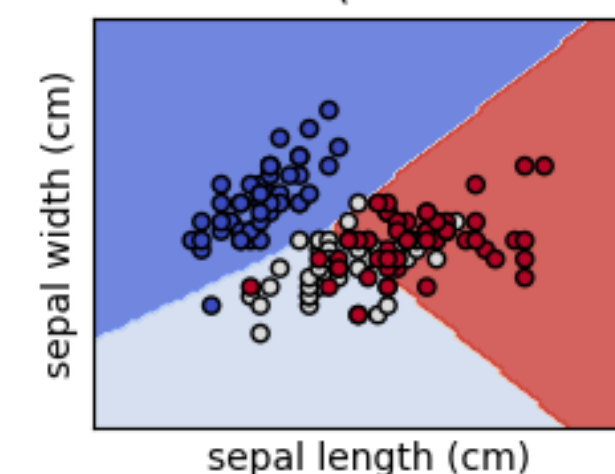
Radial basis:  $K(x, x') = \exp(-\gamma \|x - x'\|^2)$ ,

Neural network:  $K(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2)$ .

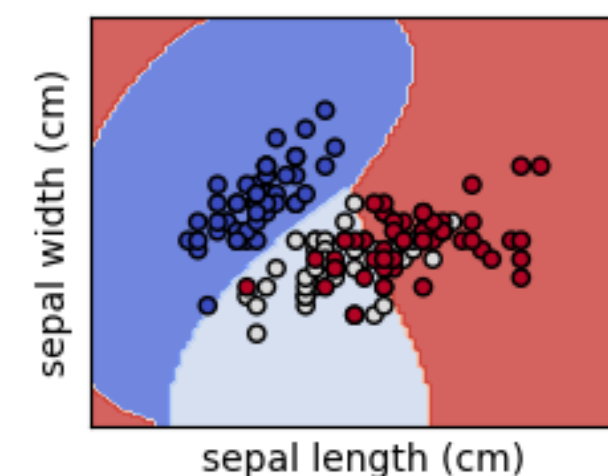
SVC with linear kernel



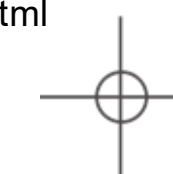
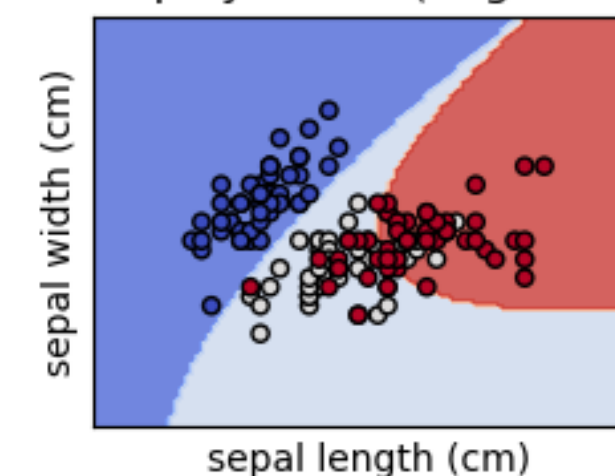
LinearSVC (linear kernel)



SVC with RBF kernel



SVC with polynomial (degree 3) kernel



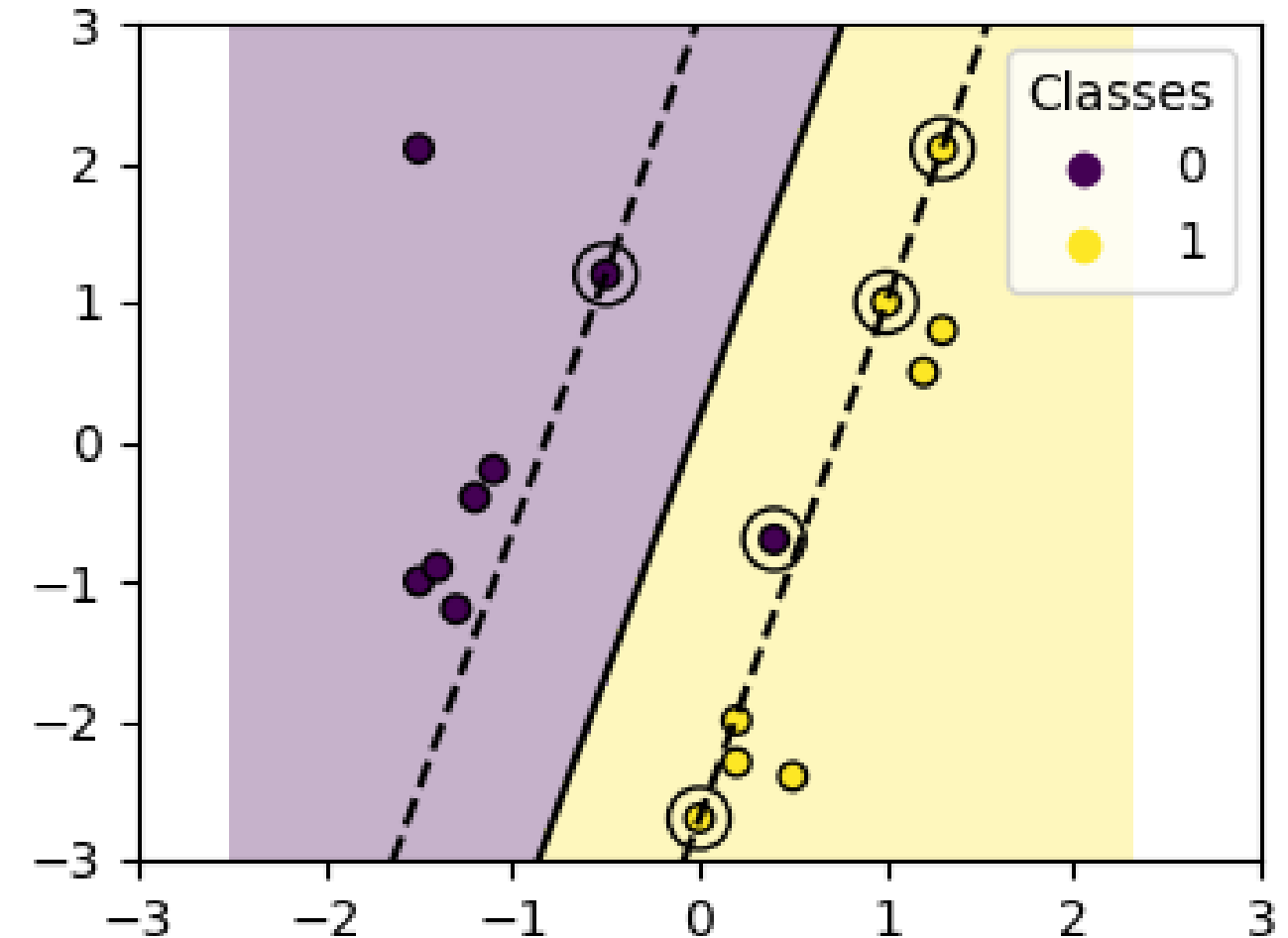
# Computing the SVM for Classification

Consideremos, por ejemplo, un espacio de características con dos entradas  $X_1$  y  $X_2$ , y un núcleo polinómico de grado 2.

Entonces:

$$\begin{aligned} K(X, X') &= (1 + \langle X, X' \rangle)^2 \\ &= (1 + X_1X'_1 + X_2X'_2)^2 \\ &= 1 + 2X_1X'_1 + 2X_2X'_2 + (X_1X'_1)^2 + (X_2X'_2)^2 + 2X_1X'_1X_2X'_2. \end{aligned}$$

Decision boundaries of linear kernel in SVC



[https://scikit-learn.org/stable/auto\\_examples/svm/plot\\_svm\\_kernels.html](https://scikit-learn.org/stable/auto_examples/svm/plot_svm_kernels.html)



# Computing the SVM for Classification

Consideremos, por ejemplo, un espacio de características con dos entradas  $X_1$  y  $X_2$ , y un núcleo polinómico de grado 2.

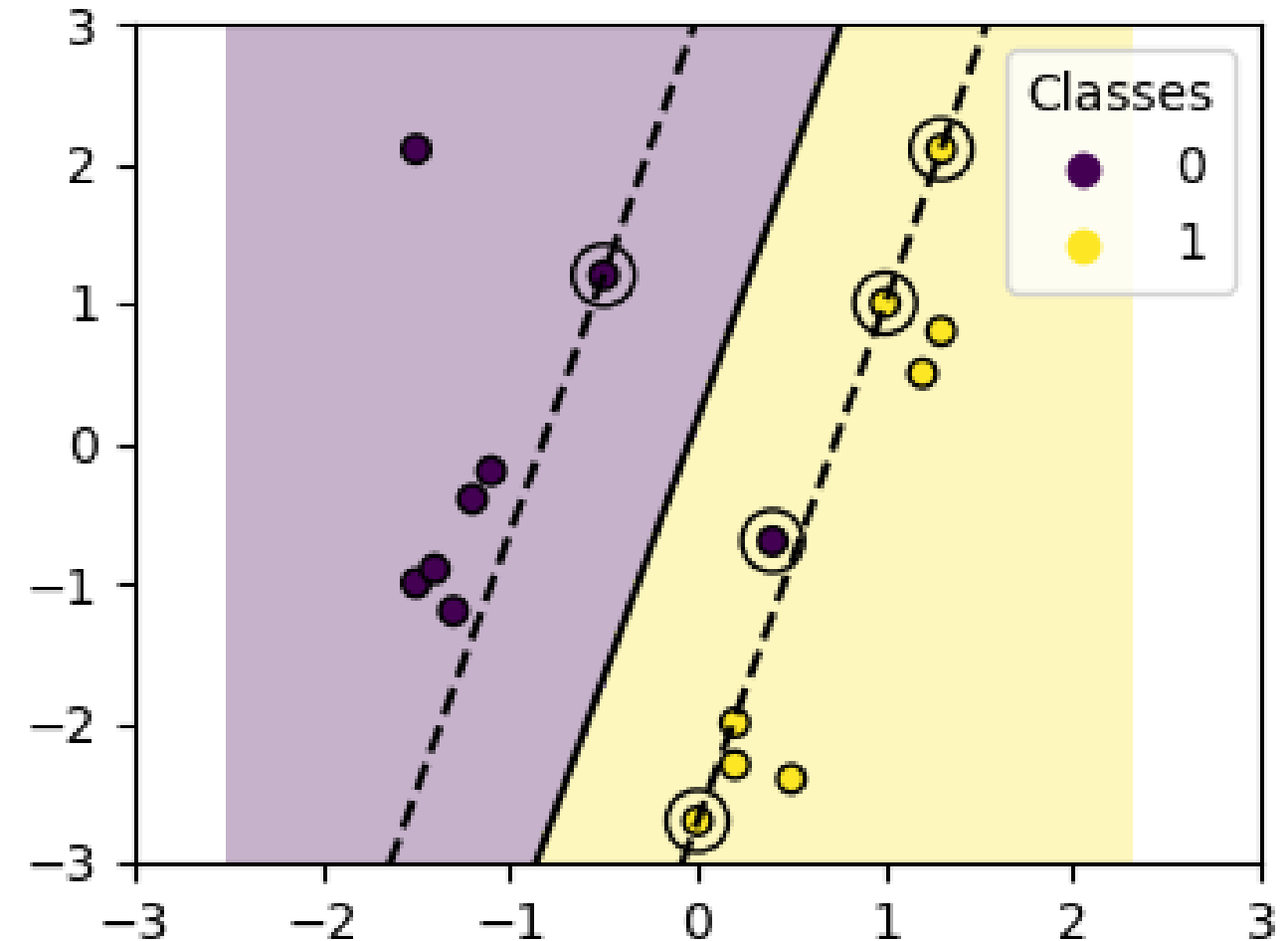
Entonces:

$$\begin{aligned} K(X, X') &= (1 + \langle X, X' \rangle)^2 \\ &= (1 + X_1X'_1 + X_2X'_2)^2 \\ &= 1 + 2X_1X'_1 + 2X_2X'_2 + (X_1X'_1)^2 + (X_2X'_2)^2 + 2X_1X'_1X_2X'_2. \end{aligned}$$

Vemos que la solución puede ser escrita:

$$\hat{f}(x) = \sum_{i=1}^N \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0.$$

Decision boundaries of linear kernel in SVC

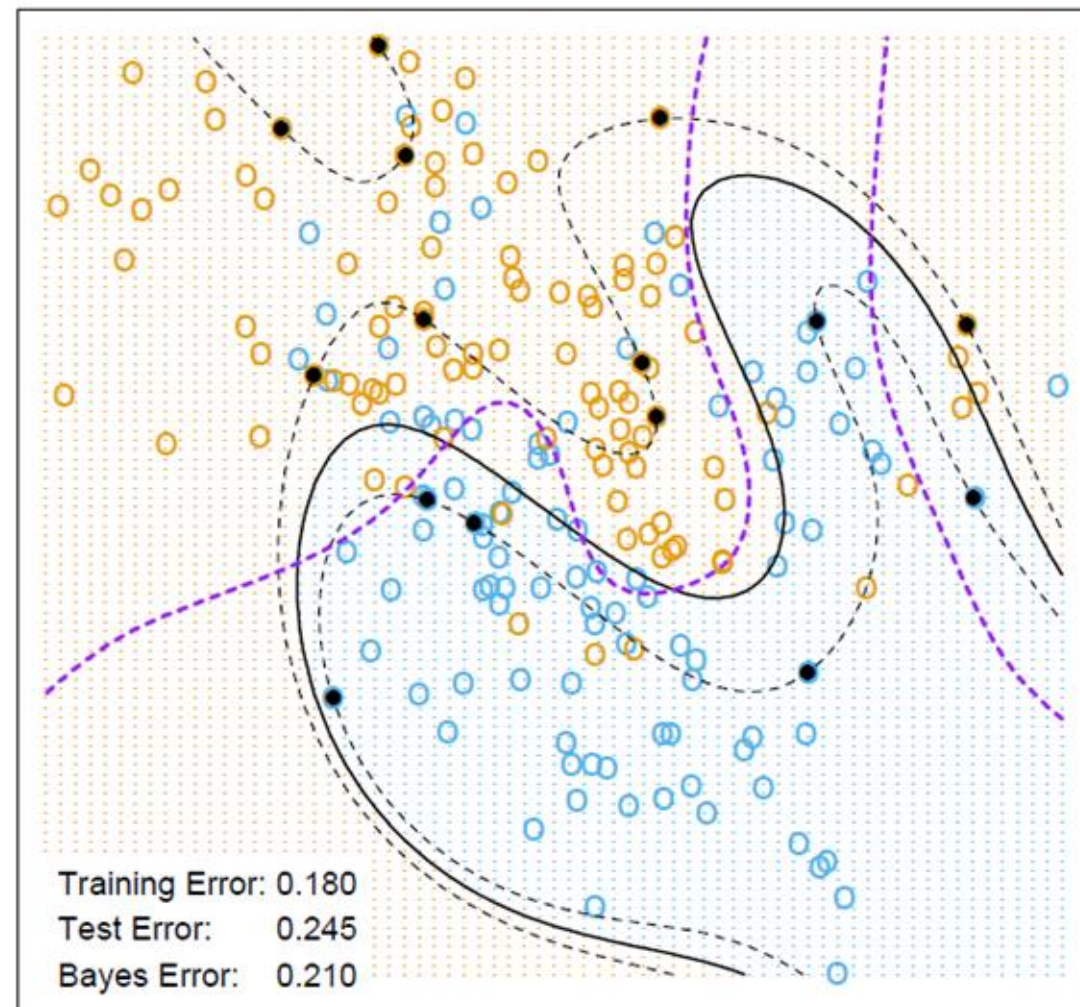


[https://scikit-learn.org/stable/auto\\_examples/svm/plot\\_svm\\_kernels.html](https://scikit-learn.org/stable/auto_examples/svm/plot_svm_kernels.html)

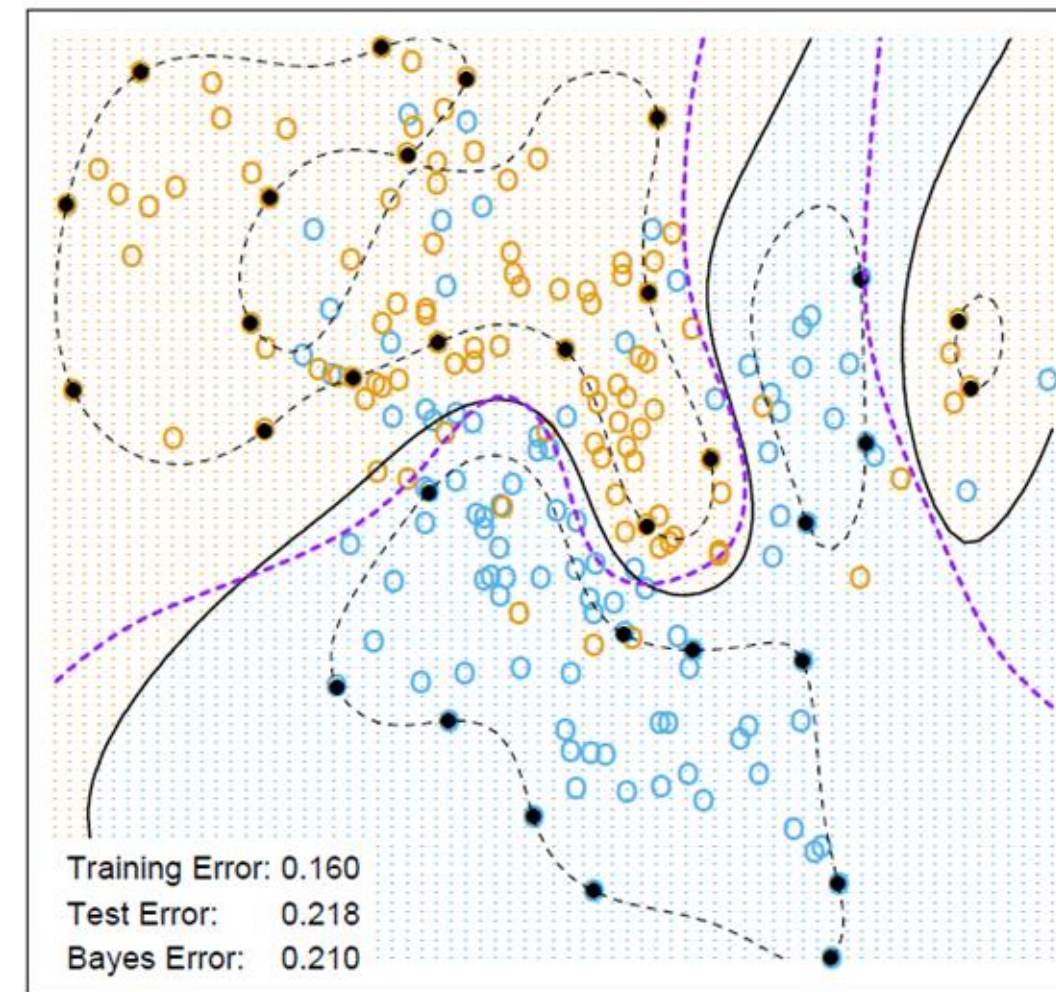


# Computing the SVM for Classification

SVM - Degree-4 Polynomial in Feature Space



SVM - Radial Kernel in Feature Space

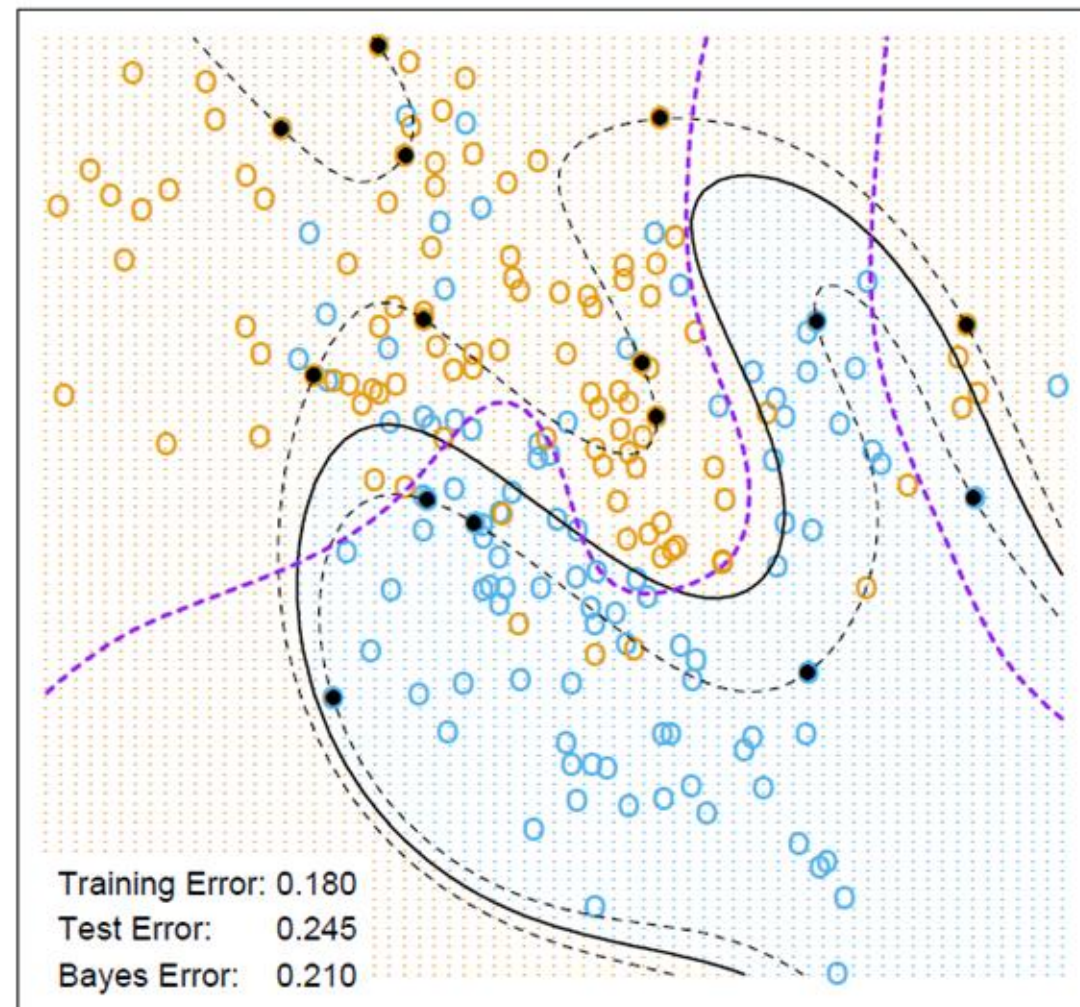


Dos SVM no lineales para los datos mixtos. El gráfico izquierdo utiliza un núcleo polinómico de cuarto grado, mientras que el inferior utiliza un núcleo de base radial (con  $\gamma = 1$ ).

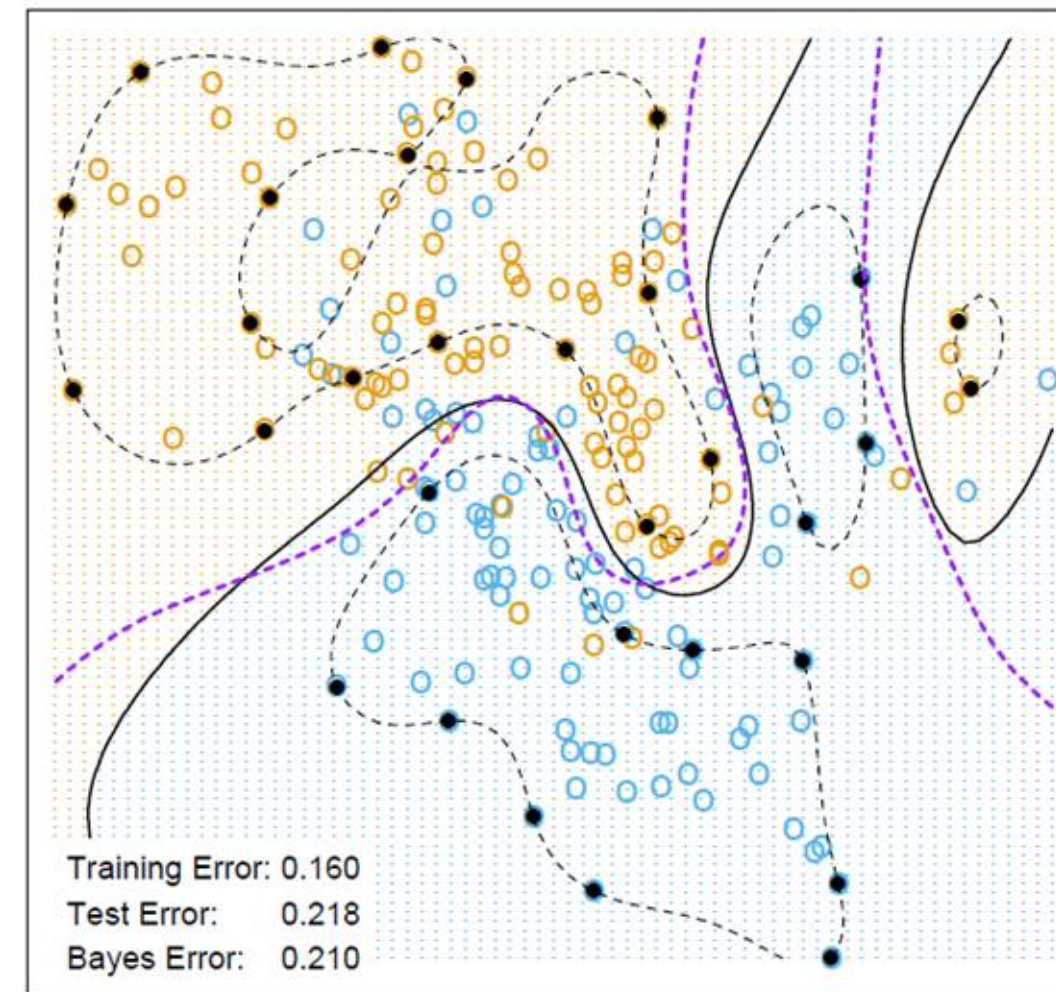


# Computing the SVM for Classification

SVM - Degree-4 Polynomial in Feature Space



SVM - Radial Kernel in Feature Space

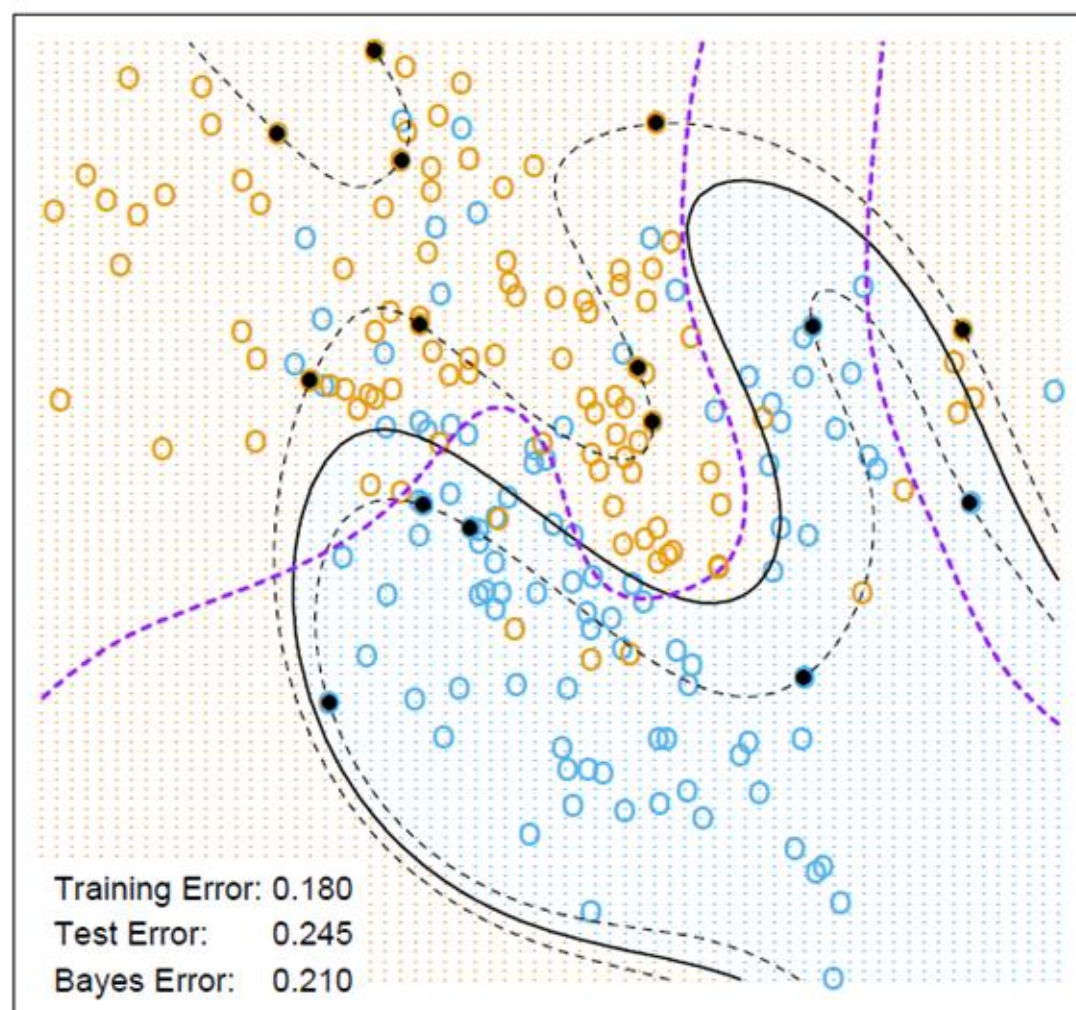


Dos SVM no lineales para los datos mixtos. El gráfico izquierdo utiliza un núcleo polinómico de cuarto grado, mientras que el inferior utiliza un núcleo de base radial (con  $\gamma = 1$ ). En cada caso,  $C$  se ajustó para lograr aproximadamente el mejor rendimiento en cuanto a errores de prueba, y  $C = 1$  funcionó bien en ambos casos.

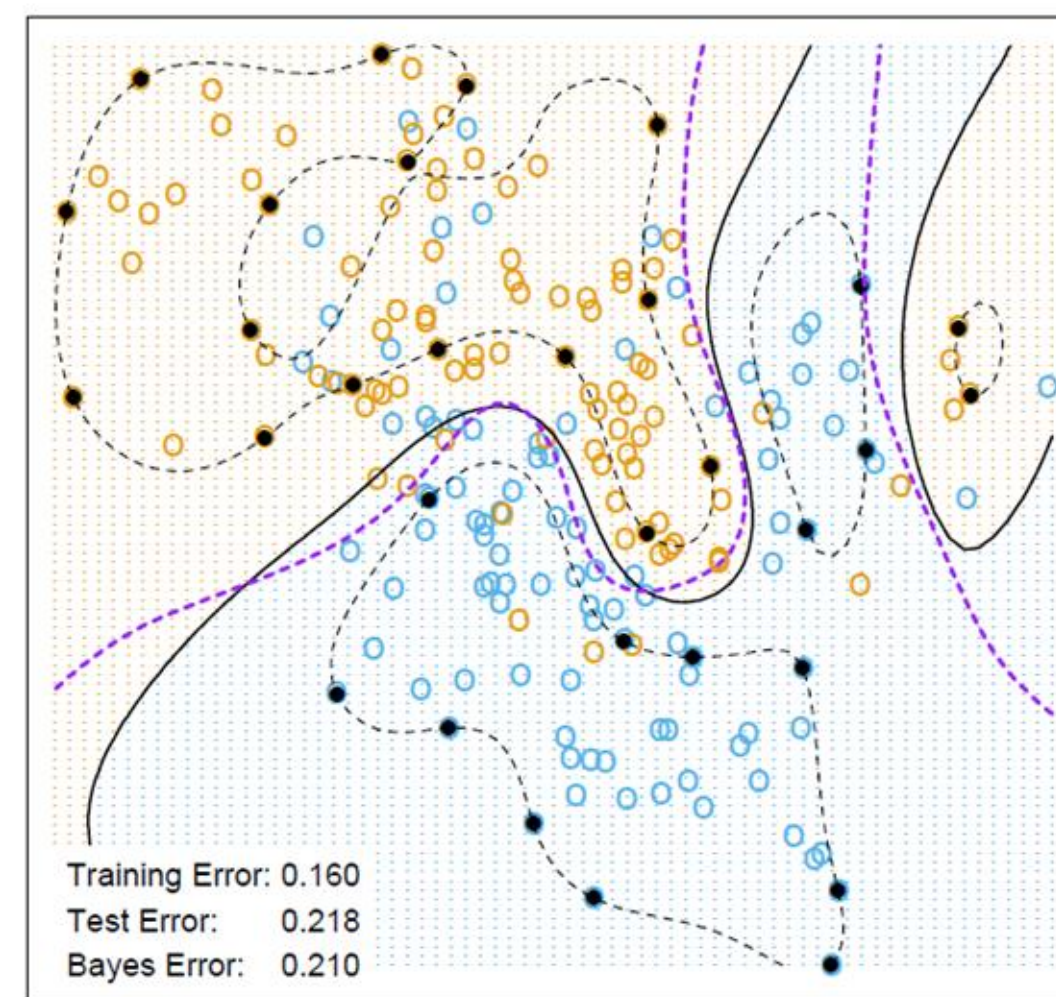


# Computing the SVM for Classification

SVM - Degree-4 Polynomial in Feature Space



SVM - Radial Kernel in Feature Space



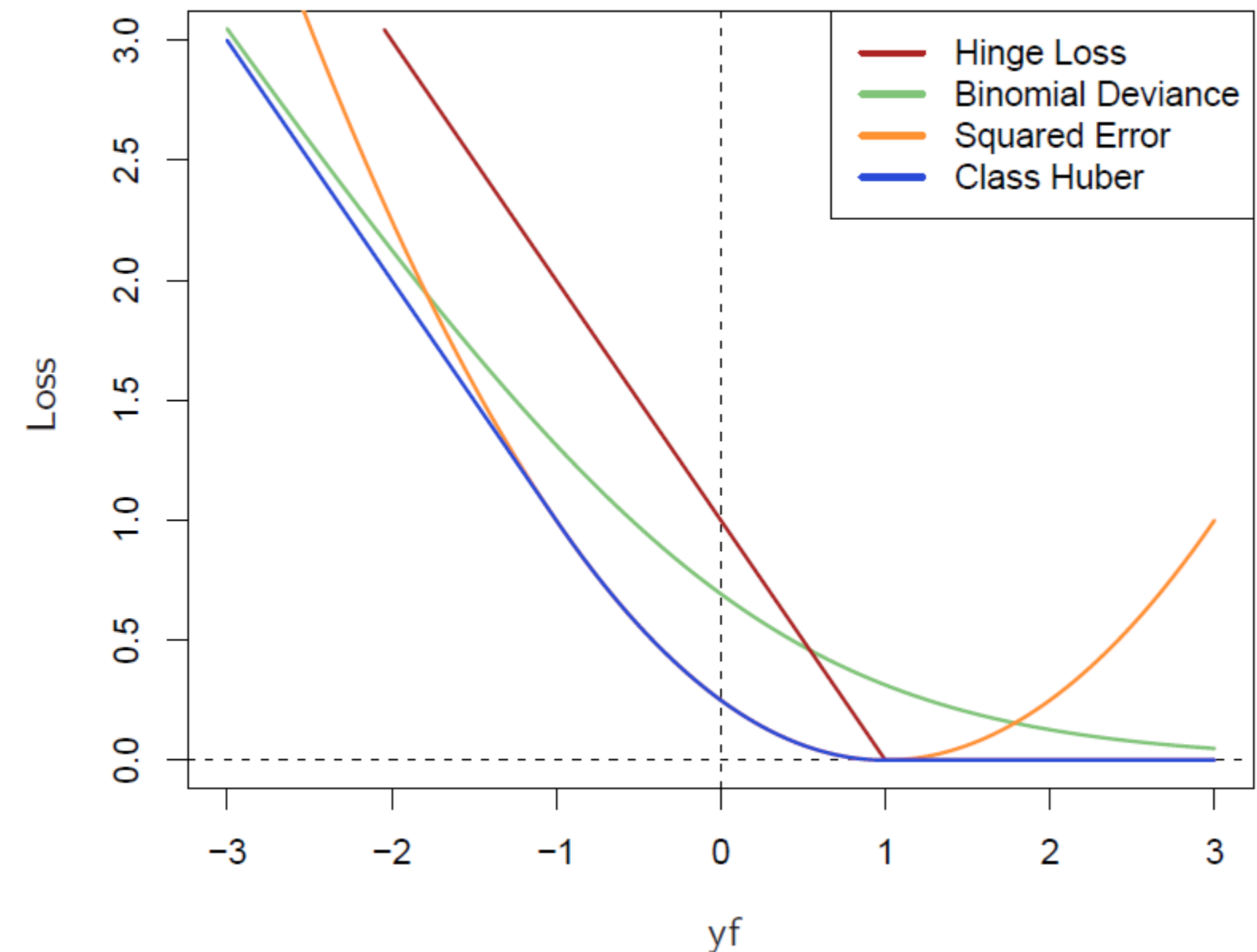
Dos SVM no lineales para los datos mixtos. El gráfico izquierdo utiliza un núcleo polinómico de cuarto grado, mientras que el inferior utiliza un núcleo de base radial (con  $\gamma = 1$ ). En cada caso,  $C$  se ajustó para lograr aproximadamente el mejor rendimiento en cuanto a errores de prueba, y  $C = 1$  funcionó bien en ambos casos. El núcleo de base radial es el que mejor rendimiento ofrece (cercano al óptimo bayesiano), como cabría esperar dado que los datos provienen de mezclas de gaussianas. La curva púrpura discontinua del fondo es el límite de decisión bayesiano.



# Computing the SVM for Classification

La función de pérdida del vector de soporte (pérdida de bisagra), en comparación con la pérdida de log-verosimilitud negativa (desviación binomial) para la regresión logística, la pérdida de error cuadrático y una versión «Huberizada» de la pérdida de bisagra cuadrática.

Todas se muestran como una función de  $yf$  en lugar de  $f$ , debido a la simetría entre los casos  $y = +1$  y  $y = -1$ . La desviación y Huber tienen las mismas asíntotas que la pérdida SVM, pero se redondean en el interior. Todas se escalan para tener la pendiente límite de la cola izquierda de  $-1$ .



# The SVM as a Penalization Method

Con  $f(x) = h(x)^T \beta + \beta_0$ , considerar el problema de optimización:

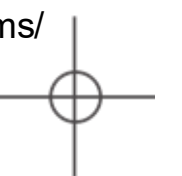
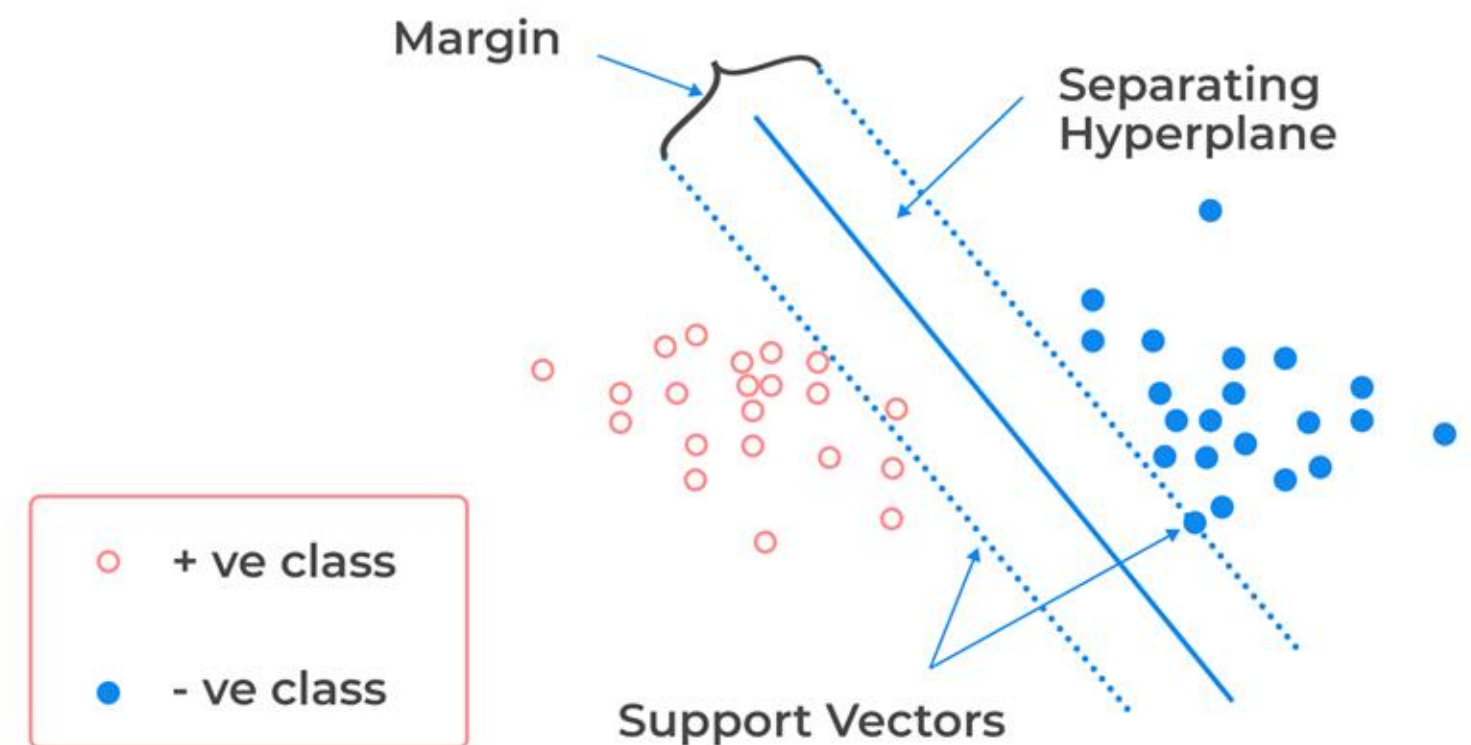
$$\min_{\beta_0, \beta} \sum_{i=1}^N [1 - y_i f(x_i)]_+ + \frac{\lambda}{2} \|\beta\|^2$$

donde el subíndice «+» indica la parte positiva. Esto tiene la forma pérdida + penalización, que es un paradigma familiar en la estimación de funciones.

Examination of the “hinge” loss function  $L(y, f) = [1 - yf]_+$  shows that it is reasonable for two-class classification, when compared to other more traditional loss functions.



## Support Vector Machine



# The SVM as a Penalization Method

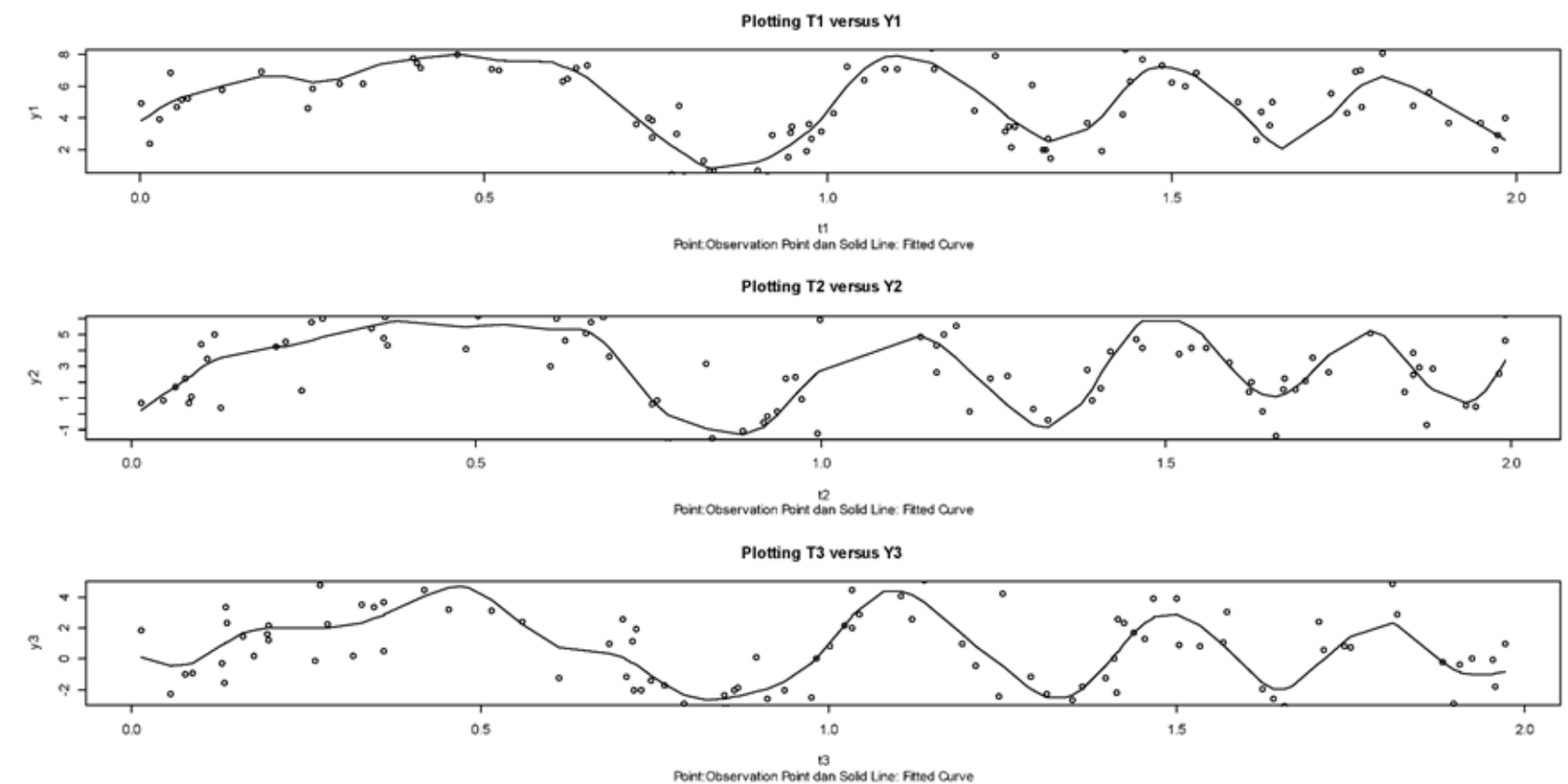
Loss Function	$L[y, f(x)]$	Minimizing Function
Binomial Deviance	$\log[1 + e^{-yf(x)}]$	$f(x) = \log \frac{\Pr(Y = +1 x)}{\Pr(Y = -1 x)}$
SVM Hinge Loss	$[1 - yf(x)]_+$	$f(x) = \text{sign}[\Pr(Y = +1 x) - \frac{1}{2}]$
Squared Error	$[y - f(x)]^2 = [1 - yf(x)]^2$	$f(x) = 2\Pr(Y = +1 x) - 1$
“Huberised” Square Hinge Loss	$-4yf(x), \quad yf(x) < -1$ $[1 - yf(x)]_+^2 \quad \text{otherwise}$	$f(x) = 2\Pr(Y = +1 x) - 1$

La regresión logística utiliza la log-verosimilitud binomial o la desviación. El análisis discriminante lineal utiliza la pérdida por error cuadrático. La pérdida de bisagra SVM estima el modo de las probabilidades de clase posteriores, mientras que las demás estiman una transformación lineal de estas probabilidades.



# Function Estimation and Reproducing Kernels

Supongamos que la base  $h$  surge de la expansión propia (posiblemente finita) de un núcleo definido positivo  $K$ ,



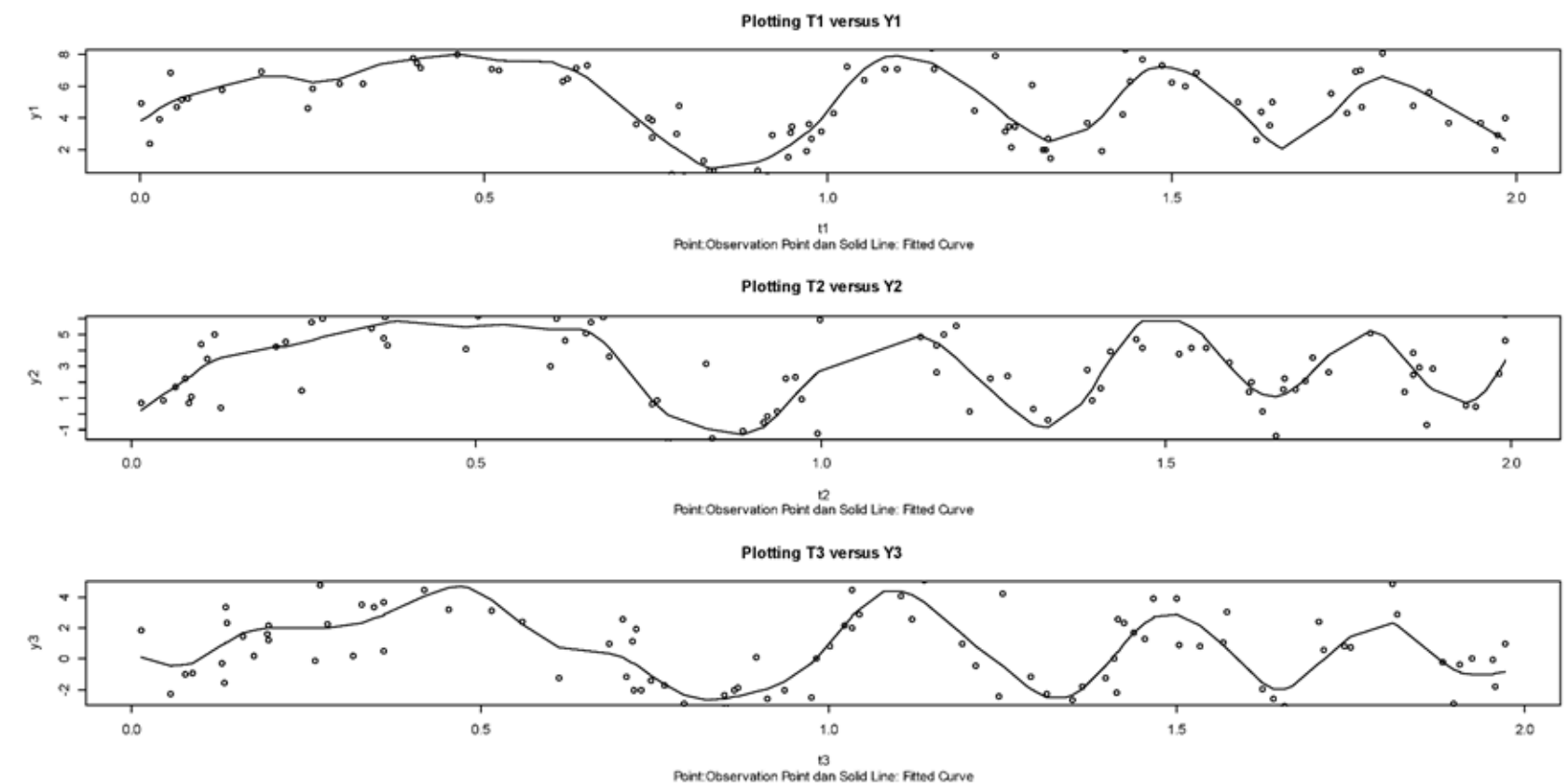
<https://www.mdpi.com/2073-8994/14/11/2227>



# Function Estimation and Reproducing Kernels

Supongamos que la base  $h$  surge de la expansión propia (posiblemente finita) de un núcleo definido positivo  $K$ ,

$$K(x, x') = \sum_{m=1}^{\infty} \phi_m(x) \phi_m(x') \delta_m$$



<https://www.mdpi.com/2073-8994/14/11/2227>

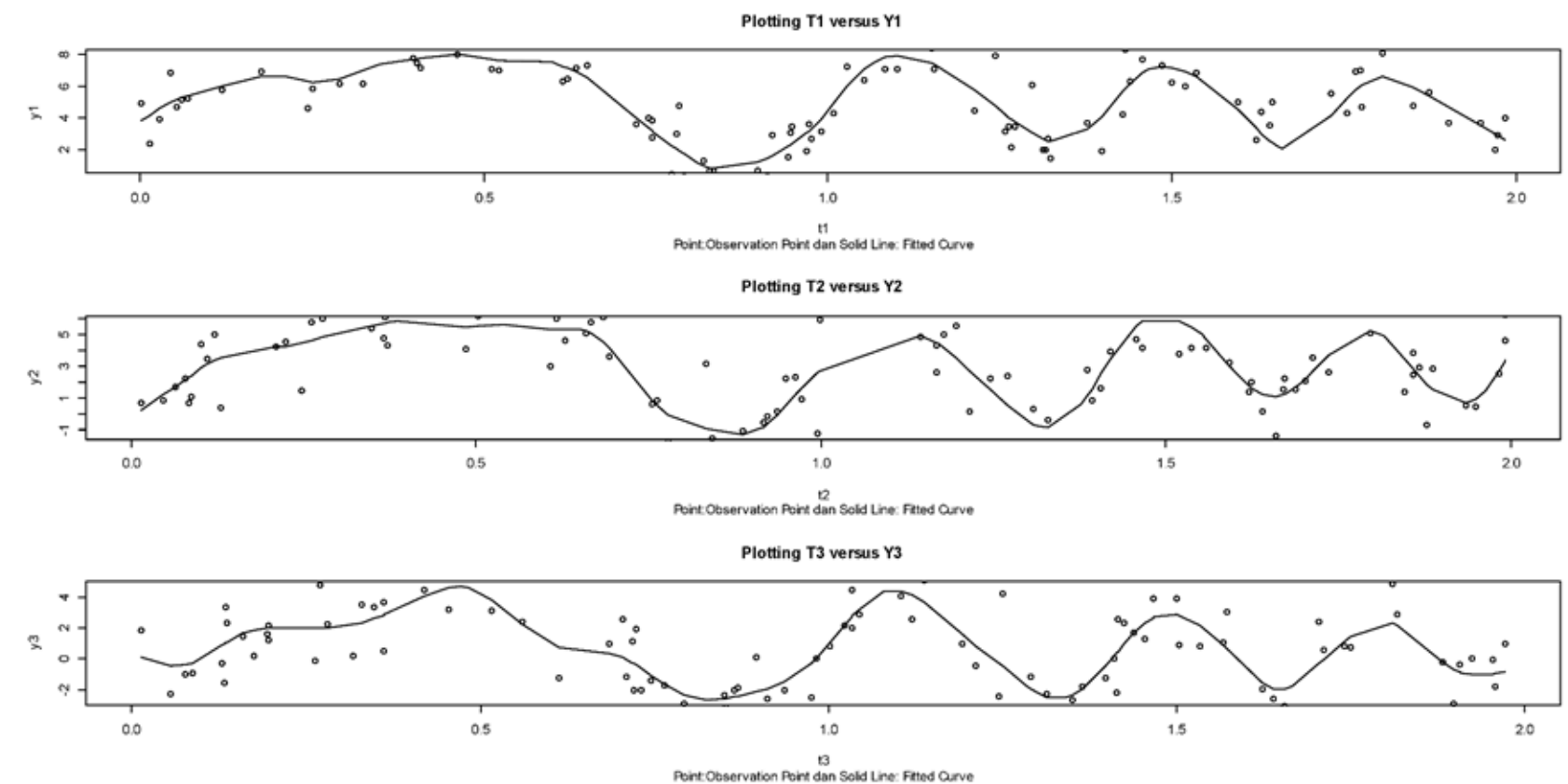


# Function Estimation and Reproducing Kernels

Supongamos que la base  $h$  surge de la expansión propia (posiblemente finita) de un núcleo definido positivo  $K$ ,

$$K(x, x') = \sum_{m=1}^{\infty} \phi_m(x) \phi_m(x') \delta_m$$

y  $h_m(x) = \sqrt{\delta_m} \phi_m(x)$ . entonces, con  $\theta_m = \sqrt{\delta_m} \beta_m$ , podemos escribir:



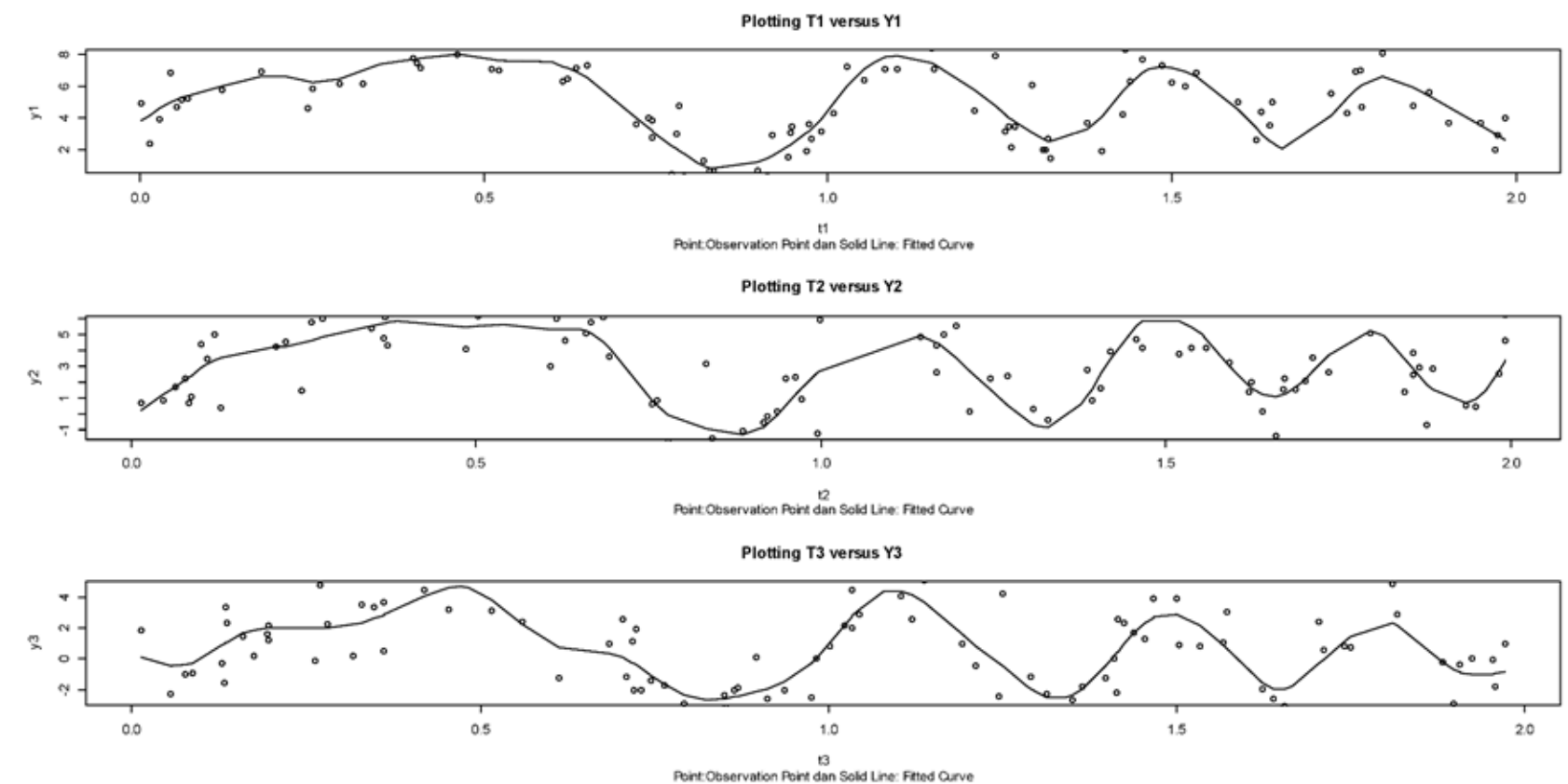
# Function Estimation and Reproducing Kernels

Supongamos que la base  $h$  surge de la expansión propia (posiblemente finita) de un núcleo definido positivo  $K$ ,

$$K(x, x') = \sum_{m=1}^{\infty} \phi_m(x) \phi_m(x') \delta_m$$

y  $h_m(x) = \sqrt{\delta_m} \phi_m(x)$ . entonces, con  $\theta_m = \sqrt{\delta_m} \beta_m$ , podemos escribir:

$$\min_{\beta_0, \theta} \sum_{i=1}^N \left[ 1 - y_i \left( \beta_0 + \sum_{m=1}^{\infty} \theta_m \phi_m(x_i) \right) \right]_+ + \frac{\lambda}{2} \sum_{m=1}^{\infty} \frac{\theta_m^2}{\delta_m}.$$

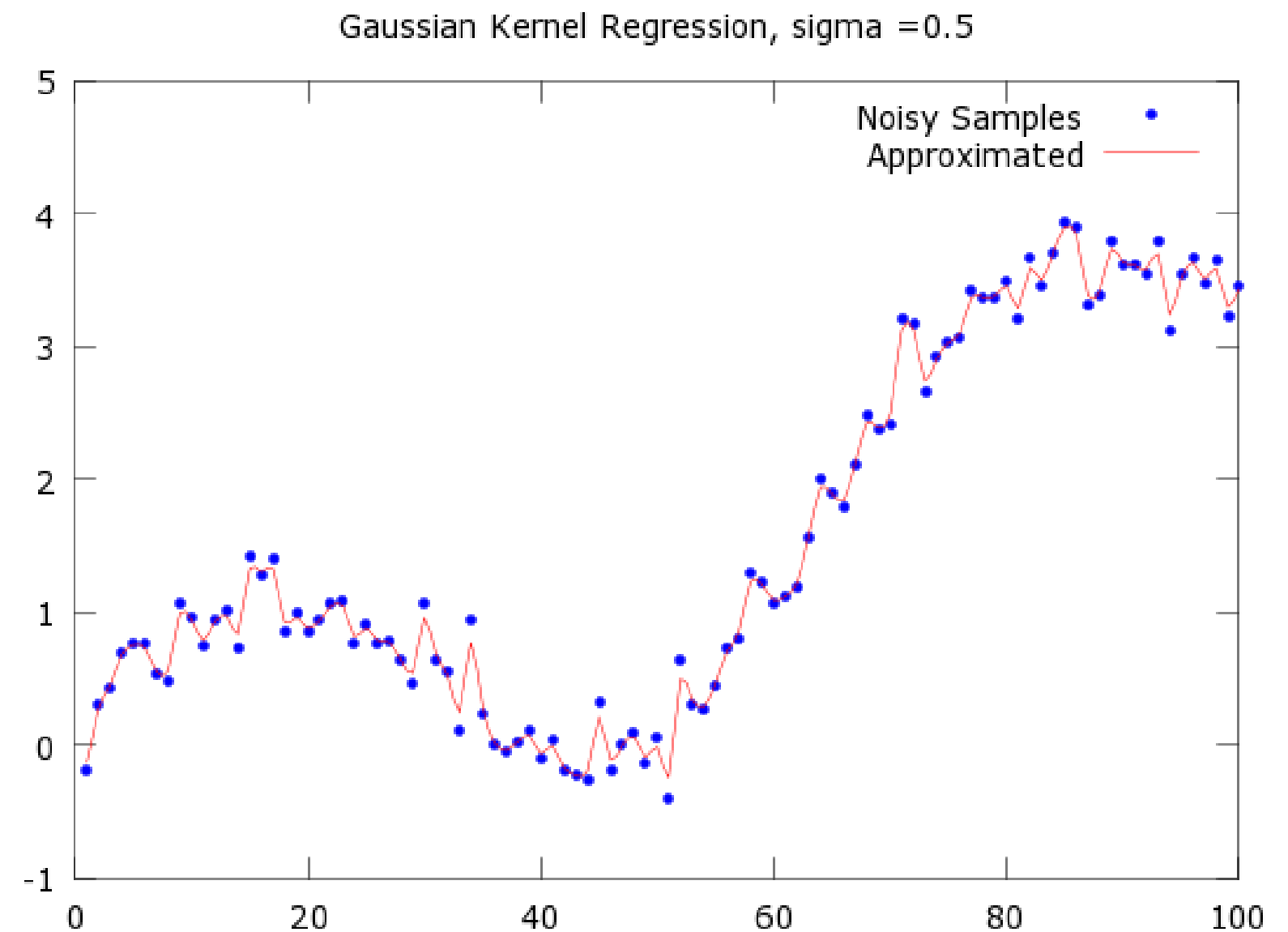


<https://www.mdpi.com/2073-8994/14/11/2227>



# Function Estimation and Reproducing Kernels

La teoría de los espacios de Hilbert con núcleo reproductor que se describe allí garantiza una solución de dimensión finita de la forma



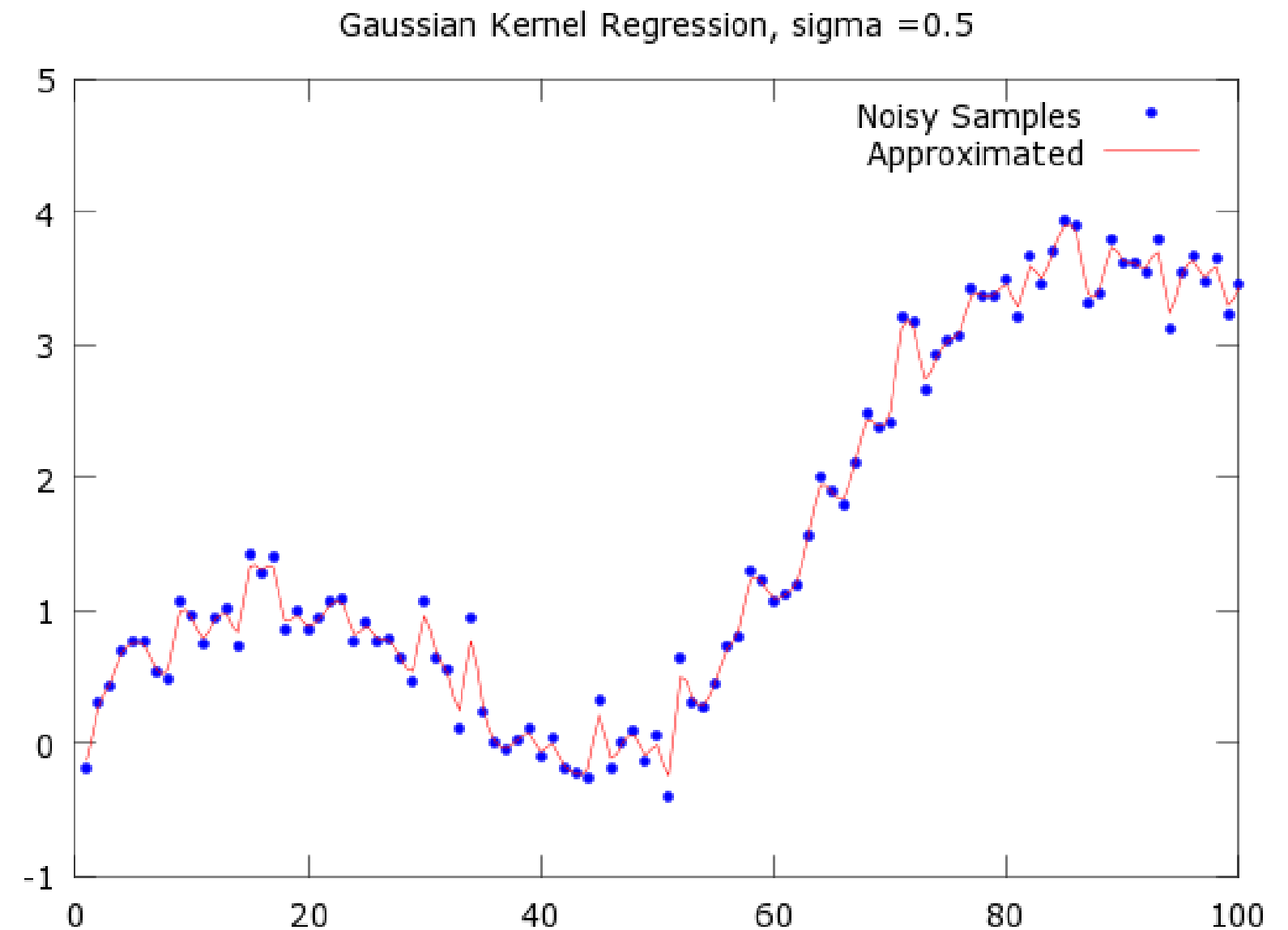
<https://chrisjmccormick.wordpress.com/2014/02/26/kernel-regression/>



# Function Estimation and Reproducing Kernels

La teoría de los espacios de Hilbert con núcleo reproductor que se describe allí garantiza una solución de dimensión finita de la forma

$$f(x) = \beta_0 + \sum_{i=1}^N \alpha_i K(x, x_i).$$



<https://chrisjmccormick.wordpress.com/2014/02/26/kernel-regression/>

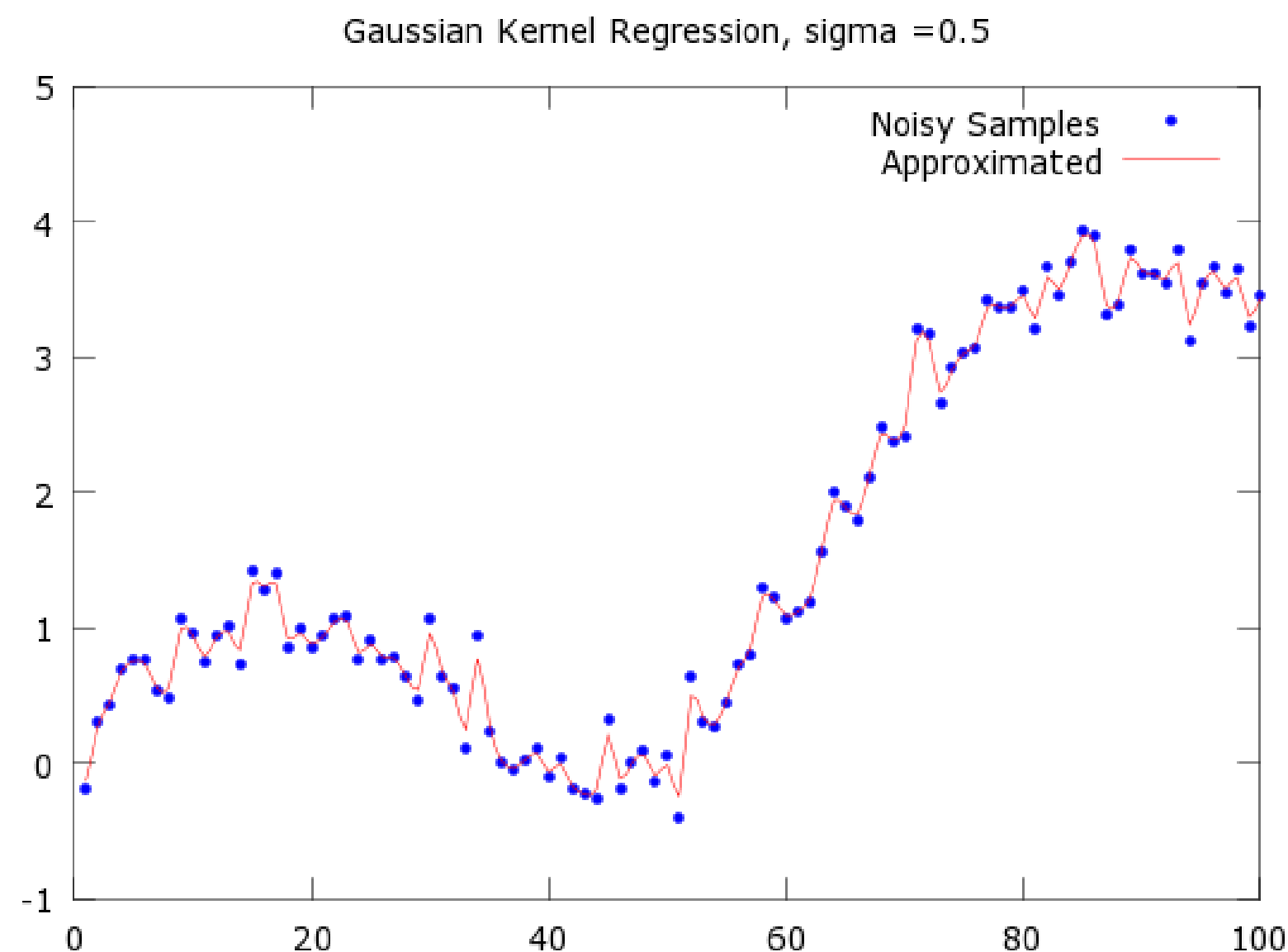


# Function Estimation and Reproducing Kernels

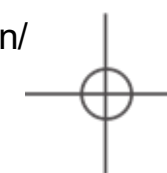
La teoría de los espacios de Hilbert con núcleo reproductor que se describe allí garantiza una solución de dimensión finita de la forma

$$f(x) = \beta_0 + \sum_{i=1}^N \alpha_i K(x, x_i).$$

$$\min_{\beta_0, \alpha} \sum_{i=1}^N (1 - y_i f(x_i))_+ + \frac{\lambda}{2} \alpha^T \mathbf{K} \alpha,$$



<https://chrisjmccormick.wordpress.com/2014/02/26/kernel-regression/>



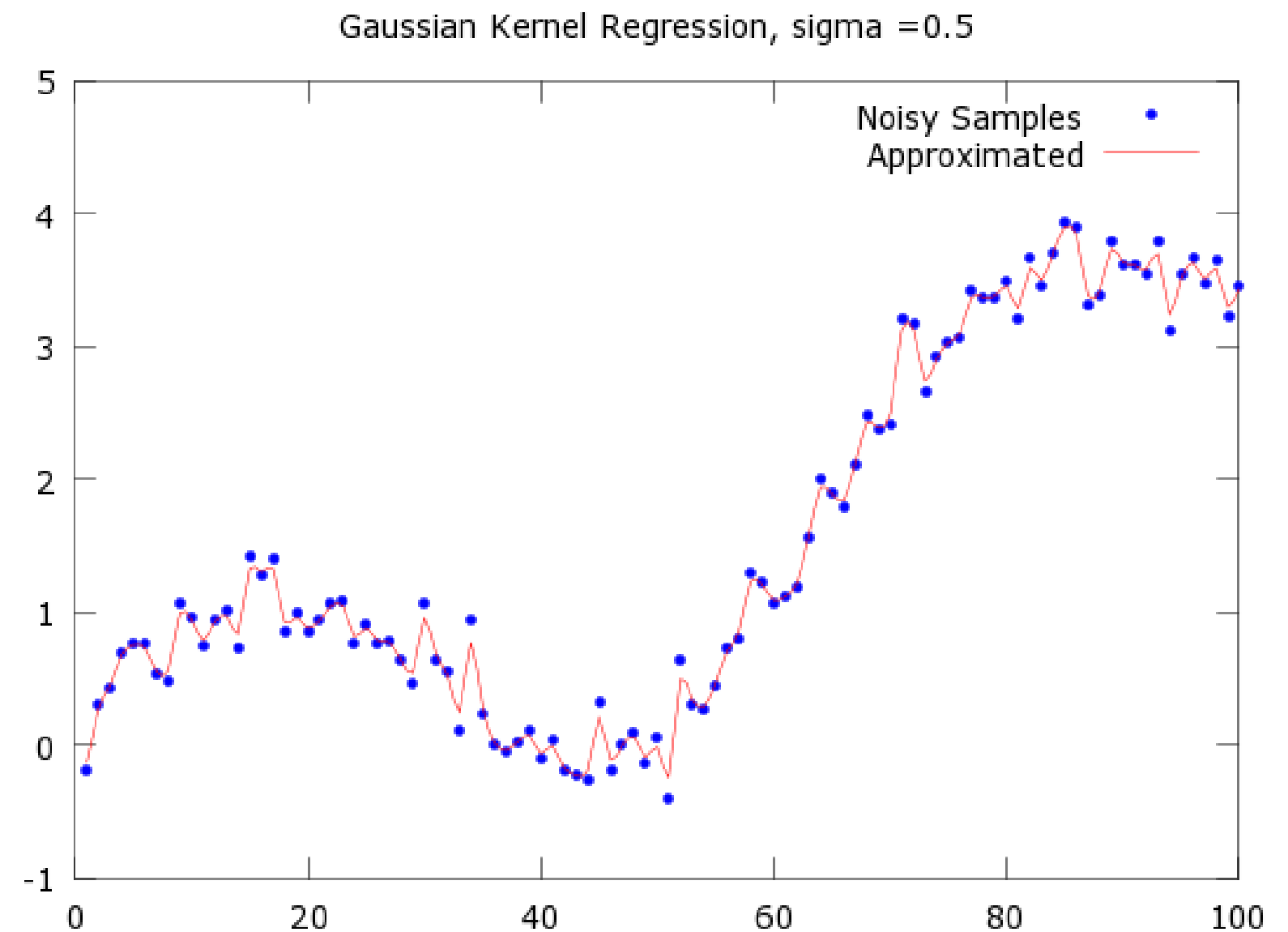
# Function Estimation and Reproducing Kernels

La teoría de los espacios de Hilbert con núcleo reproductor que se describe allí garantiza una solución de dimensión finita de la forma

$$f(x) = \beta_0 + \sum_{i=1}^N \alpha_i K(x, x_i).$$

$$\min_{\beta_0, \alpha} \sum_{i=1}^N (1 - y_i f(x_i))_+ + \frac{\lambda}{2} \alpha^T \mathbf{K} \alpha,$$

donde  $\mathbf{K}$  es la matriz  $N \times N$  de evaluaciones del núcleo para todos los pares de características de entrenamiento.



# Function Estimation and Reproducing Kernels

La teoría de los espacios de Hilbert con núcleo reproductor que se describe allí garantiza una solución de dimensión finita de la forma

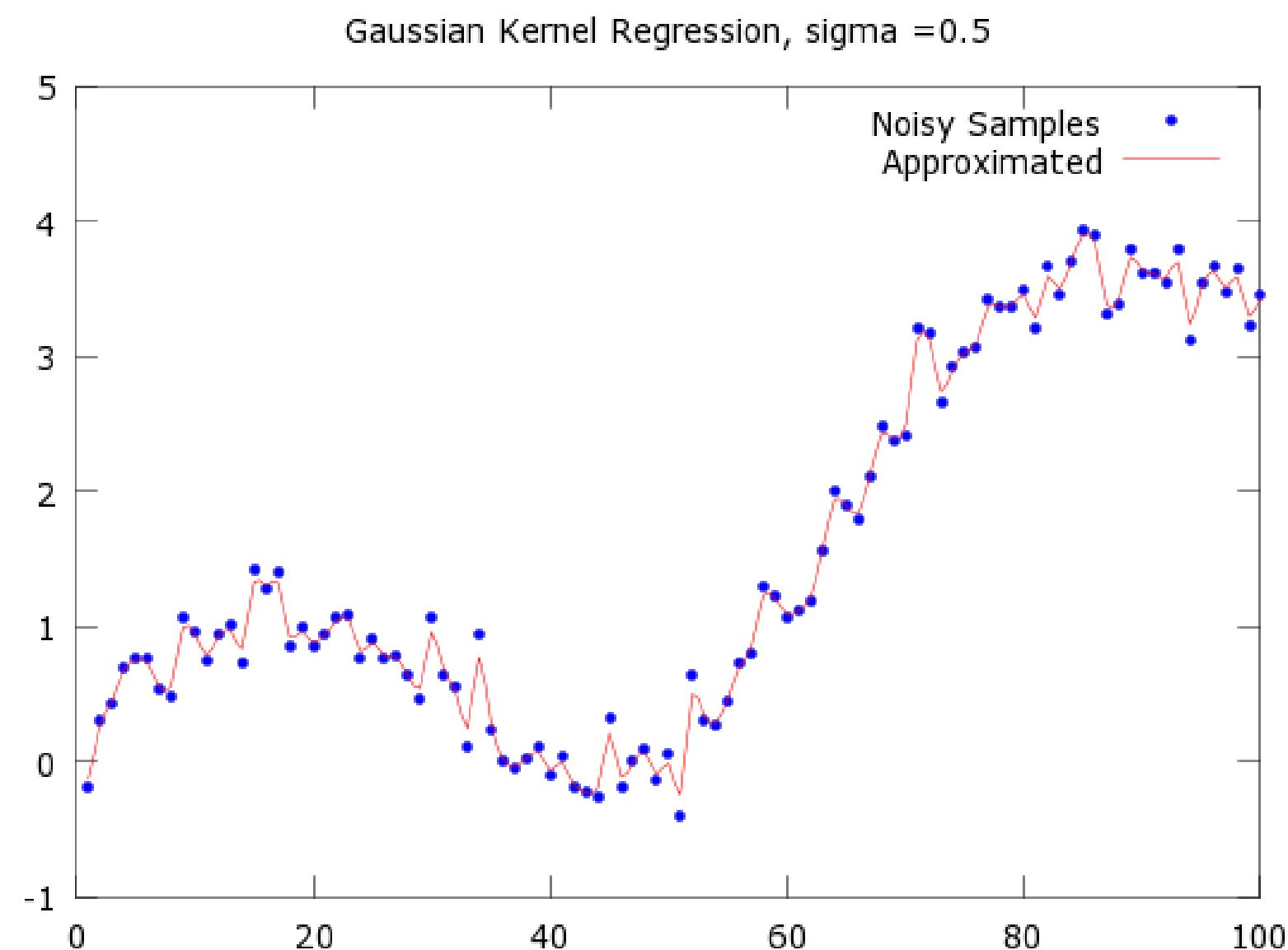
$$f(x) = \beta_0 + \sum_{i=1}^N \alpha_i K(x, x_i).$$

$$\min_{\beta_0, \alpha} \sum_{i=1}^N (1 - y_i f(x_i))_+ + \frac{\lambda}{2} \alpha^T \mathbf{K} \alpha,$$

donde  $\mathbf{K}$  es la matriz  $N \times N$  de evaluaciones del núcleo para todos los pares de características de entrenamiento.

$$\min_{f \in \mathcal{H}} \sum_{i=1}^N [1 - y_i f(x_i)]_+ + \lambda J(f),$$

donde  $\mathcal{H}$  es el espacio estructurado de funciones, y  $J(f)$  un regularizador apropiado en ese espacio.



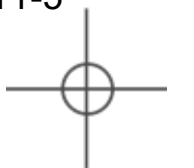
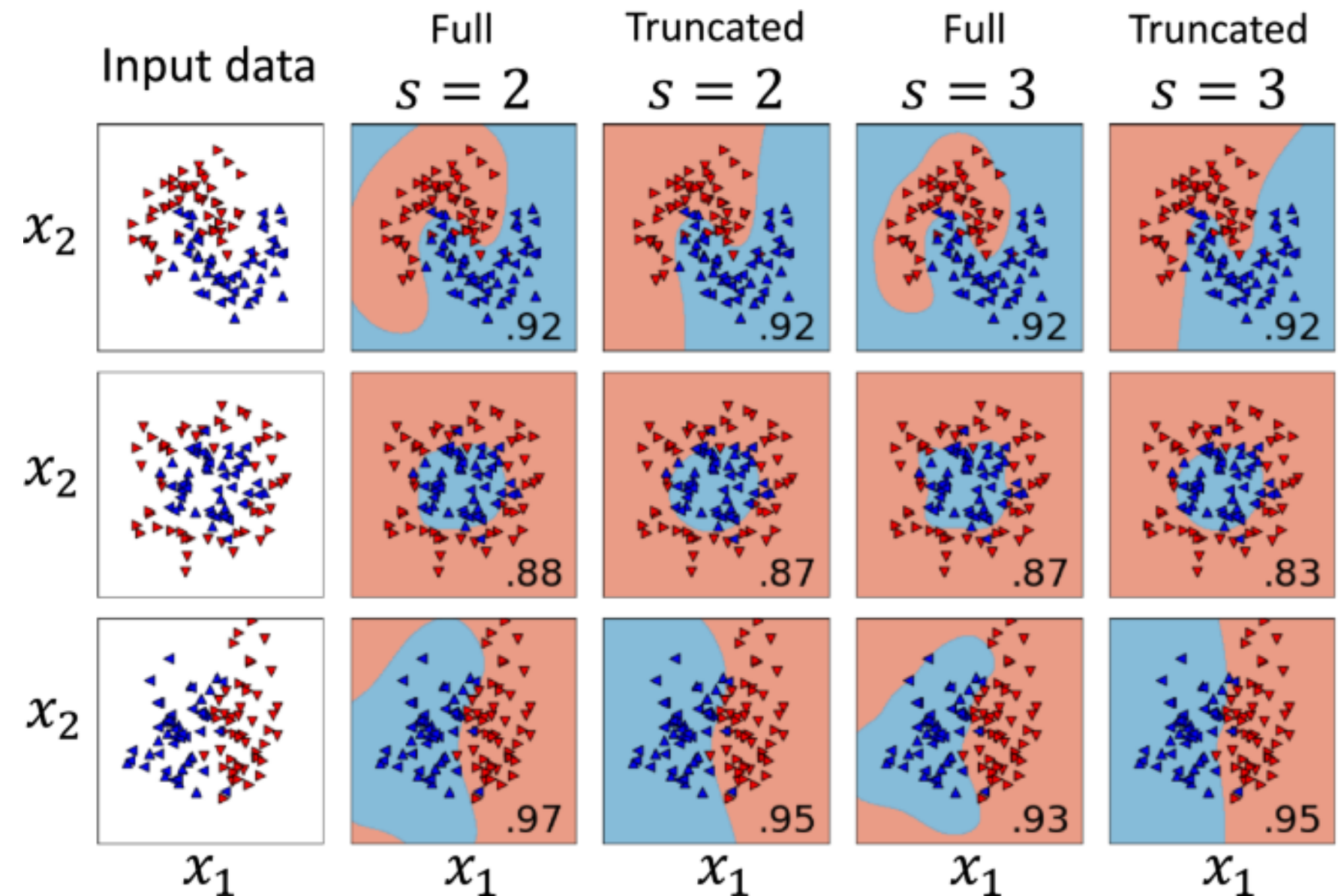
<https://chrisjmccormick.wordpress.com/2014/02/26/kernel-regression/>



# Function Estimation and Reproducing Kernels

La función ajustada es una estimación de la log-odds,

$$\begin{aligned}\hat{f}(x) &= \log \frac{\hat{\Pr}(Y = +1|x)}{\hat{\Pr}(Y = -1|x)} \\ &= \hat{\beta}_0 + \sum_{i=1}^N \hat{\alpha}_i K(x, x_i),\end{aligned}$$

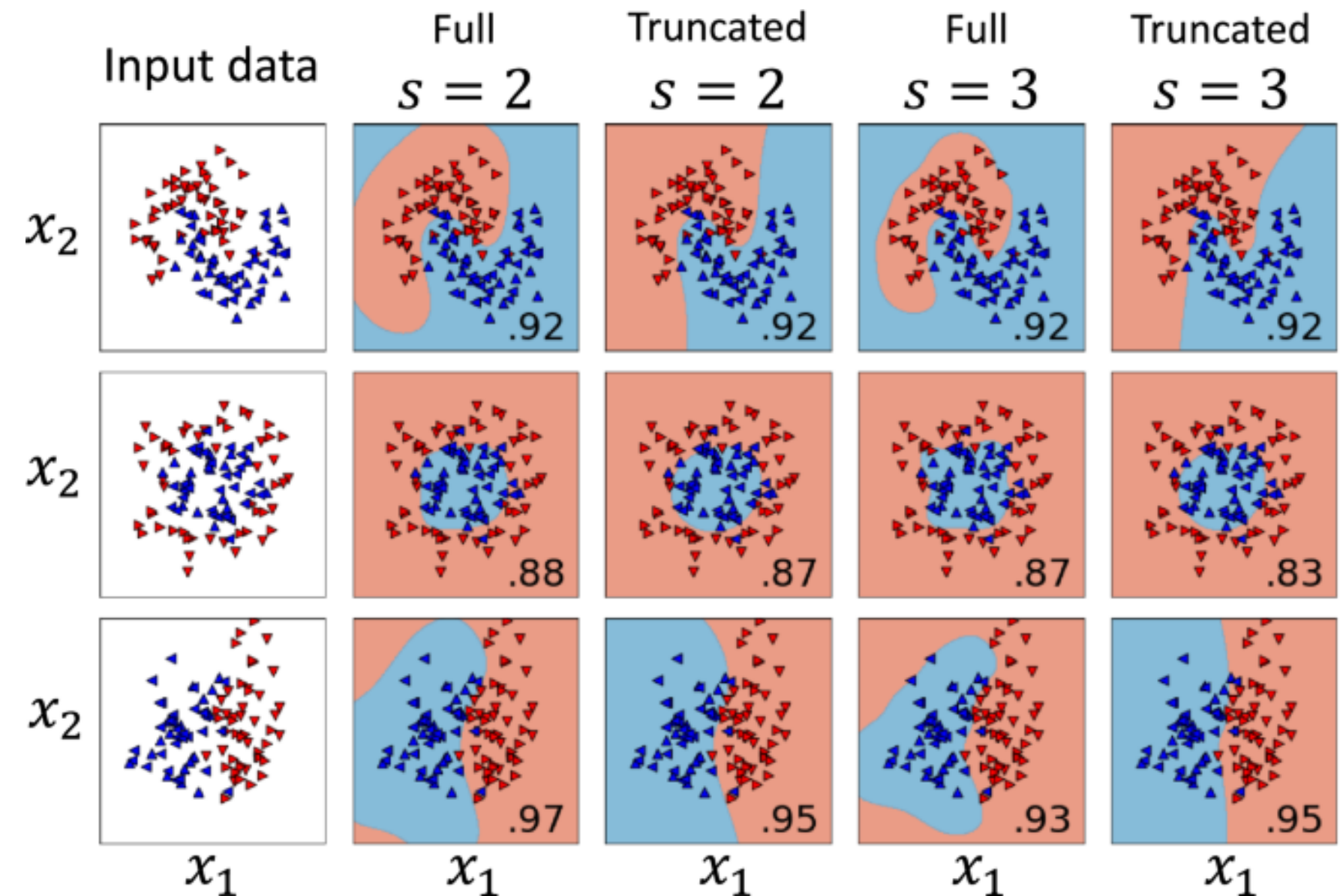


# Function Estimation and Reproducing Kernels

La función ajustada es una estimación de la log-odds,

$$\begin{aligned}\hat{f}(x) &= \log \frac{\hat{\Pr}(Y = +1|x)}{\hat{\Pr}(Y = -1|x)} \\ &= \hat{\beta}_0 + \sum_{i=1}^N \hat{\alpha}_i K(x, x_i),\end{aligned}$$

o, por el contrario, obtenemos una estimación de las probabilidades de clase.



# Function Estimation and Reproducing Kernels

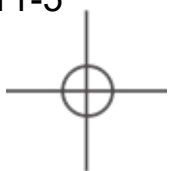
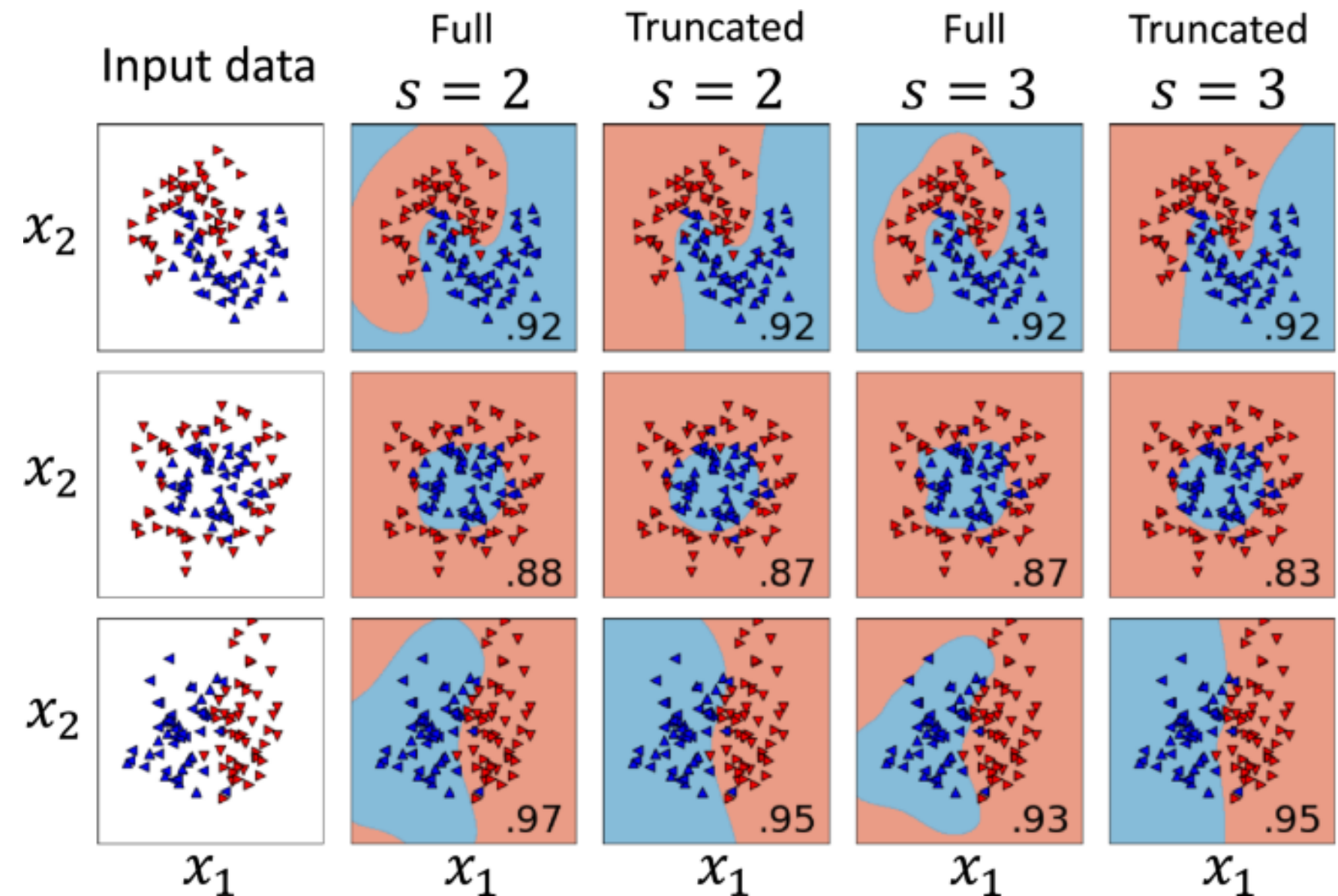
La función ajustada es una estimación de la log-odds,

$$\begin{aligned}\hat{f}(x) &= \log \frac{\hat{\text{Pr}}(Y = +1|x)}{\hat{\text{Pr}}(Y = -1|x)} \\ &= \hat{\beta}_0 + \sum_{i=1}^N \hat{\alpha}_i K(x, x_i),\end{aligned}$$

o, por el contrario, obtenemos una estimación de las probabilidades de clase.

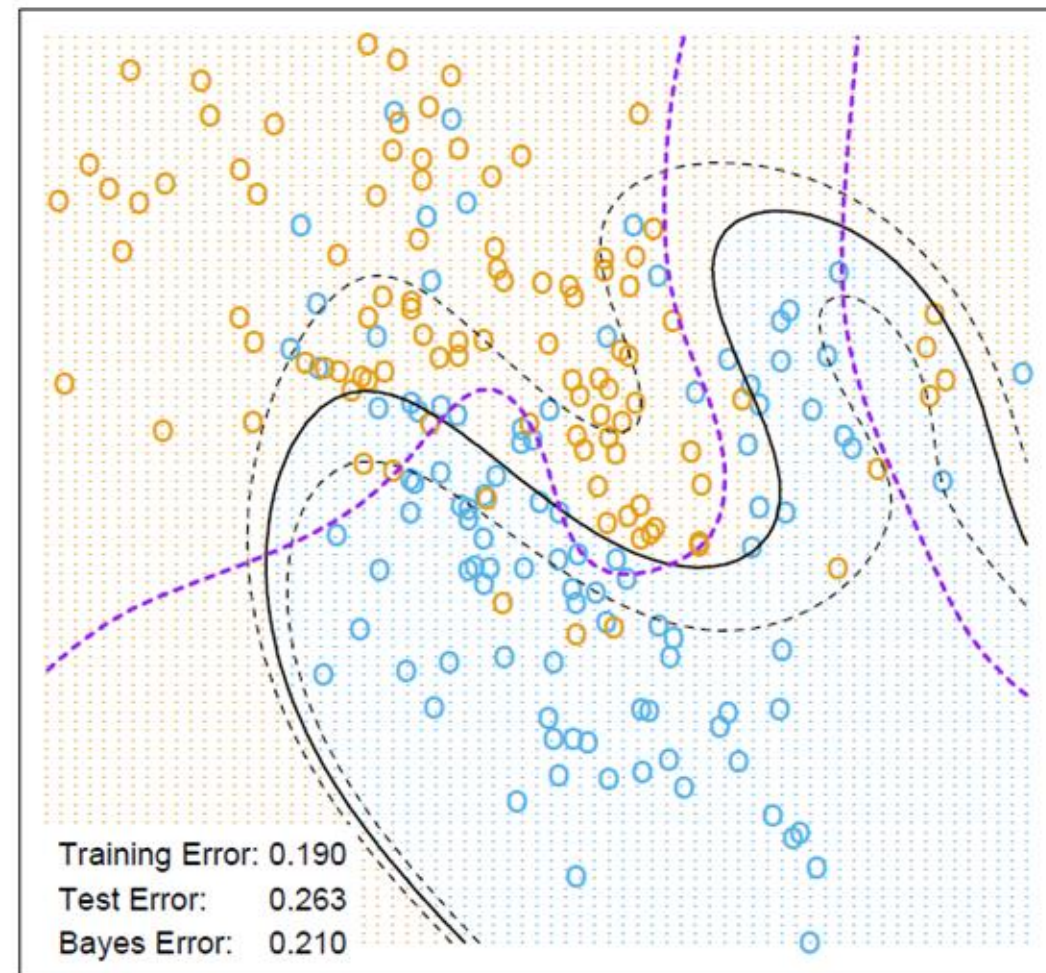
$$\hat{\text{Pr}}(Y = +1|x) = \frac{1}{1 + e^{-\hat{\beta}_0 - \sum_{i=1}^N \hat{\alpha}_i K(x, x_i)}}.$$

Los modelos ajustados son bastante similares en forma y rendimiento.

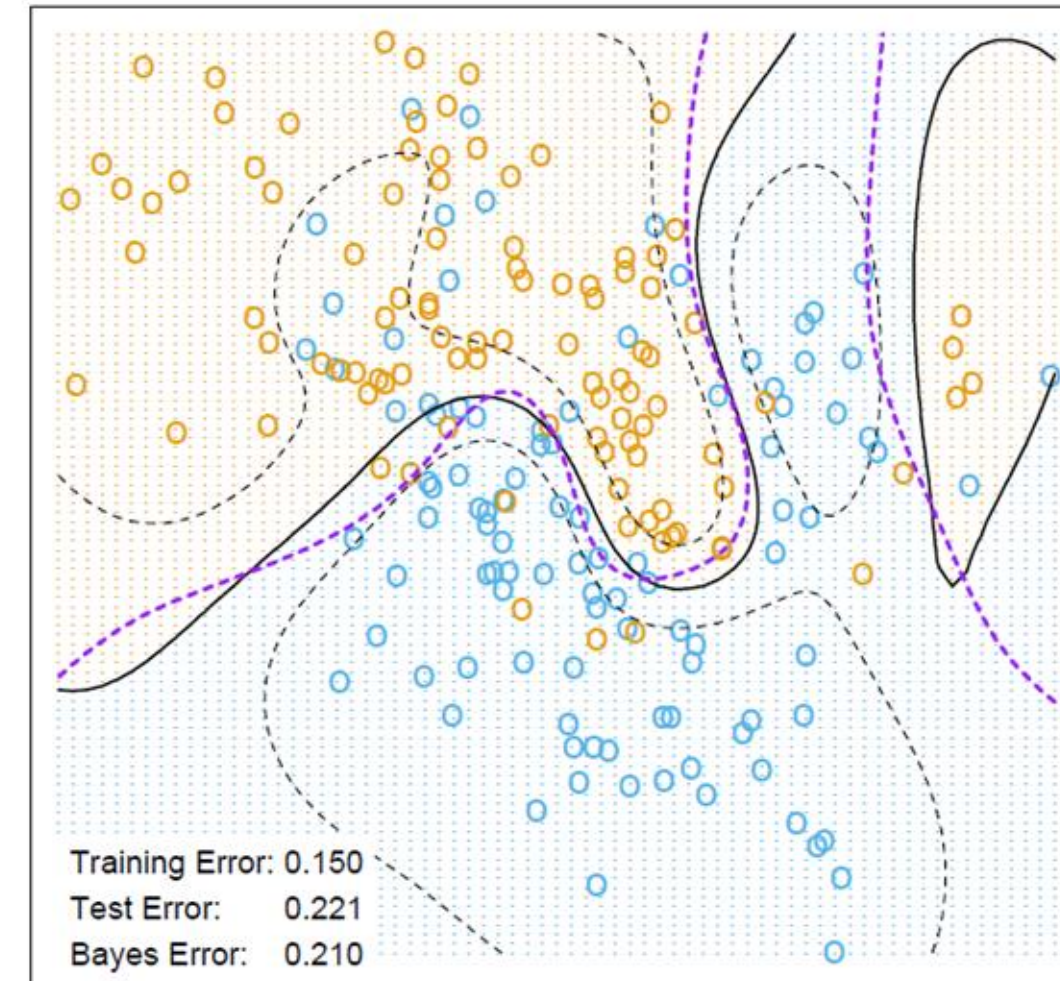


# Function Estimation and Reproducing Kernels

LR - Degree-4 Polynomial in Feature Space



LR - Radial Kernel in Feature Space



Los dos contornos discontinuos corresponden a probabilidades a posteriori de 0,75 y 0,25 para la clase +1 (o viceversa). La curva discontinua de color púrpura del fondo es el límite de decisión de Bayes.



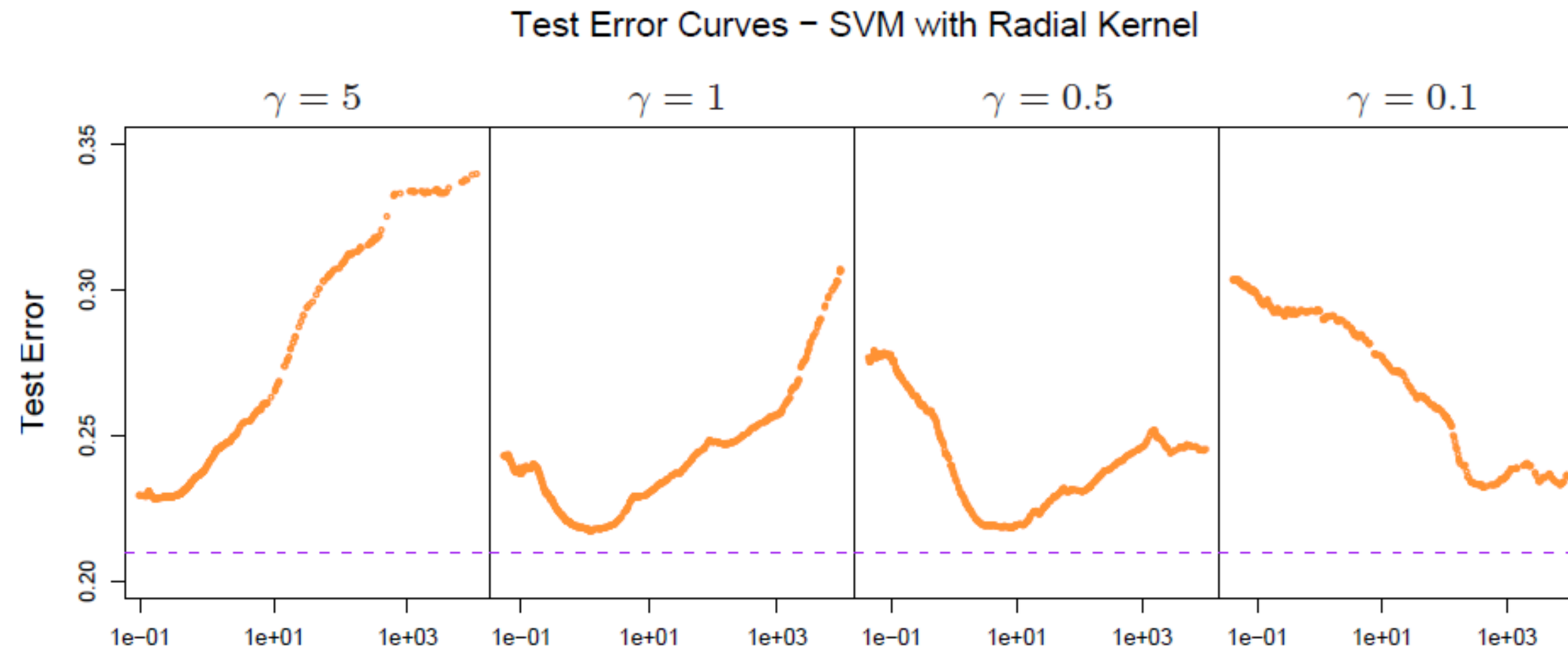
# Function Estimation and Reproducing Kernels

Method	Test Error (SE)	
	No Noise Features	Six Noise Features
1 SV Classifier	0.450 (0.003)	0.472 (0.003)
2 SVM/poly 2	0.078 (0.003)	0.152 (0.004)
3 SVM/poly 5	0.180 (0.004)	0.370 (0.004)
4 SVM/poly 10	0.230 (0.003)	0.434 (0.002)
5 BRUTO	0.084 (0.003)	0.090 (0.003)
6 MARS	0.156 (0.004)	0.173 (0.005)
Bayes	0.029	0.029

Piel de la naranja: Se muestra la media (error estándar de la media) del error de la prueba en 50 simulaciones. BRUTO ajusta un modelo spline aditivo de forma adaptativa, mientras que MARS ajusta un modelo de interacción de bajo orden de forma adaptativa.



# Function Estimation and Reproducing Kernels



Curvas de error de prueba en función del parámetro de coste  $C$  para el clasificador SVM de núcleo radial en los datos mixtos. En la parte superior de cada gráfico se encuentra el parámetro de escala  $\gamma$  para el núcleo radial:  $K_\gamma(x, y) = \exp(-\gamma\|x - y\|^2)$ . El valor óptimo para  $C$  depende en gran medida de la escala del núcleo. La tasa de error bayesiana se indica mediante las líneas horizontales discontinuas.

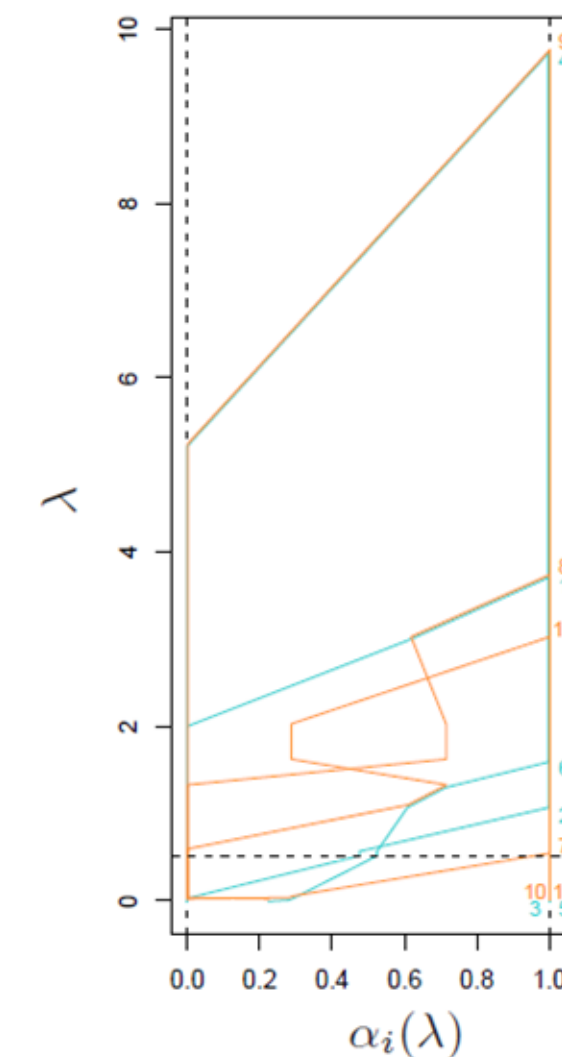
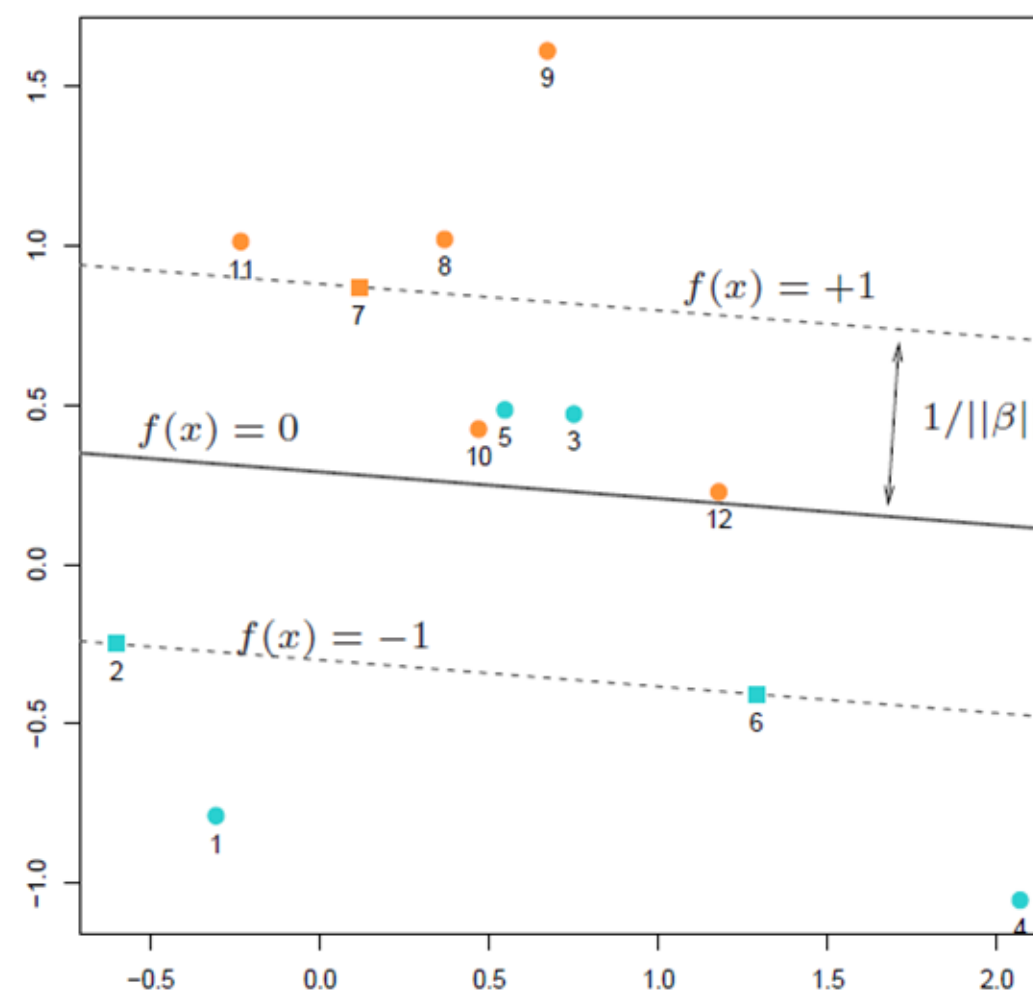


# A Path Algorithm for the SVM Classifier

Un ejemplo sencillo ilustra el algoritmo de ruta SVM. (panel izquierdo:) Este gráfico ilustra el estado del modelo en  $\lambda = 1/2$ . Los puntos «+ 1» son de color naranja y los «-1» son de color azul.

La anchura del margen suave es  $2/||\beta|| = 2 \times 0,587$ . Dos puntos azules {3, 5} están mal clasificados, mientras que los dos puntos naranjas están correctamente clasificados, pero en el lado incorrecto de su margen  $f(x) = +1$ ; cada uno de ellos tiene  $yf(x_i) < 1$ .

Los tres puntos cuadrados {2, 6, 7} están exactamente en sus márgenes. (panel derecho:) Este gráfico muestra los perfiles lineales por tramos  $\alpha_i(\lambda)$ . La línea discontinua horizontal en  $\lambda = 1/2$  indica el estado de  $\alpha_i$  para el modelo del gráfico izquierdo.

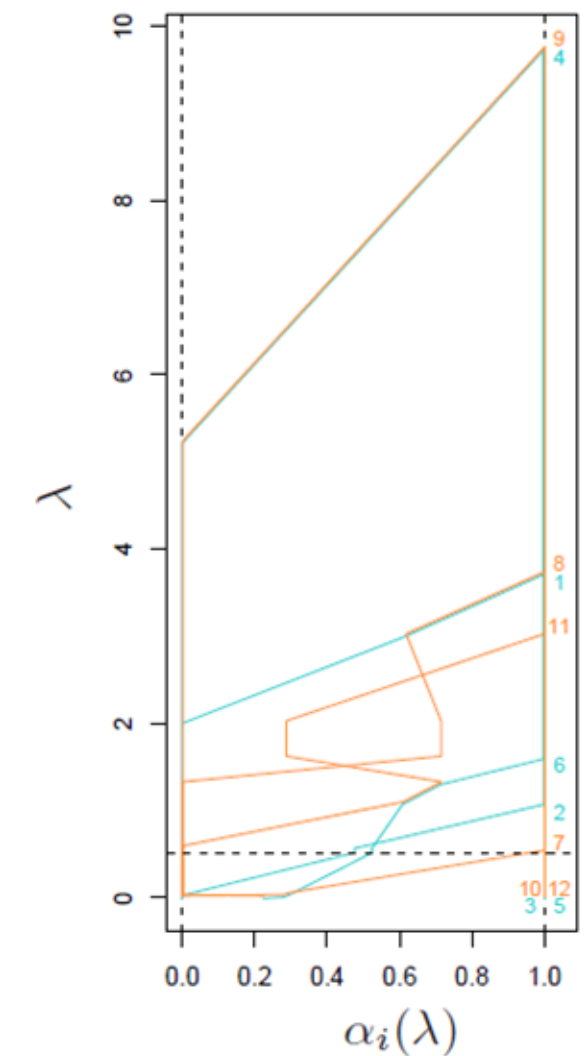
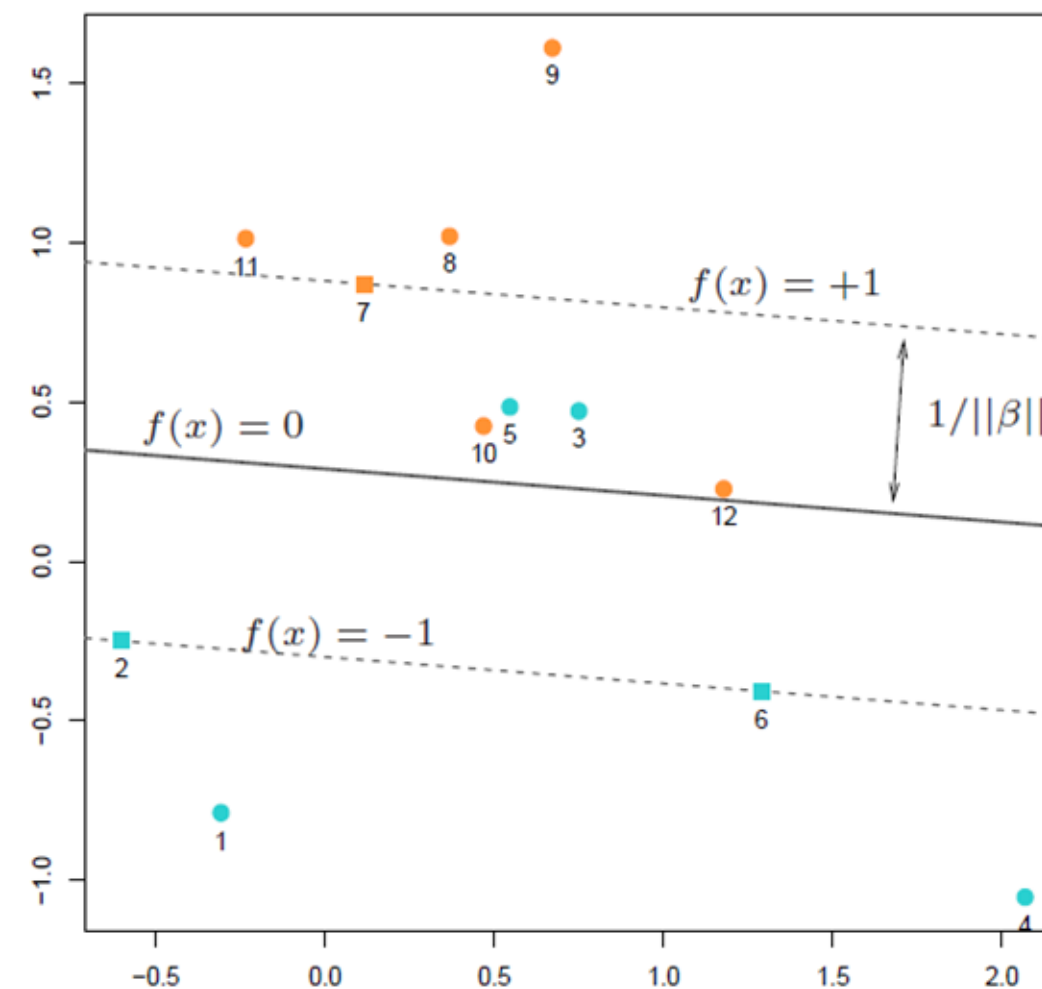


# A Path Algorithm for the SVM Classifier

Un ejemplo sencillo ilustra el algoritmo de ruta SVM. (panel izquierdo:) Este gráfico ilustra el estado del modelo en  $\lambda = 1/2$ . Los puntos «+ 1» son de color naranja y los «-1» son de color azul.

La anchura del margen suave es  $2/||\beta|| = 2 \times 0,587$ . Dos puntos azules {3, 5} están mal clasificados, mientras que los dos puntos naranjas están correctamente clasificados, pero en el lado incorrecto de su margen  $f(x) = +1$ ; cada uno de ellos tiene  $yf(x_i) < 1$ .

Los tres puntos cuadrados {2, 6, 7} están exactamente en sus márgenes. (panel derecho:) Este gráfico muestra los perfiles lineales por tramos  $\alpha_i(\lambda)$ . La línea discontinua horizontal en  $\lambda = 1/2$  indica el estado de  $\alpha_i$  para el modelo del gráfico izquierdo.

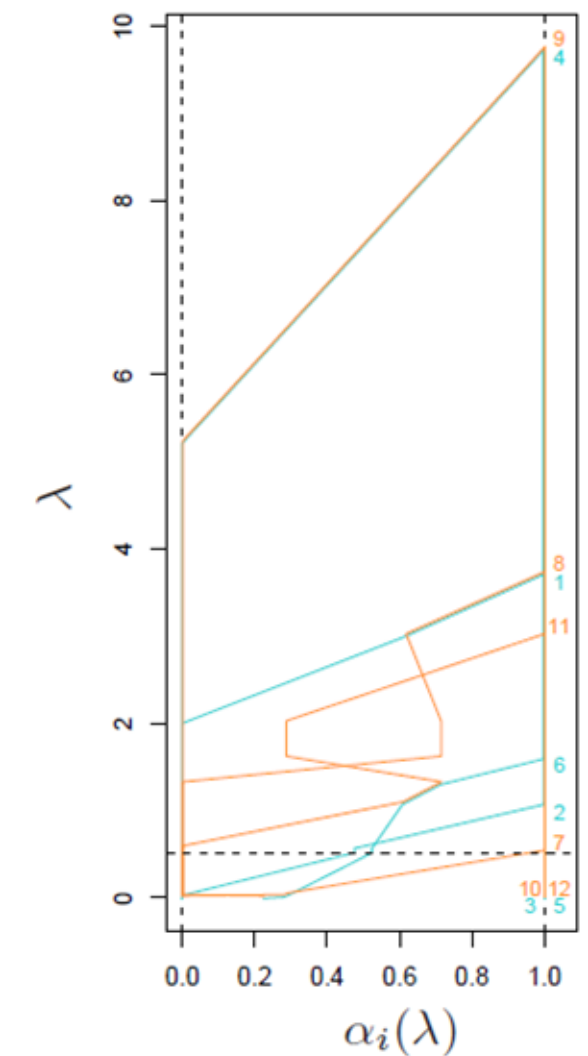
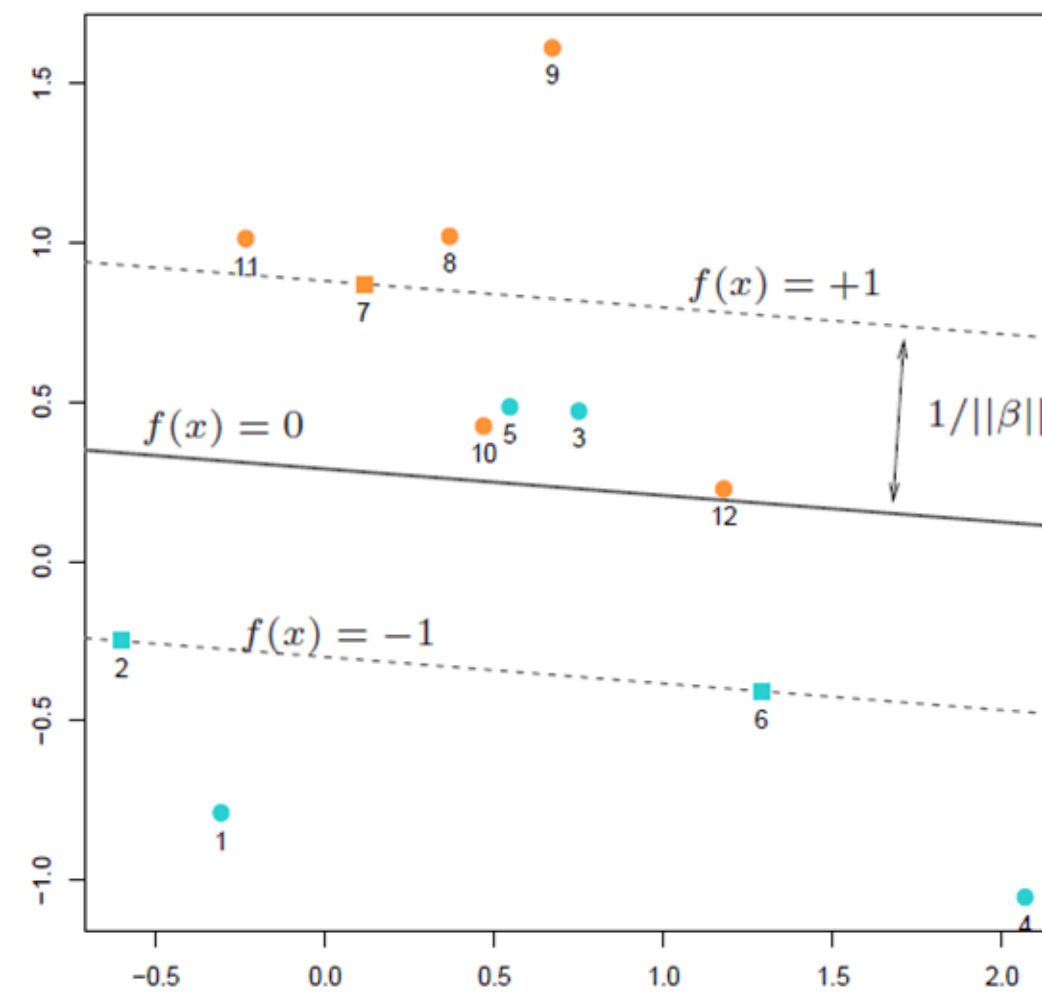


# A Path Algorithm for the SVM Classifier

Describimos un algoritmo de ruta para ajustar de manera eficiente toda la secuencia de modelos SVM obtenidos al variar  $C$ . Es conveniente utilizar la formulación de pérdida + penalización. Esto conduce a una solución para  $\beta$  en un valor dado de  $\lambda$ :

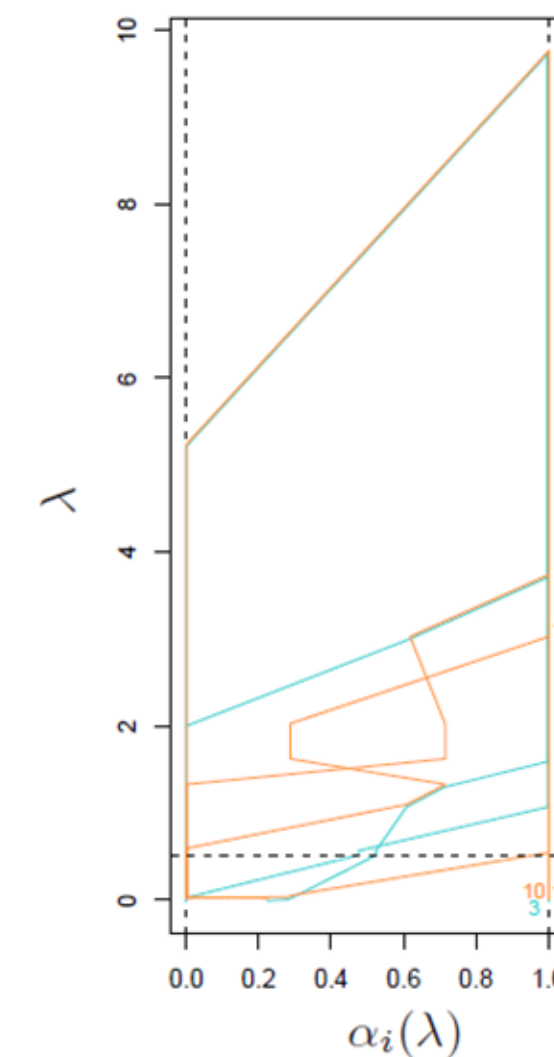
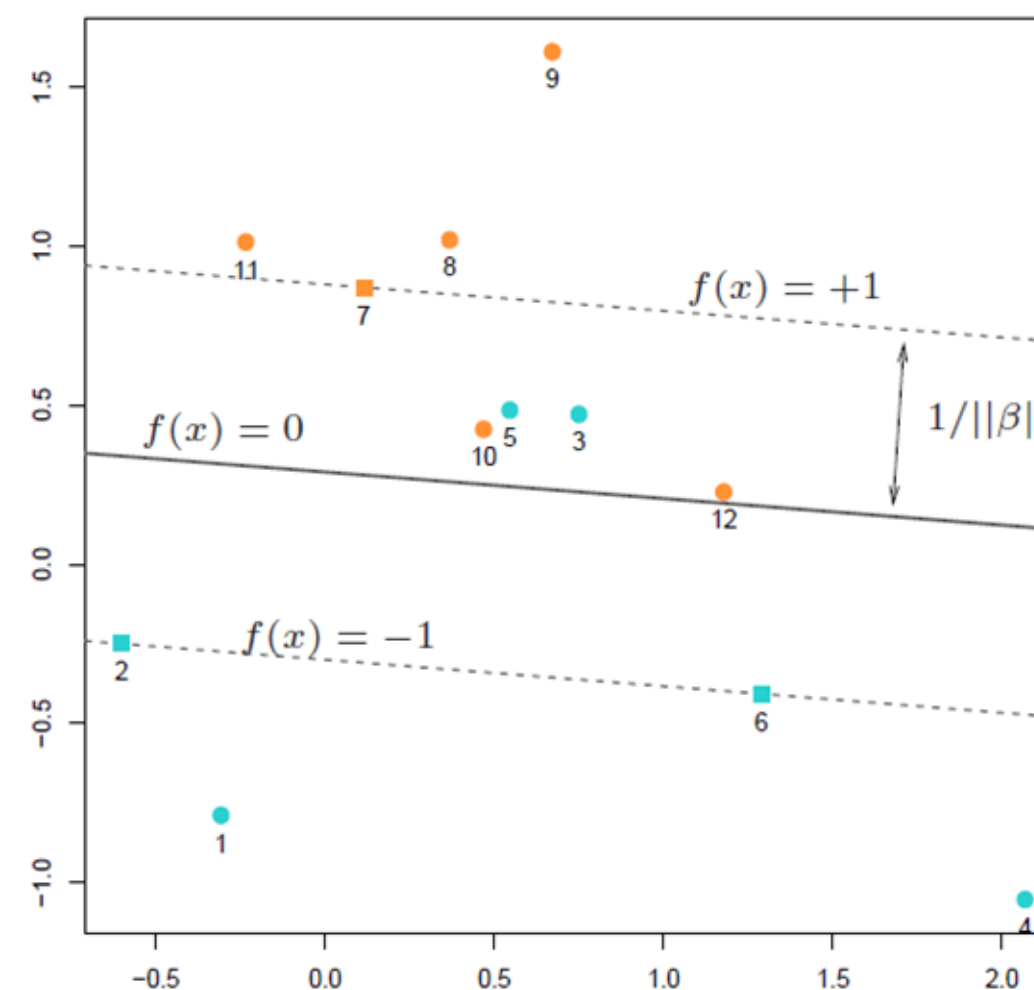
$$\beta_\lambda = \frac{1}{\lambda} \sum_{i=1}^N \alpha_i y_i x_i.$$

Los  $\alpha_i$  son nuevamente multiplicadores de Lagrange, pero en este caso todos se encuentran en  $[0, 1]$ .



# A Path Algorithm for the SVM Classifier

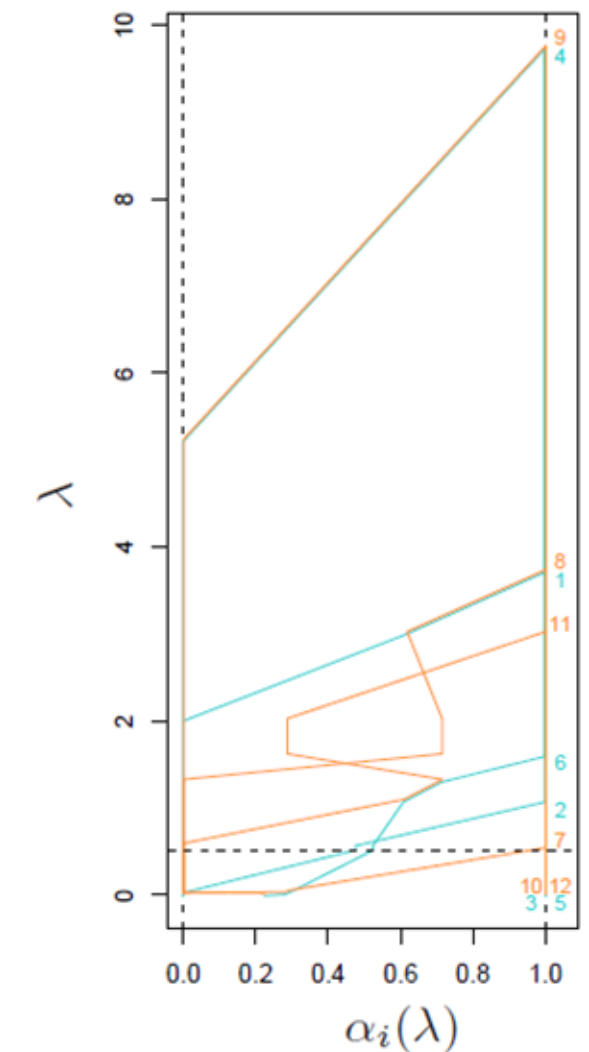
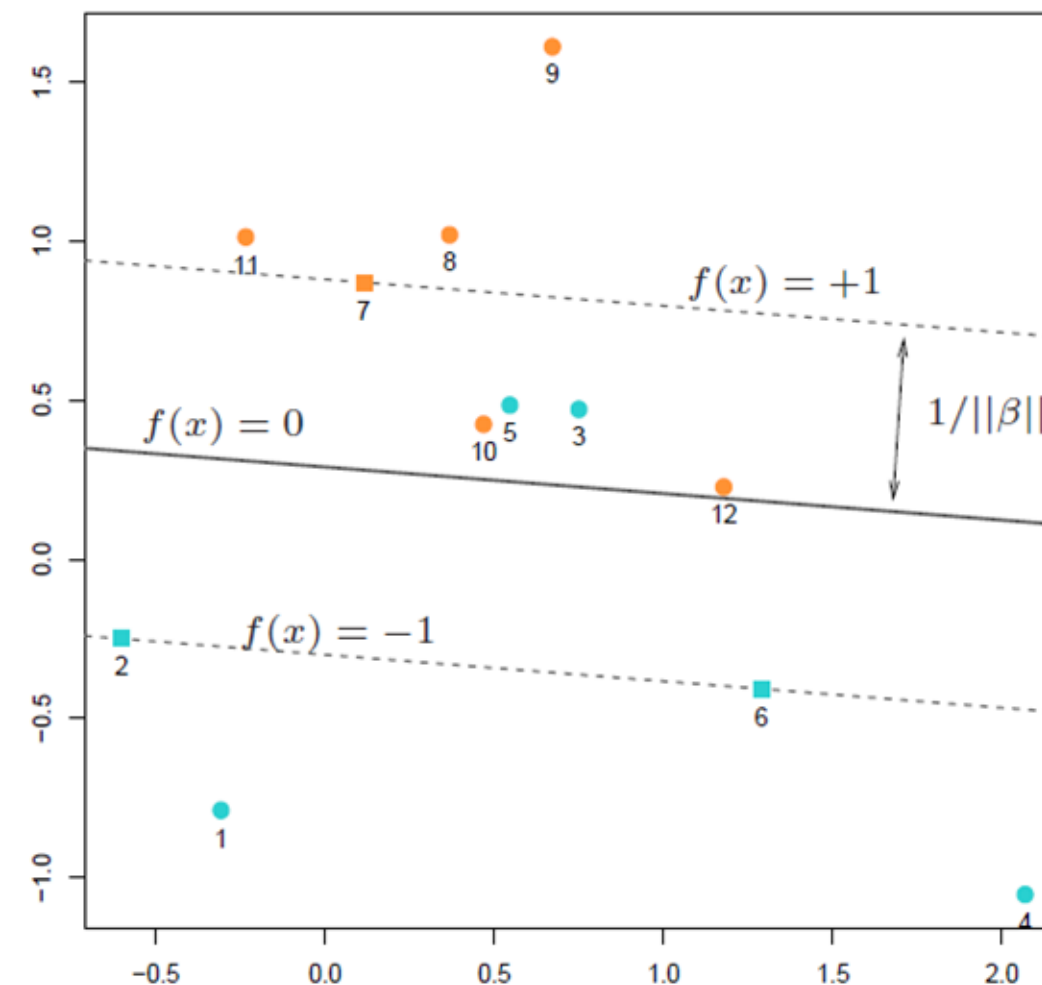
Las condiciones de optimalidad KKT implican que los puntos etiquetados  $(x_i, y_i)$  se dividen en tres grupos distintos:



# A Path Algorithm for the SVM Classifier

Las condiciones de optimalidad KKT implican que los puntos etiquetados  $(x_i, y_i)$  se dividen en tres grupos distintos:

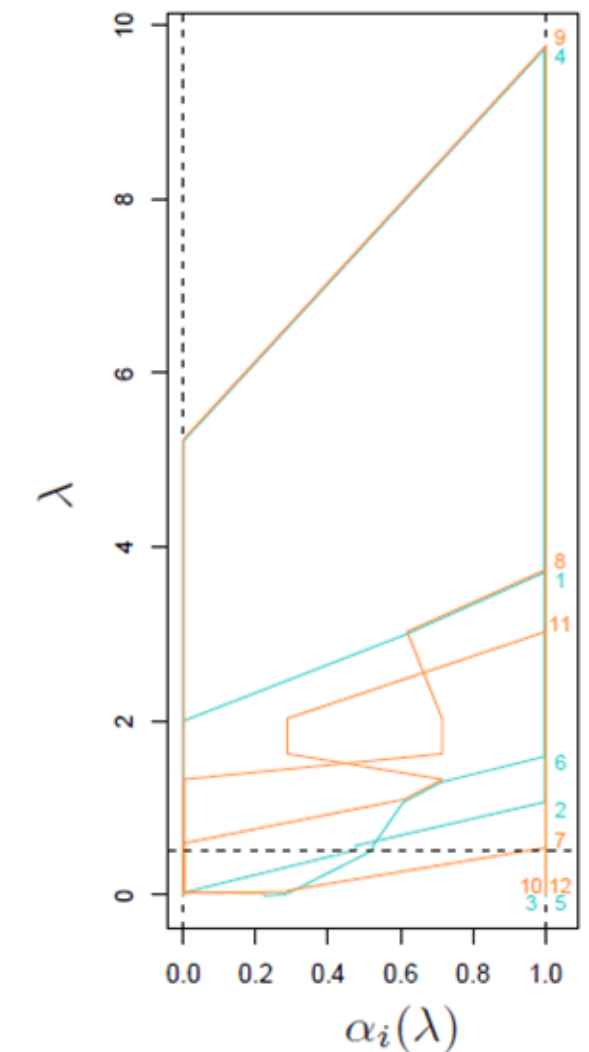
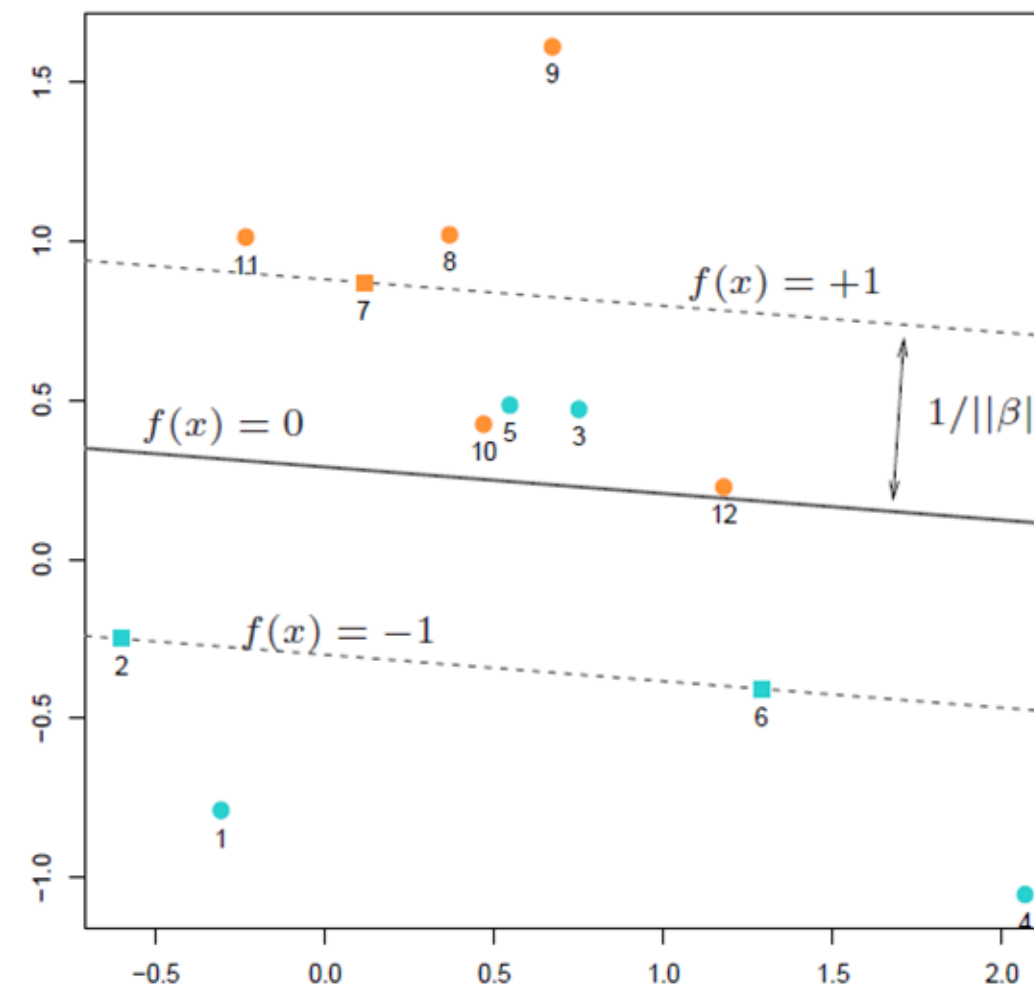
- Observaciones clasificadas correctamente y fuera de sus márgenes. Tienen  $y_i f(x_i) > 1$  y multiplicadores de Lagrange  $\alpha_i = 0$ . Algunos ejemplos son los puntos naranjas 8, 9 y 11, y los puntos azules 1 y 4.



# A Path Algorithm for the SVM Classifier

Las condiciones de optimalidad KKT implican que los puntos etiquetados  $(x_i, y_i)$  se dividen en tres grupos distintos:

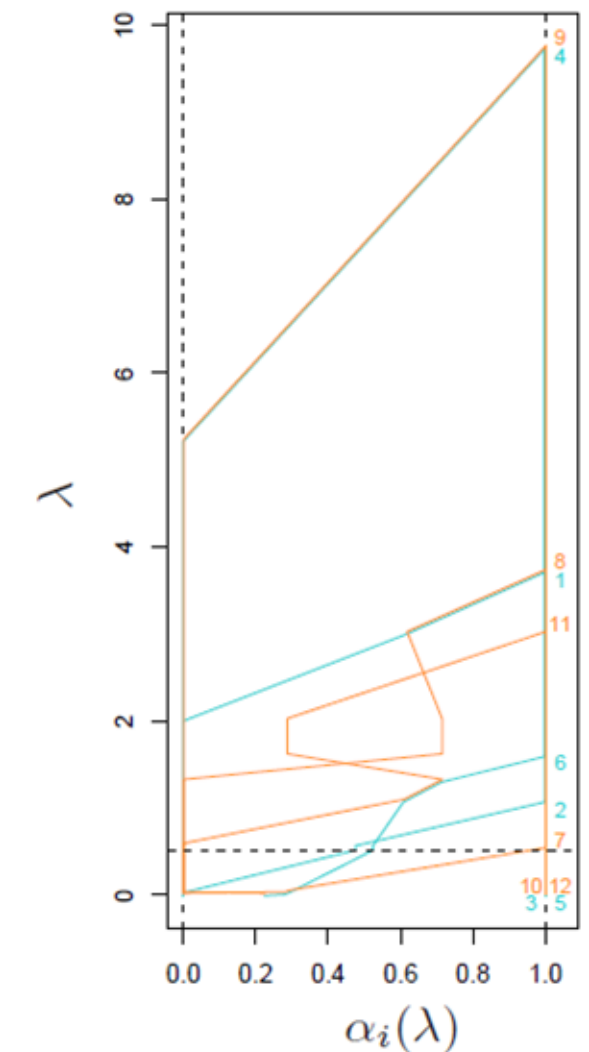
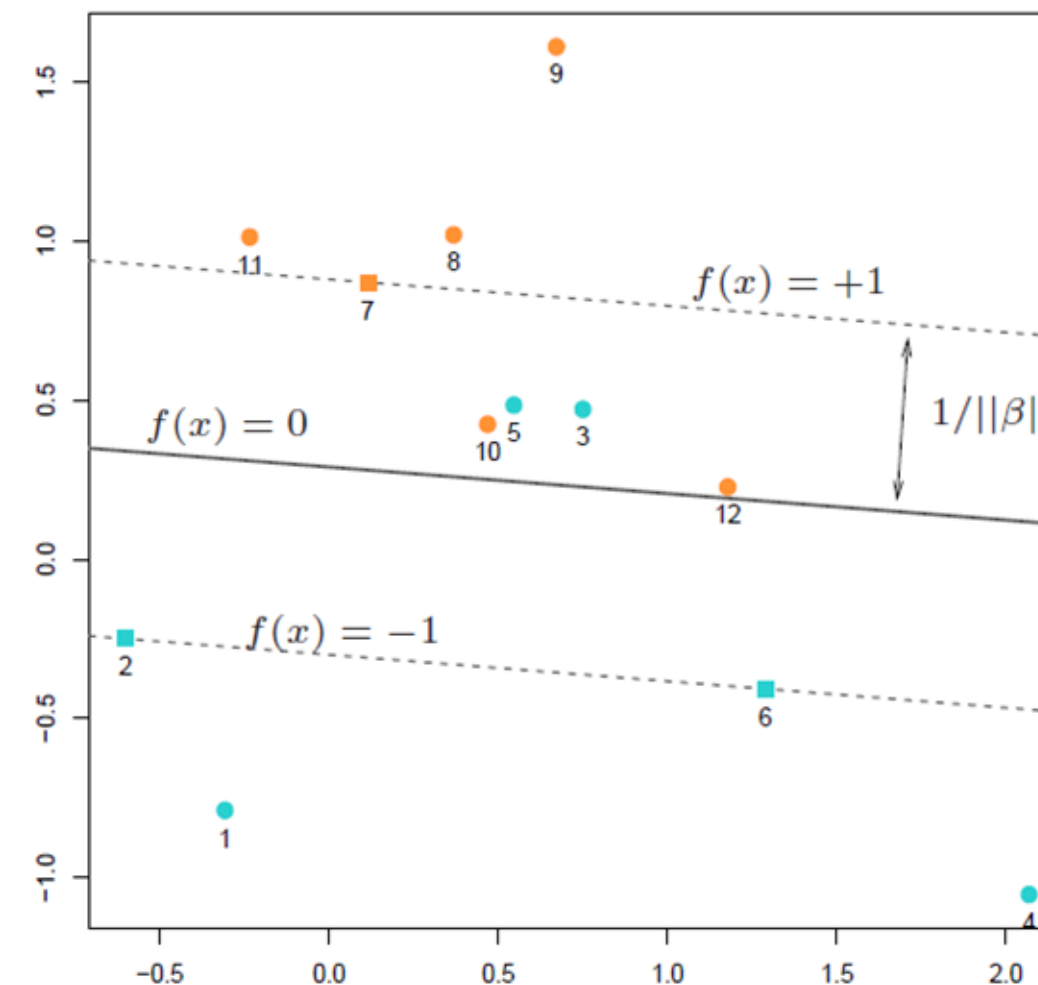
- Observaciones clasificadas correctamente y fuera de sus márgenes. Tienen  $y_i f(x_i) > 1$  y multiplicadores de Lagrange  $\alpha_i = 0$ . Algunos ejemplos son los puntos naranjas 8, 9 y 11, y los puntos azules 1 y 4.
- Observaciones situadas en sus márgenes con  $y_i f(x_i) = 1$ , con multiplicadores de Lagrange  $\alpha_i \in [0, 1]$ . Ejemplos de ello son el 7 naranja y el 2 y el 6 azules.



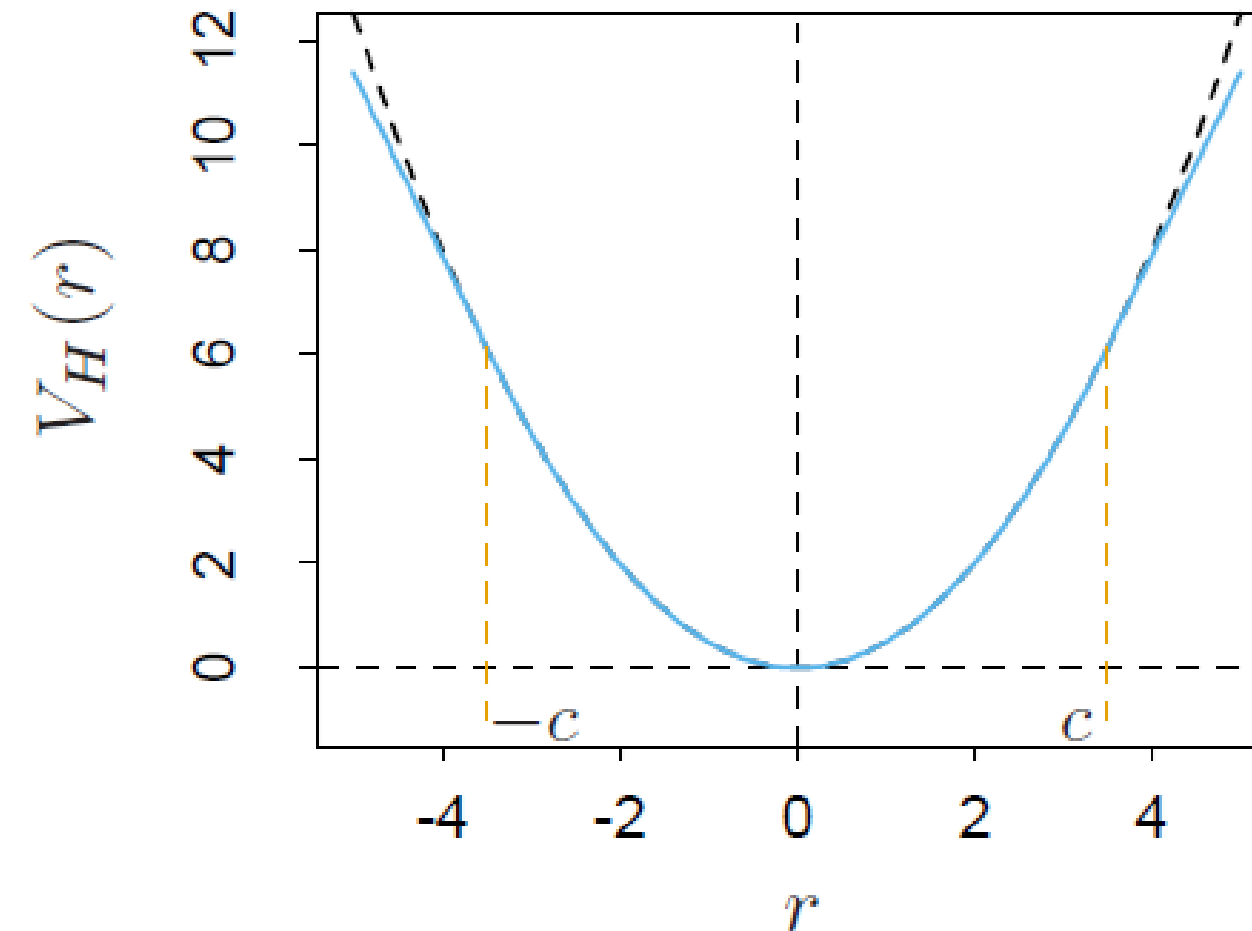
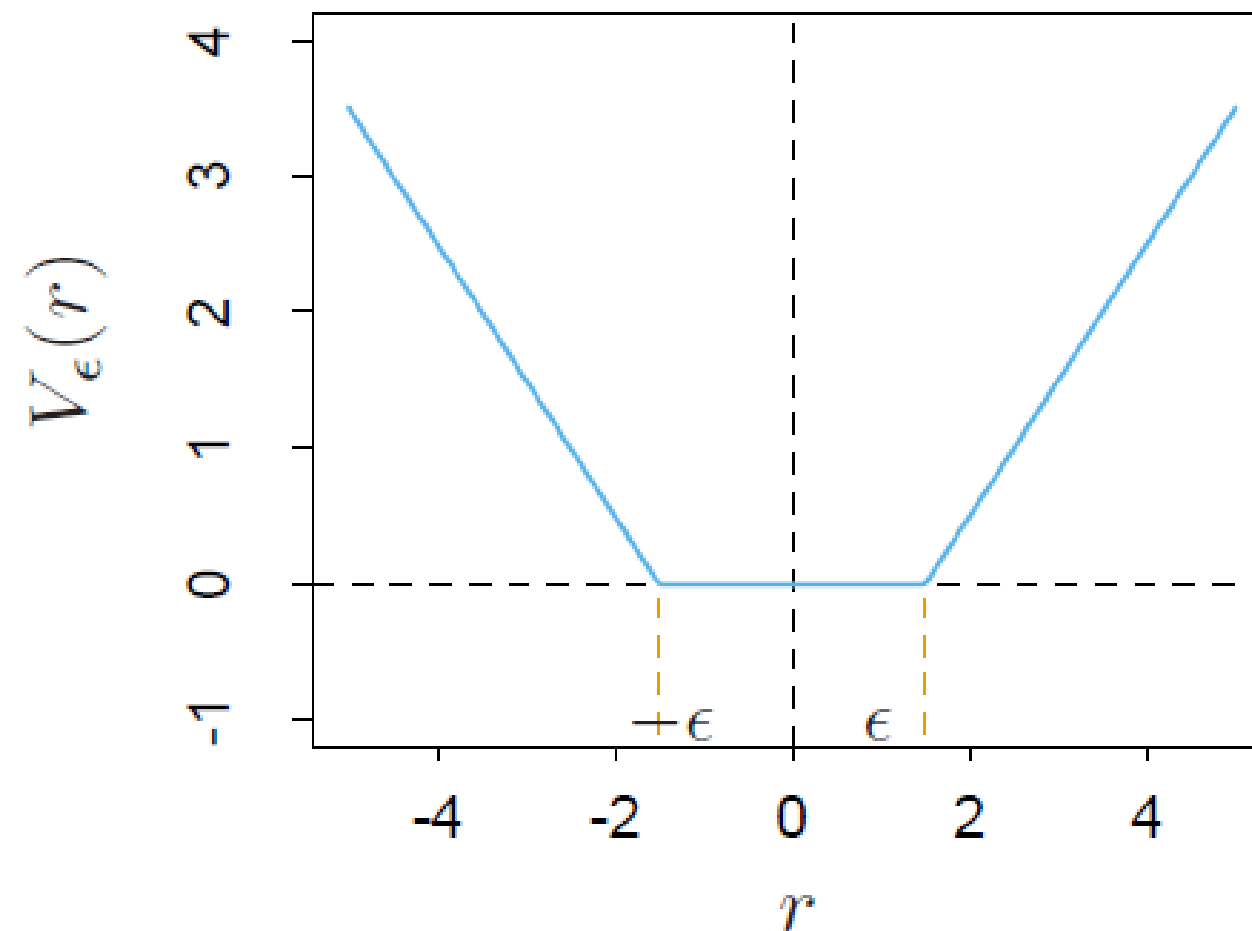
# A Path Algorithm for the SVM Classifier

Las condiciones de optimalidad KKT implican que los puntos etiquetados  $(x_i, y_i)$  se dividen en tres grupos distintos:

- Observaciones clasificadas correctamente y fuera de sus márgenes. Tienen  $y_i f(x_i) > 1$  y multiplicadores de Lagrange  $\alpha_i = 0$ . Algunos ejemplos son los puntos naranjas 8, 9 y 11, y los puntos azules 1 y 4.
- Observaciones situadas en sus márgenes con  $y_i f(x_i) = 1$ , con multiplicadores de Lagrange  $\alpha_i \in [0, 1]$ . Ejemplos de ello son el 7 naranja y el 2 y el 6 azules.
- Las observaciones dentro de sus márgenes tienen  $y_i f(x_i) < 1$ , con  $\alpha_i = 1$ . Ejemplos de ello son el 3 y el 5 azules, y el 10 y el 12 naranjas.



# Support Vector Machines for Regression



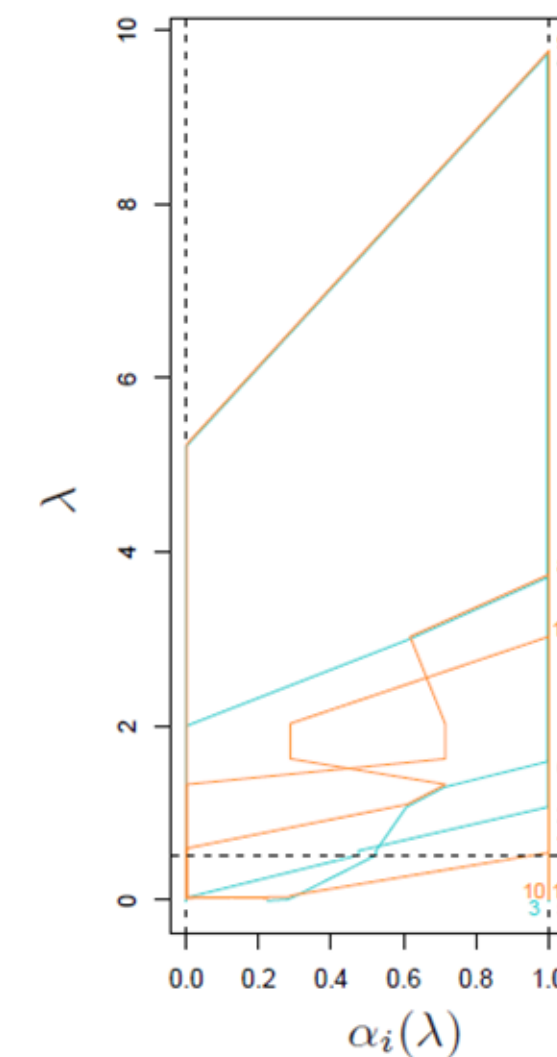
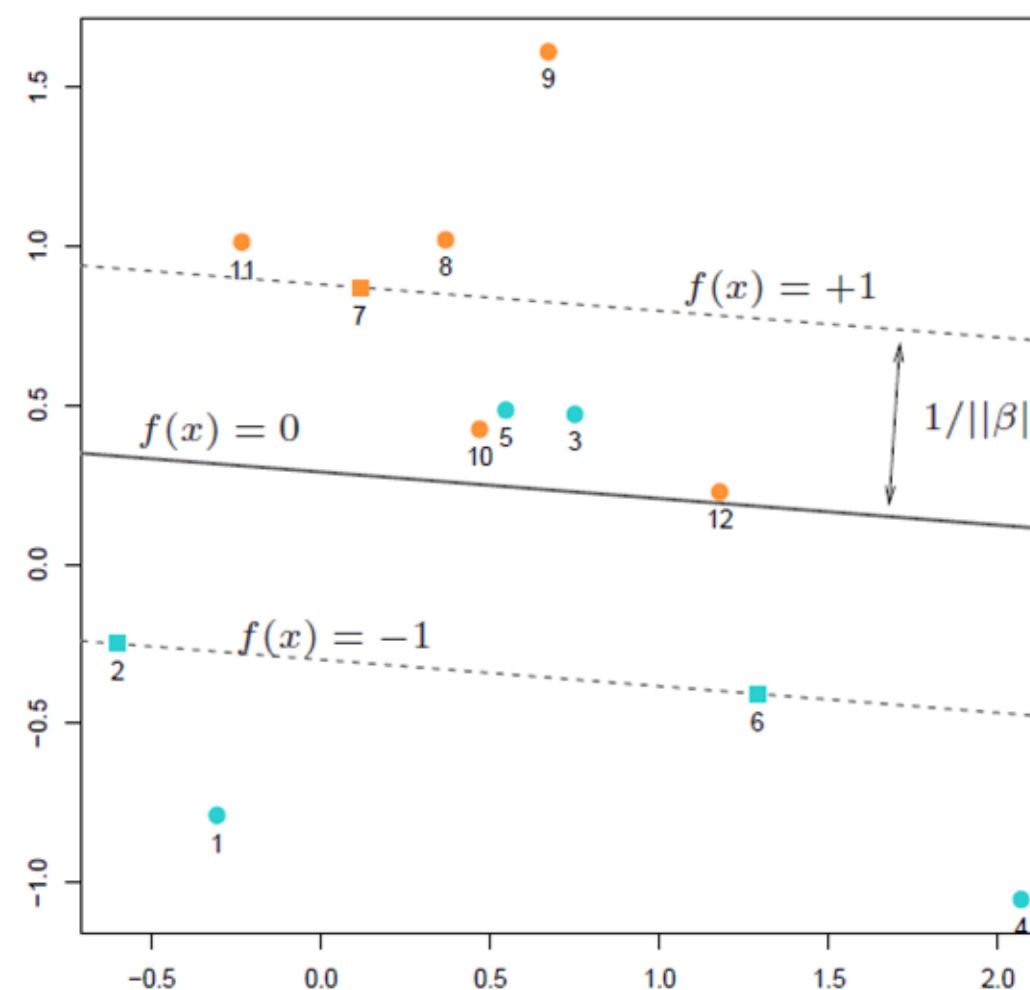
El panel izquierdo muestra la función de error insensible e utilizada por la máquina de regresión de vectores de soporte. El panel derecho muestra la función de error utilizada en la regresión robusta de Huber (curva azul). Más allá de  $|c|$ , la función cambia de cuadrática a lineal.



# Support Vector Machines for Regression

Primero discutimos el modelo de regresión lineal

$$f(x) = x^T \beta + \beta_0,$$



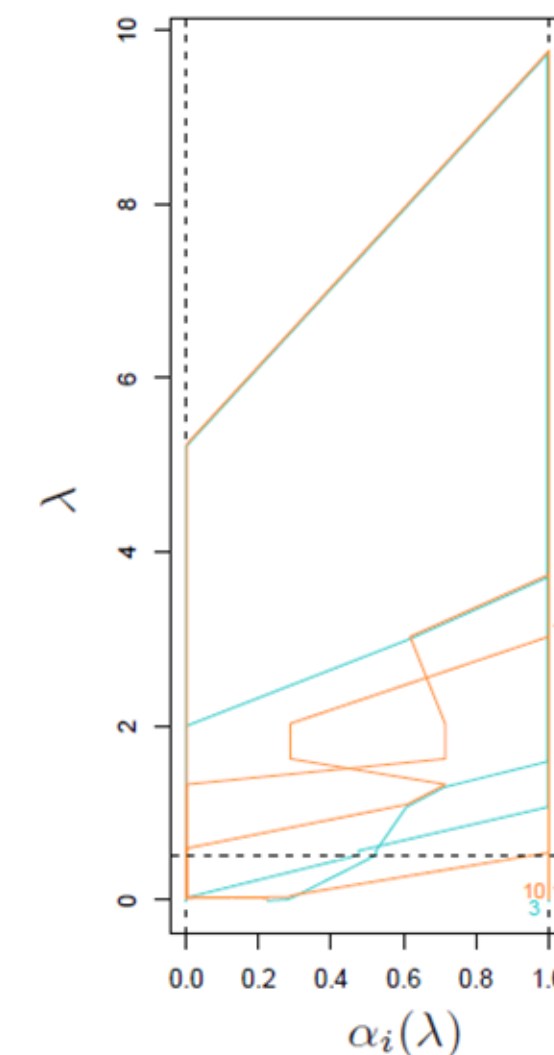
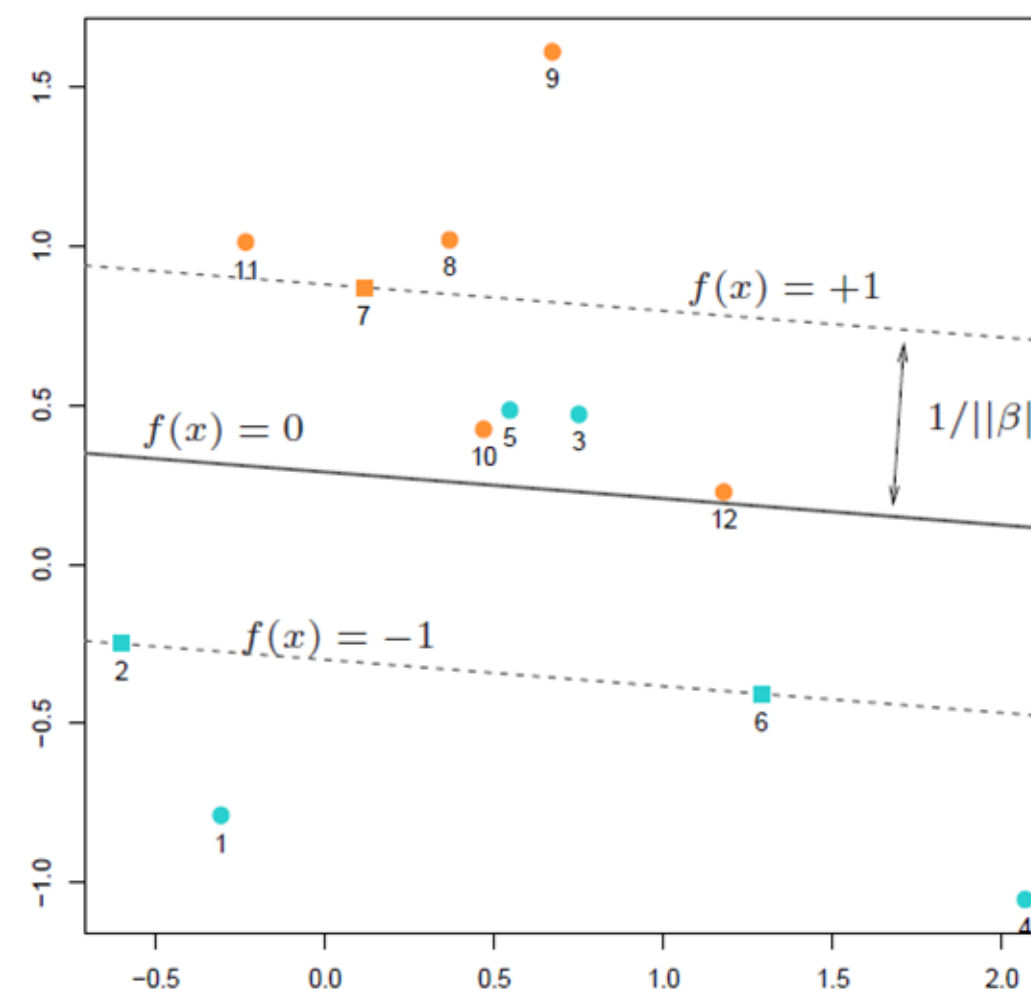
# Support Vector Machines for Regression

Primero discutimos el modelo de regresión lineal

$$f(x) = x^T \beta + \beta_0,$$

y luego tratamos las generalizaciones no lineales. Para estimar  $\beta$ , consideramos la minimización de

$$H(\beta, \beta_0) = \sum_{i=1}^N V(y_i - f(x_i)) + \frac{\lambda}{2} \|\beta\|^2,$$



# Support Vector Machines for Regression

Primero discutimos el modelo de regresión lineal

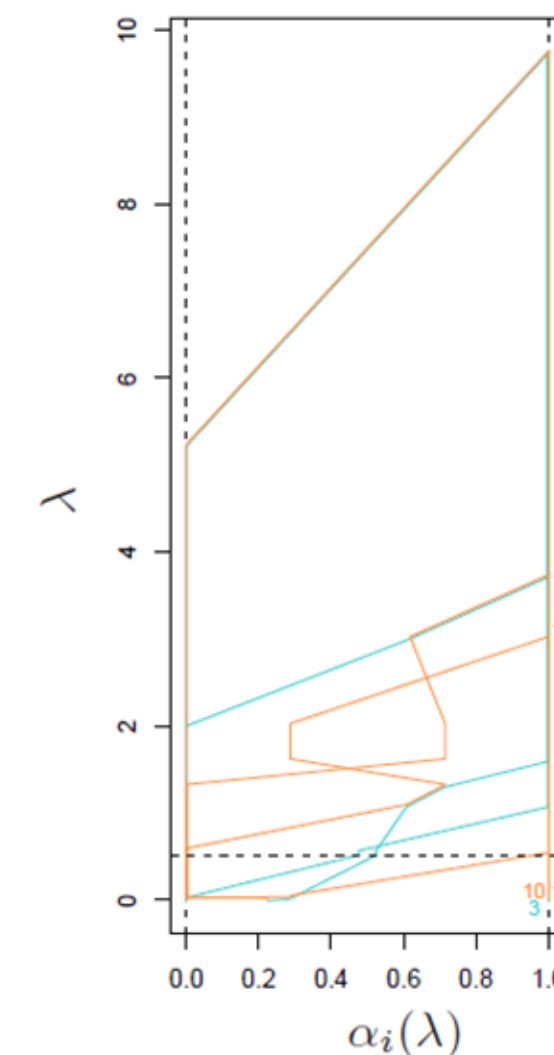
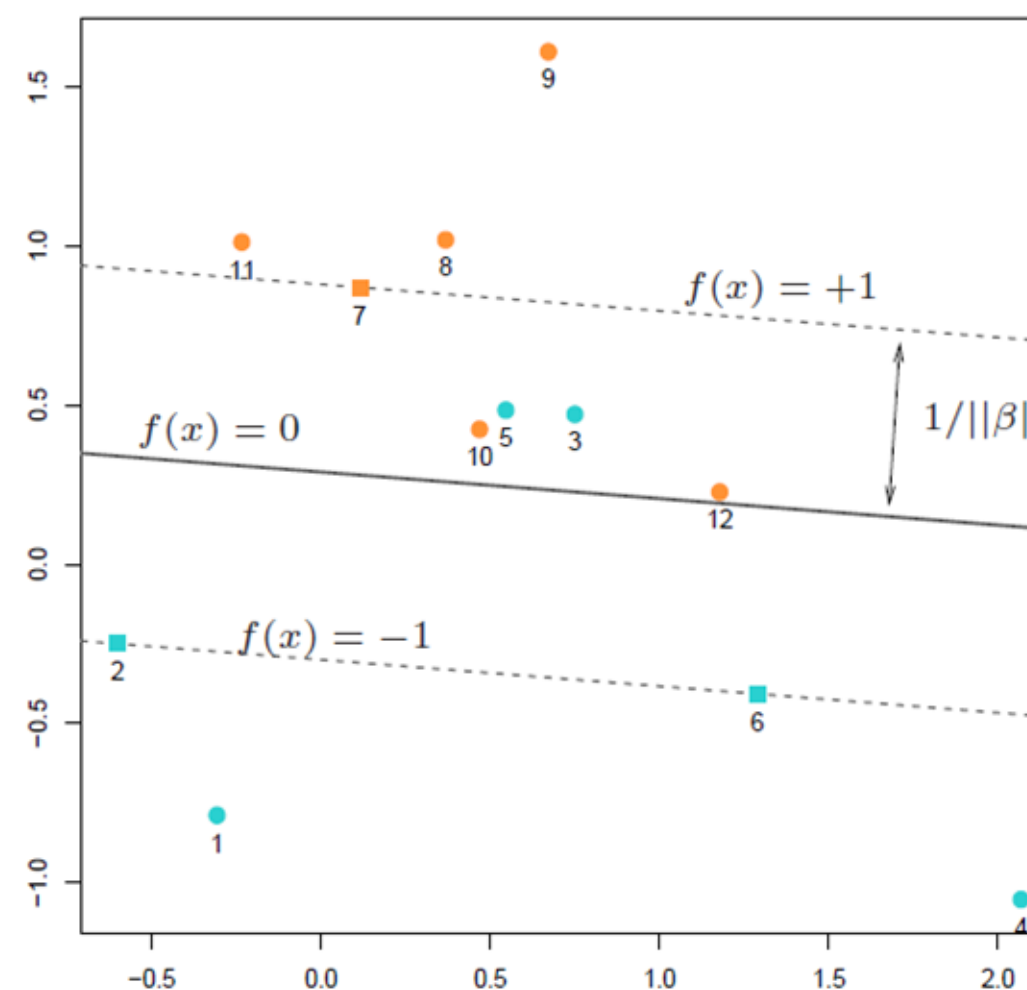
$$f(x) = x^T \beta + \beta_0,$$

y luego tratamos las generalizaciones no lineales. Para estimar  $\beta$ , consideramos la minimización de

$$H(\beta, \beta_0) = \sum_{i=1}^N V(y_i - f(x_i)) + \frac{\lambda}{2} \|\beta\|^2,$$

donde

$$V_{\epsilon}(r) = \begin{cases} 0 & \text{if } |r| < \epsilon, \\ |r| - \epsilon, & \text{otherwise.} \end{cases}$$



# Support Vector Machines for Regression

Primero discutimos el modelo de regresión lineal

$$f(x) = x^T \beta + \beta_0,$$

y luego tratamos las generalizaciones no lineales. Para estimar  $\beta$ , consideramos la minimización de

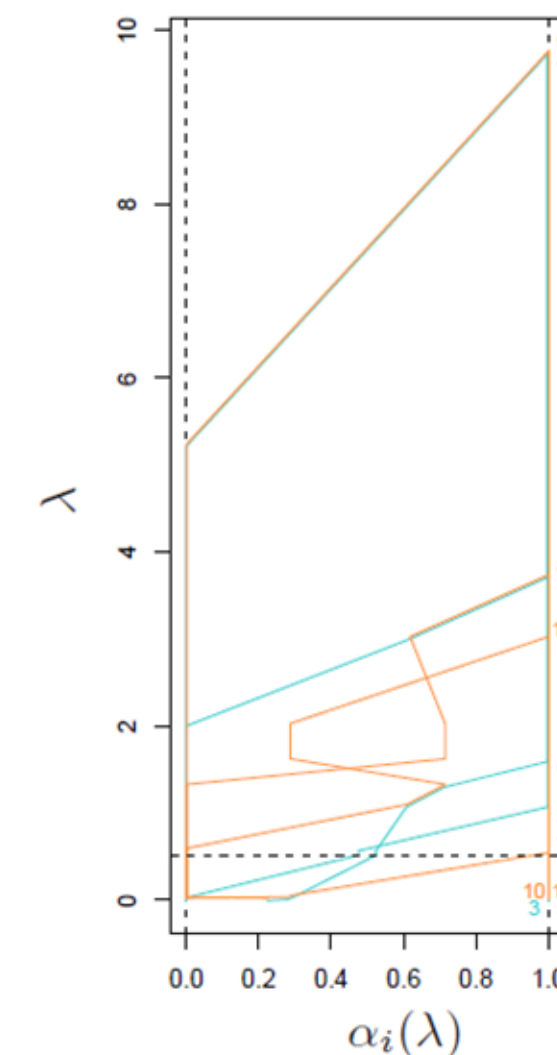
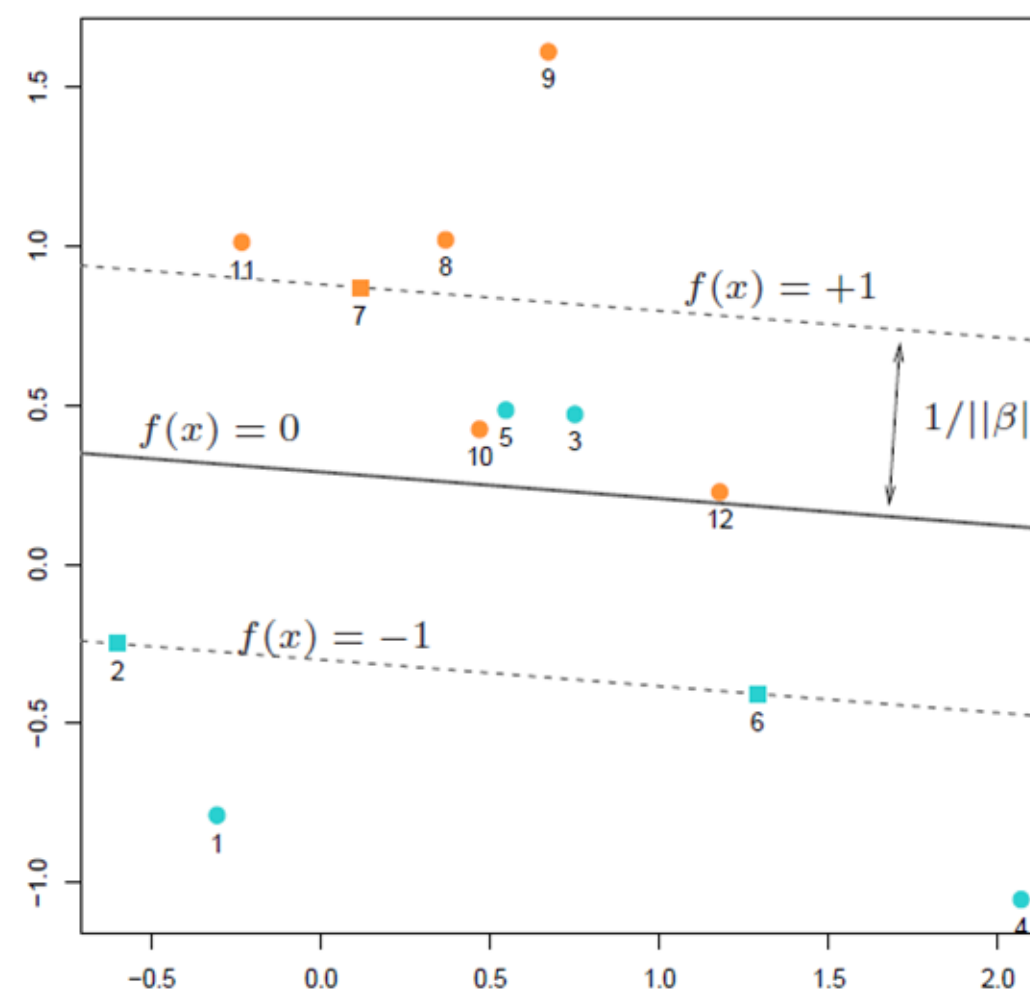
$$H(\beta, \beta_0) = \sum_{i=1}^N V(y_i - f(x_i)) + \frac{\lambda}{2} \|\beta\|^2,$$

donde

$$V_{\epsilon}(r) = \begin{cases} 0 & \text{if } |r| < \epsilon, \\ |r| - \epsilon, & \text{otherwise.} \end{cases}$$

El más popular, debido a Huber (1964), tiene la forma

$$V_H(r) = \begin{cases} r^2/2 & \text{if } |r| \leq c, \\ c|r| - c^2/2, & |r| > c, \end{cases}$$



# Support Vector Machines for Regression

Si  $\hat{\beta}$ ,  $\hat{\beta}_0$  son los minimizadores de  $H$ , se puede demostrar que la función de solución tiene la forma

$$\begin{aligned}\hat{\beta} &= \sum_{i=1}^N (\hat{\alpha}_i^* - \hat{\alpha}_i) x_i, \\ \hat{f}(x) &= \sum_{i=1}^N (\hat{\alpha}_i^* - \hat{\alpha}_i) \langle x, x_i \rangle + \beta_0,\end{aligned}$$



# Support Vector Machines for Regression

Si  $\hat{\beta}$ ,  $\hat{\beta}_0$  son los minimizadores de  $H$ , se puede demostrar que la función de solución tiene la forma

$$\begin{aligned}\hat{\beta} &= \sum_{i=1}^N (\hat{\alpha}_i^* - \hat{\alpha}_i) x_i, \\ \hat{f}(x) &= \sum_{i=1}^N (\hat{\alpha}_i^* - \hat{\alpha}_i) \langle x, x_i \rangle + \beta_0,\end{aligned}$$

donde  $\hat{\alpha}_i$ ,  $\hat{\alpha}_i^*$  son positivos y resuelven el problema de programación cuadrática

$$\min_{\alpha_i, \alpha_i^*} \epsilon \sum_{i=1}^N (\alpha_i^* + \alpha_i) - \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i) + \frac{1}{2} \sum_{i, i'=1}^N (\alpha_i^* - \alpha_i) (\alpha_{i'}^* - \alpha_{i'}) \langle x_i, x_{i'} \rangle$$



# Support Vector Machines for Regression

Si  $\hat{\beta}$ ,  $\hat{\beta}_0$  son los minimizadores de  $H$ , se puede demostrar que la función de solución tiene la forma

$$\begin{aligned}\hat{\beta} &= \sum_{i=1}^N (\hat{\alpha}_i^* - \hat{\alpha}_i) x_i, \\ \hat{f}(x) &= \sum_{i=1}^N (\hat{\alpha}_i^* - \hat{\alpha}_i) \langle x, x_i \rangle + \beta_0,\end{aligned}$$

donde  $\hat{\alpha}_i$ ,  $\hat{\alpha}_i^*$  son positivos y resuelven el problema de programación cuadrática

$$\min_{\alpha_i, \alpha_i^*} \epsilon \sum_{i=1}^N (\alpha_i^* + \alpha_i) - \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i) + \frac{1}{2} \sum_{i, i'=1}^N (\alpha_i^* - \alpha_i) (\alpha_{i'}^* - \alpha_{i'}) \langle x_i, x_{i'} \rangle$$

sujeto a las restricciones

$$\begin{aligned}0 &\leq \alpha_i, \alpha_i^* \leq 1/\lambda, \\ \sum_{i=1}^N (\alpha_i^* - \alpha_i) &= 0, \\ \alpha_i \alpha_i^* &= 0.\end{aligned}$$



# Support Vector Machines for Regression

Si  $\hat{\beta}$ ,  $\hat{\beta}_0$  son los minimizadores de  $H$ , se puede demostrar que la función de solución tiene la forma

$$\begin{aligned}\hat{\beta} &= \sum_{i=1}^N (\hat{\alpha}_i^* - \hat{\alpha}_i) x_i, \\ \hat{f}(x) &= \sum_{i=1}^N (\hat{\alpha}_i^* - \hat{\alpha}_i) \langle x, x_i \rangle + \beta_0,\end{aligned}$$

donde  $\hat{\alpha}_i$ ,  $\hat{\alpha}_i^*$  son positivos y resuelven el problema de programación cuadrática

$$\min_{\alpha_i, \alpha_i^*} \epsilon \sum_{i=1}^N (\alpha_i^* + \alpha_i) - \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i) + \frac{1}{2} \sum_{i, i'=1}^N (\alpha_i^* - \alpha_i) (\alpha_{i'}^* - \alpha_{i'}) \langle x_i, x_{i'} \rangle$$

sujeto a las restricciones

$$\begin{aligned}0 &\leq \alpha_i, \alpha_i^* \leq 1/\lambda, \\ \sum_{i=1}^N (\alpha_i^* - \alpha_i) &= 0, \\ \alpha_i \alpha_i^* &= 0.\end{aligned}$$

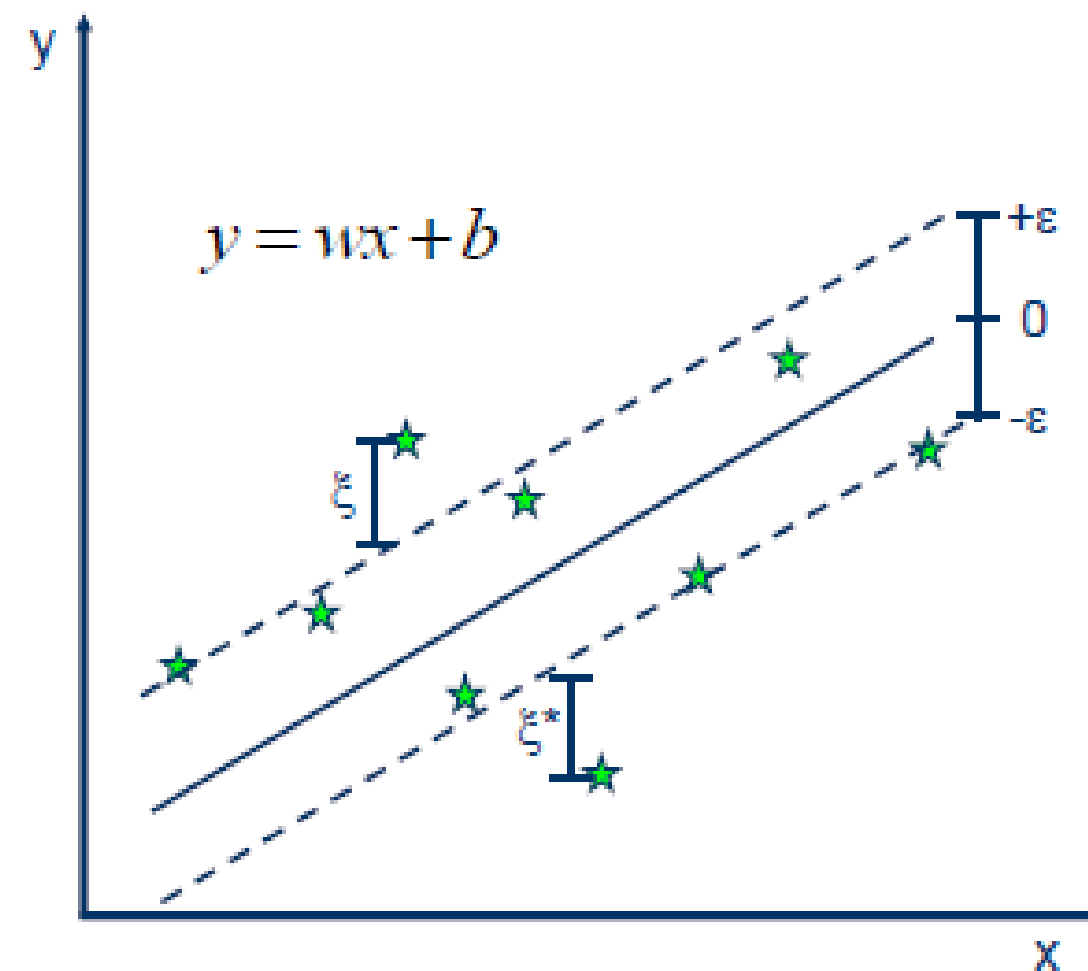
Debido a la naturaleza de estas restricciones, normalmente sólo un subconjunto de los valores de la solución ( $\hat{\alpha}_i^* - \hat{\alpha}_i$ ) son distintos de cero, y los valores de datos asociados se denominan vectores de soporte.



# Support Vector Machines for Regression

Primero discutimos el modelo de regresión lineal

$$f(x) = x^T \beta + \beta_0,$$



• Minimize:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$

• Constraints:

$$y_i - wx_i - b \leq \epsilon + \xi_i$$

$$wx_i + b - y_i \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

[http://www.saedsayad.com/support\\_vector\\_machine\\_reg.htm](http://www.saedsayad.com/support_vector_machine_reg.htm)



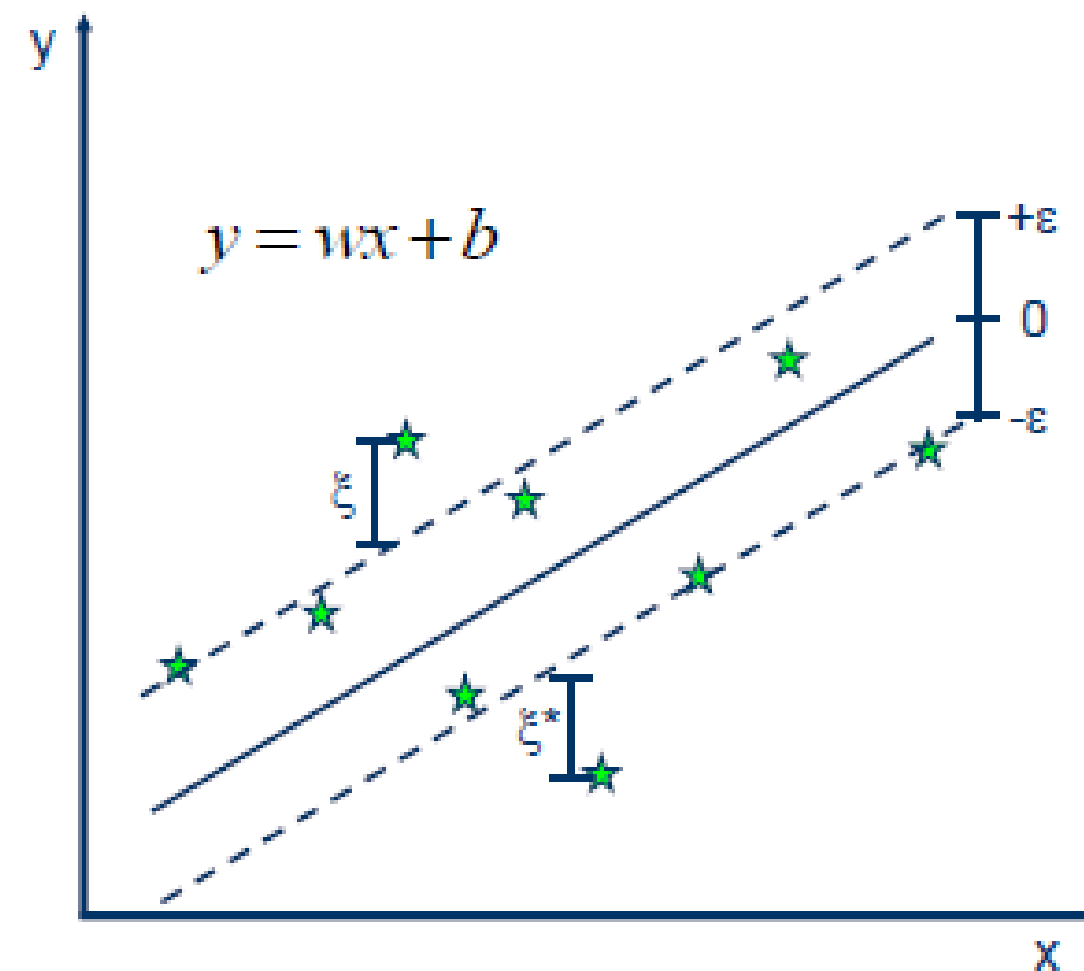
# Support Vector Machines for Regression

Primero discutimos el modelo de regresión lineal

$$f(x) = x^T \beta + \beta_0,$$

y luego tratamos las generalizaciones no lineales. Para estimar  $\beta$ , consideramos la minimización de

$$H(\beta, \beta_0) = \sum_{i=1}^N V(y_i - f(x_i)) + \frac{\lambda}{2} \|\beta\|^2,$$



• Minimize:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$

• Constraints:

$$y_i - wx_i - b \leq \varepsilon + \xi_i$$

$$wx_i + b - y_i \leq \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

[http://www.saedsayad.com/support\\_vector\\_machine\\_reg.htm](http://www.saedsayad.com/support_vector_machine_reg.htm)



# Support Vector Machines for Regression

Primero discutimos el modelo de regresión lineal

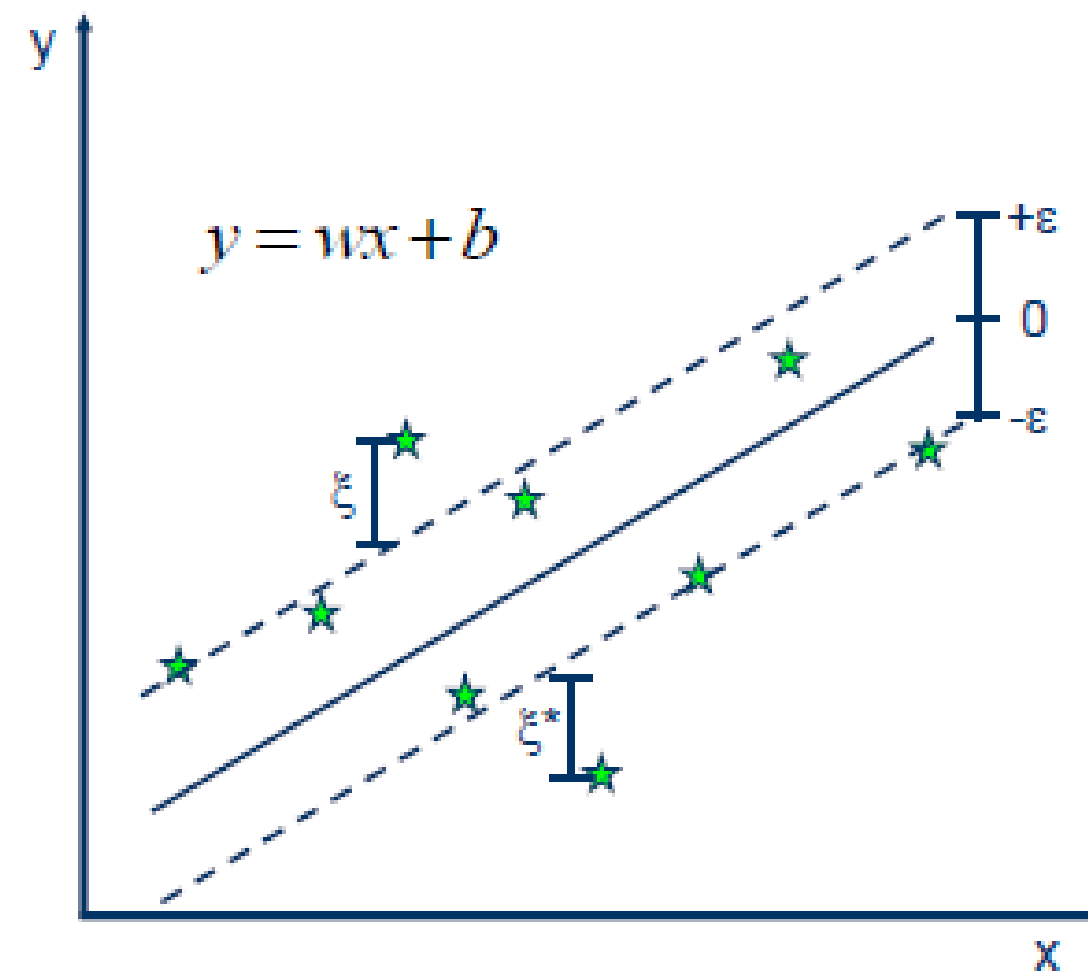
$$f(x) = x^T \beta + \beta_0,$$

y luego tratamos las generalizaciones no lineales. Para estimar  $\beta$ , consideramos la minimización de

$$H(\beta, \beta_0) = \sum_{i=1}^N V(y_i - f(x_i)) + \frac{\lambda}{2} \|\beta\|^2,$$

donde

$$V_{\epsilon}(r) = \begin{cases} 0 & \text{if } |r| < \epsilon, \\ |r| - \epsilon, & \text{otherwise.} \end{cases}$$



• Minimize:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$

• Constraints:

$$y_i - wx_i - b \leq \epsilon + \xi_i$$

$$wx_i + b - y_i \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

[http://www.saedsayad.com/support\\_vector\\_machine\\_reg.htm](http://www.saedsayad.com/support_vector_machine_reg.htm)



# Support Vector Machines for Regression

Primero discutimos el modelo de regresión lineal

$$f(x) = x^T \beta + \beta_0,$$

y luego tratamos las generalizaciones no lineales. Para estimar  $\beta$ , consideramos la minimización de

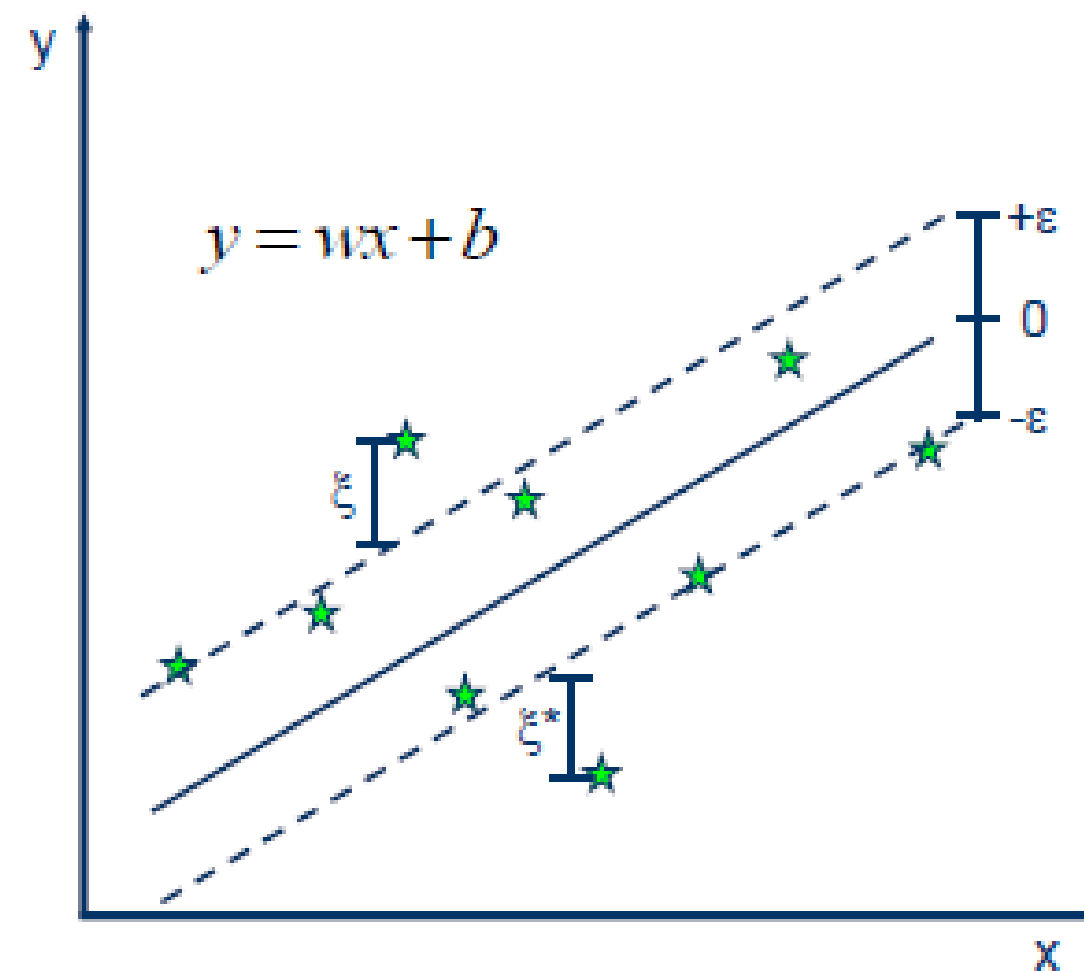
$$H(\beta, \beta_0) = \sum_{i=1}^N V(y_i - f(x_i)) + \frac{\lambda}{2} \|\beta\|^2,$$

donde

$$V_{\epsilon}(r) = \begin{cases} 0 & \text{if } |r| < \epsilon, \\ |r| - \epsilon, & \text{otherwise.} \end{cases}$$

El más popular, debido a Huber (1964), tiene la forma

$$V_H(r) = \begin{cases} r^2/2 & \text{if } |r| \leq c, \\ c|r| - c^2/2, & |r| > c, \end{cases}$$



• Minimize:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$

• Constraints:

$$y_i - wx_i - b \leq \epsilon + \xi_i$$

$$wx_i + b - y_i \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

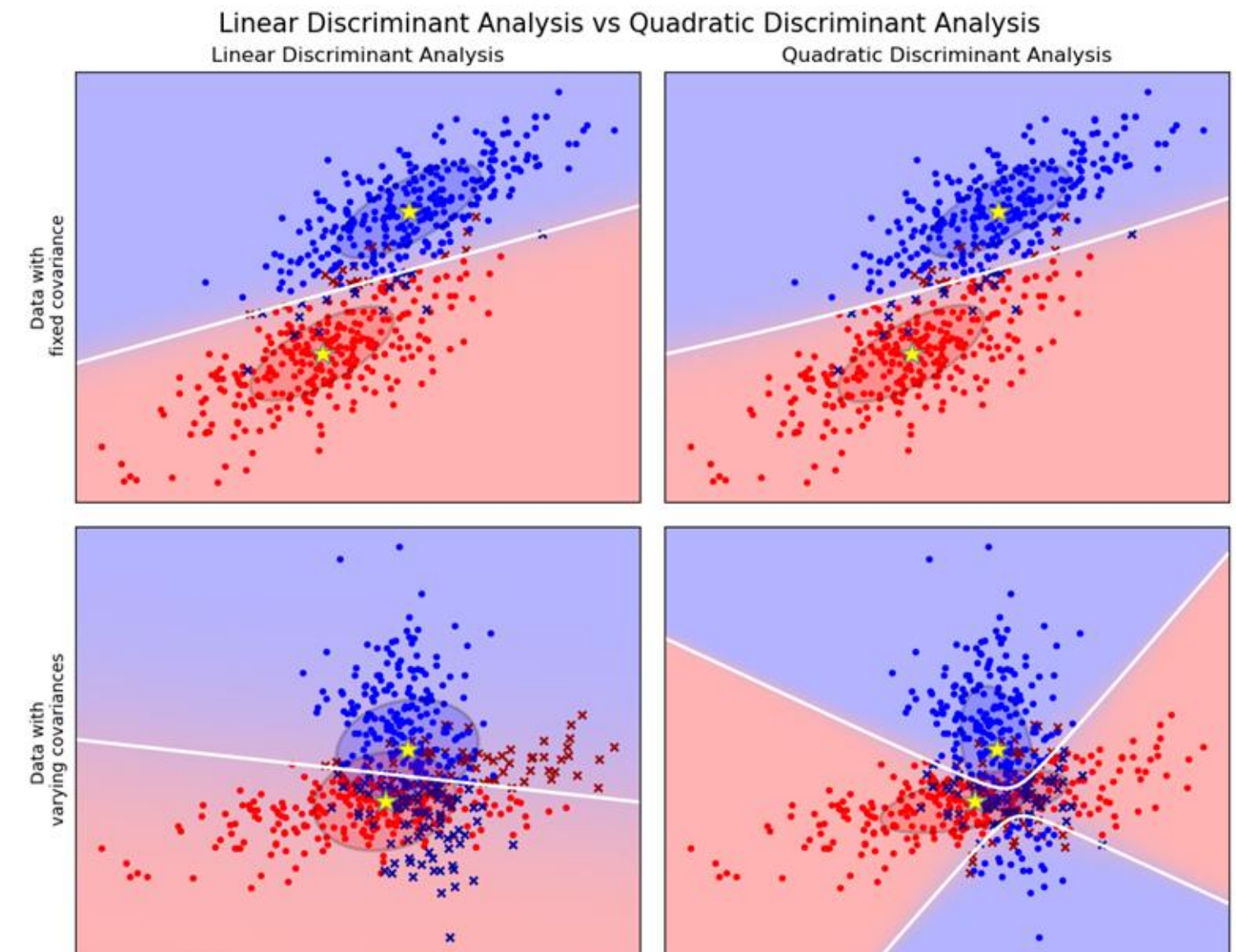
[http://www.saedsayad.com/support\\_vector\\_machine\\_reg.htm](http://www.saedsayad.com/support_vector_machine_reg.htm)



# Generalizing Linear Discriminant Analysis

Algunas de las ventajas del LDA son las siguientes:

- Es un clasificador prototipo sencillo. Una nueva observación se clasifica en la clase con el centroide más cercano. Una ligera variante es que la distancia se mide en la métrica de Mahalanobis, utilizando una estimación de covarianza agrupada.



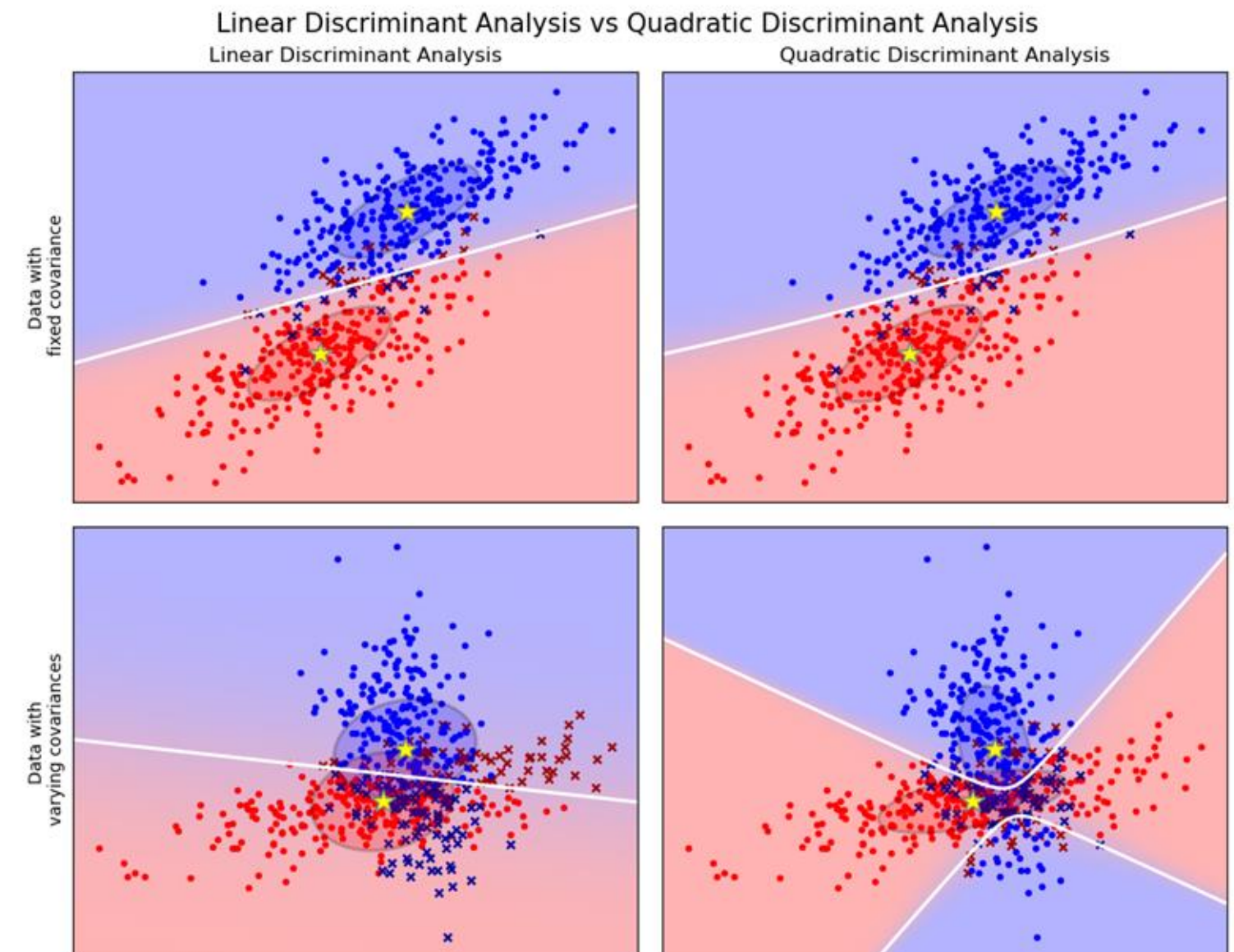
<https://ml-explained.com/blog/linear-discriminant-analysis-explained>



# Generalizing Linear Discriminant Analysis

Algunas de las ventajas del LDA son las siguientes:

- Es un clasificador prototipo sencillo. Una nueva observación se clasifica en la clase con el centroide más cercano. Una ligera variante es que la distancia se mide en la métrica de Mahalanobis, utilizando una estimación de covarianza agrupada.
- El LDA es el clasificador bayesiano estimado si las observaciones son gaussianas multivariantes en cada clase, con una matriz de covarianza común. Dado que es poco probable que esta suposición sea cierta, puede que esto no parezca una gran ventaja.



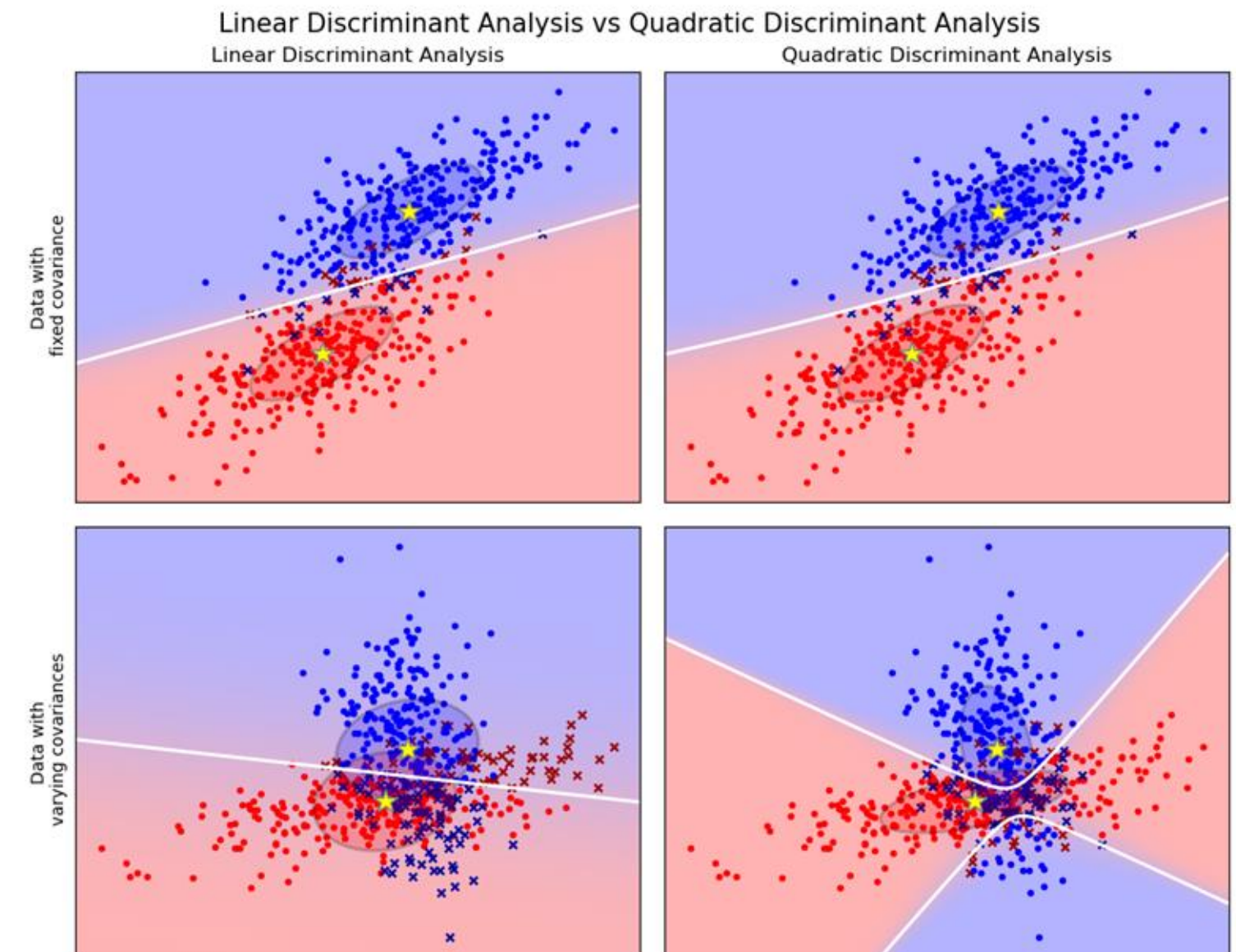
<https://ml-explained.com/blog/linear-discriminant-analysis-explained>



# Generalizing Linear Discriminant Analysis

Algunas de las ventajas del LDA son las siguientes:

- Es un clasificador prototipo sencillo. Una nueva observación se clasifica en la clase con el centroide más cercano. Una ligera variante es que la distancia se mide en la métrica de Mahalanobis, utilizando una estimación de covarianza agrupada.
- El LDA es el clasificador bayesiano estimado si las observaciones son gaussianas multivariantes en cada clase, con una matriz de covarianza común. Dado que es poco probable que esta suposición sea cierta, puede que esto no parezca una gran ventaja.
- Los límites de decisión creados por el LDA son lineales, lo que da lugar a reglas de decisión fáciles de describir y aplicar.



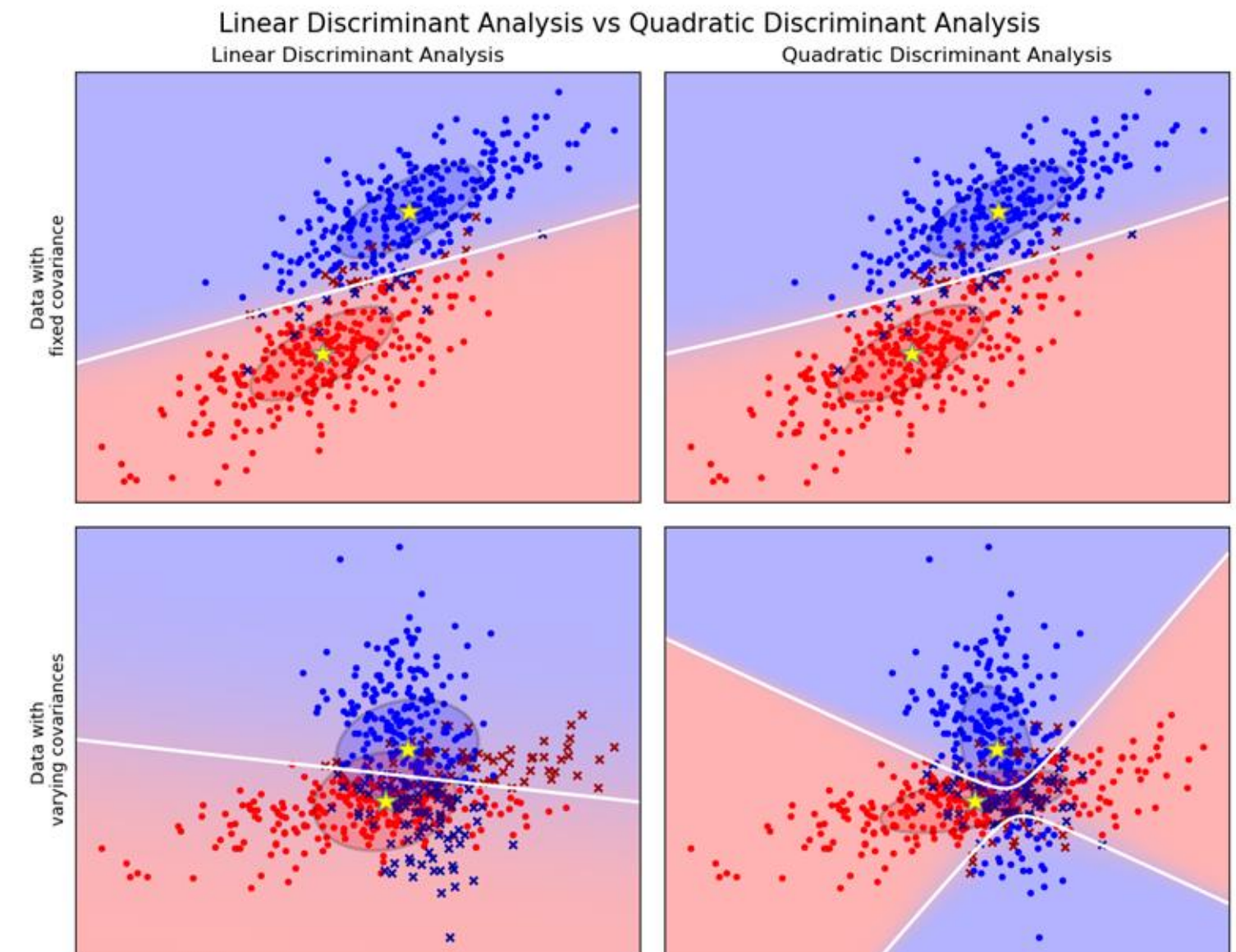
<https://ml-explained.com/blog/linear-discriminant-analysis-explained>



# Generalizing Linear Discriminant Analysis

Algunas de las ventajas del LDA son las siguientes:

- El LDA proporciona vistas naturales de baja dimensión de los datos. Por ejemplo, la figura 12.12 es una vista bidimensional informativa de datos en 256 dimensiones con diez clases.
- A menudo, el LDA produce los mejores resultados de clasificación, debido a su simplicidad y baja varianza. El LDA se encontraba entre los tres mejores clasificadores para 7 de los 22 conjuntos de datos estudiados en el proyecto STATLOG (Michie et al., 1994).



<https://ml-explained.com/blog/linear-discriminant-analysis-explained>





**UTEC** Posgrado

