MÁQUINAS DE VECTORES DE SOPORTE

Introducción

- Se tiene información de N individuos codificada de la forma $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$

- Las variables X son vectores que reúnen información numérica del individuo, las variables Y indican una de dos clases a las que pertenece.
- Asumiremos que $x_i \in \mathbb{R}^p$ y que $y_i \in \{-1, 1\}$
- Se quiere encontrar una relación entre las variables X y la clase Y. Más que esto, se quiere diseñar una regla capaz de predecir la clase de un individuo que no esté en la base, partiendo únicamente de X (sus covariantes).

Introducción

Aplicaciones

- Riesgo Crediticio
- Diagnosis Médico
- Trading Algorítmico
- Reconocimiento de dígitos

Introducción

- Defina un hiperplano como

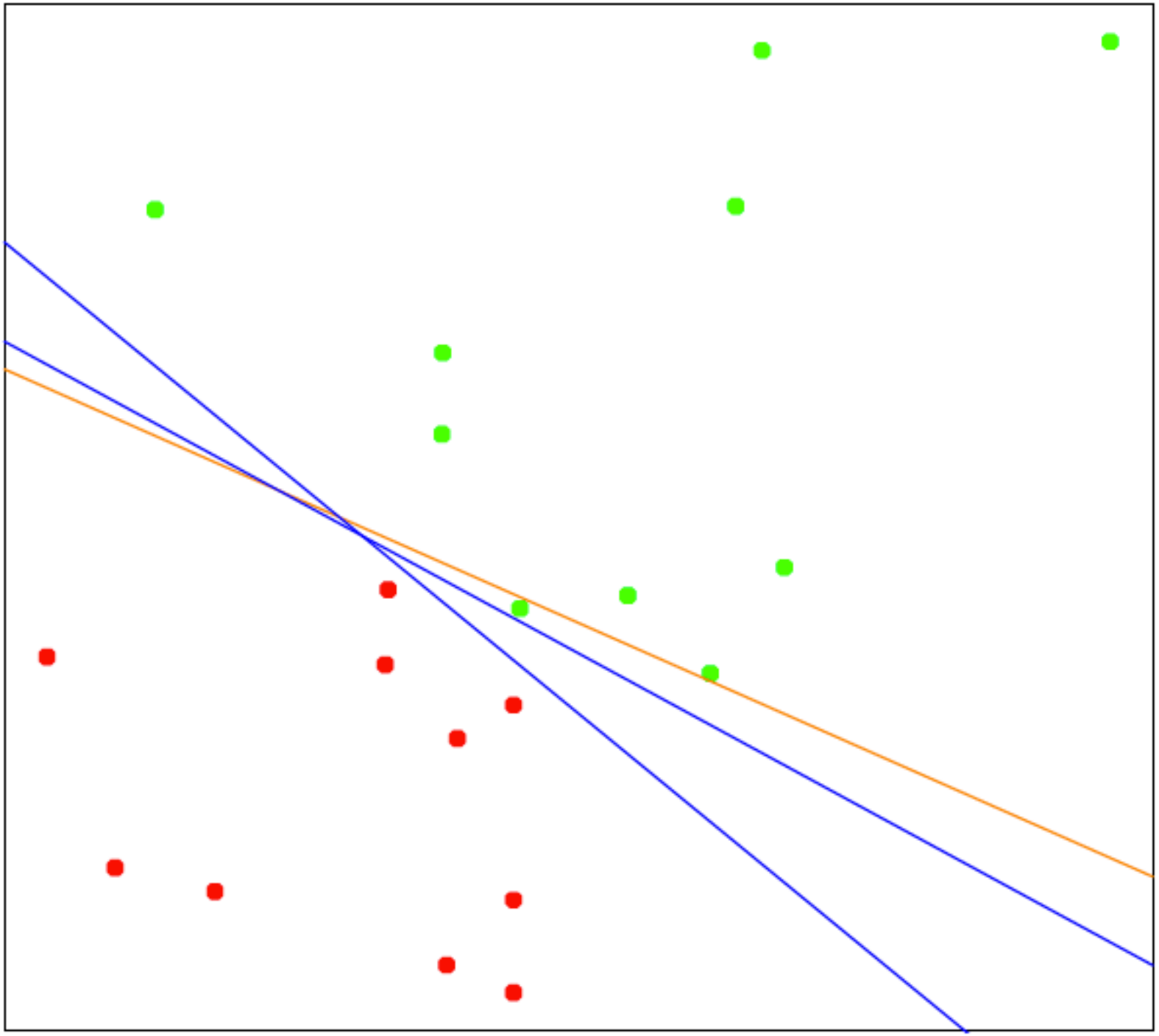
$$\{x : \ f(x) = x^T\beta + \beta_0 = 0\}$$

donde $\|\beta\| = 1$

- Una regla de clasificación podría ser

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

- Esta regla clasifica un dato como 1 o como -1 dependiendo de si está a la derecha o a la izquierda del hiperplano dibujado en el espacio de los covariantes.



Hiperplanos Separadores Óptimos

- Si los datos de las dos clases son separables linealmente por un hiperplano como el de la diapositiva 4, entonces existe un M positivo tal que

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

- Sin embargo, si tenemos un hiperplano separador, muy seguramente existen infinitos. Se busca el hiperplano óptimo. Una formulación de este problema podría ser

$$\max_{\beta,\beta_0,||\beta||=1} M$$

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

Hiperplanos Separadores Óptimos

$$\max_{\beta,\beta_0,||\beta||=1} M$$

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

La condición $\|\beta\| = 1$ se pude remover reemplazándola por

$$\frac{1}{||\beta||} y_i(x_i^T\beta + \beta_0) \geq M$$

Es decir

$$y_i(x_i^T\beta + \beta_0) \geq M||\beta||$$

Hiperplanos Separadores Óptimos

$$y_i(x_i^T\beta + \beta_0) \geq M||\beta||.$$

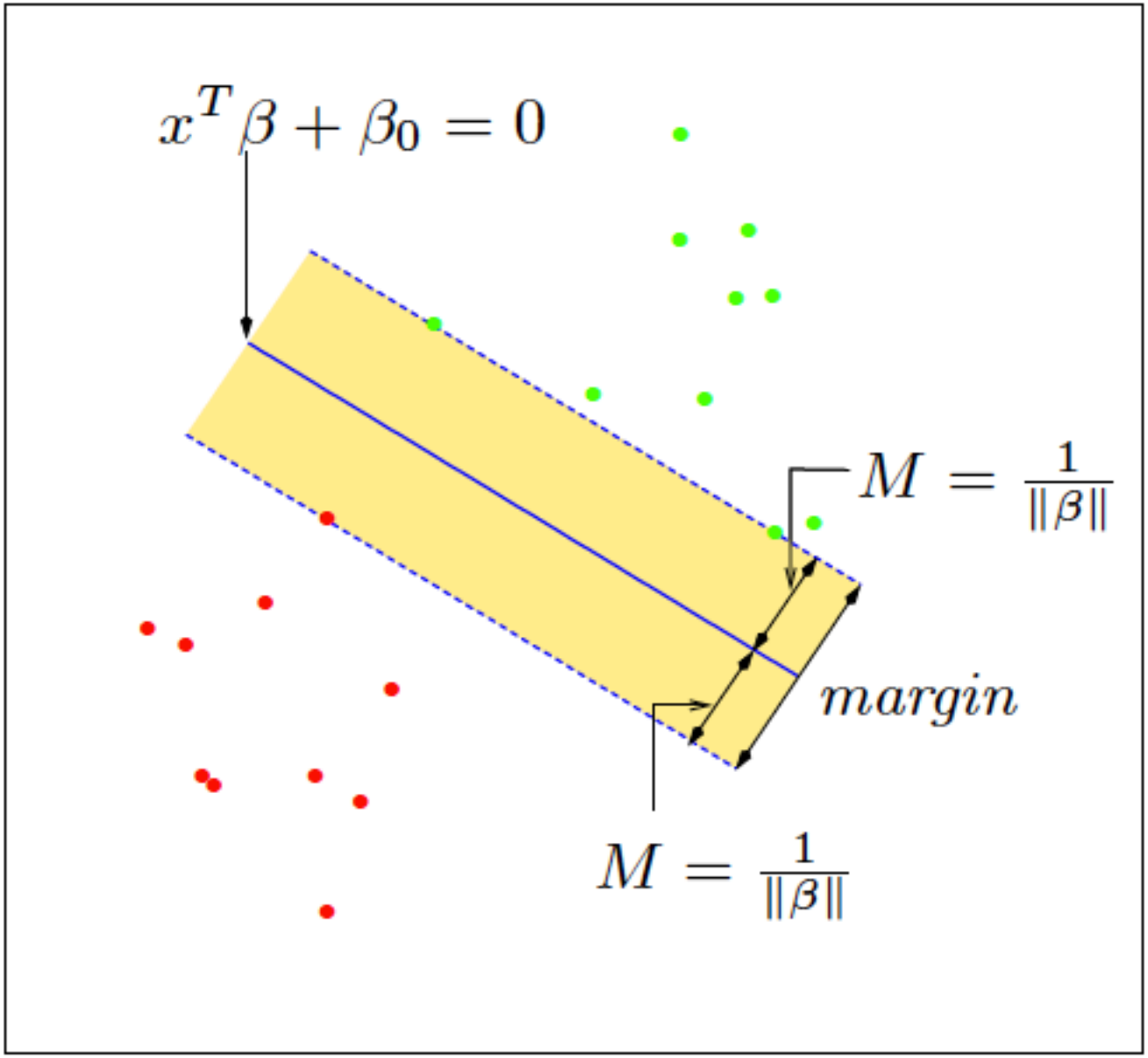
Aquí podemos asumir, sin pérdida de generalidad, que

$$||\beta|| = 1/M$$

Con lo que el problema se convierte en

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

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


$x^T\beta + \beta_0 = 0$
$M = \frac{1}{\|\beta\|}$
$margin$
$M = \frac{1}{\|\beta\|}$

Hiperplanos Separadores Óptimos

El Lagrangiano del problema está dado por

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

Al derivar respecto a β obtenemos

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

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

Reemplazando en el Lagrangiano, obtenemos el problema dual.

Hiperplanos Separadores Óptimos

- Cualquier problema de optimización tiene un problema dual. Para encontrarlo, se contruye el Lagrangiano y se minimiza para obtener la función Dual.
- El problema dual es la maximización de la función dual en el ortante positivo

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

$$\text{subject to} \qquad \alpha_i \geq 0$$

Hiperplanos Separadores Óptimos

- El problema dual es uno de optimización convexa y puede ser resuelto en tiempo polinomial.
- Es un problema mucho más sencillo que el problema primal
- La solución genera un hiperplano con el que se puede clasificar

$$\hat{f}(x) = x^T \hat{\beta} + \hat{\beta}_0$$

$$\hat{G}(x) = \text{sign}\hat{f}(x)$$

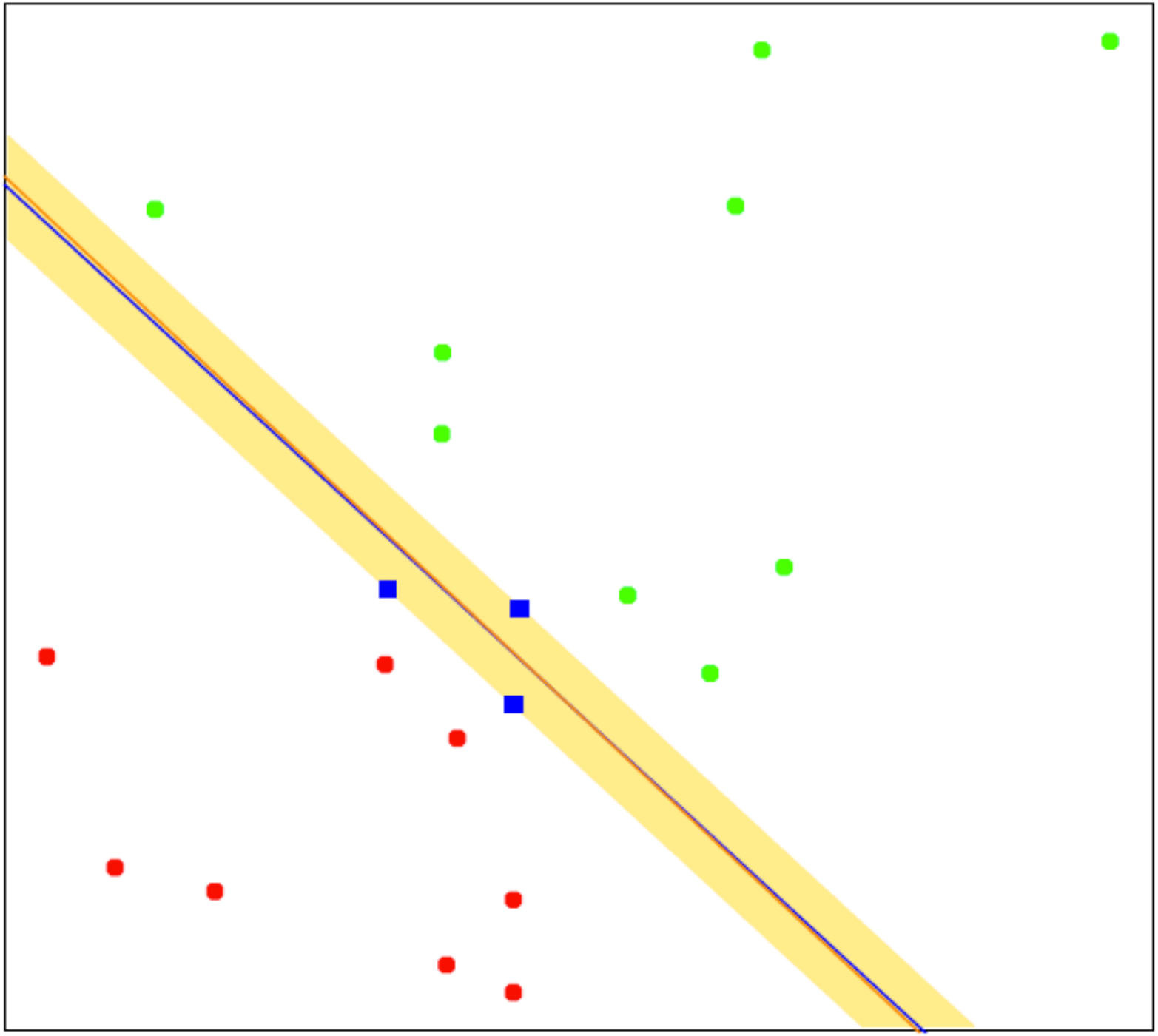
Hiperplanos Separadores Óptimos

La solución satisface las condiciones de Kuhn-Tucker.

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

Explicación

- Si $\alpha_i > 0$ entonces $y_i(x_i^T\beta + \beta_0) = 1$
- Esto quiere decir que el dato está en todo el margen. A estos datos se les llama vectores de soporte.
- Si $y_i(x_i^T\beta + \beta_0) > 1$ el dato no está en el margen y la restricción no es necesaria.



Datos No Separables

- Si los datos no son separables el problema de optimización es infactible.
- Para resolver este problema se definen variables de holgura $\xi = (\xi_1, \xi_2, \ldots, \xi_N)$ que permiten violar las restricciones.
- Las nuevas restricciones son de la forma

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

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

Datos No Separables

El problema se puede reformular de la siguiente forma

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

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

De nuevo se calcula el Lagrangiano

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

Datos No Separables

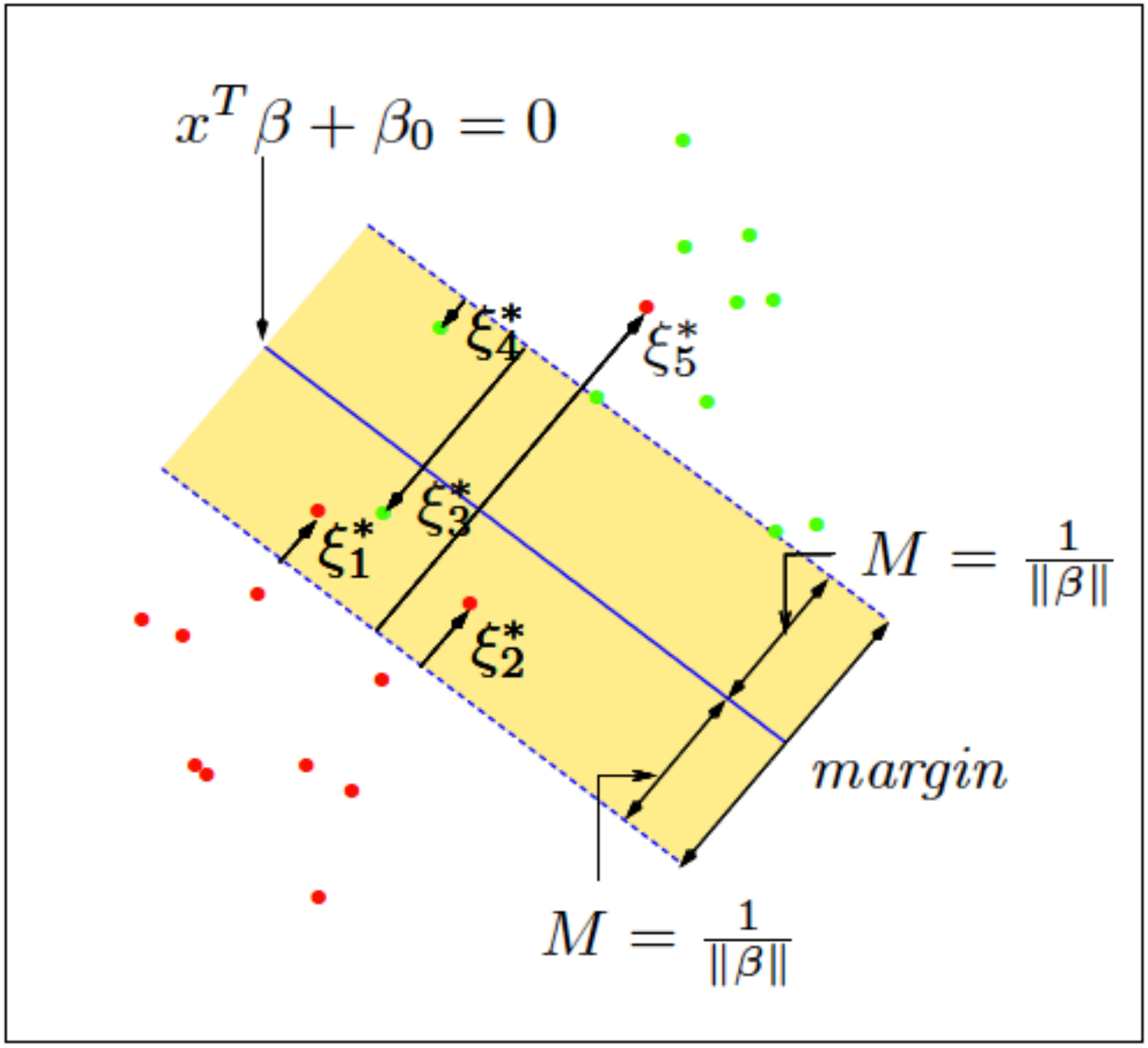
Se minimiza el Lagrangiano para obtener el problema dual: Maximizar

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

Con las restricciones

$$\begin{aligned} \beta &= \sum_{i=1}^{N} \alpha_i y_i x_i, \\ 0 &= \sum_{i=1}^{N} \alpha_i y_i, \\ \alpha_i &= C - \mu_i, \ \forall i, \end{aligned}$$

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


$x^T\beta + \beta_0 = 0$
ξ_4^*
ξ_5^*
ξ_3^*
ξ_1^*
ξ_2^*
$M = \frac{1}{\|\beta\|}$
$margin$
$M = \frac{1}{\|\beta\|}$

Datos No Separables

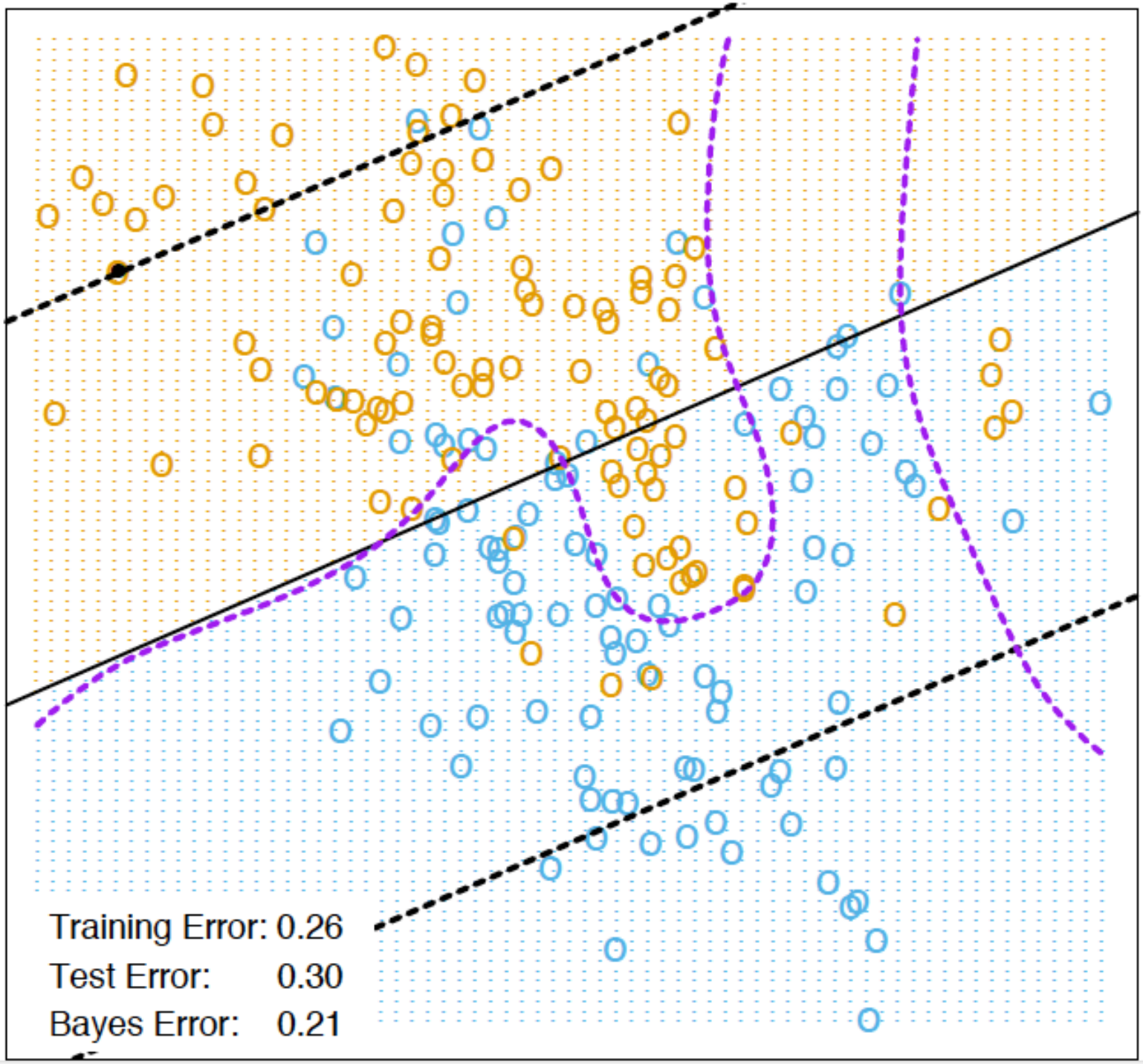
Observaciones

- Los vectores de soporte son los que tienen la restricción activa. Estos son los que están sobre el margen o dentro del margen.
- La solución sólo depende de los vectores de soporte.

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

- La solución es muy robusta, pues no cambia por los puntos muy bien clasificados y no cambia si un punto está muy mal clasificado.

$$0 < \hat{\alpha}_i < C$$


Training Error: 0.26
Test Error: 0.30
Bayes Error: 0.21

Máquinas de Vectores de Soporte

- Hasta el momento sólo se han creado regiones de separación lineales. Si se quieren otras regiones, se puede intentar expandir el espacio de covariantes.
- Si se agregan cuadrados y productos cruzados, se obtienen regiones de clasificación cuadrática.
- Suponga que se hace una transformación de las variables

$$h(x_i) = (h_1(x_i), h_2(x_i), \ldots, h_M(x_i))$$

Máquinas de Vectores de Soporte

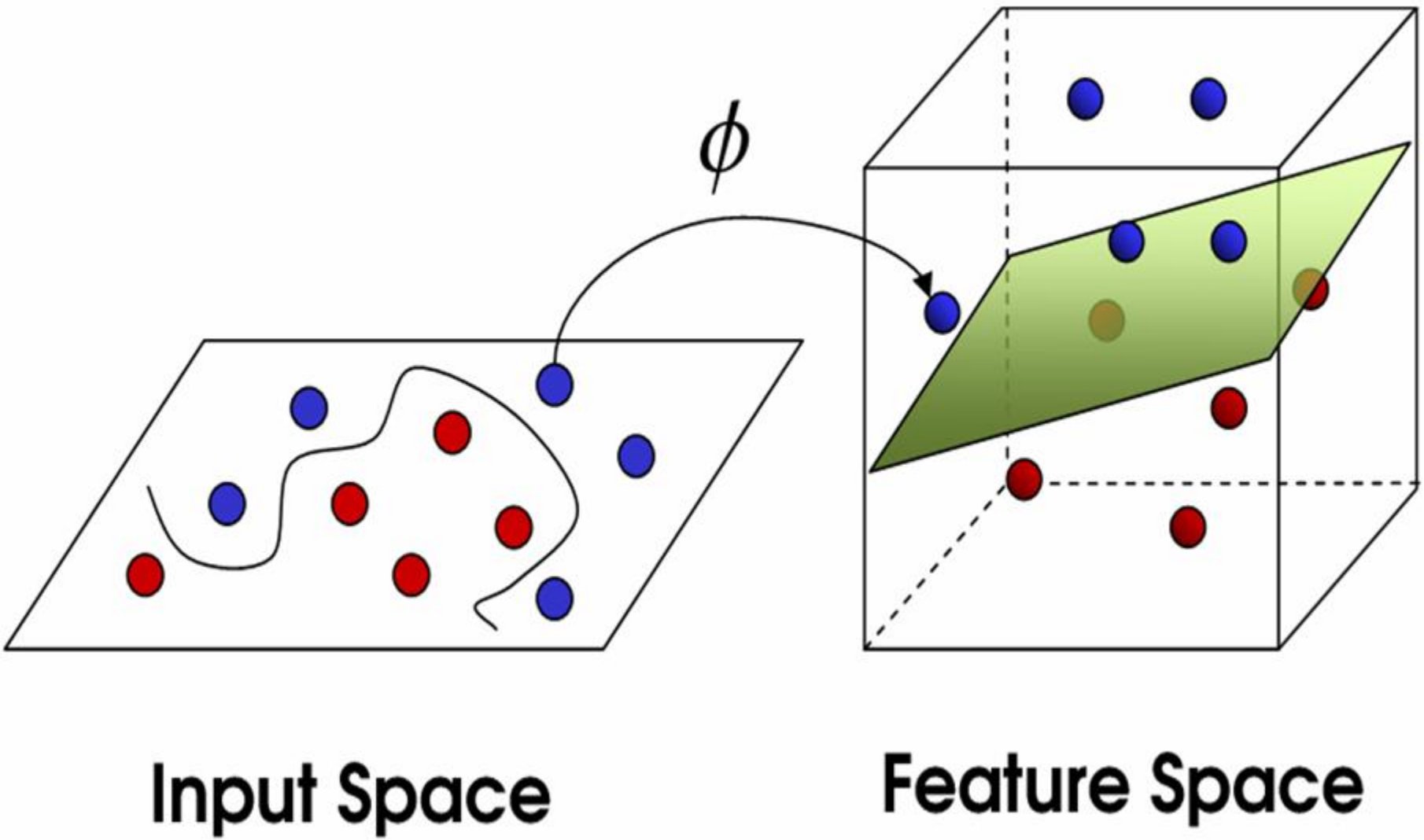
Ahora se ajusta el algoritmo para datos no separables. El problema dual sería maximizar

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

Y el hiperplano solución al problema vendría a ser

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

Dese cuenta de la dependencia del producto interno


ϕ
Input Space
Feature Space

Máquinas de Vectores de Soporte

- La gran idea de Vapnik es que el problema no depende del espacio al que mando mis variables, sino al producto interno en ese espacio.
- Me puedo ir a un espacio de dimensión gigante, y lo único que me preocupa es el producto interno

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

- De hecho, me puedo ir a un espacio de dimensión infinita, donde los datos sí sean separables, y allí aplicar el algoritmo de separación.
- Aquí el Costo juega un papel muy importante

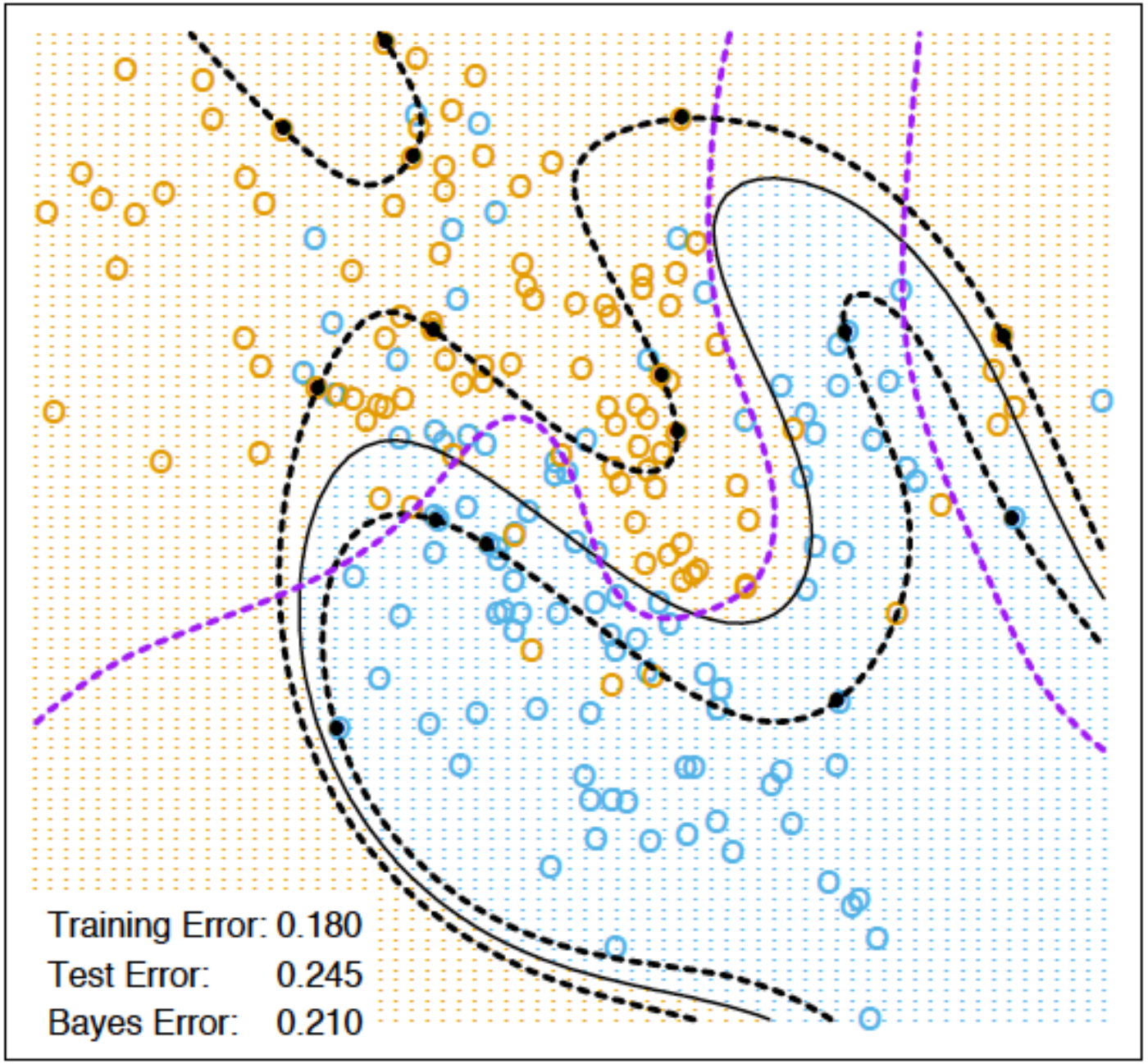
Máquinas de Vectores de Soporte

En resumen, la solución de vectores de soporte es la misma que ajustar un hiperplano con datos no separables, pero en vez de usar producto euclidiano, se usa un kernel.

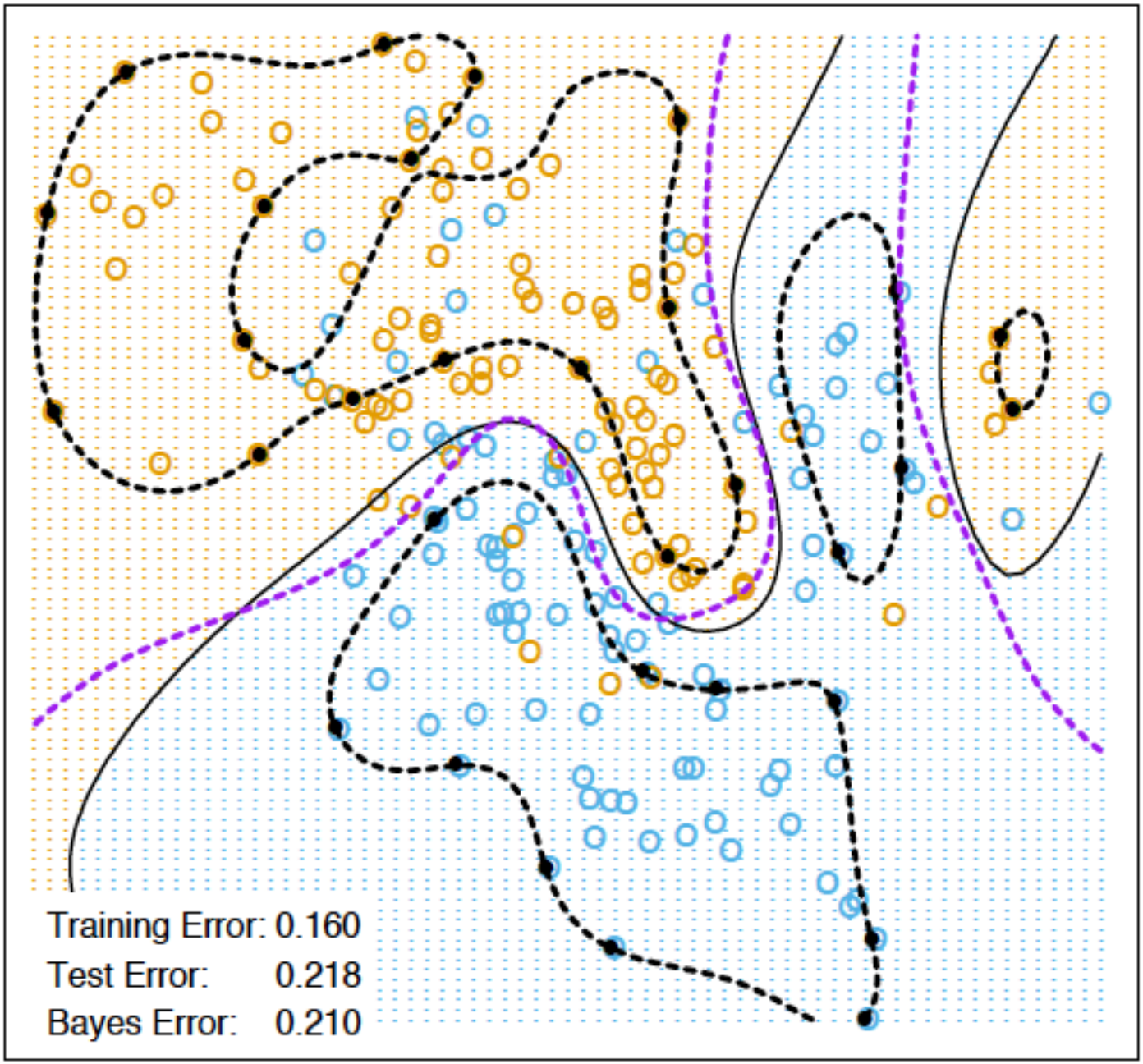
$$\begin{aligned} d\text{th-Degree polynomial: } & K(x, x') = (1 + \langle x, x' \rangle)^d, \\ \text{Radial basis: } & K(x, x') = \exp(-\gamma \|x - x'\|^2), \\ \text{Neural network: } & K(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2) \end{aligned}$$

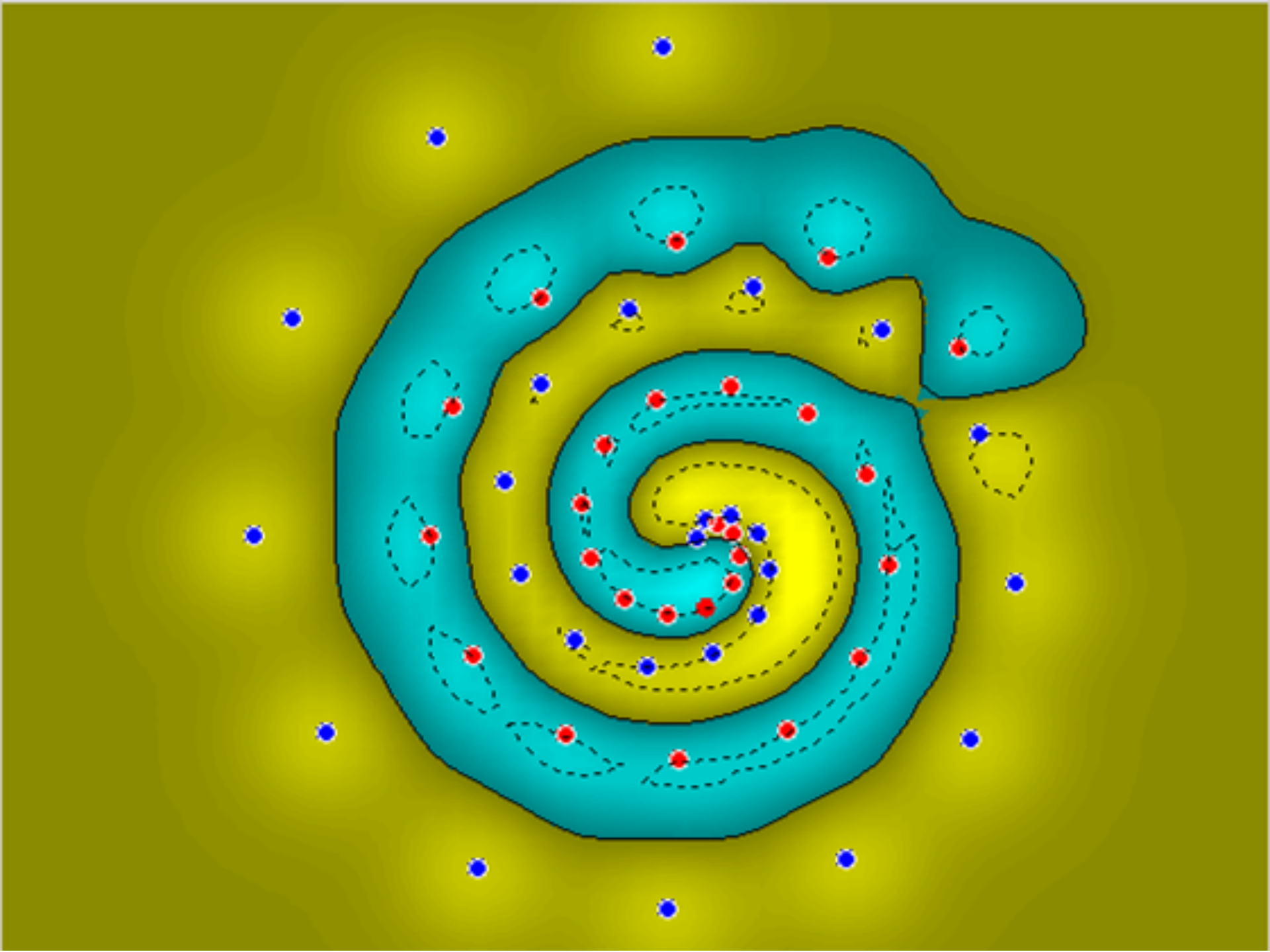
$$\hat{f}(x) = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0$$

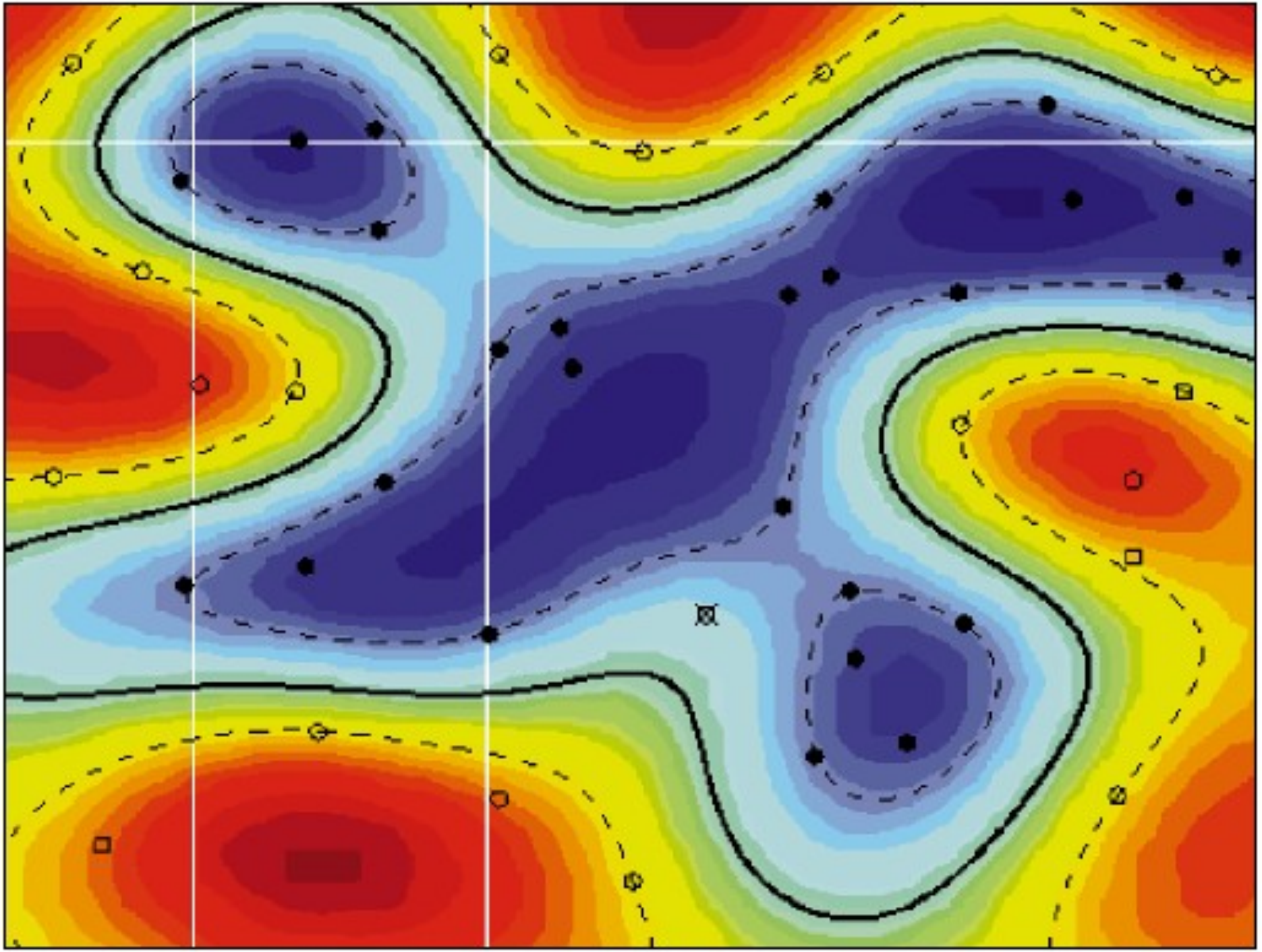
SVM - Degree-4 Polynomial in Feature Space



SVM - Radial Kernel in Feature Space







Máquinas de Vectores de Soporte

- Las máquinas de Soporte tienden a comportarse muy bien, pero la escogencia del Kernel y del costo es de suma importancia.
- Si se lleva el problema a una dimensión muy alta, es posible que haya overfit.
- Por lo general se ajusta un Kernel radial, así que los parámetros a escoger son gamma y el costo.
- Es posible realizar clasificación de varias clases usando SVM, así como regresión.

Máquinas de Vectores de Soporte

- Los algoritmos de optimización pueden no ser muy veloces. Se ajusta SVM a una base de 50000 datos con 123 variables en 20min.
- Esto se complica pues es necesario hacer cross-validation.

Implementación en R

Se usa la librería en C++ Libsvm. Un primer acercamiento es usar el Kernel radial

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

install.packages("e1071")

svm(xTrain,yTrain,kernel="radial",gamma=g,cost=c)

Es necesario calibrar gamma y el costo. Para esto se intentan

$$\gamma = 2^n d^{-2} \quad C = 2^m$$

Donde d es la mediana de la distancia entre clases

Implementación en R

$$\gamma = 2^n d^{-2} \qquad C = 2^m$$

Este código implementa una búsqueda inteligente de los parámetros óptimos por Cross-Validation usando Grid-Search.

```
tune(svm, xTrain, yTrain, xValidation, yValidation,
ranges = list(gamma=listaGammas,costo=listaCostos),
tunecontrol = tune.control(sampling ="fix"),kernel="radial")
```