Fundamentos de Aprendizaje Automático y Reconocimiento de Patrones

Actividades en clase: Práctico 6

Graciana Castro, Martín Schmidt, Federico Lecumberry, Guillermo Carbajal

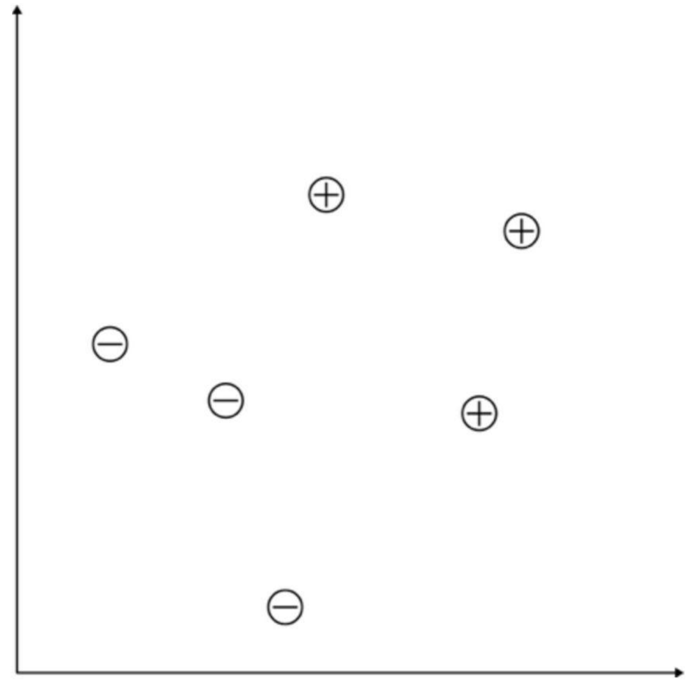
Instituto de Ingeniería Eléctrica

2024

Temario

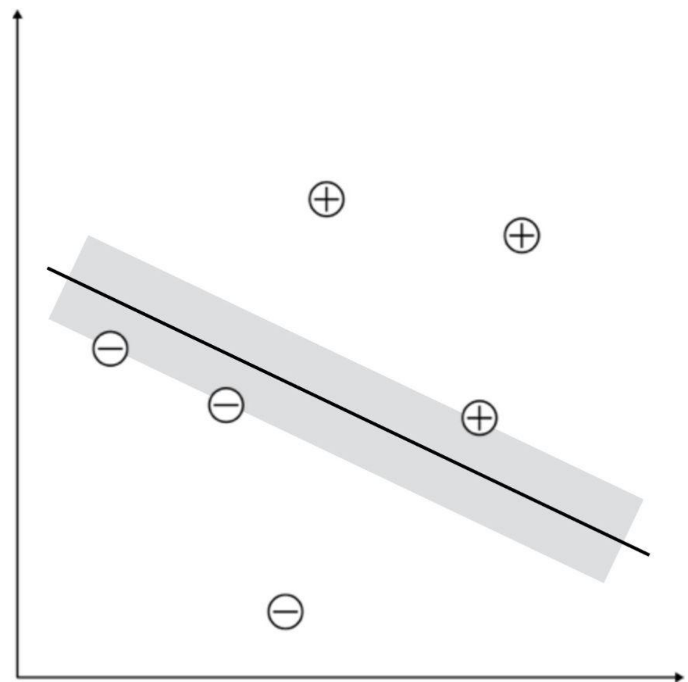
- Ejercicio 1: Formulación de C-SVM
- Ejercicio 2: Separando sonidos con características intuitivas
- Ejercicio 3: SVM en un problema de mediana escala multiclase
- Ejercicio 4: Resolviendo SVM con un paquete de optimización

Support Vector Machines (SVM)

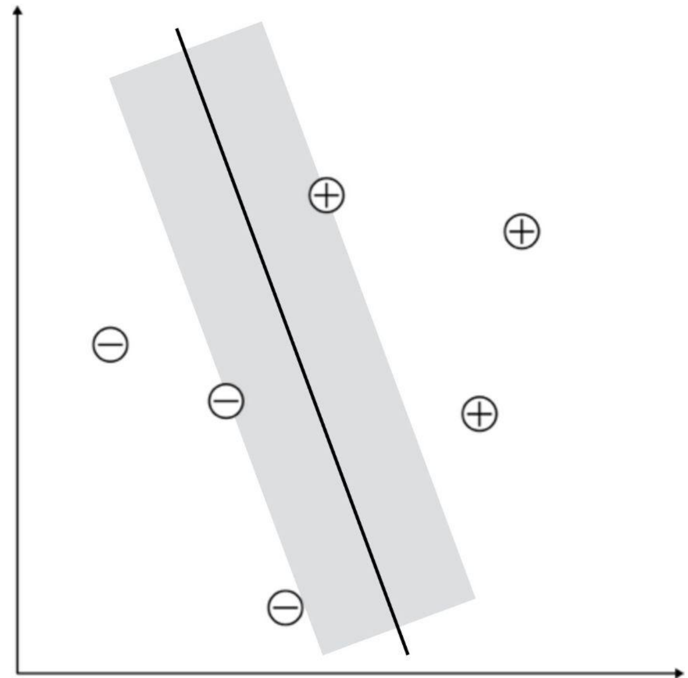


$$\{(\vec{x}_1, y_1), \cdots, (\vec{x}_n, y_n)\} \quad \vec{x}_i \in \mathbb{R}^m, y_i \in \{-1, +1\}$$

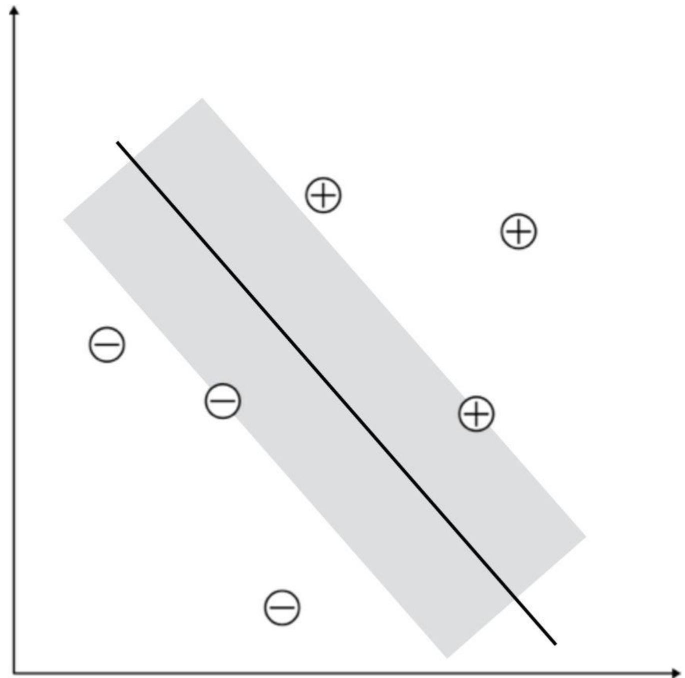
Support Vector Machines (SVM)



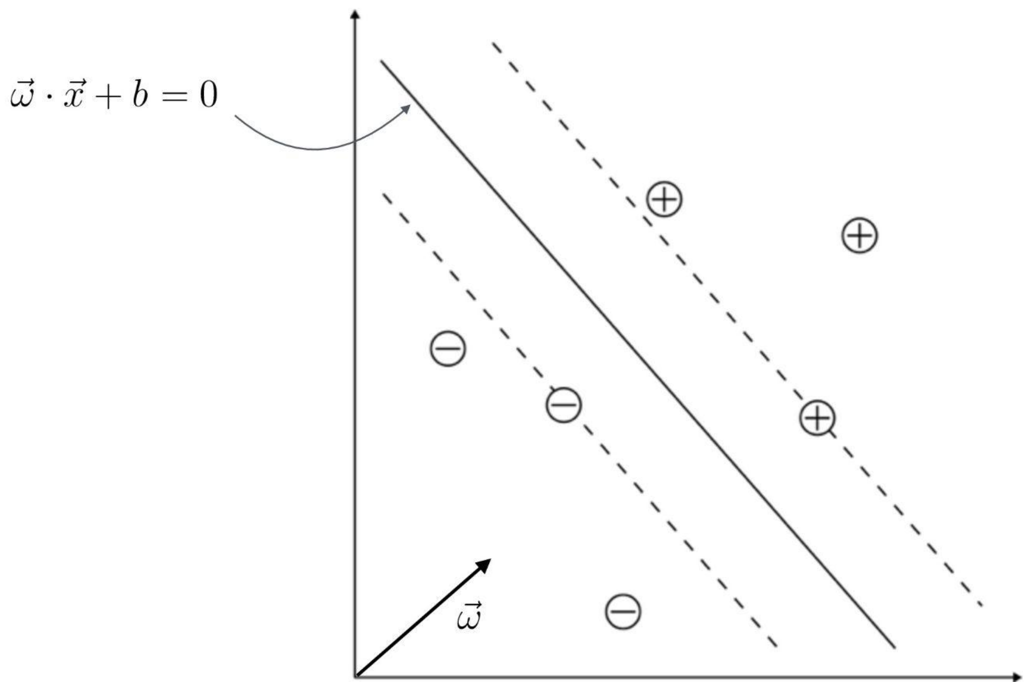
Support Vector Machines (SVM)



Support Vector Machines (SVM)



Support Vector Machines (SVM)



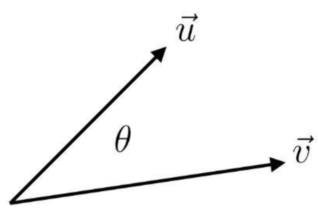
Decision rule: $\vec{\omega} \cdot \vec{u} + b \geq 0 \Rightarrow \vec{u} \in \oplus$

Support Vector Machines (SVM)

- Inner product (dot product) between two vectors

$$\vec{u} \cdot \vec{v} = \langle \vec{u}, \vec{v} \rangle = \vec{u}^\intercal \vec{v} = \vec{v}^\intercal \vec{u} = \sum_i u_i v_i = u_1 v_1 + u_2 v_2 + \cdots + u_m v_m$$

- Geometric interpretation:
  - Angle between vectors
  - Projection



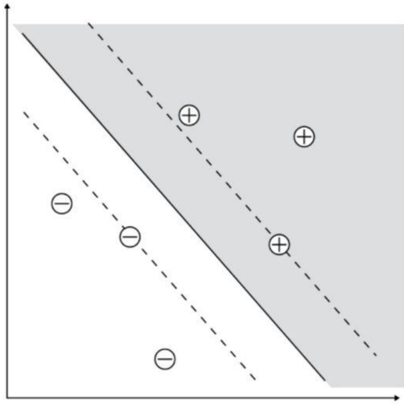
$$\vec{u} \cdot \vec{v} = \|\vec{u}\| \|\vec{v}\| \cos\theta$$

Support Vector Machines (SVM)

- The regions are defined by

$$\vec{\omega} \cdot \vec{x}_{+} + b \geq 0$$

$$\vec{\omega} \cdot \vec{x}_{-} + b \leq 0$$



Support Vector Machines (SVM)

- The regions are defined by

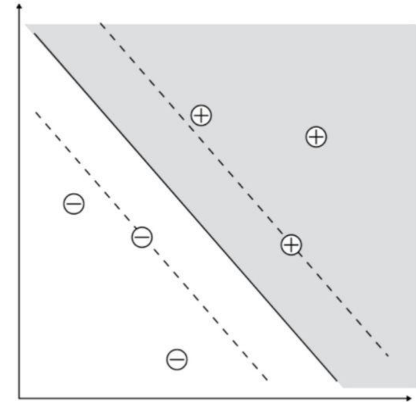
$$\vec{\omega} \cdot \vec{x}_+ + b \geq 0$$

$$\vec{\omega} \cdot \vec{x}_- + b \leq 0$$

- Increase the gap to the decision boundary

$$\vec{\omega} \cdot \vec{x}_+ + b \geq 1$$

$$\vec{\omega} \cdot \vec{x}_- + b \leq -1$$

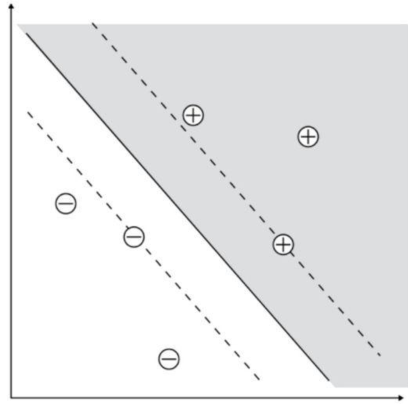


Support Vector Machines (SVM)

- The regions are defined by

$$\vec{\omega} \cdot \vec{x}_+ + b \geq 0$$

$$\vec{\omega} \cdot \vec{x}_- + b \leq 0$$



- Increase the gap to the decision boundary

$$\vec{\omega} \cdot \vec{x}_+ + b \geq 1$$

$$\vec{\omega} \cdot \vec{x}_- + b \leq -1$$

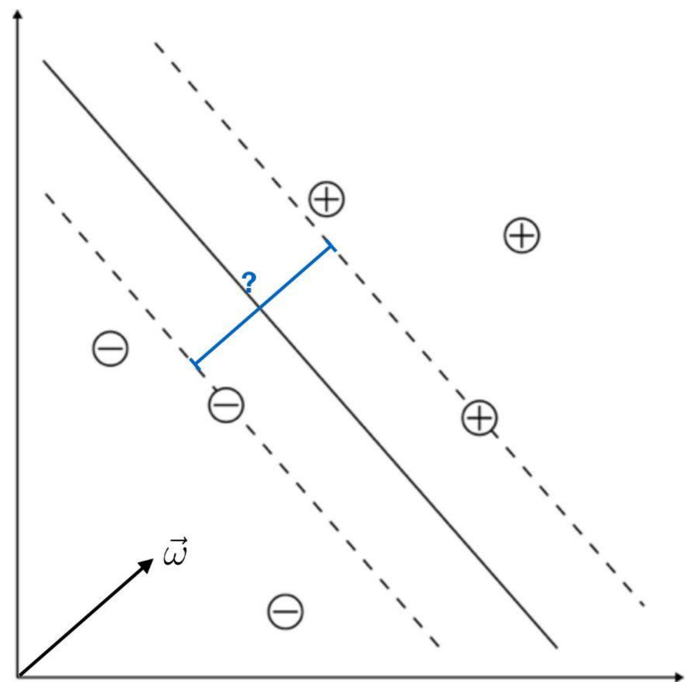
- Using

$$y_i = \begin{cases} +1 & \vec{x}_+ \in \oplus \\ -1 & \vec{x}_- \in \ominus \end{cases}$$

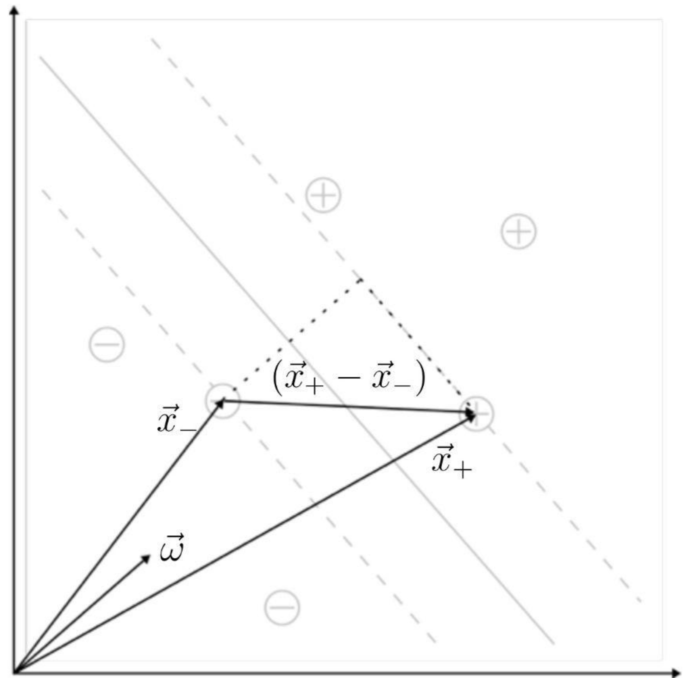
$$y_i(\vec{\omega} \cdot \vec{x} + b) - 1 \geq 0$$

and the equality holds for the samples in the “gutter" of this "street"

Support Vector Machines (SVM)



Support Vector Machines (SVM)



Support Vector Machines (SVM)

- For these two samples

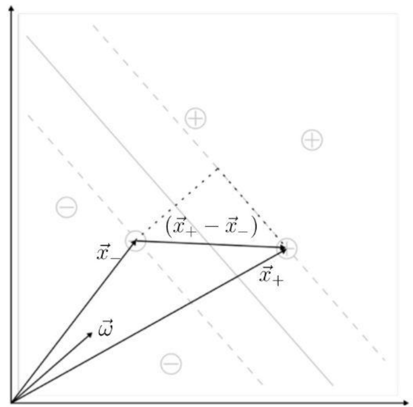
$$y_i(\vec{\omega} \cdot \vec{x} + b) - 1 = 0 \quad \Longrightarrow \quad \begin{aligned} \vec{\omega} \cdot \vec{x}_+ &= 1 - b \\ \vec{\omega} \cdot \vec{x}_- &= -(1 + b) \end{aligned}$$

- The width of the gap can be computed by

$$\begin{aligned} (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{\omega}}{\|\vec{\omega}\|} &= ((1-b) + (1+b))\frac{1}{\|\vec{\omega}\|} \\ &= \frac{2}{\|\vec{\omega}\|} \end{aligned}$$

…and we want it as big as possible.

$$\max \frac{1}{\|\vec{\omega}\|}$$



Support Vector Machines (SVM)

$$\max \frac{1}{||\vec{\omega}||}$$

Support Vector Machines (SVM)

$$\min ||\vec{\omega}||$$

Support Vector Machines (SVM)

$$\min \frac{1}{2} \|\vec{\omega}\|^2$$

Support Vector Machines (SVM)

$$\min \frac{1}{2}\|\vec{\omega}\|^2$$

With some constraints $y_i(\vec{\omega} \cdot \vec{x} + b) - 1 = 0$

Support Vector Machines (SVM)

$$\min \frac{1}{2}\|\vec{\omega}\|^2$$

With some constraints $y_i(\vec{\omega} \cdot \vec{x} + b) - 1 = 0$

- Classical constrained optimisation: Lagrange multipliers!

Support Vector Machines (SVM)

$$\min \frac{1}{2}\|\vec{\omega}\|^2$$

With some constraints $y_i(\vec{\omega} \cdot \vec{x} + b) - 1 = 0$

- Classical constrained optimisation: Lagrange multipliers!

$$\mathcal{L} = \frac{1}{2}\|\vec{\omega}\|^2 - \sum_i \lambda_i \left[y_i(\vec{\omega} \cdot \vec{x} + b) - 1\right]$$

Support Vector Machines (SVM)

$$\min \frac{1}{2}\|\vec{\omega}\|^2$$

With some constraints $\quad y_i(\vec{\omega} \cdot \vec{x} + b) - 1 = 0$

- Classical constrained optimisation: Lagrange multipliers!

$$\mathcal{L} = \frac{1}{2}\|\vec{\omega}\|^2 - \sum_i \lambda_i \left[y_i(\vec{\omega} \cdot \vec{x} + b) - 1\right]$$

$$\frac{\partial \mathcal{L}}{\partial \vec{\omega}} = \vec{\omega} - \sum_i \lambda_i y_i \vec{x}_i = 0 \Rightarrow \vec{\omega}^* = \sum_i \lambda_i y_i \vec{x}_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_i \lambda_i y_i = 0$$

Support Vector Machines (SVM)

$$\min \frac{1}{2}\|\vec{\omega}\|^2$$

With some constraints $\quad y_i(\vec{\omega} \cdot \vec{x} + b) - 1 = 0$

- Classical constrained optimisation: Lagrange multipliers!

$$\mathcal{L} = \frac{1}{2}\|\vec{\omega}\|^2 - \sum_i \lambda_i \left[y_i(\vec{\omega} \cdot \vec{x} + b) - 1\right]$$

$$\frac{\partial \mathcal{L}}{\partial \vec{\omega}} = \vec{\omega} - \sum_i \lambda_i y_i \vec{x}_i = 0 \Rightarrow \boxed{\vec{\omega}^* = \sum_i \lambda_i y_i \vec{x}_i}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_i \lambda_i y_i = 0$$

Support Vector Machines (SVM)

$$\min \frac{1}{2}\|\vec{\omega}\|^2$$

With some constraints $\quad y_i(\vec{\omega} \cdot \vec{x} + b) - 1 = 0$

- Classical constrained optimisation: Lagrange multipliers!

$$\mathcal{L} = \frac{1}{2}\|\vec{\omega}\|^2 - \sum_i \lambda_i \left[y_i(\vec{\omega} \cdot \vec{x} + b) - 1\right]$$

$$\frac{\partial \mathcal{L}}{\partial \vec{\omega}} = \vec{\omega} - \sum_i \lambda_i y_i \vec{x}_i = 0 \Rightarrow \boxed{\vec{\omega}^* = \sum_i \lambda_i y_i \vec{x}_i}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_i \lambda_i y_i = 0$$

- The optimal $\vec{\omega}^*$ (hyperplane) is a linear combination of some ($\lambda_i \neq 0$) of the sample vectors (the *support vectors*).

Temario

- Ejercicio 1: Formulación de C-SVM
- Ejercicio 2: Separando sonidos con características intuitivas
- Ejercicio 3: SVM en un problema de mediana escala multiclase
- Ejercicio 4: Resolviendo SVM con un paquete de optimización

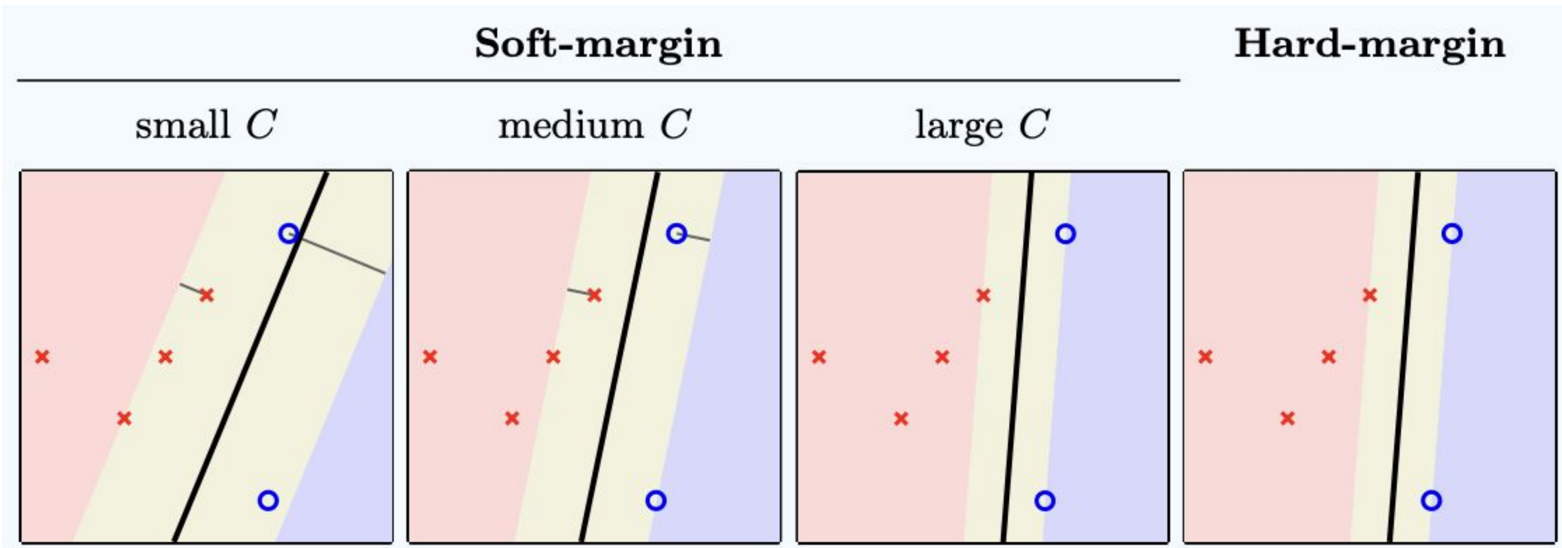
Ejercicio 1: formulación de C-SVM

El algoritmo C-SVM da lugar al siguiente problema de optimización:

$$\begin{cases} \min\limits_{\mathbf{w},b,\xi} & f(\mathbf{w},\xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \xi_n \\ \text{sujeto a} & \\ & y_n\left(\mathbf{w}^T\Phi\left(x_n\right)+b\right) \geq 1-\xi_n \quad \forall n = 1,\ldots,N \\ & \xi_n \geq 0 \end{cases}$$

margin violation

Ejercicio 1: formulación de C-SVM



Ejercicio 1: formulación de C-SVM

El algoritmo C-SVM da lugar al siguiente problema de optimización:

$$\begin{cases} \min\limits_{\mathbf{w},b,\xi} & f(\mathbf{w},\xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \\ \text{sujeto a} & \\ & y_n\left(\mathbf{w}^T\Phi\left(x_n\right)+b\right) \geq 1-\xi_n \quad \forall n=1,\ldots,N \\ & \xi_n \geq 0 \end{cases}$$

Parte (1a)

Escribir el Lagrangiano del problema.

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \cdots$$

Ejercicio 1: formulación de C-SVM

El algoritmo C-SVM da lugar al siguiente problema de optimización:

$$\begin{cases} \min\limits_{\mathbf{w},b,\xi} & f(\mathbf{w},\xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \\ \text{sujeto a} & \\ & y_n\left(\mathbf{w}^T\Phi\left(x_n\right)+b\right) \geq 1-\xi_n \quad \forall n=1,\ldots,N \\ & \xi_n \geq 0 \end{cases}$$

Parte (1a)

Escribir el Lagrangiano del problema.

$$\mathcal{L}(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\,||\mathbf{w}||^2 + \sum_n C\xi_n - \sum_n \alpha_i\left[y_n\left(\mathbf{w}^T\Phi(\mathbf{x}_n)+b\right)-1+\xi_n\right] - \sum_n \beta_n\xi_n$$

$$\text{con } \alpha_i,\, \beta_i \geq 0$$

Ejercicio 1: formulación de C-SVM

El algoritmo C-SVM da lugar al siguiente problema de optimización:

$$\begin{cases} \min\limits_{\mathbf{w},b,\xi} & f(\mathbf{w},\xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \\ \text{sujeto a} & \\ & y_n\left(\mathbf{w}^T\Phi\left(x_n\right)+b\right) \geq 1-\xi_n \quad \forall n=1,\ldots,N \\ & \xi_n \geq 0 \end{cases}$$

Parte (1a)

Escribir el Lagrangiano del problema.

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \cdots$$

Parte (1b)

Argumentar que el valor del Lagrangiano $\mathcal{L}$ está acotado superiormente por la función de costo original f.

Ejercicio 1: formulación de C-SVM

Se define la función

$$\theta_{\mathcal{P}}(\mathbf{w}, b, \boldsymbol{\xi}) = \max_{\boldsymbol{\alpha},\boldsymbol{\beta};\alpha_n \geq 0, \beta_n \geq 0} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Parte (1c)

Argumentar que resolver

$$\min_{\mathbf{w}, b, \xi} \theta_{\mathcal{P}}(\mathbf{w}, b, \boldsymbol{\xi})$$

es equivalente a encontrar el mínimo de $f(\mathbf{w}, \xi)$
con las restricciones del problema.

Ejercicio 1: formulación de C-SVM

Se define ahora la función *dual*

$$\theta_{\mathcal{D}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\mathbf{w}, b, \boldsymbol{\xi}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

En general se cumple para el óptimo del problema dual d^* que

$$d^* = \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \theta_{\mathcal{D}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \leq \min_{\mathbf{w}, b, \xi} \theta_{\mathcal{P}}(\mathbf{w}, b, \boldsymbol{\xi}) = p^*$$

(*dual gap*)

Parte (1d)

Mostrar que el problema dual puede escribirse como

$$\begin{cases} \min\limits_{\boldsymbol{\alpha}} & \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} - 1^T \boldsymbol{\alpha} \\ \text{sujeto a} & \\ & 0 \leq \alpha_n \leq C \\ & \sum\limits_n \alpha_n y_n = 0 \end{cases}$$

con $Q_{ij} = y_i y_j K(x_i, x_j)$

Temario

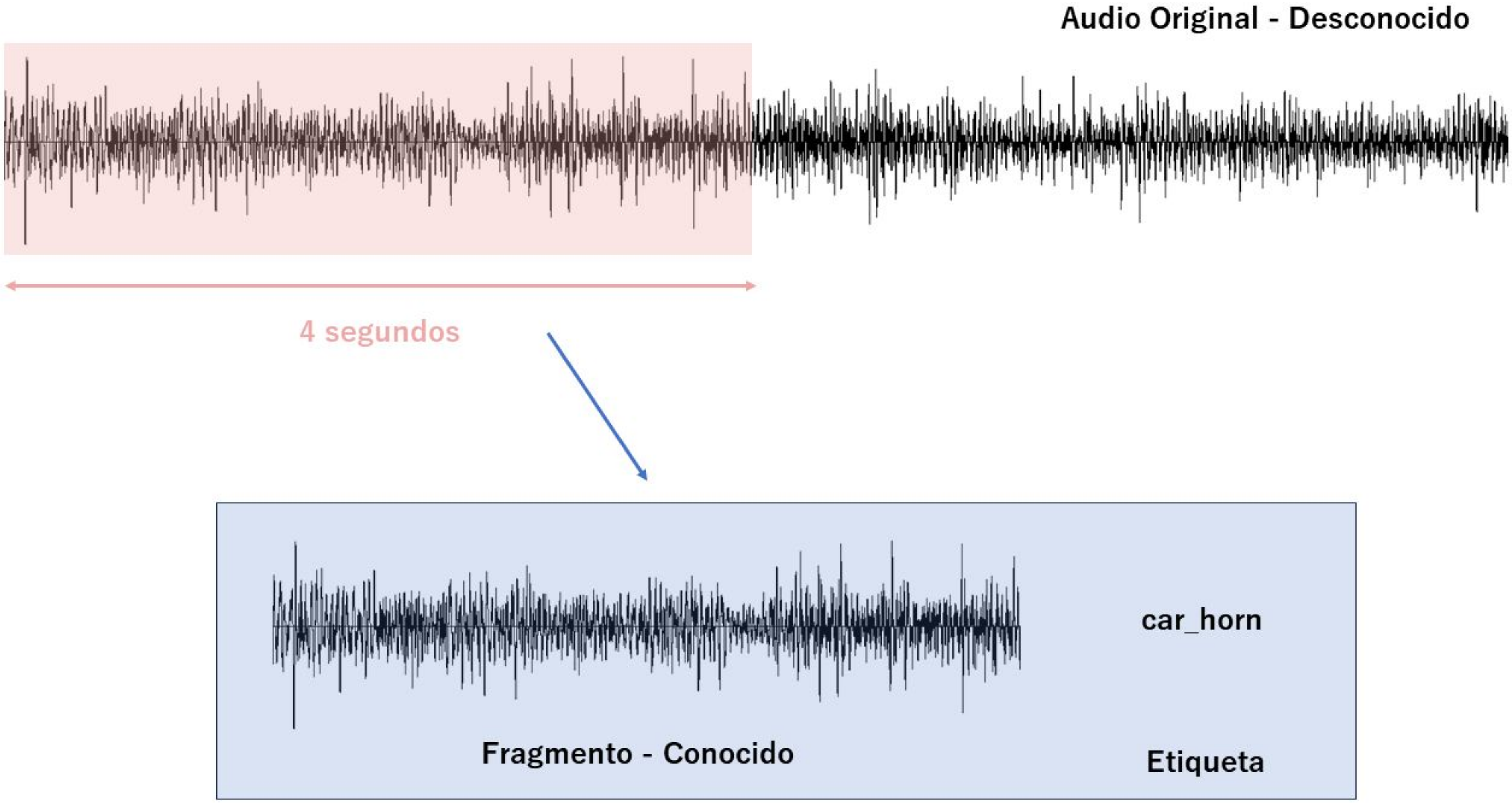
- Ejercicio 1: Formulación de C-SVM
- Ejercicio 2: Separando sonidos con características intuitivas
- Ejercicio 3: SVM en un problema de mediana escala multiclase
- Ejercicio 4: Resolviendo SVM con un paquete de optimización

Ejercicio 2: separando sonidos con características intuitivas

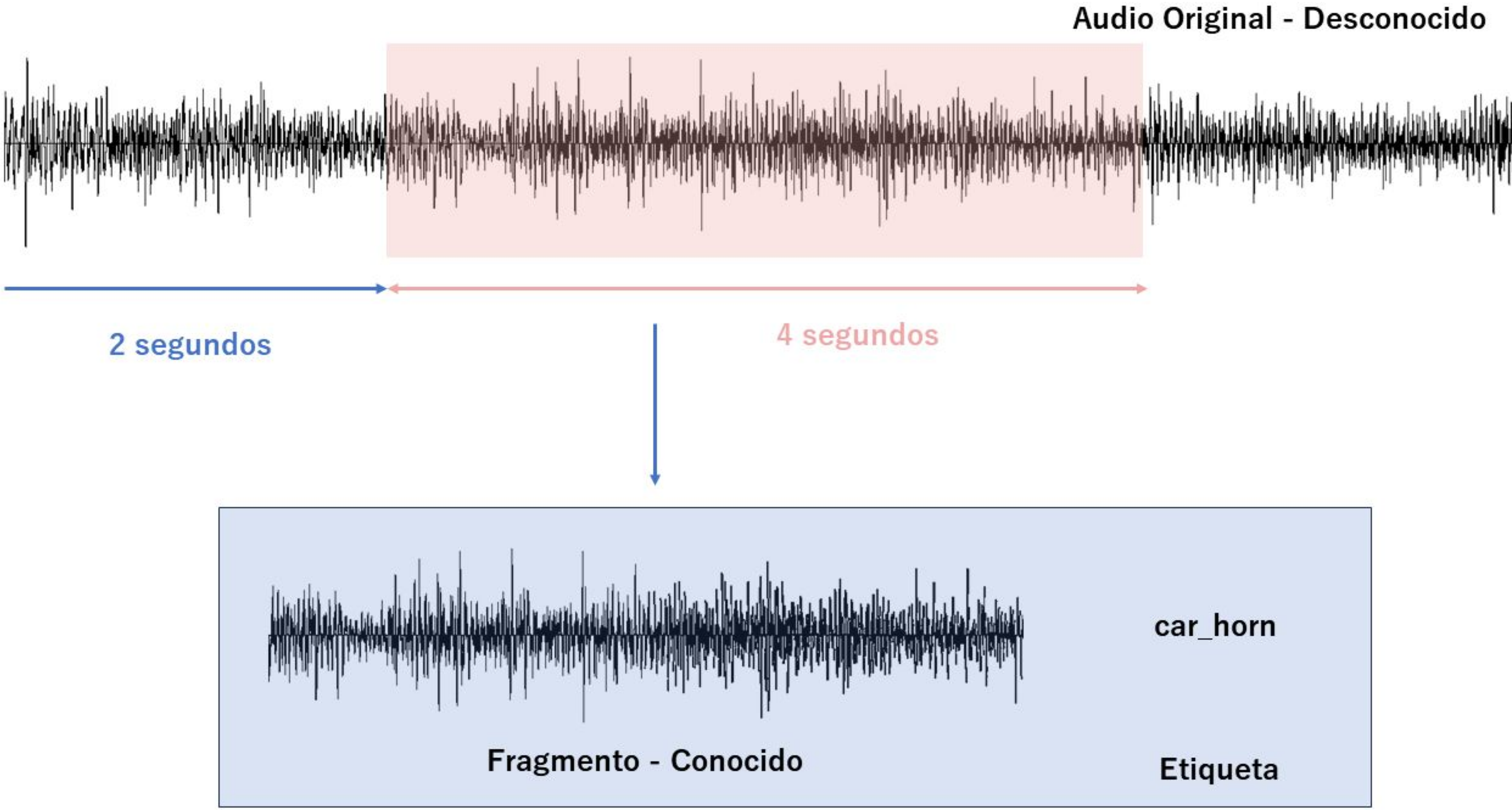
En los ejercicios 2 y 3 de este práctico se trabajará con la base de datos **Urban Sound**. Los sonidos pertenecientes a las siguientes 10 categorías:

1. aire acondicionado (air_conditioner)
2. bocina de auto (car_horn)
3. niños jugando (children_playing)
4. ladrido de perros (dog_bark)
5. taladrado (drilling)
6. motor moderando (engine_idling)
7. disparo de arma de fuego (gun_shot)
8. martillo neumático (jackhammer)
9. sirena (siren)
10. música callejera (street_music)

Ejercicio 2: separando sonidos con características intuitivas



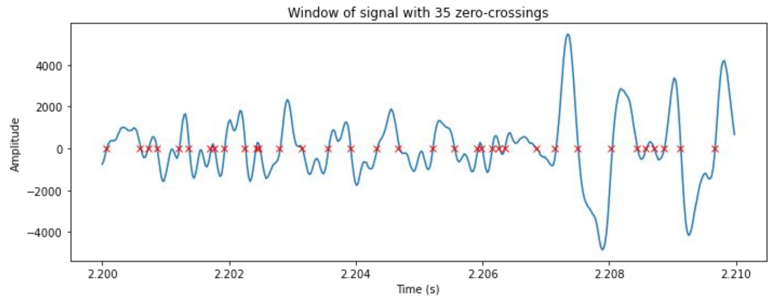
Ejercicio 2: separando sonidos con características intuitivas



Ejercicio 2: separando sonidos con características intuitivas

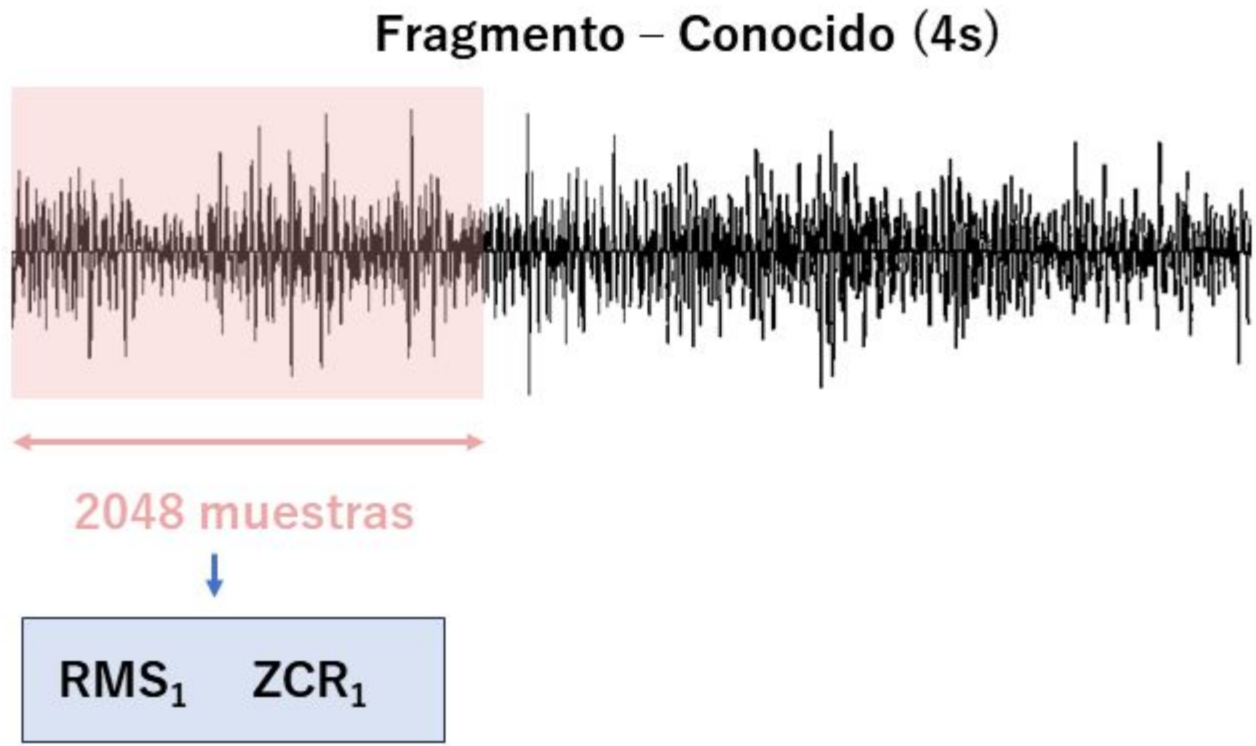
Cada fragmento se divide en ventanas solapadas en un 50% de 2048 muestras y para cada ventana se calculan las siguientes características:

- **ZCR** (Zero-Crossing-Rate): la cantidad de cruces por cero de las muestras de la señal.
- **RMS** (Root-Mean-Square): la raíz del valor cuadrático medio de las muestras de la señal, también conocida como medida de la energía de la señal.

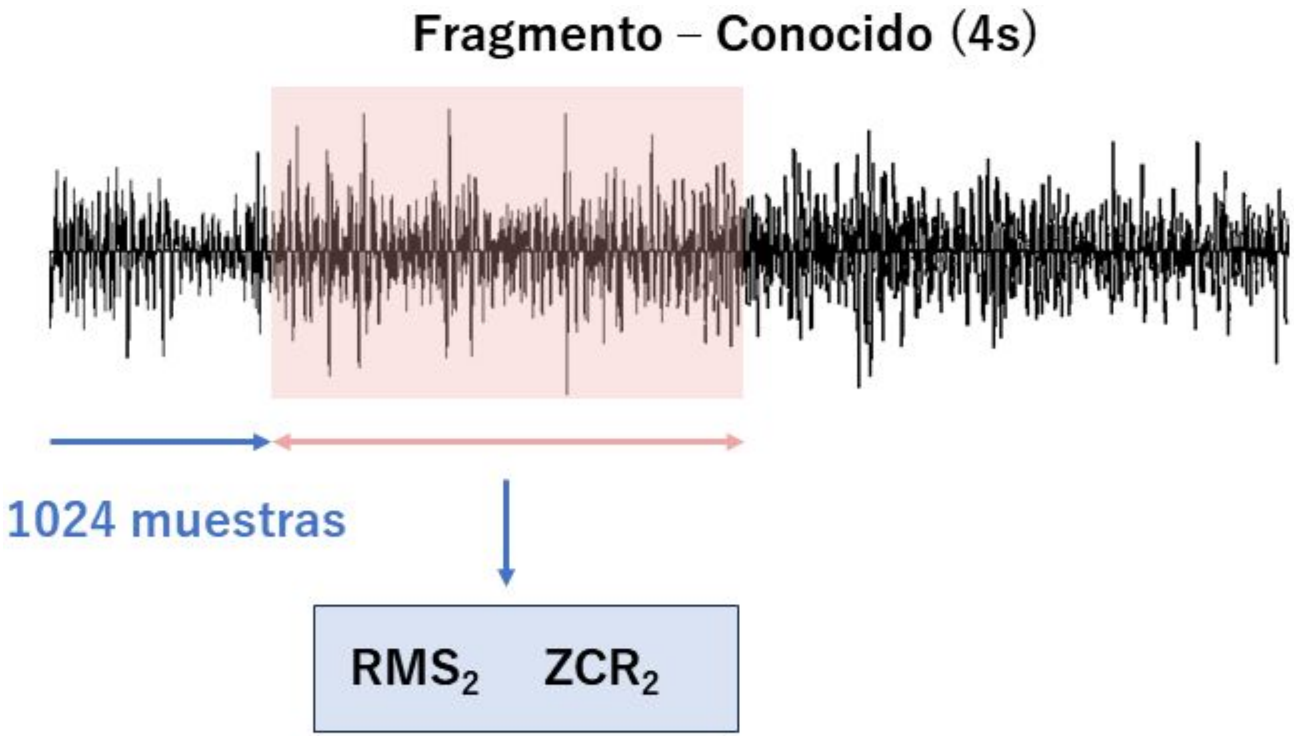


$$x_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$$

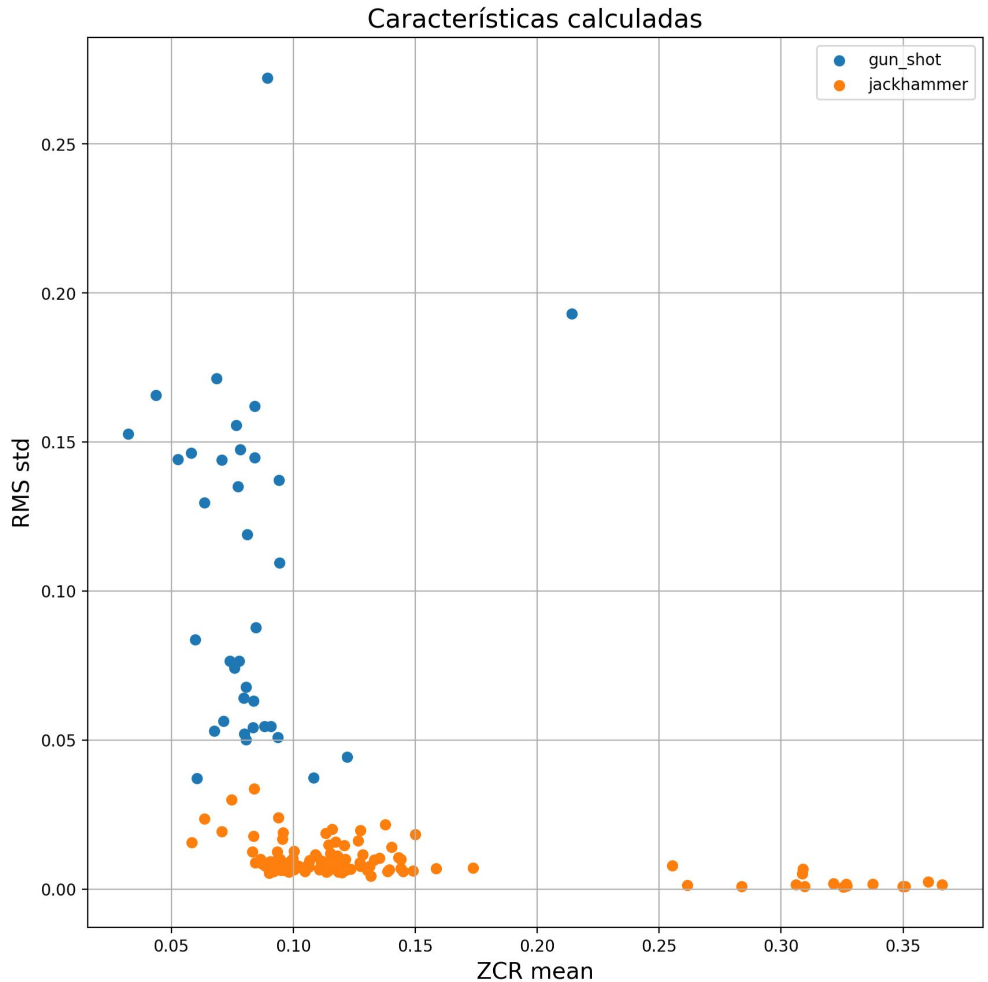
Ejercicio 2: separando sonidos con características intuitivas



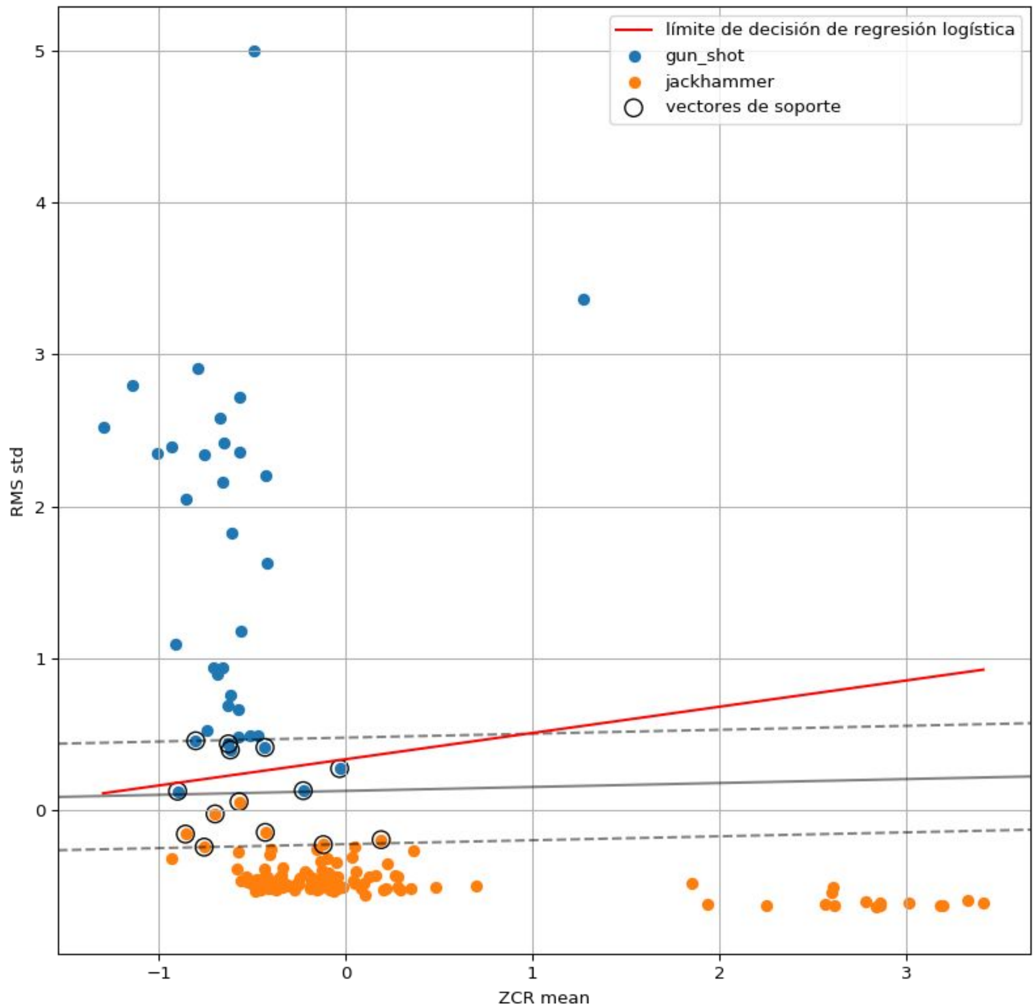
Ejercicio 2: separando sonidos con características intuitivas



Ejercicio 2: separando sonidos con características intuitivas

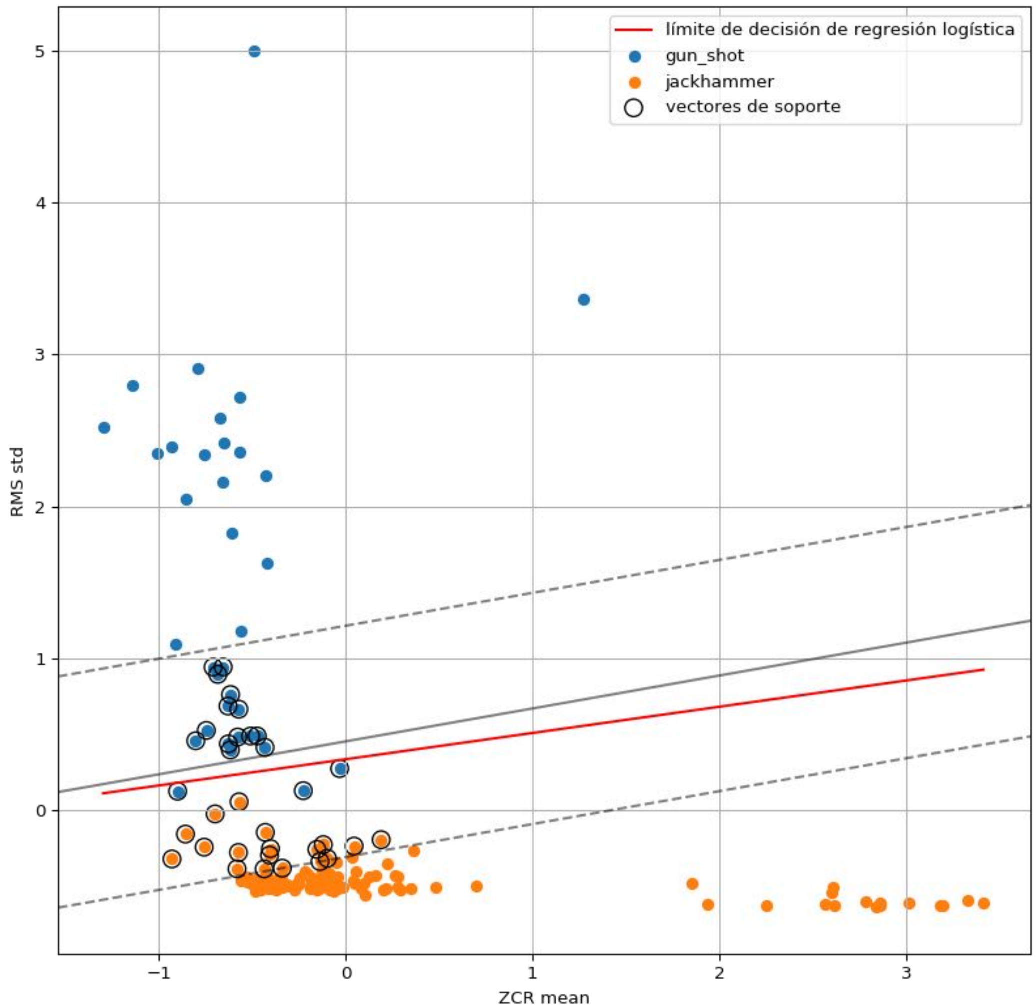


Ejercicio 2: separando sonidos con características intuitivas



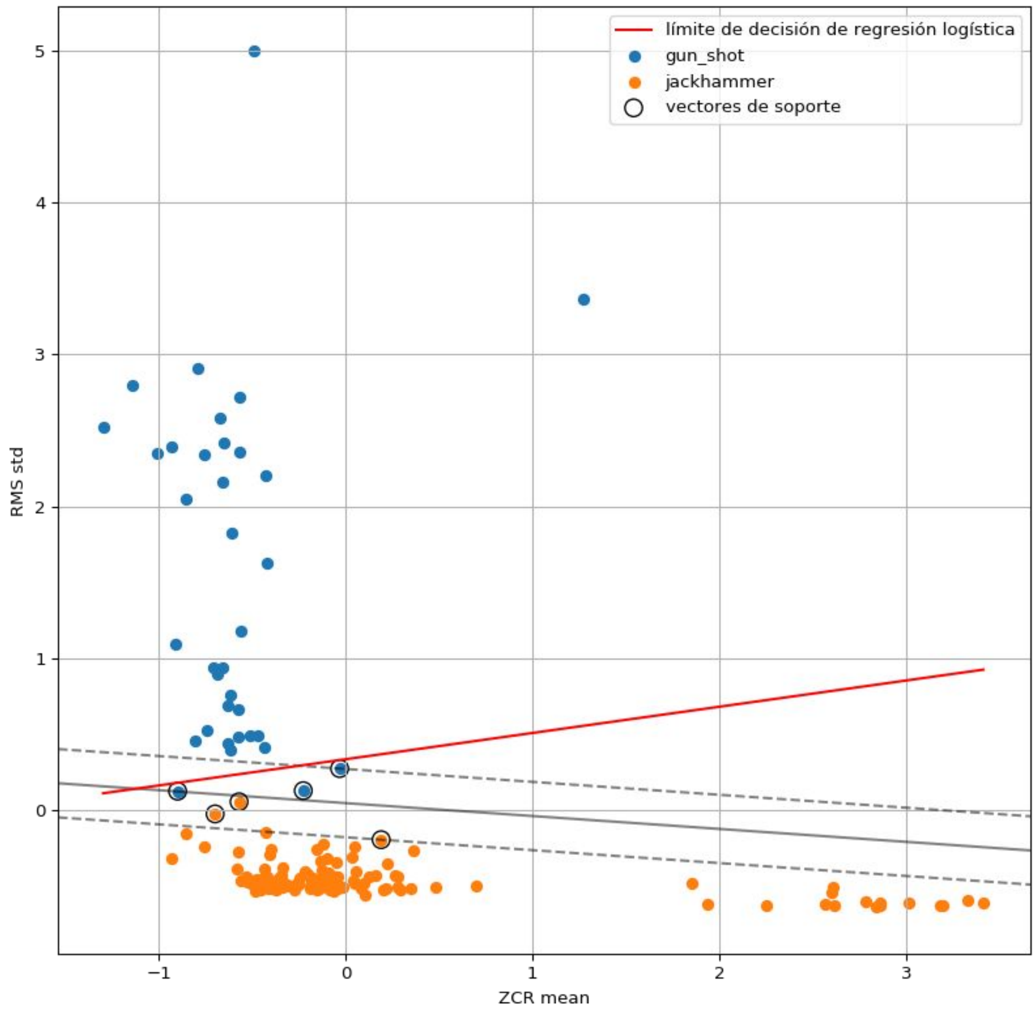
Efecto de C en la solución.
C = 1

Ejercicio 2: separando sonidos con características intuitivas



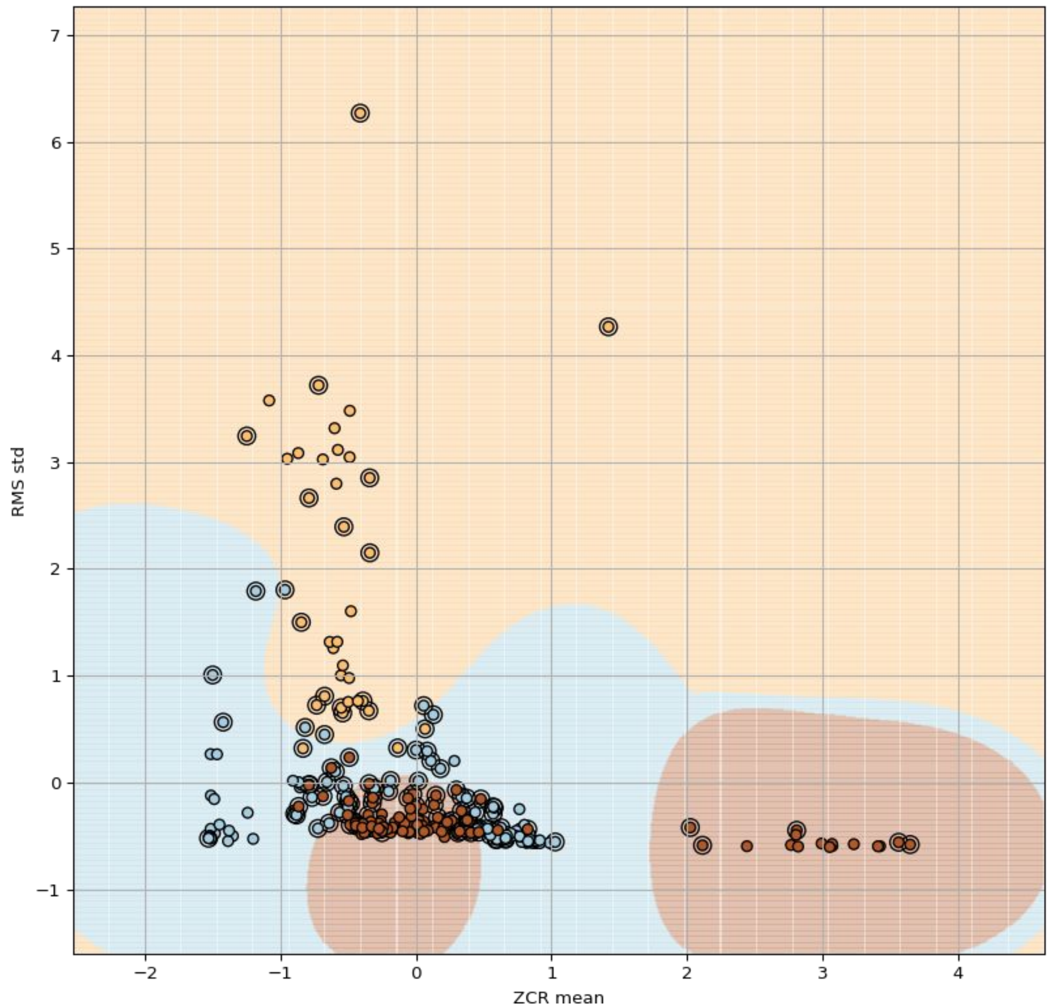
Efecto de C en la solución.
C = 0.1

Ejercicio 2: separando sonidos con características intuitivas



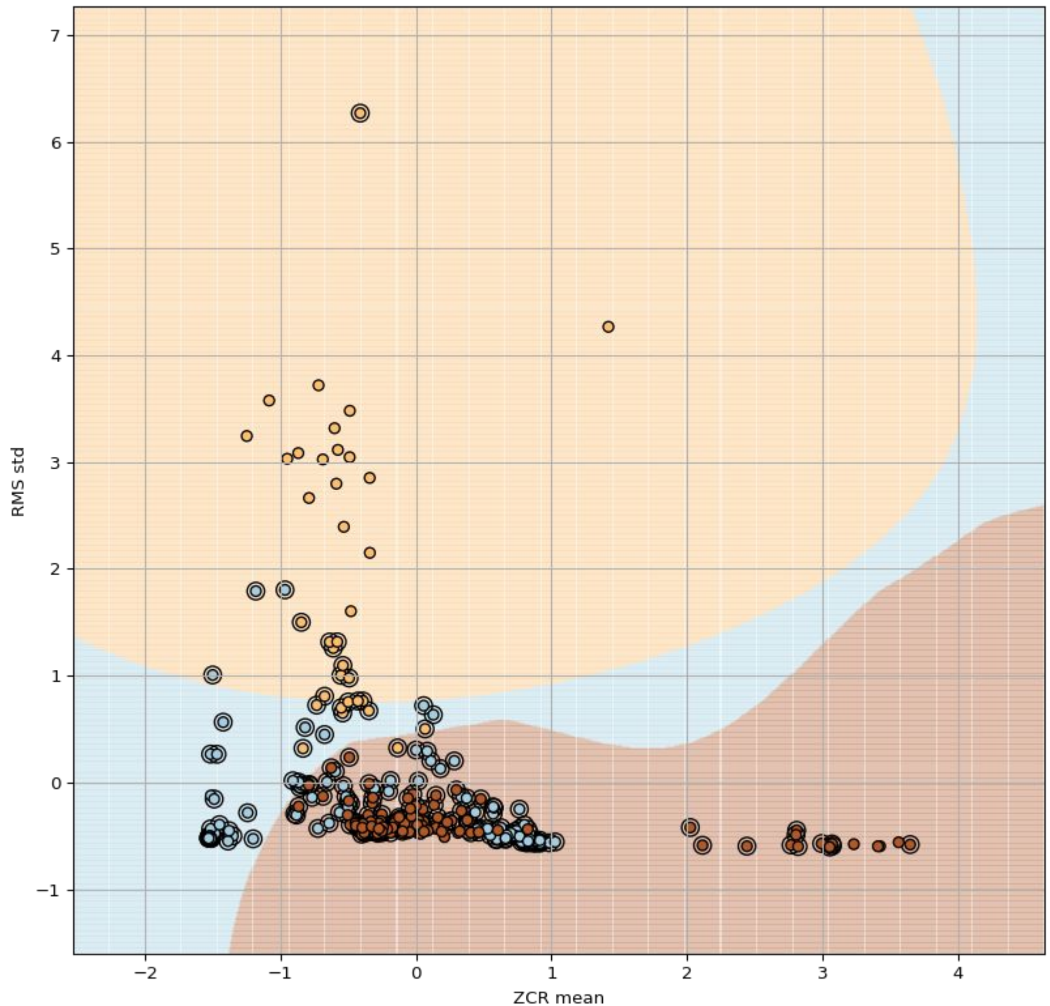
Efecto de C en la solución.
C = 10

Ejercicio 2: separando sonidos con características intuitivas



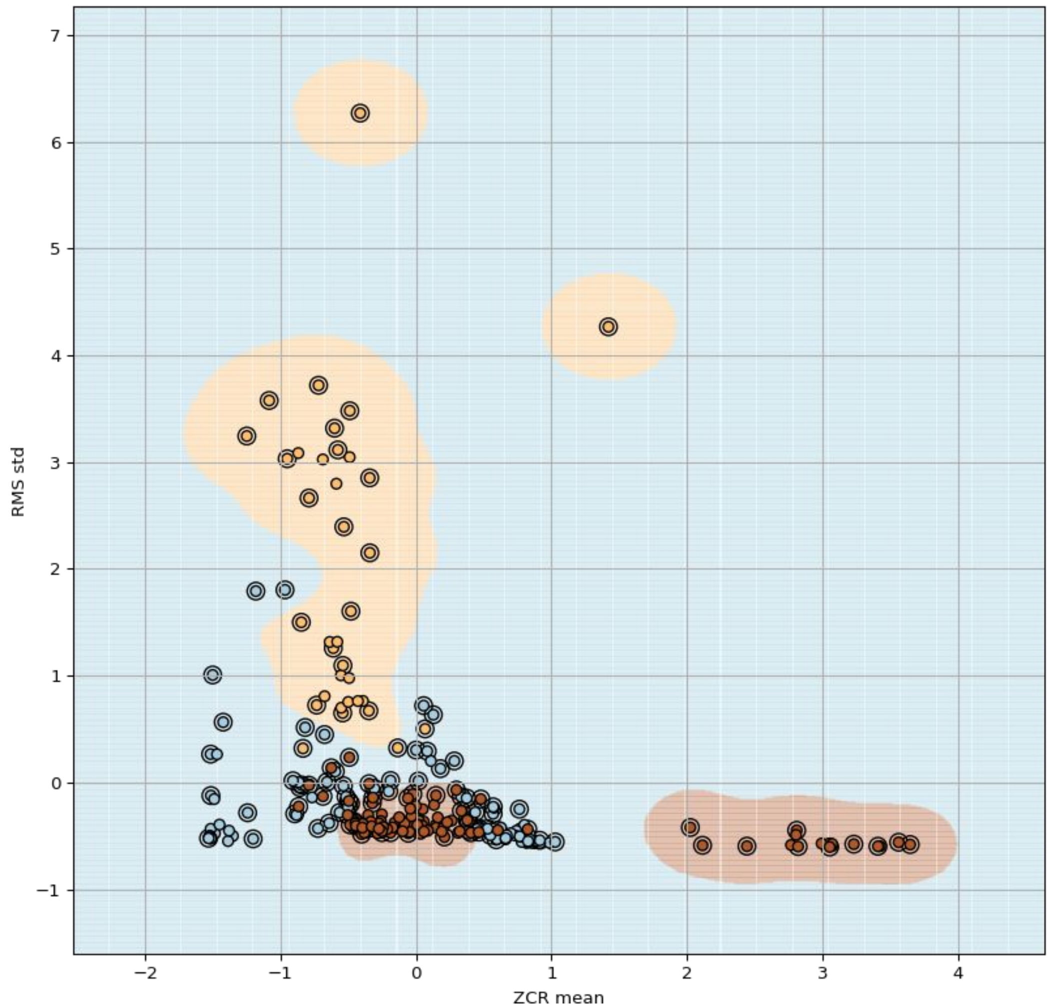
Efecto de ɣ en la solución.
ɣ = 1

Ejercicio 2: separando sonidos con características intuitivas



Efecto de ɣ en la solución.
ɣ = 0.1

Ejercicio 2: separando sonidos con características intuitivas



Efecto de ɣ en la solución.
ɣ = 10

Ejercicio 2: clasificación multiclase en SVM

Dos alternativas principales

1. *Uno contra todos*
   - se construye un clasificador por clase, etiquetando como positivas las muestras de la clase y negativas el resto
   - muy interpretable
   - para clasificar la muestra se elige la clase que haya dado positiva con mayor conanza (o probabilidad)

2. *Uno contra uno*
   - se construye un clasificador por cada par de clases
   - para clasificar una muestra se elige la clase que recibió mayor cantidad de votos
   - como se construyen $n(n-1)/2$ clasificadores suele ser más lento en tiempo de ejecución.
   - menos sensible al desbalance de clases.

La implementación SVC de scikit-learn utiliza por defecto uno contra uno

Ejercicio 2: clasificación multiclase en SVM, kernels

kernel 'linear': $\langle x, x' \rangle$

- un único parámetro C
- controla el compromiso entre minimizar el error de clasificación con el conjunto de entrenamiento y obtener una supercie de decisión simple.
- un C pequeño favorece las supercies de decisiones suaves mientras que un C grande penaliza fuertemente los errores de entrenamiento

kernel 'rbf': $\exp\left(-\gamma \|x - x'\|^2\right)$

- los parámetros son C y ɣ.
- la interpretación de C es similar al kernel 'lineal'
- ɣ define la influencia que tiene una muestra de entrenamiento.
- cuanto más grande es ɣ más rápido se desvance la influencia de una muestra sobre las demás

Cuestionario práctico

passwd: 9837

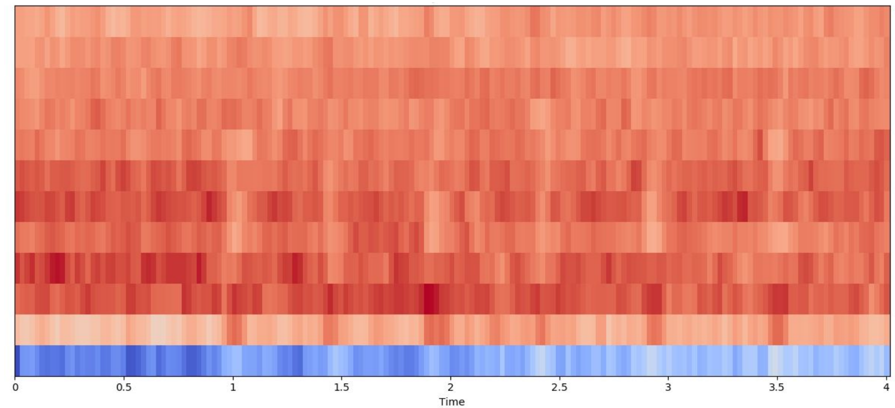
Temario

- Ejercicio 1: Formulación de C-SVM
- Ejercicio 2: Separando sonidos con características intuitivas
- Ejercicio 3: SVM en un problema de mediana escala multiclase
- Ejercicio 4: Resolviendo SVM con un paquete de optimización

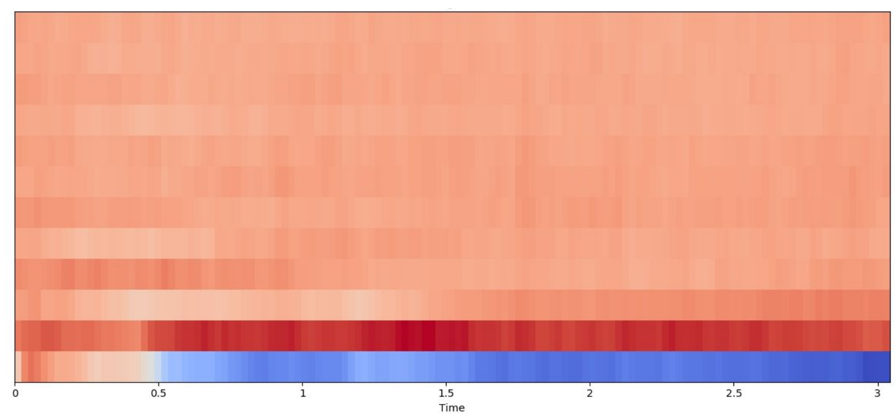
Ejercicio 3: SVM en un problema de mediana escala multiclase

- Se construirá un clasificador SVM para separar las 10 clases de la base *Urband Sound*
- El conjunto de datos original se lo dividió en:
  - 6 *folds* para entrenamiento
  - 2 *folds* para validación
  - 2 *folds* para test (nos reservamos las etiquetas)
- Como características se utilizarán estadísticas calculadas sobre los coecientes *MFCC*.

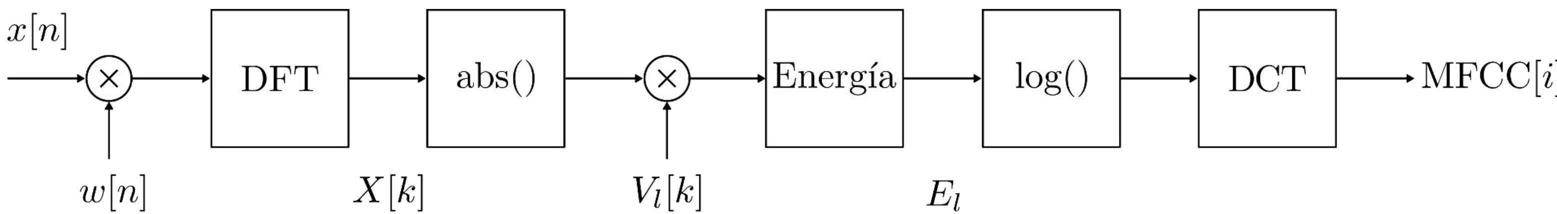
MFCC del martillo neumático



MFCC del disparo de arma de fuego



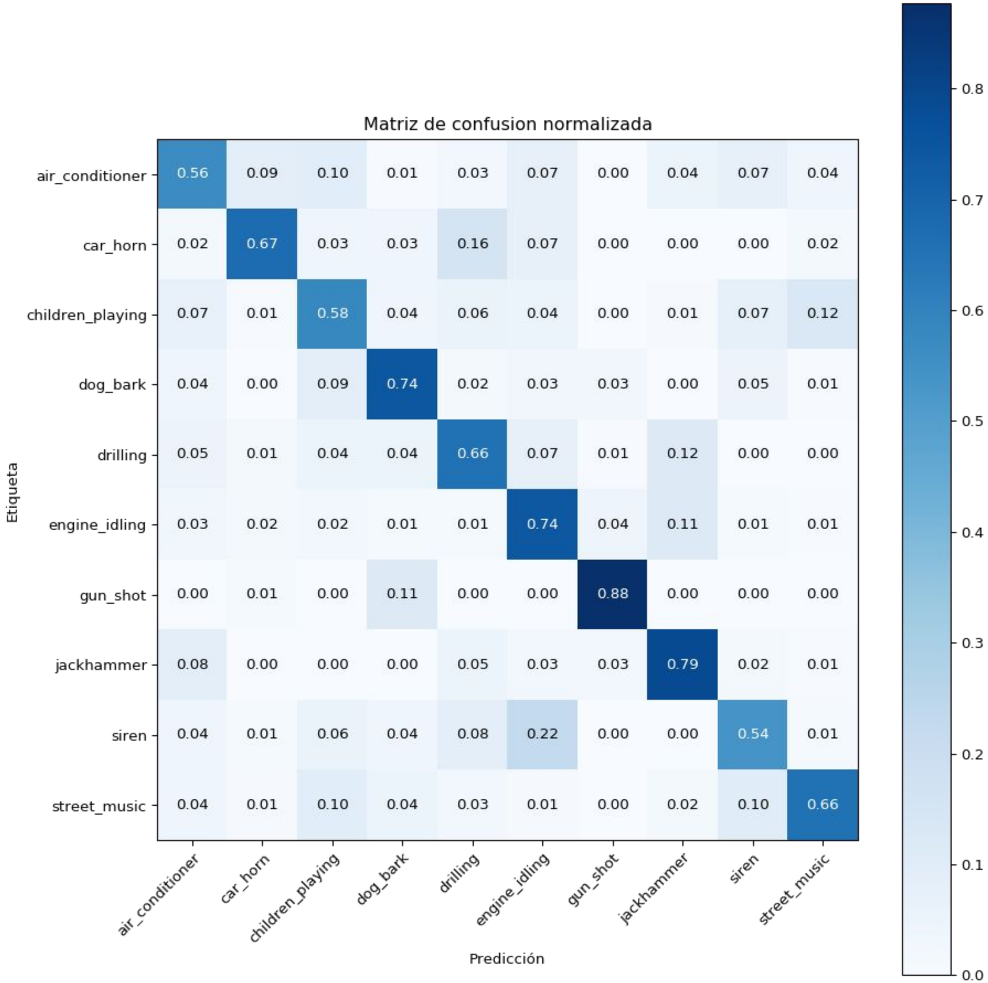
Ejercicio 3: Cálculo de los coecientes MFCC



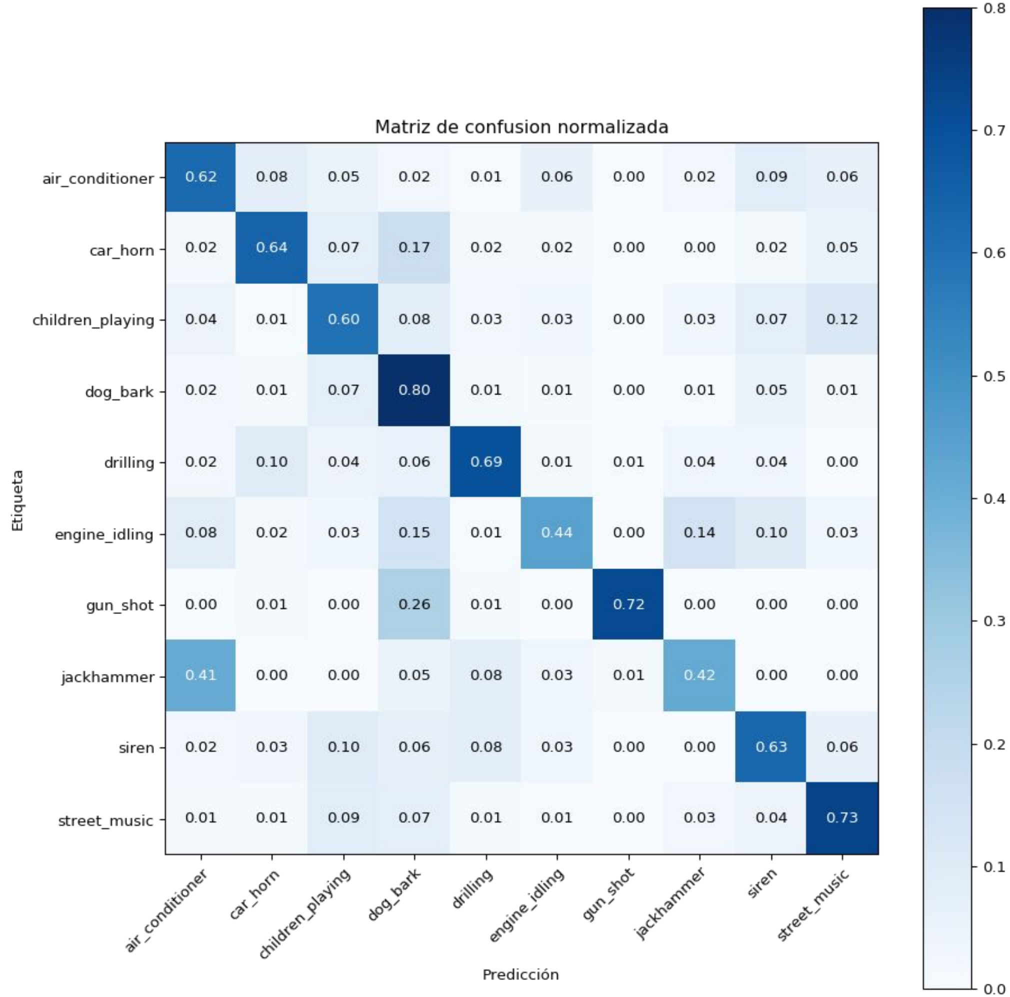
Ejercicio 3: Propuesta

- Entrenar un clasificador SVM por defecto
- Evaluar el desempeño en el **conjunto de entrenamiento** y en el **conjunto de validación**
- Estimar el desempeño del clasificador utilizando **validación cruzada**.
    - 6 *folds* por defecto
    - 6 *folds* con *ShuffleSplit*
    - 6 *folds* con *PredefinedSplit*
- Optimizar los parámetros del clasificador utilizando *GridSearch*

Ejercicio 3: matriz de confusión (kernel lineal)



Ejercicio 3: matriz de confusión (kernel RBF)



Temario

- Ejercicio 1: Formulación de C-SVM
- Ejercicio 2: Separando sonidos con características intuitivas
- Ejercicio 3: SVM en un problema de mediana escala multiclase
- Ejercicio 4: Resolviendo SVM con un paquete de optimización

Ejercicio 4: Resolviendo SVM con un paquete de optimización

El algoritmo SVM con *kernel lineal* resuelve el siguiente problema de optimización:

$$\begin{cases} \min\limits_{\mathbf{w},b} & f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{sujeto a} & \\ & y_n\left(\mathbf{w}^T\mathbf{x}_n + b\right) \geq 1 \\ & \qquad \forall n = 1, \ldots, N \end{cases}$$

En su formulación dual da lugar a la optimización

$$\begin{cases} \min\limits_{\boldsymbol{\alpha}} & \frac{1}{2}\,\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} - \mathbf{e}^{\mathbf{T}}\boldsymbol{\alpha} \\ \text{sujeto a} & \\ & \alpha_i \geq 0 \\ & \sum_i \alpha_i y_i = 0 \end{cases}$$

con $Q_{ij} = y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$

Ejercicio 4: Resolviendo SVM con un paquete de optimización

Se puede encontrar $\boldsymbol{\alpha}$ utilizando el paquete *scipy.optimize.*

Una vez encontrado $\boldsymbol{\alpha}$ se encuentran los parámetros del modelo SVM: **w**, **b**, *vectores de soporte*

Sequential Least SQuares Programming (SLSQP) Algorithm (`method='SLSQP'`)

The SLSQP method deals with constrained minimization problems of the form:

$$\begin{array}{rll} \min\limits_{x} & f(x) & \\ \text{subject to:} & c_j(x) = 0, & j \in \mathcal{E} \\ & c_j(x) \geq 0, & j \in \mathcal{I} \\ & \text{lb}_i \leq x_i \leq \text{ub}_i, \; i = 1, \ldots, N. & \end{array}$$

Where $\mathcal{E}$ or $\mathcal{I}$ are sets of indices containing equality and inequality constraints.