Problemas Multiclase

Fernando Lozano

Universidad de los Andes

10 de noviembre de 2017

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y) \ : \ \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y) \; : \; \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y) \ : \ \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:
  - Teoría:

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y) \ : \ \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:
  - Teoría:
    - Entender fundamentos.

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y) : \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:
  - Teoría:
    - Entender fundamentos.
    - Desarrollo de nuevas ideas (p.ej. boosting, SVM).

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y) \; : \; \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:
  - Teoría:
    - Entender fundamentos.
    - Desarrollo de nuevas ideas (p.ej. boosting, SVM).
  - Algoritmos:

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y)\ :\ \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:
  - Teoría:
    - Entender fundamentos.
    - Desarrollo de nuevas ideas (p.ej. boosting, SVM).
  - Algoritmos:
    - Muchos algoritmos desarrollados para caso binario.

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y)\ :\ \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:
  - Teoría:
    - Entender fundamentos.
    - Desarrollo de nuevas ideas (p.ej. boosting, SVM).
  - Algoritmos:
    - Muchos algoritmos desarrollados para caso binario.
    - Problemas de optimización más sencillos.

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y) \; : \; \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:
  - Teoría:
    - Entender fundamentos.
    - Desarrollo de nuevas ideas (p.ej. boosting, SVM).
  - Algoritmos:
    - Muchos algoritmos desarrollados para caso binario.
    - Problemas de optimización más sencillos.
- En la práctica muchos problemas son muticlase ($k > 2$):

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y) \ : \ \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:
  - Teoría:
    - Entender fundamentos.
    - Desarrollo de nuevas ideas (p.ej. boosting, SVM).
  - Algoritmos:
    - Muchos algoritmos desarrollados para caso binario.
    - Problemas de optimización más sencillos.
- En la práctica muchos problemas son muticlase ($k > 2$):
  - Reconocimiento de caracteres.

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y) \; : \; \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:
  - Teoría:
    - Entender fundamentos.
    - Desarrollo de nuevas ideas (p.ej. boosting, SVM).
  - Algoritmos:
    - Muchos algoritmos desarrollados para caso binario.
    - Problemas de optimización más sencillos.
- En la práctica muchos problemas son muticlase ($k > 2$):
  - Reconocimiento de caracteres.
  - Reconocimiento de fonemas.

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y) \ : \ \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:
  - Teoría:
    - Entender fundamentos.
    - Desarrollo de nuevas ideas (p.ej. boosting, SVM).
  - Algoritmos:
    - Muchos algoritmos desarrollados para caso binario.
    - Problemas de optimización más sencillos.
- En la práctica muchos problemas son muticlase ($k > 2$):
  - Reconocimiento de caracteres.
  - Reconocimiento de fonemas.
  - Reconocimiento de objetos

Problemas Multiclase

- Problema de clasificación: $(\mathbf{x}, y)\ :\ \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}$ con $|\mathcal{Y}| = k \geq 2$
- Caso binario ($k = 2$) es más sencillo e intuitivo:
  - Teoría:
    - Entender fundamentos.
    - Desarrollo de nuevas ideas (p.ej. boosting, SVM).
  - Algoritmos:
    - Muchos algoritmos desarrollados para caso binario.
    - Problemas de optimización más sencillos.
- En la práctica muchos problemas son muticlase ($k > 2$):
  - Reconocimiento de caracteres.
  - Reconocimiento de fonemas.
  - Reconocimiento de objetos

    $\vdots$

Cómo resolver un problema multiclase?

Cómo resolver un problema multiclase?

- Algunos métodos se pueden aplicar directamente:

Cómo resolver un problema multiclase?

- Algunos métodos se pueden aplicar directamente:
  - Red neuronal con múltiples salidas, activaciónsoftmax:

$$\sigma(\mathbf{z})_j = \frac{z_j}{\sum_i z_i}$$

Cómo resolver un problema multiclase?

- Algunos métodos se pueden aplicar directamente:
  - Red neuronal con múltiples salidas, activaciónsoftmax:

$$\sigma(\mathbf{z})_j = \frac{z_j}{\sum_i z_i}$$

  - CART
  - C4.5
  - Naive Bayes.
  - AdaBoost.MH

Cómo resolver un problema multiclase?

- Algunos métodos se pueden aplicar directamente:
  - Red neuronal con múltiples salidas, activaciónsoftmax:

$$\sigma(\mathbf{z})_j = \frac{z_j}{\sum_i z_i}$$

  - CART
  - C4.5
  - Naive Bayes.
  - AdaBoost.MH
- Utilizar métodos existentes para clasificación binaria:

Cómo resolver un problema multiclase?

- Algunos métodos se pueden aplicar directamente:
  - Red neuronal con múltiples salidas, activaciónsoftmax:

$$\sigma(\mathbf{z})_j = \frac{z_j}{\sum_i z_i}$$

  - CART
  - C4.5
  - Naive Bayes.
  - AdaBoost.MH
- Utilizar métodos existentes para clasificación binaria:
  - Convertir problema multiclase a varios problemas binarios.

Cómo resolver un problema multiclase?

- Algunos métodos se pueden aplicar directamente:
  - Red neuronal con múltiples salidas, activaciónsoftmax:

$$\sigma(\mathbf{z})_j = \frac{z_j}{\sum_i z_i}$$

  - CART
  - C4.5
  - Naive Bayes.
  - AdaBoost.MH
- Utilizar métodos existentes para clasificación binaria:
  - Convertir problema multiclase a varios problemas binarios.
  - Clasificador binario como caja negra.

Reducción a problemas binarios

Reducción a problemas binarios

- Datos: $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{n} \, : \, \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$

Reducción a problemas binarios

- Datos: $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{n} \, : \, \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$
- Procedimiento general:

Reducción a problemas binarios

- Datos: $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{n} \; : \; \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$
- Procedimiento general:
    1. Obtener problemas binarios

Reducción a problemas binarios

- Datos: $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n \ : \ \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$
- Procedimiento general:
  1. Obtener problemas binarios

$$\mathcal{S}_j = \{\mathbf{x}_i, z_i\}_{i=1}^{m_i}$$

Reducción a problemas binarios

- Datos: $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n \; : \; \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$
- Procedimiento general:
  1. Obtener problemas binarios

$$\mathcal{S}_j = \{\mathbf{x}_i, z_i\}_{i=1}^{m_i} \quad j = 1, \ldots M$$

Reducción a problemas binarios

- Datos: $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}\ :\ \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$
- Procedimiento general:
  1. Obtener problemas binarios

$$\mathcal{S}_j = \{\mathbf{x}_i, z_i\}_{i=1}^{m_i} \quad j = 1, \ldots M$$

  2. Entrenar clasificadores binarios

Reducción a problemas binarios

- Datos: $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{n} \ : \ \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$
- Procedimiento general:
  1. Obtener problemas binarios

$$\mathcal{S}_j = \{\mathbf{x}_i, z_i\}_{i=1}^{m_i} \quad j = 1, \ldots M$$

  2. Entrenar clasificadores binarios

$$\mathcal{S}_j \longrightarrow A \longrightarrow h_j$$

Reducción a problemas binarios

- Datos: $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n \; : \; \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$
- Procedimiento general:
  1. Obtener problemas binarios
  $$\mathcal{S}_j = \{\mathbf{x}_i, z_i\}_{i=1}^{m_i} \quad j = 1, \ldots M$$
  2. Entrenar clasificadores binarios
  $$\mathcal{S}_j \longrightarrow A \longrightarrow h_j$$
  3. Combinar clasificadores binarios para asignar etiquetas:

Reducción a problemas binarios

- Datos: $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n \ : \ \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$
- Procedimiento general:
  1. Obtener problemas binarios
  $$\mathcal{S}_j = \{\mathbf{x}_i, z_i\}_{i=1}^{m_i} \quad j = 1, \ldots M$$
  2. Entrenar clasificadores binarios
  $$\mathcal{S}_j \longrightarrow A \longrightarrow h_j$$
  3. Combinar clasificadores binarios para asignar etiquetas:
  $$\hat{y} = f(h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_M(\mathbf{x}))$$

Uno contra todos

Uno contra todos

- Para cada etiqueta l, se tiene un problema binario con datos $\mathcal{S}_l = \{\mathbf{x}_i, z_i\}_{i=1}^n$, y

$$z_i = \begin{cases} 1 & y_i = l \\ -1 & y_i \neq l \end{cases}$$

Uno contra todos

- Para cada etiqueta l, se tiene un problema binario con datos $\mathcal{S}_l = \{\mathbf{x}_i, z_i\}_{i=1}^n$, y

$$z_i = \begin{cases} 1 & y_i = l \\ -1 & y_i \neq l \end{cases}$$

- $M = k$ problemas binarios.

Uno contra todos

- Para cada etiqueta l, se tiene un problema binario con datos $\mathcal{S}_l = \{\mathbf{x}_i, z_i\}_{i=1}^n$, y
$$z_i = \begin{cases} 1 & y_i = l \\ -1 & y_i \neq l \end{cases}$$
- $M = k$ problemas binarios.
- Se asumen hipótesis que toman valores reales $h : \mathcal{X} \to \mathbb{R}$

Uno contra todos

- Para cada etiqueta l, se tiene un problema binario con datos $\mathcal{S}_l = \{\mathbf{x}_i, z_i\}_{i=1}^n$, y
$$z_i = \begin{cases} 1 & y_i = l \\ -1 & y_i \neq l \end{cases}$$
- $M = k$ problemas binarios.
- Se asumen hipótesis que toman valores reales $h : \mathcal{X} \to \mathbb{R}$
- Etiqueta de un nuevo dato se asigna:

Uno contra todos

- Para cada etiqueta l, se tiene un problema binario con datos $\mathcal{S}_l = \{\mathbf{x}_i, z_i\}_{i=1}^n$, y

$$z_i = \begin{cases} 1 & y_i = l \\ -1 & y_i \neq l \end{cases}$$

- $M = k$ problemas binarios.
- Se asumen hipótesis que toman valores reales $h : \mathcal{X} \to \mathbb{R}$
- Etiqueta de un nuevo dato se asigna:

$$\hat{y} = \arg\max_j h_j(\mathbf{x})$$

Uno contra todos

- Para cada etiqueta l, se tiene un problema binario con datos $\mathcal{S}_l = \{\mathbf{x}_i, z_i\}_{i=1}^n$, y

$$z_i = \begin{cases} 1 & y_i = l \\ -1 & y_i \neq l \end{cases}$$

- $M = k$ problemas binarios.
- Se asumen hipótesis que toman valores reales $h : \mathcal{X} \rightarrow \mathbb{R}$
- Etiqueta de un nuevo dato se asigna:

$$\hat{y} = \arg\max_j h_j(\mathbf{x})$$

- Problemas:
  - Escalas de h_i

Uno contra todos

- Para cada etiqueta l, se tiene un problema binario con datos $\mathcal{S}_l = \{\mathbf{x}_i, z_i\}_{i=1}^n$, y

$$z_i = \begin{cases} 1 & y_i = l \\ -1 & y_i \neq l \end{cases}$$

- $M = k$ problemas binarios.
- Se asumen hipótesis que toman valores reales $h : \mathcal{X} \rightarrow \mathbb{R}$
- Etiqueta de un nuevo dato se asigna:

$$\hat{y} = \arg\max_j h_j(\mathbf{x})$$

- Problemas:
  - Escalas de h_i
  - Problemas binarios no balanceados.

Uno contra uno

- Se construye un clasificador binario $h : \mathcal{X} \to \{1, -1\}$ para cada par de clases.

Uno contra uno

- Se construye un clasificador binario $h : \mathcal{X} \to \{1, -1\}$ para cada par de clases.
- $M = \frac{k(k-1)}{2}$ clasificadores.

Uno contra uno

- Se construye un clasificador binario $h : \mathcal{X} \to \{1, -1\}$ para cada par de clases.
- $M = \frac{k(k-1)}{2}$ clasificadores.
- Etiqueta $\hat{y}$ de un nuevo dato $\mathbf{x}$ se obtiene por votación de los M clasificadores h_i.

Uno contra uno

- Se construye un clasificador binario $h : \mathcal{X} \rightarrow \{1, -1\}$ para cada par de clases.
- $M = \frac{k(k-1)}{2}$ clasificadores.
- Etiqueta $\hat{y}$ de un nuevo dato $\mathbf{x}$ se obtiene por votación de los M clasificadores h_i.
- Problemas:
  - Número de clasificadores.
  - Ambigüedades.

Códigos de salida distribuídos (Sejnowski and Rosenberg)

Class	Code Word					
	vl	hl	dl	cc	ol	or
0	0	0	0	1	0	0
1	1	0	0	0	0	0
2	0	1	1	0	1	0
3	0	0	0	0	1	0
4	1	1	0	0	0	0
5	1	1	0	0	1	0
6	0	0	1	1	0	1
7	0	0	1	0	0	0
8	0	0	0	1	0	0
9	0	0	1	1	0	0

Column position	Abbreviation	Meaning
1	vl	contains vertical line
2	hl	contains horizontal line
3	dl	contains diagonal line
4	cc	contains closed curve
5	ol	contains curve open to left
6	or	contains curve open to right

Códigos de salida distribuídos (Sejnowski and Rosenberg)

Class	Code Word					
	vl	hl	dl	cc	ol	or
0	0	0	0	1	0	0
1	1	0	0	0	0	0
2	0	1	1	0	1	0
3	0	0	0	0	1	0
4	1	1	0	0	0	0
5	1	1	0	0	1	0
6	0	0	1	1	0	1
7	0	0	1	0	0	0
8	0	0	0	1	0	0
9	0	0	1	1	0	0

Column position	Abbreviation	Meaning
1	vl	contains vertical line
2	hl	contains horizontal line
3	dl	contains diagonal line
4	cc	contains closed curve
5	ol	contains curve open to left
6	or	contains curve open to right

- Se asigna una palabra código (string the n bits) a cada clase.

Códigos de salida distribuídos (Sejnowski and Rosenberg)

Class	Code Word					
	vl	hl	dl	cc	ol	or
0	0	0	0	1	0	0
1	1	0	0	0	0	0
2	0	1	1	0	1	0
3	0	0	0	0	1	0
4	1	1	0	0	0	0
5	1	1	0	0	1	0
6	0	0	1	1	0	1
7	0	0	1	0	0	0
8	0	0	0	1	0	0
9	0	0	1	1	0	0

Column position	Abbreviation	Meaning
1	vl	contains vertical line
2	hl	contains horizontal line
3	dl	contains diagonal line
4	cc	contains closed curve
5	ol	contains curve open to left
6	or	contains curve open to right

- Se asigna una palabra código (string the n bits) a cada clase.
- Se entrena un clasificador binario h para cada bit del código.

Códigos de salida distribuídos (Sejnowski and Rosenberg)

Class	Code Word					
	vl	hl	dl	cc	ol	or
0	0	0	0	1	0	0
1	1	0	0	0	0	0
2	0	1	1	0	1	0
3	0	0	0	0	1	0
4	1	1	0	0	0	0
5	1	1	0	0	1	0
6	0	0	1	1	0	1
7	0	0	1	0	0	0
8	0	0	0	1	0	0
9	0	0	1	1	0	0

Column position	Abbreviation	Meaning
1	vl	contains vertical line
2	hl	contains horizontal line
3	dl	contains diagonal line
4	cc	contains closed curve
5	ol	contains curve open to left
6	or	contains curve open to right

- Se asigna una palabra código (string the n bits) a cada clase.
- Se entrena un clasificador binario h para cada bit del código.
- Etiqueta de dato $\mathbf{x}$ se asigna a fila con menor distancia Hamming con $\begin{bmatrix} h_1(\mathbf{x}) & h_2(\mathbf{x}) & \dots & h_n(\mathbf{x}) \end{bmatrix}$

ECOC (output error correcting codes) (Dietterich y Bakiri)

ECOC (output error correcting codes) (Dietterich y Bakiri)

- Utiliza códigos de corrección de error para representación distribuida de las clases.

ECOC (output error correcting codes) (Dietterich y Bakiri)

- Utiliza códigos de corrección de error para representación distribuida de las clases.

$$y \longrightarrow \boxed{\text{CANAL}} \longrightarrow \hat{y}$$

ECOC (output error correcting codes) (Dietterich y Bakiri)

- Utiliza códigos de corrección de error para representación distribuida de las clases.

$$y \longrightarrow \boxed{\text{CANAL}} \longrightarrow \hat{y}$$

- Canal: representación, datos de entrenamiento, algoritmo de aprendizaje.

ECOC (output error correcting codes) (Dietterich y Bakiri)

- Utiliza códigos de corrección de error para representación distribuida de las clases.

$$y \longrightarrow \boxed{\text{CANAL}} \longrightarrow \hat{y}$$

- Canal: representación, datos de entrenamiento, algoritmo de aprendizaje.
- $\hat{y}$ es versión ruidosa de y.

ECOC (output error correcting codes) (Dietterich y Bakiri)

- Utiliza códigos de corrección de error para representación distribuida de las clases.

$$y \longrightarrow \boxed{\text{CANAL}} \longrightarrow \hat{y}$$

- Canal: representación, datos de entrenamiento, algoritmo de aprendizaje.
- $\hat{y}$ es versión ruidosa de y.
- Codigo de corrección de error permite "corregir" errores en "transmisión".

Class	Code Word														
	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
0	1	1	0	0	0	0	1	0	1	0	0	1	1	0	1
1	0	0	1	1	1	1	0	1	0	1	1	0	0	1	0
2	1	0	0	1	0	0	0	1	1	1	1	0	1	0	1
3	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1
4	1	1	1	0	1	0	1	1	0	0	1	0	0	0	1
5	0	1	0	0	1	1	0	1	1	1	0	0	0	0	1
6	1	0	1	1	1	0	0	0	0	1	0	1	0	0	1
7	0	0	0	1	1	1	1	0	1	0	1	1	0	0	1
8	1	1	0	1	0	1	1	0	0	1	0	0	0	1	1
9	0	1	1	1	0	0	0	0	1	0	1	0	0	1	1

Robustés del código

Robustés del código

- Robustés del código depende de la mínima distancia Hamming $d^* = \min_{c_1,c_2} d_H(c_1, c_2)$entre dos palabras código.

Robustés del código

- Robustés del código depende de la mínima distancia Hamming $d^* = \min_{c_1,c_2} d_H(c_1,c_2)$entre dos palabras código.
- Código puede corregir $\lfloor \frac{d^*-1}{2} \rfloor$ errores.

Robustés del código

- Robustés del código depende de la mínima distancia Hamming $d^* = \min_{c_1,c_2} d_H(c_1, c_2)$entre dos palabras código.
- Código puede corregir $\lfloor \frac{d^*-1}{2} \rfloor$ errores.
- Por ejemplo en Uno contra todos $d^* = 2$ y no hay corrección.

Robustés del código

- Robustés del código depende de la mínima distancia Hamming $d^* = \min_{c_1, c_2} d_H(c_1, c_2)$entre dos palabras código.
- Código puede corregir $\lfloor \frac{d^*-1}{2} \rfloor$ errores.
- Por ejemplo en Uno contra todos $d^* = 2$ y no hay corrección.
- d^* de códigos “naturales” tiende a ser baja.

Diseño del Código de corrección de errores

- Deseable:

Diseño del Código de corrección de errores

- Deseable:
  - Separación entre filas: $\text{mín}_{c_1,c_2} d_H(c_1, c_2) \gg$.

Diseño del Código de corrección de errores

- Deseable:
  - Separación entre filas: $\min_{c_1,c_2} d_H(c_1, c_2) \gg$.
  - Columnas no correlacionadas:

Diseño del Código de corrección de errores

- Deseable:
  - Separación entre filas: $\min_{c_1,c_2} d_H(c_1,c_2) \gg$.
  - Columnas no correlacionadas:
    - $\min_{f_1,f_2} d_H(f_1,f_2) \gg$

Diseño del Código de corrección de errores

- Deseable:
  - Separación entre filas: $\min_{c_1,c_2} d_H(c_1, c_2) \gg$.
  - Columnas no correlacionadas:
    - $\min_{f_1,f_2} d_H(f_1, f_2) \gg$
    - $\min_{f_1,f_2} d_H(\bar{f}_1, f_2) \gg$

Diseño del Código de corrección de errores

- Deseable:
  - Separación entre filas: $\min_{c_1,c_2} d_H(c_1, c_2) \gg$.
  - Columnas no correlacionadas:
    - $\min_{f_1,f_2} d_H(f_1, f_2) \gg$
    - $\min_{f_1,f_2} d_H(\bar{f}_1, f_2) \gg$
  - Ninguna columna con sólo ceros o sólo unos.

Diseño del Código de corrección de errores

- Deseable:
  - Separación entre filas: $\min_{c_1,c_2} d_H(c_1, c_2) \gg$.
  - Columnas no correlacionadas:
    - $\min_{f_1,f_2} d_H(f_1, f_2) \gg$
    - $\min_{f_1,f_2} d_H(\bar{f}_1, f_2) \gg$
  - Ninguna columna con sólo ceros o sólo unos.
- Número de columnas utilizables: $2^{k-1} - 1$.

Diseño del Código de corrección de errores

- Deseable:
  - Separación entre filas: $\min_{c_1,c_2} d_H(c_1, c_2) \gg$.
  - Columnas no correlacionadas:
    - $\min_{f_1,f_2} d_H(f_1, f_2) \gg$
    - $\min_{f_1,f_2} d_H(\bar{f}_1, f_2) \gg$
  - Ninguna columna con sólo ceros o sólo unos.
- Número de columnas utilizables: $2^{k-1} - 1$.
- Es difícil lograr condiciones para k pequeño.

Método randomizado

Método randomizado

- Generar k códigos de longitud M aletoriamente.

Método randomizado

- Generar k códigos de longitud M aletoriamente.
- $d_H(c_1, c_2)$ es distribuída binomialmente con media $L/2$.

Método randomizado

- Generar k códigos de longitud M aletoriamente.
- $d_H(c_1, c_2)$ es distribuída binomialmente con media $L/2$.
- Mejoramiento por Hill-Climbing:

Método randomizado

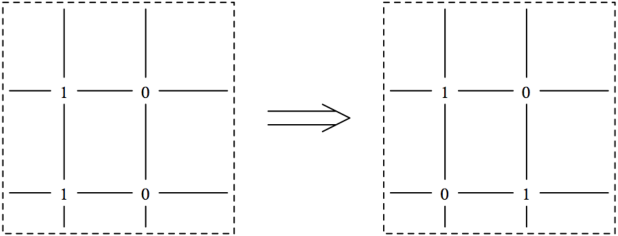
- Generar k códigos de longitud M aletoriamente.
- $d_H(c_1, c_2)$ es distribuída binomialmente con media $L/2$.
- Mejoramiento por Hill-Climbing:
  - Encontrar intersección entre el par de filas más cercanas y de columnas más lejanas.

Método randomizado

- Generar k códigos de longitud M aletoriamente.
- $d_H(c_1, c_2)$ es distribuída binomialmente con media $L/2$.
- Mejoramiento por Hill-Climbing:
  - Encontrar intersección entre el par de filas más cercanas y de columnas más lejanas.
  - Modificar bits para mejorar separación:

Método randomizado

- Generar k códigos de longitud M aletoriamente.
- $d_H(c_1, c_2)$ es distribuída binomialmente con media $L/2$.
- Mejoramiento por Hill-Climbing:
  - Encontrar intersección entre el par de filas más cercanas y de columnas más lejanas.
  - Modificar bits para mejorar separación:



Clasificadores con márgen

- Hipótesis reales $f : \mathcal{X} \to \mathbb{R}$.

Clasificadores con márgen

- Hipótesis reales $f : \mathcal{X} \to \mathbb{R}$.
- $\mathbf{x}$ es clasificado correctamente por f si $yf(\mathbf{x}) \geq 0$ y el márgen $yf(\mathbf{x})$ indica confianza en clasificación.

Clasificadores con márgen

- Hipótesis reales $f : \mathcal{X} \rightarrow \mathbb{R}$.
- $\mathbf{x}$ es clasificado correctamente por f si $yf(\mathbf{x}) \geq 0$ y el márgen $yf(\mathbf{x})$ indica confianza en clasificación.
- Clasificadores basados en márgen minimizan función de costo del márgen en los datos:

Clasificadores con márgen

- Hipótesis reales $f : \mathcal{X} \to \mathbb{R}$.
- $\mathbf{x}$ es clasificado correctamente por f si $yf(\mathbf{x}) \geq 0$ y el márgen $yf(\mathbf{x})$ indica confianza en clasificación.
- Clasificadores basados en márgen minimizan función de costo del márgen en los datos:

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i f(\mathbf{x}_i))$$

Clasificadores con márgen

- Hipótesis reales $f : \mathcal{X} \rightarrow \mathbb{R}$.
- $\mathbf{x}$ es clasificado correctamente por f si $yf(\mathbf{x}) \geq 0$ y el márgen $yf(\mathbf{x})$ indica confianza en clasificación.
- Clasificadores basados en márgen minimizan función de costo del márgen en los datos:

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i f(\mathbf{x}_i))$$

- Por ejemplo:

Clasificadores con márgen

- Hipótesis reales $f : \mathcal{X} \to \mathbb{R}$.
- $\mathbf{x}$ es clasificado correctamente por f si $yf(\mathbf{x}) \geq 0$ y el márgen $yf(\mathbf{x})$ indica confianza en clasificación.
- Clasificadores basados en márgen minimizan función de costo del márgen en los datos:

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i f(\mathbf{x}_i))$$

- Por ejemplo:

  Adaboost: $L(z) =$

Clasificadores con márgen

- Hipótesis reales $f : \mathcal{X} \to \mathbb{R}$.
- $\mathbf{x}$ es clasificado correctamente por f si $yf(\mathbf{x}) \geq 0$ y el márgen $yf(\mathbf{x})$ indica confianza en clasificación.
- Clasificadores basados en márgen minimizan función de costo del márgen en los datos:

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i f(\mathbf{x}_i))$$

- Por ejemplo:

  Adaboost: $L(z) = e^{-z}$

Clasificadores con márgen

- Hipótesis reales $f : \mathcal{X} \to \mathbb{R}$.
- $\mathbf{x}$ es clasificado correctamente por f si $yf(\mathbf{x}) \geq 0$ y el márgen $yf(\mathbf{x})$ indica confianza en clasificación.
- Clasificadores basados en márgen minimizan función de costo del márgen en los datos:

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i f(\mathbf{x}_i))$$

- Por ejemplo:

$$\begin{aligned} \text{Adaboost:}\quad & L(z) = e^{-z} \\ \text{SVM:}\quad & L(z) = \max(0, 1 - z) \end{aligned}$$

Clasificadores con márgen

- Hipótesis reales $f : \mathcal{X} \to \mathbb{R}$.
- $\mathbf{x}$ es clasificado correctamente por f si $yf(\mathbf{x}) \geq 0$ y el márgen $yf(\mathbf{x})$ indica confianza en clasificación.
- Clasificadores basados en márgen minimizan función de costo del márgen en los datos:

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i f(\mathbf{x}_i))$$

- Por ejemplo:

$$\begin{aligned} \text{Adaboost:} \quad & L(z) = e^{-z} \\ \text{SVM:} \quad & L(z) = \max(0, 1 - z) \end{aligned}$$

Clasificadores con márgen

- Hipótesis reales $f : \mathcal{X} \rightarrow \mathbb{R}$.
- $\mathbf{x}$ es clasificado correctamente por f si $yf(\mathbf{x}) \geq 0$ y el márgen $yf(\mathbf{x})$ indica confianza en clasificación.
- Clasificadores basados en márgen minimizan función de costo del márgen en los datos:

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i f(\mathbf{x}_i))$$

- Por ejemplo:

$$\begin{aligned} \text{Adaboost:} \quad & L(z) = e^{-z} \\ \text{SVM:} \quad & L(z) = \max(0, 1 - z) \\ \text{Backprop:} \quad & L(z) = (1 - z)^2 \end{aligned}$$

Clasificación muticlase con márgen (Allwein, Schapire and Singer)

Clasificación muticlase con márgen (Allwein, Schapire and Singer)

- Matriz de codificación:

$$\mathbf{M} \in \{-1, 0, 1\}^{k \times l}$$

Clasificación muticlase con márgen (Allwein, Schapire and Singer)

- Matriz de codificación:

$$\mathbf{M} \in \{-1, 0, 1\}^{k \times l}$$

- Algoritmo $\mathcal{A}$ recibe datos, produce clasificador minimizando función de costo de márgenes en los datos.

Clasificación muticlase con márgen (Allwein, Schapire and Singer)

- Matriz de codificación:

$$\mathbf{M} \in \{-1, 0, 1\}^{k \times l}$$

- Algoritmo $\mathcal{A}$ recibe datos, produce clasificador minimizando función de costo de márgenes en los datos.
- Para cada fila $s = 1, 2, \ldots l$ $\mathcal{A}$ encuentra un clasificador $f_s : \mathcal{X} \to \mathbb{R}$ con datos $\{\mathbf{x}_i, \mathbf{M}(y_i, s)\}$, ignorando datos con $\mathbf{M}(y, s) = 0$.

Clasificación muticlase con márgen (Allwein, Schapire and Singer)

- Matriz de codificación:

$$\mathbf{M} \in \{-1, 0, 1\}^{k \times l}$$

- Algoritmo $\mathcal{A}$ recibe datos, produce clasificador minimizando función de costo de márgenes en los datos.
- Para cada fila $s = 1, 2, \ldots l$ $\mathcal{A}$ encuentra un clasificador $f_s : \mathcal{X} \to \mathbb{R}$ con datos $\{\mathbf{x}_i, \mathbf{M}(y_i, s)\}$, ignorando datos con $\mathbf{M}(y, s) = 0$.

- Etiqueta de un dato $\mathbf{x}$:

- Etiqueta de un dato $\mathbf{x}$:
- Sea $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) & f_2(\mathbf{x}) & \dots & f_l(\mathbf{x}) & \end{bmatrix}$
  - Decodificación Hamming:

- Etiqueta de un dato $\mathbf{x}$:
- Sea $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) & f_2(\mathbf{x}) & \dots & f_l(\mathbf{x}) \end{bmatrix}$
  - Decodificación Hamming:

$$d_H(\mathbf{M}_r, \mathbf{f}(\mathbf{x})) = \sum_{i=1}^{l} \left(\frac{1 - \mathrm{sign}(\mathbf{M}_r f_i(\mathbf{x}))}{2} \right)$$

- Etiqueta de un dato $\mathbf{x}$:
- Sea $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) & f_2(\mathbf{x}) & \dots & f_l(\mathbf{x}) \end{bmatrix}$
  - Decodificación Hamming:

$$d_H(\mathbf{M}_r, \mathbf{f}(\mathbf{x})) = \sum_{i=1}^{l} \left(\frac{1 - \operatorname{sign}(\mathbf{M}_r f_i(\mathbf{x}))}{2} \right)$$

$$\hat{y} = \arg\min_r d_H(\mathbf{M}_r, \mathbf{f}(\mathbf{x}))$$

- Etiqueta de un dato $\mathbf{x}$:
- Sea $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) & f_2(\mathbf{x}) & \dots & f_l(\mathbf{x}) \end{bmatrix}$
  - Decodificación Hamming:

$$d_H(\mathbf{M}_r, \mathbf{f}(\mathbf{x})) = \sum_{i=1}^{l} \left(\frac{1 - \operatorname{sign}(\mathbf{M}_r f_i(\mathbf{x}))}{2} \right)$$

$$\hat{y} = \arg\min_r d_H(\mathbf{M}_r, \mathbf{f}(\mathbf{x}))$$

  - Decodificación basada en la función de pérdida L:

- Etiqueta de un dato $\mathbf{x}$:
- Sea $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) & f_2(\mathbf{x}) & \dots & f_l(\mathbf{x}) \end{bmatrix}$
  - Decodificación Hamming:

$$d_H(\mathbf{M}_r, \mathbf{f}(\mathbf{x})) = \sum_{i=1}^{l} \left(\frac{1 - \operatorname{sign}(\mathbf{M}_r f_i(\mathbf{x}))}{2} \right)$$

$$\hat{y} = \arg\min_r d_H(\mathbf{M}_r, \mathbf{f}(\mathbf{x}))$$

  - Decodificación basada en la función de pérdida L:

$$d_L(\mathbf{M}_r, \mathbf{f}(\mathbf{x})) = \sum_{s=1}^{l} L(\mathbf{M}_{rs} f_s(\mathbf{x}))$$

- Etiqueta de un dato $\mathbf{x}$:
- Sea $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) & f_2(\mathbf{x}) & \dots & f_l(\mathbf{x}) \end{bmatrix}$
    - Decodificación Hamming:

$$d_H(\mathbf{M}_r, \mathbf{f}(\mathbf{x})) = \sum_{i=1}^{l} \left(\frac{1 - \operatorname{sign}(\mathbf{M}_r f_i(\mathbf{x}))}{2} \right)$$

$$\hat{y} = \arg\min_r d_H(\mathbf{M}_r, \mathbf{f}(\mathbf{x}))$$

    - Decodificación basada en la función de pérdida L:

$$d_L(\mathbf{M}_r, \mathbf{f}(\mathbf{x})) = \sum_{s=1}^{l} L(\mathbf{M}_{rs} f_s(\mathbf{x}))$$

$$\hat{y} = \arg\min_r d_L(\mathbf{M}_r, \mathbf{f}(\mathbf{x}))$$

- Etiqueta de un dato $\mathbf{x}$:
- Sea $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) & f_2(\mathbf{x}) & \dots & f_l(\mathbf{x}) \end{bmatrix}$
  - Decodificación Hamming:

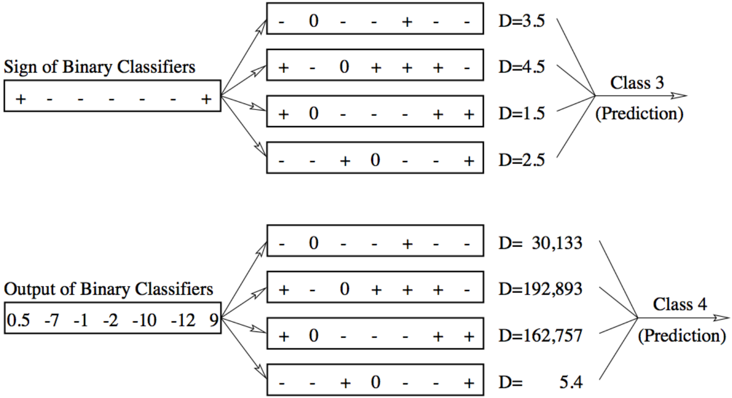
$$d_H(\mathbf{M}_r, \mathbf{f}(\mathbf{x})) = \sum_{i=1}^{l} \left(\frac{1 - \operatorname{sign}(\mathbf{M}_r f_i(\mathbf{x}))}{2} \right)$$

$$\hat{y} = \arg\min_r d_H(\mathbf{M}_r, \mathbf{f}(\mathbf{x}))$$

  - Decodificación basada en la función de pérdida L:

$$d_L(\mathbf{M}_r, \mathbf{f}(\mathbf{x})) = \sum_{s=1}^{l} L(\mathbf{M}_{rs} f_s(\mathbf{x}))$$

$$\hat{y} = \arg\min_r d_L(\mathbf{M}_r, \mathbf{f}(\mathbf{x}))$$



Sign of Binary Classifiers
+ - - - - - +
- 0 - - + - -
D=3.5
+ - 0 + + + -
D=4.5
+ 0 - - - + +
D=1.5
- - + 0 - - +
D=2.5
Class 3
(Prediction)
Output of Binary Classifiers
0.5 -7 -1 -2 -10 -12 9
- 0 - - + - -
D= 30,133
+ - 0 + + + -
D=192,893
+ 0 - - - + +
D=162,757
- - + 0 - - +
D= 5.4
Class 4
(Prediction)

Error de entrenamiento

Error de entrenamiento

- Para $\mathbf{u}, \mathbf{v} \in \{-1, 0, 1\}^l$, sea $\Delta(\mathbf{u}, \mathbf{v}) = \frac{l - \mathbf{u}^T \mathbf{v}}{2}$

Error de entrenamiento

- Para $\mathbf{u}, \mathbf{v} \in \{-1, 0, 1\}^l$, sea $\Delta(\mathbf{u}, \mathbf{v}) = \frac{l - \mathbf{u}^T \mathbf{v}}{2}$
- Sea la distancia mínima entre filas de $\mathbf{M}$:

$$\rho = \min \{\Delta(M_{r_1}, M_{r_2}) \ : \ r_1 \neq r_2\}$$

Error de entrenamiento

- Para $\mathbf{u}, \mathbf{v} \in \{-1, 0, 1\}^l$, sea $\Delta(\mathbf{u}, \mathbf{v}) = \frac{l - \mathbf{u}^T \mathbf{v}}{2}$
- Sea la distancia mínima entre filas de $\mathbf{M}$:

$$\rho = \min \{ \Delta(M_{r_1}, M_{r_2}) \ : \ r_1 \neq r_2 \}$$

- Sea la pérdida binaria promedio de las hipótesis f_s con respecto a $\mathbf{M}$:

$$\frac{1}{ml} \sum_{i=1}^{m} \sum_{s=1}^{l} L(\mathbf{M}(y_i, s) f_s(\mathbf{x}_i))$$

Teorema

Sea ϵ la pérdida binaria promedio de las hipótesis $f_1, f_2, \ldots, f_l$ en los datos $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^m$ con respecto a $\mathbf{M}$ y L. Asuma que L satisface $\frac{L(z)+L(-z)}{2} \geq L(0) > 0$, entonces el error de entrenamiento usando decodificación basada en función de pérdida es a lo sumo

$$\frac{l\epsilon}{\rho L(0)}$$

SVM multiclase 1 (Weston and Watkins)

SVM multiclase 1 (Weston and Watkins)

- Caso binario (separador lineal con máximo márgen):

$$
\begin{aligned}
\text{mín} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\zeta_i \\
\text{sujeto a} \quad & y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n \\
& \zeta_i \geq 0
\end{aligned}
$$

SVM multiclase 1 (Weston and Watkins)

- Caso binario (separador lineal con máximo márgen):

$$
\begin{aligned}
\text{mín} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\zeta_i \\
\text{sujeto a} \quad & y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) - 1 + \zeta_i \geq 0 \quad i = 1, \dots, n \\
& \zeta_i \geq 0
\end{aligned}
$$

- Resulta en el problema dual

$$
\begin{aligned}
\text{máx} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n}\alpha_i\alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle \\
\text{sujeto a} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0 \\
& 0 \leq \alpha_i \leq C
\end{aligned}
$$

SVM multiclase 1 (Weston and Watkins)

- Caso binario (separador lineal con máximo márgen):

$$\begin{aligned} \text{mín} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\zeta_i \\ \text{sujeto a} \quad & y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n \\ & \zeta_i \geq 0 \end{aligned}$$

- Resulta en el problema dual

$$\begin{aligned} \text{máx} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n}\alpha_i\alpha_j y_i y_j \left\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\right\rangle_{\mathcal{H}} \\ \text{sujeto a} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \end{aligned}$$

SVM multiclase 1 (Weston and Watkins)

- Caso binario (separador lineal con máximo márgen):

$$
\begin{aligned}
\text{mín} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\zeta_i \\
\text{sujeto a} \quad & y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n \\
& \zeta_i \geq 0
\end{aligned}
$$

- Resulta en el problema dual

$$
\begin{aligned}
\text{máx} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\
\text{sujeto a} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0 \\
& 0 \leq \alpha_i \leq C
\end{aligned}
$$

- Con k clases:

$$
\begin{aligned}
\text{mín} \quad & \frac{1}{2}\sum_{i=1}^{k}\|\mathbf{w}_k\|^2 + {\color{red}C}\sum_{i=1}^{n}\sum_{m\neq y_i}\zeta_i^m \\
\text{sujeto a} \quad & \langle \mathbf{w}_{y_i}, \mathbf{x}_i\rangle + b_{y_i} \geq \langle \mathbf{w}_m, \mathbf{x}_i\rangle + b_m + 2 - \zeta_i^m \\
& \zeta_i^m \geq 0 \\
& i = 1, \ldots, n \\
& m \in \{1, \ldots, k\} - \{y_i\}
\end{aligned}
$$

SVM multiclase 2 (Cramer and Singer)

- Hipótesis de SVM binario:

$$
\begin{aligned}
\mathbf{w}^T\mathbf{x} \geq 0 \Rightarrow y &= 1 \qquad & \mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y = 1 \\
\mathbf{w}^T\mathbf{x} \leq 0 \Rightarrow y &= -1 \qquad & -\mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y = -1
\end{aligned}
$$

SVM multiclase 2 (Cramer and Singer)

- Hipótesis de SVM binario:

$$
\begin{aligned}
\mathbf{w}^T\mathbf{x} \geq 0 \Rightarrow y &= 1 \qquad & \mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y = 1 \\
\mathbf{w}^T\mathbf{x} \leq 0 \Rightarrow y &= -1 \qquad & -\mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y = -1
\end{aligned}
$$

$$
\mathbf{M} = \begin{bmatrix} \mathbf{w}^T \\ -\mathbf{w}^T \end{bmatrix}
$$

SVM multiclase 2 (Cramer and Singer)

- Hipótesis de SVM binario:

$$\begin{aligned} \mathbf{w}^T\mathbf{x} \geq 0 \Rightarrow y &= 1 & \qquad \mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y = 1 \\ \mathbf{w}^T\mathbf{x} \leq 0 \Rightarrow y &= -1 & \qquad -\mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y = -1 \end{aligned}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{w}^T \\ -\mathbf{w}^T \end{bmatrix} \Rightarrow y = \arg\max_r \{\mathbf{M}_r \mathbf{x}\}$$

SVM multiclase 2 (Cramer and Singer)

- Hipótesis de SVM binario:

$$\begin{aligned} \mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y &= 1 \qquad\qquad \mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y = 1 \\ \mathbf{w}^T\mathbf{x} \leq 0 &\Rightarrow y &= -1 \qquad -\mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y = -1 \end{aligned}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{w}^T \\ -\mathbf{w}^T \end{bmatrix} \Rightarrow y = \arg\max_r \{\mathbf{M}_r\mathbf{x}\}$$

- Problema multiclase: $\mathbf{M} \in \mathbb{R}^{k\times d}$,

SVM multiclase 2 (Cramer and Singer)

- Hipótesis de SVM binario:

$$
\begin{aligned}
\mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y \;\; = 1 \qquad & \mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y = 1 \\
\mathbf{w}^T\mathbf{x} \leq 0 &\Rightarrow y = -1 \qquad & -\mathbf{w}^T\mathbf{x} \geq 0 &\Rightarrow y = -1
\end{aligned}
$$

$$
\mathbf{M} = \begin{bmatrix} \mathbf{w}^T \\ -\mathbf{w}^T \end{bmatrix} \Rightarrow y = \arg\max_r \{\mathbf{M}_r \mathbf{x}\}
$$

- Problema multiclase: $\mathbf{M} \in \mathbb{R}^{k \times d}$,

$$
H_{\mathbf{M}}(\mathbf{x}) = \arg\max_r \{\mathbf{M}_r \mathbf{x}\}
$$

- Para un conjunto de datos $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{n} \ : \ \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$, el error empírico de $\mathbf{M}$ es:

$$\hat{e}_S(\mathbf{M}) = \frac{1}{n} \sum_{i=1}^{n} [\![H_M(\mathbf{x}_i) \neq y_i]\!]$$

- Para un conjunto de datos $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{n} : \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$, el error empírico de $\mathbf{M}$ es:

$$\hat{e}_S(\mathbf{M}) = \frac{1}{n} \sum_{i=1}^{n} [\![H_M(\mathbf{x}_i) \neq y_i]\!]$$

- Meta es encontrar $\mathbf{M}$ con $\hat{e}_S(\mathbf{M}) \ll$

- Para un conjunto de datos $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n : \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$, el error empírico de $\mathbf{M}$ es:

$$\hat{e}_S(\mathbf{M}) = \frac{1}{n} \sum_{i=1}^{n} [\![H_M(\mathbf{x}_i) \neq y_i]\!]$$

- Meta es encontrar $\mathbf{M}$ con $\hat{e}_S(\mathbf{M}) \ll$
- no tratable computacionalmente.

- Para un conjunto de datos $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{n} \,:\, \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$, el error empírico de $\mathbf{M}$ es:

$$\hat{e}_S(\mathbf{M}) = \frac{1}{n}\sum_{i=1}^{n} [\![H_M(\mathbf{x}_i) \neq y_i]\!]$$

- Meta es encontrar $\mathbf{M}$ con $\hat{e}_S(\mathbf{M}) \ll$
- no tratable computacionalmente.
- Se acota $\hat{e}_S(\mathbf{M})$ con función lineal a trozos

- Para un conjunto de datos $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^{n} \; : \; \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$ con $|\mathcal{Y}| = k > 2$, el error empírico de $\mathbf{M}$ es:

$$\hat{e}_S(\mathbf{M}) = \frac{1}{n} \sum_{i=1}^{n} [\![H_M(\mathbf{x}_i) \neq y_i]\!]$$

- Meta es encontrar $\mathbf{M}$ con $\hat{e}_S(\mathbf{M}) \ll$
- no tratable computacionalmente.
- Se acota $\hat{e}_S(\mathbf{M})$ con función lineal a trozos $\Rightarrow$ márgen.

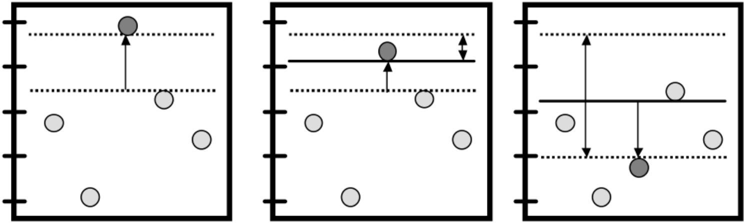
Márgen multiclase

Márgen multiclase

$$[\![H_M(\mathbf{x}) \neq y]\!] \leq \max_r \{\mathbf{M}_r\mathbf{x} + 1 - \delta_{y,r}\} - \mathbf{M}_y\mathbf{x}$$

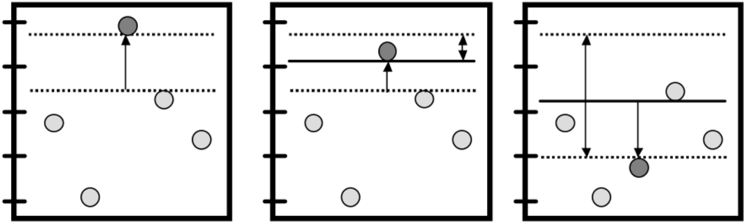
Márgen multiclase

$$[\![H_M(\mathbf{x}) \neq y]\!] \leq \max_r \{\mathbf{M}_r\mathbf{x} + 1 - \delta_{y,r}\} - \mathbf{M}_y\mathbf{x}$$



Márgen multiclase

$$\sum_{i=1}^{n} [\![H_M(\mathbf{x}_i) \neq y_i]\!] \leq \sum_{i=1}^{n} \max_r \{\mathbf{M}_r \mathbf{x}_i + 1 - \delta_{y_i,r}\} - \mathbf{M}_{y_i} \mathbf{x}_i$$



- $\mathcal{S}$ es linealmente separable por una máquina multiclase si $\exists$ una matriz $\mathbf{M}$ tal que:

- $\mathcal{S}$ es linealmente separable por una máquina multiclase si $\exists$ una matriz $\mathbf{M}$ tal que:

$$\max_r \{\mathbf{M}_r \mathbf{x}_i + 1 - \delta_{y_i,r}\} - \mathbf{M}_{y_i} \mathbf{x}_i = 0 \quad \forall i$$

- $\mathcal{S}$ es linealmente separable por una máquina multiclase si $\exists$ una matriz $\mathbf{M}$ tal que:

$$\max_r \{\mathbf{M}_r\mathbf{x}_i + 1 - \delta_{y_i,r}\} - \mathbf{M}_{y_i}\mathbf{x}_i = 0 \quad \forall i$$

o equivalentemente, $\exists\mathbf{M}$ que satisface las restricciones:

$$\mathbf{M}_{y_i}\mathbf{x}_i + \delta_{y_i,r} - \mathbf{M}_r\mathbf{x}_i \geq 1 \quad \forall i, r$$

- $\mathcal{S}$ es linealmente separable por una máquina multiclase si $\exists$ una matriz $\mathbf{M}$ tal que:

$$\max_r \{\mathbf{M}_r \mathbf{x}_i + 1 - \delta_{y_i,r}\} - \mathbf{M}_{y_i} \mathbf{x}_i = 0 \quad \forall i$$

  o equivalentemente, $\exists \mathbf{M}$ que satisface las restricciones:

$$\mathbf{M}_{y_i} \mathbf{x}_i + \delta_{y_i,r} - \mathbf{M}_r \mathbf{x}_i \geq 1 \quad \forall i, r$$

- Regularización:

- $\mathcal{S}$ es linealmente separable por una máquina multiclase si $\exists$ una matriz $\mathbf{M}$ tal que:

$$\max_r \{\mathbf{M}_r \mathbf{x}_i + 1 - \delta_{y_i,r}\} - \mathbf{M}_{y_i}\mathbf{x}_i = 0 \quad \forall i$$

  o equivalentemente, $\exists \mathbf{M}$ que satisface las restricciones:

$$\mathbf{M}_{y_i}\mathbf{x}_i + \delta_{y_i,r} - \mathbf{M}_r\mathbf{x}_i \geq 1 \quad \forall i, r$$

- Regularización:

$$\|\mathbf{M}\|^2 = \sum_{i,j} M_{ij}^2$$

- Problema de optimización:

$$\begin{aligned} \text{mín} \quad & \frac{1}{2}\|\mathbf{M}\|^2 \\ \text{sujeto a} \quad & \mathbf{M}_{y_i}\mathbf{x}_i + \delta_{y_i,r} - \mathbf{M}_r\mathbf{x}_i \geq 1 \quad \forall i, r \end{aligned}$$

- Problema de optimización:

$$
\begin{aligned}
\text{mín} \quad & \frac{1}{2}\|\mathbf{M}\|^2 \\
\text{sujeto a} \quad & \mathbf{M}_{y_i}\mathbf{x}_i + \delta_{y_i,r} - \mathbf{M}_r\mathbf{x}_i \geq 1 \quad \forall i, r
\end{aligned}
$$

- Problema de programación cuadrática con $k \times n$ restricciones.

- Problema de optimización:

$$\begin{aligned} \text{mín} \quad & \frac{1}{2}\|\mathbf{M}\|^2 \\ \text{sujeto a} \quad & \mathbf{M}_{y_i}\mathbf{x}_i + \delta_{y_i,r} - \mathbf{M}_r\mathbf{x}_i \geq 1 \quad \forall i, r \end{aligned}$$

- Problema de programación cuadrática con $k \times n$ restricciones.
- Restricciones para $r = y_i$ automáticamente satisfechas.

- En el caso no separable, se introducen variables $\xi_i \geq 0$

- En el caso no separable, se introducen variables $\xi_i \geq 0$

$$\max_r \{\mathbf{M}_r \mathbf{x}_i + 1 - \delta_{y_i,r}\} - \mathbf{M}_{y_i} \mathbf{x}_i = \xi_i \quad \forall i$$

- En el caso no separable, se introducen variables $\xi_i \geq 0$

$$\max_r \{\mathbf{M}_r \mathbf{x}_i + 1 - \delta_{y_i,r}\} - \mathbf{M}_{y_i} \mathbf{x}_i = \xi_i \quad \forall i$$

- Problema de optimización:

$$\begin{aligned} \min_{\mathbf{M},\xi_i} \quad & \frac{1}{2}\beta\|\mathbf{M}\|^2 + \sum_{i=1}^{n} \xi_i \\ \text{sujeto a} \quad & \mathbf{M}_{y_i} \mathbf{x}_i + \delta_{y_i,r} - \mathbf{M}_r \mathbf{x}_i \geq 1 - \xi_i \quad \forall i, r \end{aligned}$$

- En el caso no separable, se introducen variables $\xi_i \geq 0$

$$\max_r \{\mathbf{M}_r \mathbf{x}_i + 1 - \delta_{y_i,r}\} - \mathbf{M}_{y_i}\mathbf{x}_i = \xi_i \quad \forall i$$

- Problema de optimización:

$$\begin{aligned} \min_{\mathbf{M},\xi_i} \quad & \frac{1}{2}\beta\|\mathbf{M}\|^2 + \sum_{i=1}^{n} \xi_i \\ \text{sujeto a} \quad & \mathbf{M}_{y_i}\mathbf{x}_i + \delta_{y_i,r} - \mathbf{M}_r\mathbf{x}_i \geq 1 - \xi_i \quad \forall i, r \end{aligned}$$

- Restricciones para $r = y_i$ equivalen a $\xi_i \geq 0$

- El Lagrangiano ($\eta_{i,r} \geq 0$):

- El Lagrangiano ($\eta_{i,r} \geq 0$):

$$\mathcal{L}(\mathbf{M}, \boldsymbol{\xi}, \boldsymbol{\eta}) = \frac{1}{2}\beta \sum_r \|\mathbf{M}_r\|^2 + \sum_{i=1}^{n} \xi_i + \sum_{i,r} \eta_{i,r} \left[\mathbf{M}_r \mathbf{x}_i - \mathbf{M}_{y_i} \mathbf{x}_i - \delta_{y_i,r} + 1 - \xi_i\right]$$

- El Lagrangiano ($\eta_{i,r} \geq 0$):

$$\mathcal{L}(\mathbf{M}, \boldsymbol{\xi}, \boldsymbol{\eta}) = \frac{1}{2}\beta \sum_r \|\mathbf{M}_r\|^2 + \sum_{i=1}^n \xi_i$$
$$+ \sum_{i,r} \eta_{i,r} \left[\mathbf{M}_r \mathbf{x}_i - \mathbf{M}_{y_i} \mathbf{x}_i - \delta_{y_i,r} + 1 - \xi_i\right]$$

- Derivando con respecto a ξ_i:

$$\frac{\partial}{\partial \xi_i}\mathcal{L} = 1 - \sum_r \eta_{i,r} = 0$$

- El Lagrangiano ($\eta_{i,r} \geq 0$):

$$\mathcal{L}(\mathbf{M}, \boldsymbol{\xi}, \boldsymbol{\eta}) = \frac{1}{2}\beta \sum_r \|\mathbf{M}_r\|^2 + \sum_{i=1}^{n} \xi_i + \sum_{i,r} \eta_{i,r} \left[\mathbf{M}_r \mathbf{x}_i - \mathbf{M}_{y_i} \mathbf{x}_i - \delta_{y_i,r} + 1 - \xi_i\right]$$

- Derivando con respecto a ξ_i:

$$\frac{\partial}{\partial \xi_i}\mathcal{L} = 1 - \sum_r \eta_{i,r} = 0 \Rightarrow \sum_r \eta_{i,r} = 1$$

- Derivando con respecto a $\mathbf{M}_r$:

- Derivando con respecto a $\mathbf{M}_r$:

$$\frac{\partial}{\partial \mathbf{M}_r}\mathcal{L} = \sum_i \eta_{i,r}\mathbf{x}_i - \sum_{i,y_i=r}\left(\sum_q \eta_{i,q}\right)\mathbf{x}_i + \beta \mathbf{M}_r$$

- Derivando con respecto a $\mathbf{M}_r$:

$$\frac{\partial}{\partial \mathbf{M}_r}\mathcal{L} = \sum_i \eta_{i,r}\mathbf{x}_i - \sum_{i,y_i=r}\left(\sum_q \eta_{i,q}\right)\mathbf{x}_i + \beta\mathbf{M}_r$$

$$= \sum_i \eta_{i,r}\mathbf{x}_i - \sum_i \delta_{y_i,r}\mathbf{x}_i + \beta\mathbf{M}_r = 0$$

- Derivando con respecto a $\mathbf{M}_r$:

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{M}_r}\mathcal{L} &= \sum_i \eta_{i,r}\mathbf{x}_i - \sum_{i,y_i=r}\left(\sum_q \eta_{i,q}\right)\mathbf{x}_i + \beta\mathbf{M}_r \\
&= \sum_i \eta_{i,r}\mathbf{x}_i - \sum_i \delta_{y_i,r}\mathbf{x}_i + \beta\mathbf{M}_r = 0 \\
&\Rightarrow \mathbf{M}_r = \frac{1}{\beta}\sum_i (\delta_{y_i,r} - \eta_{i,r})\mathbf{x}_i
\end{aligned}
$$

- Derivando con respecto a $\mathbf{M}_r$:

$$\begin{aligned}\frac{\partial}{\partial \mathbf{M}_r}\mathcal{L} &= \sum_i \eta_{i,r}\mathbf{x}_i - \sum_{i,y_i=r}\left(\sum_q \eta_{i,q}\right)\mathbf{x}_i + \beta\mathbf{M}_r \\ &= \sum_i \eta_{i,r}\mathbf{x}_i - \sum_i \delta_{y_i,r}\mathbf{x}_i + \beta\mathbf{M}_r = 0 \\ &\Rightarrow \mathbf{M}_r = \frac{1}{\beta}\sum_i(\delta_{y_i,r} - \eta_{i,r})\mathbf{x}_i\end{aligned}$$

- Cada fila es una combinación lineal de los $\mathbf{x}_i$.

- Derivando con respecto a $\mathbf{M}_r$:

$$\begin{aligned}\frac{\partial}{\partial \mathbf{M}_r}\mathcal{L} &= \sum_i \eta_{i,r}\mathbf{x}_i - \sum_{i,y_i=r}\left(\sum_q \eta_{i,q}\right)\mathbf{x}_i + \beta\mathbf{M}_r \\ &= \sum_i \eta_{i,r}\mathbf{x}_i - \sum_i \delta_{y_i,r}\mathbf{x}_i + \beta\mathbf{M}_r = 0 \\ &\Rightarrow \mathbf{M}_r = \frac{1}{\beta}\sum_i(\delta_{y_i,r} - \eta_{i,r})\mathbf{x}_i\end{aligned}$$

- Cada fila es una combinación lineal de los $\mathbf{x}_i$.
- $\mathbf{x}_i$ es un vector de soporte si hay una fila r para la cual $(\delta_{y_i,r} - \eta_{i,r}) \neq 0$

$$\mathbf{M}_r = \frac{1}{\beta}\left[\sum_{i\,:\,y_i=r}(1-\eta_{i,r})\mathbf{x}_i + \sum_{i\,:\,y_i\neq r}(-\eta_{i,r})\mathbf{x}_i\right]$$

$$\mathbf{M}_r = \frac{1}{\beta}\left[\sum_{i\,:\,y_i=r}(1-\eta_{i,r})\mathbf{x}_i + \sum_{i\,:\,y_i\neq r}(-\eta_{i,r})\mathbf{x}_i\right]$$

- Un vector $\mathbf{x}_i$ de la clase $y_i = r$ es un vector de soporte si $\eta_{i,r} = \eta_{i,y_i} < 1$

$$\mathbf{M}_r = \frac{1}{\beta}\left[\sum_{i\,:\,y_i=r}(1-\eta_{i,r})\mathbf{x}_i + \sum_{i\,:\,y_i\neq r}(-\eta_{i,r})\mathbf{x}_i\right]$$

- Un vector $\mathbf{x}_i$ de la clase $y_i = r$ es un vector de soporte si $\eta_{i,r} = \eta_{i,y_i} < 1$
- Un vector con etiqueta *diferente* a $y_i = r$ es un vector de soporte si $\eta_{i,r} > 0$

$$\mathbf{M}_r = \frac{1}{\beta}\left[\sum_{i\,:\,y_i=r}(1-\eta_{i,r})\mathbf{x}_i + \sum_{i\,:\,y_i\neq r}(-\eta_{i,r})\mathbf{x}_i\right]$$

- Un vector $\mathbf{x}_i$ de la clase $y_i = r$ es un vector de soporte si $\eta_{i,r} = \eta_{i,y_i} < 1$
- Un vector con etiqueta *diferente* a $y_i = r$ es un vector de soporte si $\eta_{i,r} > 0$
- Para un dato $\mathbf{x}_i$ se puede interpretar $\{\eta_{i,1}, \eta_{i,2}, \ldots, \eta_{i,k}\}$ como una distribución de probabilidad sobre las etiquetas $1, 2, \ldots, k$.

$$\mathbf{M}_r = \frac{1}{\beta}\left[\sum_{i\,:\,y_i=r}(1-\eta_{i,r})\mathbf{x}_i + \sum_{i\,:\,y_i\neq r}(-\eta_{i,r})\mathbf{x}_i\right]$$

- Un vector $\mathbf{x}_i$ de la clase $y_i = r$ es un vector de soporte si $\eta_{i,r} = \eta_{i,y_i} < 1$
- Un vector con etiqueta *diferente* a $y_i = r$ es un vector de soporte si $\eta_{i,r} > 0$
- Para un dato $\mathbf{x}_i$ se puede interpretar $\{\eta_{i,1}, \eta_{i,2}, \ldots, \eta_{i,k}\}$ como una distribución de probabilidad sobre las etiquetas $1, 2, \ldots, k$.
- Luego $\mathbf{x}_i$ es un vector de soporte si y sólo si su distribución no está concentrada en su etiqueta y_i.

$$\mathbf{M}_r = \frac{1}{\beta}\left[\sum_{i\,:\,y_i=r}(1-\eta_{i,r})\mathbf{x}_i + \sum_{i\,:\,y_i\neq r}(-\eta_{i,r})\mathbf{x}_i\right]$$

- Un vector $\mathbf{x}_i$ de la clase $y_i = r$ es un vector de soporte si $\eta_{i,r} = \eta_{i,y_i} < 1$
- Un vector con etiqueta *diferente* a $y_i = r$ es un vector de soporte si $\eta_{i,r} > 0$
- Para un dato $\mathbf{x}_i$ se puede interpretar $\{\eta_{i,1}, \eta_{i,2}, \ldots, \eta_{i,k}\}$ como una distribución de probabilidad sobre las etiquetas $1, 2, \ldots, k$.
- Luego $\mathbf{x}_i$ es un vector de soporte si y sólo si su distribución no está concentrada en su etiqueta y_i.
- Clasificador $\mathbf{M}$ se construye con datos con etiquetas inciertas.

- Reemplazando en $\mathcal{L}$ se obtiene la función dual:

$$\mathcal{G}(\boldsymbol{\eta}) = -\frac{1}{2\beta}\sum_{i,j}\langle \mathbf{x}_i, \mathbf{x}_j\rangle \left[\sum_r (\delta_{y_i,r} - \eta_{i,r})(\delta_{y_i,r} - \eta_{i,r})\right] - \sum_{i,r}\eta_{i,r}\delta_{y_i,r}$$

- Reemplazando en $\mathcal{L}$ se obtiene la función dual:

$$\mathcal{G}(\boldsymbol{\eta}) = -\frac{1}{2\beta}\sum_{i,j}\langle \mathbf{x}_i, \mathbf{x}_j\rangle \left[\sum_r (\delta_{y_i,r} - \eta_{i,r})(\delta_{y_i,r} - \eta_{i,r})\right] - \sum_{i,r}\eta_{i,r}\delta_{y_i,r}$$

$$= -\frac{1}{2\beta}\sum_{i,j}\langle \mathbf{x}_i, \mathbf{x}_j\rangle \left\langle \mathbf{e}_{y_i} - \boldsymbol{\eta}_i, \mathbf{e}_{y_j} - \boldsymbol{\eta}_j\right\rangle - \sum_i \langle \boldsymbol{\eta}_i, \mathbf{e}_{y_i}\rangle$$

- Reemplazando en $\mathcal{L}$ se obtiene la función dual:

$$\begin{aligned}\mathcal{G}(\boldsymbol{\eta}) &= -\frac{1}{2\beta}\sum_{i,j}\langle \mathbf{x}_i, \mathbf{x}_j\rangle \left[\sum_r (\delta_{y_i,r} - \eta_{i,r})(\delta_{y_i,r} - \eta_{i,r})\right] - \sum_{i,r}\eta_{i,r}\delta_{y_i,r} \\ &= -\frac{1}{2\beta}\sum_{i,j}\langle \mathbf{x}_i, \mathbf{x}_j\rangle \left\langle \mathbf{e}_{y_i} - \boldsymbol{\eta}_i, \mathbf{e}_{y_j} - \boldsymbol{\eta}_j\right\rangle - \sum_i \langle \boldsymbol{\eta}_i, \mathbf{e}_{y_i}\rangle\end{aligned}$$

- Con el cambio de variable $\boldsymbol{\tau}_i = \mathbf{e}_{y_i} - \boldsymbol{\eta}_i$ el problema dual es:

$$\begin{aligned}\text{máx} \quad & \mathcal{G}(\boldsymbol{\tau}) = -\frac{1}{2}\sum_{i,j}\langle \mathbf{x}_i, \mathbf{x}_j\rangle \langle \boldsymbol{\tau}_i, \boldsymbol{\tau}_j\rangle + \beta\sum_i \langle \boldsymbol{\tau}_i, \mathbf{e}_{y_i}\rangle \\ \text{sujeto a} \quad & \boldsymbol{\tau}_i \leq \mathbf{e}_{y_i} \quad \forall i \\ & \langle \boldsymbol{\tau}_i, \mathbf{1}\rangle = 1 \quad \forall i\end{aligned}$$

Truco del kernel

Truco del kernel

$$\begin{aligned}\text{máx} \quad & \mathcal{G}(\boldsymbol{\tau}) = -\frac{1}{2}\sum_{i,j} \langle \mathbf{x}_i, \mathbf{x}_j \rangle \langle \boldsymbol{\tau}_i, \boldsymbol{\tau}_j \rangle + \beta \sum_i \langle \boldsymbol{\tau}_i, \mathbf{e}_{y_i} \rangle \\ \text{sujeto a} \quad & \boldsymbol{\tau}_i \leq \mathbf{e}_{y_i} \quad \forall i \\ & \langle \boldsymbol{\tau}_i, \mathbf{1} \rangle = 1 \quad \forall i\end{aligned}$$

$$H_{\mathbf{M}}(\mathbf{x}) = \arg\max_r \{\mathbf{M}_r \mathbf{x}\} = \arg\max_r \left\{ \sum_i \tau_{i,r} \langle \mathbf{x}_i, \mathbf{x} \rangle \right\}$$

Truco del kernel

$$\begin{aligned} \text{máx} \quad & \mathcal{G}(\boldsymbol{\tau}) = -\frac{1}{2}\sum_{i,j} \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle_{\mathcal{H}} \langle \boldsymbol{\tau}_i, \boldsymbol{\tau}_j\rangle + \beta \sum_i \langle \boldsymbol{\tau}_i, \mathbf{e}_{y_i}\rangle \\ \text{sujeto a} \quad & \boldsymbol{\tau}_i \leq \mathbf{e}_{y_i} \quad \forall i \\ & \langle \boldsymbol{\tau}_i, \mathbf{1}\rangle = 1 \quad \forall i \end{aligned}$$

$$H_{\mathbf{M}}(\mathbf{x}) = \arg\max_r \{\mathbf{M}_r \mathbf{x}\} = \arg\max_r \left\{ \sum_i \tau_{i,r} \langle \phi(\mathbf{x}_i), \phi(\mathbf{x})\rangle_{\mathcal{H}} \right\}$$

Truco del kernel

$$\begin{aligned} \text{máx} \quad & \mathcal{G}(\boldsymbol{\tau}) = -\frac{1}{2}\sum_{i,j} k(\mathbf{x}_i, \mathbf{x}_j) \langle \boldsymbol{\tau}_i, \boldsymbol{\tau}_j \rangle + \beta \sum_i \langle \boldsymbol{\tau}_i, \mathbf{e}_{y_i} \rangle \\ \text{sujeto a} \quad & \boldsymbol{\tau}_i \leq \mathbf{e}_{y_i} \quad \forall i \\ & \langle \boldsymbol{\tau}_i, \mathbf{1} \rangle = 1 \quad \forall i \end{aligned}$$

$$H_{\mathbf{M}}(\mathbf{x}) = \arg\max_r \{\mathbf{M}_r \mathbf{x}\} = \arg\max_r \left\{ \sum_i \tau_{i,r} k(\mathbf{x}_i, \mathbf{x}) \right\}$$