Advanced Kernel Methods for Multi-Task Learning

Tesis dirigida por José Dorronsoro y Carlos Alaíz

Carlos Ruiz Pastor

January 9, 2023

Acknowledgements 1

Acknowledgements 2

Universidad Autónoma de Madrid

Outline

o

Table of Contents

- El Aprendizaje Automático intenta automatizar el proceso de aprendizaje
- En el aprendizaje supervisado tenemos:
  - un espacio de entrada $\mathcal{X}$,
  - un espacio de salida $\mathcal{Y}$,
  - y una distribución $P(x, y)$ (desconocida) sobre $\mathcal{X} \times \mathcal{Y}$
- Dada una función $f : \mathcal{X} \to \mathcal{Y}$, definimos una función de pérdida como

$$
\begin{aligned}
\ell : \mathcal{Y} \times \mathcal{Y} &\to [0, \infty) \\
(y, f(x)) &\to \ell(y, f(x)),
\end{aligned}
$$

tal que $\ell(y, y) = 0$ para todo $y \in \mathcal{Y}$

- In classification, with the class labels $y_i \in \{-1, 1\}$, we can use:

$$\ell(y, f(x)) = [1 - yf(x)]_+ = \begin{cases} 0, & yf(x) \geq 1, \\ 1 - yf(x), & yf(x) < 1. \end{cases}$$

-

Expected Risk

- Given a space of hypothesis $\mathcal{H} = \{h\left(\cdot, \alpha\right), \alpha \in A\}$
- Definition: Expected Risk

$$R_P(\alpha) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, h\left(x, \alpha\right))dP(x, y)$$

- Our goal is to find

$$\alpha^* = \arg\min_{\alpha\in A} \left\{ R_P(\alpha) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, h\left(x, \alpha\right))dP(x, y) \right\},$$

however the distribution $P(x, y)$ is unknown

Empirical Risk

- Instead, we have a set of n instances sampled from $P(x, y)$:

$$D_n = \{(x_i, y_i),\ i = 1, \dots, n\},$$

- Definition: Empirical Risk

$$\hat{R}_{D_n}(\alpha) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i, \alpha))$$

- Instead of the Expected Risk, we minimize this empirical risk:

$$\arg\min_{\alpha \in A} \left\{ \hat{R}_D(\alpha) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i, \alpha)) \right\}$$

Table of Contents

Table of Contents

Table of Contents

Formulación Aditiva

- Una manera de implementar el MTL es combinar una parte común y otras específicas
- La formulación aditiva para el aprendizaje multitarea es

$$h_r(\cdot) = g(\cdot) + g_r(\cdot)$$

  donde
  - $g(\cdot)$ es la parte común
  - $g_r(\cdot)$ es la parte específica
- Fue propuesta para SVM lineales con los modelos

$$h_r(\cdot) = \langle w + v_r, \cdot \rangle + b_r$$

Universidad Autónoma de Madrid

Formulación Convexa

- Proponemos la siguiente formulación convexa para el aprendizaje multitarea:

$$h_r(\cdot) = \lambda_r g(\cdot) + (1 - \lambda_r) g_r(\cdot),$$

  con $\lambda_r \in [0, 1]$.
- Los hiperparámetros λ_r regulan la influencia de cada parte:
  - $\lambda_1, \ldots, \lambda_T = 0$: modelos independientes (ITL)
  - $\lambda_1, \ldots, \lambda_T = 1$: modelo común (CTL)
- La interpretación de los hiperparámetros es más sencilla

Table of Contents

Formulacion Convexa con Métodos de Kernel

- La formulación aditiva con métodos de kernel puede expresarse con los modelos:

$$h_r(\cdot) = \{\langle w, \phi(\cdot)\rangle + b\} + \{\langle v_r, \phi_r(\cdot)\rangle + d_r\}$$

- Con nuestra formulación convexa los modelos son:

$$h_r(\cdot) = \lambda_r \{\langle w, \phi(\cdot)\rangle + b\} + (1 - \lambda_r) \{\langle v_r, \phi_r(\cdot)\rangle + d_r\}$$

- Desarrollamos tres variantes de SVM:
  - L1-SVM
  - L2-SVM
  - LS-SVM

Formulación Aditiva para MTL L1-SVM

Problema Primal - L1-SVM Aditiva

$$\begin{aligned} \underset{w, \boldsymbol{v}, \boldsymbol{b}, \xi}{\arg\min} \quad & J(w, \boldsymbol{v}, \boldsymbol{b}, \xi) = C \sum_{r=1}^{T} \sum_{i=1}^{m_r} \xi_i^r + \frac{1}{2} \sum_{r=1}^{T} \|v_r\|^2 + \frac{\mu}{2} \|w\|^2 \\ \text{s.t.} \quad & y_i^r (\langle w, \phi(x_i^r) \rangle + \langle v_r, \phi_r(x_i^r) \rangle + b_r) \geq p_i^r - \xi_i^r, \\ & \xi_i^r \geq 0; \ i = 1, \dots, m_r, \ r = 1, \dots, T. \end{aligned}$$

- El parámetro μ (junto con C) regula la influencia de cada parte:
  - $\mu \to \infty$: modelos independientes (ITL)
  - $C \to 0, \ \mu \to 0$: modelo común (CTL)

Formulación Aditiva para MTL L1-SVM

Problema Dual - L1-SVM Aditiva

$$\begin{aligned}\min_{\boldsymbol{\alpha}} \quad & \Theta(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^\intercal \left(\frac{1}{\mu}Q + K\right)\boldsymbol{\alpha} - \boldsymbol{p}\boldsymbol{\alpha} \\ \text{s.t.} \quad & 0 \leq \alpha_i^r \leq C;\ i = 1, \dots, m_r,\ \ r = 1, \dots, T, \\ & \sum_{i=1}^{m_r} \alpha_i^r y_i^r = 0;\ r = 1, \dots, T.\end{aligned}$$

- El parámetro μ (junto con C) regula la influencia de cada parte:
  - $\mu \to \infty$: modelos independientes (ITL)
  - $C \to 0,\ \mu \to 0$: modelo común (CTL)

Problema Primal - L1-SVM Convexa

$$
\begin{aligned}
\min_{w, \boldsymbol{v}, b, \boldsymbol{d}, \boldsymbol{\xi}} \quad & J(w, \boldsymbol{v}, b, \boldsymbol{d}, \boldsymbol{\xi}) = C \sum_{r=1}^{T} \sum_{i=1}^{m_r} \xi_i^r + \frac{1}{2} \sum_{r=1}^{T} \|v_r\|^2 + \frac{1}{2} \|w\|^2 \\
\text{s.t.} \quad & y_i^r \left(\lambda_r \left\{\langle w, \phi(x_i^r)\rangle + b\right\} + (1 - \lambda_r) \left\{\langle v_r, \phi_r(x_i^r)\rangle + d_r\right\}\right) \geq p_i^r - \xi_i^r, \\
& \xi_i^r \geq 0, \; i = 1, \dots, m_r, \; r = 1, \dots, T.
\end{aligned}
$$

- Los hiperparámetros λ_r regulan la influencia de cada parte:
  - $\lambda_1, \dots, \lambda_T = 0$: modelos independientes (ITL)
  - $\lambda_1, \dots, \lambda_T = 1$: modelo común (CTL)
- El hiperparámetro C no interviene en la definición de los modelos

Formulación Convexa para MT L1-SVM

Problema Dual - SVM Convexa

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \Theta(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^\intercal \left(\Lambda Q \Lambda + (I_n - \Lambda)\, K \,(I_n - \Lambda)\right) \boldsymbol{\alpha} - \boldsymbol{p}\boldsymbol{\alpha} \\ \text{s.t.} \quad & 0 \leq \alpha_i^r \leq C; \; i = 1, \dots, m_r, \; r = 1, \dots, T, \\ & \sum_{i=1}^{m_r} \alpha_i^r y_i^r = 0; \; r = 1, \dots, T, \end{aligned}$$

donde

$$\Lambda = \operatorname{diag}(\overbrace{\lambda_1, \dots, \lambda_1}^{m_1}, \dots, \overbrace{\lambda_T, \dots, \lambda_T}^{m_T})$$

Formulación Convexa para MT L1-SVM

Problema Dual - SVM Convexa (λ común)

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \Theta(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^\intercal \left(\lambda^2 Q + (1-\lambda)^2 K\right)\boldsymbol{\alpha} - \boldsymbol{p}\boldsymbol{\alpha} \\
\text{s.t.} \quad & 0 \leq \alpha_i^r \leq C;\ i = 1, \dots, m_r,\ r = 1, \dots, T, \\
& \sum_{i=1}^{m_r} \alpha_i^r y_i^r = 0;\ r = 1, \dots, T,
\end{aligned}
$$

- El hiperparámetro λ regula la influencia de cada parte:
  - $\lambda = 0$: modelos independientes (ITL)
  - $\lambda = 1$: modelo común (CTL)
- El hiperparámetro C no interviene en la definición de los modelos

Proposiciones

Proposicion (Equivalencia entre formulaciones para SVM)

Para valores $\lambda \in (0, 1)$, la formulación aditiva con hiperparámetros C_{add}, μ y la formulación convexa con C_{conv} y un λ común, $\lambda_1, \dots, \lambda_T = \lambda$, son equivalentes cuando

$$C_{add} = (1 - \lambda)^2 C_{conv}, \ \mu = (1 - \lambda)^2 / \lambda^2 .$$

Proposicion (Equivalencia con CTL e ITL)

- *Para $\lambda = 0$, la formulación convexa con un λ común es equivalente a modelos independientes (ITL).*
- *Para $\lambda = 1$ la formulación convexa con un λ común es equivalente a un modelo común (CTL).*

Formulación convexa para MT L2-SVM

Problema Primal - MTL L2-SVM Convexa

$$\begin{aligned} \underset{w,\boldsymbol{v},b,\boldsymbol{d},\xi}{\arg\min} \quad & J(w, \boldsymbol{v}, b, \boldsymbol{d}, \boldsymbol{\xi}) = \frac{C}{2}\sum_{r=1}^{T}\sum_{i=1}^{m_r}(\xi_i^r)^2 + \frac{1}{2}\sum_{r=1}^{T}\|v_r\|^2 + \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i^r\left(\lambda_r\left\{\langle w, \phi(x_i^r)\rangle + b\right\} + (1-\lambda_r)\left\{\langle v_r, \phi_r(x_i^r)\rangle + d_r\right\}\right) \geq p_i^r - \xi_i^r, \end{aligned}$$

Problema Dual - MTL L2-SVM Convexa

$$
\begin{aligned}
\min_{\alpha} \quad & \Theta(\alpha) = \frac{1}{2}\boldsymbol{\alpha}^\intercal \left(\{\Lambda Q \Lambda + (I_n - \Lambda) K (I_n - \Lambda)\} + \frac{1}{C} I \right) \boldsymbol{\alpha} - \boldsymbol{p}\boldsymbol{\alpha} \\
\text{s.t.} \quad & 0 \le \alpha_i^r, \; i = 1, \dots, m_r, \; r = 1, \dots, T, \\
& \sum_{i=1}^{m_r} \alpha_i^r y_i^r = 0, \; r = 1, \dots, T.
\end{aligned}
$$

Formulación convexa para MT LS-SVM

Problema Primal - MTL LS-SVM Convexa

$$\arg\min_{w, \boldsymbol{v}, b, \boldsymbol{d}, \xi} \quad J(w, \boldsymbol{v}, b, \boldsymbol{d}, \boldsymbol{\xi}) = \frac{C}{2} \sum_{r=1}^{T} \sum_{i=1}^{m_r} (\xi_i^r)^2 + \frac{1}{2} \sum_{r=1}^{T} \|v_r\|^2 + \frac{1}{2} \|w\|^2$$

$$\text{s.t.} \quad y_i^r \left(\lambda_r \left\{\langle w, \phi(x_i^r)\rangle + b\right\} + (1 - \lambda_r) \left\{\langle v_r, \phi_r(x_i^r)\rangle + d_r\right\}\right) = p_i^r - \xi_i^r,$$

Formulación Convexa para MT LS-SVM

Problema Dual - MTL LS-SVM Convexa

$$\left[\begin{array}{c|c|c} 0 & \mathbf{0}_T^\intercal & \boldsymbol{y}^\intercal \Lambda \\ \hline \mathbf{0}_T & \mathbf{0}_{T\times T} & A^\intercal Y (I_n - \Lambda) \\ \hline \boldsymbol{y} & YA & \widehat{Q} + \frac{1}{C} I_n \end{array}\right] \begin{pmatrix} b \\ d_1 \\ \vdots \\ d_T \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{0}_T \\ \boldsymbol{p} \end{pmatrix},$$

Table of Contents

- La manera más común de adaptar las redes neuronales es el *hard sharing*
  - Capas ocultas compartidas por todas las tareas
  - Capas de salida específicas para cada tarea
- El modelo se puede expresar como:

$$h_r(\cdot) = g_r(\cdot; w_r, \Theta) = \{\langle w_r, f(\cdot; \Theta)\rangle\} + d_r$$

  - w_r, d_r son los parámetros de las capas de salida específicas
  - Θ son los parámetros de las capas ocultas compartidas

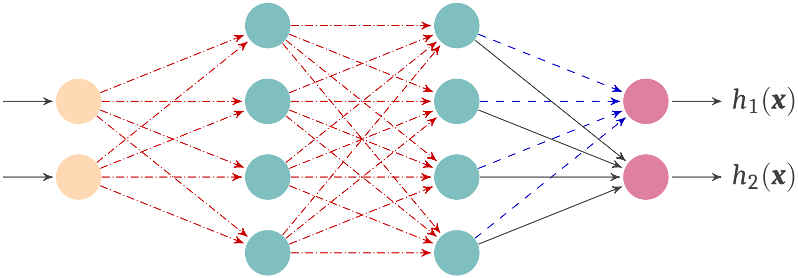


Figure: Ejemplo de *Hard Sharing* para dos tareas .

Formulación Convexa para Redes Neuronales MT

- Proponemos la formulación convexa para redes neuronales MT, combinando:
  - Una parte común $g(\cdot; w, \Theta)$
  - Una parte específica $g_r(\cdot; w_r, \Theta_r)$
- Los modelos son:

$$\begin{aligned} h_r(\cdot) &= \lambda_r g(\cdot; w, \Theta) + (1 - \lambda_r) g_r(\cdot; w_r, \Theta_r) \\ &= \lambda_r \{ \langle w, f(\cdot; \Theta) \rangle + b \} + (1 - \lambda_r) \{ \langle w_r, f_r(\cdot; \Theta_r) \rangle + d_r \}. \end{aligned}$$

  - w, Θ son los parámetros de la red común (capa de salida y ocultas)
  - w_r, Θ_r son los parámetros de las redes específicas (capa de salida y ocultas)

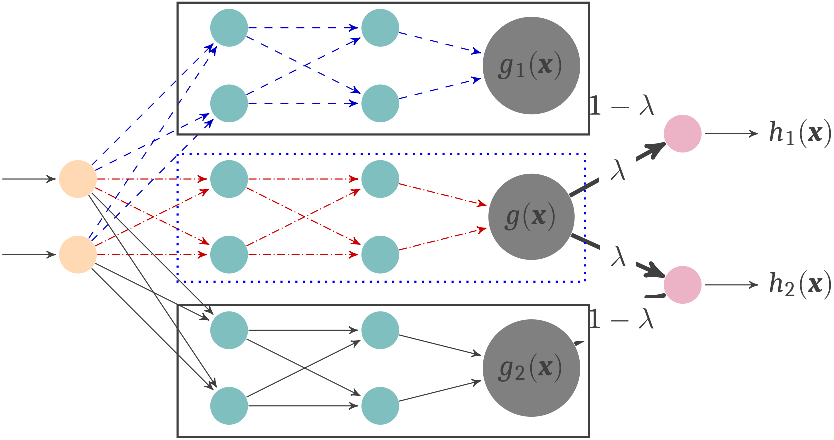


Figure: Ejemplo de formulación convexa con redes neuronales para dos tareas.

- El riesgo a minimizar en este caso es

$$\hat{R}_D = \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(h_r(x_i^r), y_i^r) + \frac{\mu}{2} \left(\|w\|^2 + \sum_{r=1}^{T} \|w_r\|^2 + \Omega(\Theta) + \Omega(\Theta_r) \right) .$$

- Se puede aplicar el descenso por gradiente con

$$\begin{aligned} \nabla_w h_t(x_i^t) &= \lambda_t f(x_i^t, \Theta), & \nabla_\Theta h_t(x_i^t) &= \lambda_t \left\langle w, \nabla_\Theta f(x_i^t, \Theta) \right\rangle ; \\ \nabla_{w_t} h_t(x_i^t) &= (1 - \lambda_t) f_t(x_i^t, \Theta), & \nabla_{\Theta_t} h_t(x_i^t) &= (1 - \lambda_t) \left\langle w, \nabla_{\Theta_t} f_t(x_i^t, \Theta_t) \right\rangle ; \\ \nabla_{w_r} h_t(x_i^t) &= 0, & \nabla_{\Theta_r} h_t(x_i^t) &= 0, \text{ for } r \neq t. \end{aligned}$$

- Los gradientes se escalan adecuadamente con λ_t y $(1 - \lambda_t)$

Algorithm 1: Pase "forward"

Input: $X_{\text{mb}}, t_{\text{mb}}$ `// Minibatch data and task labels`
Output: f `// Forward pass for the minibatch`
Data: λ `// Parameter of convex combination`
Data: $g; g_1, \dots, g_T$ `// Modules of the common and specific networks`
for $x_i, t_i \in (X_{mb}, t_{mb})$ **do**
  $f_i \leftarrow \lambda g(x_i) + (1 - \lambda) g_{t_i}(x_i)$ `// Convex combination`
end

- El pase "backward" se hace con la diferenciación automática de `PyTorch`

Table of Contents

Combinación Convexa de modelos Preentrenados

- Alternativa a la formulación convexa para aprendizaje MT
- Consideramos la combinación convexa de
  - modelo común $g(\cdot)$ entrenado
  - modelos específicos $g_r(\cdot)$ entrenados
- Minimizamos el riesgo eligiendo los hiperparámetros $\lambda_1, \dots, \lambda_T$ óptimos

$$\hat{R}_D(\lambda_1, \dots, \lambda_T) = \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(\lambda_r g(x_i^r) + (1 - \lambda_r) g_r(x_i^r), y_i^r),$$

- Hinge loss (classification):

$$\hat{R}_D(\lambda_1, \dots, \lambda_T) = \sum_{r=1}^{T} \sum_{i=1}^{m_r} \left[1 - y_i^r \left\{\lambda_r g(x_i^r) + (1 - \lambda_r) g_r(x_i^r)\right\}\right]_+ .$$

- Squared hinge loss (classification):

$$\hat{R}_D(\lambda_1, \dots, \lambda_T) = \sum_{r=1}^{T} \sum_{i=1}^{m_r} \left[1 - y_i^r \left\{\lambda_r g(x_i^r) + (1 - \lambda_r) g_r(x_i^r)\right\}\right]_+^2 .$$

- Ambas se pueden expresar como:

$$\sum_{r=1}^{T} \sum_{i=1}^{m_r} u(\lambda_r c_i^r + d_i^r), \text{ donde } c_i^r = y_i^r (g_r(x_i^r) - g(x_i^r)), \; d_i^r = 1 - y_i^r g_r(x_i^r)$$

- Absolute loss (regression):

$$\hat{R}_D(\lambda_1, \dots, \lambda_T) = \sum_{r=1}^{T} \sum_{i=1}^{m_r} \left| y_i^r - \{\lambda_r g(x_i^r) + (1 - \lambda_r) g_r(x_i^r)\} \right| .$$

- Squared loss (regression):

$$\hat{R}_D(\lambda_1, \dots, \lambda_T) = \sum_{r=1}^{T} \sum_{i=1}^{m_r} \left(y_i^r - \{\lambda_r g(x_i^r) + (1 - \lambda_r) g_r(x_i^r)\} \right)^2 .$$

- Ambas se pueden expresar como:

$$\sum_{r=1}^{T} \sum_{i=1}^{m_r} u(\lambda_r c_i^r + d_i^r), \ \text{donde } c_i^r = g(x_i^r) - g_r(x_i^r), \ d_i^r = g_r(x_i^r) - y_i^r$$

Formulación Unificada

- En todos los casos tenemos que minimizar

$$\hat{R}_D(\lambda_1, \ldots, \lambda_T) = \sum_{r=1}^{T} \sum_{i=1}^{m_r} u(\lambda_r c_i^r + d_i^r)$$

- Como es separable, tenemos en cada tarea el problema

$$\underset{\lambda_r \in [0,1]}{\arg\min}\, \mathcal{J}(\lambda_r) = \sum_{i=1}^{m_r} u(\lambda_r c_i^r + d_i^r),$$

- Usando el Teorema de Fermat

$$\lambda^* = \underset{0 \leq \lambda \leq 1}{\arg\min}\, \mathcal{J}(\lambda) \iff (0 \in \partial \mathcal{J}(\lambda^*) \text{ and } \lambda^* \in (0,1)) \text{ or } \lambda^* = 0 \text{ or } \lambda^* = 1.$$

Combinación Convexa con Error Cuadrático

- La función a minimizar es

$$\arg\min_{\lambda \in [0,1]} \mathcal{J}(\lambda) = \sum_{i=1}^{m} \left(\lambda c_i + d_i\right)^2 .$$

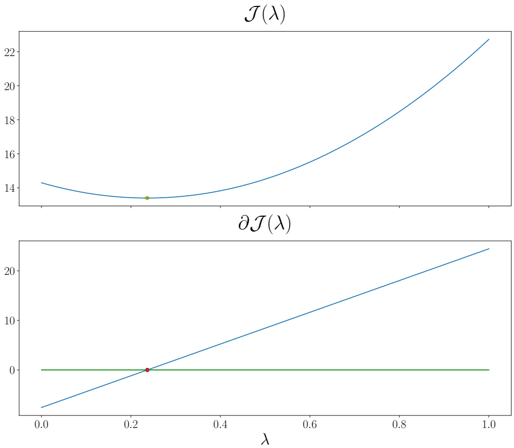
- La derivada es

$$\mathcal{J}'(\lambda) = \sum_{i=1}^{m} 2c_i(\lambda c_i + d_i).$$

- Como es derivable, resolviendo $\mathcal{J}'(\lambda) = 0$ obtenemos

$$\lambda' = -\frac{\sum_{i=1}^{m} d_i c_i}{\sum_{i=1}^{m} (c_i)^2}.$$

- La solución es entonces $\lambda^* = \max(\min(\lambda', 1), 0)$



$\mathcal{J}(\lambda)$
22
20
18
16
14
$\partial\mathcal{J}(\lambda)$
20
10
0
0.0
0.2
0.4
0.6
0.8
1.0
λ

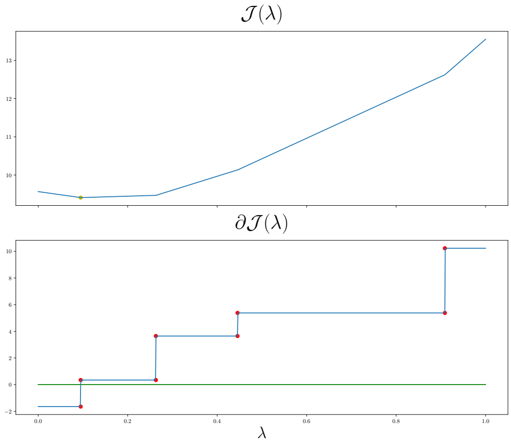
Combinación Convexa con Error Absoluto

Proposicion (λ^* óptimo para el problema con valor absoluto)

- $\lambda^* = 0$ *es óptimo si y solo si:* $-\sum_{i:\ 0>\lambda_{(i)}} \left|c_{(i)}\right| + \sum_{i:\ 0<\lambda_{(i)}} \left|c_{(i)}\right| \leq 0$
- $\lambda^* \in (0, 1)$ *es óptimo si y solo si* $0 < \lambda^* = \lambda_{(k)} < 1$ *para algún* $k = 1, \dots, m$, *y*

$$- \sum_{i:\ \lambda_{(k)} > \lambda_{(i)}} \left|c_{(i)}\right| + \sum_{i:\ \lambda_{(k)} < \lambda_{(i)}} \left|c_{(i)}\right| \in \left[- \left|c_{(k)}\right|, \left|c_{(k)}\right|\right]$$

- $\lambda^* = 1$ *es óptimo en otro caso*


$\mathcal{J}(\lambda)$
13
12
11
10
$\partial\mathcal{J}(\lambda)$
10
8
6
4
2
0
−2
0.0
0.2
0.4
0.6
0.8
1.0
λ

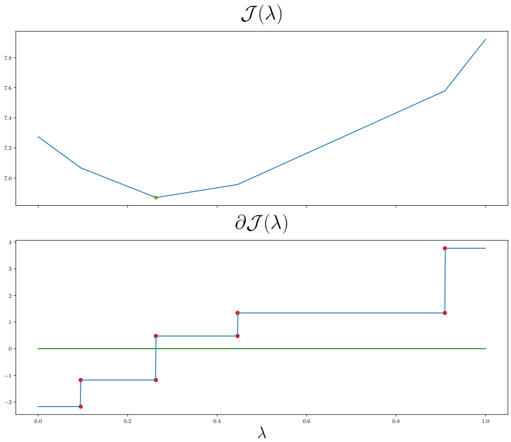
Combinación Convexa con Error Hinge

Proposicion (λ^* óptimo para el problema con error hinge)

- $\lambda^* = 0$ *es óptimo si y solo si:* $-\sum_{i:\ 0>\lambda_{(i)}} \max\left(0, c_{(i)}\right) - \sum_{0<\lambda_{(i)}} \min\left(0, c_{(i)}\right) \leq 0$
- $\lambda^* \in (0, 1)$ *es óptimo si y solo si* $0 < \lambda^* = \lambda_{(k)} < 1$ *para algún* $k = 1, \ldots, m$, *y*

$$- \sum_{i:\ \lambda_{(k)}>\lambda_{(i)}} \max\left(0, c_{(i)}\right) - \sum_{i:\ \lambda_{(k)}<\lambda_{(i)}} \min\left(0, c_{(i)}\right) \in \left[\min\left(0, c_{(k)}\right), \max\left(0, c_{(k)}\right)\right]$$

- $\lambda^* = 1$ *es óptimo en otro caso*



$\mathcal{J}(\lambda)$
7.8
7.6
7.4
7.2
7.0
$\partial\mathcal{J}(\lambda)$
4
3
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
1.0
λ

Proposicion (λ^* óptimo para el problema con error hinge cuadrático)

- $\lambda^* = 0$ *es óptimo si y solo si:* $-\sum_{i:\ 0 > c_{(i)}, 0 < \lambda_{(i)}} 2 c_i d_i - \sum_{i:\ 0 < c_{(i)}, 0 > \lambda_{(i)}} 2 c_i d_i \leq 0$
- $\lambda^* \in (0, 1)$ *es óptimo si y solo si* $0 < \lambda^* = \widehat{\lambda}_{(k)} < 1$ *para algún* $k = 1, \dots, m$*, donde*

$$\widehat{\lambda}_{(k)} = - \frac{\sum_{i:\ \lambda_{(k+1)} \geq \lambda_{(i)}} \max\left(0, c_{(i)}\right) d_{(i)} + \sum_{i:\ \lambda_{(k)} \leq \lambda_{(i)}} \min\left(0, c_{(i)}\right) d_{(i)}}{\sum_{i:\ \lambda_{(k+1)} \geq \lambda_{(i)}} \max\left(0, c_{(i)}\right)^2 + \sum_{i:\ \lambda_{(k)} \leq \lambda_{(i)}} \min\left(0, c_{(i)}\right)^2},$$

  y además $\lambda_{(k)} \leq \widehat{\lambda}_k \leq \lambda_{(k+1)}$
- $\lambda^* = 1$ *es óptimo en otro caso*

UAM
Universidad Autónoma de Madrid

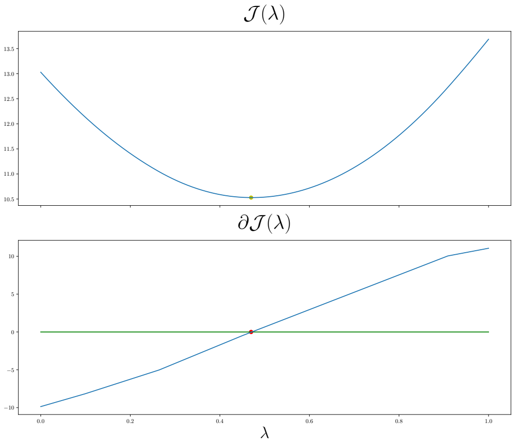


Table of Contents

- Otra manera de acoplar distintas tareas es usar una regularización Laplaciana
- Consideramos un grafo donde
  - Los nodos representan tareas
  - Las aristas y sus pesos representan las relaciones entre las tareas
- La matriz de adyacencia A tiene los pesos de las aristas
- La matriz de grados D es una matriz diagonal donde

$$(D)_{rr} = \sum_{s=1}^{T} (A)_{rs}$$

- La matriz Laplaciana se define como $L = D - A$

- Dados los modelos para cada tarea definidos como

$$h_r(\cdot) = \langle w_r, \cdot \rangle + b_r$$

- Definimos la regularización

$$\sum_{r=1}^{T} \sum_{s=1}^{T} (A)_{rs} \left\| w_r - w_s \right\|^2 ,$$

- Esta regularización se puede expresar como

$$\sum_{r=1}^{T} \sum_{s=1}^{T} (A)_{rs} \left\| w_r - w_s \right\|^2 = \sum_{r=1}^{T} \sum_{s=1}^{T} (L)_{rs} \left\langle w_r, w_s \right\rangle ,$$

Table of Contents

3 Laplaciano Adaptativo para Aprendizaje Multitarea

- Consideramos el problema de minimización

$$R(u_1, \dots, u_T) = \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle u_r, \phi(x_i^r) \rangle) + \mu \sum_r \sum_s (E)_{rs} \langle u_r, u_s \rangle \,, \tag{0}$$

- Si usamos el vector $\boldsymbol{u}^\intercal = (u_1^\intercal, \dots, u_T^\intercal)$ lo expresamos como

$$R(\boldsymbol{u}) = \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle \boldsymbol{u}, e_r \otimes \phi(x_i^r) \rangle) + \mu \left(\boldsymbol{u}^\intercal (E \otimes I) \boldsymbol{u} \right) . \tag{1}$$

Lemma

Las soluciones $u_1^*, \dots, u_T^*$ *de* (0), *o equivalentemente la solución* $\boldsymbol{u}^*$ *de* (1), *se pueden obtener minimizando*

$$S(\boldsymbol{w}) = \sum_{r=1}^{T} \sum_{i=1}^{m_r} \ell(y_i^r, \langle \boldsymbol{w}, (B_r \otimes \phi(x_i^r)) \rangle) + \mu \boldsymbol{w}^\intercal \boldsymbol{w}, \tag{2}$$

donde $\boldsymbol{w} \in \mathbb{R}^p \otimes \mathcal{H}$ *con* $p \geq T$ *y* B_r *son las columnas de* $B \in \mathbb{R}^{p \times T}$, *una matriz de rango máximo tal que* $E^{-1} = B^\intercal B$.

El kernel reproductor correspondiente es:

$$\left\langle B_r \otimes \phi(x_i^r), B_s \otimes \phi(x_j^s) \right\rangle = \left(E^{-1}\right)_{rs} k(x_i^r, x_j^s)$$

UAM
Universidad Autónoma de Madrid

Table of Contents

UAM
Universidad Autónoma de Madrid

Table of Contents

Good Luck!

- Enough for an introduction! You should know enough by now

Advanced Kernel Methods for Multi-Task Learning

Thank you for listening!

Any questions?