Entrenamiento de Support Vector Machines

Fernando Lozano

Universidad de los Andes

24 de octubre de 2022

Problema de optimización

- Primal:

$$\begin{aligned} \text{mín} \quad & P(\mathbf{w}, b, \boldsymbol{\zeta}) = \frac{1}{2}\|\mathbf{w}\|_{\mathscr{H}}^2 + {\color{red}C}\sum_{i=1}^{n}\zeta_i \\ \text{sujeto a} \quad & y_i\big(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathscr{H}} + b\big) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n \\ & \zeta_i \geq 0 \end{aligned}$$

- Dual:

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

Condiciones de Karush-Kuhn-Tucker (KKT)

1. $y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathscr{H}} + b\right) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$

Condiciones de Karush-Kuhn-Tucker (KKT)

1. $y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{H}} + b\right) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$

Condiciones de Karush-Kuhn-Tucker (KKT)

1. $y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathscr{H}} + b\right) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$

Condiciones de Karush-Kuhn-Tucker (KKT)

1. $y_i\big(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathscr{H}} + b\big) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$

Condiciones de Karush-Kuhn-Tucker (KKT)

1. $y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathscr{H}} + b\right) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\mathbf{w} = \sum_{i=1}^{n} (\alpha_i y_i) \phi(\mathbf{x}_i)$

Condiciones de Karush-Kuhn-Tucker (KKT)

1. $y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{H}} + b\right) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\mathbf{w} = \sum_{i=1}^{n} (\alpha_i y_i)\phi(\mathbf{x}_i)$
6. $\sum_{i=1}^{n} \alpha_i y_i = 0$

Condiciones de Karush-Kuhn-Tucker (KKT)

1. $y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{H}} + b\right) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\mathbf{w} = \sum_{i=1}^{n} (\alpha_i y_i)\phi(\mathbf{x}_i)$
6. $\sum_{i=1}^{n} \alpha_i y_i = 0$
7. $\alpha_i\left(y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathcal{H}} + b\right) - 1 + \zeta_i\right) = 0$

Condiciones de Karush-Kuhn-Tucker (KKT)

1. $y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathscr{H}} + b\right) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\mathbf{w} = \sum_{i=1}^{n} (\alpha_i y_i)\phi(\mathbf{x}_i)$
6. $\sum_{i=1}^{n} \alpha_i y_i = 0$
7. $\alpha_i\left(y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle_{\mathscr{H}} + b\right) - 1 + \zeta_i\right) = 0$
8. $\zeta_i \mu_i = 0$

- Denotando $f(\mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathscr{H}} + b$:

- Denotando $f(\mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathscr{H}} + b$:

$$f(\mathbf{x}_i) = \sum_{j=1}^{n} \alpha_i y_i k(\mathbf{x}_j, \mathbf{x}_i) + b$$

- Denotando $f(\mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{H}} + b$:

$$f(\mathbf{x}_i) = \sum_{j=1}^{n} \alpha_i y_i k(\mathbf{x}_j, \mathbf{x}_i) + b$$

- Se pueden reescribir las condiciones de KKT para tres casos posibles de α:

- Denotando $f(\mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{H}} + b$:

$$f(\mathbf{x}_i) = \sum_{j=1}^{n} \alpha_i y_i k(\mathbf{x}_j, \mathbf{x}_i) + b$$

- Se pueden reescribir las condiciones de KKT para tres casos posibles de α:
    1. $\alpha = 0$
    2. $0 < \alpha < C$
    3. $\alpha = C$

$\alpha = 0$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$\alpha = 0$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$$\alpha_i = 0$$

$\alpha = 0$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$$\alpha_i = 0 \Rightarrow \mu_i = C$$

$\alpha = 0$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$$\alpha_i = 0 \Rightarrow \mu_i = C \Rightarrow \zeta_i = 0$$

$\alpha = 0$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$$\alpha_i = 0 \Rightarrow \mu_i = C \Rightarrow \zeta_i = 0 \Rightarrow y_i f(\mathbf{x}_i) \geq 1$$

$0 < \alpha < C$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$0 < \alpha < C$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$$0 < \alpha < C$$

$0 < \alpha < C$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$$0 < \alpha < C \Rightarrow 0 < \mu_i < C$$

$0 < \alpha < C$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$$0 < \alpha < C \Rightarrow 0 < \mu_i < C \Rightarrow \zeta_i = 0$$

$0 < \alpha < C$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$$0 < \alpha < C \Rightarrow 0 < \mu_i < C \Rightarrow \zeta_i = 0 \Rightarrow y_i f(\mathbf{x}_i) = 1$$

$\alpha = C$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$\alpha = C$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$$\alpha_i = C$$

$\alpha = C$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$$\alpha_i = C \Rightarrow \mu_i = 0$$

$\alpha = C$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i\right) = 0$
6. $\zeta_i \mu_i = 0$

$$\alpha_i = C \Rightarrow \mu_i = 0 \Rightarrow \zeta_i \geq 0$$

$\alpha = C$

1. $y_i f(\mathbf{x}_i) - 1 + \zeta_i \geq 0 \quad i = 1, \ldots, n$
2. $\zeta_i \geq 0$
3. $\mu_i \geq 0$
4. $\alpha_i + \mu_i = C$
5. $\alpha_i \left(y_i f(\mathbf{x}_i) - 1 + \zeta_i \right) = 0$
6. $\zeta_i \mu_i = 0$

$$\alpha_i = C \Rightarrow \mu_i = 0 \Rightarrow \zeta_i \geq 0 \Rightarrow y_i f(\mathbf{x}_i) \leq 1$$

- Resolver el problema dual consiste en encontrar valores de $\alpha_i, \ldots, \alpha_n$ que satisfagan:
    1. $\alpha_i = 0 \Rightarrow y_i f(\mathbf{x}_i) \geq 1$
    2. $0 < \alpha_i < C \Rightarrow y_i f(\mathbf{x}_i) = 1$
    3. $\alpha_i = C \Rightarrow y_i f(\mathbf{x}_i) \leq 1$

- Resolver el problema dual consiste en encontrar valores de $\alpha_i, \ldots, \alpha_n$ que satisfagan:
  1. $\alpha_i = 0 \Rightarrow y_i f(\mathbf{x}_i) \geq 1$
  2. $0 < \alpha_i < C \Rightarrow y_i f(\mathbf{x}_i) = 1$
  3. $\alpha_i = C \Rightarrow y_i f(\mathbf{x}_i) \leq 1$
  4. $0 \leq \alpha_i \leq C$
  5. $\sum_{i=1}^{n} y_i \alpha_i = 0$

- Resolver el problema dual consiste en encontrar valores de $\alpha_i, \ldots, \alpha_n$ que satisfagan:
  1. $\alpha_i = 0 \Rightarrow y_i f(\mathbf{x}_i) \geq 1$
  2. $0 < \alpha_i < C \Rightarrow y_i f(\mathbf{x}_i) = 1$
  3. $\alpha_i = C \Rightarrow y_i f(\mathbf{x}_i) \leq 1$
  4. $0 \leq \alpha_i \leq C$
  5. $\sum_{i=1}^{n} y_i \alpha_i = 0$
- Para valores factibles de $\boldsymbol{\alpha}$, $L(\boldsymbol{\alpha}) \leq P(\mathbf{w}, \boldsymbol{\zeta}, b)$

- Resolver el problema dual consiste en encontrar valores de $\alpha_i, \ldots, \alpha_n$ que satisfagan:
    1. $\alpha_i = 0 \Rightarrow y_i f(\mathbf{x}_i) \geq 1$
    2. $0 < \alpha_i < C \Rightarrow y_i f(\mathbf{x}_i) = 1$
    3. $\alpha_i = C \Rightarrow y_i f(\mathbf{x}_i) \leq 1$
    4. $0 \leq \alpha_i \leq C$
    5. $\sum_{i=1}^{n} y_i \alpha_i = 0$
- Para valores factibles de $\boldsymbol{\alpha}$, $L(\boldsymbol{\alpha}) \leq P(\mathbf{w}, \boldsymbol{\zeta}, b)$
- En optimalidad $L(\boldsymbol{\alpha}^*) = P(\mathbf{w}^*, \boldsymbol{\zeta}^*, b^*)$ (brecha de dualidad es cero).

Métodos de solución

Métodos de solución

- Solución del QP por algoritmos de punto interior:

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.
  - Dificil implementación.

Métodos de solución

- Solución del QP por algoritmos de punto interior:
    - Robusto, confiable. Solución dispersa.
    - Dificil implementación.
    - Sólo problemas pequeños/medianos.

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.
  - Dificil implementación.
  - Sólo problemas pequeños/medianos.
- Variación del conjunto de trabajo:

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.
  - Dificil implementación.
  - Sólo problemas pequeños/medianos.
- Variación del conjunto de trabajo:
  - QP original con algunos $\alpha_i = 0$ es un QP más pequeño.

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.
  - Dificil implementación.
  - Sólo problemas pequeños/medianos.
- Variación del conjunto de trabajo:
  - QP original con algunos $\alpha_i = 0$ es un QP más pequeño.
  - Estrategia: Descomponer QP original en una secuencia de QP más pequeños. Resolver con rutina QP.

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.
  - Dificil implementación.
  - Sólo problemas pequeños/medianos.
- Variación del conjunto de trabajo:
  - QP original con algunos $\alpha_i = 0$ es un QP más pequeño.
  - Estrategia: Descomponer QP original en una secuencia de QP más pequeños. Resolver con rutina QP.
  - Chunking (Boser, Guyon, Vapnik, 1997).

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.
  - Dificil implementación.
  - Sólo problemas pequeños/medianos.
- Variación del conjunto de trabajo:
  - QP original con algunos $\alpha_i = 0$ es un QP más pequeño.
  - Estrategia: Descomponer QP original en una secuencia de QP más pequeños. Resolver con rutina QP.
  - Chunking (Boser, Guyon, Vapnik, 1997).
    - Comenzar con $\boldsymbol{\alpha}$ factible.

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.
  - Dificil implementación.
  - Sólo problemas pequeños/medianos.
- Variación del conjunto de trabajo:
  - QP original con algunos $\alpha_i = 0$ es un QP más pequeño.
  - Estrategia: Descomponer QP original en una secuencia de QP más pequeños. Resolver con rutina QP.
  - Chunking (Boser, Guyon, Vapnik, 1997).
    - Comenzar con $\boldsymbol{\alpha}$ factible.
    - Resuelve QP usando los $\alpha_i \neq 0$ más los peores M α_i (que no satisfacen) KKT.

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.
  - Dificil implementación.
  - Sólo problemas pequeños/medianos.
- Variación del conjunto de trabajo:
  - QP original con algunos $\alpha_i = 0$ es un QP más pequeño.
  - Estrategia: Descomponer QP original en una secuencia de QP más pequeños. Resolver con rutina QP.
  - Chunking (Boser, Guyon, Vapnik, 1997).
    - ★ Comenzar con $\boldsymbol{\alpha}$ factible.
    - ★ Resuelve QP usando los $\alpha_i \neq 0$ más los peores M α_i (que no satisfacen) KKT.
    - ★ Problemas QP de tamaño variable.

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.
  - Dificil implementación.
  - Sólo problemas pequeños/medianos.
- Variación del conjunto de trabajo:
  - QP original con algunos $\alpha_i = 0$ es un QP más pequeño.
  - Estrategia: Descomponer QP original en una secuencia de QP más pequeños. Resolver con rutina QP.
  - Chunking (Boser, Guyon, Vapnik, 1997).
    - Comenzar con $\boldsymbol{\alpha}$ factible.
    - Resuelve QP usando los $\alpha_i \neq 0$ más los peores M α_i (que no satisfacen) KKT.
    - Problemas QP de tamaño variable.
  - Algoritmo de Descomposición (Osuna, Freund, Girosi, 1997)

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.
  - Dificil implementación.
  - Sólo problemas pequeños/medianos.
- Variación del conjunto de trabajo:
  - QP original con algunos $\alpha_i = 0$ es un QP más pequeño.
  - Estrategia: Descomponer QP original en una secuencia de QP más pequeños. Resolver con rutina QP.
  - Chunking (Boser, Guyon, Vapnik, 1997).
    - ★ Comenzar con $\boldsymbol{\alpha}$ factible.
    - ★ Resuelve QP usando los $\alpha_i \neq 0$ más los peores M α_i (que no satisfacen) KKT.
    - ★ Problemas QP de tamaño variable.
  - Algoritmo de Descomposición (Osuna, Freund, Girosi, 1997)
    - ★ Mantiene QP de tamaño constante.

Métodos de solución

- Solución del QP por algoritmos de punto interior:
  - Robusto, confiable. Solución dispersa.
  - Dificil implementación.
  - Sólo problemas pequeños/medianos.
- Variación del conjunto de trabajo:
  - QP original con algunos $\alpha_i = 0$ es un QP más pequeño.
  - Estrategia: Descomponer QP original en una secuencia de QP más pequeños. Resolver con rutina QP.
  - Chunking (Boser, Guyon, Vapnik, 1997).
    - ★ Comenzar con $\boldsymbol{\alpha}$ factible.
    - ★ Resuelve QP usando los $\alpha_i \neq 0$ más los peores M α_i (que no satisfacen) KKT.
    - ★ Problemas QP de tamaño variable.
  - Algoritmo de Descomposición (Osuna, Freund, Girosi, 1997)
    - ★ Mantiene QP de tamaño constante.
    - ★ En cada etapa añade un número fijo de α_i que no satisface KKT.

Sequential Minimal Optimization (SMO) (Platt, 1998)

Sequential Minimal Optimization (SMO) (Platt, 1998)

- Resuelve secuencia de QPs de tamaño mínimo (2×2) analíticamente.

Sequential Minimal Optimization (SMO) (Platt, 1998)

- Resuelve secuencia de QPs de tamaño mínimo (2×2) analíticamente.
- No requiere subrutina de solución de QP.

Sequential Minimal Optimization (SMO) (Platt, 1998)

- Resuelve secuencia de QPs de tamaño mínimo (2×2) analíticamente.
- No requiere subrutina de solución de QP.
- Robusto numéricamente.

Sequential Minimal Optimization (SMO) (Platt, 1998)

- Resuelve secuencia de QPs de tamaño mínimo (2×2) analíticamente.
- No requiere subrutina de solución de QP.
- Robusto numéricamente.
- Aunque realiza muchas iteraciones, cada iteración es muy rápida.

$$\begin{aligned}
\text{máx} \quad & \alpha_i + \alpha_j + \sum_{k \neq i,j} \alpha_k - \frac{1}{2}(\boldsymbol{\alpha}_{\mathbf{ij}} + \boldsymbol{\alpha}_{\bar{\mathbf{ij}}})^T \mathbf{K} (\boldsymbol{\alpha}_{\mathbf{ij}} + \boldsymbol{\alpha}_{\bar{\mathbf{ij}}}) \\
\text{sujeto a} \quad & \alpha_i y_i + \alpha_j y_j + \sum_{k \neq i,j} \alpha_k y_k = 0 \\
& 0 \leq \boldsymbol{\alpha} \leq C
\end{aligned}$$

$$
\begin{aligned}
\text{máx} \quad & \alpha_i + \alpha_j + \overbrace{\sum_{k \neq i,j} \alpha_k}^{\text{constante}} - \frac{1}{2}(\boldsymbol{\alpha}_{\boldsymbol{ij}} + \boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}})^T \mathbf{K} (\boldsymbol{\alpha}_{\boldsymbol{ij}} + \boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}}) \\
\text{sujeto a} \quad & \alpha_i y_i + \alpha_j y_j + \sum_{k \neq i,j} \alpha_k y_k = 0 \\
& 0 \leq \boldsymbol{\alpha} \leq C
\end{aligned}
$$

$$
\begin{aligned}
\text{máx} \quad & \alpha_i + \alpha_j - \frac{1}{2}(\boldsymbol{\alpha}_{\boldsymbol{ij}} + \boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}})^T \mathbf{K} (\boldsymbol{\alpha}_{\boldsymbol{ij}} + \boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}}) \\
\text{sujeto a} \quad & \alpha_i y_i + \alpha_j y_j + \sum_{k \neq i,j} \alpha_k y_k = 0 \\
& 0 \leq \boldsymbol{\alpha} \leq C
\end{aligned}
$$

$$
\begin{aligned}
\text{máx} \quad & \alpha_i + \alpha_j - \frac{1}{2}\boldsymbol{\alpha}_{\overline{\boldsymbol{ij}}}^T \mathbf{K} \boldsymbol{\alpha}_{\overline{\boldsymbol{ij}}} - \frac{1}{2}\boldsymbol{\alpha}_{\boldsymbol{ij}}^T \mathbf{K} \boldsymbol{\alpha}_{\boldsymbol{ij}} - \boldsymbol{\alpha}_{\overline{\boldsymbol{ij}}}^T \mathbf{K} \boldsymbol{\alpha}_{\boldsymbol{ij}} \\
\text{sujeto a} \quad & \alpha_i y_i + \alpha_j y_j + \sum_{k \neq i,j} \alpha_k y_k = 0 \\
& 0 \leq \boldsymbol{\alpha} \leq C
\end{aligned}
$$

$$
\begin{aligned}
\text{máx} \quad & \alpha_i + \alpha_j - \cancelto{\text{constante}}{\frac{1}{2}\boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}}{}^T\mathbf{K}\boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}}} - \frac{1}{2}\boldsymbol{\alpha}_{\boldsymbol{ij}}{}^T\mathbf{K}\boldsymbol{\alpha}_{\boldsymbol{ij}} - \boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}}{}^T\mathbf{K}\boldsymbol{\alpha}_{\boldsymbol{ij}} \\
\text{sujeto a} \quad & \alpha_i y_i + \alpha_j y_j + \sum_{k \neq i,j} \alpha_k y_k = 0 \\
& 0 \leq \boldsymbol{\alpha} \leq C
\end{aligned}
$$

$$
\begin{aligned}
\text{máx} \quad & \alpha_i + \alpha_j - \frac{1}{2}\boldsymbol{\alpha}_{\boldsymbol{ij}}{}^T \mathbf{K} \boldsymbol{\alpha}_{\boldsymbol{ij}} - \boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}}{}^T \mathbf{K} \boldsymbol{\alpha}_{\boldsymbol{ij}} \\
\text{sujeto a} \quad & \alpha_i y_i + \alpha_j y_j + \sum_{k \neq i,j} \alpha_k y_k = 0 \\
& 0 \leq \boldsymbol{\alpha} \leq C
\end{aligned}
$$

$$
\begin{aligned}
\text{máx} \quad & \alpha_i + \alpha_j - \frac{1}{2}\boldsymbol{\alpha_{ij}}^T \mathbf{K} \boldsymbol{\alpha_{ij}} - \alpha_i \boldsymbol{\alpha_{\bar{ij}}}^T \mathbf{K}_i - \alpha_j \boldsymbol{\alpha_{\bar{ij}}}^T \mathbf{K}_j \\
\text{sujeto a} \quad & \alpha_i y_i + \alpha_j y_j + \sum_{k \neq i,j} \alpha_k y_k = 0 \\
& 0 \leq \alpha_i, \alpha_j \leq C
\end{aligned}
$$

$$\boldsymbol{\alpha}_{\bar{\boldsymbol{i}}\bar{\boldsymbol{j}}}{}^T \mathbf{K}_i = \sum_{k=1, k \neq i,j}^{n} \alpha_k y_i y_k k(\mathbf{x}_i, \mathbf{x}_k)$$

$$
\begin{aligned}
\boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}}{}^{T}\mathbf{K}_i &= \sum_{k=1, k\neq i,j}^{n} \alpha_k y_i y_k k(\mathbf{x}_i, \mathbf{x}_k) \\
&= y_i \left(f^{\text{old}}(\mathbf{x}_i) - b^{\text{old}} - \alpha_i^{\text{old}} y_i k(\mathbf{x}_i, \mathbf{x}_i) - \alpha_j^{\text{old}} y_j k(\mathbf{x}_j, \mathbf{x}_i) \right)
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}}{}^{T}\mathbf{K}_i &= \sum_{k=1,k\neq i,j}^{n} \alpha_k y_i y_k k(\mathbf{x}_i, \mathbf{x}_k) \\
&= y_i \left(f^{\text{old}}(\mathbf{x}_i) - b^{\text{old}} - \alpha_i^{\text{old}} y_i k(\mathbf{x}_i, \mathbf{x}_i) - \alpha_j^{\text{old}} y_j k(\mathbf{x}_j, \mathbf{x}_i) \right) \\
&= y_i v_i
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}}{}^{T}\mathbf{K}_i &= \sum_{k=1,k\neq i,j}^{n} \alpha_k y_i y_k k(\mathbf{x}_i,\mathbf{x}_k) \\
&= y_i\left(f^{\text{old}}(\mathbf{x}_i) - b^{\text{old}} - \alpha_i^{\text{old}} y_i k(\mathbf{x}_i,\mathbf{x}_i) - \alpha_j^{\text{old}} y_j k(\mathbf{x}_j,\mathbf{x}_i)\right) \\
&= y_i v_i
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{\alpha}_{\bar{\boldsymbol{ij}}}{}^{T}\mathbf{K}_j &= \sum_{k=1,k\neq i,j}^{n} \alpha_k y_j y_k k(\mathbf{x}_j,\mathbf{x}_k) \\
&= y_j\left(f^{\text{old}}(\mathbf{x}_j) - b^{\text{old}} - \alpha_j^{\text{old}} y_j k(\mathbf{x}_j,\mathbf{x}_j) - \alpha_i^{\text{old}} y_i k(\mathbf{x}_i,\mathbf{x}_j)\right) \\
&= y_j v_j
\end{aligned}
$$

$$
\begin{aligned}
\text{máx} \quad & \alpha_i + \alpha_j - \frac{1}{2}\boldsymbol{\alpha_{ij}}^T \mathbf{K} \boldsymbol{\alpha_{ij}} - y_i v_i \alpha_i - \alpha_j y_j v_j \\
\text{sujeto a} \quad & \alpha_i y_i + \alpha_j y_j + \sum_{k \neq i,j} \alpha_k y_k = 0 \\
& 0 \leq \alpha_i, \alpha_j \leq C
\end{aligned}
$$

$$
\begin{aligned}
\text{máx} \quad & \alpha_i(1 - y_i v_i) + \alpha_j(1 - y_j v_j) - \frac{1}{2}\boldsymbol{\alpha_{ij}}^T \mathbf{K} \boldsymbol{\alpha_{ij}} \\
\text{sujeto a} \quad & \alpha_i y_i + \alpha_j y_j + \sum_{k \neq i,j} \alpha_k y_k = 0 \\
& 0 \leq \alpha_i, \alpha_j \leq C
\end{aligned}
$$

$$
\begin{aligned}
\text{máx} \quad & \alpha_i(1 - y_i v_i) + \alpha_j(1 - y_j v_j) - \frac{1}{2}\mathbf{K}_{ii}{\alpha_i}^2 - \frac{1}{2}\mathbf{K}_{jj}{\alpha_j}^2 - \mathbf{K}_{ij}\alpha_i\alpha_j \\
\text{sujeto a} \quad & \alpha_i y_i + \alpha_j y_j + \sum_{k \neq i,j} \alpha_k y_k = 0 \\
& 0 \leq \alpha_i, \alpha_j \leq C
\end{aligned}
$$

$$\begin{aligned} \text{máx} \quad & \alpha_i(1 - y_i v_i) + \alpha_j(1 - y_j v_j) - \frac{1}{2}\mathbf{K}_{ii}{\alpha_i}^2 - \frac{1}{2}\mathbf{K}_{jj}{\alpha_j}^2 - \mathbf{K}_{ij}\alpha_i\alpha_j \\ \text{sujeto a} \quad & \alpha_i + s\alpha_j = \gamma \\ & 0 \le \alpha_i, \alpha_j \le C \end{aligned}$$

$$\begin{aligned} \text{máx} \quad & \alpha_i(1 - y_i v_i) + \alpha_j(1 - y_j v_j) - \frac{1}{2}\mathbf{K}_{ii}{\alpha_i}^2 - \frac{1}{2}\mathbf{K}_{jj}{\alpha_j}^2 - \mathbf{K}_{ij}\alpha_i\alpha_j \\ \text{sujeto a} \quad & \alpha_i + s\alpha_j = \gamma \\ & 0 \leq \alpha_i, \alpha_j \leq C \end{aligned}$$

donde $s = y_i y_j$ y $\gamma = \alpha_i^{\text{old}} + s\alpha_j^{\text{old}}$

$$
\begin{aligned}
\text{máx} \quad & \alpha_i(1 - y_i v_i) + \alpha_j(1 - y_j v_j) \\
& - \frac{1}{2}k(\mathbf{x}_i, \mathbf{x}_i)\alpha_i^{\,2} - \frac{1}{2}k(\mathbf{x}_j, \mathbf{x}_j)\alpha_j^{\,2} - sk(\mathbf{x}_i, \mathbf{x}_j)\alpha_i\alpha_j \\
\text{sujeto a} \quad & \alpha_i + s\alpha_j = \gamma \\
& 0 \leq \alpha_i, \alpha_j \leq C
\end{aligned}
$$

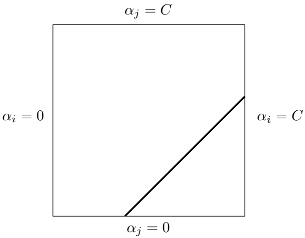
donde $s = y_i y_j$ y $\gamma = \alpha_i^{\text{old}} + s\alpha_j^{\text{old}}$

$$
\begin{aligned}
\text{máx} \quad & \alpha_i(1 - y_i v_i) + \alpha_j(1 - y_j v_j) \\
& - \frac{1}{2}k(\mathbf{x}_i, \mathbf{x}_i)\alpha_i^2 - \frac{1}{2}k(\mathbf{x}_j, \mathbf{x}_j)\alpha_j^2 - sk(\mathbf{x}_i, \mathbf{x}_j)\alpha_i\alpha_j \\
\text{sujeto a} \quad & \alpha_i + s\alpha_j = \gamma \\
& 0 \leq \alpha_i, \alpha_j \leq C
\end{aligned}
$$

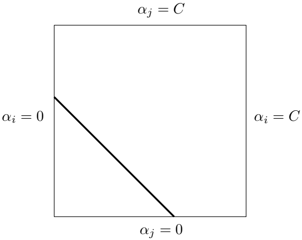
donde $s = y_i y_j$ y $\gamma = \alpha_i^{\text{old}} + s\alpha_j^{\text{old}}$

- Problema cuadrático en dos variables.

Región Factible



$y_i \neq y_j \Rightarrow \alpha_i - \alpha_j = \gamma$



$y_i = y_j \Rightarrow \alpha_i + \alpha_j = \gamma$

Actualización de α_i y α_j

- Maximizar cuadrática en segmento de línea.

Actualización de α_i y α_j

- Maximizar cuadrática en segmento de línea.
- Sustituir $\alpha_i = \gamma - s\alpha_j$ en función objetivo

Actualización de α_i y α_j

- Maximizar cuadrática en segmento de línea.
- Sustituir $\alpha_i = \gamma - s\alpha_j$ en función objetivo$\Rightarrow$ cuadrática cóncava de una variable.

Actualización de α_i y α_j

- Maximizar cuadrática en segmento de línea.
- Sustituir $\alpha_i = \gamma - s\alpha_j$ en función objetivo$\Rightarrow$ cuadrática cóncava de una variable.
- Máximo está en punto crítico y/o en uno de los extremos de la línea.

Actualización de α_i y α_j

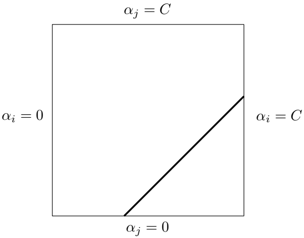
- Maximizar cuadrática en segmento de línea.
- Sustituir $\alpha_i = \gamma - s\alpha_j$ en función objetivo$\Rightarrow$ cuadrática cóncava de una variable.
- Máximo está en punto crítico y/o en uno de los extremos de la línea.
- Procedimiento:

Actualización de α_i y α_j

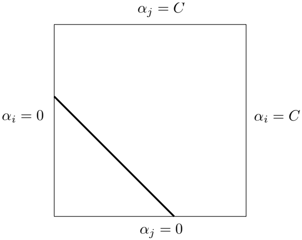
- Maximizar cuadrática en segmento de línea.
- Sustituir $\alpha_i = \gamma - s\alpha_j$ en función objetivo$\Rightarrow$ cuadrática cóncava de una variable.
- Máximo está en punto crítico y/o en uno de los extremos de la línea.
- Procedimiento:
  1. Hallar punto crítico.

Actualización de α_i y α_j

- Maximizar cuadrática en segmento de línea.
- Sustituir $\alpha_i = \gamma - s\alpha_j$ en función objetivo$\Rightarrow$ cuadrática cóncava de una variable.
- Máximo está en punto crítico y/o en uno de los extremos de la línea.
- Procedimiento:
  1. Hallar punto crítico.
  2. Recortar a la línea.



$y_i \neq y_j \Rightarrow \alpha_i - \alpha_j = \gamma$



$y_i = y_j \Rightarrow \alpha_i + \alpha_j = \gamma$

	$y_i \neq y_j$	$y_i = y_j$
L	$\text{máx}(0, \alpha_j^{\text{old}} - \alpha_i^{\text{old}})$	$\text{máx}(0, \alpha_j^{\text{old}} + \alpha_i^{\text{old}} - C)$
H	$\text{mín}(C, C + (\alpha_j^{\text{old}} - \alpha_i^{\text{old}}))$	$\text{mín}(C, \alpha_j^{\text{old}} + \alpha_i^{\text{old}})$

- Derivando e igualando a cero:

$$\alpha_j^{\text{new}} = \alpha_j^{\text{old}} - \frac{y_2 E_i - E_j}{\eta}$$

donde

- Derivando e igualando a cero:

$$\alpha_j^{\text{new}} = \alpha_j^{\text{old}} - \frac{y_2 E_i - E_j}{\eta}$$

donde

  - $E_i = f^{\text{old}}(\mathbf{x}_i) - y_i$, $E_j = f^{\text{old}}(\mathbf{x}_j) - y_j$

- Derivando e igualando a cero:

$$\alpha_j^{\text{new}} = \alpha_j^{\text{old}} - \frac{y_2 E_i - E_j}{\eta}$$

donde

- $E_i = f^{\text{old}}(\mathbf{x}_i) - y_i$, $E_j = f^{\text{old}}(\mathbf{x}_j) - y_j$
- $\eta = 2k(\mathbf{x}_i, \mathbf{x}_j) - k(\mathbf{x}_i, \mathbf{x}_i) - k(\mathbf{x}_j, \mathbf{x}_j)$

- Derivando e igualando a cero:

$$\alpha_j^{\text{new}} = \alpha_j^{\text{old}} - \frac{y_2 E_i - E_j}{\eta}$$

donde

  - $E_i = f^{\text{old}}(\mathbf{x}_i) - y_i$, $E_j = f^{\text{old}}(\mathbf{x}_j) - y_j$
  - $\eta = 2k(\mathbf{x}_i, \mathbf{x}_j) - k(\mathbf{x}_i, \mathbf{x}_i) - k(\mathbf{x}_j, \mathbf{x}_j)$

- Recortar:

$$\alpha_j^{\text{new,clip}} = \begin{cases} H & \text{si } \alpha_j^{\text{new}} \geq H \\ \alpha_j^{\text{new}} & \text{si } L < \alpha_j^{\text{new}} < H \\ L & \text{si } \alpha_j^{\text{new}} \leq L \end{cases}$$

- Derivando e igualando a cero:

$$\alpha_j^{\text{new}} = \alpha_j^{\text{old}} - \frac{y_2 E_i - E_j}{\eta}$$

donde

  - $E_i = f^{\text{old}}(\mathbf{x}_i) - y_i$, $E_j = f^{\text{old}}(\mathbf{x}_j) - y_j$
  - $\eta = 2k(\mathbf{x}_i, \mathbf{x}_j) - k(\mathbf{x}_i, \mathbf{x}_i) - k(\mathbf{x}_j, \mathbf{x}_j)$

- Recortar:

$$\alpha_j^{\text{new,clip}} = \begin{cases} H & \text{si } \alpha_j^{\text{new}} \geq H \\ \alpha_j^{\text{new}} & \text{si } L < \alpha_j^{\text{new}} < H \\ L & \text{si } \alpha_j^{\text{new}} \leq L \end{cases}$$

- Reemplazando en la restricción:

$$\alpha_i^{\text{new}} = \alpha_i^{\text{old}} + s(\alpha_j^{\text{old}} - \alpha_j^{\text{new,clip}})$$

- Derivando e igualando a cero:

$$\alpha_j^{\text{new}} = \alpha_j^{\text{old}} - \frac{y_2 E_i - E_j}{\eta}$$

donde
  - $E_i = f^{\text{old}}(\mathbf{x}_i) - y_i$, $E_j = f^{\text{old}}(\mathbf{x}_j) - y_j$
  - $\eta = 2k(\mathbf{x}_i, \mathbf{x}_j) - k(\mathbf{x}_i, \mathbf{x}_i) - k(\mathbf{x}_j, \mathbf{x}_j)$
- Recortar:

$$\alpha_j^{\text{new,clip}} = \begin{cases} H & \text{si } \alpha_j^{\text{new}} \geq H \\ \alpha_j^{\text{new}} & \text{si } L < \alpha_j^{\text{new}} < H \\ L & \text{si } \alpha_j^{\text{new}} \leq L \end{cases}$$

- Reemplazando en la restricción:

$$\alpha_i^{\text{new}} = \alpha_i^{\text{old}} + s(\alpha_j^{\text{old}} - \alpha_j^{\text{new,clip}})$$

- Actualiza b, de manera que se satisfagan KKT para i, j

Escogencia de α_i, α_j

Escogencia de α_i, α_j

1. Primer multiplicador:

Escogencia de α_i, α_j

1. Primer multiplicador:
   - α_i: $(\mathbf{x}_i, y_i)$ viola KKT:

Escogencia de α_i, α_j

1. Primer multiplicador:
   - α_i: $(\mathbf{x}_i, y_i)$ viola KKT:
     - ★ Prioridad a datos con $0 < \alpha < C$

Escogencia de α_i, α_j

1. Primer multiplicador:
   - α_i: $(\mathbf{x}_i, y_i)$ viola KKT:
     - Prioridad a datos con $0 < \alpha < C$ (cambian más).

Escogencia de α_i, α_j

1. Primer multiplicador:
   - α_i: $(\mathbf{x}_i, y_i)$ viola KKT:
     - ★ Prioridad a datos con $0 < \alpha < C$ (cambian más).
     - ★ Pasada menos frecuente sobre todos los datos.

Escogencia de α_i, α_j

1. Primer multiplicador:
   - α_i: $(\mathbf{x}_i, y_i)$ viola KKT:
     - Prioridad a datos con $0 < \alpha < C$ (cambian más).
     - Pasada menos frecuente sobre todos los datos.
2. Segundo multiplicador:

Escogencia de α_i, α_j

1. Primer multiplicador:
   - α_i: $(\mathbf{x}_i, y_i)$ viola KKT:
     - Prioridad a datos con $0 < \alpha < C$ (cambian más).
     - Pasada menos frecuente sobre todos los datos.
2. Segundo multiplicador:
   - Idealmente, se escoge α_j que maximice cambio en la función objetivo.

Escogencia de α_i, α_j

1. Primer multiplicador:
   - α_i: $(\mathbf{x}_i, y_i)$ viola KKT:
     - Prioridad a datos con $0 < \alpha < C$ (cambian más).
     - Pasada menos frecuente sobre todos los datos.
2. Segundo multiplicador:
   - Idealmente, se escoge α_j que maximice cambio en la función objetivo.
   - Heurística: Máximo $|E_1 - E_2|$

Convergencia

Convergencia

- Mantiene $\boldsymbol{\alpha}$ factible.

Convergencia

- Mantiene $\boldsymbol{\alpha}$ factible.
- Cada iteración incrementa $L(\boldsymbol{\alpha})$.

Convergencia

- Mantiene $\boldsymbol{\alpha}$ factible.
- Cada iteración incrementa $L(\boldsymbol{\alpha})$.
- Cota superior $P(\mathbf{w}^*, \boldsymbol{\zeta}^*, b^*)$

Convergencia

- Mantiene $\boldsymbol{\alpha}$ factible.
- Cada iteración incrementa $L(\boldsymbol{\alpha})$.
- Cota superior $P(\mathbf{w}^*, \boldsymbol{\zeta}^*, b^*) \Rightarrow$ convergencia asimptótica.

Convergencia

- Mantiene $\boldsymbol{\alpha}$ factible.
- Cada iteración incrementa $L(\boldsymbol{\alpha})$.
- Cota superior $P(\mathbf{w}^*, \boldsymbol{\zeta}^*, b^*) \Rightarrow$ convergencia asimptótica.
- Criterio de parada: KKT con tolerancia $\epsilon \sim 10^{-3}$

SVMs sin offset

SVMs sin offset

- Teoría indica que presencia de offset no mejora generalización (Steinwart, 2003, 2008).

SVMs sin offset

- Teoría indica que presencia de offset no mejora generalización (Steinwart, 2003, 2008).
- Sin offset el problema dual no incluye restricción lineal:

$$\begin{aligned} \text{mín} \quad & W(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\boldsymbol{\tau} \\ \text{subject to} \quad & 0 \leq \boldsymbol{\alpha} \leq C \end{aligned}$$

SVMs sin offset

- Teoría indica que presencia de offset no mejora generalización (Steinwart, 2003, 2008).
- Sin offset el problema dual no incluye restricción lineal:

$$\begin{aligned} \text{mín} \quad & W(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{K}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{\tau} \\ \text{subject to} \quad & 0 \leq \boldsymbol{\alpha} \leq C \end{aligned}$$

- Es posible actualizar sólo un α a la vez.

SVMs sin offset

- Teoría indica que presencia de offset no mejora generalización (Steinwart, 2003, 2008).
- Sin offset el problema dual no incluye restricción lineal:

$$\begin{aligned} \text{mín} \quad & W(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{K}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{\tau} \\ \text{subject to} \quad & 0 \le \boldsymbol{\alpha} \le C \end{aligned}$$

- Es posible actualizar sólo un α a la vez.
- Algoritmo de entrenamiento más simple/más rápido (Steinwart, Hush and Scovel, 2009)

Coordinate descent algorithm

Algorithm 1 Coordinate Descent

Initialize $\boldsymbol{\alpha}, \nabla W(\alpha)$
repeat
  Pick α_p to be updated.
  Update α_p
  Update $\nabla W(\alpha_p)$
until Stopping condition is met

Updating α

Updating α

- Suppose we selected α_p for updating.

Updating α

- Suppose we selected α_p for updating.
- Line search in the direction of $\mathbf{e}_p$:

Updating α

- Suppose we selected α_p for updating.
- Line search in the direction of $\mathbf{e}_p$:

$$\frac{\partial}{\partial \eta} W(\boldsymbol{\alpha} + \eta \mathbf{e}_p) = \nabla W(\boldsymbol{\alpha} + \eta \mathbf{e}_p)^T \mathbf{e}_p$$

Updating α

- Suppose we selected α_p for updating.
- Line search in the direction of $\mathbf{e}_p$:

$$\begin{aligned}\frac{\partial}{\partial \eta} W(\boldsymbol{\alpha} + \eta \mathbf{e}_p) &= \nabla W(\boldsymbol{\alpha} + \eta \mathbf{e}_p)^T \mathbf{e}_p \\ &= \left[\mathbf{K}(\boldsymbol{\alpha} + \eta \mathbf{e}_p - \boldsymbol{\tau})\right]^T \mathbf{e}_p\end{aligned}$$

Updating α

- Suppose we selected α_p for updating.
- Line search in the direction of $\mathbf{e}_p$:

$$\begin{aligned}
\frac{\partial}{\partial \eta} W(\boldsymbol{\alpha} + \eta \mathbf{e}_p) &= \nabla W(\boldsymbol{\alpha} + \eta \mathbf{e}_p)^T \mathbf{e}_p \\
&= \left[\mathbf{K}(\boldsymbol{\alpha} + \eta \mathbf{e}_p - \boldsymbol{\tau})\right]^T \mathbf{e}_p \\
&= \left[\mathbf{K}\boldsymbol{\alpha} - \boldsymbol{\tau} + \eta \mathbf{K}\mathbf{e}_p\right]^T \mathbf{e}_p
\end{aligned}$$

Updating α

- Suppose we selected α_p for updating.
- Line search in the direction of $\mathbf{e}_p$:

$$\begin{aligned}
\frac{\partial}{\partial \eta} W(\boldsymbol{\alpha} + \eta \mathbf{e}_p) &= \nabla W(\boldsymbol{\alpha} + \eta \mathbf{e}_p)^T \mathbf{e}_p \\
&= \left[\mathbf{K}(\boldsymbol{\alpha} + \eta \mathbf{e}_p - \boldsymbol{\tau})\right]^T \mathbf{e}_p \\
&= \left[\mathbf{K}\boldsymbol{\alpha} - \boldsymbol{\tau} + \eta \mathbf{K}\mathbf{e}_p\right]^T \mathbf{e}_p \\
&= \left[\nabla W(\boldsymbol{\alpha}) + \eta \mathbf{K}\mathbf{e}_p\right]^T \mathbf{e}_p
\end{aligned}$$

Updating α

- Suppose we selected α_p for updating.
- Line search in the direction of $\mathbf{e}_p$:

$$
\begin{aligned}
\frac{\partial}{\partial \eta} W(\boldsymbol{\alpha} + \eta \mathbf{e}_p) &= \nabla W(\boldsymbol{\alpha} + \eta \mathbf{e}_p)^T \mathbf{e}_p \\
&= \left[\mathbf{K}(\boldsymbol{\alpha} + \eta \mathbf{e}_p - \boldsymbol{\tau})\right]^T \mathbf{e}_p \\
&= \left[\mathbf{K}\boldsymbol{\alpha} - \boldsymbol{\tau} + \eta \mathbf{K}\mathbf{e}_p\right]^T \mathbf{e}_p \\
&= \left[\nabla W(\boldsymbol{\alpha}) + \eta \mathbf{K}\mathbf{e}_p\right]^T \mathbf{e}_p \\
&= \nabla_p W(\boldsymbol{\alpha}) + \eta \mathbf{e}_p^T \mathbf{K}\mathbf{e}_p
\end{aligned}
$$

Updating α

- Suppose we selected α_p for updating.
- Line search in the direction of $\mathbf{e}_p$:

$$\begin{aligned}
\frac{\partial}{\partial \eta} W(\boldsymbol{\alpha} + \eta \mathbf{e}_p) &= \nabla W(\boldsymbol{\alpha} + \eta \mathbf{e}_p)^T \mathbf{e}_p \\
&= \left[\mathbf{K}(\boldsymbol{\alpha} + \eta \mathbf{e}_p - \boldsymbol{\tau})\right]^T \mathbf{e}_p \\
&= \left[\mathbf{K}\boldsymbol{\alpha} - \boldsymbol{\tau} + \eta \mathbf{K}\mathbf{e}_p\right]^T \mathbf{e}_p \\
&= \left[\nabla W(\boldsymbol{\alpha}) + \eta \mathbf{K}\mathbf{e}_p\right]^T \mathbf{e}_p \\
&= \nabla_p W(\boldsymbol{\alpha}) + \eta \mathbf{e}_p^T \mathbf{K}\mathbf{e}_p = 0
\end{aligned}$$

Updating α

- Suppose we selected α_p for updating.
- Line search in the direction of $\mathbf{e}_p$:

$$
\begin{aligned}
\frac{\partial}{\partial \eta} W(\boldsymbol{\alpha} + \eta \mathbf{e}_p) &= \nabla W(\boldsymbol{\alpha} + \eta \mathbf{e}_p)^T \mathbf{e}_p \\
&= \left[\mathbf{K}(\boldsymbol{\alpha} + \eta \mathbf{e}_p - \boldsymbol{\tau})\right]^T \mathbf{e}_p \\
&= \left[\mathbf{K}\boldsymbol{\alpha} - \boldsymbol{\tau} + \eta \mathbf{K}\mathbf{e}_p\right]^T \mathbf{e}_p \\
&= \left[\nabla W(\boldsymbol{\alpha}) + \eta \mathbf{K}\mathbf{e}_p\right]^T \mathbf{e}_p \\
&= \nabla_p W(\boldsymbol{\alpha}) + \eta \mathbf{e}_p^T \mathbf{K}\mathbf{e}_p = 0 \Rightarrow \eta = \frac{-\nabla_p W(\boldsymbol{\alpha})}{\mathbf{K}_{pp}}
\end{aligned}
$$

Updating α

- Suppose we selected α_p for updating.
- Line search in the direction of $\mathbf{e}_p$:

$$
\begin{aligned}
\frac{\partial}{\partial \eta} W(\boldsymbol{\alpha} + \eta \mathbf{e}_p) &= \nabla W(\boldsymbol{\alpha} + \eta \mathbf{e}_p)^T \mathbf{e}_p \\
&= \left[\mathbf{K}(\boldsymbol{\alpha} + \eta \mathbf{e}_p - \boldsymbol{\tau})\right]^T \mathbf{e}_p \\
&= \left[\mathbf{K}\boldsymbol{\alpha} - \boldsymbol{\tau} + \eta \mathbf{K}\mathbf{e}_p\right]^T \mathbf{e}_p \\
&= \left[\nabla W(\boldsymbol{\alpha}) + \eta \mathbf{K}\mathbf{e}_p\right]^T \mathbf{e}_p \\
&= \nabla_p W(\boldsymbol{\alpha}) + \eta \mathbf{e}_p^T \mathbf{K}\mathbf{e}_p = 0 \Rightarrow \eta = \frac{-\nabla_p W(\boldsymbol{\alpha})}{\mathbf{K}_{pp}}
\end{aligned}
$$

- Update: $\alpha_p^{new} = \left[\alpha_p^{old} - \frac{\nabla_p W(\boldsymbol{\alpha})}{\mathbf{K}_{pp}}\right]_0^C$

Picking α to be updated

Picking α to be updated

- Largest change in (dual) objective function.

Picking α to be updated

- Largest change in (dual) objective function.
- Let $\delta = \alpha_p^{new} - \alpha_p^{old}$,

Picking α to be updated

- Largest change in (dual) objective function.
- Let $\delta = \alpha_p^{new} - \alpha_p^{old}$,

$$\Delta W = W(\boldsymbol{\alpha}) - W(\boldsymbol{\alpha} + \delta \mathbf{e}_p)$$

Picking α to be updated

- Largest change in (dual) objective function.
- Let $\delta = \alpha_p^{new} - \alpha_p^{old}$,

$$\begin{aligned}\Delta W &= W(\boldsymbol{\alpha}) - W(\boldsymbol{\alpha} + \delta \mathbf{e}_p) \\ &= \left(\frac{1}{2}\boldsymbol{\alpha}^T \mathbf{K}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{\tau}\right) \\ &- \left(\frac{1}{2}(\boldsymbol{\alpha} + \delta \mathbf{e}_p)^T \mathbf{K}(\boldsymbol{\alpha} + \mathbf{e}_P \delta) - (\boldsymbol{\alpha} + \delta \mathbf{e}_p)^T \boldsymbol{\tau}\right)\end{aligned}$$

Picking α to be updated

- Largest change in (dual) objective function.
- Let $\delta = \alpha_p^{new} - \alpha_p^{old}$,

$$\begin{aligned}\Delta W &= W(\boldsymbol{\alpha}) - W(\boldsymbol{\alpha} + \delta \mathbf{e}_p) \\ &= \left(\frac{1}{2}\boldsymbol{\alpha}^T \mathbf{K}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{\tau}\right) \\ &- \left(\frac{1}{2}(\boldsymbol{\alpha} + \delta \mathbf{e}_p)^T \mathbf{K}(\boldsymbol{\alpha} + \mathbf{e}_P \delta) - (\boldsymbol{\alpha} + \delta \mathbf{e}_p)^T \boldsymbol{\tau}\right) \\ &= -\frac{\delta^2}{2}\mathbf{e}_p^T \mathbf{K}\mathbf{e}_p - \delta \mathbf{e}_p^T \mathbf{K}\boldsymbol{\alpha} + \delta \mathbf{e}_p^T \boldsymbol{\tau}\end{aligned}$$

Picking α to be updated

- Largest change in (dual) objective function.
- Let $\delta = \alpha_p^{new} - \alpha_p^{old}$,

$$
\begin{aligned}
\Delta W &= W(\boldsymbol{\alpha}) - W(\boldsymbol{\alpha} + \delta \mathbf{e}_p) \\
&= \left(\frac{1}{2}\boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{\tau}\right) \\
&- \left(\frac{1}{2}(\boldsymbol{\alpha} + \delta \mathbf{e}_p)^T \mathbf{K} (\boldsymbol{\alpha} + \mathbf{e}_P \delta) - (\boldsymbol{\alpha} + \delta \mathbf{e}_p)^T \boldsymbol{\tau}\right) \\
&= -\frac{\delta^2}{2} \mathbf{e}_p^T \mathbf{K} \mathbf{e}_p - \delta \mathbf{e}_p^T \mathbf{K} \boldsymbol{\alpha} + \delta \mathbf{e}_p^T \boldsymbol{\tau} \\
&= -\frac{\delta^2 \mathbf{K}_{pp}}{2} - \delta \nabla_p W(\boldsymbol{\alpha}) = -\delta \left(\frac{\delta \mathbf{K}_{pp}}{2} + \nabla_p W(\boldsymbol{\alpha})\right)
\end{aligned}
$$

Picking α to be updated

- Largest change in (dual) objective function.
- Let $\delta = \alpha_p^{new} - \alpha_p^{old}$,

$$\begin{aligned}\Delta W &= W(\boldsymbol{\alpha}) - W(\boldsymbol{\alpha} + \delta \mathbf{e}_p) \\ &= \left(\frac{1}{2}\boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{\tau}\right) \\ &- \left(\frac{1}{2}(\boldsymbol{\alpha} + \delta \mathbf{e}_p)^T \mathbf{K}(\boldsymbol{\alpha} + \mathbf{e}_P \delta) - (\boldsymbol{\alpha} + \delta \mathbf{e}_p)^T \boldsymbol{\tau}\right) \\ &= -\frac{\delta^2}{2}\mathbf{e}_p^T \mathbf{K} \mathbf{e}_p - \delta \mathbf{e}_p^T \mathbf{K} \boldsymbol{\alpha} + \delta \mathbf{e}_p^T \boldsymbol{\tau} \\ &= -\frac{\delta^2 \mathbf{K}_{pp}}{2} - \delta \nabla_p W(\boldsymbol{\alpha}) = -\delta \left(\frac{\delta \mathbf{K}_{pp}}{2} + \nabla_p W(\boldsymbol{\alpha})\right)\end{aligned}$$

- Pick p for largest ΔW

Updating ∇W

$$\nabla W(\boldsymbol{\alpha}^{new}) = \mathbf{K}(\boldsymbol{\alpha} + \delta \mathbf{e}_p) - \tau = \nabla W(\boldsymbol{\alpha}^{old}) + \delta \mathbf{K}_p$$

Algoritmo del perceptrón

Algorithm 2 Perceptrón

Incialize $\mathbf{w}_0 = 0$
for $t = 1, 2, \ldots$ **do**
  Observe $\mathbf{x}_t$
  Prediga $\hat{y}_t = \langle \mathbf{w}_{t-1}, \mathbf{x}_t \rangle$
  Reciba y_t
  if $\hat{y}_t y_t < 0$ **then**
    $\mathbf{w}_t = \mathbf{w}_{t-1} + \mathbf{x}_t y_t$
  end if
end for

Aprendizaje en línea con Kernels

Aprendizaje en línea con Kernels

- Sea $\mathscr{L}$ el conjunto de los $\mathbf{x}$ para los cuales el algortimo del Perceptron ha cometido un error

Aprendizaje en línea con Kernels

- Sea $\mathscr{L}$ el conjunto de los $\mathbf{x}$ para los cuales el algortimo del Perceptron ha cometido un error
- El predictor obtenido por el algoritmo del Perceptrón tiene la forma:

$$f(\mathbf{x}) = \left\langle \sum_{y_i \mathbf{x}_i \in \mathscr{L}} \mathbf{x}_i, \mathbf{x} \right\rangle$$

Aprendizaje en línea con Kernels

- Sea $\mathscr{L}$ el conjunto de los $\mathbf{x}$ para los cuales el algortimo del Perceptron ha cometido un error
- El predictor obtenido por el algoritmo del Perceptrón tiene la forma:

$$f(\mathbf{x}) = \left\langle \sum_{y_i \mathbf{x}_i \in \mathscr{L}} \mathbf{x}_i, \mathbf{x} \right\rangle = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i \langle \mathbf{x}_i, \mathbf{x} \rangle$$

- Datos $\mathbf{x}_i$ sólo aparecen en productos punto

Aprendizaje en línea con Kernels

- Sea $\mathscr{L}$ el conjunto de los $\mathbf{x}$ para los cuales el algortimo del Perceptron ha cometido un error
- El predictor obtenido por el algoritmo del Perceptrón tiene la forma:

$$f(\mathbf{x}) = \left\langle \sum_{y_i \mathbf{x}_i \in \mathscr{L}} \mathbf{x}_i, \mathbf{x} \right\rangle = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i \langle \mathbf{x}_i, \mathbf{x} \rangle$$

- Datos $\mathbf{x}_i$ sólo aparecen en productos punto $\Rightarrow$ truco del kernel!

Aprendizaje en línea con Kernels

- Sea $\mathscr{L}$ el conjunto de los $\mathbf{x}$ para los cuales el algortimo del Perceptron ha cometido un error
- El predictor obtenido por el algoritmo del Perceptrón tiene la forma:

$$f(\mathbf{x}) = \left\langle \sum_{y_i \mathbf{x}_i \in \mathscr{L}} \mathbf{x}_i, \mathbf{x} \right\rangle = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i \langle \mathbf{x}_i, \mathbf{x} \rangle$$

- Datos $\mathbf{x}_i$ sólo aparecen en productos punto $\Rightarrow$ truco del kernel!
- Es decir, podemos operar en espacio de características usando un kernel:

Aprendizaje en línea con Kernels

- Sea $\mathscr{L}$ el conjunto de los $\mathbf{x}$ para los cuales el algortimo del Perceptron ha cometido un error
- El predictor obtenido por el algoritmo del Perceptrón tiene la forma:

$$f(\mathbf{x}) = \left\langle \sum_{y_i \mathbf{x}_i \in \mathscr{L}} \mathbf{x}_i, \mathbf{x} \right\rangle = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i \langle \mathbf{x}_i, \mathbf{x} \rangle$$

- Datos $\mathbf{x}_i$ sólo aparecen en productos punto $\Rightarrow$ truco del kernel!
- Es decir, podemos operar en espacio de características usando un kernel:

$$f(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i \left\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}) \right\rangle_{\mathscr{H}}$$

Aprendizaje en línea con Kernels

- Sea $\mathscr{L}$ el conjunto de los $\mathbf{x}$ para los cuales el algortimo del Perceptron ha cometido un error
- El predictor obtenido por el algoritmo del Perceptrón tiene la forma:

$$f(\mathbf{x}) = \left\langle \sum_{y_i \mathbf{x}_i \in \mathscr{L}} \mathbf{x}_i, \mathbf{x} \right\rangle = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i \langle \mathbf{x}_i, \mathbf{x} \rangle$$

- Datos $\mathbf{x}_i$ sólo aparecen en productos punto $\Rightarrow$ truco del kernel!
- Es decir, podemos operar en espacio de características usando un kernel:

$$f(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i \left\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}) \right\rangle_{\mathscr{H}} = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i k(\mathbf{x}_i, \mathbf{x})$$

Aprendizaje en línea con Kernels

- Sea $\mathscr{L}$ el conjunto de los $\mathbf{x}$ para los cuales el algortimo del Perceptron ha cometido un error
- El predictor obtenido por el algoritmo del Perceptrón tiene la forma:

$$f(\mathbf{x}) = \left\langle \sum_{y_i \mathbf{x}_i \in \mathscr{L}} \mathbf{x}_i, \mathbf{x} \right\rangle = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i \langle \mathbf{x}_i, \mathbf{x} \rangle$$

- Datos $\mathbf{x}_i$ sólo aparecen en productos punto $\Rightarrow$ truco del kernel!
- Es decir, podemos operar en espacio de características usando un kernel:

$$f(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i \left\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}) \right\rangle_{\mathscr{H}} = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i k(\mathbf{x}_i, \mathbf{x})$$

Perceptrón con Kernel

Algorithm 3 Kernel Perceptrón

Perceptrón con Kernel

Algorithm 4 Kernel Perceptrón

Require: Kernel k

Perceptrón con Kernel

Algorithm 5 Kernel Perceptrón

Require: Kernel k

  Incialize $\mathscr{L} = \phi$

Perceptrón con Kernel

Algorithm 6 Kernel Perceptrón

Require: Kernel k

Incialize $\mathscr{L} = \phi$

for $t = 1, 2, \ldots$ **do**

Perceptrón con Kernel

Algorithm 7 Kernel Perceptrón

Require: Kernel k

Incialize $\mathscr{L} = \phi$

for $t = 1, 2, \ldots$ **do**

Observe $\mathbf{x}_t$

Perceptrón con Kernel

Algorithm 8 Kernel Perceptrón

Require: Kernel k

Incialize $\mathscr{L} = \phi$

for $t = 1, 2, \ldots$ **do**

Observe $\mathbf{x}_t$

Prediga

$$\hat{y}_t = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i k(\mathbf{x}_i, \mathbf{x}_t)$$

Perceptrón con Kernel

Algorithm 9 Kernel Perceptrón

Require: Kernel k

Incialize $\mathscr{L} = \phi$

for $t = 1, 2, \ldots$ **do**

Observe $\mathbf{x}_t$

Prediga

$$\hat{y}_t = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i k(\mathbf{x}_i, \mathbf{x}_t)$$

Reciba y_t

Perceptrón con Kernel

Algorithm 10 Kernel Perceptrón

Require: Kernel k

Incialize $\mathscr{L} = \phi$

for $t = 1, 2, \ldots$ **do**

Observe $\mathbf{x}_t$

Prediga

$$\hat{y}_t = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i k(\mathbf{x}_i, \mathbf{x}_t)$$

Reciba y_t

if $\hat{y}_t y_t < 0$ **then**

Perceptrón con Kernel

Algorithm 11 Kernel Perceptrón

Require: Kernel k

Incialize $\mathscr{L} = \phi$

for $t = 1, 2, \ldots$ **do**

Observe $\mathbf{x}_t$

Prediga

$$\hat{y}_t = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i k(\mathbf{x}_i, \mathbf{x}_t)$$

Reciba y_t

if $\hat{y}_t y_t < 0$ **then**

$\mathscr{L} \leftarrow \mathscr{L} \cup \{(\mathbf{x}_t, y_t)\}$

Perceptrón con Kernel

Algorithm 12 Kernel Perceptrón

Require: Kernel k

Incialize $\mathscr{L} = \phi$

for $t = 1, 2, \ldots$ **do**

Observe $\mathbf{x}_t$

Prediga

$$\hat{y}_t = \sum_{\mathbf{x}_i \in \mathscr{L}} y_i k(\mathbf{x}_i, \mathbf{x}_t)$$

Reciba y_t

if $\hat{y}_t y_t < 0$ **then**

$\mathscr{L} \leftarrow \mathscr{L} \cup \left\{(\mathbf{x}_t, y_t)\right\}$

end if

end for