C: regularization parameter tol: numerical tolerance max_passes : max # of times to iterate over α 's without changing $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$: training data

Output:

 $\alpha \in \mathbb{R}^m$: Lagrange multipliers for solution

 $b \in \mathbb{R}$: threshold for solution

C: regularization parameter tol: numerical tolerance max_passes : max # of times to iterate over α 's without changing $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$: training data

Output:

 $\alpha \in \mathbb{R}^m$: Lagrange multipliers for solution

 $b \in \mathbb{R}$: threshold for solution

- \circ Initialize $\alpha_i = 0, \forall i, b = 0.$
- \circ Initialize passes = 0.

• **for**
$$i = 1, ... m$$
,

- \circ Calculate $E_i = f(x^{(i)}) y^{(i)}$ using (2).
- \circ if $((y^{(i)}E_i < -tol && & \alpha_i < C) || (y^{(i)}E_i > tol && & \alpha_i > 0))$

C: regularization parameter tol: numerical tolerance max_passes : max # of times to iterate over α 's without changing $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$: training data

Output:

 $\alpha \in \mathbb{R}^m$: Lagrange multipliers for solution

 $b \in \mathbb{R}$: threshold for solution

$$\circ$$
 Initialize $\alpha_i = 0, \forall i, b = 0.$

$$\circ$$
 Initialize $passes = 0$.

$$f(x) = \langle \omega, x \rangle + b$$

$$f(x) = \langle \omega, x \rangle + b$$

$$\omega = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$

$$\omega = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$

$$\circ$$
 for $i=1,\ldots m$,

$$\circ$$
 Calculate $E_i = f(x^{(i)}) - y^{(i)}$ using (2).

$$\circ$$
 if $((y^{(i)}E_i < -tol \&\& \alpha_i < C) || (y^{(i)}E_i > tol \&\& \alpha_i > 0))$

C: regularization parameter tol: numerical tolerance max_passes : max # of times to iterate over α 's without changing $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$: training data

Output:

 $\alpha \in \mathbb{R}^m$: Lagrange multipliers for solution

 $b \in \mathbb{R}$: threshold for solution

$$\circ$$
 Initialize $\alpha_i = 0, \forall i, b = 0.$

$$\circ$$
 Initialize $passes = 0$.

$$f(x) = \langle \omega, x \rangle + b$$

$$f(x) = \langle \omega, x \rangle + b$$
$$\omega = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$

$$\circ$$
 for $i = 1, ... m$,

 \circ Calculate $E_i = f(x^{(i)}) - y^{(i)}$ using (2).

$$\circ$$
 if $((y^{(i)}E_i < -tol && \alpha_i < C) || (y^{(i)}E_i > tol && \alpha_i > 0))$

- \circ Select $j \neq i$ randomly.
- \circ Calculate $E_j = f(x^{(j)}) y^{(j)}$ using (2).
- Save old α 's: $\alpha_i^{\text{(old)}} = \alpha_i$, $\alpha_j^{\text{(old)}} = \alpha_j$.
- \circ Compute L and H by (10) or (11).
- \circ if (L == H)

continue to next i.

C: regularization parameter tol: numerical tolerance max_passes : max # of times to iterate over α 's without changing $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$: training data

Output:

 $\alpha \in \mathbb{R}^m$: Lagrange multipliers for solution

 $b \in \mathbb{R}$: threshold for solution

$$\circ$$
 Initialize $\alpha_i = 0, \forall i, b = 0.$

$$\circ$$
 Initialize $passes = 0$.

$$f(x) = \langle \omega, x \rangle + b$$

$$f(x) = \langle \omega, x \rangle + b$$
$$\omega = \sum_{i=0}^{m} \alpha_i y^{(i)} x^{(i)}$$

$$\circ$$
 for $i = 1, ..., m$,

$$\circ$$
 Calculate $E_i = f(x^{(i)}) - y^{(i)}$ using (2).

$$\circ$$
 if $((y^{(i)}E_i < -tol \&\& \alpha_i < C) || (y^{(i)}E_i > tol \&\& \alpha_i > 0))$

- \circ Select $j \neq i$ randomly.
- \circ Calculate $E_i = f(x^{(j)}) y^{(j)}$ using (2).
- Save old α 's: $\alpha_i^{\text{(old)}} = \alpha_i$, $\alpha_j^{\text{(old)}} = \alpha_j$.
- \circ Compute L and H by (10) or (11).
- \circ if (L == H)

continue to next i.

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

$$\alpha_2 = C$$

$$\alpha_1 = 0$$

$$\alpha_2 = 0$$

$$\alpha_1 = C$$

$$\alpha_2 = C$$

$$\alpha_1 = 0$$

$$\alpha_2 = 0$$

$$\alpha_2 = 0$$

$$y_1 \neq y_2 \Rightarrow \alpha_1 - \alpha_2 = \gamma$$

$$y_1 \neq y_2 \Rightarrow \alpha_1 - \alpha_2 = \gamma$$
 $y_1 = y_2 \Rightarrow \alpha_1 + \alpha_2 = \gamma$

C: regularization parameter tol: numerical tolerance $max_passes: max \# of times to iterate over \alpha's without changing$ $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$: training data

Output:

 $\alpha \in \mathbb{R}^m$: Lagrange multipliers for solution

 $b \in \mathbb{R}$: threshold for solution

∘ Initialize
$$\alpha_i = 0, \forall i, b = 0.$$

∘ Initialize $passes = 0.$

$$f(x) = \langle \omega, x \rangle + b$$

$$f(x) = \langle \omega, x \rangle + b$$
$$\omega = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$

$$\circ$$
 for $i = 1, ... m$,

- \circ Calculate $E_i = f(x^{(i)}) y^{(i)}$ using (2).
- \circ if $((y^{(i)}E_i < -tol \&\& \alpha_i < C) || (y^{(i)}E_i > tol \&\& \alpha_i > 0))$
 - \circ Select $i \neq i$ randomly.
 - \circ Calculate $E_j = f(x^{(j)}) y^{(j)}$ using (2).
 - Save old α 's: $\alpha_i^{\text{(old)}} = \alpha_i$, $\alpha_j^{\text{(old)}} = \alpha_j$.
 - \circ Compute L and H by (10) or (11).
 - \circ if (L == H)

continue to next i.

 \circ Compute and clip new value for α_i

$$\alpha_j := \begin{cases} H & \text{if } \alpha_j > H \\ \alpha_j & \text{if } L \le \alpha_j \le H \\ L & \text{if } \alpha_j < L. \end{cases}$$

C: regularization parameter tol: numerical tolerance $max_passes: max \# of times to iterate over \alpha's without changing$ $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$: training data

Output:

 $\alpha \in \mathbb{R}^m$: Lagrange multipliers for solution

 $b \in \mathbb{R}$: threshold for solution

- \circ Initialize $\alpha_i = 0, \forall i, b = 0.$
- \circ Initialize passes = 0.

$$f(x) = \langle \omega, x \rangle + b$$
$$\omega = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$

$$\omega = \sum \alpha_i y^{(i)} x^{(i)}$$

- \circ for i = 1, ..., m,
 - \circ Calculate $E_i = f(x^{(i)}) y^{(i)}$ using (2).
 - \circ if $((y^{(i)}E_i < -tol \&\& \alpha_i < C) || (y^{(i)}E_i > tol \&\& \alpha_i > 0))$
 - \circ Select $i \neq i$ randomly.
 - \circ Calculate $E_j = f(x^{(j)}) y^{(j)}$ using (2).
 - Save old α 's: $\alpha_i^{\text{(old)}} = \alpha_i$, $\alpha_j^{\text{(old)}} = \alpha_j$.
 - \circ Compute L and H by (10) or (11).
 - \circ if (L == H)

continue to next i.

- \circ Compute and clip new value for α_i
- \circ if $(|\alpha_j \alpha_j^{\text{(old)}}| < 10^{-5})$

continue to next i.

 \circ Determine value for α_i

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

C: regularization parameter tol: numerical tolerance $max_passes: max \# of times to iterate over \alpha's without changing$ $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$: training data

Output:

 $\alpha \in \mathbb{R}^m$: Lagrange multipliers for solution

 $b \in \mathbb{R}$: threshold for solution

- \circ Initialize $\alpha_i = 0, \forall i, b = 0.$
- \circ Initialize passes = 0.

$$f(x) = \langle \omega, x \rangle + b$$
$$\omega = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$

$$\omega = \sum \epsilon$$

 \circ for $i=1,\ldots m$,

- \circ Calculate $E_i = f(x^{(i)}) y^{(i)}$ using (2).
- \circ if $((y^{(i)}E_i < -tol \&\& \alpha_i < C) || (y^{(i)}E_i > tol \&\& \alpha_i > 0))$
 - \circ Select $i \neq i$ randomly.
 - \circ Calculate $E_j = f(x^{(j)}) y^{(j)}$ using (2).
 - Save old α 's: $\alpha_i^{\text{(old)}} = \alpha_i$, $\alpha_i^{\text{(old)}} = \alpha_j$.
 - \circ Compute L and H by (10) or (11).
 - \circ if (L == H)

continue to next i.

- \circ Compute and clip new value for α_i
- \circ if $(|\alpha_j \alpha_j^{\text{(old)}}| < 10^{-5})$ **continue** to next i.
- \circ Determine value for α_i
- \circ Compute b

C: regularization parameter tol: numerical tolerance max_passes : max # of times to iterate over α 's without changing $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$: training data

Output:

 $\alpha \in \mathbb{R}^m$: Lagrange multipliers for solution $b \in \mathbb{R}$: threshold for solution

 \circ if (L == H)

continue to next i.

$$\begin{array}{l} \circ \text{ Initialize } \alpha_{i} = 0, \forall i, \quad b = 0. \\ \circ \text{ Initialize } passes = 0. \end{array} \qquad \qquad f(x) = \langle \omega, x \rangle + b \\ \circ \text{ Initialize } passes = 0. \end{array} \\ \begin{array}{l} \omega = \sum_{i=1}^{m} \alpha_{i} y^{(i)} x^{(i)} \\ \circ \text{ for } i = 1, \ldots m, \\ \circ \text{ Calculate } E_{i} = f(x^{(i)}) - y^{(i)} \text{ using } (2). \\ \circ \text{ if } ((y^{(i)} E_{i} < -tol & \&\& \quad \alpha_{i} < C) \mid\mid (y^{(i)} E_{i} > tol & \&\& \quad \alpha_{i} > 0)) \\ \circ \text{ Select } j \neq i \text{ randomly.} \\ \circ \text{ Calculate } E_{j} = f(x^{(j)}) - y^{(j)} \text{ using } (2). \\ \circ \text{ Save old } \alpha \text{ 's: } \alpha_{i}^{(\text{old})} = \alpha_{i}, \quad \alpha_{j}^{(\text{old})} = \alpha_{j}. \\ \circ \text{ Compute } L \text{ and } H \text{ by } (10) \text{ or } (11). \end{array}$$

Compute and clip new value for α_j
if (|α_j - α_j^(old)| < 10⁻⁵)
continue to next i.
Determine value for α_i
Compute b

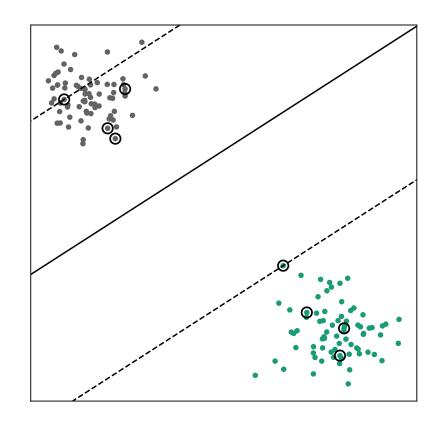
 $\circ num_changed_alphas := num_changed_alphas + 1.$ \circ end if

 \circ end \circ end for

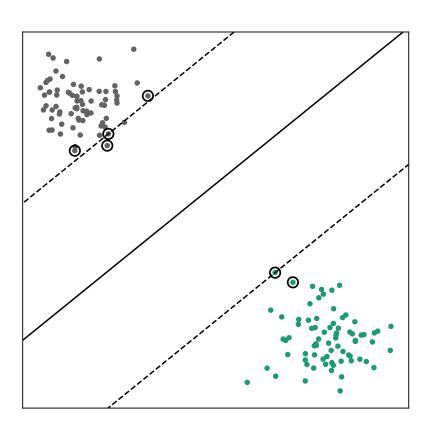
Input: C: regularization parameter tol: numerical tolerance max_passes : max # of times to iterate over α 's without changing $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$: training data **Output:** $\alpha \in \mathbb{R}^m$: Lagrange multipliers for solution $b \in \mathbb{R}$: threshold for solution

 $f(x) = \langle \omega, x \rangle + b$ $\omega = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$ \circ Initialize $\alpha_i = 0, \forall i, b = 0.$ \circ Initialize passes = 0. \circ while (passes < max_passes) $\circ num_changed_alphas = 0.$ \circ for i = 1, ... m, \circ Calculate $E_i = f(x^{(i)}) - y^{(i)}$ using (2). \circ if $((y^{(i)}E_i < -tol \&\& \alpha_i < C) || (y^{(i)}E_i > tol \&\& \alpha_i > 0))$ \circ Select $i \neq i$ randomly. \circ Calculate $E_i = f(x^{(j)}) - y^{(j)}$ using (2). • Save old α 's: $\alpha_i^{\text{(old)}} = \alpha_i$, $\alpha_j^{\text{(old)}} = \alpha_j$. \circ Compute L and H by (10) or (11). \circ if (L == H)**continue** to next i. \circ Compute and clip new value for α_i \circ if $(|\alpha_j - \alpha_j^{\text{(old)}}| < 10^{-5})$ **continue** to next i. \circ Determine value for α_i \circ Compute b $\circ num_changed_alphas := num_changed_alphas + 1.$ \circ end if \circ end for \circ if $(num_changed_alphas == 0)$ passes := passes + 1 \circ else passes := 0

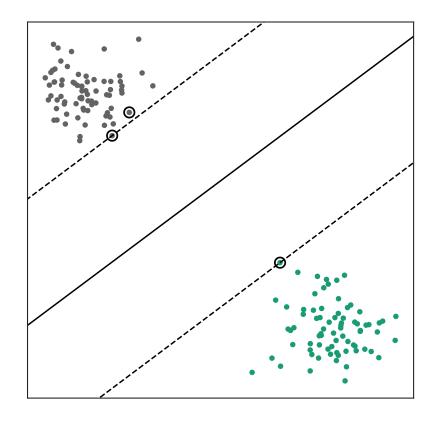
o end while



max_passes = 1



max_passes = 15



max_passes = 30