A. The Proof of Theorem 2

Here, we prove the theorem by explicitly deriving our non-SV screening rule in section 3.

The proof First, we proove (15) and (16) by solving the minimization problem in (13) with the Lagrange multiplier method. The minimization problem can be calculated as follows:

$$\begin{aligned}
&\min_{w\in\Theta_{[s_b,s_a]}} y_i f(x_i)\\
&= \min_{w} w^\top y_i (w^\top x_i)\\
&\quad \text{s.t. } \frac{1}{2}||w||^2 \le \gamma_b, w^*(s_a)^\top (w - w^*(s_a)) \ge 0\\
&= \min_{w}\Big(\max_{\mu\ge 0,\nu\ge 0}\big(y_i w^\top x_i\\
&\quad + \mu(\frac{1}{2}||w||^2 - \gamma_b) - \nu w^*(s_a)^\top (w - w^*(s_a))\big)\Big)\\
&= \max_{\mu\ge 0,\nu\ge 0}\Big(-\mu\gamma_b + \nu||w^*(s_a)||^2\\
&\quad + \min_{w}\big(\frac{\mu}{2}||w||^2 + (y_i x_i - \nu w^*(s_a))^\top w\big)\Big)\\
&= \max_{\mu\ge 0,\nu\ge 0}\Big(-\mu\gamma_b + \nu||w^*(s_a)||^2\\
&\quad - \frac{1}{2\mu}||y_i x_i - \nu w^*(s_a)||^2\Big),
\end{aligned}$$

where $\mu \ge 0$ and $\nu \ge 0$ are the Lagrange multipliers, and the maximand in the last line is so-called Lagrangian:

$$\begin{aligned}
L(\mu,\nu) :=& -\mu\gamma_b + \nu||w^*(s_a)||^2\\
&- \frac{1}{2\mu}||y_i x_i - \nu w^*(s_a)||^2. \qquad (22)
\end{aligned}$$

At the optimal solution, we must satisfy

$$\frac{\partial L}{\partial \nu} = 0 \;\Leftrightarrow\; 2\gamma_a + \frac{y_i f^*(x_i|s_a) - 2\nu\gamma_a}{\mu} = 0.$$

Since the Lagrange multiplier $\nu \ge 0$, we can write

$$\nu = \max\big(0, \frac{y_i f^*(x_i|s_a)}{2\gamma_a} + \mu\big).$$

Thus, if $\mu \le -\frac{y_i f^*(x_i|s_a)}{2\gamma_a}$, then

$$\nu = 0. \qquad (23)$$

In this case, noting that the Lagrange multiplier $\mu \ge 0$, we have

$$\frac{\partial L}{\partial \mu} = 0 \;\Leftrightarrow\; \mu = \frac{||x_i||}{\sqrt{2\gamma_b}}. \qquad (24)$$

On the other hand, if $\mu > -\frac{y_i f^*(x_i|s_a)}{2\gamma_a}$, then

$$\nu = \frac{y_i f^*(x_i|s_a)}{2\gamma_a} + \mu. \qquad (25)$$

In this case, we have

$$\frac{\partial L}{\partial \mu} = 0 \;\Leftrightarrow\; \mu = \frac{1}{2}\sqrt{\frac{2||x_i||^2\gamma_a - f^*(x_i|s_a)^2}{\gamma_a\gamma_b - \gamma_a^2}}. \qquad (26)$$

By plugging the pair of (μ,ν) in ((24), (23)) or in ((26), (25)) into $L(\mu,\nu)$ in (22), the lower bound can be summarized as

$$\begin{aligned}
&\min_{w\in\Theta_{[s_b,s_a]}} y_i f(x_i)\\
&:= \begin{cases}
-\sqrt{2\gamma_b}||x_i||, \\
\qquad \text{if } -\frac{y_i f^*(x_i|s_a)}{||x_i||} \ge \frac{\sqrt{2}\gamma_a}{\sqrt{\gamma_b}}, \\
y_i f^*(x_i|s_a) \\
\quad - \sqrt{\frac{\gamma_b-\gamma_a}{\gamma_a}(2\gamma_a||x_i||^2 - f^*(x_i|s_a)^2)}, \\
\qquad \text{otherwise.}
\end{cases}
\end{aligned}$$

In the former case when $-\frac{y_i f^*(x_i|s_a)}{||x_i||} \ge \frac{\sqrt{2}\gamma_a}{\sqrt{\gamma_b}}$, the lower bound $\min_{\Theta_{[s_b,s_a]}} y_i f(x_i) = -\sqrt{2\gamma_b}||x_i||^2$ is clearly less than 1, and there is no hope to screen out this instance as a non-SV. It suggests that we can only consider the latter case. In addition, if we look into the lower bound in the latter case, the first term is the value of $y_i f(x_i)$ at s_a, while the second squared-root term is obviously non-negative. It means that we have a chance to screen out the ith instance only when $y_i f^*(x_i|s_a) > 1$. This is why the screening rule for $\mathcal{R}$ has the form in (15) and (16).

Next, we prove (15) and (16). In the same way, we can write the upper bound of $y_i f(x_i)$ as follows:

$$\begin{aligned}
&\max_{w\in\Theta_{[s_b,s_a]}} y_i f(x_i)\\
&:= \begin{cases}
\sqrt{2\gamma_b}||x_i||, \\
\qquad \text{if } \frac{y_i f^*(x_i|s_a)}{||x_i||} \ge \frac{\sqrt{2}\gamma_a}{\sqrt{\gamma_b}}, \\
y_i f^*(x_i|s_a) \\
\quad + \sqrt{\frac{\gamma_b-\gamma_a}{\gamma_a}(2\gamma_a||x_i||^2 - f^*(x_i|s_a)^2)}, \\
\qquad \text{otherwise.}
\end{cases}
\end{aligned}$$

Here, we might have a chance to screen out the ith instance also in the former case when $\frac{y_i f^*(x_i|s_a)}{||x_i||} \ge \frac{\sqrt{2}\gamma_a}{\sqrt{\gamma_b}}$. In the latter case, we only have a chance to screen out the ith instance if $y_i f^*(x_i|s_a) < 1$ since the second squared-root term is non-negative. However, for notational simplicity, we just write the screening rule for $\mathcal{L}$ in the form of (17) and (18). **Q.E.D.**

B. The Proof of Lemma 3

In the proof, we use index sets to represent subvectors and submatrices. For example, $v_{\mathcal{A}}$ for a vector v indicates a subvector of v having only the elements in the index set $\mathcal{A}$, and $M_{\mathcal{A},\mathcal{B}}$ for a matrix M indicates a submatrix of M having only the rows in the index set $\mathcal{A}$ and the columns in the index set $\mathcal{B}$. In addition, we write $\mathbf{1}_n$ to represent an n-dimensional vector of 1s.

In order to prove Lemmas 3, we first summarize the optimality condition of the problem (7) in s-form. Then, based on them, we clarify the sensitivity of C in C-form to s in s-form.

Proposition 4 *Consider the optimal solution $f^*(\cdot|s)$ of the problem (7) at a certain s (we assume that there exists a feasible solution with the s). Then, the dual problem is formulated as*

$$\max_{\alpha,C} \ -\frac{1}{2}\alpha^\top Q\alpha + \mathbf{1}^\top\alpha - Cs \text{ s.t. } \alpha \in [0,C]^n, \quad (27)$$

where $\alpha \in \mathbb{R}^n$ and $C \in \mathbb{R}$ is the Lagrange multipliers. Using these Lagrange multipliers, the optimal classifier is written as

$$f^*(x|s) = \sum_{i\in\mathbb{N}} y_i\alpha_i K(x,x_i), \quad (28)$$

If we define the following three index sets:

$$\mathcal{R} := \{i \in \mathbb{N} | y_i f^*(x_i|s) > 1\}, \quad (29a)$$
$$\mathcal{E} := \{i \in \mathbb{N} | y_i f^*(x_i|s) = 1\}, \quad (29b)$$
$$\mathcal{L} := \{i \in \mathbb{N} | y_i f^*(x_i|s) < 1\}, \quad (29c)$$

then, the Lagrange multipliers $\alpha \in \mathbb{R}^n$ satisfy

$$i \in \mathcal{R} \ \Rightarrow \ \alpha_i = 0, \quad (30a)$$
$$i \in \mathcal{E} \ \Rightarrow \ \alpha_i \in [0,C], \quad (30b)$$
$$i \in \mathcal{L} \ \Rightarrow \ \alpha_i = C, \quad (30c)$$

where, remember that, C is the optimal Lagrange multiplier for the constraint (7b) which is also obtained after solving the optimization problem (27).

At the optimal solution, unless all the training instances are correctly classified with the margin greater than 1 (i.e., unless the set $\mathcal{L}$ is empty), the constraint (7b) is active, i.e.,

$$\sum_{i\in\mathbb{N}} [1 - y_i f^*(x_i|s)] = s. \quad (31)$$

We omit the proof of this proposition because it is easily shown by using standard Lagrange multiplier theory (Boyd & Vandenberghe, 2004).

The Proof In order to prove the lemma, we derive the optimality conditions of the problem (20) at s. Given the three index set $\mathcal{L}$, $\mathcal{E}$, $\mathcal{R}$ in (30), the optimality conditions of the problem are written as

$$i \in \mathcal{R} \Rightarrow y_i f(x_i) > 1, \alpha_i = 0, \quad (32a)$$
$$i \in \mathcal{E} \Rightarrow y_i f(x_i) = 1, \alpha_i \in [0,C], \quad (32b)$$
$$i \in \mathcal{L} \Rightarrow y_i f(x_i) < 1, \alpha_i = C, \quad (32c)$$
$$\sum_{i\in\mathbb{N}} [1 - y_i f(x_i)]_+ = s. \quad (32d)$$

If we rewrite these conditions in matrix vector form, we have

$$Q_{\mathcal{R}\mathcal{E}}\alpha_{\mathcal{E}} + Q_{\mathcal{R}\mathcal{L}}\mathbf{1}_{|\mathcal{L}|}C > \mathbf{1}_{|\mathcal{R}|}, \quad (33a)$$
$$Q_{\mathcal{L}\mathcal{E}}\alpha_{\mathcal{E}} + Q_{\mathcal{L}\mathcal{L}}\mathbf{1}_{|\mathcal{L}|}C < \mathbf{1}_{|\mathcal{L}|}, \quad (33b)$$
$$Q_{\mathcal{E}\mathcal{E}}\alpha_{\mathcal{E}} + Q_{\mathcal{E}\mathcal{L}}\mathbf{1}_{|\mathcal{L}|}C = \mathbf{1}_{|\mathcal{E}|}, \quad (33c)$$
$$\mathbf{1}_{|\mathcal{L}|}^\top(\mathbf{1}_{|\mathcal{L}|} - Q_{\mathcal{L}\mathcal{E}}\alpha_{\mathcal{E}} - Q_{\mathcal{L}\mathcal{L}}\mathbf{1}_{|\mathcal{L}|}C) = s. \quad (33d)$$

From (33c),

$$\alpha_{\mathcal{E}} = Q_{\mathcal{E}\mathcal{E}}^{-1}\mathbf{1}_{|\mathcal{L}|} - Q_{\mathcal{E}\mathcal{E}}^{-1}Q_{\mathcal{E}\mathcal{L}}\mathbf{1}_{|\mathcal{L}|}C. \quad (34)$$

Substituting (34) into (33d), we have

$$C = -\frac{1}{\mathbf{1}_{|\mathcal{L}|}^\top(Q_{\mathcal{L}\mathcal{L}} - Q_{\mathcal{L}\mathcal{E}}Q_{\mathcal{E}\mathcal{E}}^{-1}Q_{\mathcal{E}\mathcal{L}})\mathbf{1}_{|\mathcal{L}|}}s + \text{constant}. \quad (35)$$

Since the matrix $Q_{\mathcal{L}\mathcal{L}} - Q_{\mathcal{L}\mathcal{E}}Q_{\mathcal{E}\mathcal{E}}^{-1}Q_{\mathcal{E}\mathcal{L}}$ is the Schur complement of the block $Q_{\mathcal{E}\mathcal{E}}$ of a positive semi-definite matrix

$$Q_{(\mathcal{E}\cup\mathcal{L})(\mathcal{E}\cup\mathcal{L})} = \begin{bmatrix} Q_{\mathcal{E}\mathcal{E}} & Q_{\mathcal{E}\mathcal{L}} \\ Q_{\mathcal{L}\mathcal{E}} & Q_{\mathcal{L}\mathcal{L}} \end{bmatrix}, \quad (36)$$

this is also positive semi-definite (Boyd & Vandenberghe, 2004). It indicates that

$$\frac{\partial C}{\partial s} = -\frac{1}{\mathbf{1}_{|\mathcal{L}|}^\top(Q_{\mathcal{L}\mathcal{L}} - Q_{\mathcal{L}\mathcal{E}}Q_{\mathcal{E}\mathcal{E}}^{-1}Q_{\mathcal{E}\mathcal{L}})\mathbf{1}_{|\mathcal{L}|}} \leq 0. \quad (37)$$

It suggests that, as long as the three index sets $\mathcal{L}, \mathcal{E}, \mathcal{R}$ are unchanged, C is linearly decreasing with s. When one or more instances move from one index set to the other, then, the decreasing rate $-\frac{1}{\mathbf{1}_{|\mathcal{L}|}^\top(Q_{\mathcal{L}\mathcal{L}} - Q_{\mathcal{L}\mathcal{E}}Q_{\mathcal{E}\mathcal{E}}^{-1}Q_{\mathcal{E}\mathcal{L}})\mathbf{1}_{|\mathcal{L}|}}$ might be changed, but C is still linearly decreasing with s. It follows that C is a piecewise-linear function of s and each linear segment has non-positive slope, meaning that C is monotonically decreasing with s. **Q.E.D.**

C. Additional information of Pathwise Computation Algorithm

Here, we provide the pseudo-codes of FindSb and ComputeSubPath functions in Algorithm 2 and some additional information about the algorithm.

Algorithm 3 FindSb

1: **Input:** $D_{\mathbb{N}}$, $C^{(t+M)}$, $w^*(s_a)$;
2: **Output:** s_b, $w^*(s_b)$;
3: $s_b \leftarrow$ InitializeSb;
4: `isSbSafe` $\leftarrow$ `false`;
5: **while** `isSbSafe` = `false` **do**
6: Compute $\hat{w}(s_b)$ by solving (21);
7: $\{\tilde{\mathcal{L}}, \tilde{\mathcal{R}}, \tilde{\mathcal{Z}}\} \leftarrow$ Screen$(D_{\mathbb{N}}, w^*(s_a), \hat{w}(s_b))$;
8: $w^*(C^{(t+M)}) \leftarrow$ SVMSolver$(D_{\tilde{\mathcal{Z}}}, C^{(t+M)})$;
9: `isOpt` $\leftarrow$ CheckOptimality$(w^*(C^{(t+M)}))$;
10: **if** `isOpt` **then**
11: Compute $s^{(t+M)}$ by (19);
12: $s_b \leftarrow s^{(t+M)}$;
13: `isSbSafe` = `true`;
14: **else**
15: $s_b \leftarrow$ DecreaseSb(s_b);
16: **end if**
17: **end while**

Algorithm 4 ComputeSubPath

1: **Input:** $D_{\tilde{\mathcal{Z}}}$, $\{C^{(u)}\}_{u=u_1}^{u_2}$
2: **Output:** $\{w^*(C^{(u)})\}_{u=u_1}^{u_2}$;
3: $u \leftarrow u_1$
4: **while** $u \leq u_2$ **do**
5: $w^*(C^{(u)}) \leftarrow$ SVMSolver$(D_{\tilde{\mathcal{Z}}}, C^{(u)})$;
6: $u \leftarrow u + 1$;
7: **end while**

- We need to plug-in SVMSolver that can solve the SVM optimization problems (2) or (3). When we use this solver for the subset $D_{\tilde{\mathcal{Z}}}$ (the set of instances that are not screened out by the rule), it must take into account the fact that $\alpha_i = C$ for $i \in \tilde{\mathcal{L}}$. For simplicity, we do not explicitly write this dependency in the pseudo-code.

- At line 4, the optimal solution at $C^{(1)}$ is computed by using an SVM solver. This is because we do not have the optimal solution at any $C \leq C^{(1)}$. Training an SVM with very small C is usually very fast. Similarly, at line 6, the optimal solution at $C^{(T)}$ is computed. We compute this as a feasible solution at $C^{(T)}$, which is needed to efficiently compute other feasible solutions in later steps (see section 4.1).

- At line 11, FindSb function is called. This function returns the tightest upper bound of s_b, i.e., $s_b \equiv s^{(t+M)}$ and the optimal solution $w^*(s_b) \equiv w^*(s^{(t+M)})$. As described in Algorithm 3, this function is a bit complicated because we find $s_b \leq s^{(t+M)}$ in a trial-and-error manner. Actually, if we simply set $s_b = s^{(T)}$, this requirement is satisfied because $s^{(T)} \leq s^{(t+M)}$. However, in order to make the rule more effective, we would like to find a tighter lower bound of $s^{(t+M)}$ here. To this end, we first estimate an s_b using InitializeSb function. Then, we construct a non-SV screening rule, and the solution at $C^{(t+M)}$ is computed by using the rule. If the solution is optimal (confirmed by checking the optimality conditions), then we just set $s_b = s^{(t+M)}$. Otherwise, it indicates that our current estimate of s_b is greater than $s^{(t+M)}$, then we decrease it by DecreaseSb function, and try the above process again.

- At line 12, we construct a non-SV screening rule defined on $[s_b, s_a]$ which can be used of computing the pathwise solutions at $C^{(t+1)}, \ldots, C^{(t+M-1)}$. Computing these pathwise solutions with the subset $D_{\tilde{\mathcal{Z}}}$ is couducted in ComputeSubPath function described in Algorithm 4.