ML Session 2, Part 2: Support Vector Machines

Javen Shi

29 Nov. 2018

Table of Contents I

Refresh concepts

Inner product

For vectors $\mathbf{x} = [x^1, x^2, \cdots, x^d]^\top, \mathbf{w} = [w^1, w^2, \cdots, w^d]^\top$, inner product

$$\langle \mathbf{x}, \mathbf{w} \rangle = \sum_{i=1}^{d} x^i w^i.$$

We write $\mathbf{x} \in \mathbb{R}^d, \mathbf{w} \in \mathbb{R}^d$ to say they are d-dimensional real number vectors. We consider all vectors as column vectors by default. $\top$ is the transpose. We also use the matlab syntax that $[x^1;x^2;\cdots;x^d]$ as column vector.

Refresh concepts

Inner product

For vectors $\mathbf{x} = [x^1, x^2, \cdots, x^d]^\top, \mathbf{w} = [w^1, w^2, \cdots, w^d]^\top$, inner product

$$\langle \mathbf{x}, \mathbf{w} \rangle = \sum_{i=1}^{d} x^i w^i.$$

We write $\mathbf{x} \in \mathbb{R}^d, \mathbf{w} \in \mathbb{R}^d$ to say they are d-dimensional real number vectors. We consider all vectors as column vectors by default. $\top$ is the transpose. We also use the matlab syntax that $[x^1; x^2; \cdots; x^d]$ as column vector.

Example: $a = [1; 3; 1.5], b = [2; 1; 1]$. $\langle a, b \rangle = ?$

Refresh concepts

Inner product

For vectors $\mathbf{x} = [x^1, x^2, \cdots, x^d]^\top, \mathbf{w} = [w^1, w^2, \cdots, w^d]^\top$, inner product

$$\langle \mathbf{x}, \mathbf{w} \rangle = \sum_{i=1}^{d} x^i w^i.$$

We write $\mathbf{x} \in \mathbb{R}^d, \mathbf{w} \in \mathbb{R}^d$ to say they are d-dimensional real number vectors. We consider all vectors as column vectors by default. $\top$ is the transpose. We also use the matlab syntax that $[x^1;x^2;\cdots;x^d]$ as column vector.

Example: $a = [1; 3; 1.5], b = [2; 1; 1]$. $\langle a, b \rangle =?$
$= 1 \times 2 + 3 \times 1 + 1.5 \times 1 = 6.5$

Refresh concepts

Sign function

For any scalar $a \in \mathbb{R}$,

$$\text{sign}(a) = \begin{cases} 1 & \text{if } a > 0 \\ -1 & \text{otherwise} \end{cases}$$

Refresh concepts

Sign function

For any scalar $a \in \mathbb{R}$,

$$\text{sign}(a) = \begin{cases} 1 & \text{if } a > 0 \\ -1 & \text{otherwise} \end{cases}$$

Examples: $\text{sign}(20) = ?$, $\text{sign}(-5) = ?$, $\text{sign}(0) = ?$.

Refresh concepts

Sign function

For any scalar $a \in \mathbb{R}$,

$$\text{sign}(a) = \begin{cases} 1 & \text{if } a > 0 \\ -1 & \text{otherwise} \end{cases}$$

Examples: $\text{sign}(20) = ?$, $\text{sign}(-5) = ?$, $\text{sign}(0) = ?$.
$\text{sign}(20) = 1$, $\text{sign}(-5) = -1$, $\text{sign}(0) = -1$.

Decision functions

Typical decision functions for classification [1] :

$$\text{Binary-class} \quad g(\mathbf{x}; \mathbf{w}) = \operatorname{sign}(\langle \mathbf{x}, \mathbf{w} \rangle).$$

$$\text{Multi-class} \quad g(\mathbf{x}; \mathbf{w}) = \operatorname*{argmax}_{y \in \mathcal{Y}}(\langle \mathbf{x}, \mathbf{w}_y \rangle).$$

where $\mathbf{w}, \mathbf{w}_y$ are the parameters, and $\mathbf{x} \in \mathbb{R}^d, \mathbf{w} \in \mathbb{R}^d, \mathbf{w}_y \in \mathbb{R}^d$.

[1] for $b \in \mathbb{R}$, more general form $\langle \mathbf{x}, \mathbf{w} \rangle + b$ can be rewritten as $\langle [\mathbf{x}; 1], [\mathbf{w}; b] \rangle$

Decision functions

Typical decision functions for classification [1] :

$$\text{Binary-class} \quad g(\mathbf{x};\mathbf{w}) = \mathrm{sign}(\langle \mathbf{x}, \mathbf{w} \rangle).$$

$$\text{Multi-class} \quad g(\mathbf{x};\mathbf{w}) = \operatorname*{argmax}_{y \in \mathcal{Y}}(\langle \mathbf{x}, \mathbf{w}_y \rangle).$$

where $\mathbf{w}, \mathbf{w}_y$ are the parameters, and $\mathbf{x} \in \mathbb{R}^d, \mathbf{w} \in \mathbb{R}^d, \mathbf{w}_y \in \mathbb{R}^d$.
Example 1: $\mathbf{x} = [1; 3.5], \mathbf{w} = [2; -1]$. $g(\mathbf{x};\mathbf{w}) = ?$

[1] for $b \in \mathbb{R}$, more general form $\langle \mathbf{x}, \mathbf{w} \rangle + b$ can be rewritten as $\langle [\mathbf{x}; 1], [\mathbf{w}; b] \rangle$

Decision functions

Typical decision functions for classification [1] :

$$\text{Binary-class} \quad g(\mathbf{x}; \mathbf{w}) = \operatorname{sign}(\langle \mathbf{x}, \mathbf{w} \rangle).$$

$$\text{Multi-class} \quad g(\mathbf{x}; \mathbf{w}) = \operatorname*{argmax}_{y \in \mathcal{Y}}(\langle \mathbf{x}, \mathbf{w}_y \rangle).$$

where $\mathbf{w}, \mathbf{w}_y$ are the parameters, and $\mathbf{x} \in \mathbb{R}^d, \mathbf{w} \in \mathbb{R}^d, \mathbf{w}_y \in \mathbb{R}^d$.
Example 1: $\mathbf{x} = [1; 3.5], \mathbf{w} = [2; -1]$. $g(\mathbf{x}; \mathbf{w}) = ?$
$= \operatorname{sign}(1 \times 2 + 3.5 \times (-1)) = \operatorname{sign}(-1.5) = -1.$

[1] for $b \in \mathbb{R}$, more general form $\langle \mathbf{x}, \mathbf{w} \rangle + b$ can be rewritten as $\langle [\mathbf{x}; 1], [\mathbf{w}; b] \rangle$

Decision functions

Typical decision functions for classification [1] :

$$\text{Binary-class} \quad g(\mathbf{x}; \mathbf{w}) = \text{sign}(\langle \mathbf{x}, \mathbf{w} \rangle).$$

$$\text{Multi-class} \quad g(\mathbf{x}; \mathbf{w}) = \operatorname*{argmax}_{y \in \mathcal{Y}}(\langle \mathbf{x}, \mathbf{w}_y \rangle).$$

where $\mathbf{w}, \mathbf{w}_y$ are the parameters, and $\mathbf{x} \in \mathbb{R}^d, \mathbf{w} \in \mathbb{R}^d, \mathbf{w}_y \in \mathbb{R}^d$.
Example 1: $\mathbf{x} = [1; 3.5], \mathbf{w} = [2; -1]$. $g(\mathbf{x}; \mathbf{w}) = ?$
$= \text{sign}(1 \times 2 + 3.5 \times (-1)) = \text{sign}(-1.5) = -1$.
Example 2:
$\mathbf{x} = [1; 3.5], \mathbf{w}_1 = [2; -1], \mathbf{w}_2 = [1; 2], \mathbf{w}_3 = [3; 2], y = 1, 2, 3$.
$g(\mathbf{x}; \mathbf{w}) = ?$

[1] for $b \in \mathbb{R}$, more general form $\langle \mathbf{x}, \mathbf{w} \rangle + b$ can be rewritten as $\langle [\mathbf{x}; 1], [\mathbf{w}; b] \rangle$

Decision functions

Typical decision functions for classification [1] :

$$\text{Binary-class} \quad g(\mathbf{x};\mathbf{w}) = \operatorname{sign}(\langle \mathbf{x}, \mathbf{w} \rangle).$$

$$\text{Multi-class} \quad g(\mathbf{x};\mathbf{w}) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}(\langle \mathbf{x}, \mathbf{w}_y \rangle).$$

where $\mathbf{w}, \mathbf{w}_y$ are the parameters, and $\mathbf{x} \in \mathbb{R}^d, \mathbf{w} \in \mathbb{R}^d, \mathbf{w}_y \in \mathbb{R}^d$.
Example 1: $\mathbf{x} = [1; 3.5], \mathbf{w} = [2; -1]$. $g(\mathbf{x};\mathbf{w}) = ?$
$= \operatorname{sign}(1 \times 2 + 3.5 \times (-1)) = \operatorname{sign}(-1.5) = -1$.
Example 2:
$\mathbf{x} = [1; 3.5], \mathbf{w}_1 = [2; -1], \mathbf{w}_2 = [1; 2], \mathbf{w}_3 = [3; 2], y = 1, 2, 3$.
$g(\mathbf{x};\mathbf{w}) = ?$ $\langle \mathbf{x}, \mathbf{w}_1 \rangle = -1.5, \langle \mathbf{x}, \mathbf{w}_2 \rangle = 8, \langle \mathbf{x}, \mathbf{w}_3 \rangle = 10$. Thus
$g(\mathbf{x};\mathbf{w}) = \operatorname{argmax}_{y \in \{1,2,3\}} \langle \mathbf{x}, \mathbf{w}_y \rangle = 3$.

[1] for $b \in \mathbb{R}$, more general form $\langle \mathbf{x}, \mathbf{w} \rangle + b$ can be rewritten as $\langle [\mathbf{x}; 1], [\mathbf{w}; b] \rangle$

Convexity

- Convexity for a function
- Convexity for a set

Illustrate using the whiteboard or the document camera.

Lagrange multipliers and function

To solve a convex minimisation problem,

$$\min_{\mathbf{x}} f_0(\mathbf{x})$$
$$\text{s.t.} \quad f_i(\mathbf{x}) \leq 0, i = 1, \cdots, m, \qquad \text{(Primal)}$$

where f_0 is convex, and the feasible set (let's call it A) is convex (equivalent to all f_0, f_i are convex). $\mathbf{x}$ are called primal variables.

Lagrange function:

$$L(\mathbf{x}, \alpha) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i f_i(\mathbf{x}),$$

where $\alpha_i \geq 0$ are called Lagrange multipliers also known as (a.k.a) dual variables.

Dual problem

$L(\mathbf{x}, \alpha)$ produces the primal objective:

$$f_0(\mathbf{x}) = \max_{\alpha \geq 0} L(\mathbf{x}, \alpha).$$

$L(\mathbf{x}, \alpha)$ produces the dual objective:

$$D(\alpha) = \min_{\mathbf{x} \in A} L(\mathbf{x}, \alpha).$$

The following problem is called the (Lagrangian) dual problem,

$$\begin{aligned} & \max_{\alpha} D(\alpha) \\ \text{s.t.} \quad & \alpha_i \geq 0, i = 1, \cdots, m. \qquad \text{(Dual)} \end{aligned}$$

Primal and Dual relation

In general:

$$\min_{\mathbf{x}\in A} f_0(\mathbf{x}) = \min_{\mathbf{x}\in A}(\max_{\alpha\geq 0} L(\mathbf{x},\alpha)) \geq \max_{\alpha\geq 0}(\min_{\mathbf{x}\in A} L(\mathbf{x},\alpha)) = \max_{\alpha\geq 0} D(\alpha).$$

Since $L(\mathbf{x},\alpha)$ is convex w.r.t. $\mathbf{x}$, and concave w.r.t. α, we have

$$\min_{\mathbf{x}\in A} f_0(\mathbf{x}) = \min_{\mathbf{x}\in A}(\max_{\alpha\geq 0} L(\mathbf{x},\alpha)) = \max_{\alpha\geq 0}(\min_{\mathbf{x}\in A} L(\mathbf{x},\alpha)) = \max_{\alpha\geq 0} D(\alpha).$$

To solve the primal $\min_{\mathbf{x}\in A} f_0(\mathbf{x})$, one can solve the dual $\max_{\alpha\geq 0} D(\alpha)$.

Duality

The following always holds
$D(\alpha) \leq f_0(\mathbf{x}), \ \forall \mathbf{x}, \alpha$ (so called weak duality)

Sometimes (not always) below holds
$\max_\alpha D(\alpha) = \min_\mathbf{x} f_0(\mathbf{x})$ (so called strong duality)
Strong duality holds for SVM.

How to do it?

Given a problem, how to get its dual form?

1. transform the problem to a standard form
2. write down the Lagrange function
3. use optimality conditions to get equations
   - 1st order condition
   - complementarity conditions
4. remove the primal variables.

Examples.

Perceptron Algorithm

Assume $g(\mathbf{x};\mathbf{w}) = \text{sign}(\langle \mathbf{x}, \mathbf{w} \rangle)$, where $\mathbf{x}, \mathbf{w} \in \mathbb{R}^d$, $y \in \{-1, 1\}$.

Input: training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, step size η, #iter T
Initialise $w_1 = \mathbf{0}$
for $t = 1$ **to** T **do**

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta \sum_{i=1}^{n} (y_i \, \mathbf{x}_i \, \mathbf{1}_{\{y_i \langle \mathbf{x}_i, \mathbf{w}_t \rangle < 0\}}) \tag{1}$$

end for
Output: $\mathbf{w}^* = \mathbf{w}_T$

The class of $\mathbf{x}$ is predicted via

$$y^* = \text{sign}(\langle \mathbf{x}, \mathbf{w}^* \rangle)$$

View it in ERM

$$\min_{\mathbf{w},\xi} \frac{1}{n}\sum_{i=1}^{n} \xi_i, \quad \text{s.t.} \quad y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq -\xi_i, \xi_i \geq 0$$

whose unconstrained form is

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \max\{0, -y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\} \Leftrightarrow \min_{\mathbf{w}} R_n(\mathbf{w}, \ell_{pern})$$

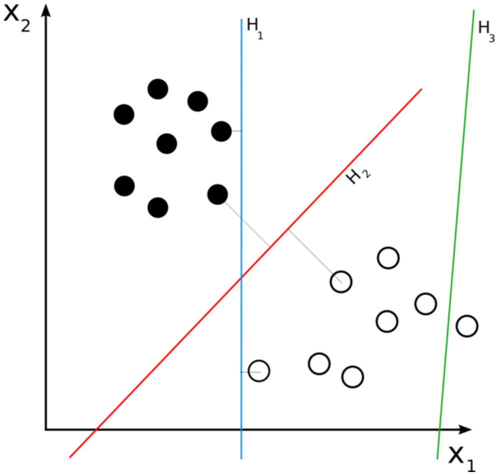
with Loss $\ell_{pern}(\mathbf{x}, y, \mathbf{w}) = \max\{0, -y \langle \mathbf{x}, \mathbf{w} \rangle\}$ and
Empirical Risk $R_n(\mathbf{w}, \ell_{pern}) = \frac{1}{n}\sum_{i=1}^{n} \ell_{pern}(\mathbf{x}_i, y_i, \mathbf{w})$.

$$\text{Sub-gradient} \quad \frac{\partial R_n(\mathbf{w}, \ell_{pern})}{\partial \mathbf{w}} = -\frac{1}{n}\sum_{i=1}^{n}(y_i \mathbf{x}_i \mathbf{1}_{\{y_i(\langle \mathbf{x}_i, \mathbf{w}_t \rangle)<0\}}).$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta' \frac{\partial R_n(\mathbf{w}, \ell_{pern})}{\partial \mathbf{w}} = \mathbf{w}_t + \eta' \frac{1}{n}\sum_{i=1}^{n}(y_i \mathbf{x}_i \mathbf{1}_{\{y_i(\langle \mathbf{x}_i, \mathbf{w}_t \rangle)<0\}})$$

Letting $\eta = \eta' \frac{1}{n}$ recovers the equation (1).

Max Margin



Picture courtesy of wikipedia

Max Margin Formulation

One form of soft margin binary Support Vector Machines (SVMs) (a primal form) is

$$\min_{\mathbf{w},b,\gamma,\xi} -\gamma + C\sum_{i=1}^{n}\xi_i \tag{2}$$
$$\text{s.t. } y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle + b) \geq \gamma - \xi_i, \xi_i \geq 0, \| \mathbf{w} \|^2 = 1$$

For a testing $\mathbf{x}'$, given the learnt $\mathbf{w}^*, b^*$, the predicted label

$y^* = g(\mathbf{x}'; \mathbf{w}^*) = \mathrm{sign}(\langle \mathbf{x}', \mathbf{w}^*\rangle + b^*)$.

Primal

A more popular version is (still a primal form)

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\| \mathbf{w} \|^2 + C \sum_{i=1}^{n} \xi_i,$$

$$\text{s.t.} \quad y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \cdots, n,$$

This is equivalent to the previous form and $\gamma = 1/\| \mathbf{w} \|$.

View in in ERM hinge loss $\ell_H(\mathbf{x}, y, \mathbf{w}) = \max\{0, 1 - y(\langle \mathbf{x}, \mathbf{w} \rangle + b)\}$, and $\Omega(\mathbf{w}) = \frac{1}{2}\| \mathbf{w} \|^2$ with a proper λ.

It is often solved by using Lagrange multipliers and duality.

Lagrangian function

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \alpha_i [1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b)] + \sum_{i=1}^{n} \beta_i (-\xi_i)$$

Optimise Lagrangian function — 1st order condition

To get $\inf_{\mathbf{w},b,\xi}\{L(\mathbf{w},b,\xi,\alpha,\beta)\}$, by 1st order condition

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial\,\mathbf{w}} = 0 \Rightarrow \mathbf{w}^* - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i = 0 \tag{3}$$

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial \xi_i} = 0 \Rightarrow C - \alpha_i - \beta_i = 0 \tag{4}$$

$$\frac{\partial L(\mathbf{w},b,\xi,\alpha,\beta)}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{5}$$

Optimise Lagrangian function — Complementarity conditions

Complementarity conditions

$$\alpha_i[1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b)] = 0, \forall i \tag{6}$$

$$\beta_i \xi_i = 0, \forall i \tag{7}$$

Dual

$$
\begin{aligned}
&L(\mathbf{w}^*, b^*, \xi^*, \alpha, \beta) \\
&= \frac{1}{2}\langle \mathbf{w}^*, \mathbf{w}^* \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{w}^* \rangle \\
&+ \sum_{i=1}^{n} \xi_i^* (C - \alpha_i - \beta_i) + b(\sum_{i=1}^{n} \alpha_i y_i) \\
&= \frac{1}{2}\langle \mathbf{w}^*, \mathbf{w}^* \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{w}^* \rangle \quad \text{via eq(4) and eq(5)} \\
&= \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \;\; \text{via eq(3)} \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle
\end{aligned}
$$

Dual

$\max_\alpha \inf_{\mathbf{w},b,\xi}\{L(\mathbf{w}, b, \xi, \alpha, \beta)\}$ gives the dual form:

$$
\begin{aligned}
\max_\alpha \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\
\text{s.t.} \quad 0 \le \alpha_i \le C, i = 1, \cdots, n, \quad \text{(via eq(4))} \\
\sum_{i=1}^{n} \alpha_i y_i = 0
\end{aligned}
$$

Let α^* be the solution.

From dual to primal variables

How to compute $\mathbf{w}^*, b^*$ from α^*?
Via eq(3), we have

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i. \tag{8}$$

Via comp condition eq(6), we have $\alpha_i[1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b)] = 0, \forall i$.
When $\alpha_i > 0$, we know $1 - \xi_i - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) = 0$. It will be great if $\xi_i = 0$ too. When will it happen? $\beta_i > 0 \Rightarrow \xi_i = 0$ because of comp condition eq(7). Since $C - \alpha_i - \beta_i = 0$ (4), $\beta_i > 0$ means $\alpha < C$.
For any i, s.t. $0 < \alpha_i < C$, $1 - y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) = 0$, so (multiple y_i on both sides, and the fact that $y_i^2 = 1$)

$$b^* = y_i - \langle \mathbf{x}_i, \mathbf{w}^* \rangle \tag{9}$$

Numerically wiser to take the average over all such training points (Burges tutorial).

Support Vectors

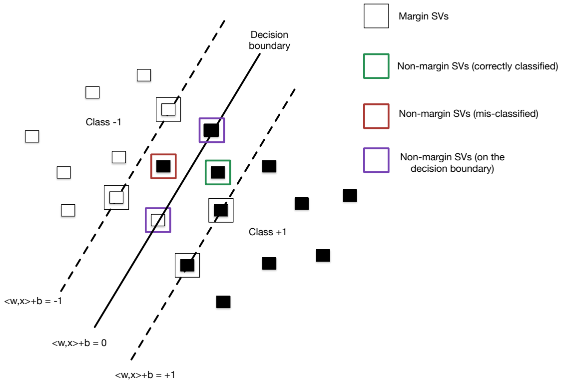
$y^* = \text{sign}(\langle x, \mathbf{w}^* \rangle + b^*) = \text{sign}(\sum_{i=1}^n \alpha_i^* y_i \langle \mathbf{x}_i, x \rangle + b^*)$.

It turns out many $\alpha_i^* = 0$. Those $\mathbf{x}_j$ with $\alpha_j^* > 0$ are called support vectors. Let $S = \{j : \alpha_j^* > 0\}$

$$y^* = \text{sign}(\sum_{j \in S} \alpha_j^* y_j \langle \mathbf{x}_j, \mathbf{x} \rangle + b^*)$$

Note now y can be predicted without explicitly expressing $\mathbf{w}$ as long as the support vectors are stored.

Support Vectors



Two types of SVs:

- Margin SVs: $0 < \alpha_i < C$ ($\xi_i = 0$, on the dash lines)
- Non-margin SVs: $\alpha_i = C$ ($\xi_i > 0$, thus violating the margin. More specifically, when $1 > \xi_i > 0$, correctly classified; when $\xi_i > 1$, it's mis-classified; when $\xi_i = 1$, on the decision boundary)

Dual

All derivation holds if one replaces $\mathbf{x}_j$ with $\phi(\mathbf{x}_j)$ and let kernel function $\kappa(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$. This gives

$$
\begin{aligned}
\max_{\alpha} & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \\
\text{s.t.} \quad & 0 \leq \alpha_i \leq C, i = 1, \cdots, n \\
& \sum_{i=1}^{n} \alpha_i y_i = 0
\end{aligned}
$$

$$
y^* = \text{sign}[\sum_{j \in S} \alpha_j^* y_j \kappa(\mathbf{x}_j, \mathbf{x}) + b^*].
$$

This leads to non-linear SVM and more generally kernel methods (will be covered in later lectures).

Theoretical justification

An example of generalisation bounds is below (just to give you an intuition, no need to fully understand it for now).

Theorem (VC bound)

Denote h as the VC dimension, for all $n \geq h$, for any $\delta \in (0,1)$, with probability at least $1-\delta$, $\forall g \in \mathcal{G}$

$$R(g) \leq R_n(g) + 2\sqrt{2\frac{h \log \frac{2en}{h} + \log(\frac{2}{\delta})}{n}}.$$

Margin $\gamma = 1/\| \mathbf{w} \|$, $h \leq \min\{D, \lceil \frac{4R^2}{\gamma^2} \rceil\}$, where the radius $R^2 = \max_{i=1}^{n} \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_i) \rangle$ (assuming data are already centered)

Theoretical justification

Other tighter bounds such as Rademacher bounds, PAC-Bayes bounds *etc.*.

That's all

Thanks!