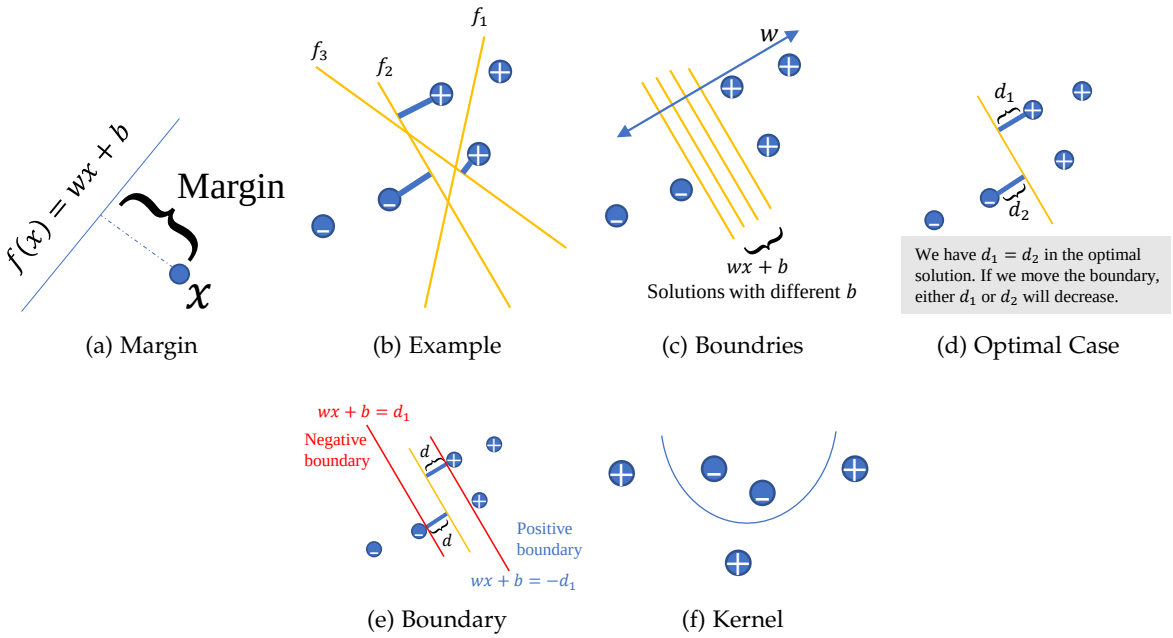
# CISC Introduction to Machine Learning
# Lecture Notes

Dr. Guangmo (Amo) Tong

## Support Vector Machine




(a) Margin (b) Example (c) Boundries (d) Optimal Case

(e) Boundary (f) Kernel

Figure 1: SVM.

### What is SVM?

Support vector machine (SVM), without using a kernel, is a linear classifier. Therefore, its decision boundary is of the form $w^T x + b$, where $x$ is the input, $w$ is the weight parameter, and $b$ is the bias parameter. The classification rule of $f(x) = w^T x + b$ is

$$Predict(x) = \begin{cases} 1 & \text{if } w^T x + b \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

## How to train SVM?

### Formulation

Basically, we need to compute the parameters $w$ and $b$. Given a linearly separable dataset, there can be multiple correct linear decision boundaries. Which one should we choose? SVM selects the one that can **maximize the margin from the data points to the boundary**. For one data point $x$ and a linear classifier $f$, its margin $M(x, f)$ is the length of the perpendicular drawn from $x$ to $f$, see Fig. 1a. For a dataset $D = \{x_1, ..., x_n\}$ of $n$ points and one linear classifier $f$, the margin is define as $M(D, f) = \min_{x \in D} M(x, f)$ - the minimum margin among all the points. The solution of SVM is

$$\max_f M(D, f) \tag{1}$$

$$\text{subject to } w^T x + b \geq 0 \text{ for positive points} \tag{2}$$

$$w^T x + b < 0 \text{ for negative points.} \tag{3}$$

**Example 1.** Consider the example in Fig. 1b. $f_1$ cannot correctly classify all the points, and we may take its margin as $M(D, f_1) = -\infty$. $f_2$ and $f_3$ are correct boundaries, and $f_2$ has a larger margin.

To solve the equations, let us consider how to draw a linear boundary. Typically, we can first decide $w$, which gives the direction of the line, and then decide $b$, which gives the position along the direction (See Fig. 1c). Supposing a direction $w$ is decided, in the optimal case, **the margin from the positive side must be equal to the margin from the negative side**. That is, the optimal solution is in the middle. Intuitively, if we move the mid boundary along the given direction to either the positive or the negative side, we see that $M(D, f) = \min_{x \in D} M(x, f)$ will decrease (See Fig. 1d). Therefore, our equations become

$$\max_f M(D, f) = d \tag{4}$$

$$\text{subject to } w^T x + b \geq d_1 \text{ for positive points} \tag{5}$$

$$w^T x + b < -d_1 \text{ for negative points,} \tag{6}$$

where $d$ is the margin to want to maximize and $d_1$ is an extra bias between the mid boundary and positive boundary. The relationship between $d$, $d_1$ and $w$ is

$$d = d_1 / \parallel W \parallel,$$

so we have

$$\max \frac{d_1}{\parallel W \parallel} \tag{7}$$

$$\text{subject to } wx + b \geq d_1 \text{ for positive points} \tag{8}$$

$$w^T x + b < -d_1 \text{ for negative points.} \tag{9}$$

Note that the whole system has some freedom, and we can scale it by $1/|d_1|$ and have

$$\max \frac{1}{\| W \|} \tag{10}$$

$$\text{subject to } w^T x + b \geq 1 \text{ for positive points} \tag{11}$$

$$w^T x + b < -1 \text{ for negative points.} \tag{12}$$

Note that $w$ and $b$ are learnable parameters so there is no need to put a factor before them. Finally, let $y_i \in \{+1, -1\}$ be the class value of $x_i$, and the constrains can be simplified as

$$\max \frac{1}{\| W \|} \tag{13}$$

$$\text{subject to } y_i \cdot (w^T x + b) \geq 1 \text{ for each } x_i. \tag{14}$$

$\frac{1}{\|W\|}$ is not a friendly form for optimization, and we consider the following problem:

$$\min \frac{1}{2} \| W \|^2 \tag{15}$$

$$\text{subject to } y_i \cdot (w^T x_i + b) \geq 1 \text{ for each } x_i. \tag{16}$$

This problem is a quadratic programming with $n$ (number of training points) linear constraints. At this point, we may use any software to solve it. However, we want to see the properties of the SVM, and let us use the Lagrange multiplier method.

**The Dual Problem**

For Lagrange multiplier method, see Canvas for a brief introduction. Using Lagrange multipliers, Eqs. (15) and (16) can be equivalently transferred as

$$\min \; L(w, b, \alpha_i) = \frac{1}{2} \| w \|^2 - \sum_i \alpha_i \cdot (y_i \cdot (w^T x_i + b) - 1) \tag{17}$$

$$\text{subject to } \alpha_i \geq 0, \tag{18}$$

where each constraints in Eq (16) is associated with one Lagrange multiplier $\alpha_i$. Setting the derivatives of $L(w, b, \alpha_i)$ with respect to $w$ and $b$ (please try this yourself), the solution SVM must satisfy

$$w = \sum_i \alpha_i \cdot (y_i \cdot x_i) \tag{19}$$

and

$$0 = \sum_i \alpha_i \cdot y_i.$$

Putting Eq. (19) into Eqs (17) and (18) (please try this yourself), we have the dual form

$$\min \; L(\alpha_i) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{20}$$

$$\text{subject to } \alpha_i \geq 0, \tag{21}$$

$$\sum_i \alpha_i \cdot y_i = 0, \tag{22}$$

where we denote $L(w, b, \alpha_i)$ as $L(\alpha_i)$ since $w$ and $b$ are now not in the equations. This is the dual form of Eqs. (15) and (16). The above transformation is useful for two purpose: (a) understanding the support vector and (b) applying the kernel trick.

### Support Vector

Applying the Karush-Kuhn-Tucker (KKT) condition to our problem Eqs. (17) and (18), the SVM solution must satisfy

$$\alpha_i \geq 0, \tag{23}$$

$$y_i \cdot (w^T x_i + b) - 1 \geq 0, \tag{24}$$

$$\alpha_i \cdot (y_i \cdot (w^T x_i + b) - 1) = 0. \tag{25}$$

Eqs. (24) and (25) imply that for each $x_i$, either $\alpha_i = 0$, which means $(x_i, y_i)$ does not affect the solution to Eqs. (20),(21) and (22), or $y_i \cdot (w^T x_i + b) - 1 = 0$, which means $(x_i, y_i)$ is on the boundary (positive or negative) and it is a *support vector*. If we look at Fig. 1e, we see that the SVM solution is deductively determined by the points on the boundary - the support vectors.

### Solutions and Prediction

We see that the solution to Eqs. (20), (21) and (22) is $\{\alpha_i\}$, but we need to find out $w$ and $b$. $w$ can be derived from $\alpha_i$ by Eq. (19) as $\sum_i \alpha_i \cdot (y_i \cdot x_i)$, and $b$ can be derived by any support vector $(x^*, y^*)$ according to Eq. (25). That is, $y^* \cdot (\sum_i \alpha_i y_i x_i^T x^* + b) = 1$.

For a new instance $x_{new}$, the prediction made according to

$$\text{sign}\left(w^T x_{new} + b - 1\right) = \text{sign}\left(\sum_i (\alpha_i \cdot y_i \cdot x_i^T \cdot x_{new}) + b - 1)\right).$$

As we can see here, there is in fact no need to recover the parameter $w$ as the prediction can be made by $\alpha_i$ and $b$.

### Kernel Trick

If the data is not linearly separable, we may consider other curves that can separate them into two correct areas. For example, in Fig. 1f, a quadratic function $\phi(x)$ can classify the given data points, with areas $\phi(x) \geq 0$ and $\phi(x) < 0$. From another perspective, we are using $\phi(x)$ to map the data into another space so that they are linearly separable i.e., using $w^T \phi(x) + b = 1$ as the decision boundary.

Considering any map $\phi(x)$, the kernel is defined as $K(x,y) = \phi(x)^T\phi(y)$ for each pair $x, y$. Why we consider $K(x,y)$ instead of the map $\phi(x)$? This is because the Eqs. (20), (21) and (22) only need the inner product $x_i^T x_j$ of the points, so does the process of recovering $w$ and $b$ as well as the prediction. The kernel trick is that for any mapping we want to use $\phi(x)$, we consider the kernel function $K(x,y) = \phi(x)^T\phi(y)$ and solve the following equations:

$$\min \ L(\alpha_i) = \sum_i \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j K(x_i, x_j) \tag{26}$$

$$\text{subject to } \alpha_i \geq 0, \tag{27}$$

$$\sum_i \alpha_i \cdot y_i = 0, \tag{28}$$

and the predictions are made by

$$\text{sign}\big(\sum_i (\alpha_i \cdot y_i \cdot K(x_i, x_{new})) + b - 1)\big).$$