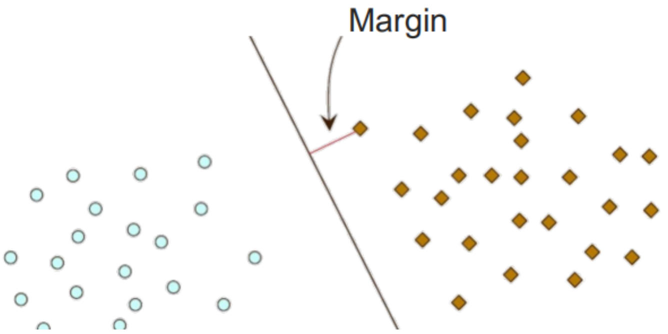
# ch7: Support Vector Machine

> **The Idea:** Find a decision boundary that **maximizes the margin** between two classes




## Problem Formulation(P6-9)

### Maximizing the Margin(P7)

$$\boxed{\boldsymbol{w}^{\mathrm{T}}x+w_0 = \begin{cases} 1 & \text{for the closest points on one side} \\ -1 & \text{for the closest points on the other} \end{cases}}$$

- Let $x^{(1)}$ and $x^{(2)}$ be two closest points on each side of the hyperplane.
- Note that

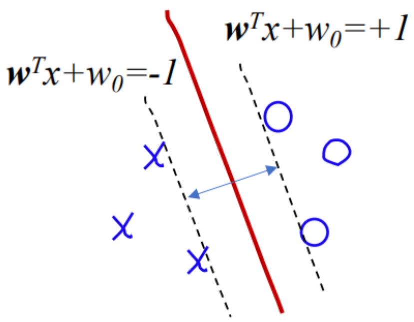
$$\boldsymbol{w}^{\mathrm{T}}x^{(1)} + w_0 = +1$$
$$\boldsymbol{w}^{\mathrm{T}}x^{(2)} + w_0 = -1$$

which imply

$$\boldsymbol{w}^{\mathrm{T}}(x^{(1)} - x^{(2)}) = 2.$$

Hence, the margin can be given by

$$\text{margin} = \frac{\boldsymbol{w}^{\mathrm{T}}(x^{(1)} - x^{(2)})}{\|\boldsymbol{w}\|} = \frac{2}{\|\boldsymbol{w}\|}$$

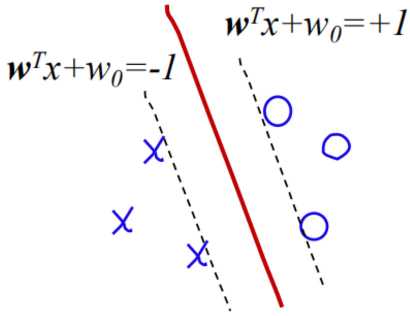



Maximizing the margin is equivalent to minimizing $\frac{1}{2}\|\boldsymbol{w}\|$

### Inequality Constraints(P8)

- Given: $D = \{(x^{(\ell)}, y^{(\ell)}), \ldots, (x^{(N)}, y^{(N)})\}$, we want $\boldsymbol{w}$ and $w_0$ to satisfy

$$\boldsymbol{w}^{\mathrm{T}}x^{(\ell)}+w_0 \begin{cases} \geq +1 & \text{if } y^{(\ell)}=+1 \\ \leq -1 & \text{if } y^{(\ell)}=-1 \end{cases}$$




- Or, equivalently,

$$y^{(\ell)}\,(\boldsymbol{w}^{\mathrm{T}}x^{(\ell)}+w_0) \geq 1$$

**SVM = Solving an Optimization Problem(P9)**

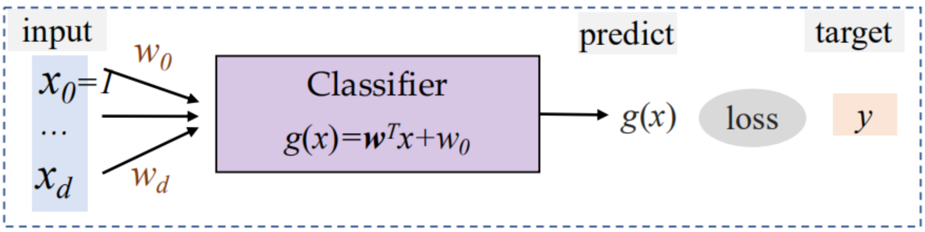
In summary, SVM aims to solve a constrained optimization problem:

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\text{subject to} && y^{(\ell)}(\boldsymbol{w}^{\mathrm{T}}x^{(\ell)}+w_0)\geq 1,\ \ell=1,\ldots,\mathrm{N} \end{aligned}$$

- This is a quadratic programming (QP) problem, which is one type of convex optimization problem.
- The complexity depends on the dimensionality $d$ of inputs

**Model Architecture(P10)**

- The same architecture as Perceptron.




- Train:
  - optimize the parameters $\boldsymbol{w}$ and $w_0$ using data
- Test:
  - calculate $g(x) = \boldsymbol{w}^{\mathrm{T}}x+w_0$ and choose $C_1$ if $g(x) > 1$ or choose $C_2$ if $g(x)<-1$.

## Loss Function(P11)

- For a given input $x$, the model outputs a score $g(x) = \boldsymbol{w}^{\mathrm{T}}x + w_0$. Let $y \in \{-1, +1\}$ be the label of the real class ($y = +1$: $x \in C_1$, $y = -1$: $x \in C_2$):
  - if $y(\boldsymbol{w}^{\mathrm{T}}x + w_0) < 1$: we aim to maximize $y(\boldsymbol{w}^{\mathrm{T}}x + w_0)$ until reaching 1, cost is $1 - y(\boldsymbol{w}^{\mathrm{T}}x + w_0)$
  - if $y(\boldsymbol{w}^{\mathrm{T}}x + w_0) > 1$: outlier points, no need to optimize, cost is **0**

- Given: $D = \{(x^{(1)}, r^{(1)}), \ldots, (x^{(N)}, r^{(N)})\}$, the loss over the dataset is defined as:

$$L(\boldsymbol{w}, w_0 \mid D) = \frac{1}{N}\sum_{\ell=1}^{N} \max(0,\ 1 - y^{(\ell)}(\boldsymbol{w}^{\mathrm{T}}x^{(\ell)} + w_0)) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2$$

## Optimization – Gradient Descend(P12)

$$\boxed{\boldsymbol{w}_{\text{new}} = \boldsymbol{w}_{\text{old}} + \eta\frac{\partial L}{\partial w}}$$

What is $\dfrac{\partial L}{\partial w}$?

$$L(\boldsymbol{w}, w_0 \mid D) = \begin{cases} \frac{\lambda}{2}\|\boldsymbol{w}\|^2 & \textbf{if}\ \ y^{(\ell)}(\boldsymbol{w}^{\mathrm{T}}x^{(\ell)} + w_0) \geq 1 \\ \frac{1}{N}\sum_{\ell=1}^{N} 1 - y^{(\ell)}(\boldsymbol{w}^{\mathrm{T}}x^{(\ell)} + w_0) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 & \text{otherwise} \end{cases}$$

For each $\boldsymbol{w}_j$ ($j = 0, \ldots, d$):

$$\frac{\partial L}{\partial w_j} = \begin{cases} \lambda w_j & \text{if } y^{(\ell)}(\boldsymbol{w}^{\mathrm{T}}x^{(\ell)} + w_0) \geq 1 \\ \lambda w_j - y^{(\ell)}x^{(\ell)} & \text{o.w.} \end{cases}$$

$$\frac{\partial L}{\partial w_0} = \begin{cases} 0 & \text{if } y^{(\ell)}(\boldsymbol{w}^{\mathrm{T}}x^{(\ell)} + w_0) \geq 1 \\ -y^{(\ell)} & \text{o.w.} \end{cases}$$

## Algorithm(P13)

**Gradient Descend for SVM**

**Input:** $D = \{(\mathrm{x}^{(l)}, \mathrm{y}^{(l)})\}$ $(l = 1{:}N)$
**for** $j = 0, \ldots, d$
  $\boldsymbol{w}_{\mathrm{j}} \leftarrow rand\,(-0.01, 0.01)$
**repeat**
  **for** $j = 0, \ldots, d$
    $\Delta\boldsymbol{w}_{\mathrm{j}} \leftarrow 0$
  **for** $l = 1, \ldots, N$
    $a \leftarrow 0$
    **for** $j = 0, \ldots, d$
      $a \leftarrow a + \boldsymbol{w}_{\mathrm{j}} x_{\mathrm{j}}^{(l)}$
    **for** $j = 0, \ldots, d$
     $\Delta\boldsymbol{w}_{\mathrm{j}} \leftarrow \Delta\boldsymbol{w}_{\mathrm{j}} + \lambda\boldsymbol{w}_{\mathrm{j}}$
     $\Delta\boldsymbol{w}_{0} \leftarrow 0$
     **if** $y^{(l)}a<1$: $\Delta\boldsymbol{w}_{\mathrm{j}} \leftarrow \Delta\boldsymbol{w}_{\mathrm{j}} - y^{(l)}x^{(l)}$
  $\Delta\boldsymbol{w}_{\mathrm{j}} = \Delta\boldsymbol{w}_{\mathrm{j}}/\mathrm{N}$
  **for** $j = 0, \ldots, d$
    $w_{\mathrm{j}} \leftarrow w_{\mathrm{j}} + \eta\Delta w_{\mathrm{j}}$
**until** convergence

# Dual Problem

## Lagrangian(P15)

- 定义

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \le 0, & \text{for } i = 1,\dots,m. \\ & h_i(x) = 0, & \text{for } i = 1,\dots,k. \end{array}$$

**Idea:** relax/soften the constraints by replacing each with a linear "penalty term" or "cost" in the objective.

**The Lagrangian Function**

$$L(x, \lambda, v) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{k} v_i h_i(x)$$

- $\lambda_i$ is the **lagrange multiplier** for the $i$-th inequality constraint
  - required to be nonnegative
- $v_i$ is the **lagrange multiplier** for the $i$-th equality constraint
  - allowed to be arbitrary sign

- **Lagrange Dual Problem**

This is the problem of finding the best lower bound on OPT(primal) obtained from the Lagrange dual function

$$\begin{array}{ll} \underset{\lambda,\nu}{\text{maximize}} & g(\lambda, \nu) \\ \text{subject to} & \lambda \ge 0. \end{array}$$

- $(\lambda^\star, v^\star)$ solving the above are referred to as the **dual optimal solution.**
- **Note**: this is a convex optimization problem, regardless of whether primal problem was convex

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le 0, \quad \text{for } i = 1,\dots,\text{m}. \\ & h_i(x) = 0, \quad \text{for } i = 1,\dots,\text{k}. \end{array}$$

**Idea**: creates the **lower bound** of the primal optimum subject to the softened constraints.

**The Lagrange Dual Function**

$$g(\lambda, v) = \inf_{x \in \mathcal{D}} L(x, \lambda, v) = \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{k} v_i h_i(x)\right)$$

## The Dual Problem for SVM(P16-17)

### Dual optimization problem:

$$\begin{array}{ll} \text{maximize} & \sum_{\ell=1}^{N} \alpha_\ell - \frac{1}{2} \sum_{\ell=1}^{N} \sum_{\ell'=1}^{N} \alpha_\ell \alpha_{\ell'} y^{(\ell)} y^{(\ell')} (x^{(\ell)})^T x^{(\ell')} \\ \text{subject to} & \sum_{\ell=1}^{N} \alpha_\ell y^{(\ell)} = 0 \\ & \alpha_\ell \ge 0, \ \ell = 1 \dots N \end{array}$$

- This is also a QP problem, but its complexity depends on the sample size $N$ (rather than the input dimensionality $d$)

**Primal**

$$\begin{array}{ll} \min & \frac{1}{2} \|\boldsymbol{w}\|^2 \\ \text{s.t.} & y^{(l)}(\boldsymbol{w}^T \mathrm{x}^{(l)} + w_0) \ge 1, \ \forall l \end{array}$$

The complexity depends on the dimensionality $d$ of inputs

**Dual**

$$\begin{array}{ll} \max & \Sigma_l \alpha_l - \frac{1}{2} \Sigma_l \Sigma_{l'} \alpha_l \alpha_{l'} y^{(l)} y^{(l')} (x^{(l)})^T x^{(l')} \\ \text{s.t.} & \Sigma_l \alpha_l y^{(l)} = 0 \\ & \alpha_l \ge 0, \ l = 1 \dots \text{N} \end{array}$$

The complexity depends on the sample size $N$

- **It turns out to be more convenient to solve the dual problem** than solving the **primal problem** ($N < d$)
- We can firstly solve **Dual** to obtain $\{\alpha_\ell\}$, and then obtain the $W$ in **Primal**

## Support Vectors(P19)

### Suppose the optimal $\{\alpha_\ell\}$ have been obtained

- Patterns for which $y^{(\ell)}(wx^{(\ell)}+w_0) > 1$

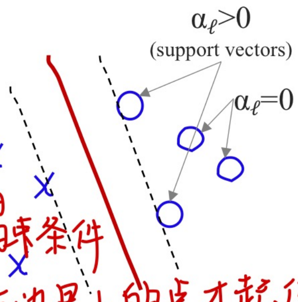
  不起作用的约束条件

  $\alpha_\ell=0$ (inactive constraints) $\Rightarrow x^{(\ell)}$ irrelevant

  recall complimentary slackness: $\lambda_i^* g_i(x^*)=0$

- Patterns that have $\alpha_\ell>0$ (active constraints) 起作用的约束条件

  $y^{(\ell)}(wx^{(\ell)}+w_0) = 1 \Rightarrow x^{(\ell)}$ lies on margin




只有边界上的点才起作用

- Most of the dual variables vanish with $\alpha_\ell=0$. They are points lying beyond the margin with **no effect** on the hyperplane.
- Solution is only determined by the examples on the margin (support vectors), i.e., $x^{(\ell)}$ with $\alpha_\ell>0$, hence the name **support vector machine (SVM).**

在边界上的点:支持向量