Recall the hard-margin SVM

- It assumes the data is perfectly separable
- The constraints are $y_i \langle w, x_i \rangle \geq 1 \quad \forall i$
  - not only correctly classified
  - but also $\geq 1$ away from the separating hyperplane
- The optimization minimizes $\frac{1}{2}\|w\|^2$ (1.9)

$\Rightarrow$ Soft-margin SVM, because data is not always perfectly separable

## Soft-margin SVM

Slack variables $\xi_i \geq 0$

Constraints now become $y_i \langle w, x_i \rangle \geq 1 - \xi_i$

- $\xi_i = 0 \Rightarrow$ hard-margin
- $\xi_i \in (0, 1] \Rightarrow$ on the correct side but inside the margin
- $\xi_i > 1 \Rightarrow$ misclassified.

Objective: $\min_{w, \{\xi_i\}_{i=1}^n} \underbrace{\frac{1}{2}\|w\|^2}_{\geq \frac{1}{2}\|w\|^2} + \lambda \sum_{i=1}^n \xi_i$

s.t. $\xi_i \geq 0$

$y_i \langle w, x_i \rangle \geq 1 - \xi_i \quad \forall i$

$\Rightarrow$ Unconstrained opt. problem with the hinge loss: $\ell_{hinge}(y, \hat{y}) = \max\{0, 1 - y\hat{y}\}$

$\Rightarrow \min_w \frac{1}{2}\|w\|^2 + \lambda \sum_i \ell_{hinge}(y_i, \langle w, x_i \rangle)$

For one data point, hinge loss is

$$\ell(y, \langle w, x \rangle) = \max\{0, 1 - y\langle w, x \rangle\}$$

Gradient:

$$\nabla_w \ell = \begin{cases} 0 & y\langle w, x \rangle \geq 1 \\ -yx & y\langle w, x \rangle < 1 \end{cases}$$

$\Rightarrow$ Hinge loss updates not only when a point is misclassified, but also when it is correctly classified but creeping into the margin zone.

Compare with the perceptron gradient

$$\nabla_w \ell_{perc} = \begin{cases} 0 & y\langle w, x \rangle > 0 \\ -yx & y\langle w, x \rangle \leq 0 \end{cases}$$

$\Rightarrow$ Perceptron updates only on outright misclassifications.

One more important piece:

SVM adds the regularization term $\frac{1}{2}\|w\|^2$

- Gradient is simply $w$

$\Rightarrow$ The overall update combines:

① A perceptron-like correction from hinge loss

② A force pulling $w$ to smaller norm, encouraging a large margin.

## Kernels

Let $k_1, k_2$ denote two kernel functions

WTS: $k^+(x,y) = k_1(x,y) + k_2(x,y)$ is a valid kernel function

$$k^+(y,x) = k_1(y,x) + k_2(y,x)$$
$$= k_1(x,y) + k_2(x,y)$$
$$= k^+(x,y) \quad \text{Symmetry } \checkmark$$

Let $K^l$ be the Gram matrix

$$K^l_{ij} = k_l(x_i, x_j) \quad \forall l = 1,2$$

$K^+$ G.m. s.t. $K^+_{ij} = k^+(x_i, x_j)$

$K$ $\quad \langle v, Kv \rangle \geq 0$

$$\langle v, K^+ v \rangle = \langle v, (K^1 + K^2) v \rangle$$
$$= \underbrace{\langle v, K^1 v \rangle}_{\geq 0} + \underbrace{\langle v, K^2 v \rangle}_{\geq 0}$$

$\geq 0$

$\Rightarrow K^+$ P.S.D. ✓

$\Rightarrow k^+(x,y)$ is a valid kernel function.

WTS: $k^*(x,y) = k_1(x,y) \cdot k_2(x,y)$ is a valid kernel function.

$k^*(y,x) = k_1(y,x) \cdot k_2(y,x)$

$= k_1(x,y) \cdot k_2(x,y) = k^*(x,y)$

Let $K^*$ be the Gram matrix s.t.

$K^*_{ij} = k^*(x_i, x_j)$

Hint: Hadamard product of two PSD matrices is also PSD.

$(A \odot B)_{ij} = A_{ij} \cdot B_{ij}$

$K^* = K^1 \odot K^2$ is also PSD

($K^1$ PSD by def, $K^2$ PSD by def)

$\Rightarrow k^*(x,y)$ is also a valid kernel function.

## Polynomial kernel

$k^p(x,y) = (\langle x,y \rangle + \alpha)^p \quad \alpha \geq 0 \quad p \in \mathbb{N}_+$

$k^1(x,y) = \langle x,y \rangle + \alpha$ is a valid kernel function

($\langle x,y \rangle$: kernel; $\alpha$: kernel)

$k^2(x,y) = k^1(x,y) \cdot k^1(x,y)$ is also a kernel

($k^1(x,y)$: kernel; $k^1(x,y)$: also kernel)

$k^3(x,y)$ is also a kernel by multiplication rule

$\vdots$

$k^p(x,y)$ is a kernel by induction