Support Vector Machines and Kernels

September 28, 2022

1 SVM

- **Classification:**
  N training vectors $\{(x_i, y_i)\}$, where $x \in \mathbb{R}^D$ and $y \in \{-1, 1\}$.
- **Classifier:**

$$f(x) = w^T \phi(X) + b$$
$$\hat{y} = sign(f(x))$$

- **Maximum Margin Method:** for a linearly separable dataset:

$$\max_{w,b} \min_i dist(x_i, w, b)$$
$$s.t.: \ \forall i: \ y_i(w^T \phi(x_i) + b \geq 0)$$

  Distance from a datapoint $\phi(x_i)$ to a hyperplane $w^T \phi(X) + b = 0$ is:

$$\frac{|w^T \phi(x) + b|}{\|w\|} = \frac{y_i(w^T \phi(x) + b)}{\|w\|}$$

  Assuming w, b such that nearest point to the hyperplane have $y_i(w^T \phi(x) + b) = 1$ our objective becomes:

$$\max_{w,b} \frac{1}{\|w\|} \equiv \min_{w,b} \frac{1}{2}\|w\|^2 \qquad \text{(A quadratic program)}$$
$$s.t.: \ \forall i: \ y_i(w^T \phi(x_i) + b \geq 1)$$

- **Support Vectors:** Datapoints close to margin.
- **Slack Variables:** We need those for non-separable datasets.
  A slack variable for each datapoint, ξ, that shows how much that datapoint can violate the *margin constraint* (and of course will be punished accordingly!). New optimization problem:

$$\min_{w,b,\xi_{1:N}} \sum_i \xi_i + \lambda \frac{1}{2}\|w\|^2$$
$$s.t.: \ \forall i: \ y_i(w^T \phi(x_i) + b \geq 1 - \xi_i \text{ and } \xi_i \geq 0)$$

- **Loss Functions**
  - 0-1 Loss:

$$L_{0-1}(x,y) = \begin{cases} 1 & yf(x) < 0 \\ 0 & \text{otherwise} \end{cases}$$

  - Logistic Regression:

$$L_{LR} = \ln(1 + e^{-yf(x)}) \tag{1}$$

  - Hinge Loss:

$$\begin{cases} 1 - yf(X) & yf(x) < 0 \\ 0 & \text{otherwise} \end{cases}$$

2 Largrangian and the Kernel Trick

Using Lagrangian (and assuming linearly separability)

$$L(w, b, a_{1:N}) = \frac{1}{2}\|w\|^2 - \sum_i a_i(y_i(w^T\phi(x_i) + b) - 1)$$
$$\Rightarrow \begin{cases} w = \sum_i a_i y_i \phi(x_i) \\ \sum_i y_i a_i = 0 \end{cases}$$

Dual Lagrangian:

$$L(a_{1:N}) = \sum_i a_i - \frac{1}{2}\sum_i\sum_j a_i a_j y_1 y_2 \phi(x_i)^T\phi(x_j)$$

With using kernel funcion:

$$L(a_{1:N}) = \sum_i a_i - \frac{1}{2}\sum_i\sum_j a_i a_j y_1 y_2 k(x_i, x_j)$$

- **Popular kernels:**
  - Polynomial kernel:

$$k(x,z) = (x \cdot z + 1)^d$$

  - Gaussian kernel:

$$k(x,z) = e^{-\frac{\|x-z\|^2}{2\sigma^2}}$$

  - RBF kernel:

$$k(x,z) = e^{-\gamma\|x-z\|^2}$$

- **Predicting:**

$$\begin{aligned} f(x_{new}) &= w^T \phi(x_{new}) + b \\ &= \left(\sum_i a_i y_i \phi(x_i) \right)^T \phi(x_j) + b \\ &= \sum_i a_i y_i k(x_i, x_{new}) + b \end{aligned}$$