08_Support_Vector_Machine

For **linear seperable** training sample:
Hyperplane(decision boundary): separate dataset with maximal margins

Location of $Hyperplane$: **weights** $\vec{\omega}$ (orientation of boundary)the **bias** $\vec{b}$(bias).
Find the values for $\vec{\omega}$ and $\vec{b}$ which $maximizes\ the\ margin$, i.e. the distance to any datapoint.

Optimization problem:

$$
\begin{aligned}
& argmin_{\vec{\omega},\vec{b}} \frac{1}{2}\|\vec{\omega}\|^2 \\
& s.t. \quad \vec{t}_i(\vec{\omega}\ \phi(\vec{x}_i) - b) \geq 1, \forall i \\
\\
& \vec{x}_i : \quad i_{th}\ data\ point \\
& \vec{t}_i : \quad 2\ class for\ \vec{x}_i, positive = 1, negetive = -1 \\
& \phi(\vec{x}_i) : \quad transformation for\ dataset,\ i.e. Kernell
\end{aligned}
$$

Once solving the optimazation problem, it gets $\vec{\alpha}\&b$.The points with $none\ zero\ \alpha_i$ is so called $Support\ Vector$ which is exactly on the margin. Utilizing SV $\vec{s}$ to classfy a new data point $\vec{x}$ by the indicator function:

$$ind(\vec{x}) = \omega\ \phi(\vec{x}) - b$$

Transform to **Dual Formulation** with computational advantages:

$$
\begin{aligned}
& argmin_{\alpha_i} \ \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j t_i t_j K(\vec{x}_i, \vec{x}_j) - \sum_i \vec{\alpha} \\
& s.t. \quad 0 \leq \alpha_i \leq C, \quad \sum_i \alpha_i t_i = 0 \\
& ind(\vec{x}) = \sum_{i=1}^{n} \alpha_i t_i K(\vec{x}, \vec{s}_i) - b, \quad \vec{x} : new\ data\ point, \vec{s} : support\ vectors
\end{aligned}
$$

Parameter C is problem called soft margin or slack variable. For non-linear separable training sample, slack allows datapoints to be miss-classified if it results in a substantially wider margin.

Kernel Function:
By transforming the input data non-linearly to a high-dimensional space, more complex decision boundaries can be utilized.

1. Linear Kernel:

   $$K(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T \vec{x}_j + 1$$

   Linear Kernel returns the scalar product between the two points.Result in a linear separation.

2. Polynomial Kernel:

$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i^T \vec{x}_j + 1)^p$$

Polynomial kernel allows for curved decision boundaries.P(a positive integer) controls the degree of the polynomials.Higher p results in more complex shapes.

3. Radial Basis Function RBF Kernel:

$$K(\vec{x}_i, \vec{x}_j) = e^{-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2}}$$

RBF kernel uses the explicit euclidian distance between the two datapoints,results in very good boundaries. The parameter sigma is used tocontrol the smoothness of the boundary. High sigma-smooth. Low sigma-rough.

Implement see in Lab2