SMO Algorithm

For dual problem of SVM:

$$D(\alpha, \mathbf{x}, y) \max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{i=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}$$

$$s.t. \quad \sum_{i=1}^{m} \alpha_i y_i = 0, \quad \alpha_i \geq 0, i = 1, 2, \cdots, m.$$

The KKT condition for hard margin is given as:

$$\begin{cases} y_i(\omega_i^T \mathbf{x_i} + b) - 1 > 0, (\alpha_i = 0), \\ y_i(\omega_i^T \mathbf{x_i} + b) - 1 = 0, (\alpha_i > 0) \quad i = 1, 2, \cdots, m. \end{cases}$$

And the KKT condition for soft margin is given by:

$$\begin{cases} y_i(\omega_i^T \mathbf{x_i} + b) - 1 > 0, (\alpha_i = 0) \\ y_i(\omega_i^T \mathbf{x_i} + b) - 1 = 0, (0 < \alpha_i < C) \quad i = 1, 2, \cdots, m. \\ y_i(\omega_i^T \mathbf{x_i} + b) - 1 < 0, (\alpha_i = C) \end{cases}$$

The main idea of SMO algorithm is to select two variables α_i, α_j while other variables are fixed. By selecting and updating different pairs of variables until all the variables converge, the model can be optimized.

Suppose α_i, α_j are selected and we have:

$$y_i \alpha_i + y_j \alpha_j = M := \sum_{k \neq i,j}^{m} \alpha_k y_k, \quad K \text{ is constant}$$
$$\alpha_j = y_j M - y_i y_j \alpha_i,$$

As α_i, α_j should satisfy KKT condition, for the new $\alpha_{i,new}$, to be updated, we have the boundaries for hard margin:

$$0 \leq \alpha_{i,new} \leq \alpha_i + \alpha_j, \quad (y_i y_j = 1)$$
$$\max(\alpha_i - \alpha_j, 0) \leq \alpha_{i,new}, \quad (y_i y_j = -1)$$

And for soft margin:

$$\max(\alpha_i + \alpha_j - C, 0) \leq \alpha_{i,new} \leq \min(\alpha_i + \alpha_j, C), \quad (y_i y_j = 1)$$
$$\max(\alpha_i - \alpha_j, 0) \leq \alpha_{i,new} \leq \min(C + \alpha_i - \alpha_j, C), \quad (y_i y_j = -1)$$

If we define:

$$K_{i,j} = \mathbf{x_i}^T \mathbf{x_j} (\text{for kernel } \phi(\mathbf{x_i})^T \phi(\mathbf{x_j}))$$
$$E_i = \omega^T \mathbf{x_i} + b - y_i = \sum_{k=1}^{m} \alpha_k y_k K_{i,k} - y_k$$

The dual problem will become an easy quadratic optimization problem which only contains α_i.

And the new α_i can be calculated:

$$\hat{\alpha}_{i,new} = \alpha_{i,old} + \frac{y_i(E_i - E_j)}{K_{1,1} + K_{2,2} - 2K_{1,2}}$$

The calculated result $\hat{\alpha}_{i,new}$ should satisfy the boundary condition above. When it is smaller than the lower boundary, it should be updated as the lower boundary and be updated as the upper boundary if greater than the upper boundary.

The algorithm can be describe as follows:

```
SMO:
Input: x[m][d], y[m],alpha[m]
Output: convergent alpha[m]

while(alpha[m] unconvergent){
    pick i,j
    calculate alpha_new.
    update the alpha_i with boundary condition.
}
```