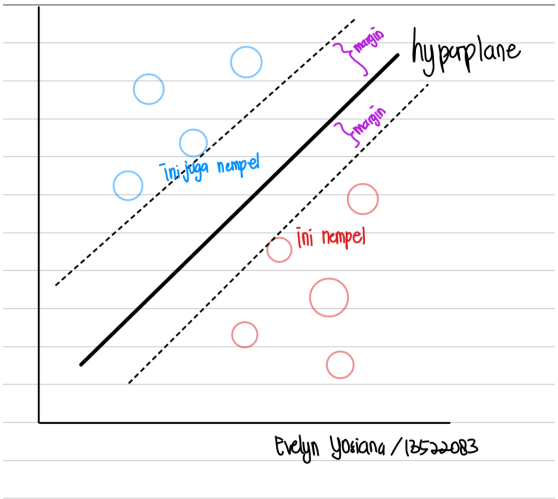
Support Vector Classifier (SVC)

Evelyn Yosiana / 13522083

1. Konsep SVC pada dasarnya yaitu mencari hyperplane (2D, 3D, dan seterusnya) untuk memaksimalkan margin antarkelas data. Margin sendiri merupakan jarak antara hyperplane (garis pemisah) dengan titik data paling dekat dengan hyperplane tersebut. Untuk ilustrasinya sebagai berikut.



Proses train: untuk proses train, perhitungan data dilakukan untuk mencari weight dan bias. Persamaan yang digunakan adalah

$$weight = weight - learning\ rate\ (2 * lamda * w)$$

bias tetap

ketika suatu titik data sudah diklasifikasikan dengan benar berdasarkan hyperplane yang telah dibuat (condition TRUE jika dalam kode), atau

$$weight = weight - learning\ rate\ (2 * lamda * weight - [x_i\ y_i'])$$

$$bias = bias - learning\ rate * y_i'$$

jika titik yang sedang diiterasi kondisinya sudah benar (condition TRUE).

Jadi pada dasarnya, weight dan bias digunakan untuk membentuk hyperplane (garis pemisah).

Proses predict: Setelah didapatkan weight dan bias, data di-mapping ke dimensi yang lebih tinggi menggunakan kernel function. Kemudian tiap data diklasifikasikan berdasarkan threshold yang digunakan.

Parameter:

- learning rate: untuk mengontrol perubahan weight, semakin besar maka akan semakin cepat konvergen, namun bisa jadi melewatkan titik tertentu.
- lamda_param: untuk mengontrol besaran penalti.
- N_iterations: banyak iterasi yang dilakukan.
- kernel: untuk mapping data ke dimensi yang lebih tinggi.
- threshold: ambang batas untuk menentukan prediksinya 0 atau 1.

4. Pada model ini, hasil perbandingan model yang saya buat dan model dari library kurang sama. Hal ini dapat disebabkan oleh beberapa faktor, antara lain:
   - **Pendefinisian threshold**: pada kode saya, threshold didefinisikan secara manual, sedangkan dalam library, threshold ditentukan secara otomatis berdasarkan hyperplane yang terbentuk.
   - **Penalty term**: regularisasi atau penalty term di library dibuat lebih optimal dan otomatis sehingga dapat meminimalisasi overfitting dan underfitting. Dalam kode saya, regularisation (lamda_param) diatur secara manual dan tidak berubah seiring dengan pengembangan model.
   - **Implementasi kernel function**: dalam kode di library, kernel function telah dioptimasi sedemikian rupa sehingga lebih efisien dan kompleks. Kode saya sendiri menggunakan implementasi kernel function yang relatif sederhana.
   - **Stopping criteria**: kode yang saya buat menggunakan iteration sebagai batas iterasi, sedangkan kode dalam library menggunakan batas iterasi berdasarkan konvergensi.

5. Improvement yang dapat saya lakukan antara lain:
   - **Menambahkan stopping criteria** untuk menghentikan proses training apabila model sudah konvergen selama proses training sehingga dapat menurunkan resiko overfitting.
   - **Hyperparameter tuning** untuk mendapatkan kombinasi parameter yang optimal (salah satunya dengan menggunakan metode grid search atau library optuna).
   - Menggunakan **confusion metrics** untuk model evaluation.

Lampiran

Contoh perhitungan manual:

Evelyn Yosiana / 13522083

SVC

$x\ train = \begin{bmatrix} 1 & 3 \\ 4 & 2 \\ 5 & 2 \end{bmatrix}$ $y\ train = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ $lr = 0{,}001$ $\lambda = 0{,}01$

transforming : $y'\ train = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$

FORMULA!

condition $= y' \cdot (w \cdot x - b)$

update $w = w - lr\ (2\lambda w - xy')$

update $b = b - lr \cdot y'$

approx $= w \cdot x - b$

$w = [0, 0]$

$b = 0$

Sample 1 : $x_1 = [1,3]$ $y_1' = -1$

condition $= -1\ (0 \cdot [1,3] - 0) = 0 \longrightarrow 0 \geq 1$ FALSE

update $w = [0,0] - 0{,}001\ (0 - [-1,-3]) = [0.001, 0.003]$

$b = 0 - 0{,}001\ (-1) = 0{,}001$

Sample 2 : $x_2 = [4,2]$ $y_2' = 1$

condition $= 1\ ([0.001, 0.003] \cdot [4,2] - 0{,}001) = 0{,}001 - 0{,}001 = 0 \geq 1$ FALSE

update $w = [0.001, 0.003] - 0.001\ (2 \cdot 0.01\ [0.001, 0.003] - [4,2]) = [0.005, 0.005]$

$b = 0.001 - 0.001\ (1) = 0$

Sample 3 : $x_3 = [5,2]$ $y_3' = -1$

condition $= -1\ ([0.005, 0.005]\ [5,2] - 0) = -0.035 \geq 1$ FALSE

update $w = [0.005, 0.005] - 0.001\ ([0.001, 0.001] - [-5,-2]) = [0.01, 0.007]$

$b = 0 - 0.001\ (-1) = 0.001$

$w = [0.01, 0.007]$ $b = 0.001$

predict [3,7] :

approx $= [0.01, 0.007]\ [3,7] - 0.001$

$= 0{,}051 \longrightarrow$ karena $0{,}051 > 0$ maka hasil prediksinya 1