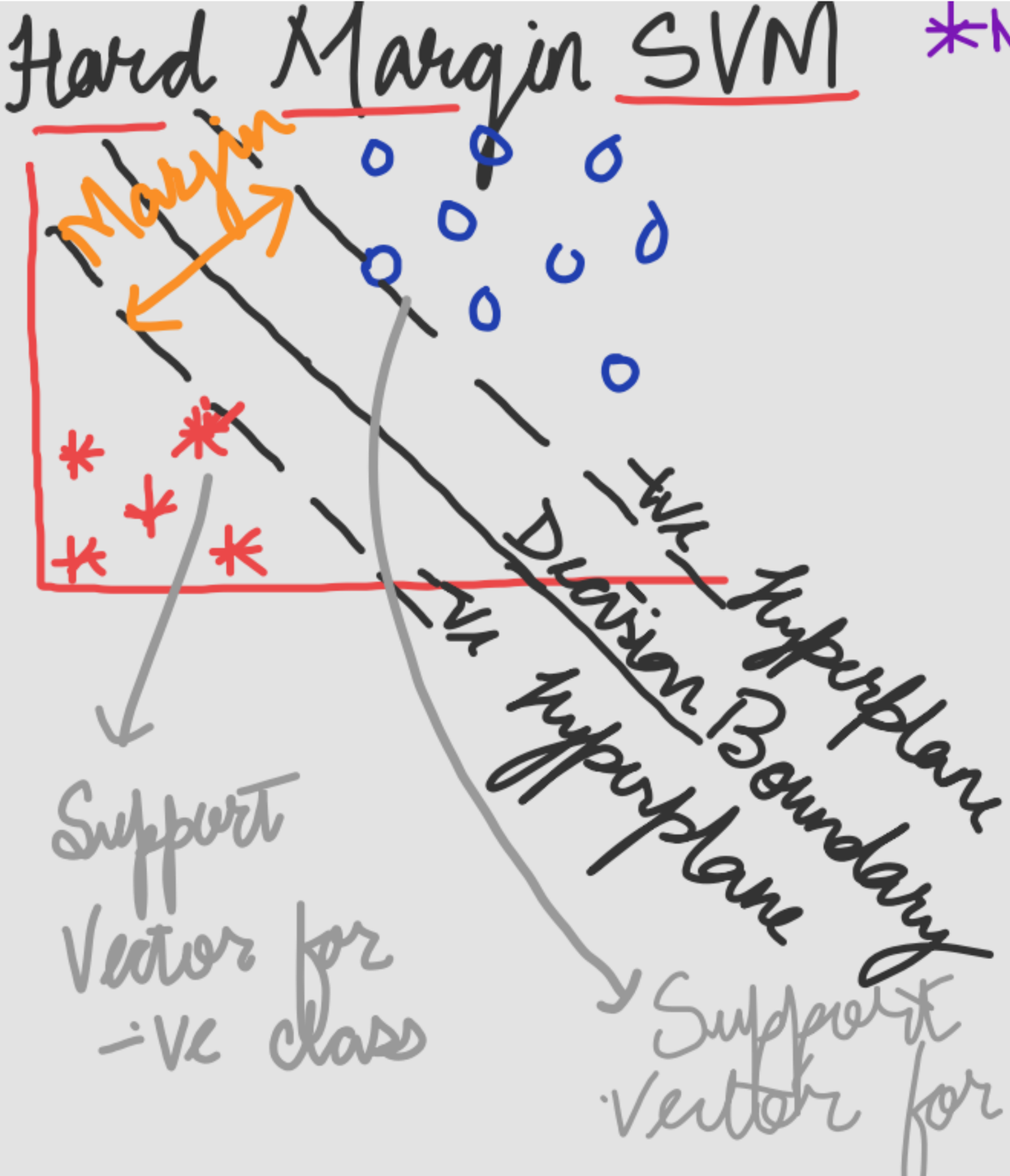
Hard Margin SVM



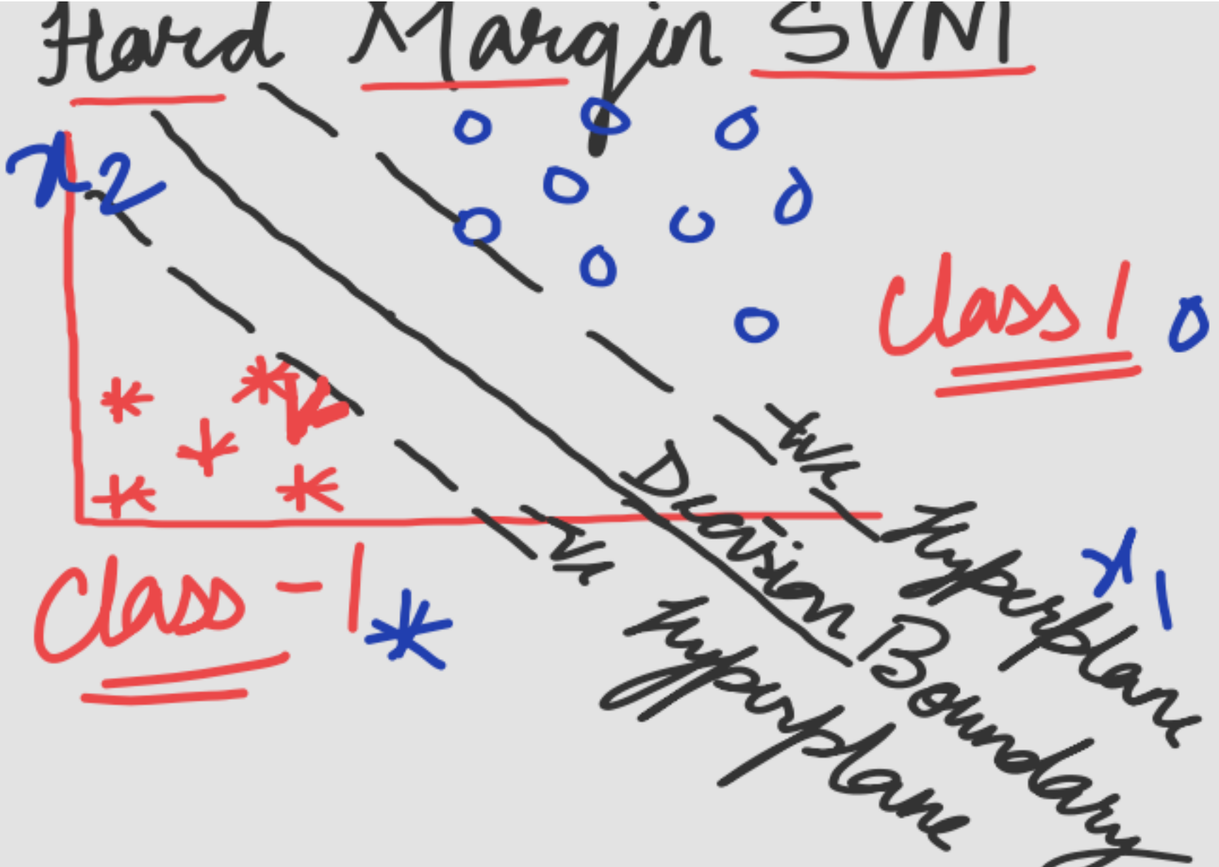
*Note : Refer class notes

Aim

To come up with the decision boundary that can act as max. margin classifier i.e. distance of support vectors from decision boundary is max.

Margin : Distance of Support Vectors (closest pt. from division boundary) from the decision boundary

Hard Margin SVM



Eqn of Decision Boundary

$$w_1 x_1 + w_2 x_2 + b = 0$$

or

$$\vec{w}\vec{x} + b = 0$$

where $w = [w_1 \quad w_2]$, $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Eqn of positive Hyperplane : $\vec{w}\vec{x} + b = 1$

Eqn of negative Hyperplane : $\vec{w}\vec{x} + b = -1$

Thus, the Constraints that training data should Satisfy:

If $y = 1$, then $\vec{w}\vec{x} + b \geq 1$

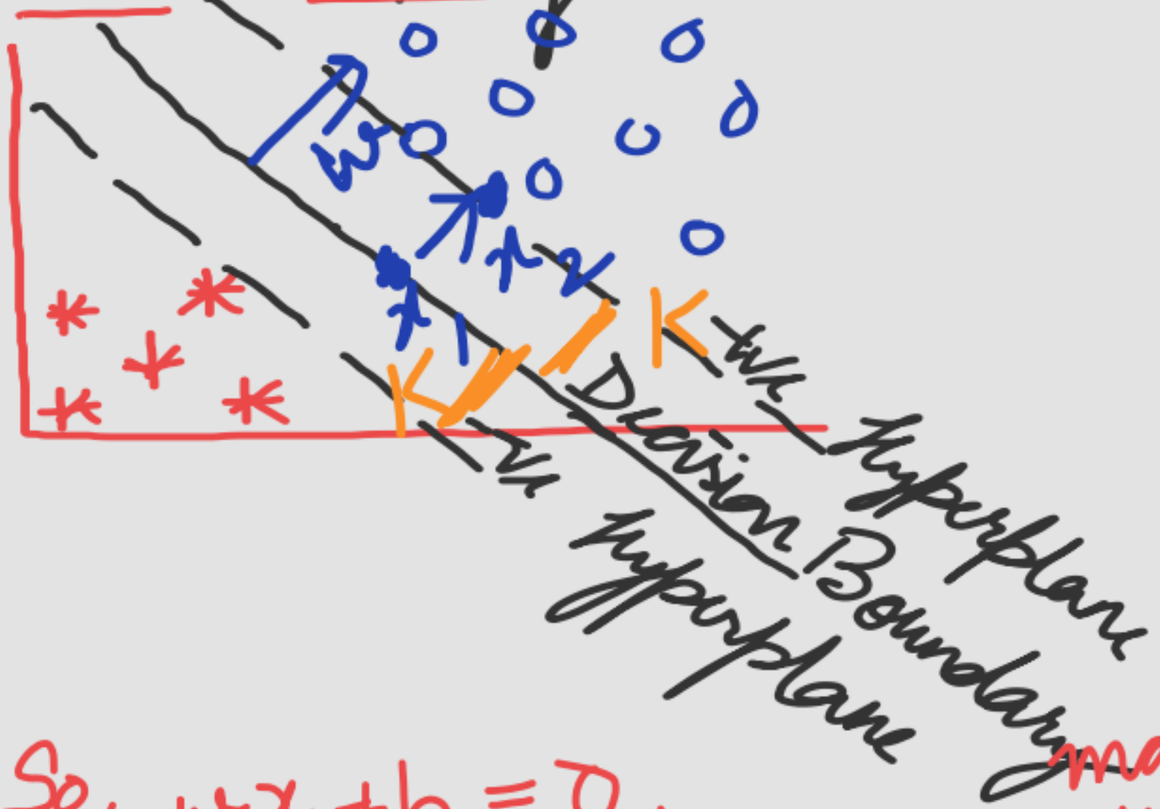
If $y = -1$, then $\vec{w}\vec{x} + b \leq -1$

OR

$$(y)\cdot(\vec{w}\vec{x} + b) \geq 1$$

$\therefore$

Hard Margin SVM



w : vector which is normal or perpendicular to the decision boundary

Consider pt x_1 on decision boundary & x_2 on +ve hyperplane which is K units away

So, $wx_1 + b = 0$

To reach pt x_2 we go the K. margin can be found as

$w\left(x_1 + K\frac{w}{\|w\|}\right) + b = 1$

$\left(\therefore x_2 = x_1 + K\frac{w}{\|w\|}\right)$

K units in → direction of w

Refer class Notes!

$$\Rightarrow \vec{w}\vec{x}_1 + \frac{K\,\vec{w}\cdot\vec{w}}{\|w\|} + b = 1 \qquad \left(\because \vec{w}\vec{x} + b = 0\right)$$

$$\Rightarrow K = \frac{\|w\|}{\vec{w}\cdot\vec{w}} \qquad \left(\because \vec{w}\vec{w} = \|w\|^2\right)$$

$$= \frac{\|w\|}{\|w\|^2} = \frac{1}{\|w\|}$$

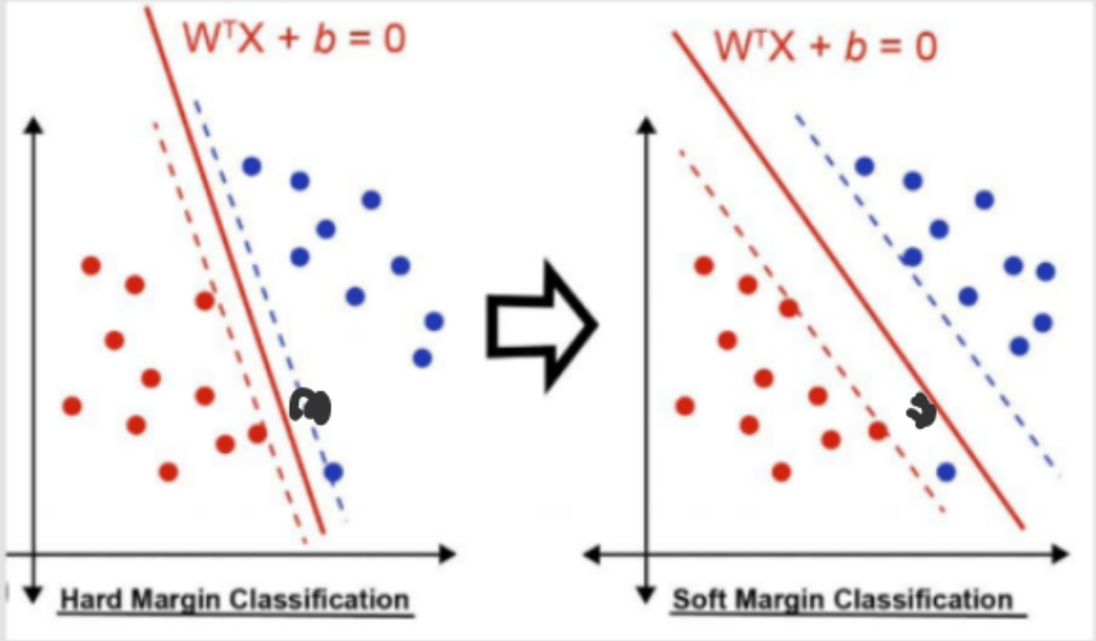
Total Margin $= 2K = \frac{2}{\|w\|}$

Optimization Problem of Hard margin SVM:

Max $\frac{2}{\|w\|}$ or Min. $\frac{\|w\|}{2}$

subject to $y^{(i)}\left(\vec{w}\vec{x}^{(i)} + b\right) \geq 1 \quad \forall i = 1, 2, 3, \ldots N$

Hard Margin vs Soft Margin



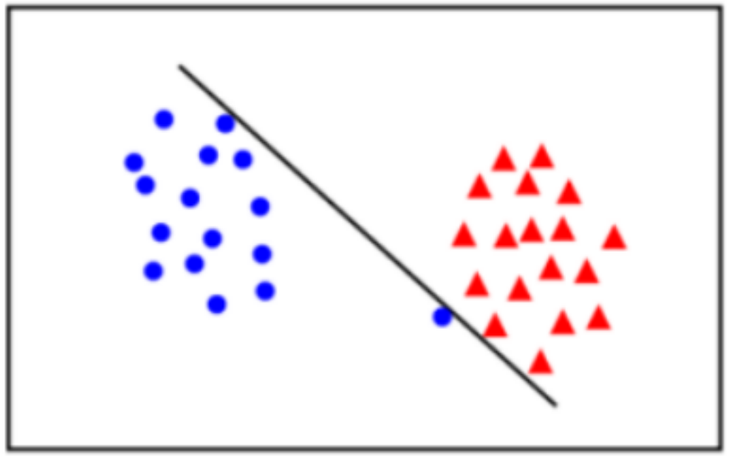
Soft Margin: try to find a line to separate, but tolerate one or few misclassified dots (e.g. the dots circled in red)

which case denotes overfitting?

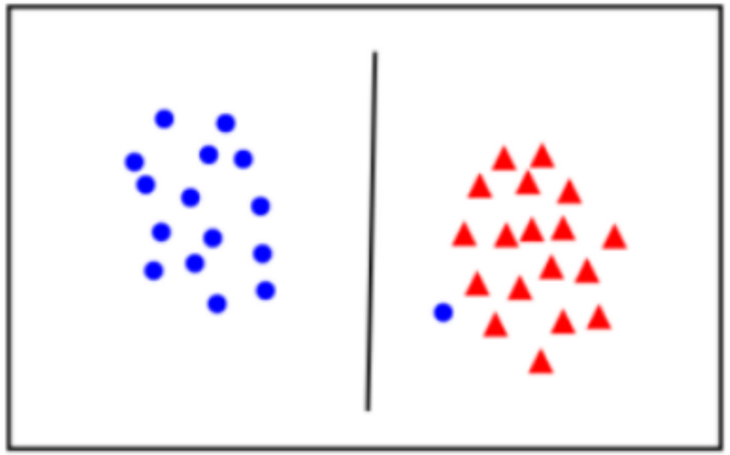
→ Hard Margin

→ Soft Margin

* Generally data is not linearly separable



• the points can be linearly separated but there is a very narrow margin



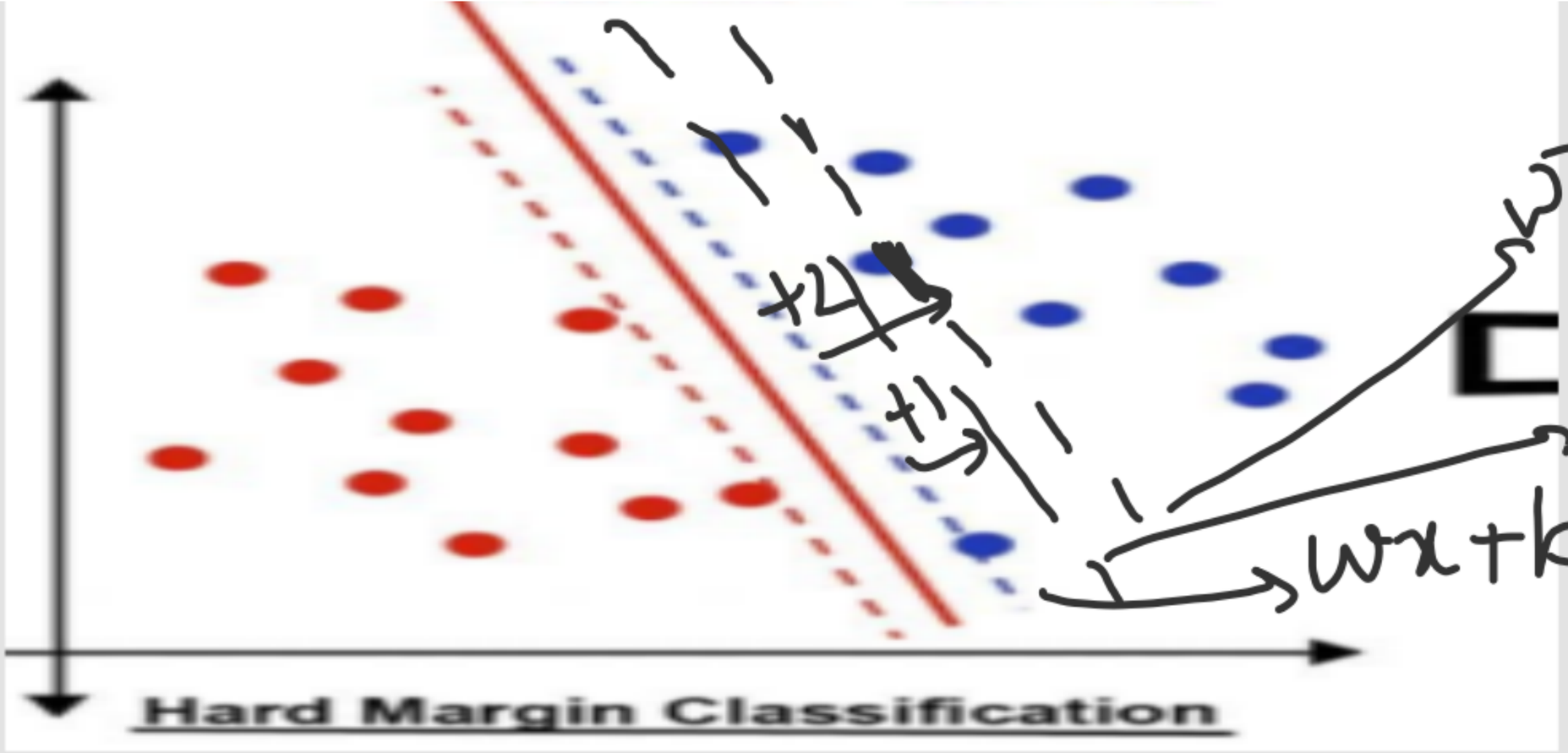
• but possibly the large margin solution is better, even though one constraint is violated

In general there is a trade off between the margin and the number of mistakes on the training data

Idea: Give certain penalties for mistakes based on how big they are

→ Avoid overfitting of our model by introducing slack var ξ_i one for each example which allow some of the points to be inside margin

Aim: Max. margin & min. misclassifications / margin error



Optimization Problem of Soft margin SVM:

Max $\Rightarrow$ or Min. $\left(\frac{\|w\|}{2} + C \sum_{i=0}^{N} \xi_i \right)$

subject to $y^{(i)}(\vec{w}\vec{x}^{(i)}+b) \geq 1+\xi_i \quad \forall i=1,2,3,\ldots N$

C: How much misclassifications you meant to avoid

$C \uparrow$, $\xi_i \downarrow$ → Less misclassifications
↳ small margin

$C \downarrow$ / Misclassification

Optimization Problem of Soft margin SVM:

$$\min \frac{\|w\|}{2} + C \sum_{i=0} \xi_i$$

subject to $y^{(i)}(\vec{w}\vec{x}^{(i)} + b) \geq 1 + \xi_i \quad \forall i = 1, 2, 3, \ldots N$

C: Large Eg, ↓

↳ Lesser misclassification

→ Optimizer will choose smaller margin hyperplane

C: Small

↳ Entire focus on first term

↳ Optimizer will look for large margin hyperplane.

Regularization

<Refer class Notes>

→ Technique to avoid overfitting by shrinking the value of parameters.

$f(x) = 29x - 385x^2 + 39x^3 - 174x^4 + 10$

$f(x) = 13x - 0.23x^2 + 0.00014x^3 - 0.0001x^4 + 10$

Idea:

Update cost func. by penalizing weight parameters

L1 Regularization (Lasso Regression)

$$J(w,b) = \frac{1}{2N} \sum_{i=1}^{N} \left(y^{(i)} - f(x^{(i)})\right)^2 + \frac{\lambda}{2N} \sum_{j=1}^{2} |w_j|$$

L2 Regularization (Ridge Regression)

$$J(w,b) = \frac{1}{2N} \sum_{i=1}^{N} \left(y^{(i)} - f(x^{(i)})\right)^2 + \frac{\lambda}{2N} \sum_{j=1}^{2} w_j^2$$

Always Optional

Ques. Why λ should not be too small or too high!

If $\lambda = 0$ ⇒ No regularization
⇒ Fit Overly wiggly, Overly complex curve.
↳ Effect of overfitting data
Low Train Error, Large Generalization / Prediction Error

If $\lambda \to$ Very Very large ⇒ Placing heavy weight on regularization term
⇒ learning algo will choose $w_1, w_2, \ldots$ to be extremely close to 0
⇒ $f(x^{(i)})$ is basically equal to b.
↳ Learning algo will fit a straight line ↑ horizontal
if it underfits
Large Train Error & Large Generalization / Prediction Error

Choose λ that balances first & second terms of its objective
↳ Min MSE & keeps the parameters small

The differences between L1 and L2 regularization:

L1 regularization penalizes the sum of absolute values of the weights

↳ Lasso shrinks weights of less imp. features to zero.
↳ has the effect of feature selection

L2 regularization penalizes the sum of squares of the weights.

$$J = \frac{1}{2N}\sum_{i=1}^{N}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \frac{\lambda}{2N}\sum_{j=1}^{d} w_j^2$$

$\hat{y}^{(i)} \rightarrow w_1x_1 + w_2x_2 + \dots + w_dx_d + b$

Gradient Descent for

Linear Regression

repeat until Convergence {

$$b = b - \alpha \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}^{(i)} - y^{(i)}\right)$$

$$w_j = w_j - \alpha \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}^{(i)} - y^{(i)}\right)x_j^{(i)}$$

}

$$w_1 = w_1 - \alpha \frac{\partial J}{\partial w_1}$$

Regularized Linear Regression

SAME

$$w_j = w_j - \alpha \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}^{(i)} - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{N} w_j$$

$$\frac{\partial J}{\partial w_1} = \frac{\partial}{\partial w_1}\left(\frac{1}{2N}\left(w_1^2 + w_2^2 + \dots + b\right)\right)$$

(Term)

$$B = \frac{\lambda}{2N} \sum_{j=1}^{d} \left(w_j\right)^2$$

$$\frac{\partial B}{\partial w_1} = \frac{\partial}{\partial w_1}\left(\frac{\lambda}{2N}\left(w_1^2 + w_2^2 + \cdots + w_d^2\right)\right)$$

$$= \frac{\lambda}{2N} \times 2w_1 = \frac{\lambda}{N} w_1$$

$$J = -\frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} \log\left(f(x^{(i)})\right) + (1-y^{(i)}) \log\left(1 - f(x^{(i)})\right) \right) + \frac{\lambda}{2N} \sum_{j=1}^{d} w_j^2$$

$$f(x^{(i)}) = \frac{1}{1 + e^{-(w_1 x_1 + w_2 x_2 + \cdots w_d x_d + b)}}$$

Gradient Descent for

Logistic Regression

repeat until Convergence {

$$b = b - \alpha \frac{1}{N} \sum \left(f(x^{(i)}) - y^{(i)} \right)$$

$$w_j = w_j - \alpha \frac{1}{N} \sum_{i=1}^{N} \left(f(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

}

Regularized Logistic Regr

{ ⟶ SAME

$$w_j = w_j - \alpha \frac{1}{N} \sum_{i=1}^{N} \left(f(x^{(i)}) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{N} w_j$$

}