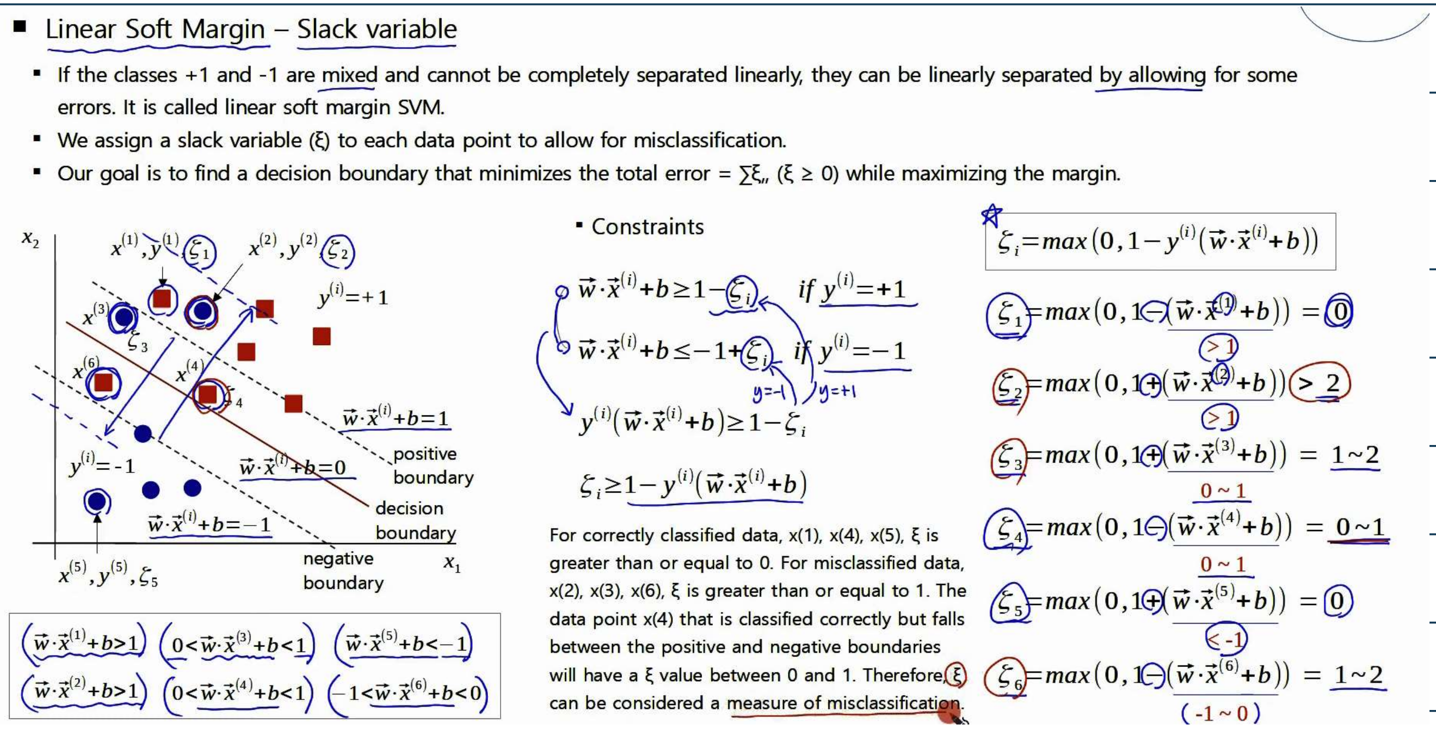
Soft Margin Revisited

Thursday 23 May 2024 8:22 AM



Linear Soft Margin – Slack variable

- If the classes +1 and -1 are mixed and cannot be completely separated linearly, they can be linearly separated by allowing for some errors. It is called linear soft margin SVM.
- We assign a slack variable (ξ) to each data point to allow for misclassification.
- Our goal is to find a decision boundary that minimizes the total error = ∑ξ,, (ξ ≥ 0) while maximizing the margin.

x_2
$x^{(1)}, y^{(1)}, \xi_1$
$x^{(2)}, y^{(2)}, \xi_2$
$y^{(i)}=+1$
$x^{(3)}$
ξ_3
$x^{(6)}$
$x^{(4)}$
ξ_4
$\vec{w}\cdot\vec{x}^{(i)}+b=1$
positive boundary
$y^{(i)}=-1$
$\vec{w}\cdot\vec{x}^{(i)}+b=0$
decision boundary
$\vec{w}\cdot\vec{x}^{(i)}+b=-1$
negative boundary
x_1
$x^{(5)}, y^{(5)}, \xi_5$

$(\vec{w}\cdot\vec{x}^{(1)}+b>1)$ $(0<\vec{w}\cdot\vec{x}^{(3)}+b<1)$ $(\vec{w}\cdot\vec{x}^{(5)}+b<-1)$

$(\vec{w}\cdot\vec{x}^{(2)}+b>1)$ $(0<\vec{w}\cdot\vec{x}^{(4)}+b<1)$ $(-1<\vec{w}\cdot\vec{x}^{(6)}+b<0)$

- Constraints

$$\vec{w}\cdot\vec{x}^{(i)}+b\geq 1-\xi_i \quad if\ y^{(i)}=+1$$

$$\vec{w}\cdot\vec{x}^{(i)}+b\leq -1+\xi_i \quad if\ y^{(i)}=-1$$

y=-1 y=+1

$$y^{(i)}(\vec{w}\cdot\vec{x}^{(i)}+b)\geq 1-\xi_i$$

$$\xi_i\geq 1-y^{(i)}(\vec{w}\cdot\vec{x}^{(i)}+b)$$

For correctly classified data, x(1), x(4), x(5), ξ is greater than or equal to 0. For misclassified data, x(2), x(3), x(6), ξ is greater than or equal to 1. The data point x(4) that is classified correctly but falls between the positive and negative boundaries will have a ξ value between 0 and 1. Therefore, ξ can be considered a measure of misclassification.

$$\boxed{\xi_i=max(0,1-y^{(i)}(\vec{w}\cdot\vec{x}^{(i)}+b))}$$

$$\xi_1=max(0,1-\underset{>1}{(\vec{w}\cdot\vec{x}^{(1)}+b)})=0$$

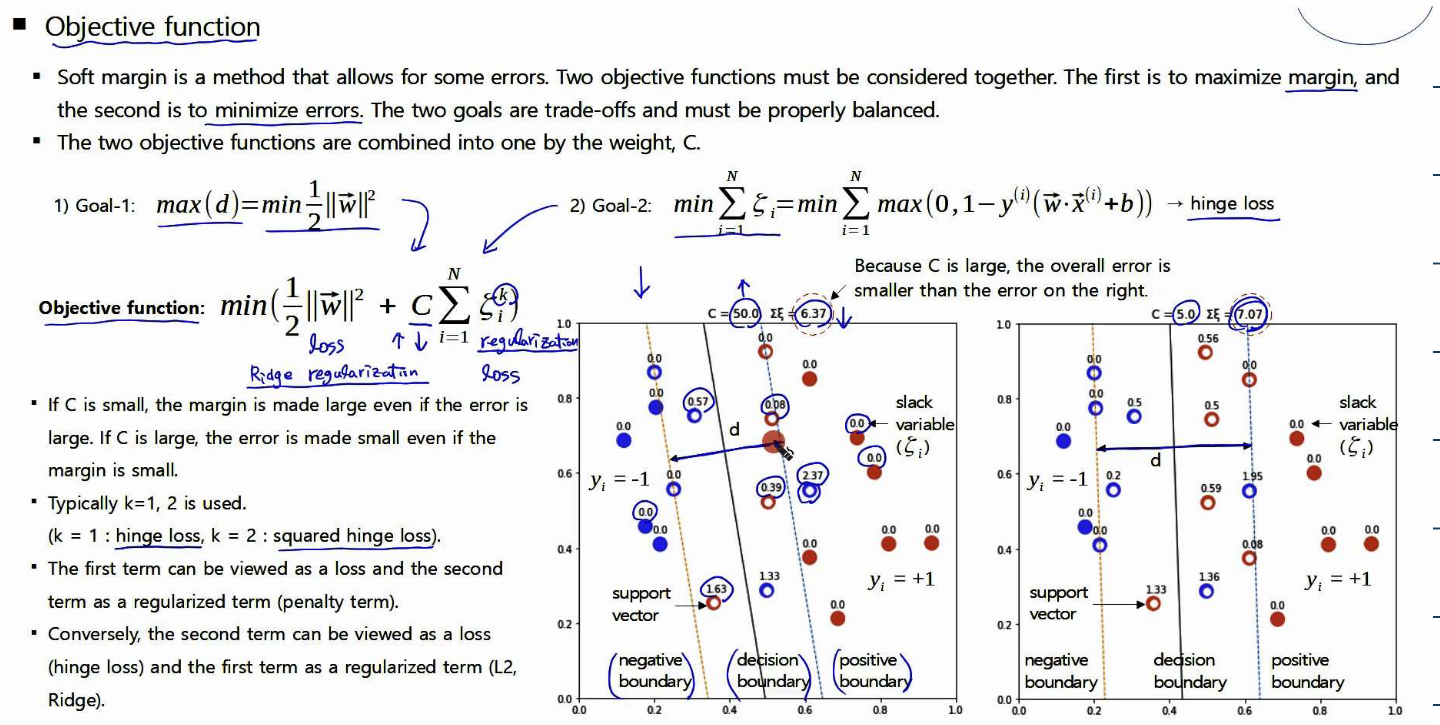
$$\xi_2=max(0,1+\underset{>1}{(\vec{w}\cdot\vec{x}^{(2)}+b)})>2$$

$$\xi_3=max(0,1+\underset{0\sim1}{(\vec{w}\cdot\vec{x}^{(3)}+b)})=1\sim2$$

$$\xi_4=max(0,1-\underset{0\sim1}{(\vec{w}\cdot\vec{x}^{(4)}+b)})=0\sim1$$

$$\xi_5=max(0,1+\underset{<-1}{(\vec{w}\cdot\vec{x}^{(5)}+b)})=0$$

$$\xi_6=max(0,1-\underset{(-1\sim0)}{(\vec{w}\cdot\vec{x}^{(6)}+b)})=1\sim2$$



Objective function

- Soft margin is a method that allows for some errors. Two objective functions must be considered together. The first is to maximize margin, and the second is to minimize errors. The two goals are trade-offs and must be properly balanced.
- The two objective functions are combined into one by the weight, C.

1) Goal-1: $max(d)=min\frac{1}{2}\|\vec{w}\|^2$

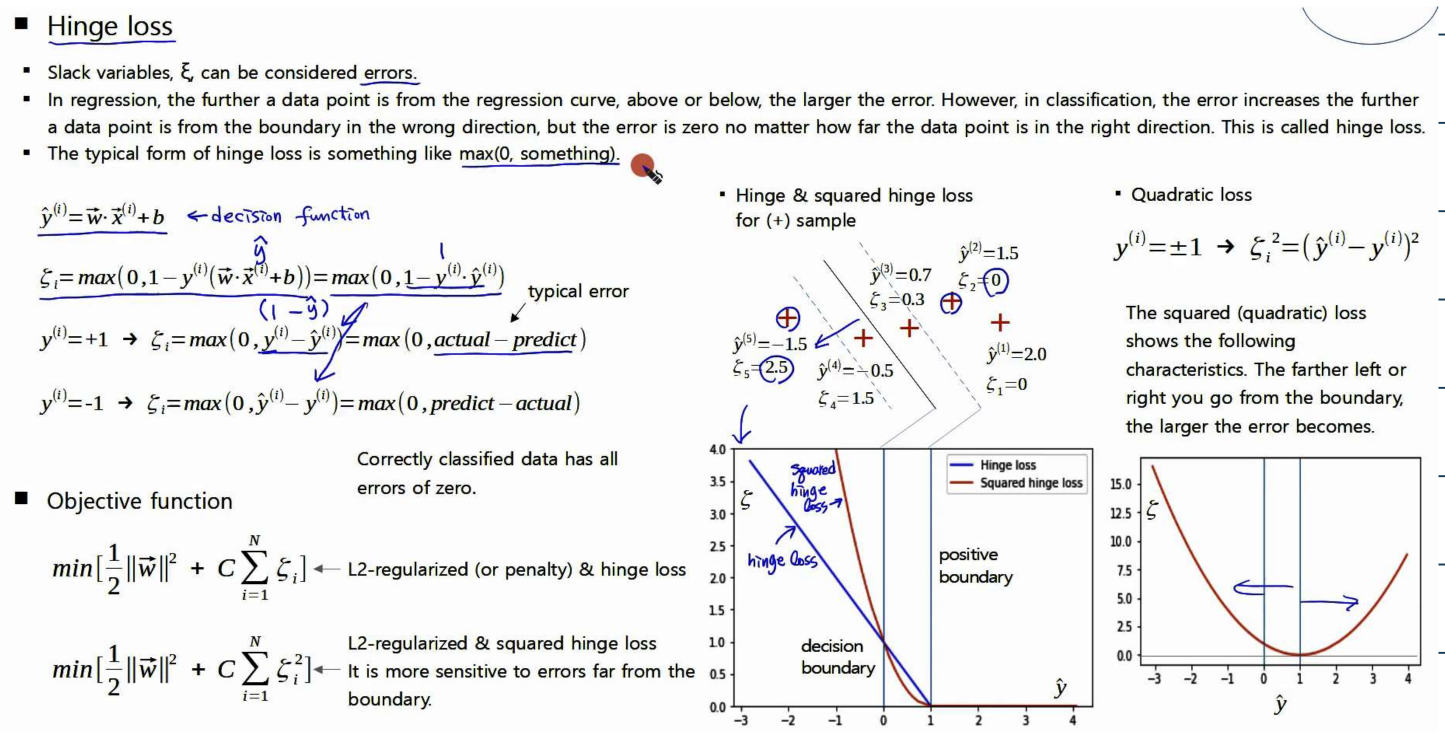
2) Goal-2: $min\sum_{i=1}^{N}\xi_i=min\sum_{i=1}^{N}max(0,1-y^{(i)}(\vec{w}\cdot\vec{x}^{(i)}+b))$ → hinge loss

Objective function: $min(\underset{\text{loss}}{\frac{1}{2}\|\vec{w}\|^2}+C\sum_{i=1}^{N}\underset{\text{regularization}}{\xi_i^{(k)}})$

Ridge regularization loss

- If C is small, the margin is made large even if the error is large. If C is large, the error is made small even if the margin is small.
- Typically k=1, 2 is used. (k = 1 : hinge loss, k = 2 : squared hinge loss).
- The first term can be viewed as a loss and the second term as a regularized term (penalty term).
- Conversely, the second term can be viewed as a loss (hinge loss) and the first term as a regularized term (L2, Ridge).

Because C is large, the overall error is smaller than the error on the right.



Hinge loss

- Slack variables, ξ, can be considered errors.
- In regression, the further a data point is from the regression curve, above or below, the larger the error. However, in classification, the error increases the further a data point is from the boundary in the wrong direction, but the error is zero no matter how far the data point is in the right direction. This is called hinge loss.
- The typical form of hinge loss is something like max(0, something).

$\hat{y}^{(i)}=\vec{w}\cdot\vec{x}^{(i)}+b$ ← decision function

$\xi_i=max(0,1-y^{(i)}(\overset{\hat{y}}{\vec{w}\cdot\vec{x}^{(i)}+b}))=max(0,1-\overset{1}{y^{(i)}}\cdot\hat{y}^{(i)})$ typical error

(1 − ŷ)

$y^{(i)}$=+1 → $\xi_i=max(0,y^{(i)}-\hat{y}^{(i)})=max(0,actual-predict)$

$y^{(i)}$=-1 → $\xi_i=max(0,\hat{y}^{(i)}-y^{(i)})=max(0,predict-actual)$

Correctly classified data has all errors of zero.

Objective function

$min[\frac{1}{2}\|\vec{w}\|^2+C\sum_{i=1}^{N}\xi_i]$ ← L2-regularized (or penalty) & hinge loss

$min[\frac{1}{2}\|\vec{w}\|^2+C\sum_{i=1}^{N}\xi_i^2]$ ← L2-regularized & squared hinge loss It is more sensitive to errors far from the boundary.

- Hinge & squared hinge loss for (+) sample

- Quadratic loss

$$y^{(i)}=\pm1 \rightarrow \xi_i^2=(\hat{y}^{(i)}-y^{(i)})^2$$

The squared (quadratic) loss shows the following characteristics. The farther left or right you go from the boundary, the larger the error becomes.

Optimization: Lagrange primal function

- objective

$$\min\left[\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\zeta_i\right]$$

- constraints

$$1-\zeta_i - y^{(i)}(\vec{w}\cdot\vec{x}^{(i)}+b)\le 0 \qquad y^{(i)} = \begin{cases} 1 & for \quad "+" \\ -1 & for \quad "-" \end{cases} \qquad -\zeta\le 0$$

- inequality constrained optimization problem

$$\min_x f(x), \quad s.t \quad h(x)\le 0$$

- Lagrangian primal function

$$L_p(x,\lambda) = f(x)+\lambda h(x) \;\rightarrow\; \lambda\ge 0$$

$$\min_x L_p(x,\lambda), \quad s.t \quad h(x)\le 0$$

$$\frac{\partial L_p}{\partial x} = 0$$

$$L_p = \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\zeta_i + \sum_{i=1}^{N}\lambda_i\{1-\zeta - y^{(i)}(\vec{w}\cdot\vec{x}^{(i)}+b)\} + \sum_{i=1}^{N}\{-\mu_i\zeta_i\} \qquad (\zeta\ge 0,\ \underline{\lambda_i\ge 0},\ \underline{\mu_i\ge 0})$$

$$L_p = \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\zeta_i - \sum_{i=1}^{N}\lambda_i\{y^{(i)}(\vec{w}\cdot\vec{x}^{(i)}+b) - 1 + \zeta\} - \sum_{i=1}^{N}\mu_i\zeta_i$$

$$\frac{\partial L_p}{\partial \vec{w}} = 0 \;\rightarrow\; \underline{\vec{w} = \sum_{i=1}^{N}\lambda_i y^{(i)}\vec{x}^{(i)}} \qquad \frac{\partial L_p}{\partial b} = 0 \;\rightarrow\; \underline{\sum_{i=1}^{N}\lambda_i y^{(i)} = 0}$$

$$\frac{\partial L_p}{\partial \zeta_i} = \underline{C-\lambda_i-\mu_i = 0 \;\rightarrow C = \lambda_i+\mu_i} \qquad \underline{\lambda_i = C-\mu_i} \;\rightarrow\; \underline{0\le\lambda_i\le C}$$

Optimization: Lagrange dual function

- The Lagrange dual function for the soft margin is obtained in the same way as for the hard margin.
- The dual function is the same as that of the hard margin, but has different constraints.

Primal function

$$L_p = \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\zeta_i - \sum_{i=1}^{N}\lambda_i\{y^{(i)}(\vec{w}\cdot\vec{x}^{(i)}+b) - 1 + \zeta\} - \sum_{i=1}^{N}\mu_i\zeta_i \qquad (\zeta\ge 0,\ \lambda_i\ge 0,\ \mu_i\ge 0)$$

$$\vec{w} = \sum_{i=1}^{N}\lambda_i y^{(i)}\vec{x}^{(i)} \qquad \sum_{i=1}^{N}\lambda_i y^{(i)} = 0 \qquad C-\lambda_i-\mu_i = 0 \;\rightarrow C = \lambda_i+\mu_i \;\rightarrow\; 0\le\lambda_i\le C$$

$$L_D = \frac{1}{2}\left(\sum_{i=1}^{N}\lambda_i y^{(i)}\vec{x}^{(i)}\right)\left(\sum_{j=1}^{N}\lambda_j y^{(j)}\vec{x}^{(j)}\right) + C\sum_{i=1}^{N}\zeta_i - \sum_{i=1}^{N}\left[\lambda_i y^{(i)}\left(\sum_{i=1}^{N}\lambda_i y^{(i)}\vec{x}^{(i)}\right)\cdot\vec{x}^{(i)} + \lambda_i y^{(i)} b - \lambda_i + \lambda_i\zeta_i\right] - \sum_{i=1}^{N}\mu_i\zeta_i$$

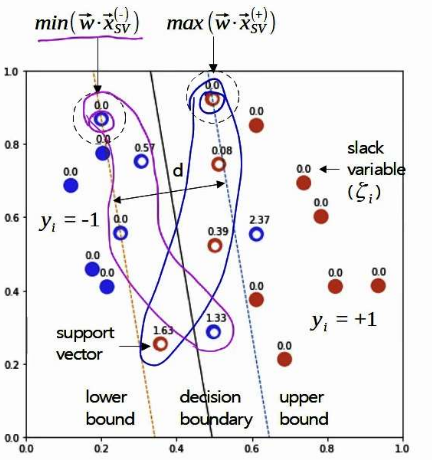
$$L_D = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y^{(i)}y^{(j)}\vec{x}^{(i)}\cdot\vec{x}^{(j)} + \sum_{i=1}^{N}\overset{=0}{\cancel{(C-\lambda_i-\mu_i)}}\zeta_i + \sum_{i=1}^{N}\lambda_i - \sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y^{(i)}y^{(j)}\vec{x}^{(i)}\cdot\vec{x}^{(j)} + b\overset{=0}{\cancel{\sum_{i=1}^{N}\lambda_i y^{(i)}}}$$

$$\boxed{\begin{array}{l} L_D = \sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y^{(i)}y^{(j)}\vec{x}^{(i)}\cdot\vec{x}^{(j)} \\ \text{constraints:} \quad \underline{0\le\lambda_i\le C}, \quad \sum_{i=1}^{N}\lambda_i y^{(i)} = 0 \end{array}}$$

- These are the Lagrange dual function and the constraints for the linear soft margin SVM.
- As in the hard margin SVM, the λ that maximizes this equation can be obtained through quadratic programming.

Decision function

- We can find the lambda from the dual function, and then use the lambda to find w. And we can use the support vectors to find b.
- We use the w and b to obtain the decision function. And we use the decision function to predict the class of the test data.



- Use λ to find w

$$\vec{w} = \sum_{i=1}^{N}\lambda_i y^{(i)}\vec{x}^{(i)}$$

- Use the support vectors (SV) to find b. SVs lie between the positive and negative boundaries.

$$\vec{w}\cdot\vec{x}_i^{(+)} + b\ge 1-\zeta_i \quad \longleftarrow \quad \text{"+" sample i satisfies this inequality.}$$

$$\vec{w}\cdot\vec{x}_j^{(-)} + b\le -1+\zeta_j \quad \longleftarrow \quad \text{"-" sample i satisfies this inequality.}$$

The SV(+) sample with the largest wx is on the positive boundary, and the SV(-) sample with the smallest wx is on the negative boundary (ξi = ξj = 0). Calculate b using these two samples.

$$b = -\frac{\left(max(\vec{w}\cdot\vec{x}_{SV}^{(+)}) + min(\vec{w}\cdot\vec{x}_{SV}^{(-)})\right)}{2}$$

- Decision function: We use this function to predict the class of the test data.

$$\hat{y} = w_1x_1 + w_2x_2 + \dots + b$$

Quadratic Programming for linear soft margin SVM

- It is the same as for hard margin SVM. However, only the constraints on λ are different.

$$L_D=\sum_{i=1}^{N}\lambda_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j \underline{y^{(i)}y^{(j)}\vec{x}^{(i)}\cdot\vec{x}^{(j)}}$$

$\underline{H_{i,j}}=y^{(i)}y^{(j)}\vec{x}^{(i)}\cdot\vec{x}^{(j)}$ ← definition

$$\underline{H}=\begin{bmatrix}y^{(1)}y^{(1)} & y^{(1)}y^{(2)}\\ y^{(2)}y^{(1)} & y^{(2)}y^{(2)}\end{bmatrix}\times\begin{bmatrix}\vec{x}^{(1)}\cdot\vec{x}^{(1)} & \vec{x}^{(1)}\cdot\vec{x}^{(2)}\\ \vec{x}^{(2)}\cdot\vec{x}^{(1)} & \vec{x}^{(2)}\cdot\vec{x}^{(2)}\end{bmatrix}$$

← For N=2 element wise product

H = np.outer(y, y) * np.dot(x, x.T) ← Python code

$$L_D=\sum_{i=1}^{N}\lambda_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j H_{i,j}$$

$$L_D=\sum_{i=1}^{N}\lambda_i-\frac{1}{2}\begin{bmatrix}\lambda_1 & \lambda_2\end{bmatrix}\cdot H\cdot\begin{bmatrix}\lambda_1\\ \lambda_2\end{bmatrix}$$

$$\underline{L_D}=\sum_{i=1}^{N}\lambda_i-\frac{1}{2}\lambda^T H\cdot\lambda$$

$$\text{s.t}\ -\lambda_i\le 0,\ \underline{\lambda_i\le C}$$

$$\sum_{i=1}^{N}\lambda_i y^{(i)}=0$$

← This constraint is added to the hard margin SVM.

- Standard form of QP

$$\underset{x}{argmin}\ \frac{1}{2}x^T P x+q^T x$$

$$\text{s.t}\ \ Gx\le h$$

$$Ax=b$$

- Lagrange dual function for linear soft margin SVM

$$\underset{\lambda}{argmin}\ \frac{1}{2}\lambda^T H\lambda-1^T\lambda_i$$

$$\text{s.t}\ -\lambda_i\le 0$$

$$\lambda_i\le C$$

$$\underline{\sum_{i=1}^{N}\lambda_i y^{(i)}=0}$$

← $\underline{\underset{\lambda}{argmax}\ L_D}$ → $\underline{\underset{\lambda}{argmin}\ (-L_D)}$

← Expressed in the same format as the standard form of QP.

$P\ :=\ H$ size = N x N

$q\ :=\ -\vec{1}$ = [[-1], [-1], …] - size = N x 1

$A\ :=\ y$ size = N x 1

$b\ :=\ 0$ scalar

$$G\cdot\lambda\le h\ \longrightarrow\ \underbrace{\begin{bmatrix}-1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&-1\\1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}}_{G}\cdot\begin{bmatrix}\lambda_1\\\lambda_2\\\lambda_3\\\lambda_4\end{bmatrix}=\begin{bmatrix}-\lambda_1\\-\lambda_2\\-\lambda_3\\-\lambda_4\\\lambda_1\\\lambda_2\\\lambda_3\\\lambda_4\end{bmatrix}\le\underbrace{\begin{bmatrix}0\\0\\0\\0\\C\\C\\C\\C\end{bmatrix}}_{h}\begin{matrix}\left.\begin{matrix}\\ \\ \\ \\ \end{matrix}\right\}-\lambda_i\le 0\\ \left.\begin{matrix}\\ \\ \\ \\ \end{matrix}\right\}\lambda_i\le C\end{matrix}$$

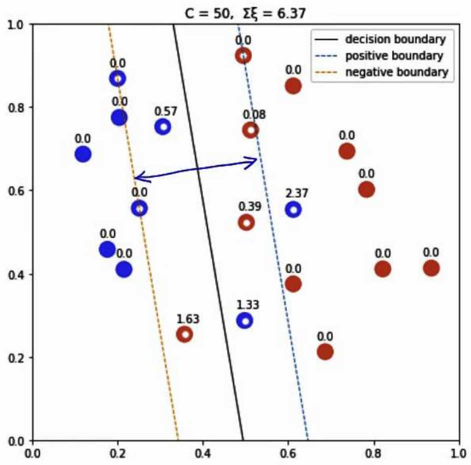
Since the constraints have changed, G and h are different from those of the hard margin SVM.

Observation of changes in decision boundary according to changes in C

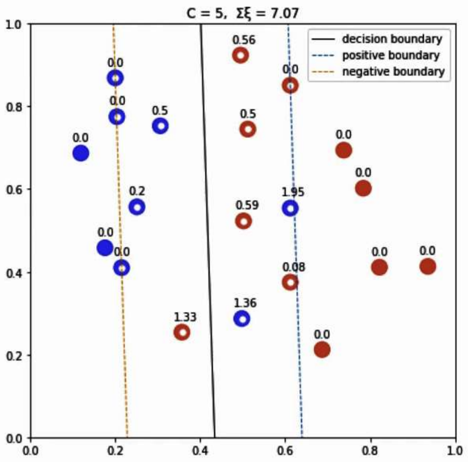
- As C gets smaller, the margin and ∑ξ get bigger. This is because it focuses more on the goal of maximizing the margin. Conversely, as C increases, it focuses more on the goal of minimizing ∑ξ, so ∑ξ decreases but the margin also becomes smaller.

$$min[\frac{1}{2}\|\vec{w}\|^2\ +\ C\sum_{i=1}^{N}\zeta_i]$$

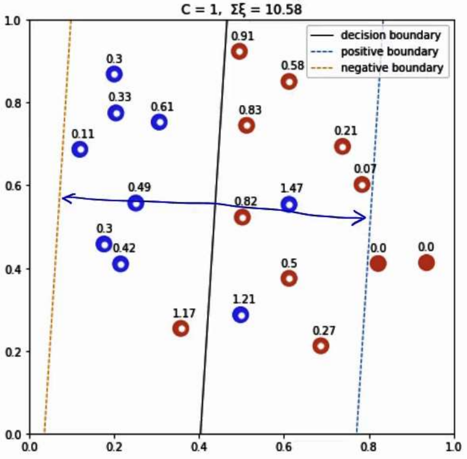
- C = 50, ∑ξ = 6.37



- C = 5, ∑ξ = 7.07



- C = 1, ∑ξ = 10.58


C = 1, Σξ = 10.58
decision boundary
positive boundary
negative boundary