Two-class

Perceptron learning rule for classification

Training set $\mathcal{L} = ((\underset{\sim}{x}_1, i_1), \ldots (\underset{\sim}{x}_n, i_n))$

$i_j \in \{1, 2\}$. $\forall j = (1)n$; $\underset{\sim}{x}_i \in \mathcal{X}$. $p \times 1$

Pop^{n}s π_1 & π_2.

A linear classifier

$$\underset{\sim}{\omega}' \underset{\sim}{x} + \omega_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \underset{\sim}{x} \in \begin{cases} \pi_1 \\ \pi_2 \end{cases}$$

$\underset{\sim}{v} = (\omega_0, \omega_1, \ldots \omega_p)'$ & $\underset{\sim}{z} = (1, x_1, \ldots x_p)'$.

or

$$\underset{\sim}{v}' \underset{\sim}{z} \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \underset{\sim}{x} \in \begin{cases} \pi_1 \\ \pi_2 \end{cases}$$

Note : $\underset{\sim}{z}$ can also be $= (1, \phi_1(\underset{\sim}{x}), \ldots \phi_D(\underset{\sim}{x}))'$.

Define

$$\underset{\sim}{y}_i' = (1, \underset{\sim}{x}_i'), \text{ if } \underset{\sim}{x}_i \in \pi_1$$

$$\& \quad \underset{\sim}{y}_i' = (-1, -\underset{\sim}{x}_i'), \text{ if } \underset{\sim}{x}_i \in \pi_2.$$

Ideally, we want a solution for $\underset{\sim}{v} \ni$

$\underset{\sim}{v}' \underset{\sim}{y} > 0$ for as many samples as possible.

If $\exists$ a $\underset{\sim}{v} \ni \underset{\sim}{v}' \underset{\sim}{y} > 0$ $\forall$ patterns then the data are said to be 'linearly separable.

$\frac{14}{4}$ -3 $\frac{21}{4}$ $\frac{7}{4}+2$

If $\underset{\sim}{v}' \underset{\sim}{y}_i < 0$ then a misclassification occur.

Define the perceptron criterion function as

$$J_p(\underset{\sim}{v}) = \sum_{\underset{\sim}{y}_i \in \mathcal{Y}} (-\underset{\sim}{v}' \underset{\sim}{y}_i)$$

$\mathcal{Y} \rightarrow$ set of all misclassified patterns

* $J_p(\underline{v})$ is prop to the sum of distances of the misclassified samples to the decision boundary.

Objective is to find. $v \ni J_p(\underset{\sim}{v})$ is minimised.
gradient-descent procedure used to solve $\min\limits_{\underset{\sim}{v}} J_p(\underline{v})$

$$\frac{\partial J_p(\underline{v})}{\partial \underset{\sim}{v}} = \sum_{\underset{\sim}{y}_i \in \mathcal{Y}} (-\underset{\sim}{y}_i)$$

Iteration steps

(i) $$v_{K+1} = v_K + \rho \sum_{\underset{\sim}{y}_i \in \mathcal{Y}} \underset{\sim}{y}_i$$

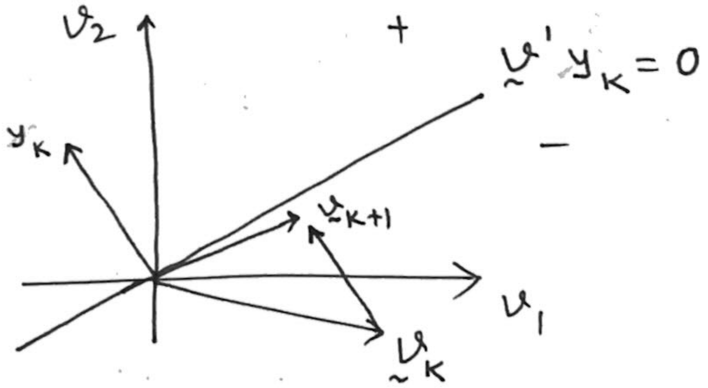
batch updation mode or many pattern adaption

(ii) $$v_{K+1} = v_K + \rho\, \underset{\sim}{y}_i$$ — ρ : step size, possible to have ρ_K also

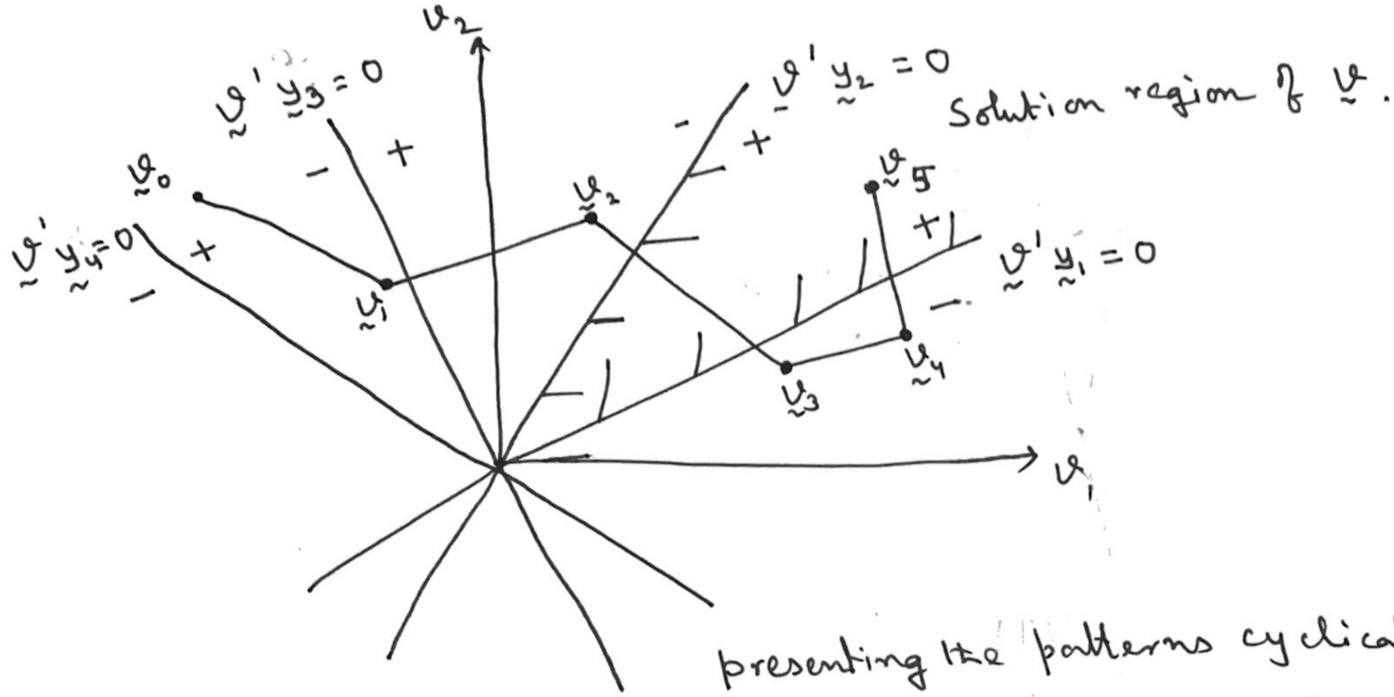
$\underset{\sim}{y}_i$ is a training pattern that is misclassified.

instantaneous updation mode or single-pattern adaption

weight updation in weight space



example with 4 patterns on single-pattern adaption mode



presenting the patterns cyclically

y_1, y_2, y_3, y_4.

Starting with v_0, first updation for y_2 takes v_0 to v_1; next updation for y_3 ($v_1 \to v_2$); next updation for y_2 ($v_2 \to v_3$) - - - - .

A solution of $J_p(v) = 0$. will be obtained for linearly separable patterns.

Note : An important variation of the above perceptron learning criterion is through introduction of a 'margin' $b > 0$.

A weight vector is updated whenever

$$v' y_i \le b$$

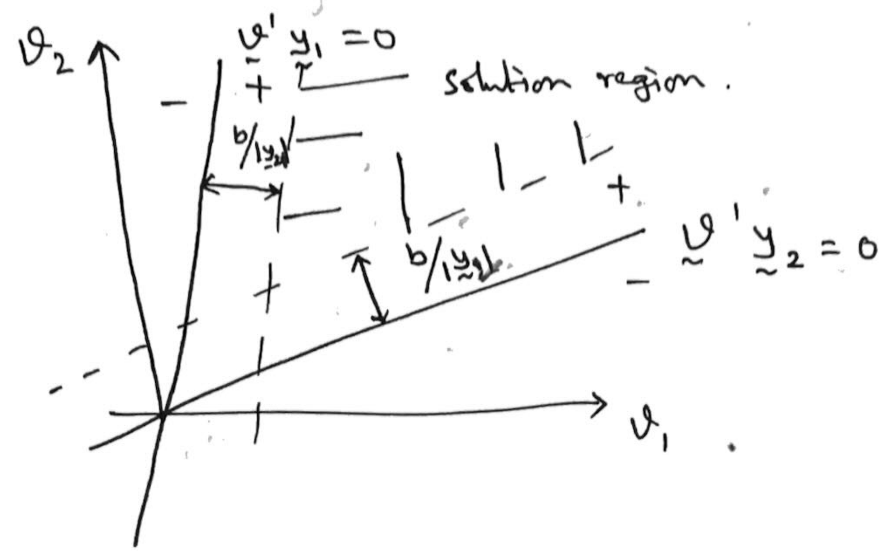
$\Rightarrow$ the solution vector v must lie at a distance greater than $b / |y_i|$ from hyperplane $v' y_i = 0$

(In term of org feature space $b/|w|$)

Aim of the margin is to improve generalization without the margin some points may lie too close to the separating boundary ie the separating hyperplane.

Maximal margin classifier $\rightarrow$ Support Vector Machine (SVM) classifier.

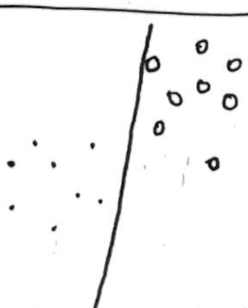
Solution region with a margin



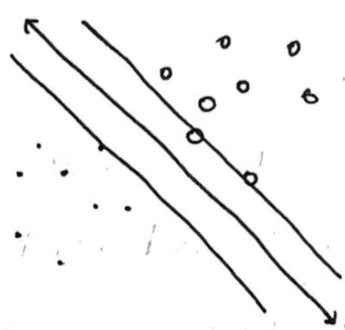
SVM

Construction of 'best' separating hyperplane, the maximal margin hyperplane

Basic SVM model.



A separating hyperplane for separable sets



maximal margin hyperplane

Suppose the data is linearly separable

Two classes π_1 & π_2 → labels $y_i = \pm 1$

$$\mathcal{L} = ((\underset{\sim}{x}_1, y_1), \ldots (\underset{\sim}{x}_n, y_n)).$$

linear discriminant f^n

$$g(\underset{\sim}{x}) = \underset{\sim}{\omega}'\underset{\sim}{x} + \omega_0$$

Decision rule

$$\underset{\sim}{w}'\underset{\sim}{x} + w_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \underset{\sim}{x} \in \begin{cases} \pi_1 & \text{with } y_i = +1 \\ \pi_2 & \text{with } y_i = -1 \end{cases}$$

$\Rightarrow$ Training pts are correctly classified if

$$y_i(\underset{\sim}{w}'\underset{\sim}{x}_i + w_0) > 0 \quad \forall i$$

Let $b > 0$ be a margin $\ni$ we seek a solution $\ni$

$$y_i(\underset{\sim}{w}'\underset{\sim}{x}_i + w_0) \geq b$$

Perceptron learning algorithm yeilds a solution for which all points $\underset{\sim}{x}_i$ are at a distance greater than or equal to $b/|\underset{\sim}{w}|$ from the separating hyperplane

<u>Note</u>: A scaling of b, w_0 and $\underset{\sim}{w}$ leaves $b/|\underset{\sim}{w}|$ unaltered and

$y_i(\underset{\sim}{w}'\underset{\sim}{x}_i + w_0) \geqslant b$ is still satisfied

w.l.o.g. we take $b = 1 \rightarrow$ cannonical hyperplanes

CH: $\quad H_1: \underset{\sim}{w}'\underset{\sim}{x} + w_0 = +1$

$\quad\quad H_2: \underset{\sim}{w}'\underset{\sim}{x} + w_0 = -1$

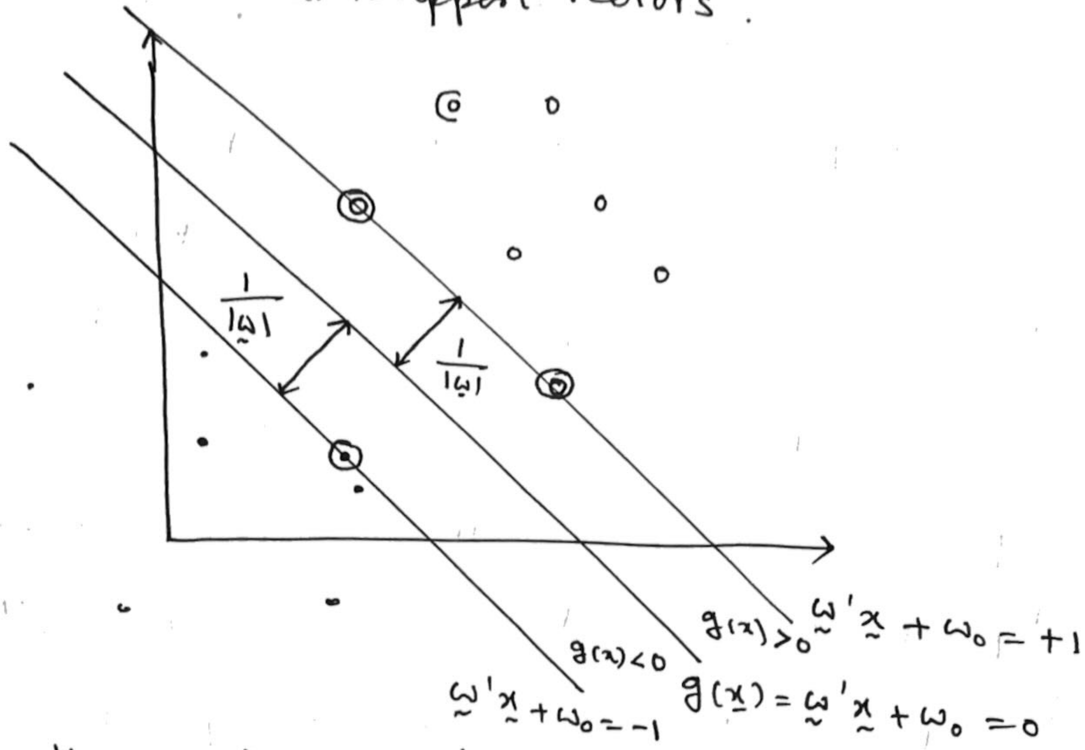
and

$$\underset{\sim}{w}'\underset{\sim}{x}_i + w_0 \geqslant +1 \quad \text{for } y_i = +1$$

$$\underset{\sim}{w}'\underset{\sim}{x}_i + w_0 \leq -1 \quad \text{for } y_i = -1$$

The distance of H_1 & H_2 from the separating hyperplane $g(\underset{\sim}{x}) = \underset{\sim}{w}'\underset{\sim}{x} + w_0 = 0$ is $1/|\underset{\sim}{w}|$ and is termed ~~the~~ the margin betn H_1/H_2 cannonical hyperplane and the sep hyperplane

The points that lie on the ~~separating~~ cannonical hyperplanes H_1 and H_2 are called 'support vectors'.



Objective

Maximizing the margin subject to the constraint that the separating hyperplane separates the linearly separable groups

i.e. Max $\frac{1}{|w|}$ i.e. Min $|\underset{\sim}{w}|$ $\ni$ the constraint

$$C : y_i \left(\underset{\sim}{w}' \underset{\sim}{x}_i + w_0 \right) \geq 1 \quad \forall\ i = 1(1)n$$

$$L_p = \frac{1}{2} \underset{\sim}{w}' w - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\underset{\sim}{w}' \underset{\sim}{x}_i + w_0 \right) - 1 \right).$$

where $\alpha_i \geq 0$; $i = 1(1)n$ are Lagrange multiplier.

$(p+1)$ Primal parameter $w_0, w_1, \ldots w_p$.

Note (i) solution obtained using quadratic programming

(ii) Case of linearly non-separable data solved using appropriate slack variables

(iii) Concepts can be extended for multiclass problem