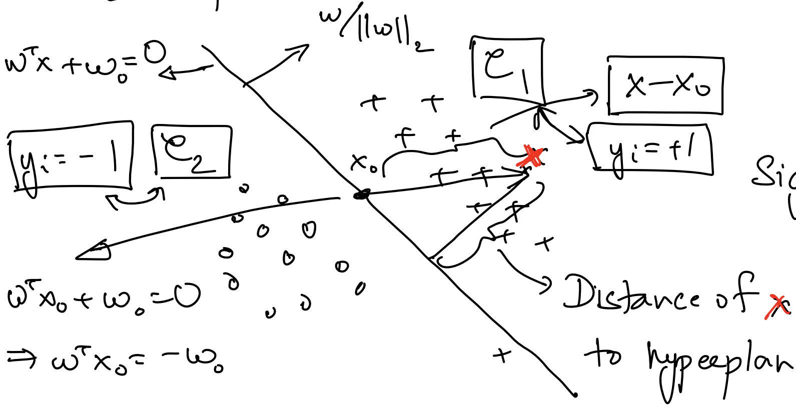
Classification

Perceptron – First proposed in 1962

(x_i, y_i), $i = 1, 2, \ldots N$



Signed Distance of x to the hyperplane

$$\left(\frac{w}{\|w\|}\right)^T (x - x_0)$$

$$= \frac{w^T x - w^T x_0}{\|w\|} = \frac{w^T x + w_0}{\|w\|}$$

For class C_1,

$w^T x_i + w_0 > 0$ for all points x_i in C_1 that are correctly classified

For class C_2, $w^T x_i + w_0 < 0$ for all points in C_2 that are correctly classified

For points that are correctly classified (x_i in C_1 or C_2)

$$y_i (w^T x_i + w_0) > 0$$

For misclassified points,

$$y_i (w^T x_i + w_0) < 0$$

Perceptron Criterion:

$$\min_{w, w_0} D(w, w_0) = \boxed{-\sum_{i \in M} y_i (w^T x_i + w_0)}$$

M indexes the misclassified points

$$\nabla_w D(w, w_0) = -\sum_{x \in M} y_i x_i = -\left(\sum_{i \in M \cap C_1} x_i - \sum_{j \in M \cap C_2} x_j\right)$$

$$\nabla_{w_0} D(w, w_0) = -\sum_{i \in M} y_i = -(N_1 - N_2)$$

N_i be the number of misclassified points in C_i

Perceptron Update Rule

$$\begin{bmatrix} w \\ w_0 \end{bmatrix} \leftarrow \begin{bmatrix} w \\ w_0 \end{bmatrix} - \eta \begin{bmatrix} -y_i x_i \\ -y_i \end{bmatrix} \quad \text{if } x_i \text{ is misclassified}$$

$$\hookrightarrow \begin{bmatrix} w \\ w_0 \end{bmatrix} \leftarrow \begin{bmatrix} w \\ w_0 \end{bmatrix} + \eta \begin{bmatrix} y_i x_i \\ y_i \end{bmatrix}$$

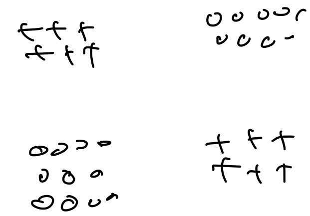
— Not gradient descent but stochastic gradient descent

Gradient Descent :

$$\begin{bmatrix} w \\ w_0 \end{bmatrix} = \begin{bmatrix} w \\ w_0 \end{bmatrix} + \eta \begin{bmatrix} \sum_i y_i x_i \\ \sum_i y_i \end{bmatrix}$$

<u>Stochastic Gradient Descent</u> : Look at x_i and if x_i is misclassified then go in direction of the negative gradient (but only contribution to gradient by x_i)

<u>Perceptron Algorithm</u> (x_i, y_i), $i = 1, 2, \ldots n$ is the training data



Start with some w, w_0

```
Repeat
    for i=1 to n
        if  y_i(w^T x_i + w_0) < 0  then
            w  <- w + η y_i x_i
            w_0 <- w_0 + η y_i
        endif
    end for
until there are no misclassifications (mistakes)
    within the for loop
```

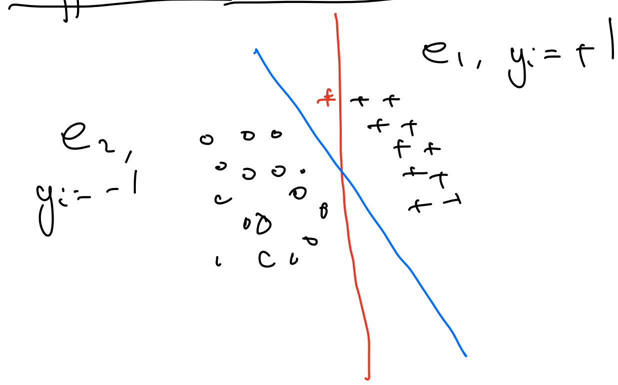
Perceptron algorithm is guaranteed to find <u>a</u> separating hyperplane <u>if</u> the data is linearly separable

<u>Drawbacks of Perceptron</u>

1. If the data is linearly separable, the hyperplane that is output by the perceptron depends on the order in which points (data) is presented to the algorithm.
2. Number of iterations might be large
3. If the classes are not linearly separable, then the

algorithm might not converge – cycles can develop that are not easy to detect

Support Vector Machines (SVM)



Signed distance of x to hyperplane

$$\frac{w^T x - w^T x_0}{\|w\|} = \frac{w^T x + w_0}{\|w\|}$$

$$y_i \left(\frac{w^T x + w_0}{\|w\|}\right)$$ – Distance of training point x_i to the hyperplane

Suppose we put the requirement that each of these distances is greater than C

$$y_i \frac{(w^T x + w_0)}{\|w\|} \geq C$$

$$\Rightarrow \quad y_i (w^T x + w_0) \geq C \cdot \|w\|$$

In linear support vector machines, the goal is to maximize C:

$$\max_{w, w_0} \; C$$

such that $y_i (w^T x_i + w_0) \geq C \|w\|$, $i = 1, 2, \ldots n$

Fix $C \|w\| = 1$ (since we can arbitrarily scale w and w_0)

$$\max_{w, w_0} \; 1/\|w\|_2$$

such that $y_i (w^T x_i + w_0) \geq 1$, $i = 1, 2, \ldots n$

Primal SVM Problem (Constrained Optimization Problem)

$$\boxed{\begin{array}{l} \min_{w, w_0} \; \|w\|_2^2 \\ \text{such that } y_i (w^T x_i + w_0) \geq 1, \quad i = 1, 2, \ldots n \end{array}}$$

$1/\|w\| \rightarrow$ margin

$2/\|w\|$

SVM finds hyperplane with maximum margin

General Constrained Optimization Problem

$$\boxed{\begin{aligned} &\min_x \ f_0(x) \\ &\text{st} \quad f_i(x) \le 0, \quad i = 1, 2, \ldots m \\ &\qquad h_i(x) = 0, \quad i = 1, 2, \ldots p \end{aligned}} \quad (1)$$

Primal problem

Here $x \in \mathbb{R}^n$ (i.e., $f_i : \mathbb{R}^n \to \mathbb{R}$, $h_i : \mathbb{R}^n \to \mathbb{R}$)

Lagrangian $L: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$

$$L(\underline{x}, \underline{\lambda}, \underline{\nu}) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

where λ_i is Lagrange multiplier for i-th inequality constraint

& ν_i is Lagrange multiplier for i-th equality constraint

$\underline{\lambda}$ and $\underline{\nu}$ are also called dual variables

Lagrange Dual Function

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu) = \inf_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right)$$

λ & ν are dual feasible if $\lambda \geq 0$ & $g(\lambda, \nu) > -\infty$

Fact 1: $g(\lambda, \nu) \le p^*$ for any dual feasible λ, ν

(where p^* is optimal value of the primal problem (1), i.e., $p^* = f_0(x^*)$)

Fact 2: If $\exists$ dual feasible λ^*, ν^* ($\lambda^* \geq 0$) and primal feasible x^* such that

$g(\lambda^*, \nu^*) = p^* = f_0(x^*)$ then strong duality is said to hold