Perceptron Algo Continued

Recall:

- linear decision function $f(x) = wx$ (no α for simplicity)
- decision boundary: $\{x : f(x) = 0\}$ (hyperplane through origin)
- sample points $x_1 \dots x_n \in \mathbb{R}^d$, class labels $y_1 \dots y_n \Rightarrow \pm 1$
- goal: find weights such that $y_i x_i w \geq 0$ (constraints are satisfied, all points classified properly)
- goal, revised: find w that minimizes $R(w) := \sum_{i \in V} -y_i x_i w$, V = set of i where $y_i x_i w < 0$

w is in a different space than our hyperplane "feature space"

"w space"

Objects in x-space (feature space) transform to objects in w-space

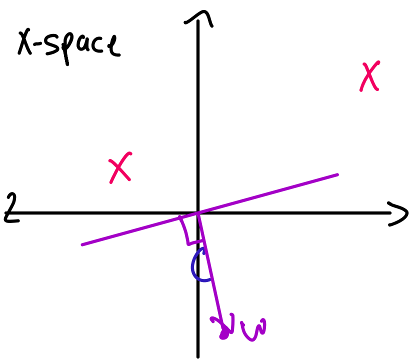
x-space	w-space
hyperplane: $\{z : w \cdot z = 0\}$	point: w (normal vector)
point: x	hyperplane: $\{z : x \cdot z = 0\}$

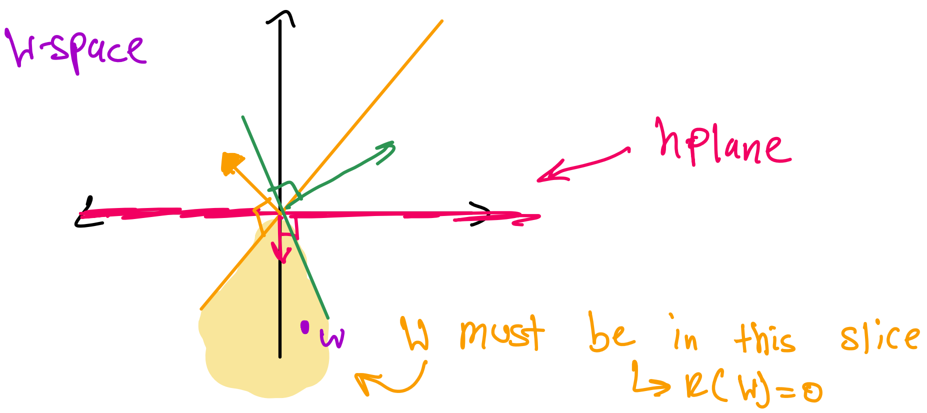
Consider point x on h-plane $\{z : wz = 0\} \iff wx = 0$

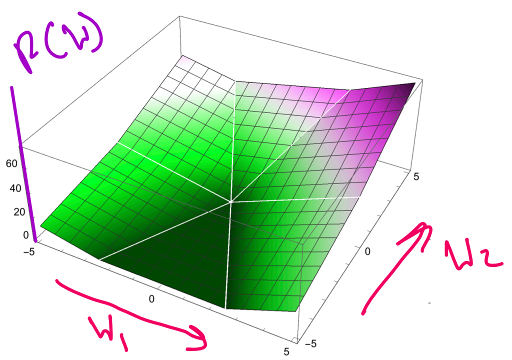
$\iff$ point w on h-plane $\{z : xz = 0\}$

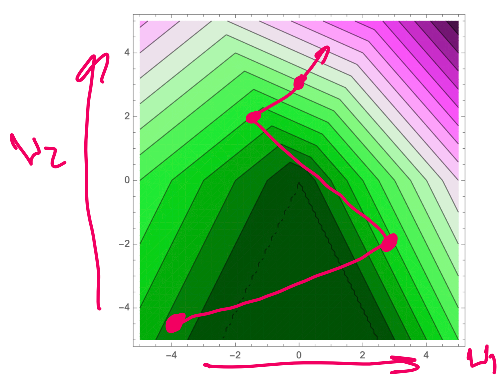
If we want $x_i w \geq 0 \rightarrow$ in x-space, x should be on same side as w if x is a positive sample

in w-space, w should be on same side of h-plane as x, x is the normal vector









no gradient @ $w = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Optimization Algorithm: gradient descent on R

- Given a starting point w, find gradient of R w.r.t. w (direction of steepest ascent)
- take step in opposite direction

Recall: $\nabla R(w) = \begin{bmatrix} \frac{\partial R}{\partial w_1} \\ \vdots \\ \frac{\partial R}{\partial w_d} \end{bmatrix}$ and $\nabla_w (z \cdot w) = \begin{bmatrix} z_1 \\ \vdots \\ z_d \end{bmatrix} = \vec{z}$

$\rightarrow \nabla R(w) = \sum_{i \in V} \nabla - y_i x_i w = \sum_{i \in V} - y_i x_i$

At any point w, walk downhill in direction of steepest descent ($-\nabla R(w)$)

doesn't work if not linearly separable

1) arbitrary $w \neq \vec{0}$ (good choice is any $y_i x_i$)
   - ↳ will classify at least 1 point correctly
2) while $R(w) > 0$

   V = set of indices i s.t. $y_i X_i < 0$

   $w = w + \epsilon \underbrace{\sum_{i \in V} y_i X_i}_{-\nabla R(w)}$
3) return w

Newton's method?
↳ not on piecewise linear function

$\epsilon > 0$ is the step size / learning rate

PROBLEM: slow algo. each step takes $O(nd)$ time

2nd optimization algo: stochastic gradient descent

Idea: in each step, pick one misclassified point X_i;

do gradient descent on $L(X_i \cdot w, y_i)$

Called the perceptron algorithm. each step takes $O(d)$ time

dim of sample points = # features

While some $y_i X_i \cdot w < 0$:

$w = w + \epsilon y_i X_i$

return w

* SGD can't always replace vanilla gradient descent

What if separating h-plane doesn't pass through origin?

↳ add a fictitious dimension

↳ $f(x) = w \cdot x + \alpha = \begin{bmatrix} w_1 & w_2 & \alpha \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$

now: Sample points are in $\mathbb{R}^{d+1}$, $x_{d+1} = 1$

→ run perceptron algo in $(d+1)$-dimensional space

online algo: If new data comes along while training, you can incorporate them

Perceptron Convergence thm.

$$O\left(\frac{r^2}{\gamma^2}\right) \qquad r = \max \|x_i\| \quad \text{"radius of data"}$$

$$\gamma = \text{max margin}$$

---

maximum margin classifiers

The margin of a linear classifier is the distance from the decision boundary to the nearest sample point

enforce some constraints: $\swarrow$ 0 in perceptron algo.

$$y_i [w \cdot x_i + \alpha] \geq 1 \quad \text{for } i \in [1 \dots n]$$

makes it impossible for $w = \vec{0}$

Recall: If $\|\vec{w}\| = 1$, then signed distance from h-plane to x_i is $w \cdot x_i + \alpha$

For $\|\vec{w}\| \neq 1$: Signed distance is $\frac{w}{\|\vec{w}\|} \cdot x_i + \frac{\alpha}{\|\vec{w}\|}$

$\rightarrow$ margin is $\min_i \frac{1}{\|\vec{w}\|} \cdot |\underbrace{w \cdot x_i + \alpha}_{\geq 1}| \geq \frac{1}{\|\vec{w}\|}$

$\rightarrow$ maximize margin $\frac{1}{\|\vec{w}\|}$ = minimizing $\|\vec{w}\|$ Subject to $y_i[w \cdot x_i + \alpha] \geq 1$

Optimization problem: find w & α that minimize $\|\vec{w}\|^2$ s.t. $y_i[x_i \cdot w + \alpha] \geq 1 \ \forall i \in [1 \dots n]$

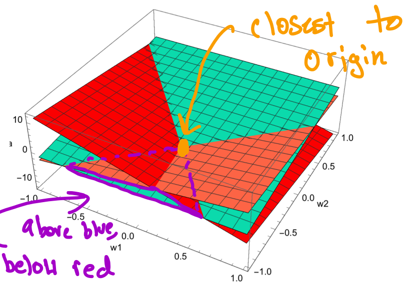
$\|w\|^2$ is smooth, even at origin

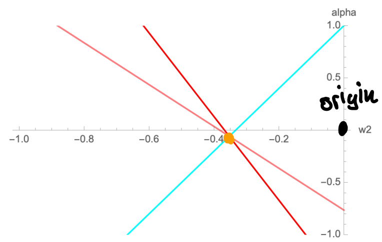
Called a quadratic program in $\mathbb{R}^{d+1}$ - (Since α brings 1 dimension)
- n constraints

One unique solution * if data is linearly seperable

Solution: maximum margin classifier / hard-margin SVM

- At the optimal solution, the margin is exactly $\frac{1}{\|\vec{w}\|}$ when $\|\vec{w}\|$ is maximized
- there is a slab w/ width $\frac{2}{\|\vec{w}\|}$ w/ no data points





$w_1 = 1/17$