Learning Linear Separators

Perceptron, Margins

Maria-Florina (Nina) Balcan

01/25/2019

Admin, Logistics

Hwk 0: due today.

Hwk 1: out today, due on February 8.

Learning Linear Seaprators

Today: Perceptron a simple learning algorithm for supervised classification analyzed via geometric margins in the 50's [Rosenblatt'57] .

Later in the course: Support Vector Machines (SVMs) an algorithm that also does well when data has large margin, one of the practically most effective classification algorithms in machine learning.

The Perceptron Algorithm

Originally introduced in the online learning scenario.

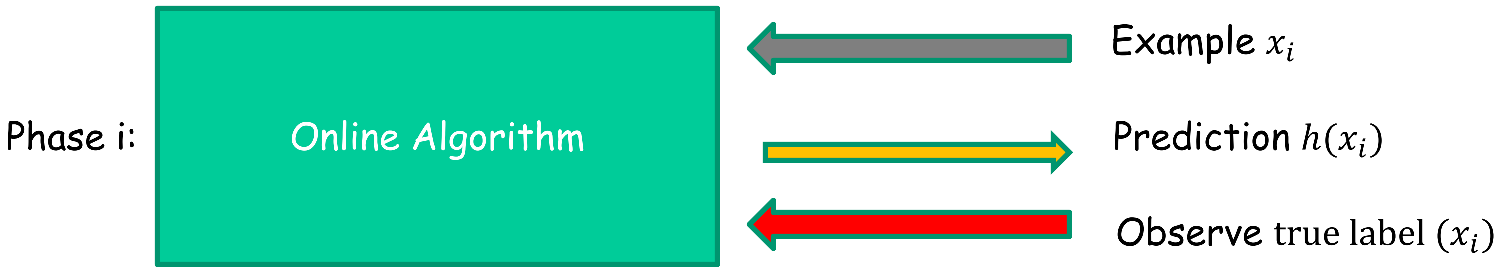
- Online learning model
- Perceptron algorithm
- Its guarantees under large margins

The Online Learning Model

- Example arrive sequentially.
- We need to make a prediction.

  Afterwards observe the outcome.

For i=1, 2, …, :



Phase i:

Online Algorithm

Example x_i

Prediction $h(x_i)$

Observe true label (x_i)

Mistake bound model

- **Goal**: Minimize the number of mistakes.
- Analysis wise, make no distributional assumptions, consider a worst sequence of examples.

The Online Learning Model. Motivation

Many possible applications

- Email classification (distribution of both spam and regular mail changes over time, but the target function stays fixed - last year's spam still looks like spam).

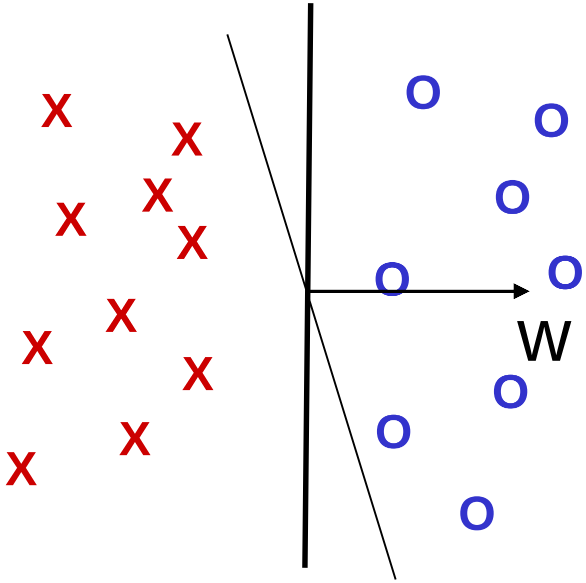
- Recommendation systems. Recommending movies, etc.

- Predicting whether a user will be interested in a new news article or not.

- Add placement in a new market.

Linear Separators

- Feature space $\mathrm{X} = \mathrm{R}^{\mathrm{d}}$
- Hypothesis class of linear decision surfaces in R^d.
  - $\mathrm{h(x) = w \cdot \ x \ + w_0}$, if $h(x) \geq \ 0$, then label **x** as **+**, otherwise label it as **-**



Trick: Without loss of generality $\mathrm{w_0 = 0}$.

Proof: Can simulate a non-zero threshold with a dummy input feature x_0 that is always set up to 1.

- $x = (x_1, \dots, x_d) \rightarrow \tilde{x} = (x_1, \dots, x_d, 1)$
- $\mathrm{w \cdot \ x \ + w_0} \geq 0$ iff $(w_1, \dots, w_d, \mathrm{w_0}\,) \cdot \tilde{x} \geq 0$

where $\mathrm{w} = (w_1, \dots, w_d)$

Linear Separators: Perceptron Algorithm

- Set t=1, start with the all zero vector w_1.
- Given example x, predict positive iff $w_t \cdot x \geq 0$
- On a mistake, update as follows:
  - Mistake on positive, then update $w_{t+1} \leftarrow w_t + x$
  - Mistake on negative, then update $w_{t+1} \leftarrow w_t - x$

Natural greedy procedure:

- If true label of x is +1 and w_t incorrect on x we have $w_t \cdot x < 0$, $w_{t+1} \cdot x \leftarrow w_t \cdot x + x \cdot x = w_t \cdot x + ||x||^2$, so more chance w_{t+1} classifies x correctly.
- Similarly for mistakes on negative examples.

Linear Separators: Perceptron Algorithm

- Set t=1, start with the all zero vector w_1.
- Given example x, predict positive iff $w_t \cdot x \geq 0$
- On a mistake, update as follows:
  - Mistake on positive, then update $w_{t+1} \leftarrow w_t + x$
  - Mistake on negative, then update $w_{t+1} \leftarrow w_t - x$

Note: w_t is weighted sum of incorrectly classified examples

$$w_t = a_{i_1} x_{i_1} + \cdots + a_{i_k} x_{i_k}$$

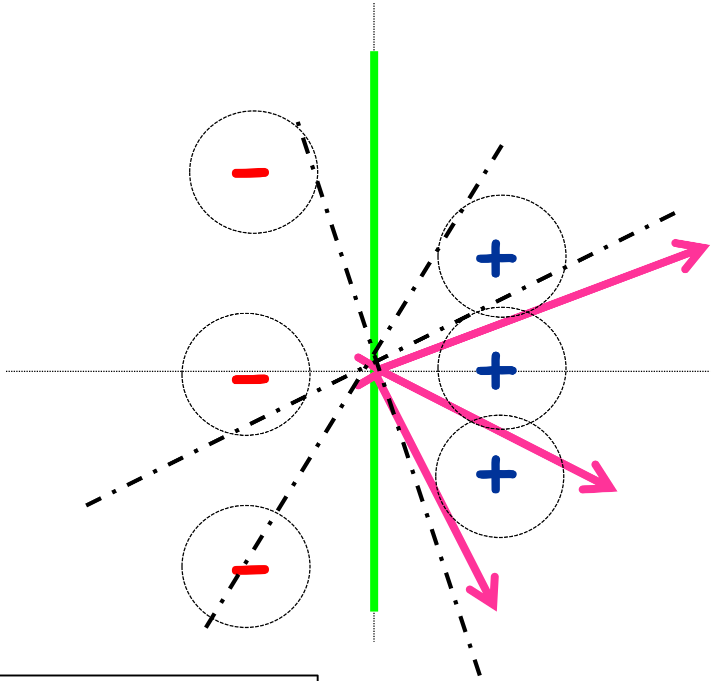
$$w_t \cdot x = a_{i_1} x_{i_1} \cdot x + \cdots + a_{i_k} x_{i_k} \cdot x$$

Important later in the course when we talk about kernels.

Perceptron Algorithm: Example

Example:

$(-1,2)$	$-$	✗
$(1,0)$	$+$	✓
$(1,1)$	$+$	✗
$(-1,0)$	$-$	✓
$(-1,-2)$	$-$	✗
$(1,-1)$	$+$	✓



$w_1 = (0,0)$

$w_2 = w_1 - (-1,2) = (1,-2)$

$w_3 = w_2 + (1,1) = (2,-1)$

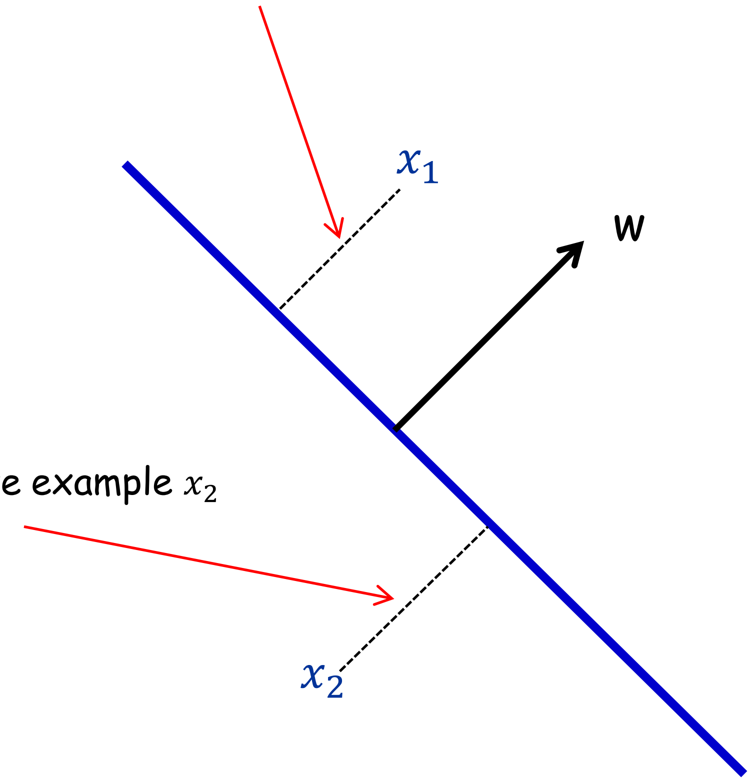
$w_4 = w_3 - (-1,-2) = (3,1)$

Algorithm:

- Set t=1, start with all-zeroes weight vector w_1.
- Given example x, predict positive iff $w_t \cdot x \geq 0$.
  - On a mistake, update as follows:
    - Mistake on positive, update $w_{t+1} \leftarrow w_t + x$
    - Mistake on negative, update $w_{t+1} \leftarrow w_t - x$

Geometric Margin

Definition: The margin of example x w.r.t. a linear sep. w is the distance from x to the plane $w \cdot x = 0$ (or the negative if on wrong side)



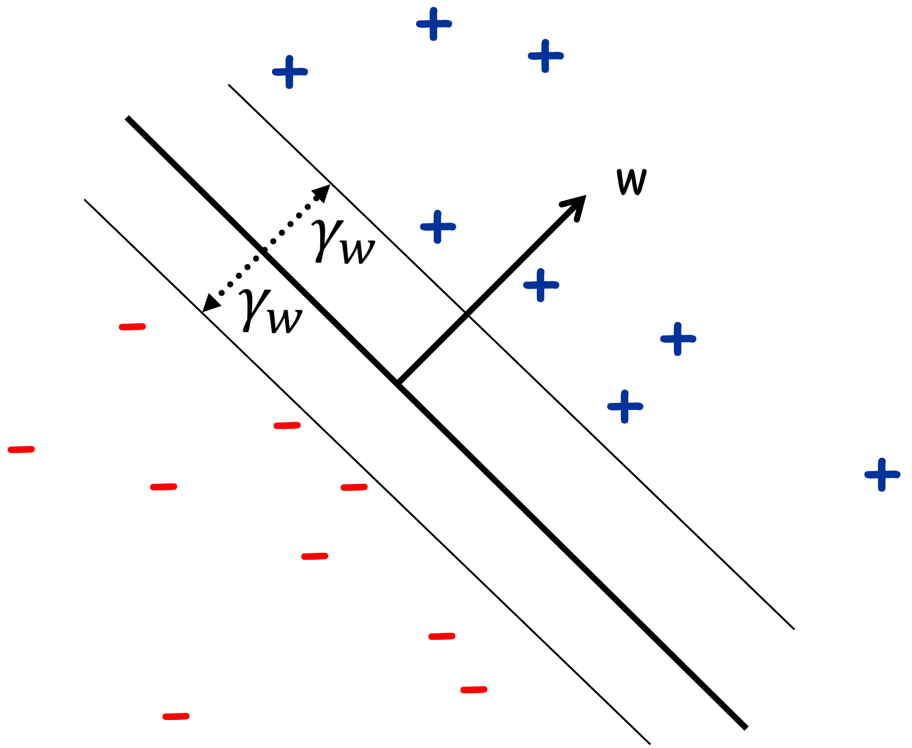
Margin of positive example x_1

Margin of negative example x_2

Geometric Margin

Definition: The margin of example x w.r.t. a linear sep. w is the distance from x to the plane $w \cdot x = 0$ (or the negative if on wrong side)

Definition: The margin γ_w of a set of examples S wrt a linear separator w is the smallest margin over points $x \in S$.

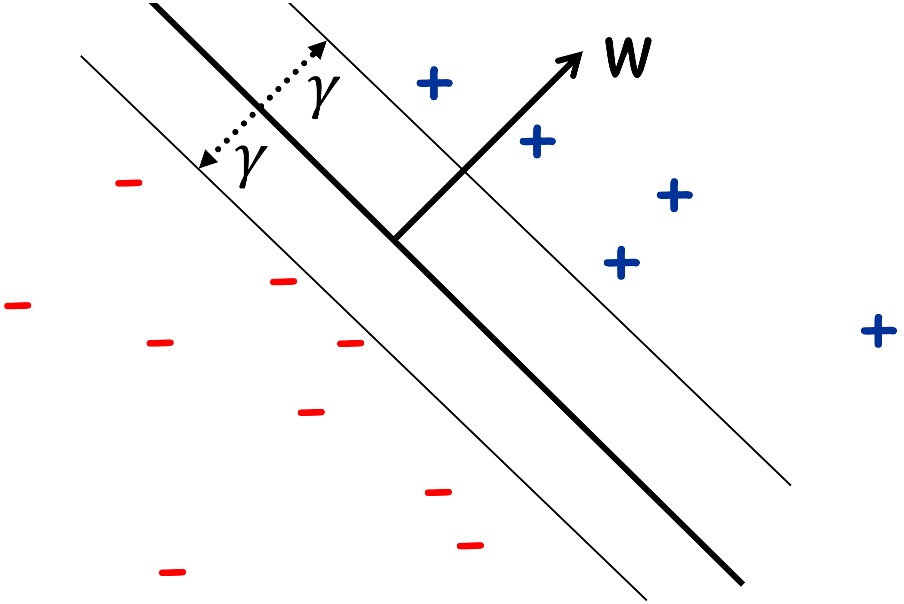


Geometric Margin

Definition: The margin of example x w.r.t. a linear sep. w is the distance from x to the plane $w \cdot x = 0$ (or the negative if on wrong side)

Definition: The margin γ_w of a set of examples S wrt a linear separator w is the smallest margin over points $x \in S$.

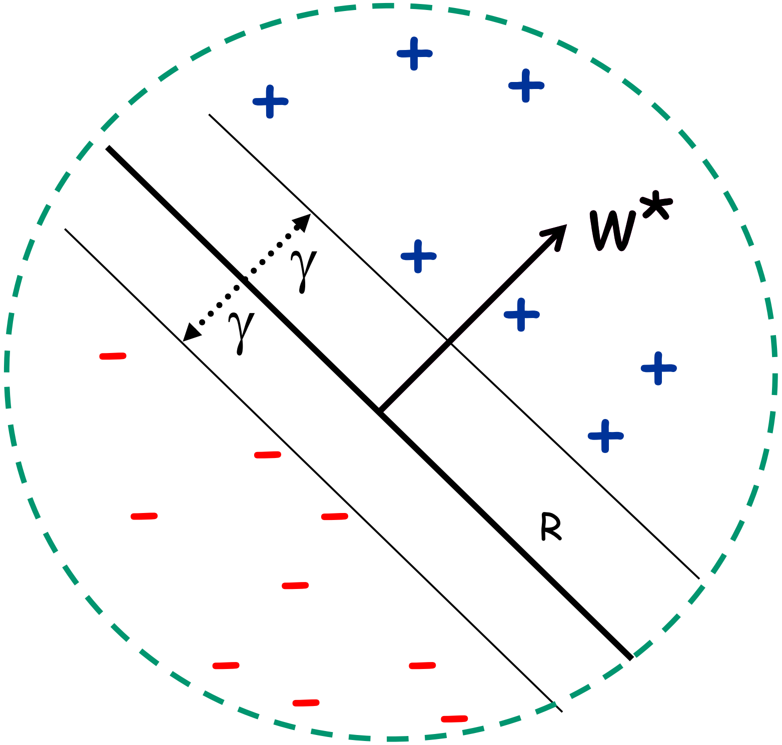
Definition: The margin γ of a set of examples S is the maximum γ_w over all linear separators w.



Perceptron: Mistake Bound

Guarantee: If data has margin γ and all points inside a ball of radius R, then Perceptron makes $\leq (R/\gamma)^2$ mistakes.

(Normalized margin: multiplying all points by 100, or dividing all points by 100, doesn't change the number of mistakes; algo is invariant to scaling.)



Perceptron Algorithm: Analysis

Guarantee: If data has margin γ and all points inside a ball of radius R, then Perceptron makes $\leq (R/\gamma)^2$ mistakes.

Update rule:

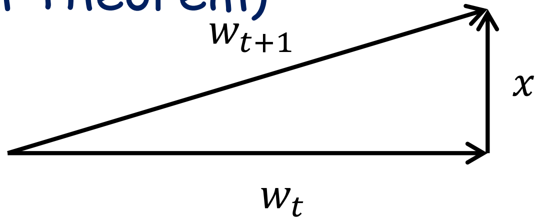
- Mistake on positive: $w_{t+1} \leftarrow w_t + x$
- Mistake on negative: $w_{t+1} \leftarrow w_t - x$

Proof:

Idea: analyze $w_t \cdot w^*$ and $\|w_t\|$, where w^* is the max-margin sep, $\|w^*\| = 1$.

Claim 1: $w_{t+1} \cdot w^* \geq w_t \cdot w^* + \gamma$. (because $l(x)x \cdot w^* \geq \gamma$)

Claim 2: $\|w_{t+1}\|^2 \leq \|w_t\|^2 + R^2$. (by Pythagorean Theorem)

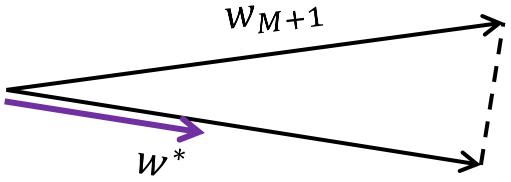


After M mistakes:

$w_{M+1} \cdot w^* \geq \gamma M$ (by Claim 1)

$\|w_{M+1}\| \leq R\sqrt{M}$ (by Claim 2)

$w_{M+1} \cdot w^* \leq \|w_{M+1}\|$ (since w^* is unit length)



So, $\gamma M \leq R\sqrt{M}$, so $M \leq \left(\frac{R}{\gamma}\right)^2$.

Perceptron Extensions

- Can use Perceptron in a batch setting too, to find a consistent linear separator given a set S of labeled examples that is linearly separable by margin γ.
  - We repeatedly feed the whole set S of labeled examples into the Perceptron algorithm up to $\left(\frac{R}{\gamma}\right)^2 + 1$ rounds, until we get to a point where the current hypothesis is consistent/correct with the whole set S. Note we are guaranteed to reach such a point.
  - The runnning time is then polynomial in $\frac{R}{\gamma^2}$ and |S|.

Perceptron Discussion

- Can also be adapted to the case where there is no perfect separator as long as the so called hinge loss (i.e., the total distance needed to move the points to classify them correctly large margin) is small.

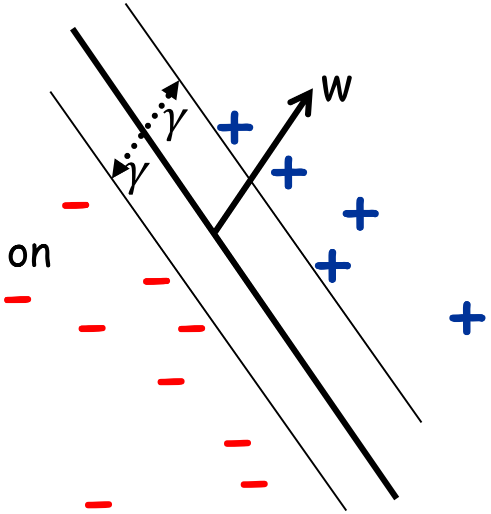
- Simple, but very useful in applications like branch prediction; it also has interesting extensions to structured prediction.

- Can be kernelized to handle non-linear decision boundaries!
    - See future lecture!!!

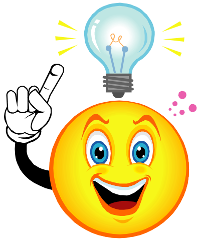
Margin Important Theme in ML

- If large margin, # mistakes Peceptron makes is small (independent on the dim of the ambient space)!

- Large margin can help prevent overfitting.
  - If large margin γ and if alg. produces a large margin classifier, then amount of data needed depends only on R/γ [Bartlett & Shawe-Taylor '99].



- Why not directly search for a large margin classifier?

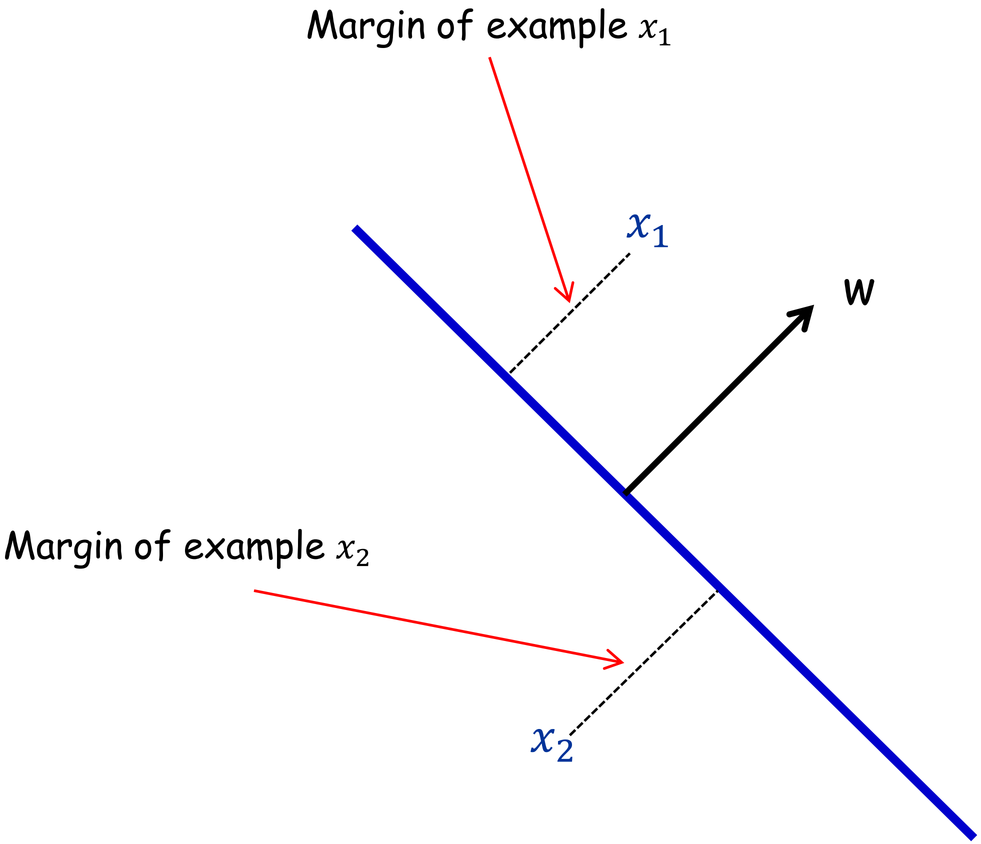


Support Vector Machines (SVMs).

Geometric Margin

WLOG homogeneous linear separators $[w_0 = 0]$.

Definition: The margin of example x w.r.t. a linear sep. w is the distance from x to the plane $w \cdot x = 0$.



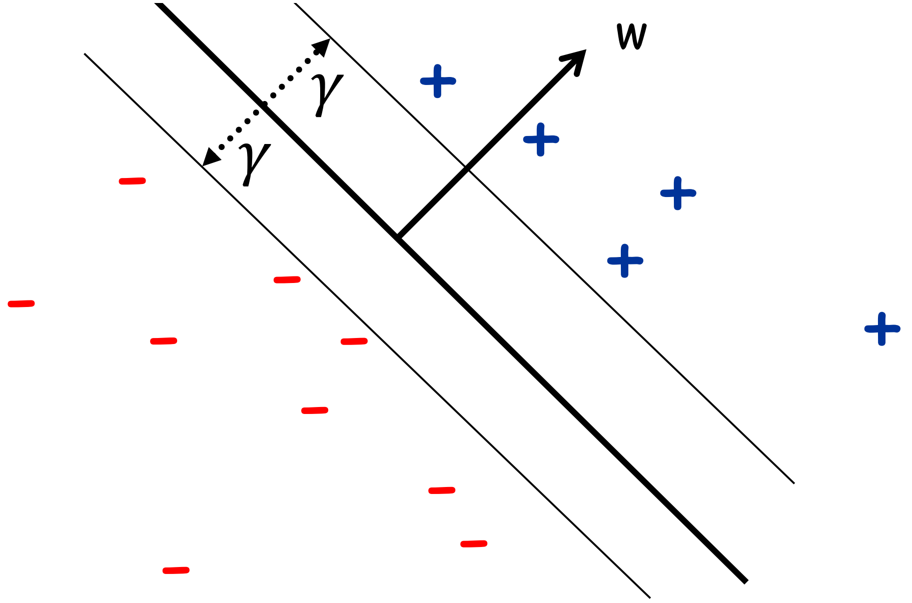
If $||w|| = 1$, margin of x w.r.t. w is $|x \cdot w|$.

Geometric Margin

Definition: The margin of example x w.r.t. a linear sep. w is the distance from x to the plane $w \cdot x = 0$.

Definition: The margin γ_w of a set of examples S wrt a linear separator w is the smallest margin over points $x \in S$.

Definition: The margin γ of a set of examples S is the maximum γ_w over all linear separators w.



Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs

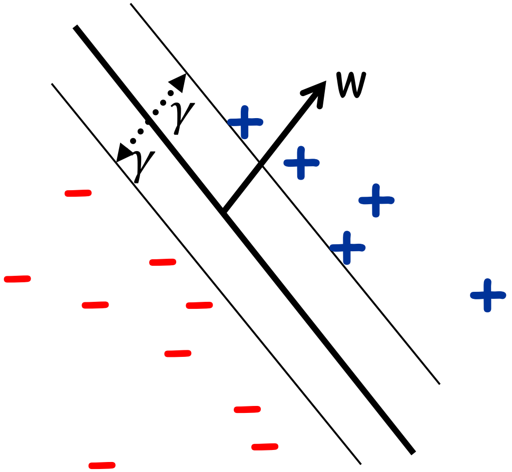
First, assume we know a lower bound on the margin γ

Input: γ, S={(x_1, y_1), …,$(\mathrm{x}_\mathrm{m}, y_\mathrm{m})$};

Find: some w where:

- $||\mathrm{w}||^2 = 1$
- For all i, $y_i w \cdot x_i \geq \gamma$

Output: w, a separator of margin γ over S



The case where the data is truly linearly separable by margin γ

Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs

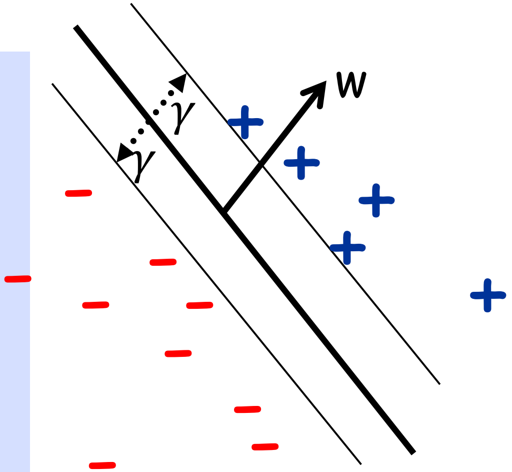
E.g., search for the best possible γ

Input: S={(x_1, y_1), …,$(\mathrm{x}_\mathrm{m}, y_\mathrm{m})$};

Find: some w and maximum γ where:

- $||\mathrm{w}||^2 = 1$
- For all i, $y_i w \cdot x_i \geq \gamma$

Output: maximum margin separator over S



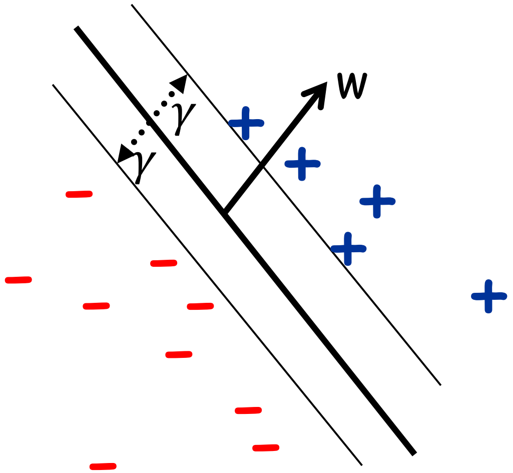
Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs

Input: $S=\{(x_1, y_1), \ldots, (x_m, y_m)\}$;

Maximize γ under the constraint:

- $||w||^2 = 1$
- For all i, $y_i w \cdot x_i \geq \gamma$



Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs

Input: $S=\{(x_1, y_1), \ldots, (x_m, y_m)\}$;

Maximize γ under the constraint:

- $||w||^2 = 1$
- For all i, $y_i w \cdot x_i \geq \gamma$

objective function

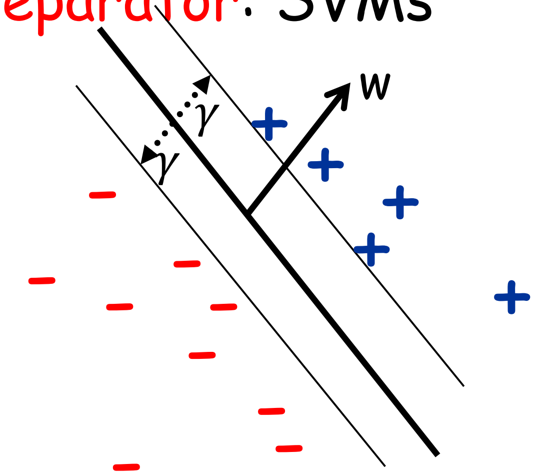
constraints

This is a **constrained optimization** problem.

- Famous example of constrained optimization: **linear programming**, where objective fn is linear, constraints are linear (in)equalities

Support Vector Machines (SVMs)

Directly optimize for the maximum margin separator: SVMs



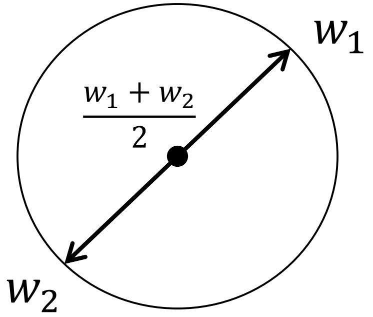
Input: S={(x_1, y_1), …,(x_m, y_m)};

Maximize γ under the constraint:

- $||w||^2 = 1$
- For all i, $y_i w \cdot x_i \geq \gamma$

This constraint is non-linear.

In fact, it's even non-convex



Can re-write it in a convex way, also make it handle non-separable data. Will discuss it later in the course.

What you should know

- Perceptron simple online algo for learning linear separators with good guarantees when data has large geometric margin.
- The importance of margins in machine learning.