Do not distribute course material

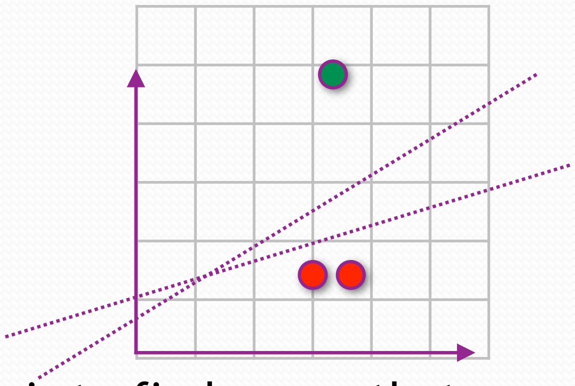
Lecture
Support Vector Machines continued

PROF. LINDA SELLIE

SOME SLIDES FROM PROF. RANGAN

We can compare two hyperplanes by comparing their geometric margins.

Which hyperplane has a larger margin:



- Given the following data:

```
x(1) = [3.2   4.7]        y(1) = -1
x(2) = [3.5   1.4 ]       y(2) = 1
x(3) = [3.    1.4 ]       y(3) = 1
```

- What is the geometric margin for:

Goal is to find $\mathbf{w}, w_0$ that has the largest γ such that $y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0)/\|\mathbf{w}\|_2 \geq \gamma$

$$w_0 = 1/2 \qquad \mathbf{w} = \begin{bmatrix} 2/3 \\ -1 \end{bmatrix}$$

$$\overbrace{(-1)\left([2/3 \quad -1]\begin{bmatrix} 3.2 \\ 4.7 \end{bmatrix} + 1/2)\right)/\sqrt{4/9+1}}^{y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0)/\|\mathbf{w}\|_2} \quad = 1.7$$

$$(1)\left([2/3 \quad -1]\begin{bmatrix} 3.5 \\ 1.4 \end{bmatrix} + 1/2)\right)/\sqrt{4/9+1} \quad = 1.2$$

$$(1)\left([2/3 \quad -1]\begin{bmatrix} 3 \\ 1.4 \end{bmatrix} + 1/2)\right)/\sqrt{4/9+1} \quad = 0.9$$

$\gamma = 0.9$

$$y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0)/\|\mathbf{w}\|_2 \geq 0.9$$

$$w_0 = 1 \qquad \mathbf{w} = \begin{bmatrix} 1/3 \\ -1 \end{bmatrix}$$

$$(-1)\left([1/3 \quad -1]\begin{bmatrix} 3.2 \\ 4.7 \end{bmatrix} + 1)\right)/\sqrt{4/9+1} \quad = 2.2$$

$$(1)\left([1/3 \quad -1]\begin{bmatrix} 3.5 \\ 1.4 \end{bmatrix} + 1)\right)/\sqrt{4/9+1} \quad = 0.6$$

$$(1)\left([1/3 \quad -1]\begin{bmatrix} 3 \\ 1.4 \end{bmatrix} + 1)\right)/\sqrt{4/9+1} \quad = 0.5$$

$\gamma = 0.5$

$$y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0)/\|\mathbf{w}\|_2 \geq 0.5$$

Objective function

Goal

$$\max_{w_0, \boldsymbol{w}} \ \gamma$$

subject to $y^{(i)}(w_0 + \mathbf{w}^T \boldsymbol{x}^{(i)}) \geq \gamma$ for all i=1,...,N

$$\|\mathbf{w}\|_2 = 1$$

Not yet the form we need

Difficult to work with constraints that are not linear.

Let us write our objective function in a different way.

We want our constrained object function to return a unique $\mathbf{w}, w_0$

Crazy idea next!

Instead of requiring $\mathbf{w}$ to be a unit vector (i.e. $\|\mathbf{w}\| = 1$),
We will define the idea of a "functional margin" and require that to be 1.

Steps to understanding "functional margin":

1. Simple observation: *rescaling* the parameters doesn't change the *decision boundary*.
2. How to rescale the parameters/weights, so the functional margin is 1. We call these weights/ parameters *canonical weights*.

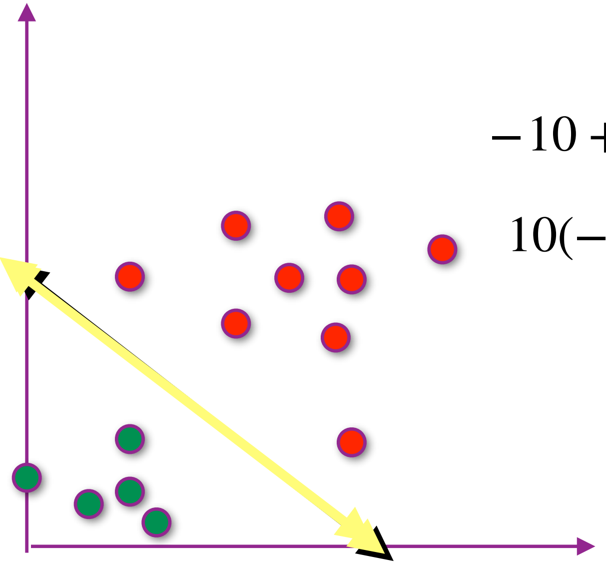
Step 1. We can write a hyperplane in many ways

$-10 + 3x_1 + 4x_2 > 0$

$10(-10 + 3x_1 + 4x_2) > 0$

$-10 + 3x_1 + 4x_2 = 0$

$-100 + 30x_1 + 40x_2 = 10(-10 + 3x_1 + 4x_2) = 0$



$-10 + 3x_1 + 4x_2 < 0$

$10(-10 + 3x_1 + 4x_2) < 0$

Pair share: Do we change the classification if we multiply $-10 + 3x_1 + 4x_2 = 0$ by 10?

Step 1 conclusion

Rescaling the parameters doesn't change the line (decision boundary)!

$$\mathbf{w}^T\mathbf{x} + w_0 = 0 = c\mathbf{w}^T\mathbf{x} + cw_0$$

Step 2

What is another way to constrain the problem so that we get a unique solution?

$y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) = 1$ for the closest point to the hyperplane

The functional margin of $(\mathbf{w}, w_0)$ with respect to ***a point*** $\mathbf{x}^{(i)}$ is

$$\gamma^{(i)} = y^{(i)}(\mathbf{w}^T\boldsymbol{x}^{(i)} + w_0)$$

The *Functional margin* of $(\mathbf{w}, w_0)$ with respect to ***a set*** S is

$$\gamma = \min\{\gamma^{(1)}, \gamma^{(2)}, \ldots, \gamma^{(N)}\}$$

Step 2

Canonical weights

● +1
● -1

Functional margin of $(\mathbf{w}, w_0)$ with respect to ***a point*** $\mathbf{x}^{(i)}$ is

$$\gamma^{(i)} = y^{(i)}\left(\frac{\mathbf{w}^T}{3}\boldsymbol{x}^{(i)} + \frac{w_0}{3}\right)$$

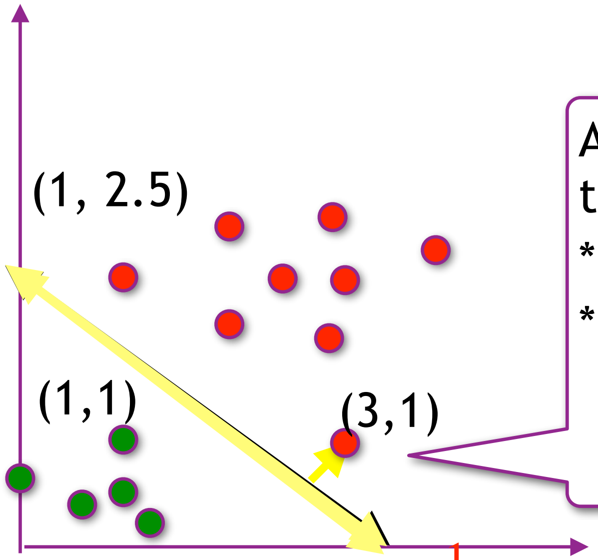
Functional margin of $(\mathbf{w}, w_0)$ with respect to ***a set*** S is

$$\gamma = \min\{\gamma^{(1)}, \gamma^{(2)}, \ldots, \gamma^{(N)}\}$$

$$\gamma^{(2)} = (1)\left(\left[\tfrac{3}{3}\ \ \tfrac{4}{3}\right]\begin{bmatrix}3\\1\end{bmatrix} - \frac{10}{3}\right) = (1)\frac{(3)}{3}$$

$$\gamma^{(1)} = (1)\left(\left[\tfrac{3}{3}\ \ \tfrac{4}{3}\right]\begin{bmatrix}1\\2.5\end{bmatrix} - \frac{10}{3}\right) = (1)\frac{(3)}{3}$$

$$\gamma^{(N)} = (-1)\left(\left[\tfrac{3}{3}\ \ \tfrac{4}{3}\right]\begin{bmatrix}1\\1\end{bmatrix} - \frac{10}{3}\right) = (-1)\frac{(-3)}{3}$$



$$\frac{1}{3}(-10 + 3x_1 + 4x_2) = 0$$

After scaling, the points closest to the decision boundary have:

* functional margin of 1
* Euclidean distance for this point is $\dfrac{1}{\|\mathbf{w}\|_2}$

$$-10 + 3x_1 + 4x_2 = 0 = -10/3 + 3/3x_1 + 4/3x_2$$

We can make γ = 1
The canonical weights

Step 2 conclusion

For any hyperplane that separates the data, we can make its functional margin any value we want.

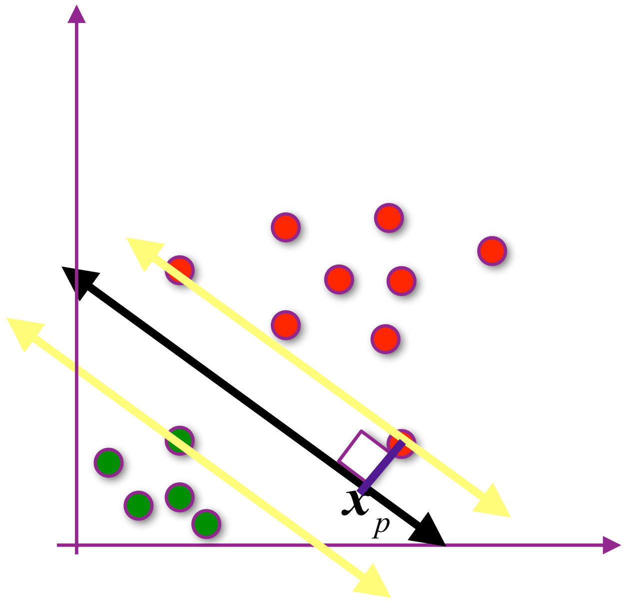
Canonical weights are when the functional margin is 1 for the set of training examples

$$\min_i y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) = 1$$

Example: given a hyperplane $w = (3,4)^T$, $w_0 = -10$ which has a functional margin of 3, rescale the parameters so the functional margin is 1

Next: Many equivalent versions of our objective function

Hard-Margin SVM



1) Constrained optimization problem:

$$\max_{w_0, \boldsymbol{w}} \ \gamma$$

$$\text{subject to } \mathrm{y}^{(i)}(w_0 + \mathbf{w}^T \boldsymbol{x}^{(i)}) \geq \gamma \ \text{ for all i=1,...,N}$$

$$\|\mathbf{w}\|_2 = 1$$

$\frac{\gamma}{\|\mathbf{w}\|_2}$ = Geometric margin

2) Another formulation:

$$\max_{w_0, \boldsymbol{w}} \frac{\gamma}{\|\mathbf{w}\|_2} \quad = r$$

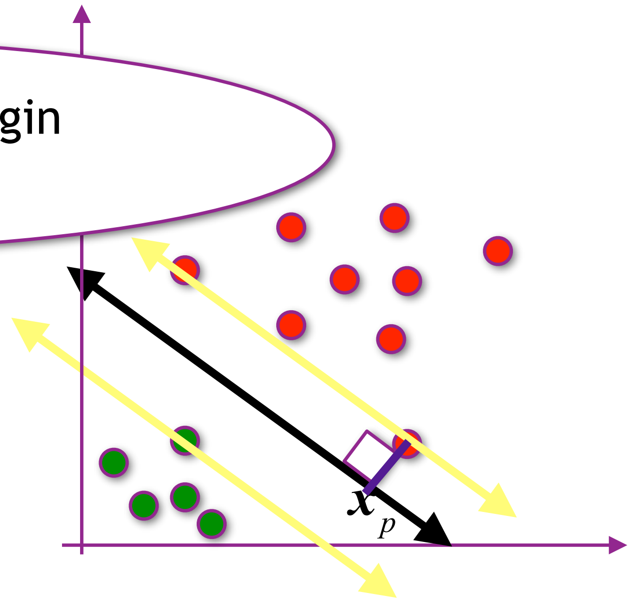
$$\text{subject to } \mathrm{y}^{(i)}(w_0 + \mathbf{w}^T \boldsymbol{x}^{(i)}) \geq \gamma \ \text{ for all i=1,...,N}$$

Hard-Margin SVM

$$\frac{\gamma}{\|\mathbf{w}\|_2} = \text{Geometric margin}$$

2) $$\max_{w_0, \boldsymbol{w}} \frac{\gamma}{\|\mathbf{w}\|_2} = r$$

$$\text{subject to } \mathrm{y}^{(i)}(w_0 + \mathbf{w}^T \boldsymbol{x}^{(i)}) \geq \gamma \text{ for all i=1,...,N}$$



Canonical weights!!!

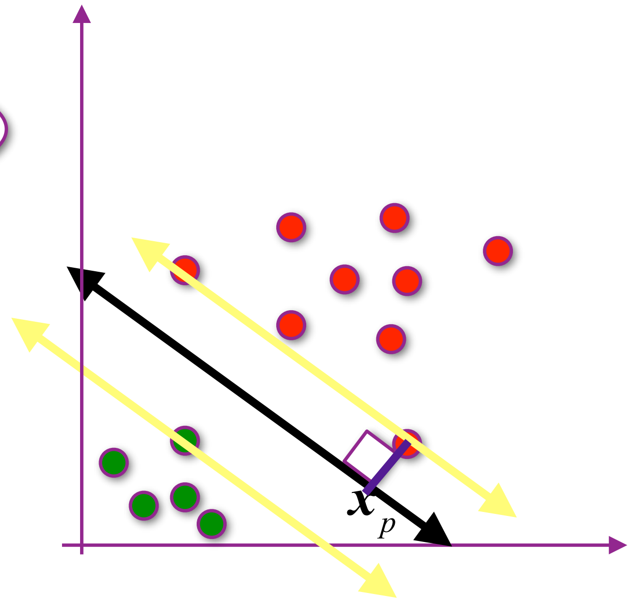
Idea: we can rescale our margin to anything we want by rescaling our coefficients

notice that $\max \gamma / \|\mathbf{w}\|_2$ equals $\max(\gamma/\gamma)/\|\mathbf{w}/\gamma\|_2$

$$\text{subject to } \mathrm{y}^{(i)}(w_0/\gamma + \mathbf{w}^T/\gamma \;\; \boldsymbol{x}^{(i)}) \geq \gamma/\gamma \text{ for all i=1,...,N}$$

Hard-Margin SVM

Canonical weights!!!



$\max \gamma / \|\mathbf{w}\|_2 \qquad \max(\gamma/\gamma)/\|\mathbf{w}/\gamma\|_2$

subject to $y^{(i)}(w_0/\gamma + \mathbf{w}^T/\gamma \ \boldsymbol{x}^{(i)}) \geq \gamma/\gamma$ for all i=1,...,N

3) We set $w_0 := w_0/\gamma$, and $\mathbf{w} := \mathbf{w}/\gamma$ Notice we now want to max $1/\|\mathbf{w}\|_2$

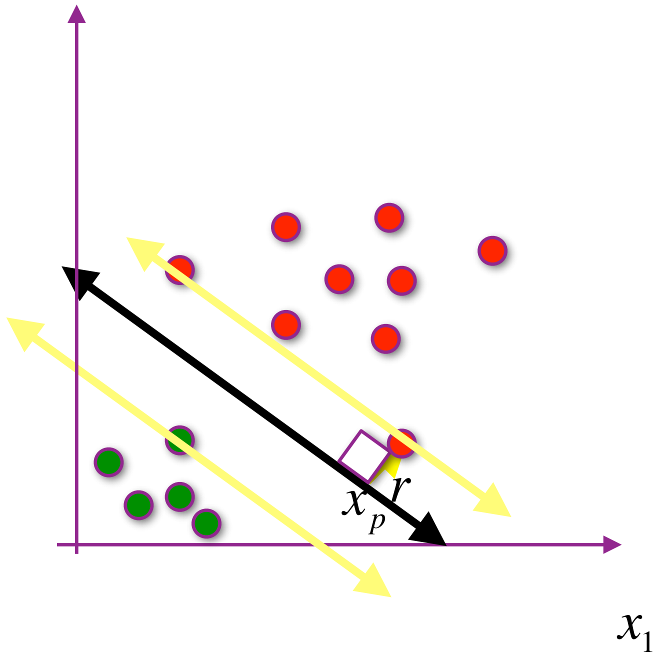
Using this idea we rewrite the formula as

$\max_{w_0,\mathbf{w}} 1/\|\mathbf{w}\|_2$ now $\gamma = 1$

$\dfrac{1}{\|\mathbf{w}\|_2}$ = margin

Subject to $y^{(i)}(w_0 + \mathbf{w}^T\mathbf{x}^{(i)}) \geq 1$ for all $i = 1,\ldots,N$

Hard-Margin SVM



3) Constrained optimization problem:

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

Subject to $y^{(i)}(w_0 + \mathbf{w}^T \mathbf{x}^{(i)}) \geq 1$ for all $i = 1, \ldots, N$

4) Notice $\max 1/\|\mathbf{w}\|_2$ is the same as $\min \|\mathbf{w}\|_2$

Notice $\min \|\mathbf{w}\|_2$ is the same as $\min \|\mathbf{w}\|_2^2$

$$\min \|\mathbf{w}\|_2^2 = \min(w_1^2 + w_2^2 + \cdots + w_d^2)$$

Subject to $y^{(i)}(w_0 + \mathbf{w}^T \mathbf{x}^{(i)}) \geq 1$ for all $i = 1, \ldots, N$

Solvable in polynomial time!

Objective function is convex and points satisfying constraints are convex

A constrained quadratic optimization problem!

Example Hard-Margin SVM

($\mathbf{x}^T$,y): ((1, 2.5),1), ((2, 2),1), ((3,1),1),…,((0, 0.75),-1), ((1,1),-1)}

The constrained quadratic optimization function is:

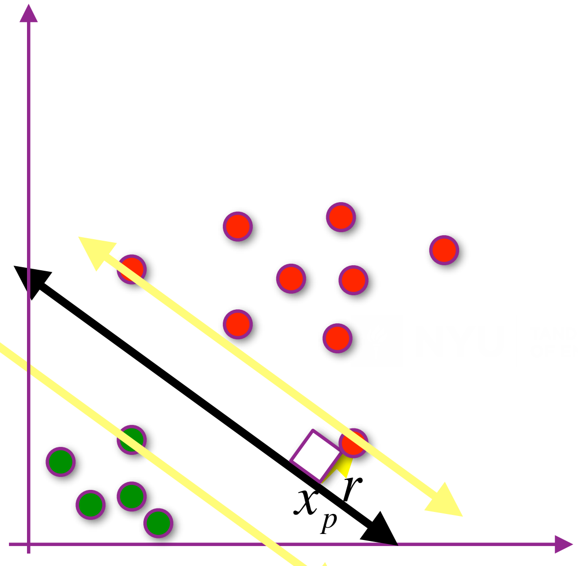
$$\min_{w_0,\boldsymbol{w}} \left\|\mathbf{w}\right\|_2^2 = w_1^2 + w_2^2$$

$$\text{subject to } (1)\left(w_0 + \mathbf{w}^T \begin{bmatrix} 1 \\ 2.5 \end{bmatrix} \right) \geq 1$$

$$(1)\left(w_0 + \mathbf{w}^T \begin{bmatrix} 2 \\ 2 \end{bmatrix} \right) \geq 1$$

$$\vdots$$

$$(-1)\left(w_0 + \mathbf{w}^T \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \geq 1$$



Example Hard-Margin SVM

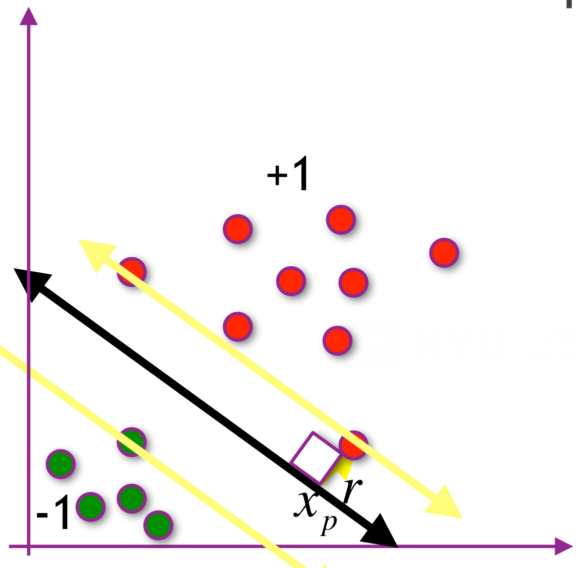
$(\mathbf{x}^T, y)$: ((1, 2.5),1), ((2, 2),1), ((3,1),1),…,((0, 0.75),-1), ((1,1),-1)}

The optimal hyperplane is: $\mathbf{w} = (1,4/3)^T$, $w_0 = -10/3$

- $f(\mathbf{x}) = (1,4/3)\mathbf{x} - 10/3$
- Predict +1 if $f(\mathbf{x}) > 0$
- Predict -1 if $f(\mathbf{x}) < 0$

Two types of training data:

- $y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) = 1.$ Points on the margin called support vectors
- $y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) > 1.$ If we remove these points, the solution doesn't change



Could it be possible for

$y^{(i)}(w_0 + \mathbf{w}^T \boldsymbol{x}^{(i)}) > 1$ for all i=1,…,N

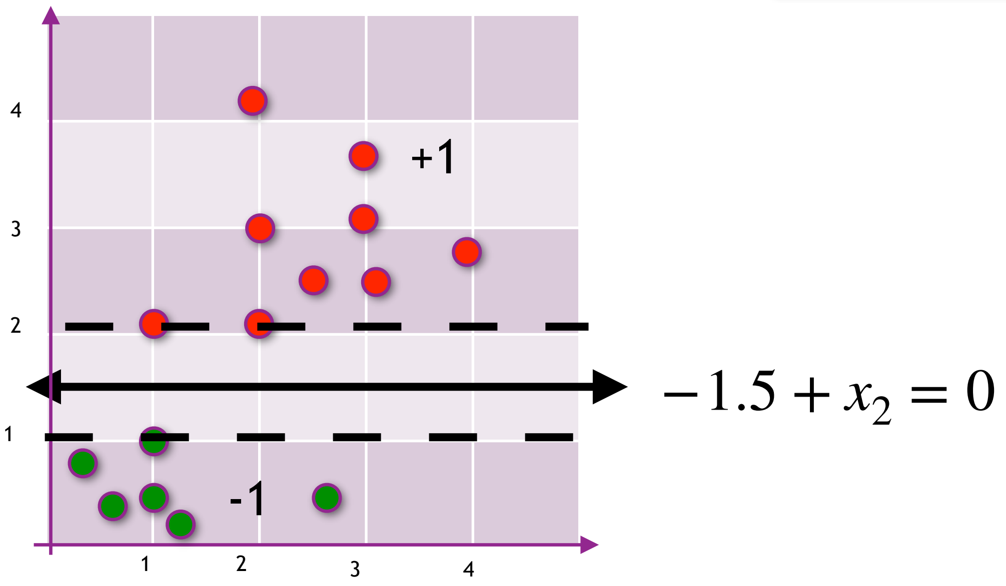
No!

We could scale w and w_0 to find a smaller $||\mathbf{w}||_2^2$

So the functional margin with respect to the set of training examples will be 1

Hard margin example

Pair share: How can modify our decision boundary to have a functional margin of 1?

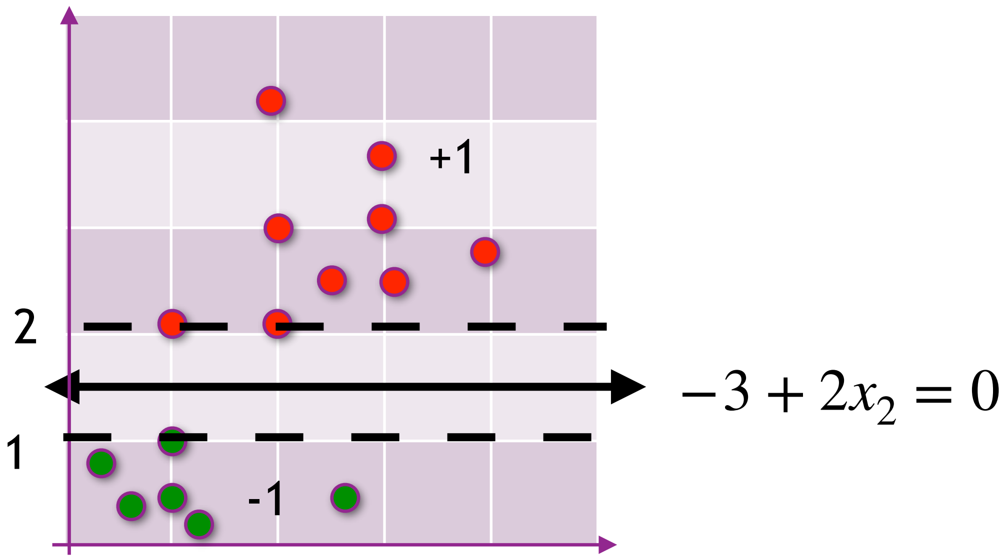


Decision boundary is $\mathbf{w} = [0,1]^T, w_0 = -1.5$

Is this the form we wanted?

The support vectors are supposed to have a functional margin of 1: $y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) = 1$

Example

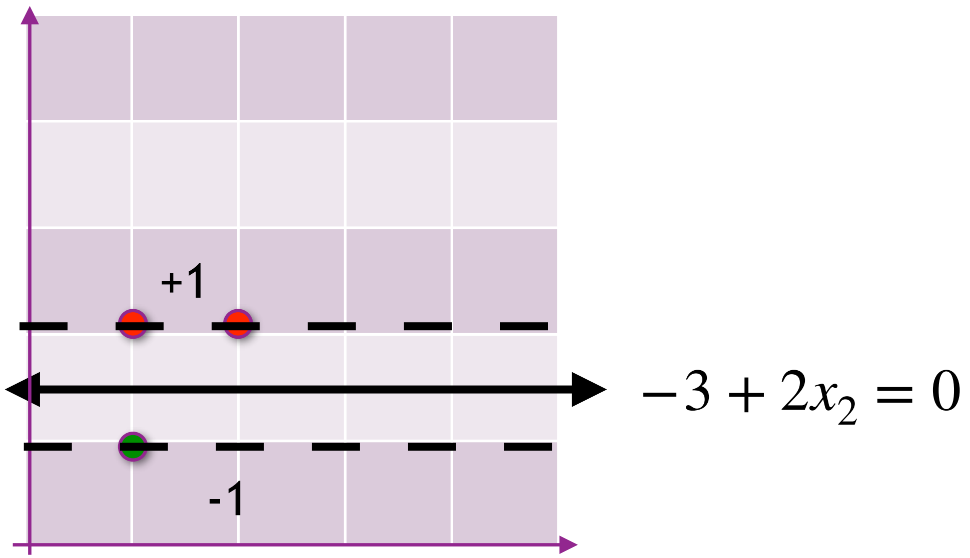


Decision boundary is $\mathbf{w} = [0,2]^T, w_0 = -3$

The support vectors have a functional margin of 1: $y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) = 1$

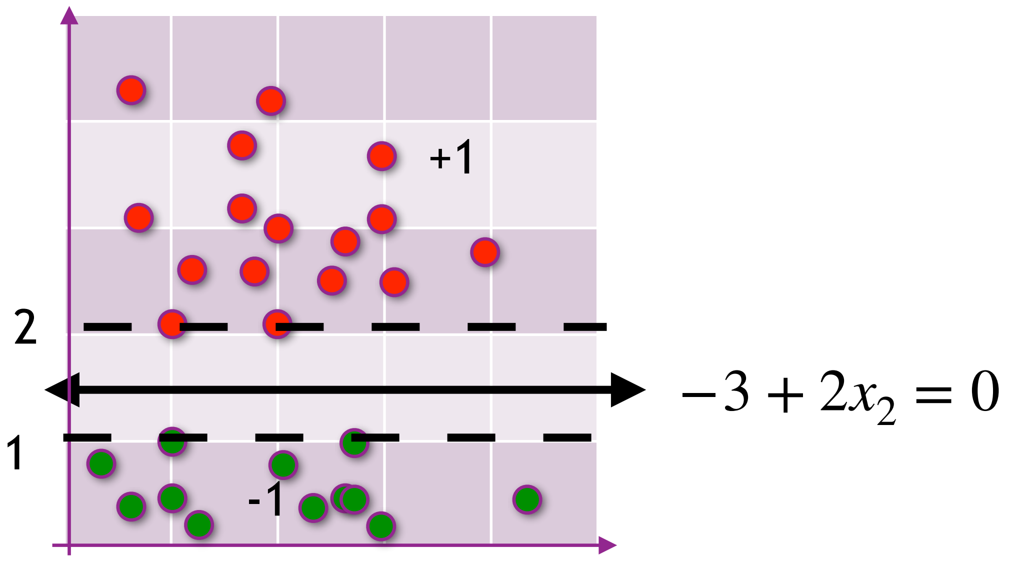
Approach taken in this slide is from CMU 18-661

Example



The boundary doesn’t change if I remove points with a functional margin > 1

Example



The solution doesn't change if I add points whose functional margin is ≥ 1

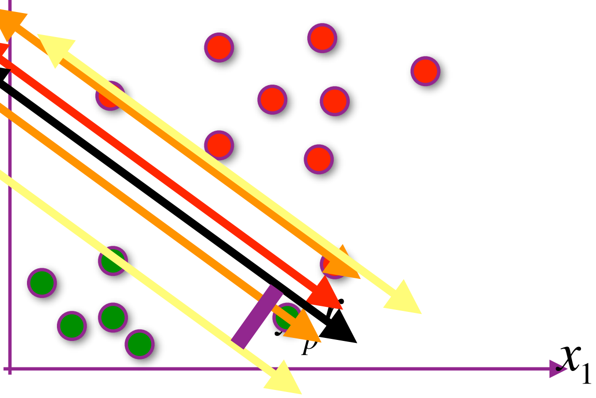
Approach taken in this slide is from CMU 18-661

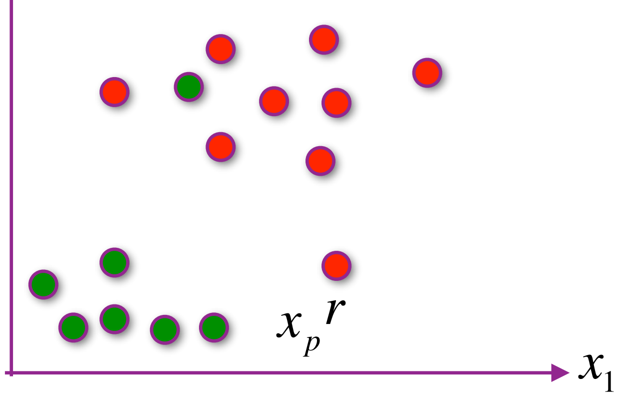
Outline

- ❑ Notation change, intuition, and finding how to compare hyperplanes - mathematically how do compare hyperplanes to find the one with the maximum margin. Can we turn this way of comparing hyperplanes into an objective function
- ❑ Support vector machines
  - ★ hard margin - find the constrained objective function when the data is linearly separable
  - ★ Dealing with non-linear data - “Soft” margins for SVM - New constrained objective function for the case where the data is not linearly separable
  - ★ Pegasos algorithm. Optimizer for soft margin SVM
  - ★ Dealing with non-linear data - feature transformation with the kernel trick - Show two popular feature maps

Non-Linear Data

1. Soft margin
2. Transform features vector $\mathbf{x}$ into a new feature space $\Phi(\mathbf{x})$



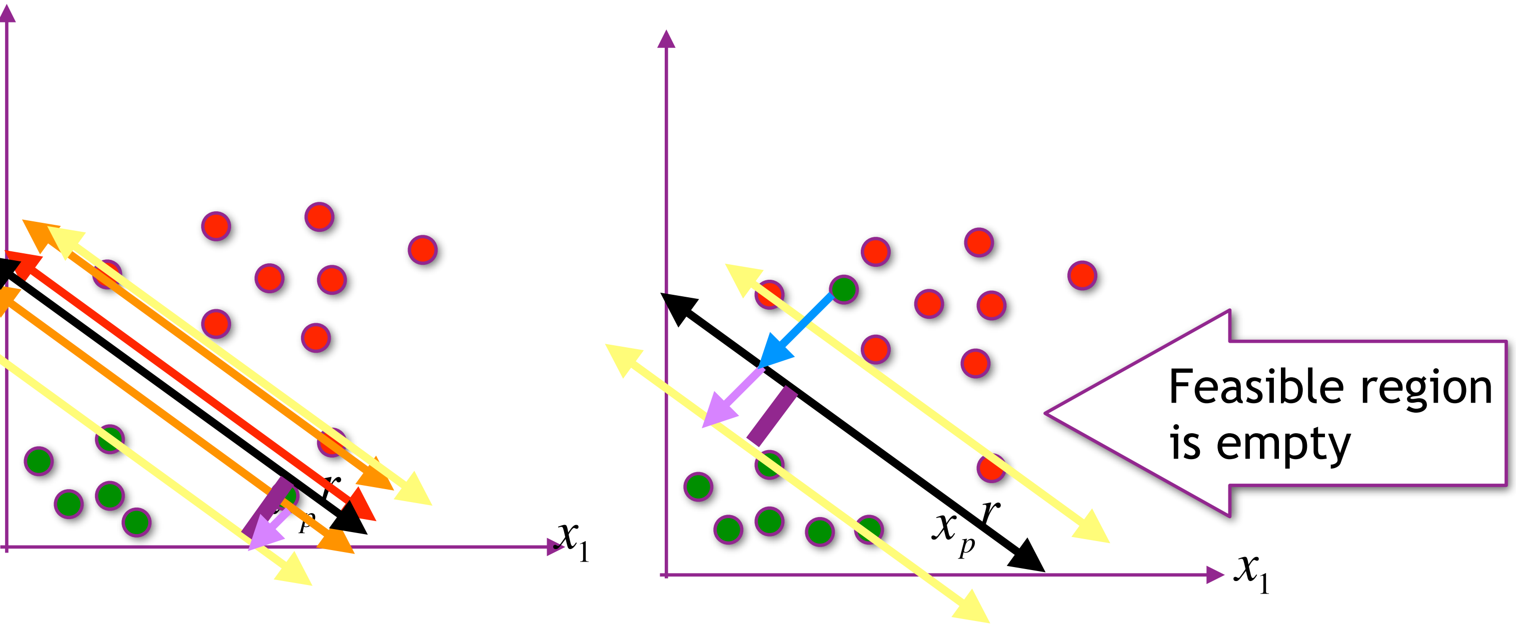


An outlier can make the data inseparable or not generalize well

What if the data isn’t linearly separable

WE CAN MAKE OUR MODEL MORE FLEXIBLE BY ADDING A COST FUNCTION FOR THE POINTS THAT ARE MISCLASSIFIED

Soft-Margin SVM



How can we still find find the optimal hyperplane where we allow for a few points to either be misclassified or within the margin?

We could incur a cost $\xi^{(i)}$ for how far the $x^{(i)}$ is away from the margin.

We will create a slack variable $\xi^{(i)}$ for each training example $\mathbf{x}^{(i)}$

The hyperplane must satisfy

$$y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) \geq 1 - \xi^{(i)}$$

$\xi^{(i)} =$ +

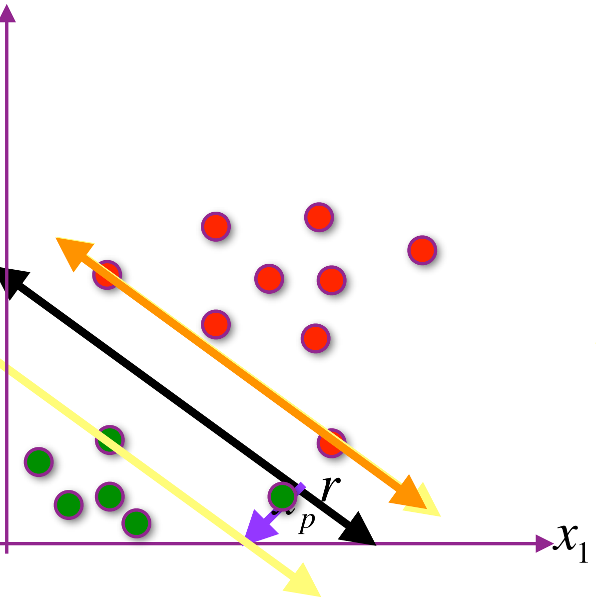
$\xi^{(i)} =$

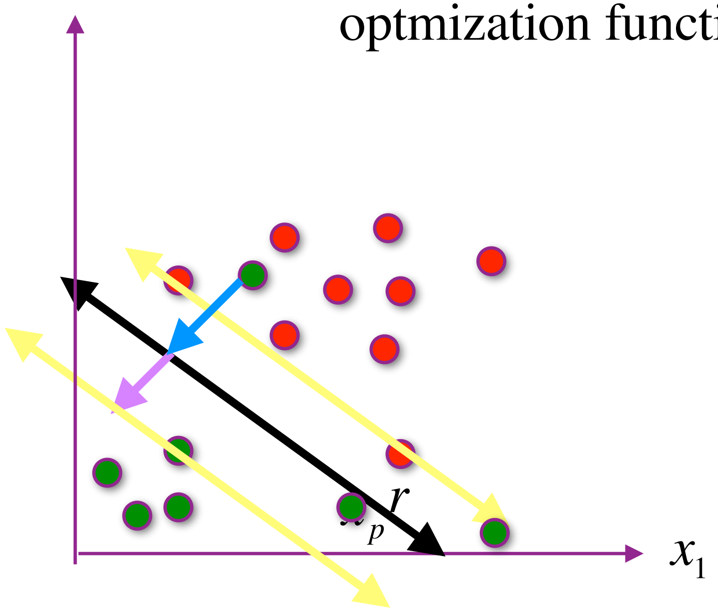
What function should we use for

Which cost function?

SHOULD ALL THE POINTS BE CHARGED, OR ONLY THOSE THAT ARE INSIDE THE MARGIN OR INCORRECTLY CLASSIFIED?

Soft-Margin SVM



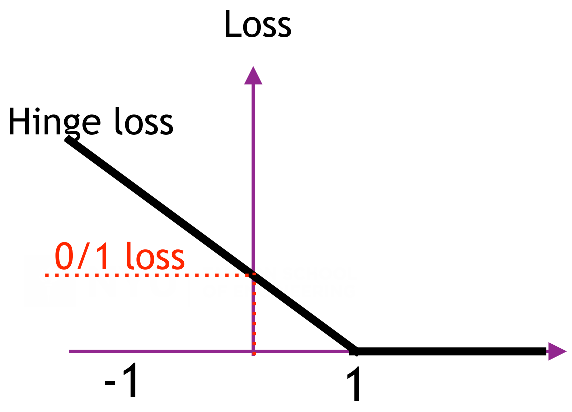


optmization function $$\min_{w_0, \mathbf{w}, \{\xi^{(i)}\}_{i=1}^N} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^{N} \xi^{(i)}$$

C is a tunable parameter. Gives relative importance of the error term

$$\text{subject to } y^{(i)}(w_0 + \mathbf{w}^T \mathbf{x}^{(i)}) \geq 1 - \xi^{(i)} \quad \text{for all } i = 1, \ldots, N$$

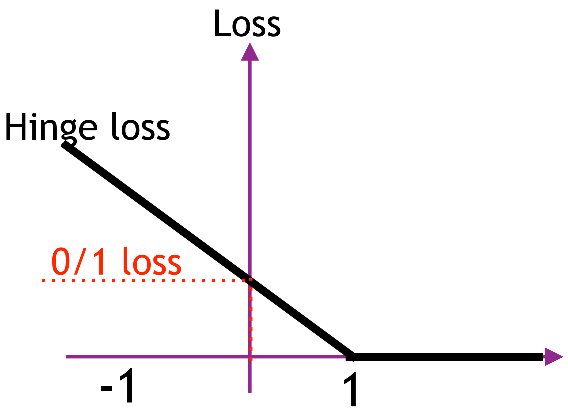
$$\xi^{(i)} \geq 0$$



$$\xi^{(i)} = \begin{cases} 0 & y^{(i)}(\mathbf{w}^T \mathbf{x}^{(i)} + w_0) \geq 1 \\ 1 - y^{(i)}(\mathbf{w}^T \mathbf{x}^{(i)} + w_0) & \textit{otherwise} \end{cases}$$

Hinge Loss

Introduced one slack variable $\xi^{(i)}$ for each training example.



$$\xi^{(i)} = \begin{cases} 0 & y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) \geq 1 \\ 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) & otherwise \end{cases}$$

0-1 loss: No penalty if correctly classified. Cost of 1 for incorrectly classified points

Hinge loss: Penalty upper bounds 0/1 loss

- Penalizes correct predictions that are too close to the margin
- Penalty linearly increases for incorrect predictions (and close correct predictions)

No penalty on correct predictions that are far away from the margin

optmization function

$$\min_{w_0, \mathbf{w}, \{\xi^{(i)}\}_{i=1}^{N}} ||\mathbf{w}||_2^2 + C \sum_{i=1}^{N} \xi^{(i)}$$

$$\text{subject to } y^{(i)}(w_0 + \mathbf{w}^T \mathbf{x}^{(i)}) \geq 1 - \xi^{(i)}$$

$$\xi^{(i)} \geq 0$$

Pair share: What do you know about the functional margin for $\mathbf{x}$ if:

1) $\xi \geq 1$

2) $0 < \xi < 1$

3) $\xi = 0$

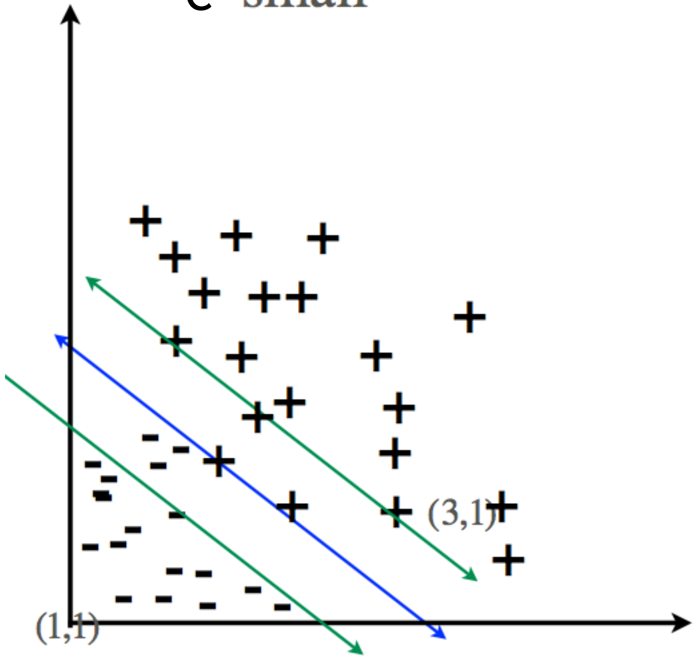
Pair share: Do you think that $\sum_{i=1}^{N} \xi^{(i)}$ is an upper bound on the number of training errors (i.e. number of points misclassified incorrectly)?

Pair share:

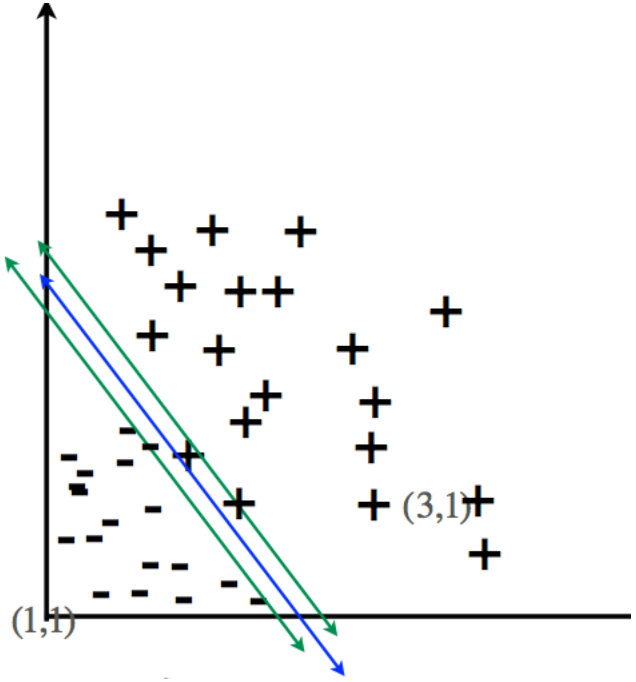
1) What happens to the margin if I make C large?

2) What happens to the margin if I make C small?

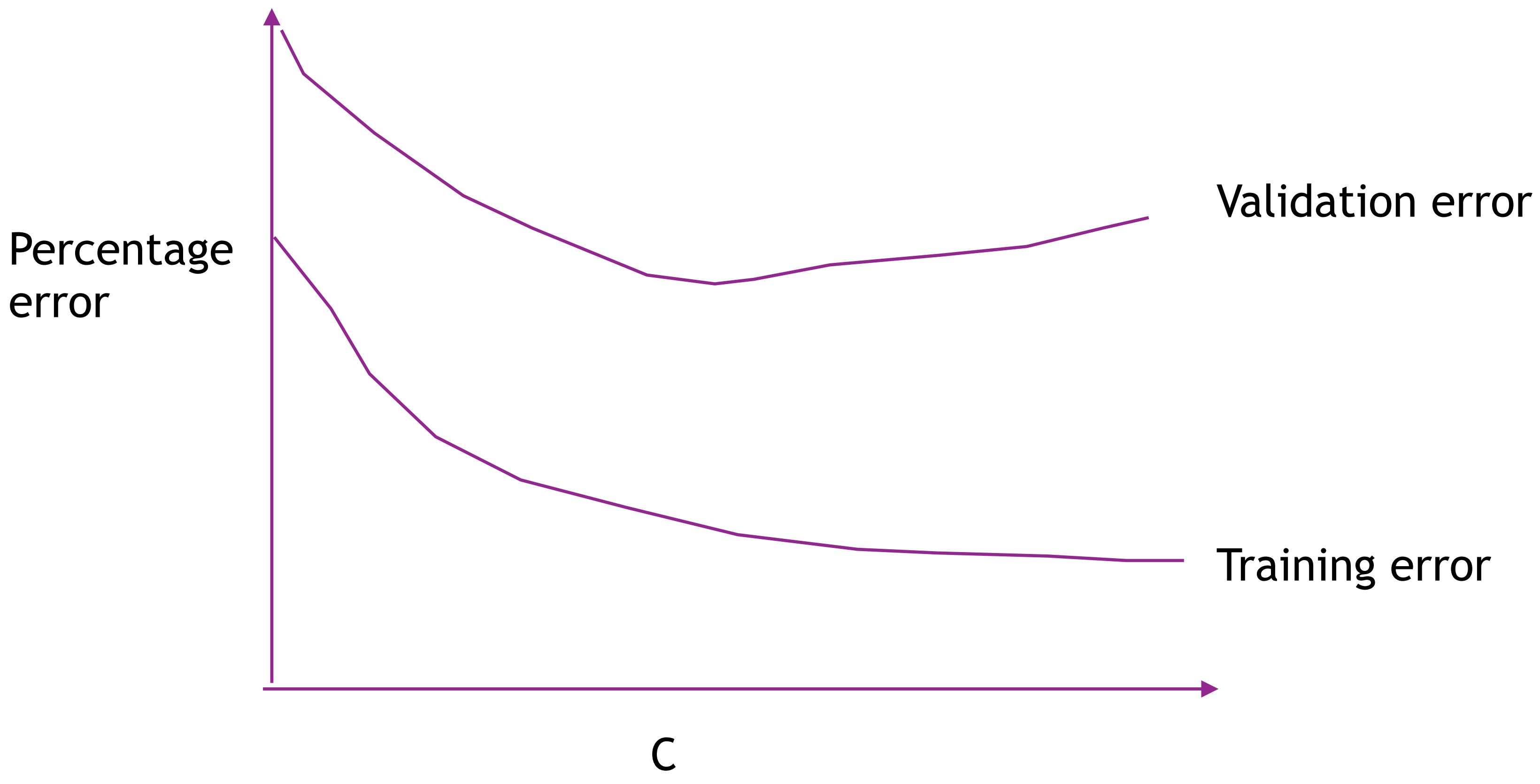
C small



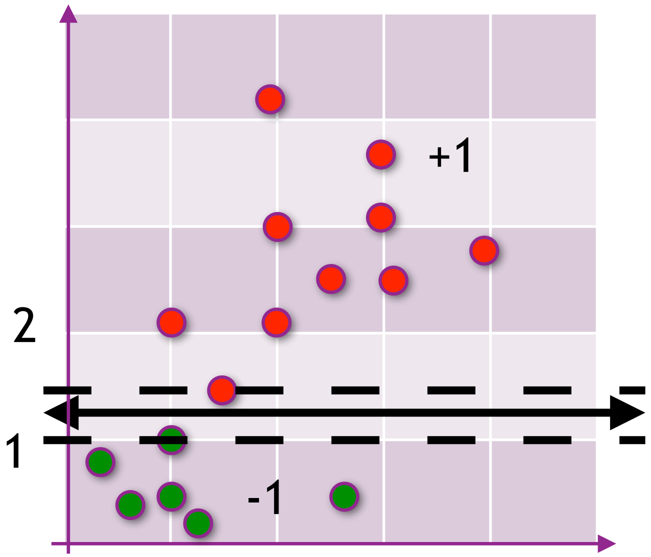
Larger C



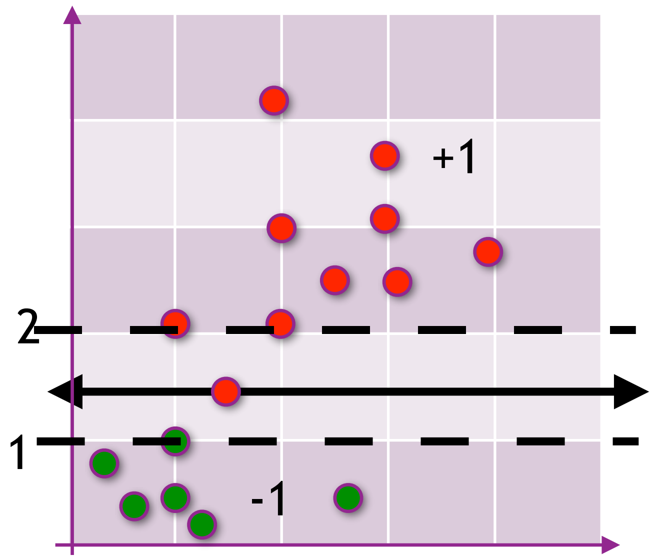
What if $C = \infty$?


Percentage error
Validation error
Training error
C

Example



Our margin becomes smaller if we have an outlier



$$\min_{w_0,\mathbf{w},\{\xi^{(i)}\}_{i=1}^{N}} \|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N} \xi^{(i)}$$

$$\text{subject to } y^{(i)}(w_0 + \mathbf{w}^T\mathbf{x}^{(i)}) \geq 1 - \xi^{(i)}$$

Approach taken in this slide is from CMU 18-661

Outline

- ❑ Notation change, intuition, and finding how to compare hyperplanes - mathematically how do compare hyperplanes to find the one with the maximum margin. Can we turn this way of comparing hyperplanes into an objective function
- ❑Support vector machines
  - ★ hard margin - find the constrained objective function when the data is linearly separable
  - ★ Dealing with non-linear data - “Soft” margins for SVM - New constrained objective function for the case where the data is not linearly separable
  - ★ Pegasos algorithm. Optimizer for soft margin SVM
  - ★ Dealing with non-linear data - feature transformation with the kernel trick - Show two popular feature maps

Simplifying our objective function

Rewriting our SVM objective function

$$\min_{w_0, \mathbf{w}, \{\xi^{(i)}\}_{i=1}^{N}} \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^{N} \xi^{(i)}$$

$$\text{subject to } y^{(i)}(w_0 + \mathbf{w}^T \mathbf{x}^{(i)}) \geq 1 - \xi^{(i)}$$

$$\xi^{(i)} \geq 0$$

Same as: $\xi^{(i)} \geq 1 - y^{(i)} \left(\mathbf{w}^T \mathbf{x}^{(i)} + w_0 \right)$

Our SVM objective function with hinge loss:

Setting $\lambda = 1/C$

$$\min_{\mathbf{w}, w_0} \frac{\lambda}{2} \underbrace{\|\mathbf{w}\|_2^2}_{\text{regularizer}} + \cancel{C} \sum_{i=1}^{N} \underbrace{\max \left(0, 1 - y^{(i)}(\mathbf{w}^T \mathbf{x}^{(i)} + w_0) \right)}_{\text{hinge Loss}}$$

Since $\xi^{(i)}$ is as small as possible

A balance between loss function and regularizer.

*In our optimizer, we will ignore the intercept term to make things easier

Our objective function is convex but not differentiable

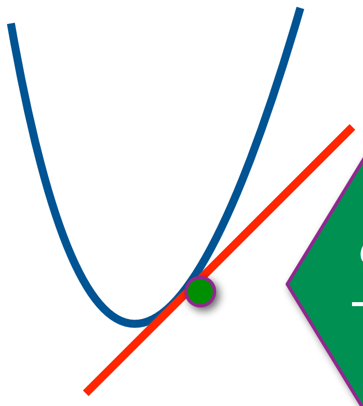
We can use a sub-gradient. Derivation is beyond the scope of course.

Derivative

Sub-derivative of the hinge loss

$$\frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

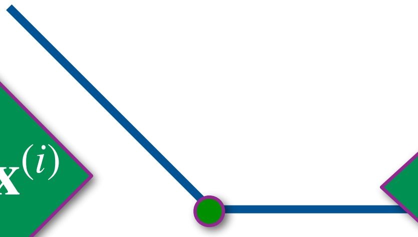
hinge loss $= \max\left(0, 1 - y^{(i)}\mathbf{w}^T\mathbf{x}\right)$



$$\frac{\partial \frac{\lambda}{2}\|\mathbf{w}\|_2^2}{\partial \mathbf{w}} = \lambda \mathbf{w}$$

+

$$\frac{\partial \text{hinge loss}}{\partial \mathbf{w}} = -y^{(i)}\mathbf{x}^{(i)}$$



?

$$\frac{\partial \text{hinge loss}}{\partial \mathbf{w}} = 0$$

$$J(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{N} \max\left(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0)\right)$$

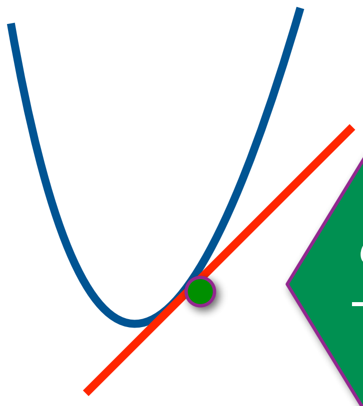
$$\nabla J(\mathbf{w}) = \lambda \mathbf{w} + \begin{cases} 0 & \text{if } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) \geq 1 \\ -y^{(i)}\mathbf{x}^{(i)} & \text{otherwise} \end{cases}$$

Derivative

Sub-derivative of the hinge loss

$$\frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

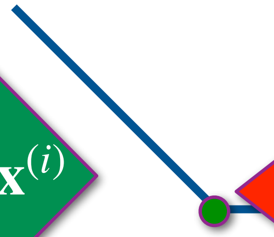
hinge loss $= \max\left(0, 1 - y^{(i)}\mathbf{w}^T\mathbf{x}\right)$



$$\frac{\partial \frac{\lambda}{2}\|\mathbf{w}\|_2^2}{\partial \mathbf{w}} = \lambda \mathbf{w}$$

+

$$\frac{\partial \text{hinge loss}}{\partial \mathbf{w}} = -y^{(i)}\mathbf{x}^{(i)}$$



?

Sub-gradient: Linear global underestimate at this point

$$J(\mathbf{w}) = \underline{\frac{\lambda}{2}\|\mathbf{w}\|_2^2} + \underline{\sum_{i=1}^{N} \max\left(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0)\right)}$$

$$\nabla J(\mathbf{w}) = \lambda \mathbf{w} + \begin{cases} 0 & \textbf{if } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0) \geq 1 \\ -y^{(i)}\mathbf{x}^{(i)} & \textbf{otherwise} \end{cases}$$

$$J(\mathbf{w}) = \underbrace{\frac{\lambda}{2}\|\mathbf{w}\|_2^2}_{\text{regularizer}} + \sum_{i=1}^{N} \underbrace{\max\left(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + w_0)\right)}_{\text{hinge Loss}}$$

$$\textbf{subgradient}(\mathbf{w}) = \begin{cases} \lambda\mathbf{w} & \textbf{if } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} \qquad) \geq 1 \\ \lambda\mathbf{w} - y^{(i)}\mathbf{x}^{(i)} & \textbf{otherwise} \end{cases}$$

"We did not incorporate a bias term in any of our experiments. We found that including an un-regularized bias term does not significantly change the predictive performance for any of the data sets used. Furthermore, most methods we compare to, including [21, 24, 37, 18], do not incorporate a bias term either. Nonetheless, there are clearly learning problems where the incorporation of the bias term could be beneficial." /https://www.cs.huji.ac.il/w~shais/papers/ShalevSiSrCo10.pdf

To keep it simple, we will not include a bias unit.

If N is large, batch gradient is slow

We will use stochastic sub-gradient descent
with an adaptive learning rate

Stochastic Gradient Descent

$\mathbf{w}$ = random initialization

For t = 1,2,…,T:

Pick a random training example $(\mathbf{x}^{(i)}, y^{(i)})$

$$\# \ \hat{J}(\mathbf{w}) = \frac{\lambda}{2}||\mathbf{w}||_2^2 + \max\left(0, 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} \qquad)\right)$$

Use only one training example in objective function, N=1

$$\mathbf{w} = \mathbf{w} - \alpha \nabla \hat{J}(\mathbf{w})$$

The Pegasos Algorithm

$$\textbf{subgradient}(\mathbf{w}) = \begin{cases} \lambda \mathbf{w} & \textbf{if } y^{(i)}(\mathbf{w}^T \mathbf{x}^{(i)}) \geq 1 \\ \lambda \mathbf{w} - y^{(i)} \mathbf{x}^{(i)} & \textbf{otherwise} \end{cases}$$

$\mathbf{w}$ = random initialization

For t = 1,2,…,T:

Pick a random training example $(\mathbf{x}^{(i)}, y^{(i)})$

Decrease the learning rate every iteration of the algorithm

Update the parameters by moving a small amount in the opposite direction of the sub gradient

Set

$T = \tilde{O}\left(\frac{1}{\delta \lambda \epsilon}\right)$

err w* $<$ w $+ \epsilon$

with probability

$1 - \delta$

To keep it simple, we will not include a bias unit.

The Pegasos Algorithm

$$\textbf{subgradient}(\mathbf{w}) = \begin{cases} \lambda \mathbf{w} & \textbf{if } y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)}) \geq 1 \\ \lambda \mathbf{w} - y^{(i)}\mathbf{x}^{(i)} & \textbf{otherwise} \end{cases}$$

$\mathbf{w}$ = random initialization

For t = 1,2,...,T:

Pick a random training example $(\mathbf{x}^{(i)}, y^{(i)})$

$$\alpha = \frac{1}{\lambda \cdot t}$$

if $y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)}) \geq 1$

$$\mathbf{w} = \mathbf{w} - \alpha\lambda\mathbf{w} \text{ \# weight decay}$$

else

$$\mathbf{w} = \mathbf{w} - \alpha(\lambda\mathbf{w} - y^{(i)}\mathbf{x}^{(i)})$$

Set $T = \tilde{O}\left(\frac{1}{\delta\lambda\epsilon}\right)$

err w* < w $+ \epsilon$

with probability

$1 - \delta$

Pair share: If α is small enough, will the function converge to a minimum value if enough iterations occur?

To keep it simple, we will not include a bias unit.

Modified Pegasos for Homework

$\mathbf{w}$ = 0, t = 0

For iter = 1,2,…,num_iters:

    For j = 1, 2, …, N:

        t=t+1

$$\alpha = \frac{1}{\lambda \cdot t}$$

if $y^{(j)}(\mathbf{w}^T\mathbf{x}^{(j)}) \geq 1$

$$\mathbf{w} = \mathbf{w} - \alpha\lambda\mathbf{w}$$ # weight decay

else

$$\mathbf{w} = \mathbf{w} - \alpha(\lambda\mathbf{w} - y^{(j)}\mathbf{x}^{(j)})$$

To keep it simple, we will not include a bias unit.