# Applied Machine Learning

## Support Vector Machines

Reihaneh Rabbany

School of Computer Science

# Learning objectives

geometry of linear classification

margin maximization and support vectors

hinge loss and relation to logistic regression

# geometry of the **separating hyperplane**

A linear decision boundary is a hyperplane with one dimension lower than D (number of features)

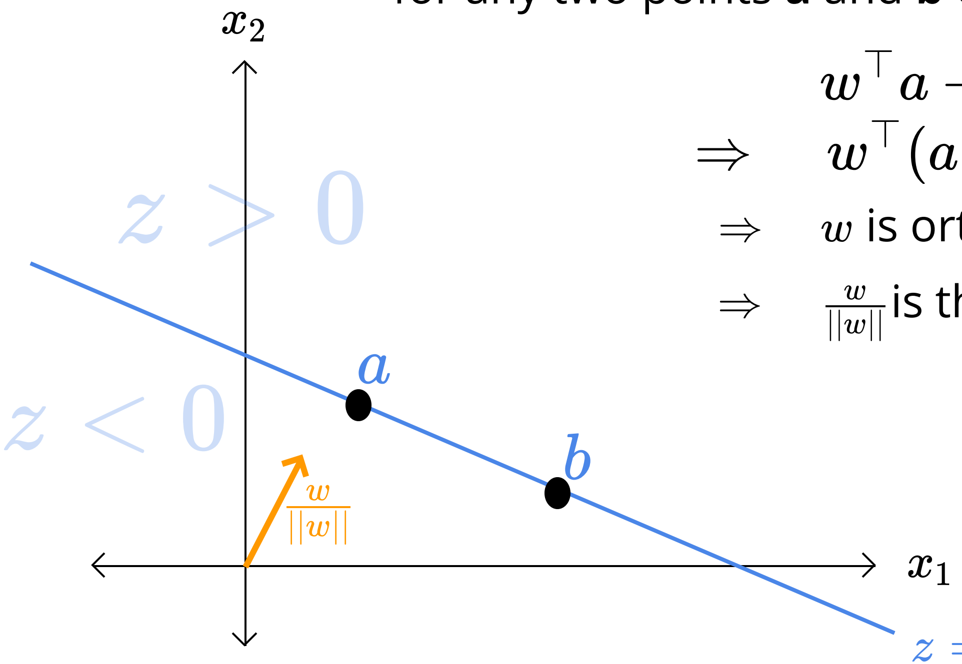
for any two points **a** and **b** on this line, we have:

$$w^\top a + w_0 = w^\top b + w_0 = 0$$

$$\Rightarrow \quad w^\top (a - b) + w_0 - w_0 = 0$$

$\Rightarrow$ $w$ is orthogonal to the line

$\Rightarrow$ $\frac{w}{||w||}$ is the unit vector normal to the line




$$z = w^\top x + w_0 = w_2 x_2 + w_1 x_1 + w_0 = 0$$

## geometry of the **separating hyperplane**

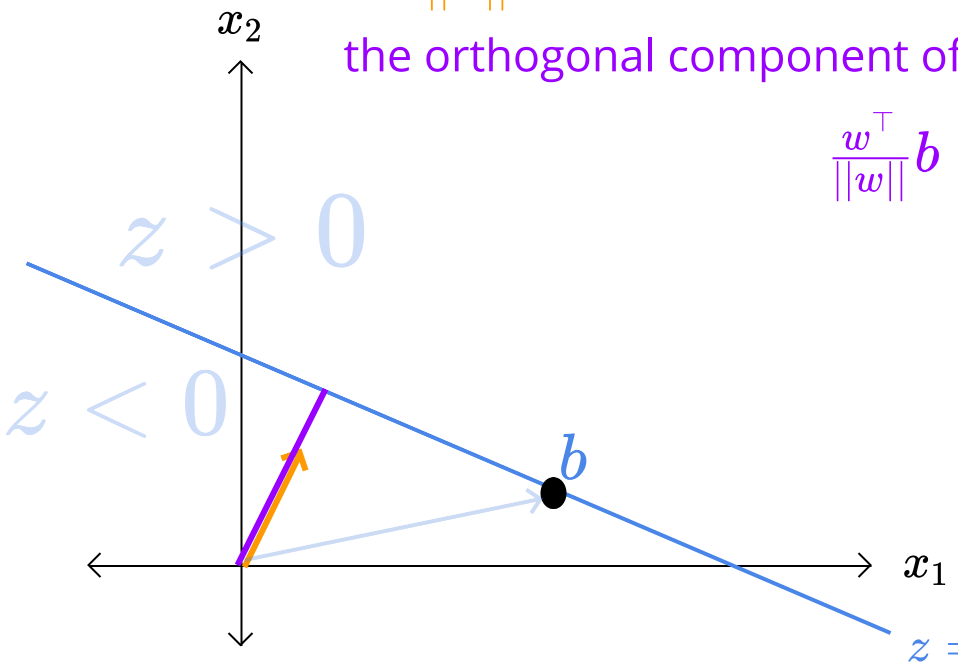
A linear decision boundary is a hyperplane with one dimension lower than D (number of features)

so $\frac{w}{||w||}$ is the unit normal vector to the line

the orthogonal component of any point on the line

$$\frac{w^\top}{||w||}b = \frac{w^\top}{||w||}b + \frac{w_0}{||w||} - \frac{w_0}{||w||} = -\frac{w_0}{||w||}$$

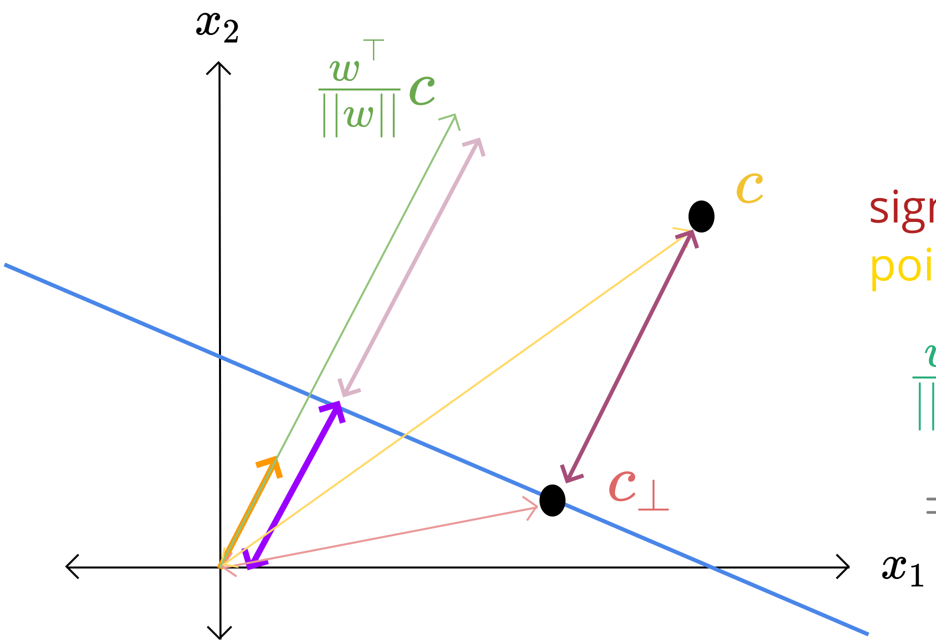
$$w^\top b + w_0 = 0$$




$$z = w^\top x + w_0 = w_2 x_2 + w_1 x_1 + w_0 = 0$$

geometry of the **separating hyperplane**

the orthogonal component of any point on the line $\frac{w^\top}{||w||}b = -\frac{w_0}{||w||}$ :(**)




signed distance of any point (c) from the line

$$\frac{w^\top}{||w||}c - \frac{w^\top}{||w||}c_\perp$$

$$= \frac{w^\top}{||w||}c + \frac{w_0}{||w||}$$

$$= \frac{1}{||w||}(w^\top c + w_0)$$

$\frac{w}{||w||}$

$\frac{w^\top}{||w||}c$

$c_\perp$

$\frac{w^\top}{||w||}c_\perp = -\frac{w_0}{||w||}$ (**)

# Perceptron: objective

Reminder

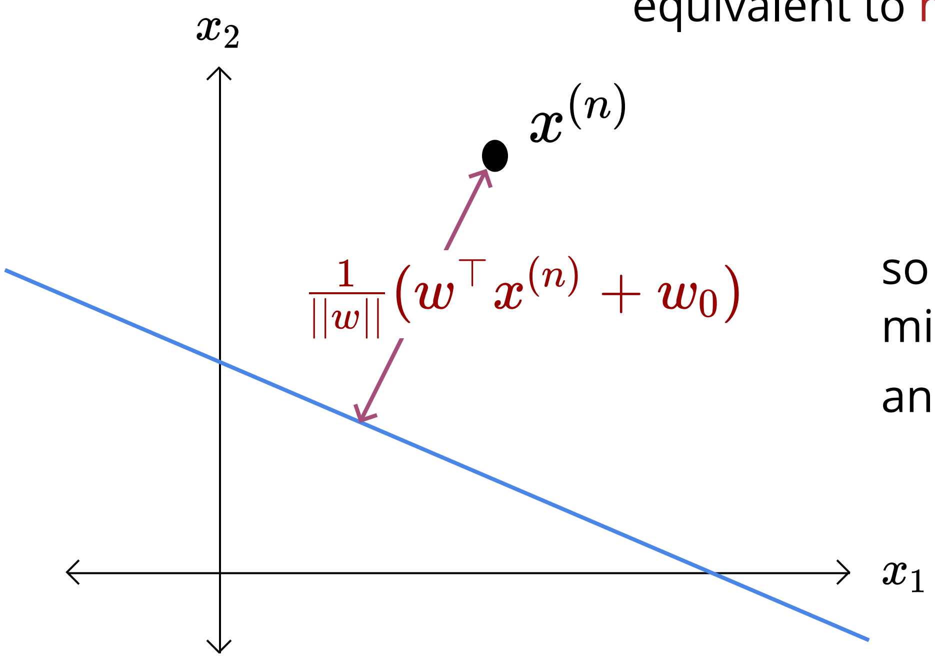
note that $y$ is -1 or 1 instead of 0 or 1

if $y^{(n)}\hat{y}^{(n)} < 0$ try to make it positive

label and prediction have different signs $\hat{y}^{(n)} = \text{sign}(w^\top x^{(n)} + w_0)$

equivalent to minimizing $-y^{(n)} \underbrace{(w^\top x^{(n)} + w_0)}_{\text{distance to the boundary}}$

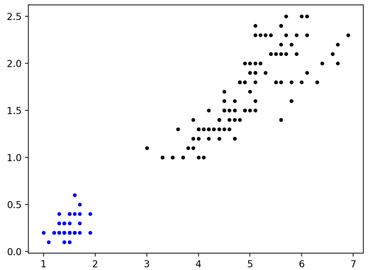
this is positive for points that are on the wrong side




so perceptron tries to minimize the distance of misclassified points from the decision boundary

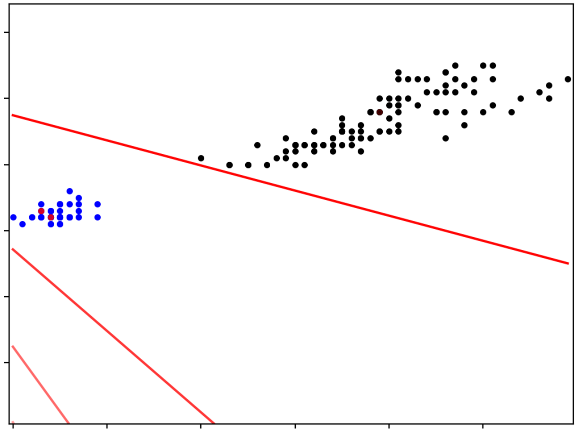
and push them to the right side

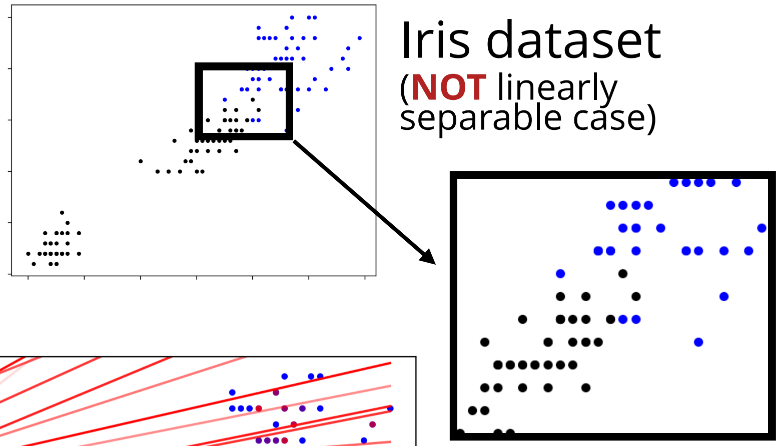
# Perceptron: example



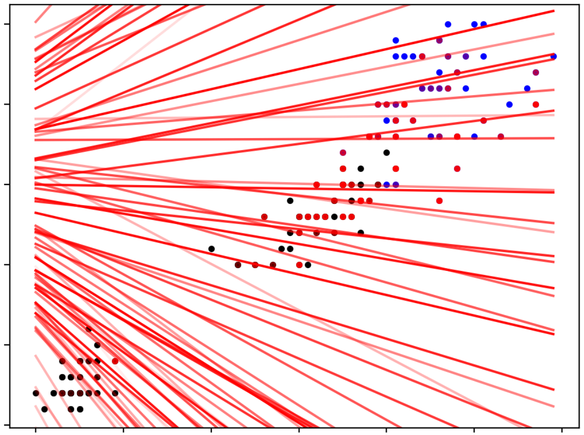

Iris dataset
(linearly separable case)

converged at iteration 10





Iris dataset
(**NOT** linearly separable case)



the algorithm does not converge

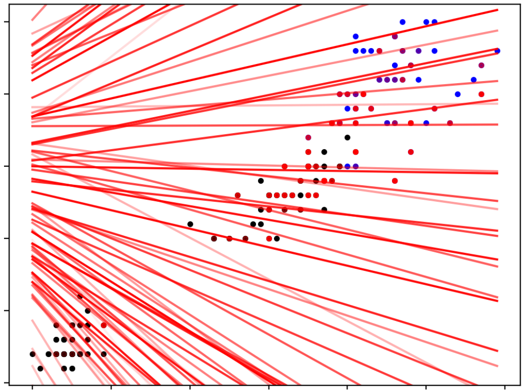
there is always a wrong prediction and the weights will be updated

# Perceptron: issues

Perceptron is not expressive enough

increase the model's expressiveness by adaptive nonlinear bases, discussed in previously in MLP

previously



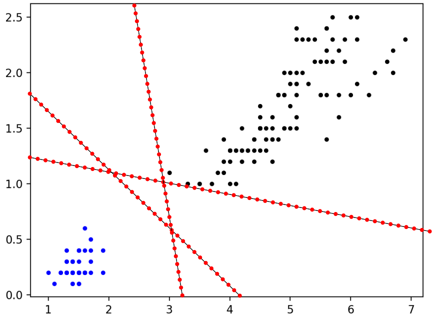
even if linearly separable

convergence could take many iterations

the decision boundary may be suboptimal

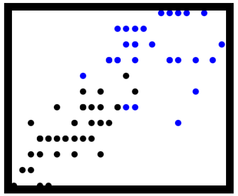
let's fix this problem first

assume linear separability




cyclic updates if the data is not perfectly linearly separable

- data may be inherently noisy



# Margin

the margin of a classifier (assuming correct classification)

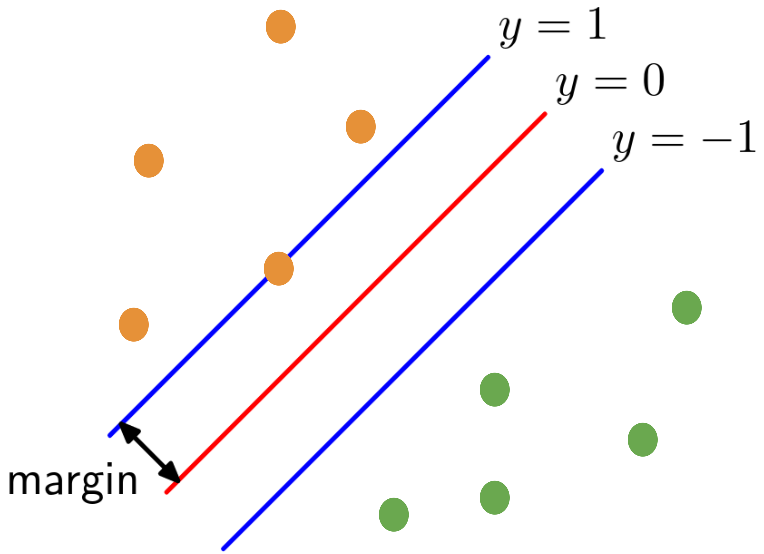
is the distance of the closest point to the decision boundary

signed distance is $\frac{1}{||w||}(w^\top x^{(n)} + w_0)$

adjust so that correctly classified points have positive margin

$$\underbrace{\frac{1}{||w||}(w^\top x^{(n)} + w_0)}_{\hat{y}^{(n)}\text{=distance to the boundary}} y^{(n)}$$

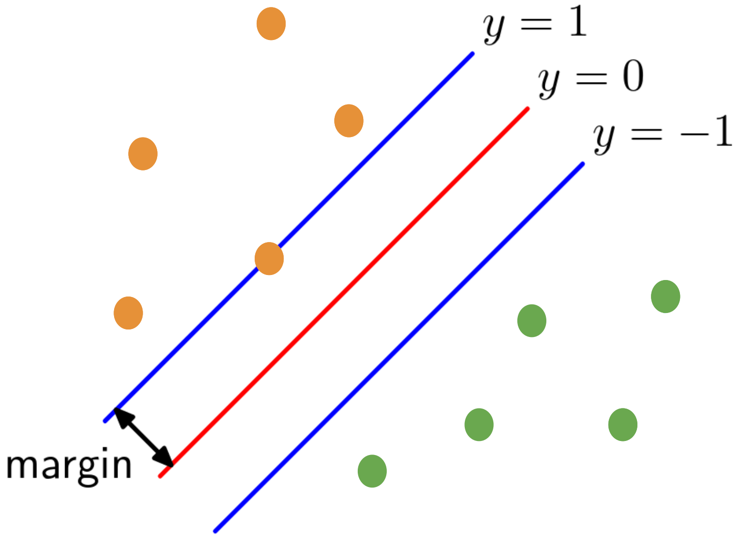
this is positive for points that are on the right side



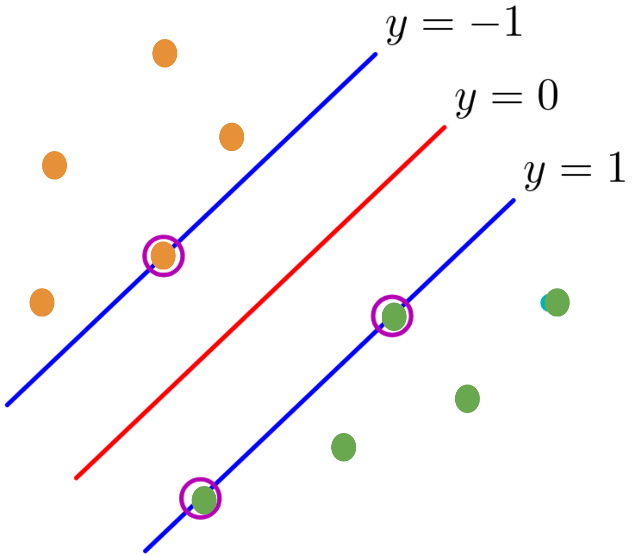
# Max margin classification

find the decision boundary with maximum margin

margin is not maximal

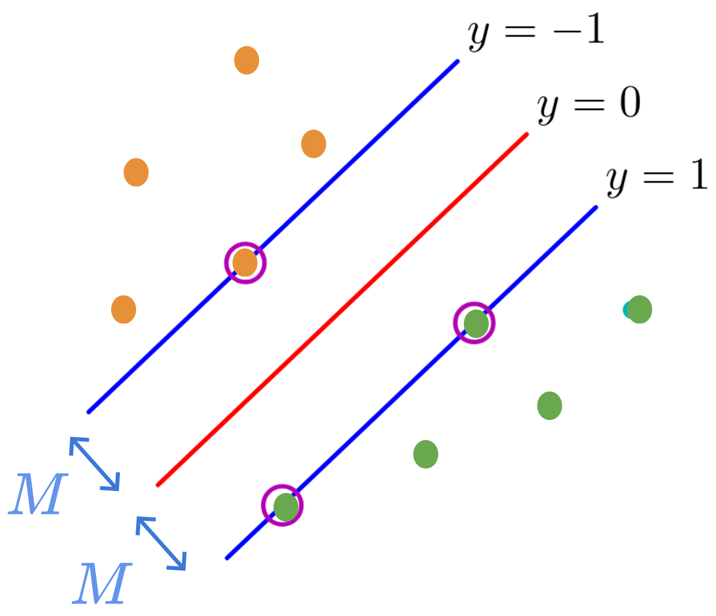



maximum margin



# Max margin classification

find the decision boundary with maximum margin




$$\begin{cases} \max_{w,w_0} M \\ M \leq \frac{1}{||w||_2} y^{(n)}(w^\top x^{(n)} + w_0) \quad \forall n \end{cases}$$

only the points (n) with

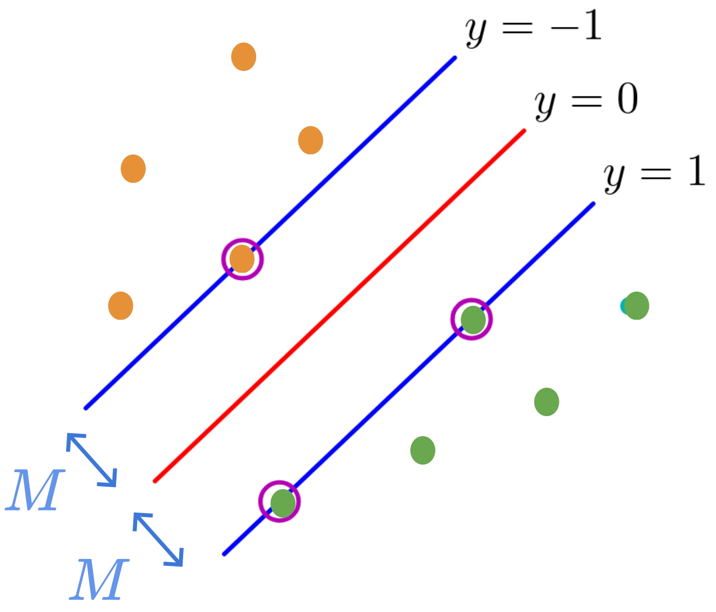
$M = \frac{1}{||w||_2} y^{(n)}(w^\top x^{(n)} + w_0)$ matter in finding the boundary

these are called **support vectors**

max-margin classifier is called **support vector machine** (SVM)

# Support Vector Machine

find the decision boundary with maximum margin




$$\begin{cases} \max_{w,w_0} M \\ M \leq \frac{1}{||w||_2} y^{(n)}(w^\top x^{(n)} + w_0) \quad \forall n \end{cases}$$

**observation**

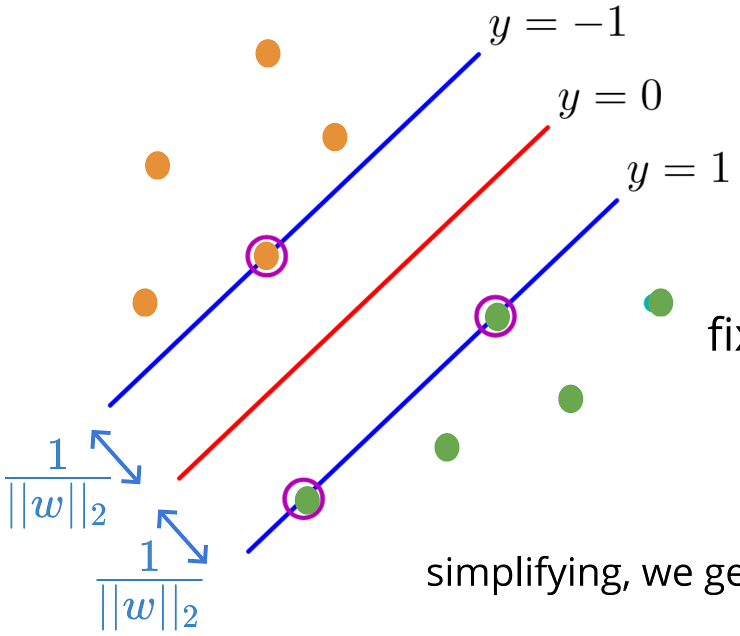
if $w^*, w_0^*$ is an optimal solution then

$20w^*, 20w_0^*$ is also optimal (same margin)

fix the norm of w to avoid this $||w||_2 = \frac{1}{M}$

# Support Vector Machine

find the decision boundary with maximum margin




$$\begin{cases} \max_{w,w_0} M \\ M \leq \frac{1}{||w||_2} y^{(n)}(w^\top x^{(n)} + w_0) \quad \forall n \end{cases}$$

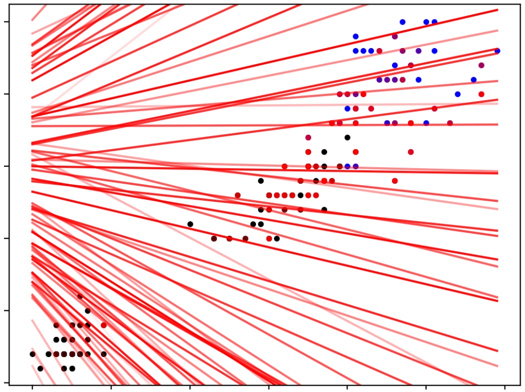
fixing $||w||_2 = \frac{1}{M}$

$$\begin{cases} \max_{w,w_0} \frac{1}{||w||_2} \\ \frac{1}{||w||_2} \leq \frac{1}{||w||_2} y^{(n)}(w^\top x^{(n)} + w_0) \quad \forall n \end{cases}$$

simplifying, we get hard margin SVM objective

$$\begin{cases} \min_{w,w_0} ||w||_2^2 \\ y^{(n)}(w^\top x^{(n)} + w_0) \geq 1 \quad \forall n \end{cases}$$

# Perceptron: issues



## Perceptron is not expressive enough

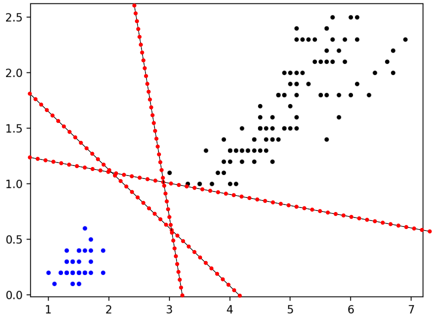
increase the model's expressiveness by adaptive nonlinear bases, discussed in previously in MLP

previously

even if linearly separable

convergence could take many iterations

the decision boundary may be suboptimal



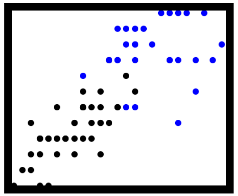

maximize the **hard** margin

cyclic updates if the data is not perfectly linearly separable

- data may be inherently noisy

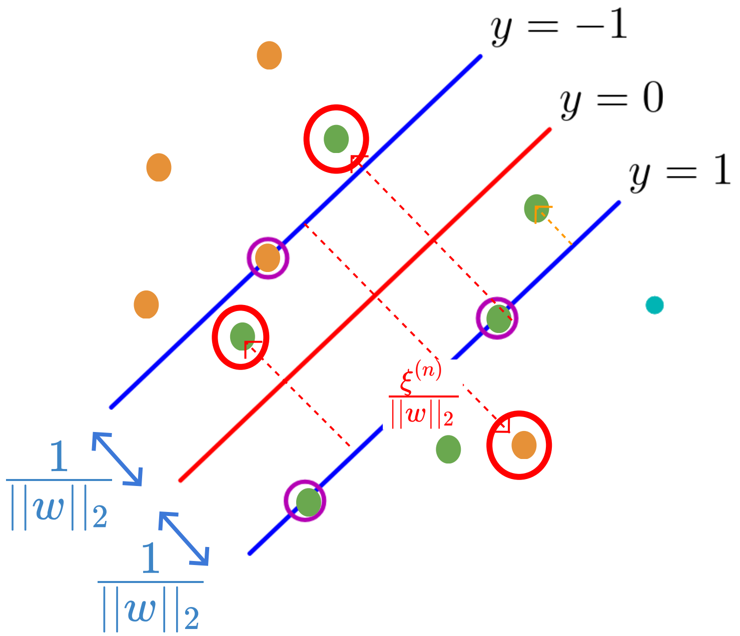
now lets fix this problem

maximize a **soft** margin



# Soft margin constraints

allow points inside the margin and on the wrong side
but penalize them




instead of hard constraint $y^{(n)}(w^\top x^{(n)} + w_0) \geq 1 \quad \forall n$

use $y^{(n)}(w^\top x^{(n)} + w_0) \geq 1 - \xi^{(n)} \quad \forall n$

$\xi^{(n)} \geq 0$ slack variables (one for each n)

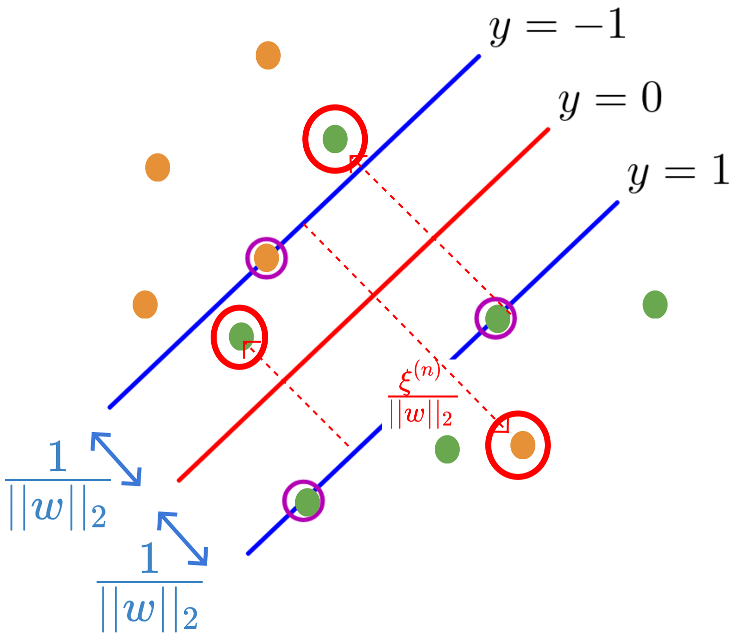
$\xi^{(n)} = 0$ zero if the point satisfies original margin constraint

$0 < \xi^{(n)} < 1$ if correctly classified but inside the margin

$\xi^{(n)} > 1$ incorrectly classified

# Soft margin constraints

allow points inside the margin and on the wrong side but penalize them




soft-margin objective

$$\min_{w,w_0} \frac{1}{2}||w||_2^2 + \gamma \sum_n \xi^{(n)}$$

$$y^{(n)}(w^\top x^{(n)} + w_0) \geq 1 - \xi^{(n)} \quad \forall n$$

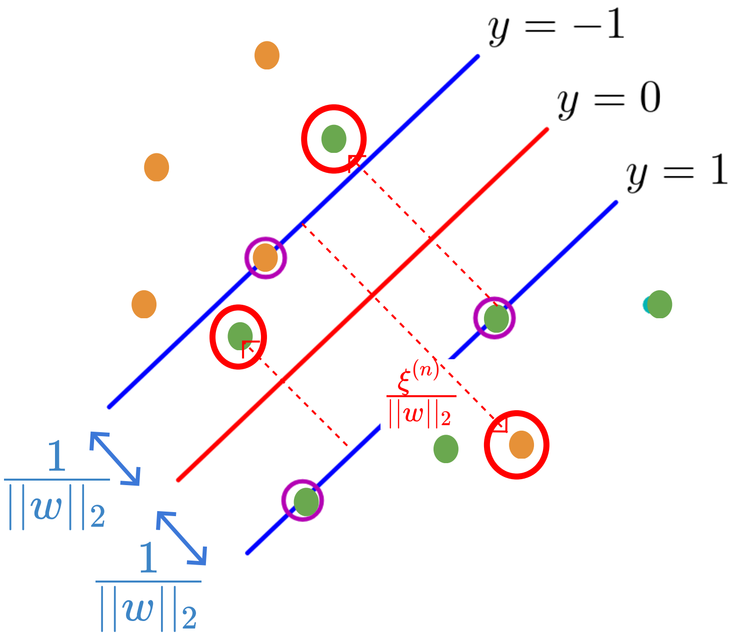
$$\xi^{(n)} \geq 0 \quad \forall n$$

$\gamma$ is a hyper-parameter that defines the importance of constraints

for very large $\gamma$ this becomes similar to hard margin svm

# Hinge loss

would be nice to turn this into an unconstrained optimization




$$\min_{w,w_0} \frac{1}{2}||w||_2^2 + \gamma \sum_n \xi^{(n)}$$

$$y^{(n)}(w^\top x^{(n)} + w_0) \geq 1 - \xi^{(n)}$$

$$\xi^{(n)} \geq 0 \quad \forall n$$

> if point satisfies the margin $y^{(n)}(w^\top x^{(n)} + w_0) \geq 1$
> minimum slack is $\xi^{(n)} = 0$

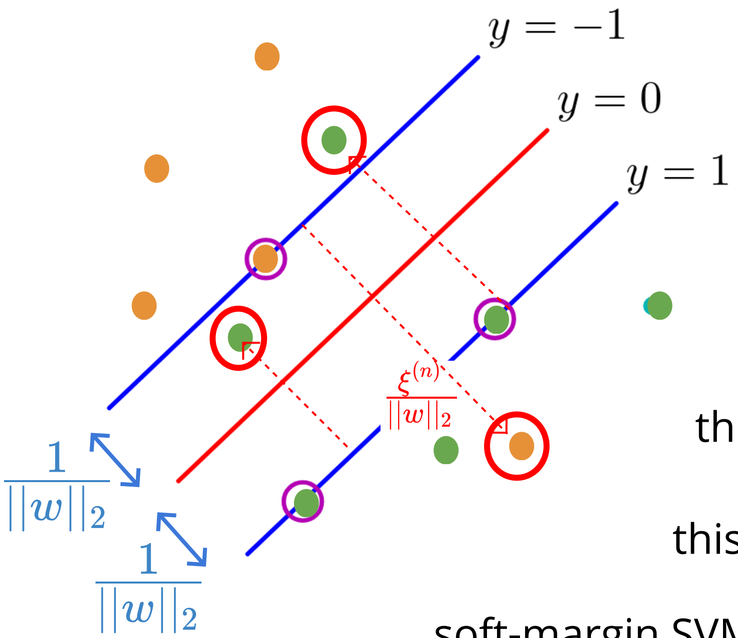
> otherwise $y^{(n)}(w^\top x^{(n)} + w_0) < 1$
> the smallest slack is $\xi^{(n)} = 1 - y^{(n)}(w^\top x^{(n)} + w_0)$

so the optimal slack satisfying both cases

$$\xi^{(n)} = \max(0, 1 - y^{(n)}(w^\top x^{(n)} + w_0))$$

# Hinge loss

would be nice to turn this into an unconstrained optimization




$$\min_{w,w_0} \tfrac{1}{2}||w||_2^2 + \gamma \textstyle\sum_n \xi^{(n)}$$

$$y^{(n)}(w^\top x^{(n)} + w_0) \geq 1 - \xi^{(n)}$$

$$\xi^{(n)} \geq 0 \quad \forall n$$

replace $\xi^{(n)} = \max(0, 1 - y^{(n)}(w^\top x^{(n)} + w_0))$

we get $\min_{w,w_0} \frac{1}{2}||w||_2^2 + \gamma \sum_n \max(0, 1 - y^{(n)}(w^\top x^{(n)} + w_0))$

the same as $\min_{w,w_0} \sum_n \max(0, 1 - y^{(n)}(w^\top x^{(n)} + w_0)) + \frac{1}{2\gamma}||w||_2^2$

this is called the hinge loss $L_{hinge}(y, z) = \max(0, 1 - yz)$

soft-margin SVM is doing L2 regularized hinge loss minimization

# Perceptron vs. SVM

## Perceptron

cost

if correctly classified evaluates to zero

otherwise it is $-y^{(n)}(w^\top x^{(n)} + w_0))$

can be written as

$\sum_n \max(0, -y^{(n)}(w^\top x^{(n)} + w_0))$

optimization

finds some linear decision boundary if exists

stochastic gradient descent with fixed learning rate

## SVM

cost

$\sum_n \max(0, \boxed{1} - y^{(n)}(w^\top x^{(n)} + w_0)) + \frac{\lambda}{2}||w||_2^2$

*so this is the difference!*

*(plus regularization)*

optimization

for small lambda finds the max-margin decision boundary

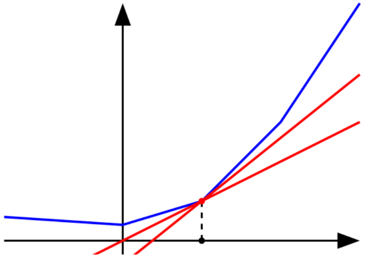
depending on the formulation we have many choices

# Perceptron vs. SVM

cost $\quad J(w) = \sum_n \max(0, 1 - y^{(n)} w^\top x^{(n)}) + \frac{\lambda}{2}||w||_2^2$

now we included bias in w

check that the cost function is convex in w(?)



hinge loss is not smooth (piecewise linear)

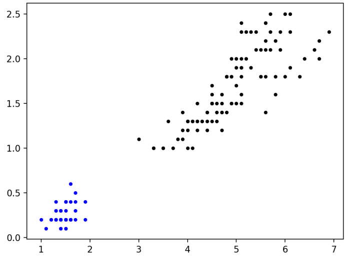
if we use **"stochastic" sub-gradient** descent

the update will look like Perceptron

if $\quad y^{(n)} \hat{y}^{(n)} < 1$ minimize $\quad -y^{(n)}(w^\top x^{(n)}) + \frac{\lambda}{2}||w||_2^2$

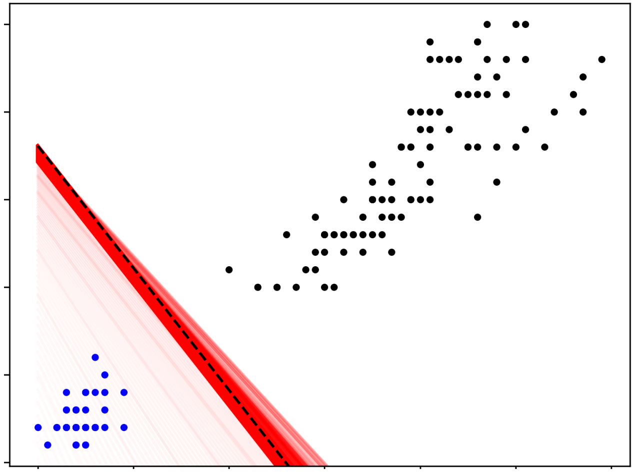
otherwise, do nothing

# Example: linearly separable

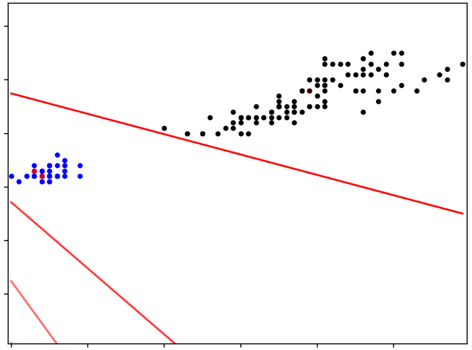



Iris dataset (D=2)

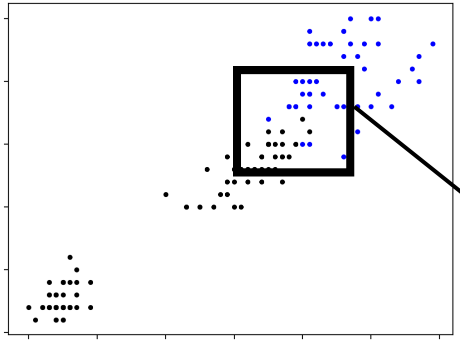
max-margin boundary (using small lambda $\lambda = 10^{-8}$ )



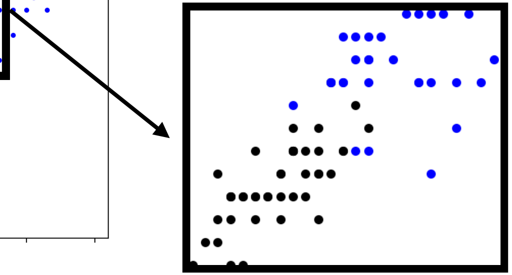
compare to Perceptron's decision boundary



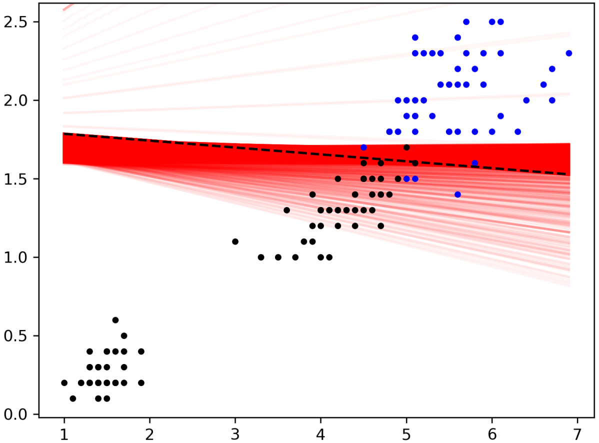
# Example: not linearly separable



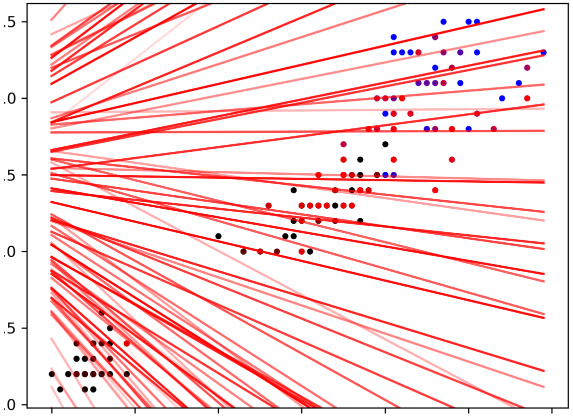
Iris dataset (D=2)



soft margins using small lambda $\lambda = 10^{-8}$




Perceptron does not converge



# SVM vs. logistic regression

recall: **logistic regression** simplified cost for $y \in \{0,1\}$

$$J(w) = \sum_{n=1}^{N} y^{(n)} \log\left(1 + e^{-z^{(n)}}\right) + (1 - y^{(n)}) \log\left(1 + e^{z^{(n)}}\right) \quad \text{where} \quad z^{(n)} = w^\top x^{(n)}$$

includes the bias

for $y \in \{-1,+1\}$ we can write this as

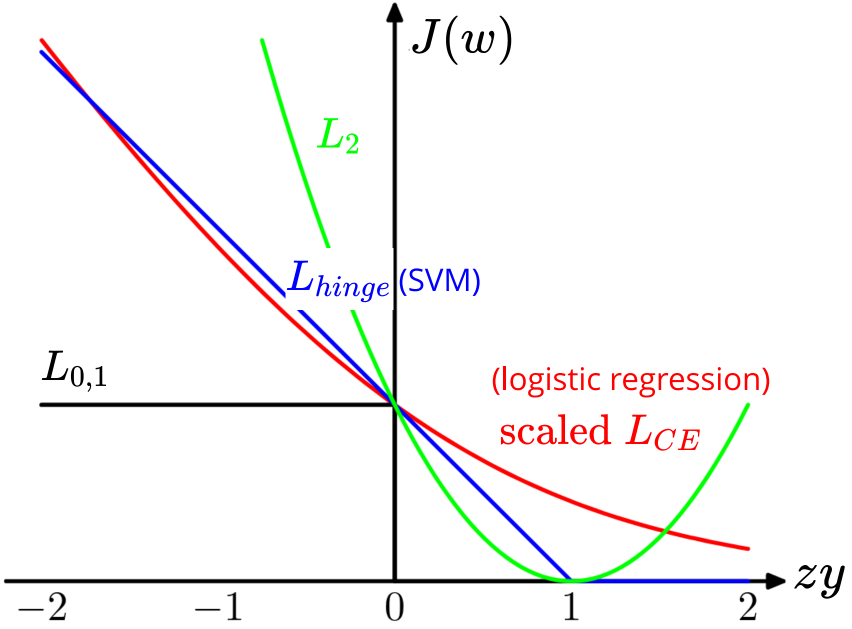
$$J(w) = \sum_{n=1}^{N} \log\left(1 + e^{-y^{(n)} z^{(n)}}\right) + \frac{\lambda}{2}||w||_2^2$$

*also added L2 regularization*

compare to **SVM cost** for $y \in \{-1,+1\}$

$$J(w) = \sum_n \max(0, 1 - y^{(n)}(z^{(n)})) + \frac{\lambda}{2}||w||_2^2$$

they both try to approximate 0-1 loss (accuracy)



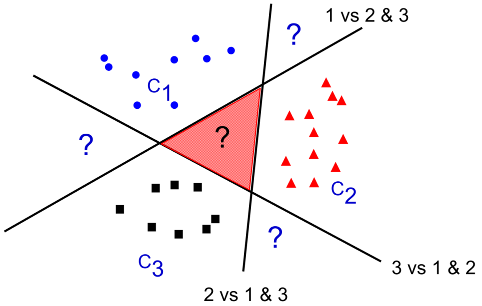
# Multiclass classification

can we use multiple binary classifiders?

**one versus the rest**

**training:**

train C different 1-vs-(C-1) classifiers $z_c(x) = w_c^\top x$




**test time:**

choose the class with the highest score

$$z^* = \arg\max_c z_c(x)$$

**problems:**

class imbalance

not clear what it means to compare $z_c(x)$ values, trained on different tasks

image credit: Andrew Zisserman

# Multiclass classification

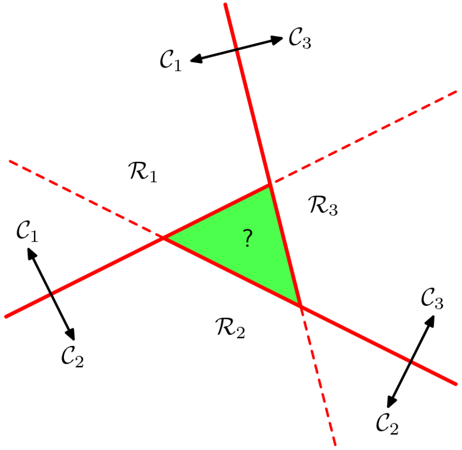
can we use multiple binary classifiders?

**one versus one**

**training:**
train $\frac{C(C-1)}{2}$ classifiers for each class pair

**test time:**
choose the class with the highest vote

**problems:**
computationally more demanding for large C
ambiguities in the final classification



# Summary

- geometry of linear classification
- distance to the decision boundary (margin)
- max-margin classification
- support vectors
- hard vs soft SVM
- relation to perceptron
- hinge loss and its relation to logistic regression
- some ideas for max-margin multi-class classification