CS2109S: Introduction to AI and Machine Learning

Lecture 7:
Support Vector Machines

13 October 2023

Announcements

Final Assessment

- **Correction** of release and due date:
  - Release Date: **November 25, 2023 (Saturday), at 20:00**
  - Due Date: **November 26, 2023 (Sunday) at 23:59**
- Format: Take-home exam
- We will calibrate it such that it **can be completed within 3-5 hours**
- **Mock assessment will be available ~2 weeks before** the final exam
  - People can have a chance to try

Quick notes on Midterm

- Q1: Standard uninformed search formulation
- Q2: A* search + Heuristics
- Q3: Decision Trees
- Q4: Game tree + $\alpha\beta$-pruning
- Q5: For fun.

Solutions uploaded!

Grading on progress, maybe end of next week…

Midterm Survey

Due Today

1. How do you find the lectures?

** Required*

S/N	Multiple Choice Option	Count	Percentage
1	Lectures are clear and I am able to follow the material quite well	123	51.9%
2	Lectures are no different from the other classes at campus	67	28.3%
3	I have no idea what's happening half the time	28	11.8%
4	Lectures are way cool. Easily the best class that I've taken at NUS	13	5.5%
5	I have no clue what the lecturer is talking about most of the time	4	1.7%
6	Sorry, I have not been attending the lectures.	2	0.8%

Sort By Percentage

RI

Highlights some of the points

Certainly! Here are some highlighted points from the feedback:

1. **Audio Quality and Volume:**
   - 2 The lecture audio is quite bad
   - 27 Not sure if its a issue on my laptop (Windows) only, but the audio for the lectures are always very low
   - 77 Audio quality is so bad. can't listen through speaker
   - 96 Volume of the lecture recordings is too low.
2. **Professor's Teaching Style and Clarity:**
   - 6 Prof Rizki can explain the concepts very clearly and easy to understand
   - 11 Prof Rizki is a good lecturer and explains things clearly!
   - 32 You are very considerate. Honestly he cares a lot about whether we get the content of the lecture
   - 105 He is clear about the concepts and content that he is delivering
   - 138 Clear and concise
   - 177 Prof Rizki is very eloquent, his lectures are very well paced and easily understandable
   - 188 Professor Rizki is one of the best profs I have ever seen. He is able to explain concepts clearly and it is a joy listening to his lectures.

3. **Slide Content and Presentation:**
   - 4 + for pronunciation and articulation - for slides as some "implied" information is not really implied
   - 24 Should look through slides before presenting. Sometimes there are mistakes in the slides
   - 33 may be the slides could have been annotated better
   - 55 Lectures are generally easy to follow
   - 145 explain materials very clearly, makes lectures interesting
4. **Specific Topics:**
   - 35 Nice pace of lectures, especially love the TED talk during the break to have a change of atmosphere.
   - 57 hes good at teaching
   - 61 I think Prof Rizki is very passionate about his field of expertise, which makes learning under him an enjoyable and uplifting experience.
   - 78 Very clear teaching
   - 110 Excellent!

These are the highlighted points that represent the key feedback themes.

Handpicked feedback

- Some topics need elaboration and more examples
  - E.g., alpha beta pruning
- Notations can be made clearer
- Regarding answering questions during lecture
  - “Having so many questions answered during the lecture is a bit distracting.”
  - “Lecture is quite good. However, sometimes he is too nice to answer all students' questions.”
  - “Thanks for taking time to answer the questions during lecture.”

3. How effective are the Tutorials in facilitating your learning?

** Required*

S/N	Multiple Choice Option	Count	Percentage
1	Tutorials are helpful for my learning	129	54.4%
2	Tutorials are okay	57	24.1%
3	Tutorials are brilliant. Every module at NUS should have them!	43	18.1%
4	I have no idea what's going on/I have no idea what the Tutor or my peers are saying most of the time	7	3.0%
5	They are a complete waste of my time...	1	0.4%
6	Sorry, I haven't been attending the tutorials.	0	0.0%

Sort By Percentage

1. Comment on the difficulty of the midterm.

** Required*

S/N	Multiple Choice Option	Count	Percentage
1	Somewhat hard	129	54.4%
2	Just right	69	29.1%
3	Way over your head	36	15.2%
4	Somewhat easy	2	0.8%
5	Too easy	1	0.4%

Sort By Percentage

2. Comment on time allocated for the midterm.

** Required*

S/N	Multiple Choice Option	Count	Percentage
1	Time is somewhat short	110	46.4%
2	Way too little. Too long, too little time.	93	39.2%
3	Time allocated is just nice	34	14.3%
4	Too much time, too little to do	0	0.0%
5	I can nap for an hour during the midterm and still finish every question	0	0.0%

Sort By Percentage

Materials

Recap

- Logistic Regression
  - Classification with Continuous Inputs
  - Cross-entropy Loss
  - Logistic Regression with Gradient Descent
- Logistic Regression: Challenges and Solutions
  - Logistic Regression with Many Attributes
  - Dealing with Non-Linear Decision Boundary
- Multi-class Classification
- (More) Performance Measure
  - Receiver Operating Characteristic (ROC)
  - Area under ROC (AUC)
- Model Evaluation & Selection
  - Bias & Variance
- Hyperparameter Tuning

ERRATA

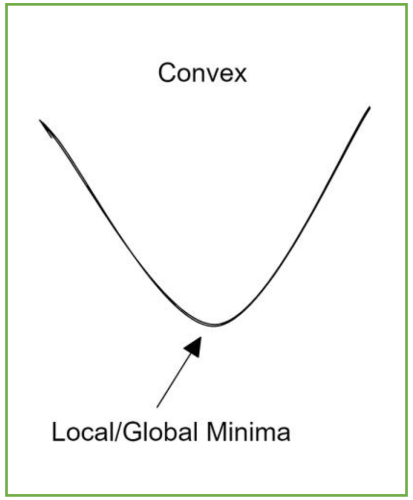
Logistic Regression with Cross-Entropy Loss

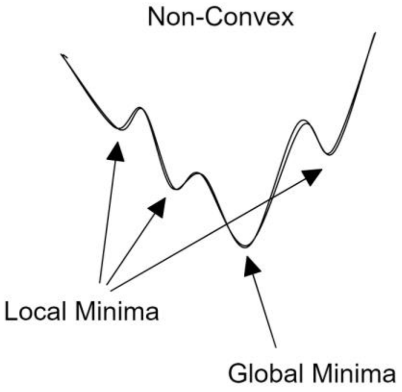
For a set of m examples $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$

we can compute the **binary cross entropy loss** as follows.

$$J_{BCE}(w) = \frac{1}{m}\sum_{i=1}^{m} BCE\left(y^{(i)}, h_w\left(x^{(i)}\right)\right)$$

$h_w(x) = \sigma(w_0 + w_1x_1 + w_2x_2)$ (Probability output)





$$BCE(y, \hat{y})$$
$$= -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$
$$= -y\log\left(\frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2)}}\right)$$
$$-(1-y)\log\left(1-\frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2)}}\right)$$

Linear! → convex

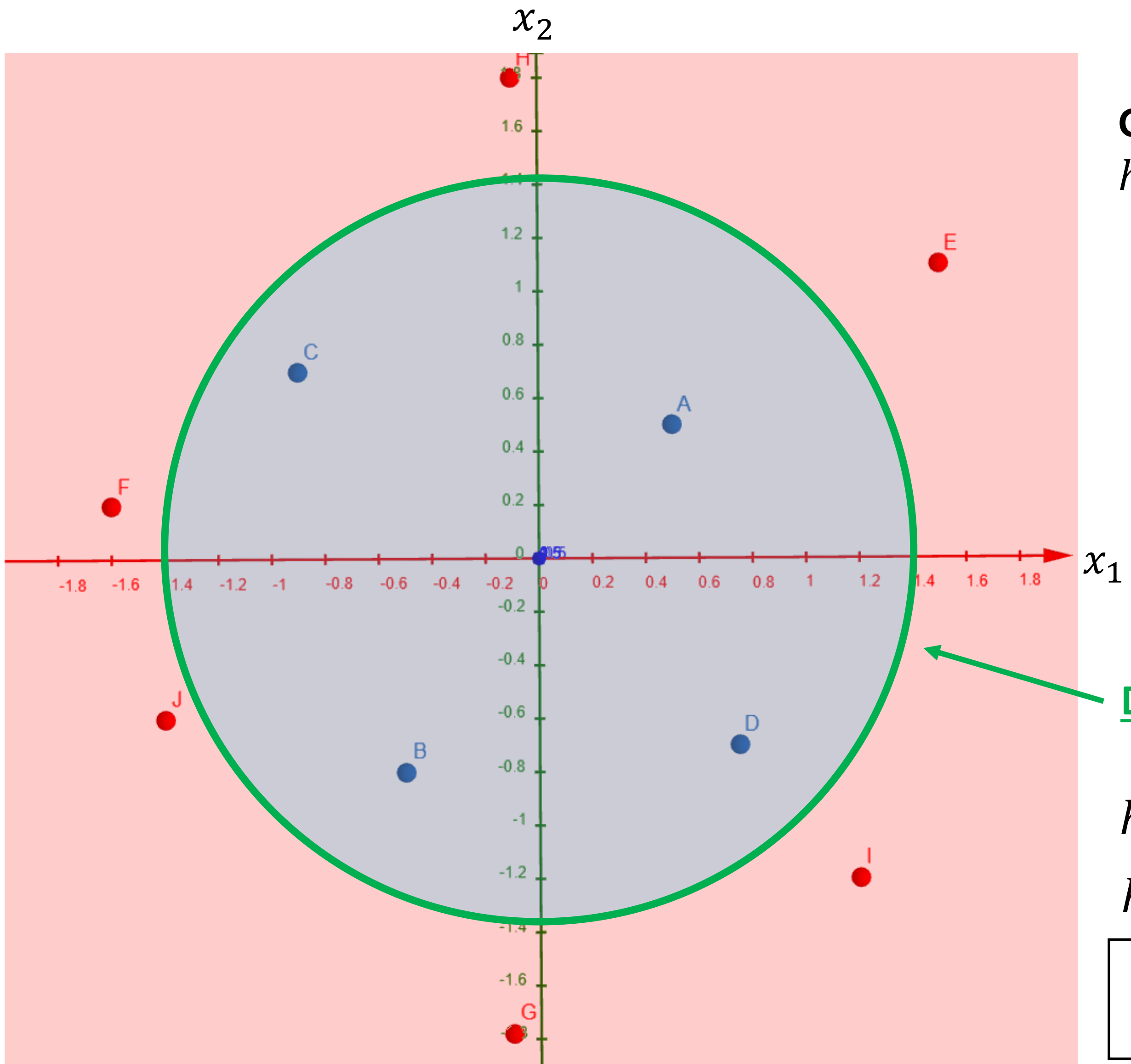
$\log(e^{a}) = ca$, where c is constant

Proof: Use second order definition of convexity

Show that the eigenvalues of this loss function's Hessian matrix are all always nonnegative.

ERRATA

Dealing with Non-Linear Decision Boundary



Generally:

$h_w(x) = w_0 + w_1 f_1 + w_2 f_2 + w_3 f_3 + \cdots + w_n f_n$

Transformed features:

$e.g., f_1 = x_1, f_2 = x_2, f_3 = x_1^2, f_4 = x_2^2$

Decision boundary

$$h_w(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2)$$

$$h_w(x) = \sigma(-2 + 0x_1 + 0x_2 + 1x_1^2 + 1x_2^2)$$

$$\boxed{\sigma(z) = \frac{1}{1 + e^{-z}}}$$

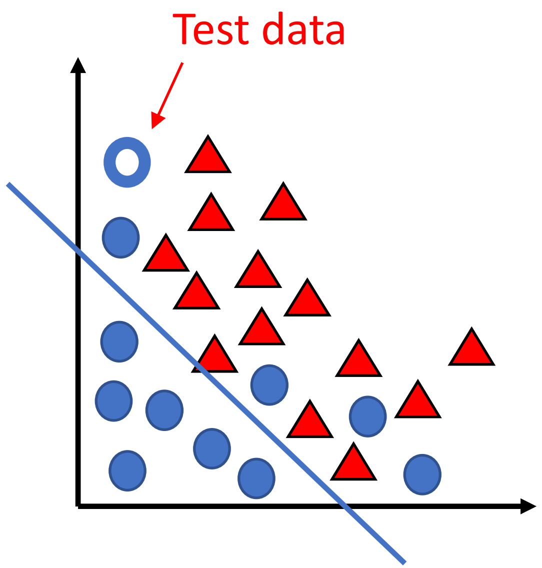
Outline

- The problem of overfitting
- Linear regression with regularization
- Logistic regression with regularization
- Support Vector Machines
  - Hard-margin SVM
  - Soft-margin SVM
- Kernel Methods & Kernel Trick

Outline

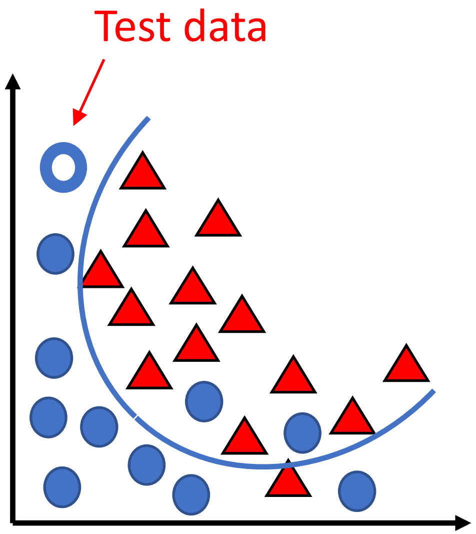
- **The problem of overfitting**
- Linear regression with regularization
- Logistic regression with regularization
- Support Vector Machines
    - Hard-margin SVM
    - Soft-margin SVM
- Kernel Methods & Kernel Trick

The problem of **overfitting**: logistic regression



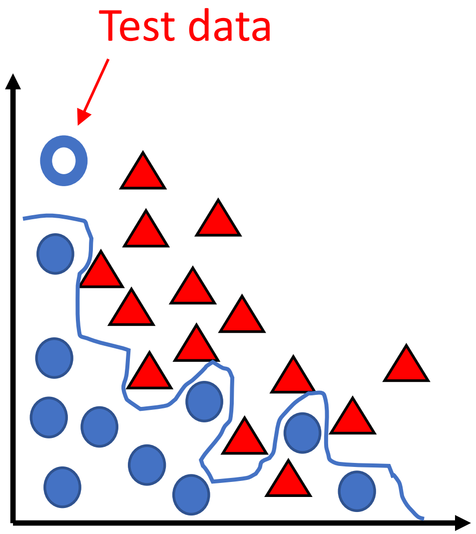
$$\sigma(w_0 + w_1 x_1 + w_2 x_2)$$

Underfitting



$$\sigma(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1 x_2)$$

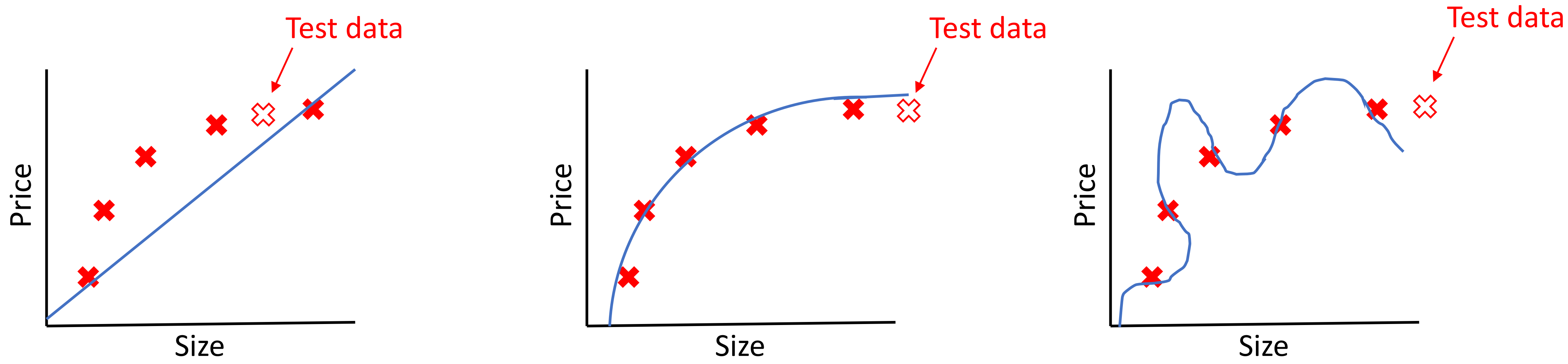
"Just Right"



$$\sigma(w_0 + w_1 x_1 + w_2 x_1^2 + w_3 x_1^2 x_2 + w_4 x_1^2 x_2^2 + \cdots)$$

Overfitting

The problem of **overfitting**: linear regression



$w_0 + w_1 x$

$w_0 + w_1 x + w_2 x^2$

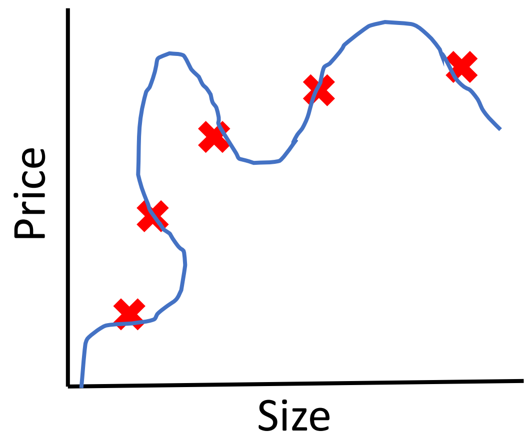
$w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$

Underfitting

“Just Right”

Overfitting

Addressing **overfitting**



1. Reduce the number of features
   - High degree polynomial → low degree

$$w_0 + w_1x + w_2x^2 + \cancel{w_3x^3 + w_4x^4}$$

2. Regularization
   - Keep all features, but reduce the magnitude w_j

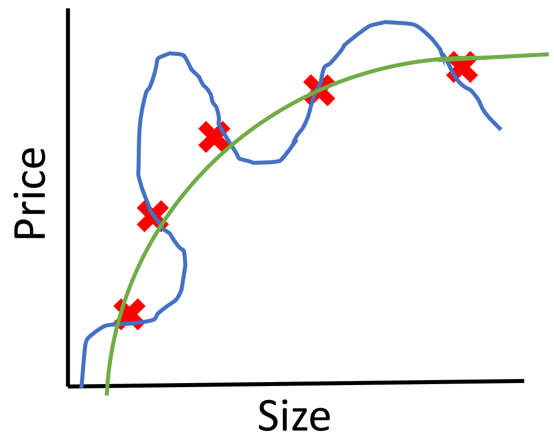
How to reduce?

$$w_0 + w_1x + w_2x^2 + w_3x^3 + w_4x^4$$

small

Occam's razor: simple is *usually* better

Addressing **overfitting**: regularization



Suppose that we want w_3 and w_4 to be really small

$$w_0 + w_1 x + w_2 x^2 + \cancel{w_3 x^3 + w_4 x^4}$$

small

How to know which ones to penalize?

Linear regression objective:

Don't know, **penalize all of them!**

$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right)^2 + 1000 w_3^2 + 1000 w_4^2\right]$$

Wants this to be small

If predicts correctly ~~then J(w) will be small~~

Not anymore!

Needs to be small! Profit!

Addressing **overfitting**: regularization

Suppose that we want w_3 and w_4 to be really small

$$w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

small

How to know which ones to penalize?

Linear regression objective:

Don't know, **penalize all of them!**

$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_w(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{n} w_i^2\right]$$

Outline

- The problem of overfitting
- **Linear regression with regularization**
- Logistic regression with regularization
- Support Vector Machines
    - Hard-margin SVM
    - Soft-margin SVM
- Kernel Methods & Kernel Trick

Linear Regression with Regularization

Hypothesis:

$$h_w(x): w^T x$$

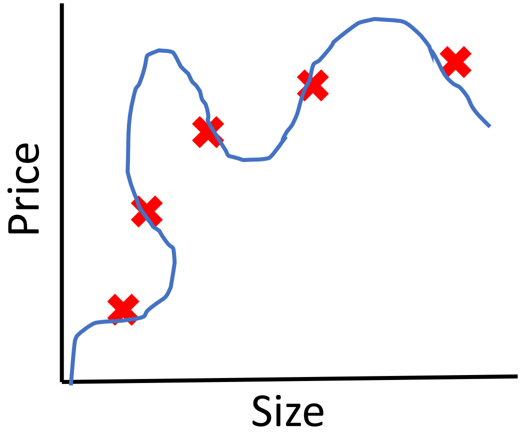
Cost function:

regularization parameter

$$J(w) = \frac{1}{2m}\left[\underbrace{\sum_{i=1}^{m}\left(h_w(x^{(i)}) - y^{(i)}\right)^2}_{\text{fitting data "well"}} + \underbrace{\lambda \sum_{i=1}^{n} w_i^2}_{\text{avoid "over-fitting"}}\right]$$

OG linear regression

Linear Regression with Regularization

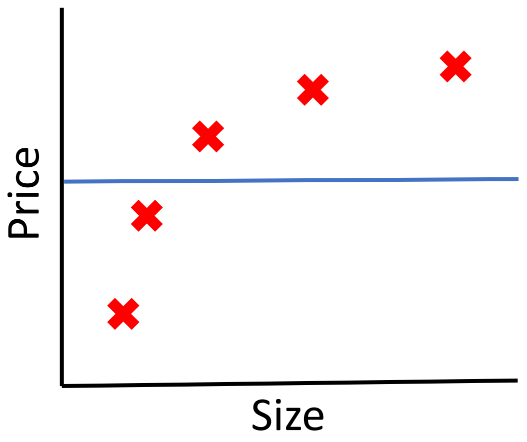


$w_0 + w_1x + w_2x^2 + w_3x^3 + w_4x^4$

$\lambda = 0$ $\lambda = 1000$

$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)})^2 + 0\sum_{i=1}^{n} w_i^2\right]$$

Linear Regression with Regularization

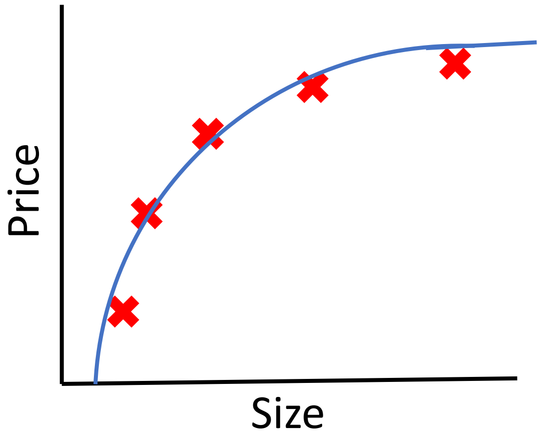


$$w_0 + \cancel{w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4}$$

$\lambda = 0$ $\lambda = 1000$

$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_w(x^{(i)}) - y^{(i)}\right)^2 + 1000\sum_{i=1}^{n} w_i^2\right]$$

Linear Regression with Regularization



$$w_0 + w_1 x + w_2 x^2 + \cancel{w_3 x^3 + w_4 x^4}$$

$\lambda = 0$ $\lambda = 1000$

$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_w(x^{(i)}) - y^{(i)}\right)^2 + 2\sum_{i=1}^{n} w_i^2\right]$$

Linear Regression with Regularization

Hypothesis:

$$h_w(x): w^T x$$

Cost Function:

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right)^2 \quad \boxed{+\frac{\lambda}{2m}\sum_{i=1}^{n} w_i^2}$$

Gradient Descent:

$$w_n := w_n - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right).x_n^{(i)} \boxed{-\frac{\lambda}{m}w_n}$$

Linear Regression w/ Regz: Optimization

$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{i=1}^{n} w_i^2\right]$$

Optimization goal: $\min_{w} J(w)$

1. Gradient Descent
2. Normal Equation

Linear Regression w/ Regz: Gradient Descent

$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{i=1}^{n} w_i^2\right]$$

$\frac{\partial J(w_0, w_{1,\ldots})}{\partial w_j}$

Optimization goal: $\min_{w} J(w)$

$\frac{\partial J(w_0, w_{1,\ldots})}{\partial w_j}$

Repeat {

$$w_0 := w_0 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)}).x_0^{(i)}$$

$$w_1 := w_1 - \alpha \frac{1}{m}\left[\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)}).x_1^{(i)} - \lambda w_1\right]$$

$$\vdots$$

$$w_n := w_n - \alpha \frac{1}{m}\left[\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)}).x_n^{(i)} - \lambda w_n\right]$$

}

Why does this work?

Linear Regression w/ Regz: Gradient Descent

$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_w(x^{(i)}) - y^{(i)}\right)^2 + \lambda\sum_{i=1}^{n} w_i^2\right]$$

Optimization goal: $\min_{w} J(w)$

These are usually small

Intuition: **shrink parameters!**

$$:= \left(1 - \frac{\alpha\lambda}{m}\right) w_n - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right).x_n^{(i)}$$

$$w_n := w_n - \alpha\frac{1}{m}\left[\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right).x_n^{(i)} - \lambda w_n\right]$$

slightly < 1

"regular" gradient descent

Why does this work?

Linear Regression w/ Regz: Gradient Descent

$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{i=1}^{n} w_i^2\right]$$

Optimization goal: $\min_{w} J(w)$

$$w_n := w_n - \alpha \frac{1}{m}\left[\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)}).\, x_n^{(i)} - \lambda w_n\right]$$

These are usually small

Intuition: **shrink parameters!**

$$:= \underbrace{\left(1 - \frac{\alpha\lambda}{m}\right)}_{\text{slightly} < 1} w_n - \underbrace{\alpha \frac{1}{m}\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)}).\, x_n^{(i)}}_{\text{"regular" gradient descent}}$$

Why does this work?

Linear Regression w/ Regz: Normal Equation

$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{i=1}^{n} w_i^2\right]$$

Optimization goal: $\min_w J(w)$

Do a bunch of math

(n+1) x (n+1)

m x (n+1)

m

$$w = (X^T X + \lambda \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix})^{-1} X^T Y$$

This works even if $X^T X$ is non-invertible if λ>0!

Outline

- The problem of overfitting
- Linear regression with regularization
- **Logistic regression with regularization**
- Support Vector Machines
  - Hard-margin SVM
  - Soft-margin SVM
- Kernel Methods & Kernel Trick

Logistic Regression with Regularization

Hypothesis:

$$h_w(x): \frac{1}{1+e^{-w^T x}}$$

Cost Function:

$$J(w) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_w\left(x^{(i)}\right) + \left(1-y^{(i)}\right)\ \log\left(1-h_w\left(x^{(i)}\right)\right) + \frac{\lambda}{2m}\sum_{i=1}^{n} w_i^2$$

Gradient Descent:

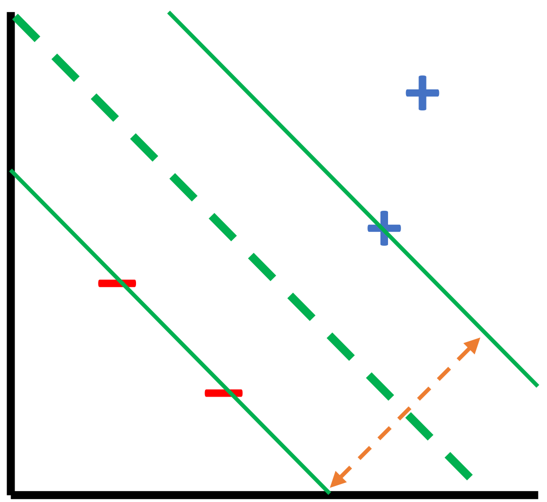
$$w_n := w_n - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right)-y^{(i)}\right).\,x_n^{(i)} - \alpha\frac{\lambda}{m}w_n$$

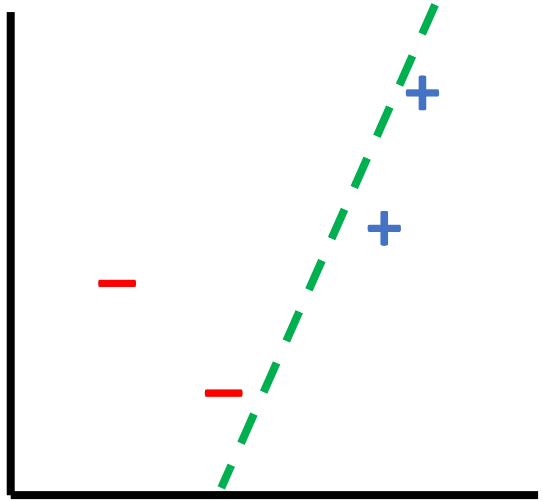
Outline

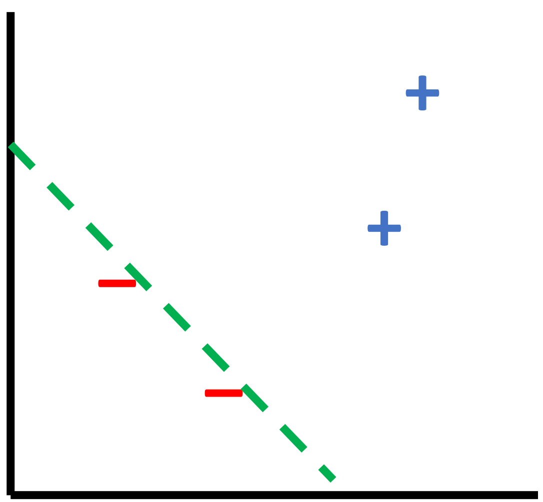
- The problem of overfitting
- Linear regression with regularization
- Logistic regression with regularization
- **Support Vector Machines**
  - Hard-margin SVM
  - Soft-margin SVM
- Kernel Methods & Kernel Trick

Decision Boundaries

Best



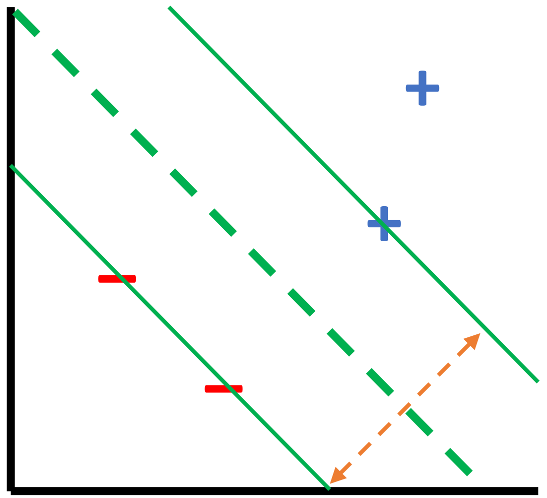




Maximize the *margin* between + and –

Support Vector Machines (SVM)

"Widest street approach"



Maximize the *margin* between + and –

"Maximize the width of the street"

How do we get a model that maximizes the margin?

1. Define the appropriate decision rule

$$w \cdot x + b \geq 0 \text{ then } +$$

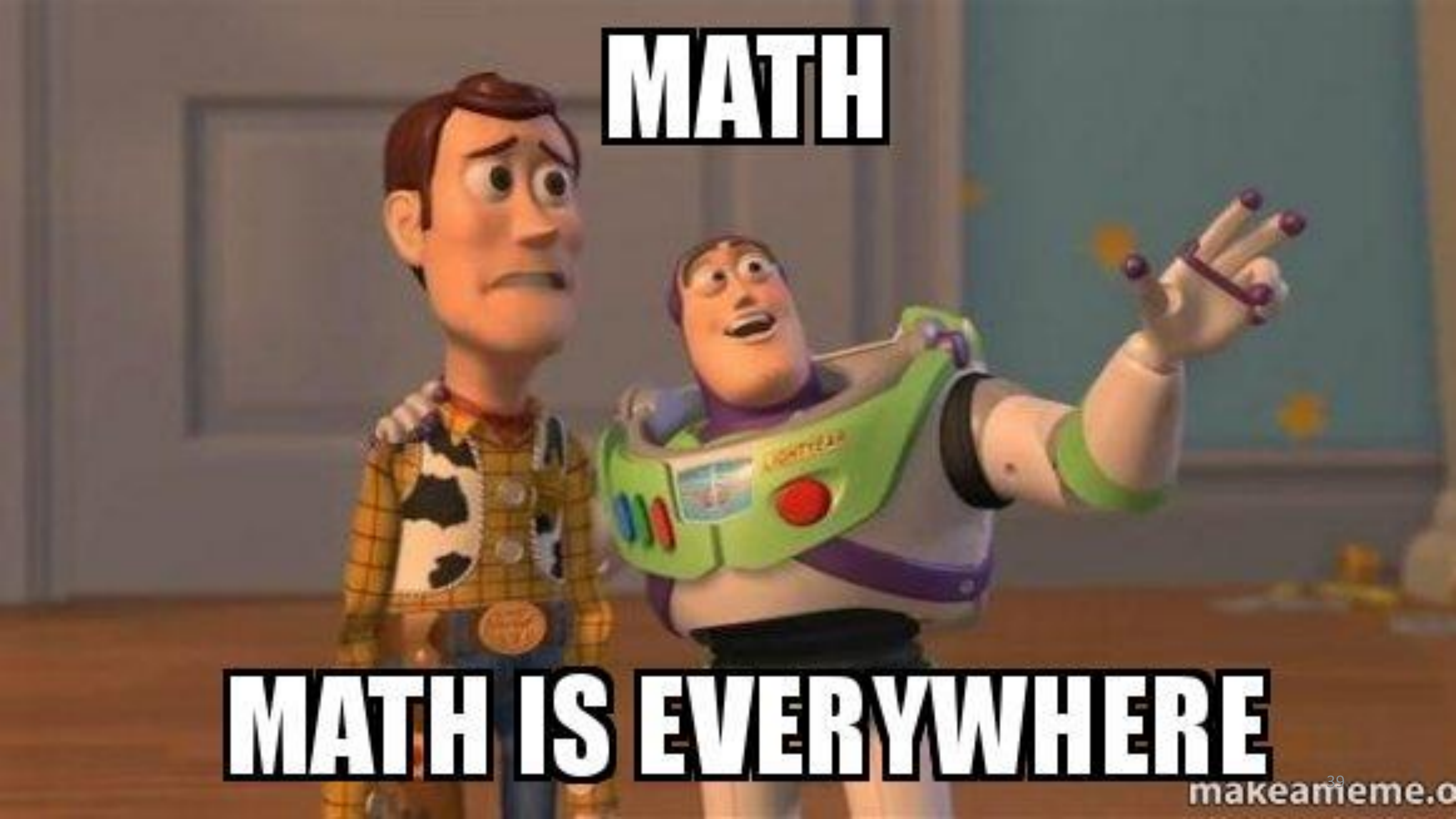
2. Find the equation of the margin

$$\text{margin} = \frac{2}{\|w\|}$$

3. Derive the objective that maximizes the margin

$$\max_{w} \frac{2}{\|w\|}$$

$$\text{s.t. } \bar{y}^{(i)}(w \cdot x^{(i)} + b) - 1 \geq 0$$

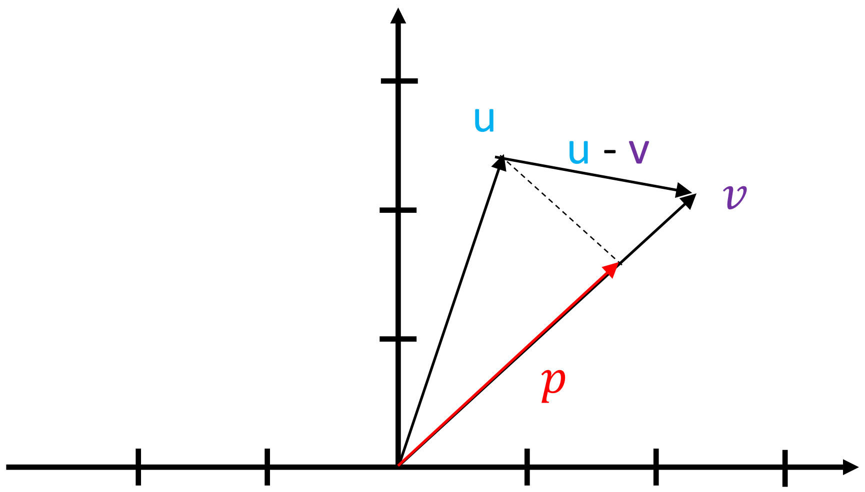

MATH
MATH IS EVERYWHERE

Background: Linear Algebra

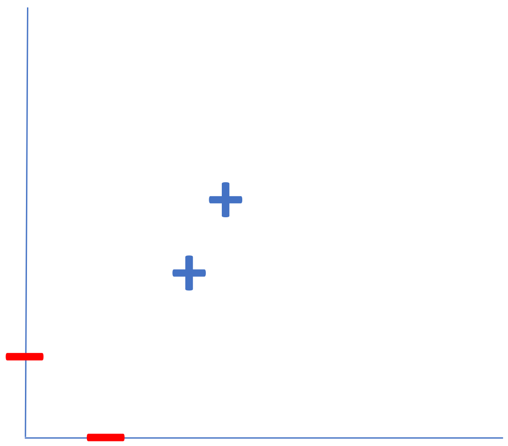
$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$

$||u|| = \sqrt{u_1^2 + u_2^2} \qquad \in R$

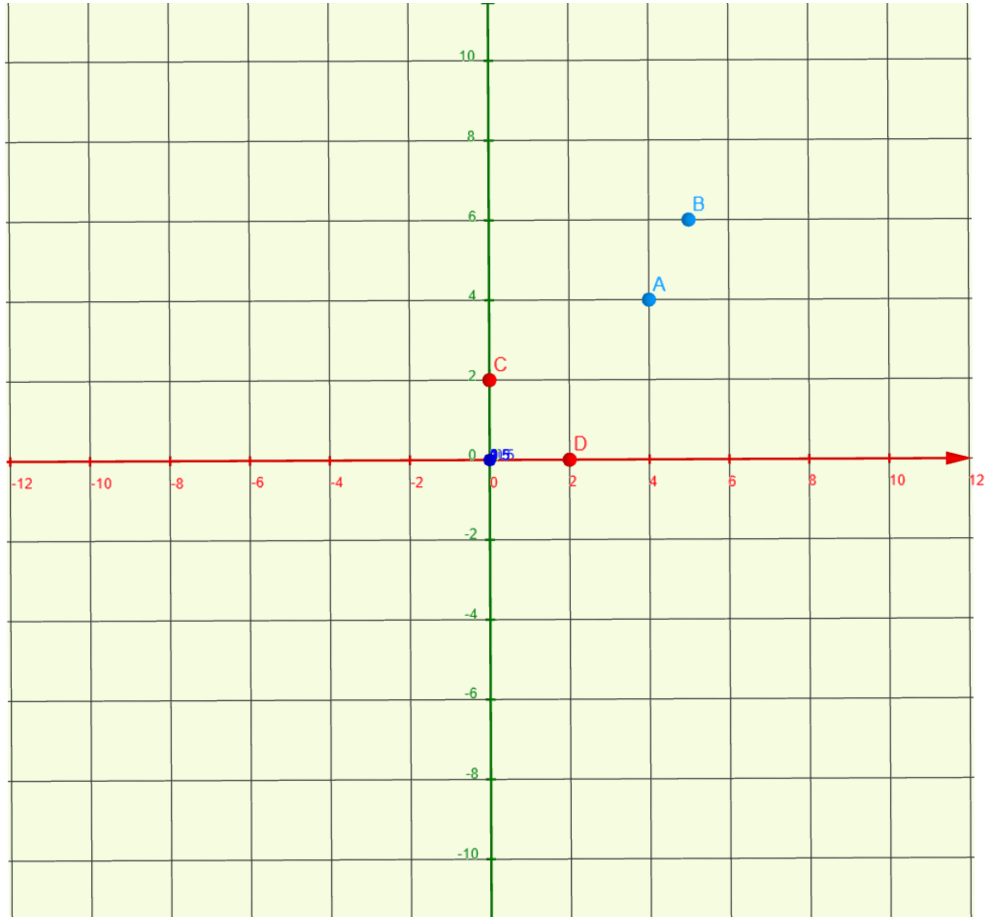
$u \cdot v = u^T v = p||v||, \qquad p \in R$



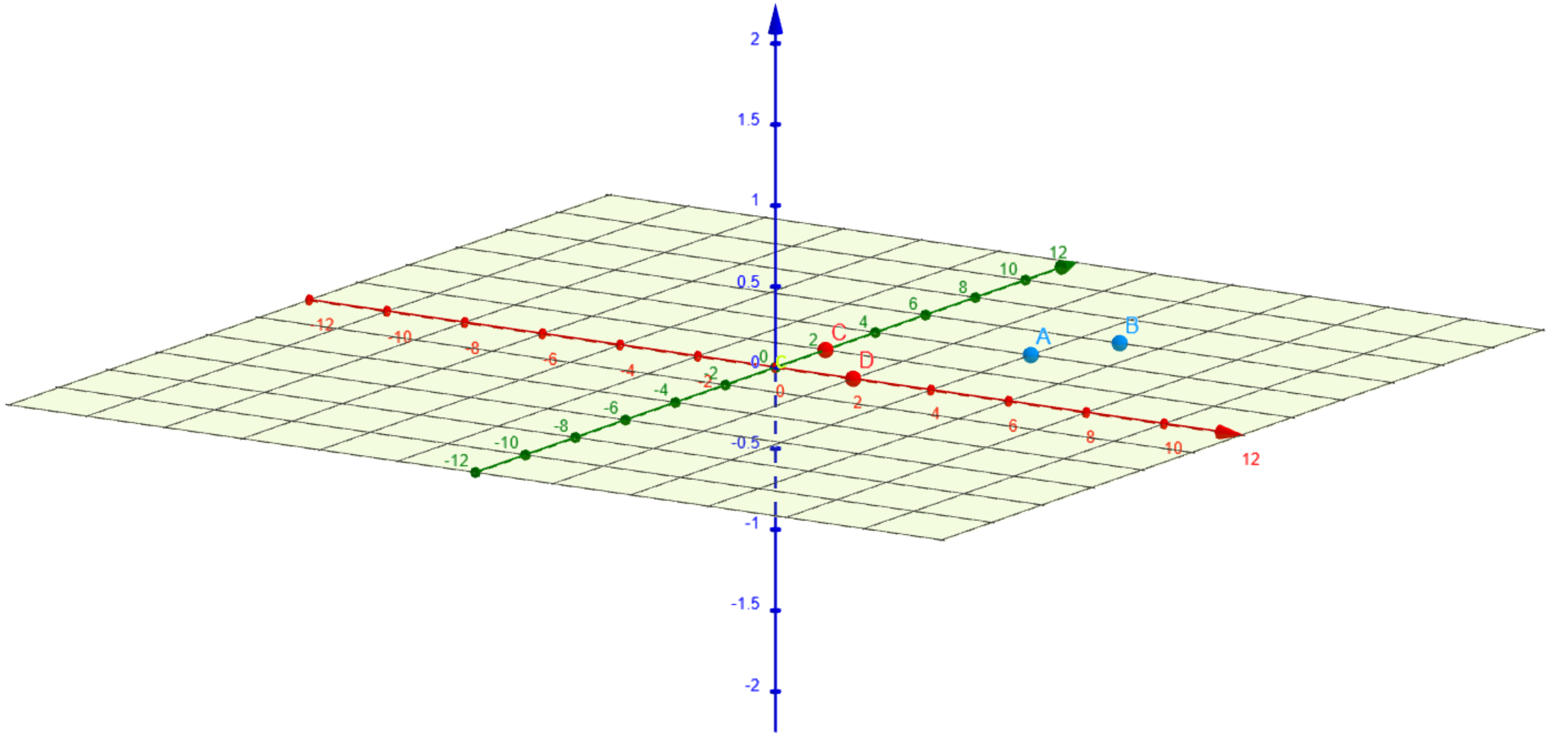
Support Vector Machines (SVM)



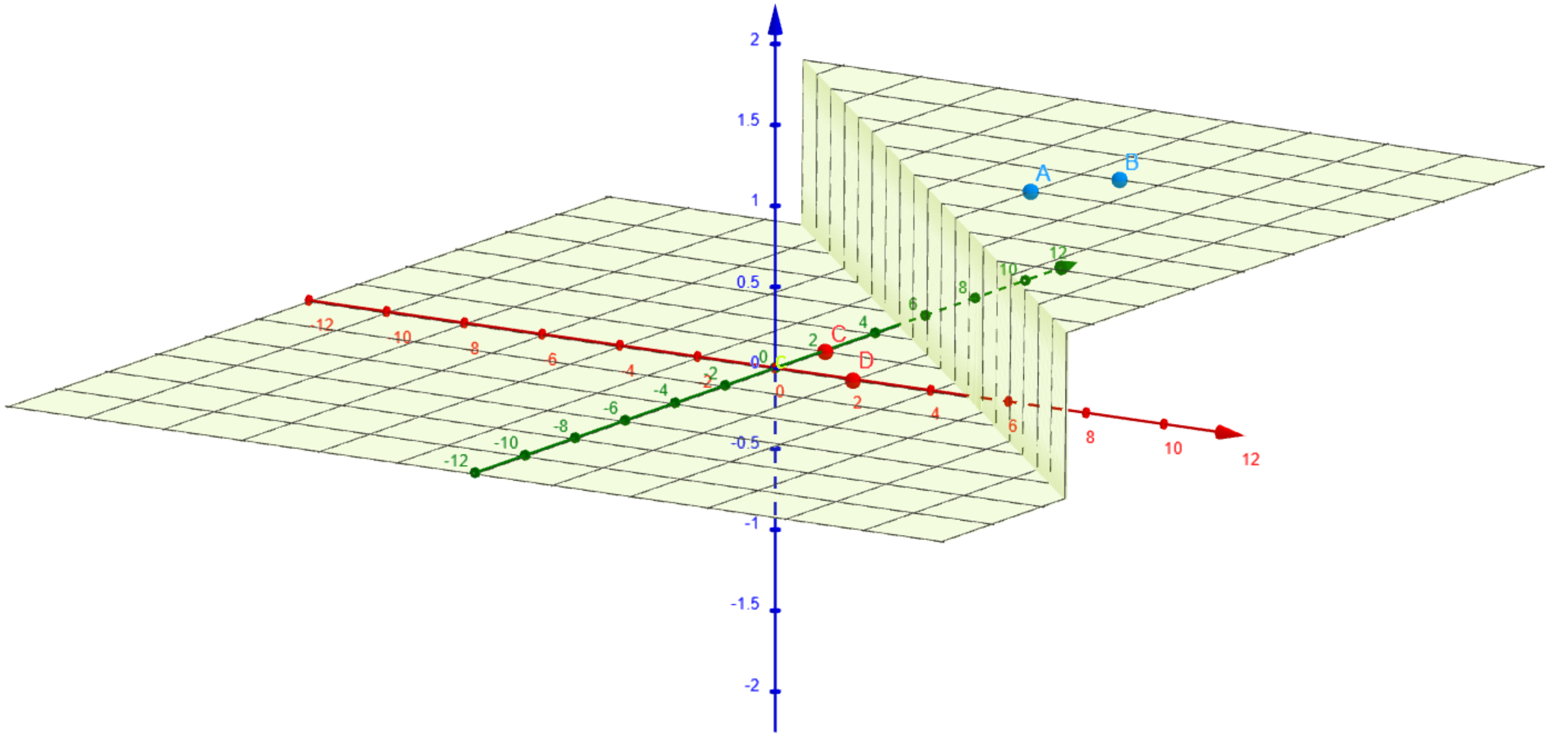
Support Vector Machines (SVM)



Support Vector Machines (SVM)



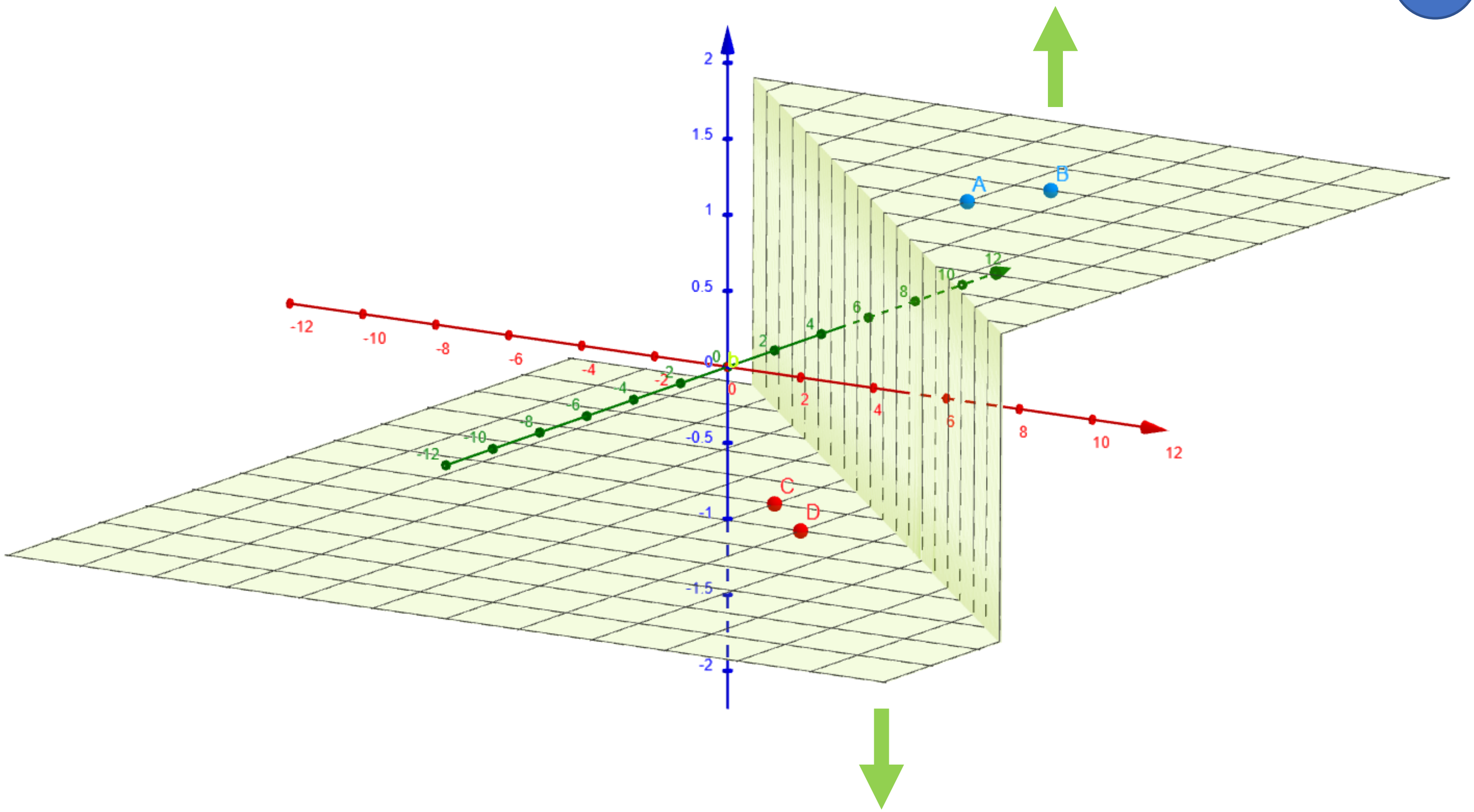
Support Vector Machines (SVM)



1 *Decision Rule:*

$$w \cdot x \geq c \text{ then } +$$
$$w \cdot x + b \geq 0 \text{ then } +$$

Support Vector Machines (SVM)



1 *Decision Rule:*

$$w \cdot x \geq c \text{ then } +$$

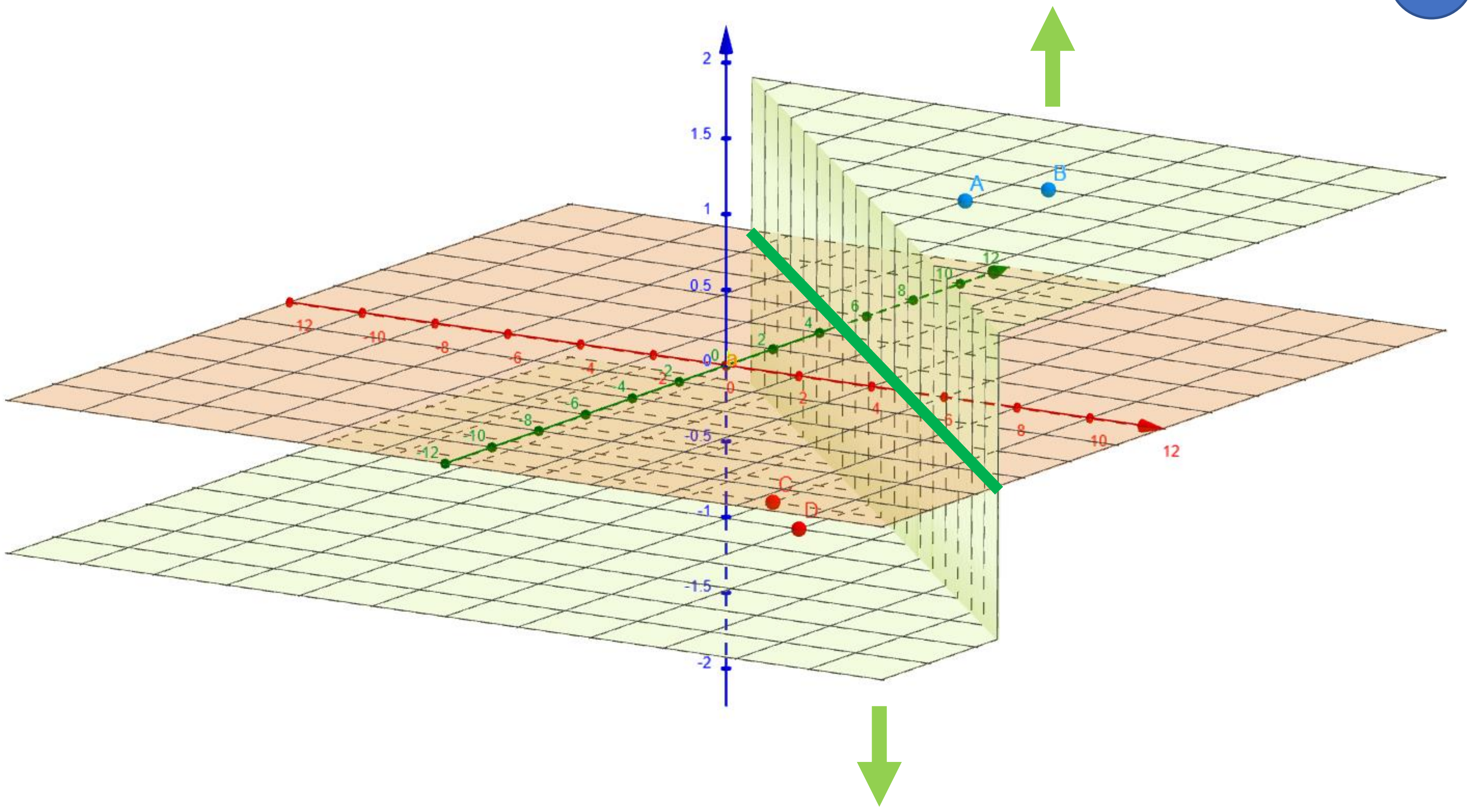
$$w \cdot x + b \geq 0 \text{ then } +$$

Add constraints:

$$w \cdot x^{+} + b \geq 1$$

$$w \cdot x^{-} + b \leq -1$$

Support Vector Machines (SVM)



1 *Decision Rule:*

$w \cdot x \geq c$ then +

$w \cdot x + b \geq 0$ then +

Add constraints:

$w \cdot x^{+} + b \geq 1$

$w \cdot x^{-} + b \leq -1$

Define:

$\bar{y}^{(i)} = +1$ for + samples

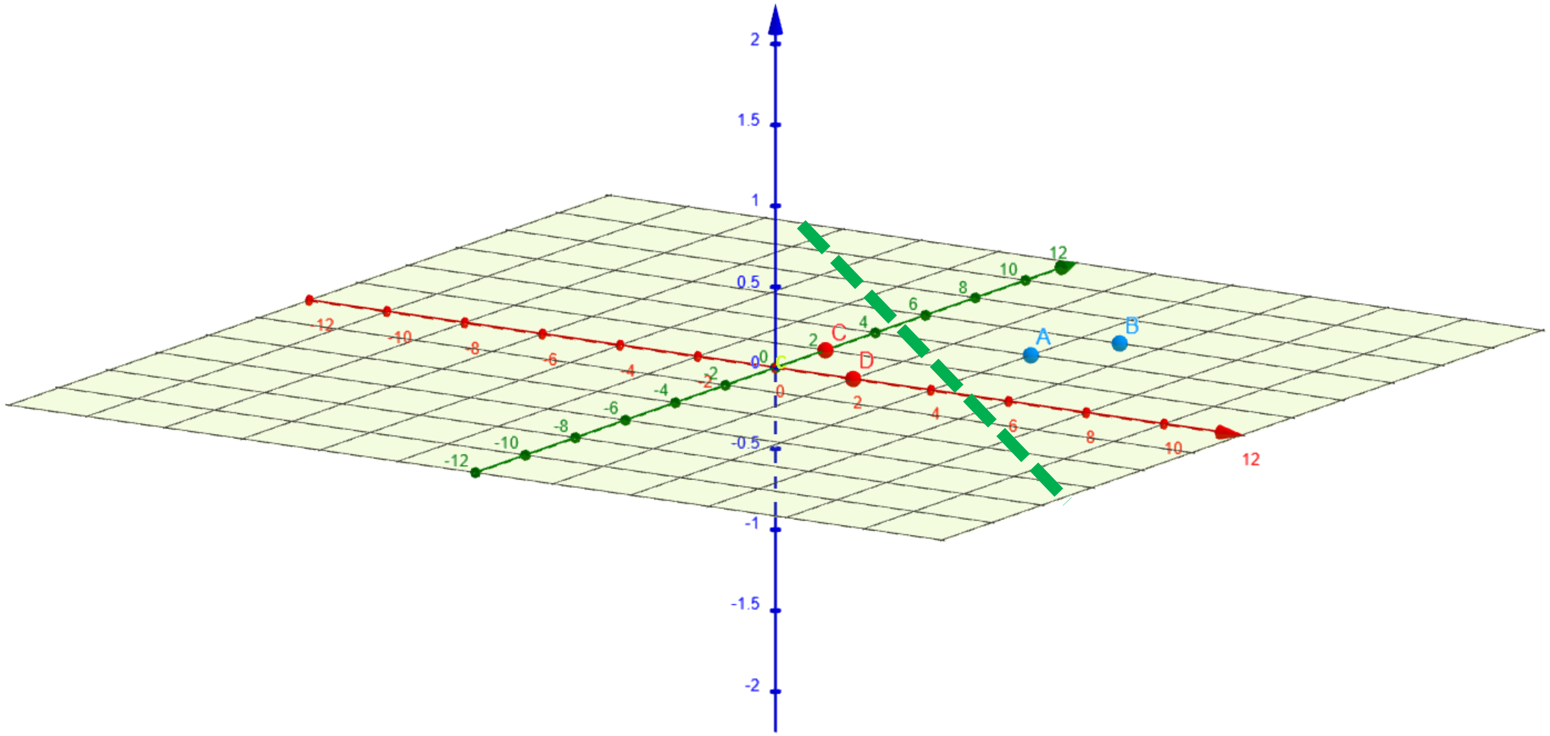
$= -1$ for − samples

$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) \geq 1$$

$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 \geq 0$$

$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 = 0$ for all $x^{(i)}$ in margin

Support Vector Machines (SVM)



1 *Decision Rule:*

$$w \cdot x \geq c \text{ then } +$$
$$w \cdot x + b \geq 0 \text{ then } +$$

Add constraints:

$$w \cdot x^{+} + b \geq 1$$
$$w \cdot x^{-} + b \leq -1$$

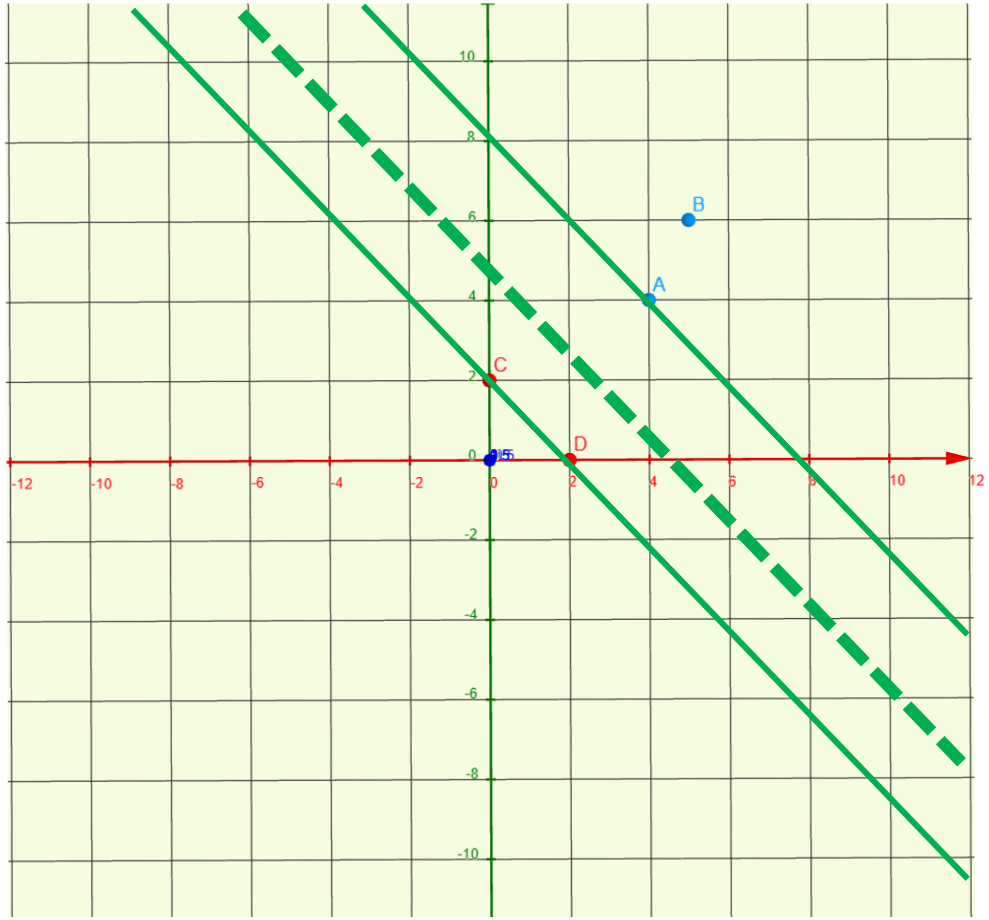
Define:

$$\bar{y}^{(i)} = +1 \text{ for } + \text{ samples}$$
$$= -1 \text{ for } - \text{ samples}$$

$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) \geq 1$$
$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 \geq 0$$

$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 = 0 \text{ for all } x^{(i)} \text{ in margin}$$

Support Vector Machines (SVM)



1 *Decision Rule:*

$$w \cdot x \geq c \text{ then } +$$
$$w \cdot x + b \geq 0 \text{ then } +$$

Add constraints:

$$w \cdot x^{+} + b \geq 1$$
$$w \cdot x^{-} + b \leq -1$$

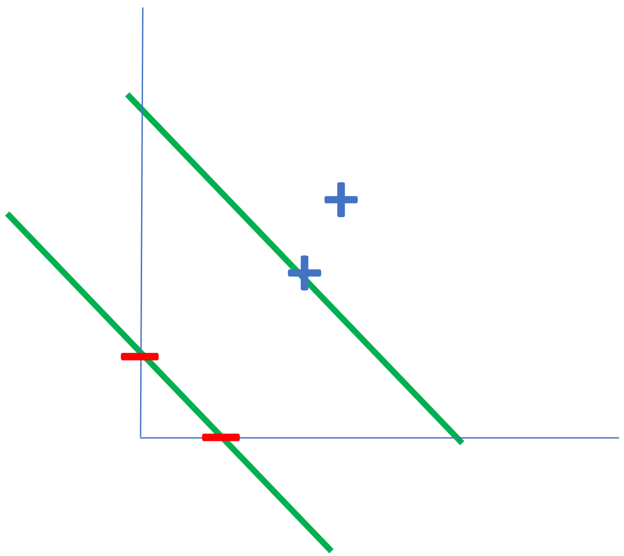
Define:

$$\bar{y}^{(i)} = +1 \text{ for } + \text{ samples}$$
$$= -1 \text{ for } - \text{ samples}$$

$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) \geq 1$$
$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 \geq 0$$

$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 = 0 \text{ for all } x^{(i)} \text{ in margin}$$

Support Vector Machines (SVM)



1 *Decision Rule:*

$$w \cdot x \geq c \text{ then } +$$
$$w \cdot x + b \geq 0 \text{ then } +$$

Add constraints:

$$w \cdot x^{+} + b \geq 1$$
$$w \cdot x^{-} + b \leq -1$$

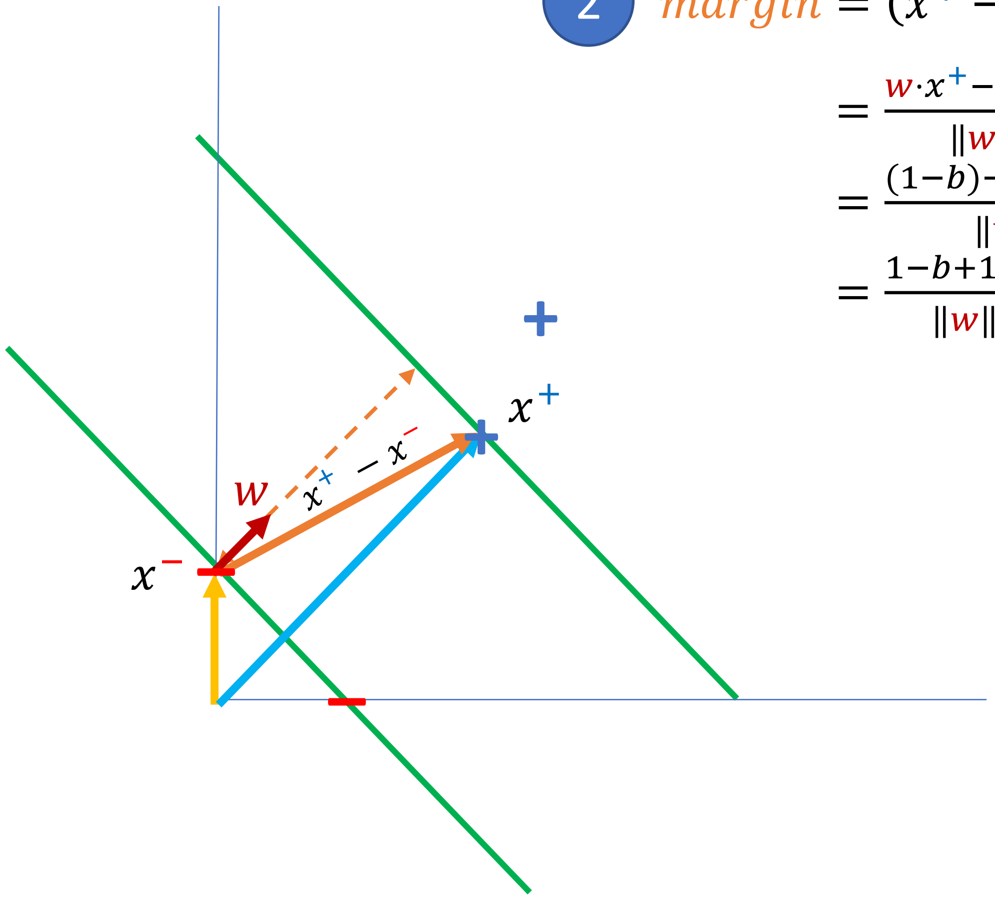
Define:

$$\bar{y}^{(i)} = +1 \text{ for } + \text{ samples}$$
$$= -1 \text{ for } - \text{ samples}$$

$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) \geq 1$$
$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 \geq 0$$

$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 = 0$ for all $x^{(i)}$ in margin

Support Vector Machines (SVM)



2 $margin = (x^+ - x^-) \cdot \dfrac{w}{\|w\|}$

$$= \frac{w \cdot x^+ - w \cdot x^-}{\|w\|}$$

$$= \frac{(1-b)-(-1-b)}{\|w\|}$$

$$= \frac{1-b+1+b}{\|w\|} = \frac{2}{\|w\|}$$

1 *Decision Rule:*

$$w \cdot x \geq c \text{ then } +$$

$$w \cdot x + b \geq 0 \text{ then } +$$

Add constraints:

$$w \cdot x^+ + b \geq 1$$

$$w \cdot x^- + b \leq -1$$

Define:

$$\bar{y}^{(i)} = +1 \text{ for } + \text{ samples}$$

$$= -1 \text{ for } - \text{ samples}$$

$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) \geq 1$$

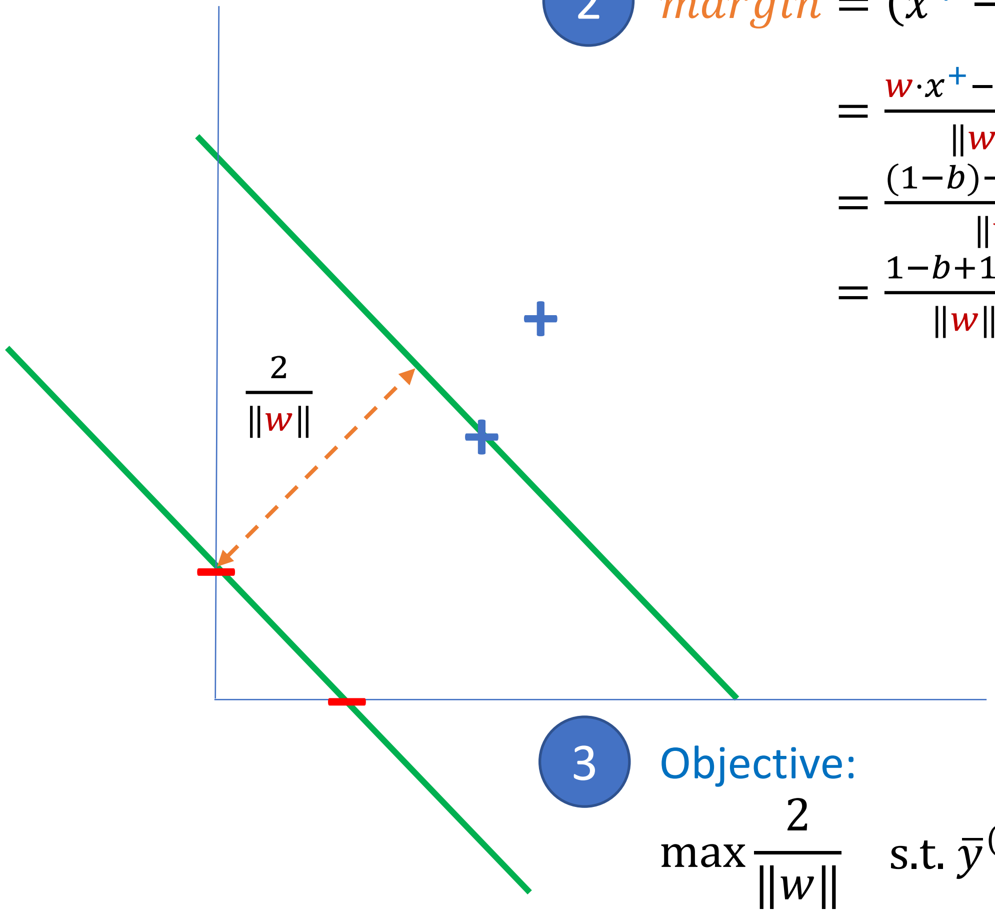
$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 \geq 0$$

$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 = 0 \text{ for all } x^{(i)} \text{ in margin}$$

$$\left(w \cdot x^{(i)} + b\right) = \frac{1}{\bar{y}^{(i)}}$$

$$\boxed{\begin{aligned} w \cdot x^+ + b &= +1 \\ w \cdot x^- + b &= -1 \end{aligned}} \quad \boxed{\begin{aligned} w \cdot x^+ &= 1 - b \\ w \cdot x^- &= -1 - b \end{aligned}}$$

Support Vector Machines (SVM)



2 $margin = (x^{+} - x^{-}) \cdot \frac{w}{\|w\|}$

$$= \frac{w \cdot x^{+} - w \cdot x^{-}}{\|w\|}$$

$$= \frac{(1-b)-(-1-b)}{\|w\|}$$

$$= \frac{1-b+1+b}{\|w\|} = \frac{2}{\|w\|}$$

1 *Decision Rule:*

$w \cdot x \geq c$ then +

$w \cdot x + b \geq 0$ then +

Add constraints:

$w \cdot x^{+} + b \geq 1$

$w \cdot x^{-} + b \leq -1$

Define:

$\bar{y}^{(i)} = +1$ for + samples

$= -1$ for − samples

$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) \geq 1$$

$$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 \geq 0$$

$\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 = 0$ for all $x^{(i)}$ in margin

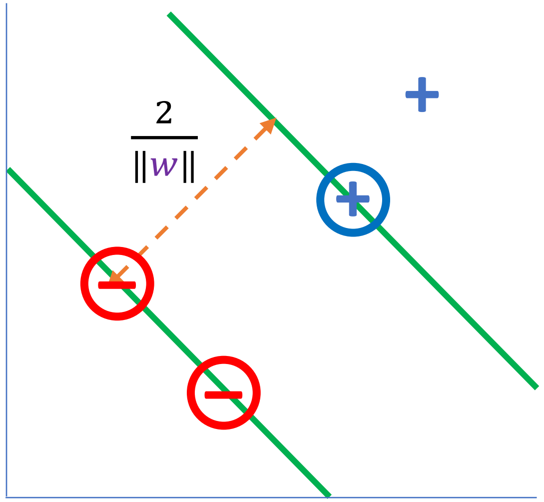
3 Objective:

$$\max \frac{2}{\|w\|} \quad \text{s.t. } \bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 \geq 0$$

"Maximize margin" "Classify correctly"

Support Vector Machines (SVM)

3



Objective:

$$\max \frac{2}{\|w\|} \rightarrow \max \frac{1}{\|w\|} \rightarrow \min \|w\| \rightarrow \min \frac{1}{2}\|w\|^2$$ "Maximize gap"

$$\text{s.t. } \bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 \geq 0$$ "Classify correctly"

Objective (Dual):

$$L(w,\alpha) = \frac{1}{2}\|w\|^2 - \sum_i \alpha^{(i)}\left[\bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1\right], \forall_i\, \alpha^{(i)} \geq 0$$

$$\frac{\partial L(w,\alpha)}{\partial w} = w - \sum_i \alpha^{(i)}\bar{y}^{(i)}x^{(i)} = 0 \quad \Rightarrow \quad \boxed{w = \sum_i \alpha^{(i)}\bar{y}^{(i)}x^{(i)}}$$

$$\frac{\partial L(w,\alpha)}{\partial b} = \sum_i \alpha^{(i)}\bar{y}^{(i)} = 0$$

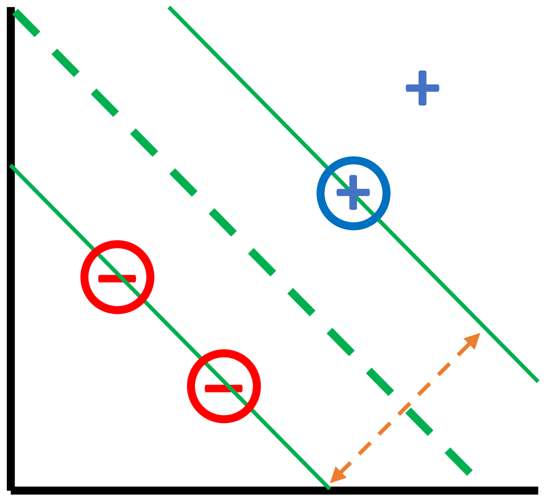
Samples with non-zero $\alpha^{(i)}$= support vectors

… a few math later …

Maximize →
$$L(\alpha) = \sum_i \alpha^{(i)} - \frac{1}{2}\sum_i\sum_j \alpha^{(i)}\alpha^{(j)}\bar{y}^{(i)}\bar{y}^{(j)}x^{(i)} \cdot x^{(j)}$$

Hard-Margin

Support Vector Machines (SVM)



How do we get a model that maximizes the margin?

1. Define the appropriate decision rule

$$w \cdot x + b \geq 0 \text{ then } +$$

2. Find the equation of the margin

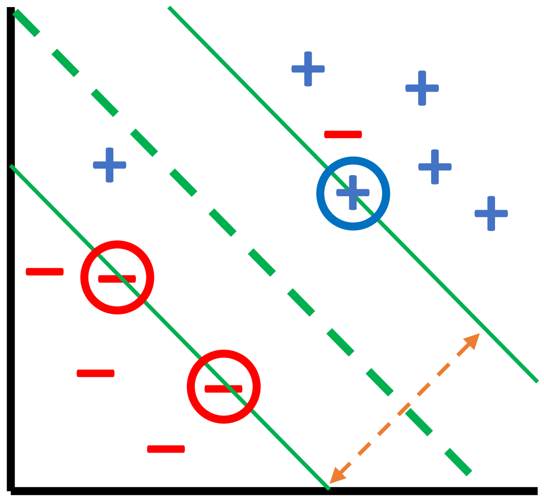
$$\text{margin} = \frac{2}{\|w\|}$$

3. Derive the objective that maximizes the margin

$$\min \frac{1}{2} \|w\|^2$$
$$\text{s.t. } \bar{y}^{(i)}(w \cdot x^{(i)} + b) - 1 \geq 0$$

…or…

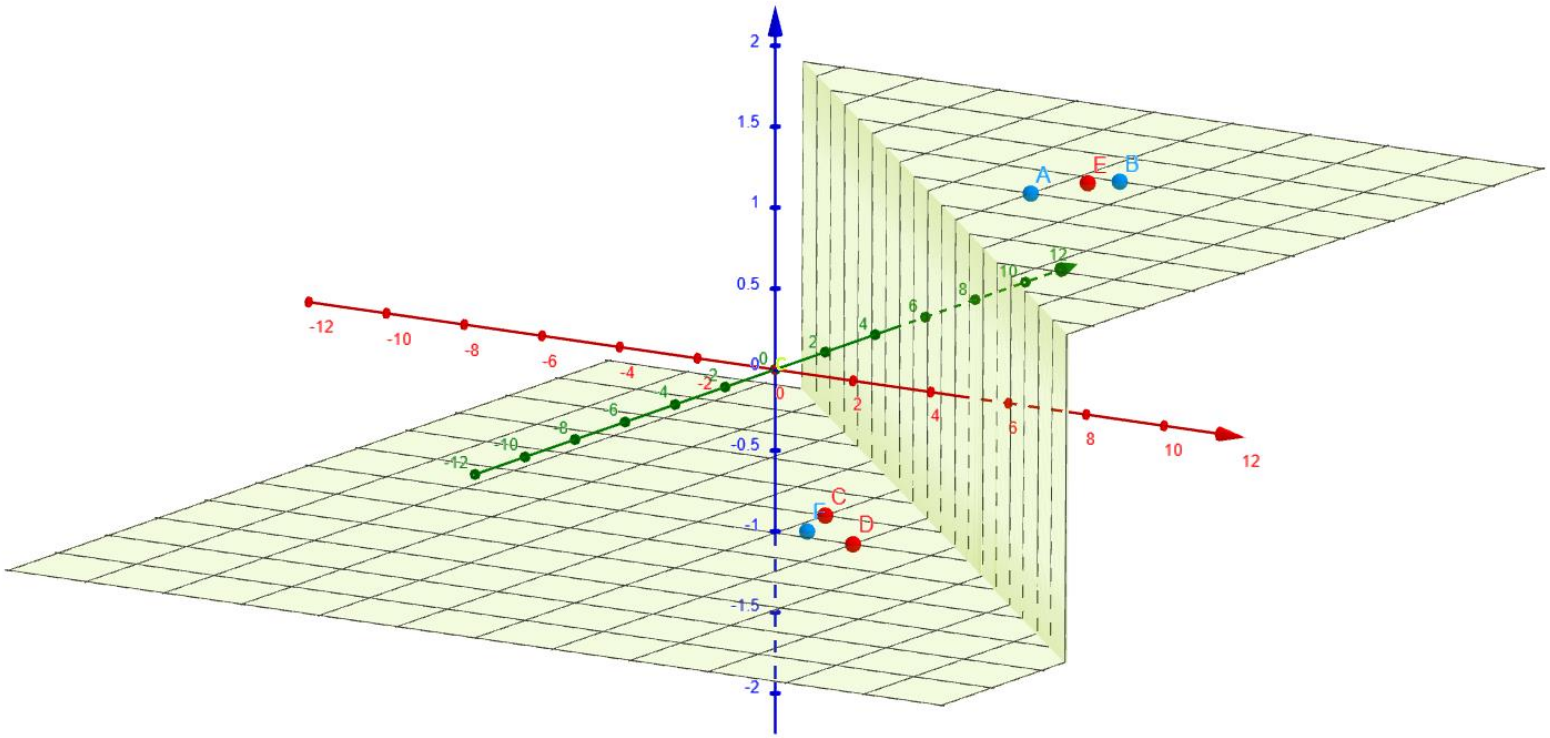
$$\max_{a} \sum_{i} \alpha^{(i)} - \frac{1}{2} \sum_{i} \sum_{j} \alpha^{(i)} \alpha^{(j)} \bar{y}^{(i)} \bar{y}^{(j)} x^{(i)} \cdot x^{(j)}$$



What if the data is not **linearly-separable**?

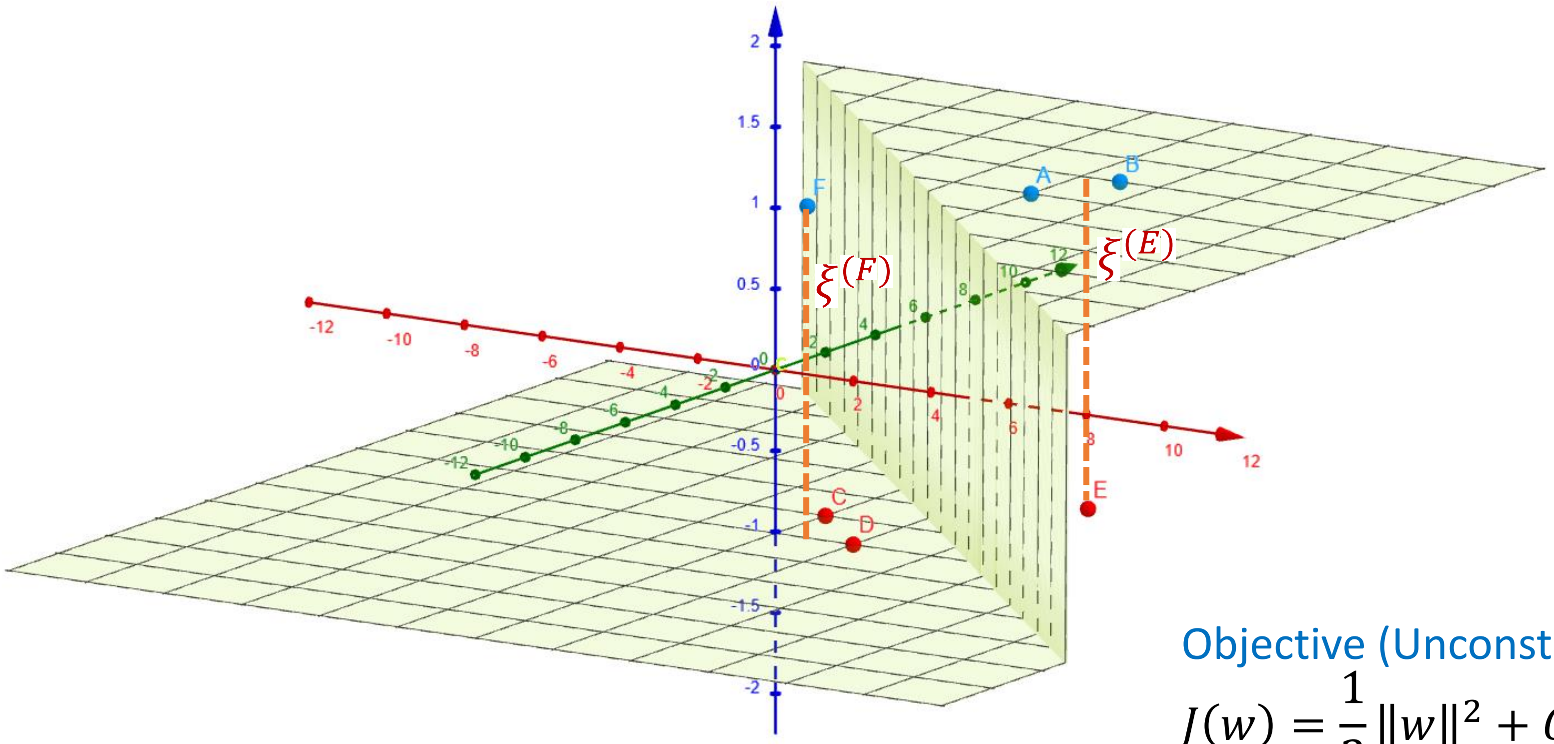
Soft-Margin

Support Vector Machines (SVM)



Soft-Margin

Support Vector Machines (SVM)



Introduce slack variables:

$$w \cdot x^{+} + b \geq 1 - \xi^{(i)}$$
$$w \cdot x^{-} + b \leq -1 + \xi^{(i)}$$

Objective:

$$\min \frac{1}{2}\|w\|^2 + C \sum_i \xi^{(i)}$$

Maximize gap and allow slack
Higher C → less slack

$$\text{s.t. } \bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right) - 1 \geq 0 - \xi^{(i)}$$
$$\forall_i \; \xi^{(i)} \geq 0$$

Classify correctly
but allow misclassifications

$$\boxed{\xi^{(i)} \geq 1 - \bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right)}$$

Objective (Unconstrained):

$$J(w) = \frac{1}{2}\|w\|^2 + C \sum_i \max\{0, 1 - \bar{y}^{(i)}\left(w \cdot x^{(i)} + b\right)\}$$

Image: https://jhui.github.io/2017/01/15/Machine-learning-regression-and-logistic-regression/

Soft-Margin

Support Vector Machines (SVM)

Recover the standard notation: $b = w_0$

Hypothesis:

$$h_w(x) = \begin{cases} 1, & \text{if } \ w^T x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Objective (Unconstrained):

$\frac{1}{2}\|w\|^2$

$$J(w) = \frac{1}{2}\sum_{i=1}^{n} w_i^2 + C\sum_{i} \max\{0, 1 - \bar{y}^{(i)}\left(w^T x^{(i)}\right)\}$$

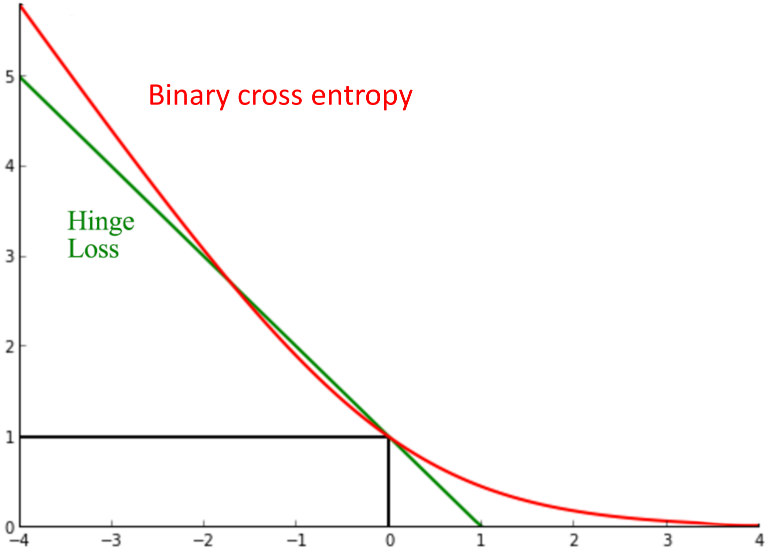
$$= C\sum_{i} \max\{0, 1 - \bar{y}^{(i)}\left(w^T x^{(i)}\right)\} + \frac{1}{2}\sum_{i=1}^{n} w_i^2$$

$$= C\sum_{i=1}^{m} y^{(i)} cost_1\left(w^T x^{(i)}\right) + \left(1 - y^{(i)}\right) cost_0\left(w^T x^{(i)}\right) + \frac{1}{2}\sum_{i=1}^{n} w_i^2$$

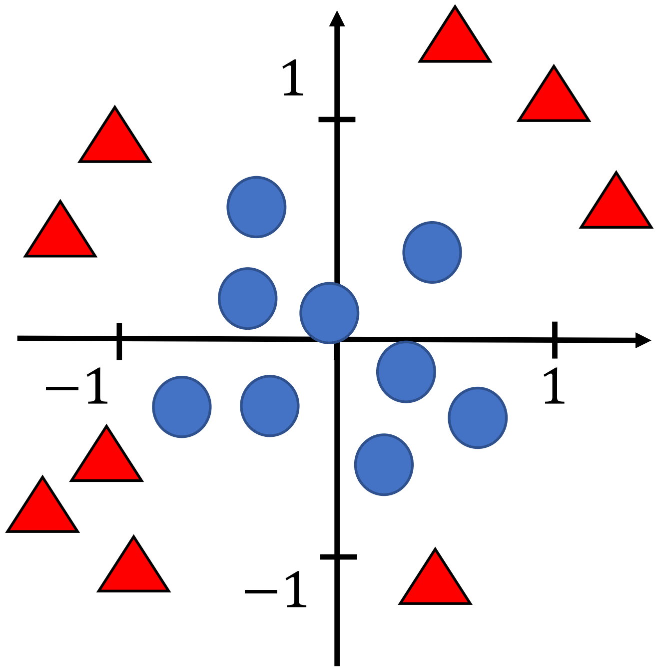
Soft-margin SVM

$$J(w) = \frac{1}{m}\sum_{i=1}^{m} y^{(i)}\left(-\log\left(h_w\left(x^{(i)}\right)\right)\right) + \left(1 - y^{(i)}\right)\left(-\log\left(1 - h_w\left(x^{(i)}\right)\right)\right) + \frac{\lambda}{2m}\sum_{i=1}^{n} w_i^2$$

Logistic Regression with regularization



$$cost_1(z) = \max\{0, 1 - z\}$$
$$cost_0(z) = \max\{0, 1 + z\}$$

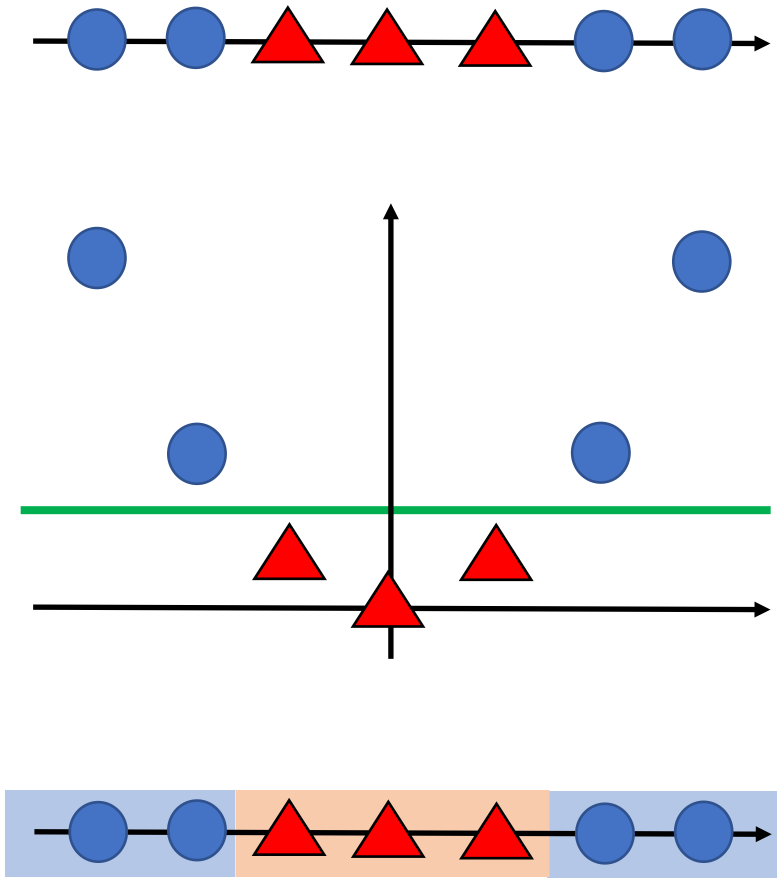


What if the data is truly <u>not</u> **linearly-separable**?

Outline

- The problem of overfitting
- Linear regression with regularization
- Logistic regression with regularization
- Support Vector Machines
  - Hard-margin SVM
  - Soft-margin SVM
- **Kernel Methods & Kernel Trick**

Handling **non-linear decision boundary (1D)**



$$\phi(x) = [x]^T$$

$$\boxed{w_0 + w_1 x \geq 0 \text{ then } +}$$

FAIL

$$\phi(x) = [x, x^2]^T$$

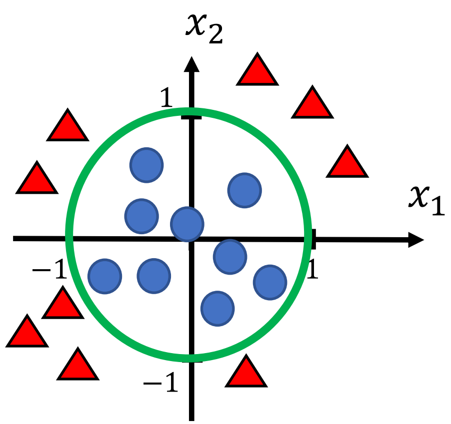
$$\boxed{w_0 + w_1 x + w_2 x^2 \geq 0 \text{ then } +}$$

SUCCESS

$$\boxed{w^T \phi(x) \geq 0 \text{ then } +}$$

Feature map

Handling **non-linear decision boundary (2D)**

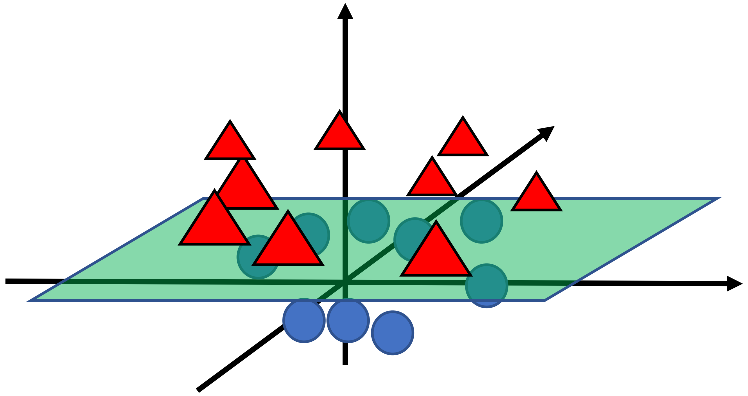


$\phi(x) = [x_1, x_2]^T$

$\boxed{w^T\phi(x) \geq 0 \text{ then } +}$

$\boxed{w_0 + w_1x_1 + w_2x_2 \geq 0 \text{ then } +}$

FAIL



$\phi(x) = [x_1, x_2, x_1x_2, x_1^2, x_2^2, \ldots]^T$

$\boxed{w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 + \cdots \geq 0 \text{ then } +}$

SUCCESS

ϕ can produce a huge number of features!
Not scalable!

Kernels

Polynomial degree 1:

$$K(u,v) = \phi(u)\cdot\phi(v) = [\mathrm{u}_1, \mathrm{u}_2]^T \cdot [v_1, v_2]^T = u_1 v_1 + u_2 v_2 = u \cdot v$$

Polynomial degree 2:

$$K(u,v) = \phi(u)\cdot\phi(v) = \left[\mathrm{u}_1^2, \sqrt{2}u_1u_2, u_2^2\right]^T \cdot \left[v_1^2, \sqrt{2}v_1v, v_2^2\right]^T = \cdots = (u\cdot v)^2$$

d=6, n=100, about 1.6 billion terms!

Polynomial degree d (n^d terms):

$$K(u,v) = \phi(u)\cdot\phi(v) = (u\cdot v)^d$$

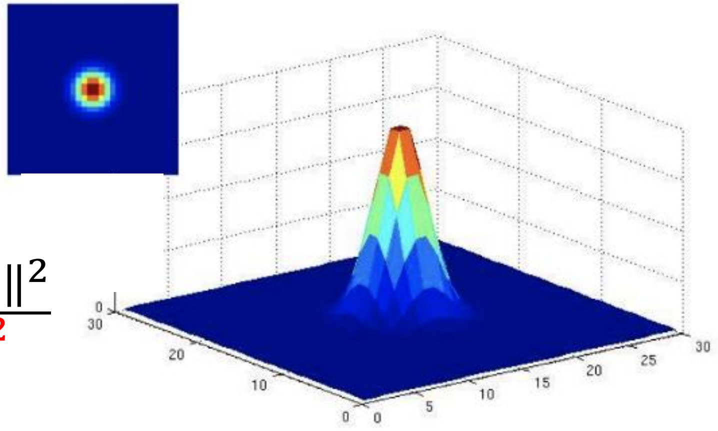
Kernel

Kernels

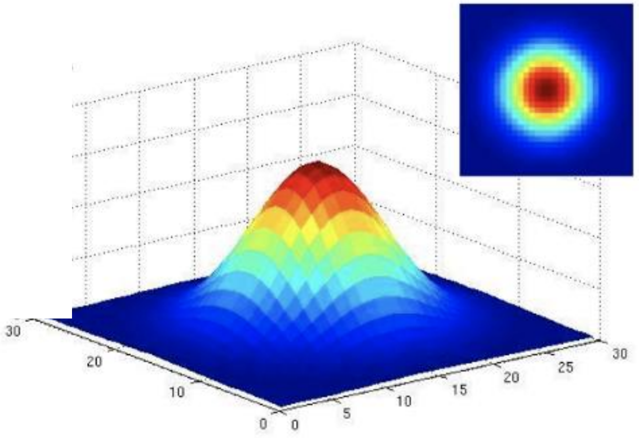
Radial Basis Function

Gaussian (RBF) Kernel:

$$K(u,v) = \phi(u) \cdot \phi(v) = e^{-\frac{\|u-v\|^2}{2\sigma^2}}$$



$\sigma^2 = small$



$\sigma^2 = large$

$\phi(u)$ maps to **infinite-dimensional** features (discussed in Tutorial)

So what?

Source: https://medium.com/jun94-devpblog/cv-2-gaussian-and-median-filter-separable-2d-filter-2d11ee022c66

SVM with Kernel Trick

From Before: $\boxed{w = \sum_i \alpha^{(i)}\, \hat{y}^{(i)} x^{(i)}}$

Objective:

$$J(w) = \frac{1}{2}\|w\|^2 - \sum_i \alpha^{(i)} [\bar{y}^{(i)} (w \cdot \phi(x^{(i)}) + b) - 1]$$

$$\frac{\partial J(w)}{\partial w} = w - \sum_i \alpha^{(i)} \bar{y}^{(i)} \phi(x^{(i)}) = 0$$

$$\frac{\partial J(w)}{\partial b} = \sum_i \alpha^{(i)} \bar{y}^{(i)} = 0$$

… a few math later …

$$J(w) = \sum_i \alpha^{(i)} - \frac{1}{2} \sum_i \sum_j \alpha^{(i)} \alpha^{(j)} \bar{y}^{(i)} \bar{y}^{(j)} \phi(x^{(i)}) \cdot \phi(x^{(j)})$$

$$J(w) = \sum_i \alpha^{(i)} - \frac{1}{2} \sum_i \sum_j \alpha^{(i)} \alpha^{(j)} \bar{y}^{(i)} \bar{y}^{(j)} K(x^{(i)}, x^{(j)})$$

Decision Rule:

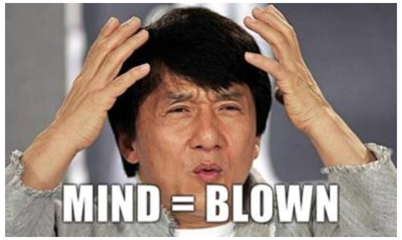
$$w \cdot \phi(x) + b \geq 0 \text{ then } +$$

$$\sum_i \alpha^{(i)}\, \hat{y}^{(i)} \phi(x^{(i)}) \cdot \phi(x) + b \geq 0 \text{ then } +$$

$$\sum_i \alpha^{(i)}\, \hat{y}^{(i)} \quad K(x^{(i)}, x) \quad + b \geq 0 \text{ then } +$$

There is <u>no need</u> to compute the transformed features **explicitly**!

Can have SVM with **infinite-dimensional** features!



Family of kernels

- Not all similarity functions yield valid kernels (aka might not converge)
- Need to satisfy Mercer's theorem
  - (i.e., continuous, symmetric, positive semidefinite)
- Other kernels:
  - String kernel
  - Chi-squared kernel
  - tanh kernel

Summary

- Overfitting
- Regularization
  - Linear and logistic regression
- Support Vector Machine (SVM)
  - Hard-margin SVM
  - Soft-margin SVM
- Kernel
  - SVM with Kernel Trick

Logistic Regression / SVM
With x as features

Logistic Regression / SVM
With $\boldsymbol{\phi}(\boldsymbol{x})$ as features

SVM with Kernel Trick
With $\boldsymbol{\phi}$(x) mapping to
finite-dimensional
features

SVM with Kernel Trick
With $\boldsymbol{\phi}$(x) mapping to
infinite-dimensional
features

Coming Up Next Week

- Perceptron
  - Perceptron Update Rule
- Gradient Descent with Perceptron
- Neural Networks
  - Multi-layer neural networks

To Do

- **Lecture Training 7**
  - +100 Free EXP
  - +50 Early bird bonus
- **Midterm Survey**
  - Due tonight 25:59