Machine Learning Review

講者：Isaac

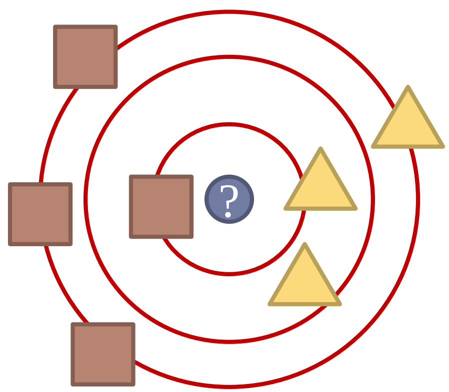
Outline

- K-Nearest Neighbor
- Logistic regression
- Naive Bayes
- Support Vector Machine

K-Nearest Neighbor

What's K-Nearest Neighbor

- A non-parametric method used for classification and regression
- Also called kNN
  - "k" mean how many neighbors should be considered to help classification/regression

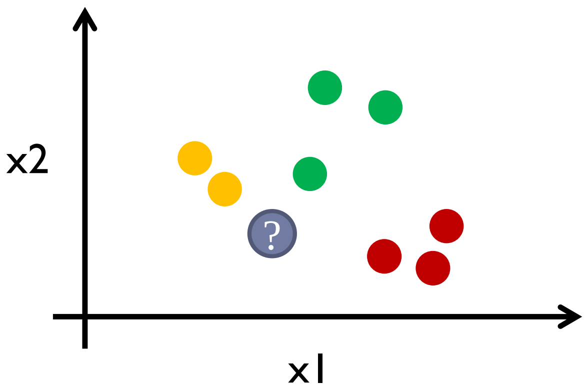


- k=1:
  - Belongs to square class
- k=3
  - Belongs to triangle class
- k=7
  - Belongs to square class

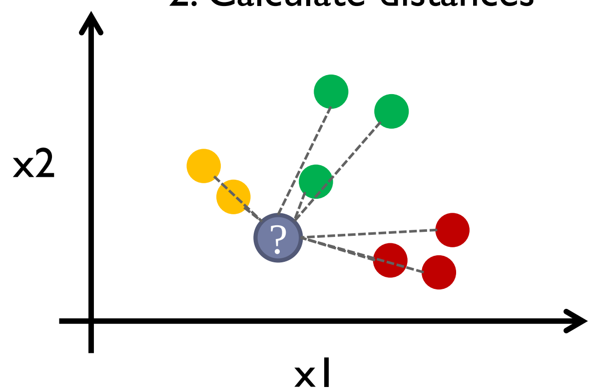
kNN intuitive concept

K-Nearest Neighbor

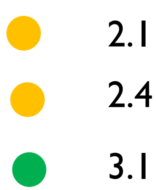
1. Look at the data



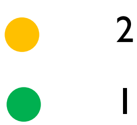
2. Calculate distances



3. Find neighbors



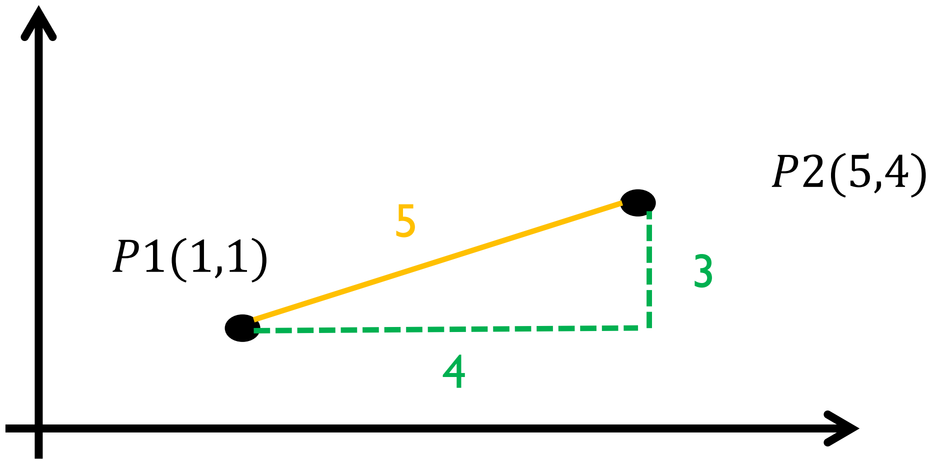
4. vote from labels



guess it is yellow class

How to Define Distance

- L1 distance (Manhattan distance)
- L2 distance (Euclidean distance)

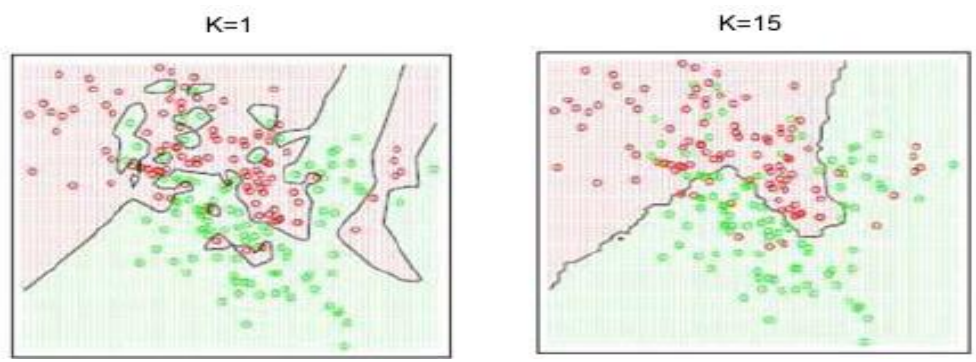


$$Euclidean\ distance = \sqrt{(5-1)^2+(4-1)^2} = 5$$

$$Manhattan\ distance = |5-1| + |4-1| = 7$$

How to choose K?

- K is small
  - sensitive to noise points
- K is large
  - neighborhood may include points from other classes
  - smoother boundary
  - If too large, machine always predict majority class



- http://vision.stanford.edu/teaching/cs231n-demos/knn/

- 1-NN
  - Voronoi Diagram

Logistic regression

Logistic regression

$$Model\colon\ h_\theta = \frac{1}{1+e^{-\theta^T X}}\ where\ \theta = \begin{bmatrix}\theta_0\\ \vdots\\ \theta_n\end{bmatrix}, X = \begin{bmatrix}x_0\\ \vdots\\ x_n\end{bmatrix}$$

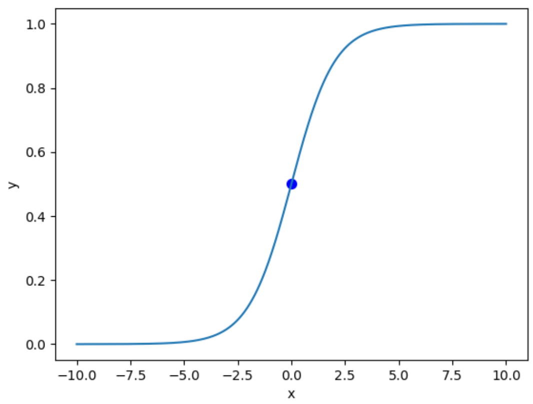
$$Learned\ Parameters\colon\ \theta_0, \theta_1, \dots, \theta_n$$

$$Cost\ Function\colon C(\theta_0, \theta_1, \dots, \theta_n)$$

$$= \frac{1}{2m}\sum_{i=1}^{m} y^{(i)}\log(h^\theta\left(x^{(i)}\right)) + \left(1 - y^{(i)}\right)\log(1 - h^\theta\left(x^{(i)}\right))$$

Logistic regression

- Sigmoid function
  - Output is [0, 1]



$$y = \frac{1}{1 + e^{-x}}$$

Sigmoid function

Logistic regression

- Actually, cost function in logistic regression is cross-entropy
  - note that cross-entropy can be used when each of output is probability distribution

$$\mathbf{p} \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad H(\mathrm{p}, \mathrm{q}) = -\sum_i p_i \ln(\mathrm{q}_i) \qquad \mathbf{q} \quad \begin{bmatrix} 0.1 \\ 0.5 \\ 0.4 \end{bmatrix}$$

Logistic regression

- Information
  - $log\left(\frac{1}{p_i}\right)$ $where\ p_i\ is\ probability\ of\ an\ event$

Sun rises in the east tomorrow	It will rain tomorrow in Taiwan

Which is more informative?

Entropy V.S. Cross-entropy

- Entropy
  - Expected value(mean) of information contained in each message
- Entropy can be seen as index of uncertainty
  - Bigger mean more chaos
- Cross-entropy
  - Measurement on the difference between two probability distribution
  - Different distribution apply on entropy
  - Cross-entropy is greater than entropy

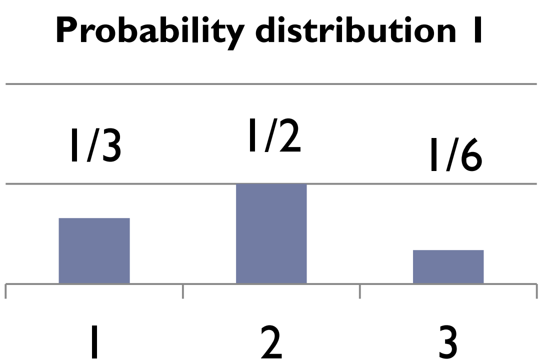
$$H(y) = \sum_i y_i \log\left(\frac{1}{y_i}\right) = -\sum_i y_i \log(y_i)$$

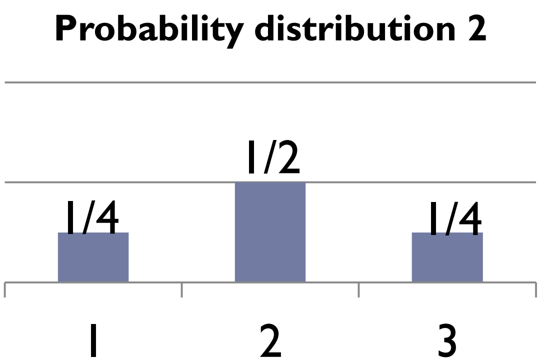
Entropy

$$H(y, \hat{y}) = -\sum_i y_i \log(\widehat{y_i})$$

Cross-entropy

Example





Entropy on distribution 1
= 1/3 * log(3) + 1/2 * log(2) + 1/6 * log(6)

Entropy on distribution 2
= 1/4 * log(4) + 1/2 * log(2) + 1/4 * log(4)

Cross-entropy on distribution 1 over distribution 2
= 1/3 * log(4) + 1/2 * log(2) + 1/6 * log(4)

Cross-entropy on distribution 2 over distribution 1
= 1/4 * log(3) + 1/2 * log(2) + 1/4 * log(6)

Example

Entropy on distribution 1
$= 1/3 * \log(3) + 1/2 * \log(2) + 1/6 * \log(6)$
$= 0.439$

Entropy on distribution 2
$= 1/4 * \log(4) + 1/2 * \log(2) + 1/4 * \log(4)$
$=0.452$

Cross-entropy on distribution 1 over distribution 2
$= 1/3 * \log(4) + 1/2 * \log(2) + 1/6 * \log(4) = 0.456$

Cross-entropy on distribution 2 over distribution 1
$= 1/4 * \log(3) + 1/2 * \log(2) + 1/4 * \log(6) = 0.464$

- Cross-entropy is greater than entropy
  Cross-entropy on distribution 1 over 2 > Entropy on distribution 1
  Cross-entropy on distribution 2 over 1 > Entropy on distribution 2
- If two distribution become closer
  - Value of cross-entropy is closer to entropy

Logistic regression

- Learning in logistic regression
  - Use gradient descent(same as linear regression)

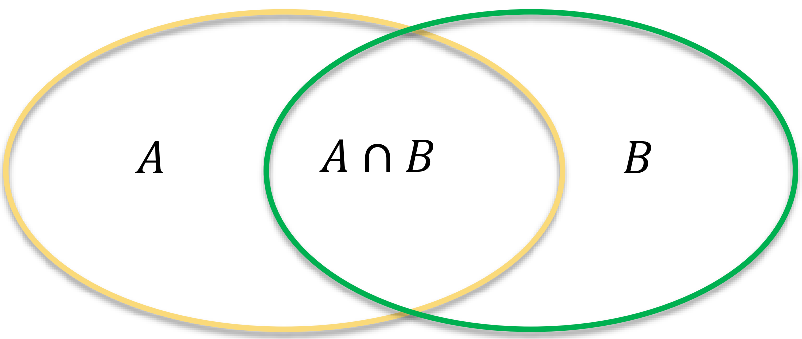
Naive Bayes

What's Naive Bayes

- A family of probabilistic classifiers based on applying Bayes' theorem
- Naive Bayes classifiers are highly scalable
  - require parameters linear in the number of variables features
- Common used in document classification

Thomas Bayes

probability basic



$$p(A \cup B) = p(A) + p(B) - p(A \cap B)$$

independent probability

- Two event are independent if

$$P(A \cap B) = P(A) * P(B)$$

- Example
  - Given a dice, if we toss the dice twice, what's probability that the first toss is even number and the second toss is odd number

$$P(first\ toss\ is\ even\ number\ \cap second\ toss\ is\ odd\ number)$$
$$= \frac{1}{2} * \frac{1}{2}$$
$$= P(first\ toss\ is\ even\ number) * P(\ second\ toss\ is\ odd\ number)$$

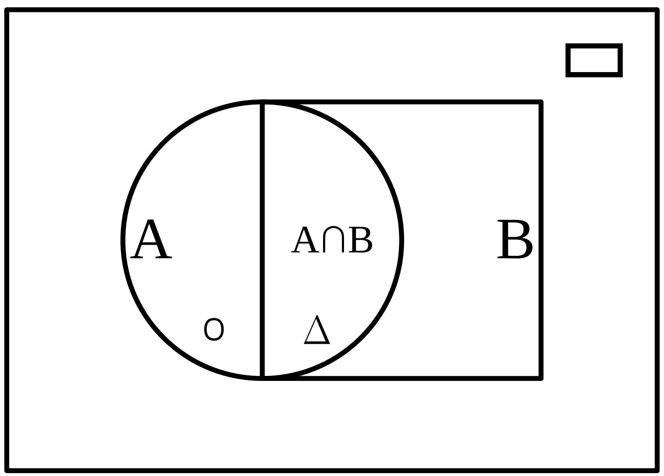
tossing the dice each time is independent event

Naive Bayes

- Bayes' Theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{P(B|A) * P(A) + P(B|\bar{A}) * P(\bar{A})}$$

Bayes' Theorem



$$P(A \cap B) = \frac{\triangle}{\boxed{}}$$

$$P(B|A) \cdot P(A) = \frac{\triangle}{\bigcirc} \times \frac{\bigcirc}{\boxed{}} = \frac{\triangle}{\boxed{}}$$

$$P(A|B) \cdot P(B) = \frac{\triangle}{\square} \times \frac{\square}{\boxed{}} = \frac{\triangle}{\boxed{}}$$

$$P(B|A) \cdot P(A) = P(A \cap B)$$

$$\Rightarrow P(B|A) = \frac{P(A \cap B)}{P(A)}$$

$$\Rightarrow P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$$

Example

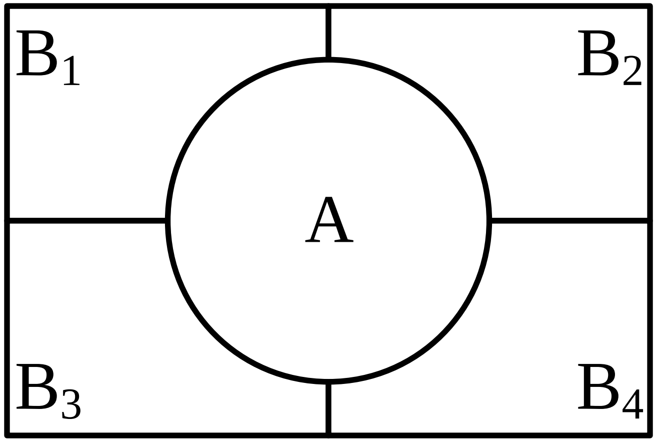
- There are 4000 phones in total.
- There are 2000 phones in B1 box and 10% of them are broken
- There are 500 phones in B2 box and 20% of them are broken
- There are 500 phones in B3 box and 30% of them are broken
- There are 1000 phones in B4 box and 40% of them are broken

If we randomly choose a broken phone,
what's the probability that this phone is from box B3?

Example

$P(B_1) = P(B_2) = P(B_3) = P(B_4) = \frac{1}{4}$

$$
\begin{aligned}
P(B_3|A) &= \frac{P(B_3 \cap A)}{P(A)} \\
&= \frac{P(A|B_3)P(B_3)}{P(A|B_1)P(B_1)+P(A|B_2)P(B_2)+P(A|B_3)P(B_3)+P(A|B_4)P(B_4)} \\
&= \frac{30\% \cdot \frac{1}{4}}{10\% \cdot \frac{1}{4} + 20\% \cdot \frac{1}{4} + 30\% \cdot \frac{1}{4} + 40\% \cdot \frac{1}{4}} \\
&= \frac{0.075}{0.025+0.05+0.075+0.1} \\
&= \frac{0.075}{0.25} \\
&= 30\%
\end{aligned}
$$



How Naive Bayes Classifier Work

- Assume
  - there are three attributes A1, A2, A3 and two class C0 and C1

$$p(C_0| A_1, A_2, A_3) > p(C_1| A_1, A_2, A_3)$$

⇩

Guess it is class 0

$$p(C_0| A_1, A_2, A_3) < p(C_1| A_1, A_2, A_3)$$

⇩

Guess it is class 1

A1	A2	A3	Class
1	0	1	1
0	1	1	0
1	1	0	0
1	1	1	0
1	0	0	1

How Naive Bayes Classifier Work

$$p(C_0 | A_1, A_2, A_3) = \frac{p(A_1, A_2, A_3 | C_0) * p(C_0)}{p(A_1, A_2, A_3)}$$

Assume attributes are independent

$$p(C_0 | A_1, A_2, A_3) = \frac{p(A_1 | C_0) * p(A_2 | C_0) * p(A_3 | C_0) * p(C_0)}{p(A_1, A_2, A_3)}$$

$$p(C_1 | A_1, A_2, A_3) = \frac{p(A_1, A_2, A_3 | C_1) * p(C_1)}{p(A_1, A_2, A_3)}$$

Assume attributes are independent

$$p(C_1 | A_1, A_2, A_3) = \frac{p(A_1 | C_1) * p(A_2 | C_1) * p(A_3 | C_1) * p(C_1)}{p(A_1, A_2, A_3)}$$

How Naive Bayes Classifier Work

- Assume
  - there are three attributes A1, A2, A3 and two class C0 and C1

$$p(A_1|C_0) * p(A_2|C_0) * p(A_3|C_0) * p(C_0) > p(A_1|C_1) * p(A_2|C_1) * p(A_3|C_1) * p(C_1)$$

$\Downarrow$

Guess it is class 0

$$p(A_1|C_0) * p(A_2|C_0) * p(A_3|C_0) * p(C_0) < p(A_1|C_1) * p(A_2|C_1) * p(A_3|C_1) * p(C_1)$$

$\Downarrow$

Guess it is class 1

Example

Goal: predict if the text is about sport

Text	Category
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

Example

- If we want to predict if sentence "a very close game" is sports or not sports, we need to compare the following two term

$$\boxed{p(sports|a\ very\ close\ game\ ) \qquad p(Not\ sports|a\ very\ close\ game)}$$

$$\boxed{p(a\ very\ close\ game\ |sports) * p(sports) p(a\ very\ close\ game\ |Not\ sports)\ p(Not\ sports)}$$

Example

$p(sports|a\ very\ close\ game\ )$ $\qquad$ $p(Not\ sports|a\ very\ close\ game)$

use previous concept, we can compare the following term instead of origin one

$$p(a\ |sports) * p(very\ |sports) * p(close\ |sports) * p(game|sports) * p(sports)$$

$$p(a\ |Not\ sports) * p(very\ |Not\ sports) * p(close\ |Not\ sports) * p(game|Not\ sports) * p(Not\ sports)$$

Example

$$p(a\ |sports) * p(very\ |sports) * p(close\ |sports) * p(game|sports) * p(sports)$$

$$p(a\ |Not\ sports) * p(very\ |Not\ sports) * p(close\ |Not\ sports) * p(game|Not\ sports) * p(Not\ sports)$$

How to calculate each term?

Example

Text	Category
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

$$p(sports) = \frac{3}{5}$$

$$p(Not\ sports) = \frac{2}{5}$$

How to calculate $p(word|Sports)$? The most intuitive way is like the following

$$p(game|Sports) = \frac{2}{11}$$

$$\vdots$$

$$\boxed{p(close\ |Sports) = 0}$$

We don't want this!!!

Example

Text	Category
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

$$p(sports) = \frac{3}{5}$$

$$p(Not\ sports) = \frac{2}{5}$$

In order to deal with zero count problem, we use Laplace smoothing method to calculate $p(word|Sports)$

Example

Text	Category
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

$$p(sports) = \frac{3}{5}$$

$$p(Not\ sports) = \frac{2}{5}$$

$$p(game|Sports) = \frac{2 + \boxed{1}}{11 + \boxed{14}}$$

add one to every count

add # of different words

Multinomial Naive Bayes

Example

Text	Category
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

$$p(sports) = \frac{3}{5}$$

$$p(Not\ sports) = \frac{2}{5}$$

Word	P(word | sports)	P(word | not sports)
a	$\frac{2+1}{11+14}$	$\frac{1+1}{9+14}$
very	$\frac{1+1}{11+14}$	$\frac{0+1}{9+14}$
close	$\frac{0+1}{11+14}$	$\frac{1+1}{9+14}$
game	$\frac{2+1}{11+14}$	$\frac{0+1}{9+14}$

Example

$$p(a\ |sports) * p(very\ |sports) * p(close\ |sports) * p(game|sports) * p(sports)$$
$$= 0.0000276$$

$$p(a\ |Not\ sports) * p(very\ |Not\ sports) * p(close\ |Not\ sports) * p(game|Not\ sports)$$
$$* p(Not\ sports)$$
$$= 0.00000572$$

So, our classifier guess "a very nice game" is sports category

Different probability assumption

Text	Category
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

$$\boxed{p(game|Sports) = \frac{2+1}{11+14}}$$

Why we calculate condition probability like this?

Different probability assumption

Actually, we can assume conditional probability as different probability distribution

$$p(attribute\ 1|class\ 1) = N(attribute1|\ \mu, \sigma)$$
$$where\ N\ is\ gaussian\ distribution$$

Assume conditional probability as gaussian distribution

Example

Height	Weight	Shoe size	Gender
6.00	180	12	Male
5.92	190	11	Male
5.58	170	12	Male
5.92	165	10	Male
5.00	100	6	Female
5.50	150	8	Female
5.42	130	7	Female
5.75	150	9	Female

$$p(male) = p(female) = \frac{1}{2}$$

$$p(height|male) = N(height|\mu_{hm}, \sigma_{hm})$$

$$p(weight|male) = N(weight|\mu_{wm}, \sigma_{wm})$$

$$p(shoe|male) = N(shoe|\mu_{sm}, \sigma_{sm})$$

$$p(height|female) = N(height|\mu_{hf}, \sigma_{hf})$$

$$p(weight|female) = N(weight|\mu_{wf}, \sigma_{wf})$$

$$p(shoe|female) = N(shoe|\mu_{sf}, \sigma_{sf})$$

Example

$$p(height|male) = N(height|\mu_{hm}, \sigma_{hm})$$

$$p(weight|male) = N(weight|\mu_{wm}, \sigma_{wm})$$

$$p(shoe|male) = N(shoe|\mu_{sm}, \sigma_{sm})$$

$$p(height|female) = N(height|\mu_{hf}, \sigma_{hf})$$

$$p(weight|female) = N(weight|\mu_{wf}, \sigma_{wf})$$

$$p(shoe|female) = N(shoe|\mu_{sf}, \sigma_{sf})$$

	height mean	height variance	weight mean	weight variance	shoe size mean	shoe size variance
$male$	$\mu_{hm} = 5.855$	$\sigma^2_{hm} = .0350$	$\mu_{wm} = 176.25$	$\sigma^2_{wm} = 122.9$	$\mu_{sm} = 11.25$	$\sigma^2_{sm} = .9167$
$female$	$\mu_{hf} = 5.418$	$\sigma^2_{hf} = .0972$	$\mu_{wf} = 132.5$	$\sigma^2_{wf} = 558.3$	$\mu_{sf} = 7.5$	$\sigma^2_{sf} = 1.667$

Example

If a sample with height = 6, weight=130, and shoe=8, predict if it is male or female?

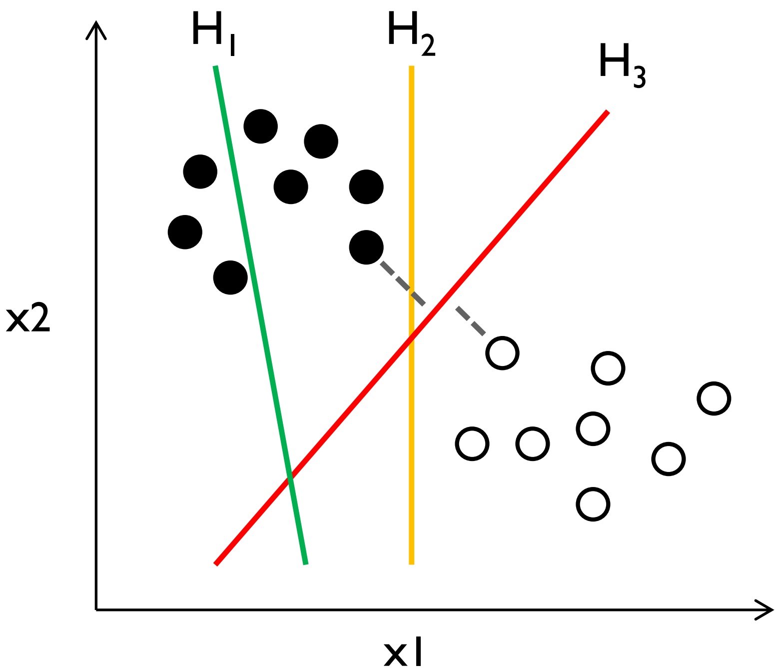
$$
\begin{aligned}
p(male \mid height, weight, shoe) &\propto p(male)\, p(height \mid male)\, p(weight \mid male)\, p(shoe \mid male) \\
&\propto p(male)\, \mathcal{N}(height \mid \mu_{hm}, \sigma_{hm})\, \mathcal{N}(weight \mid \mu_{wm}, \sigma_{wm})\, \mathcal{N}(shoe \mid \mu_{sm}, \sigma_{sm}) \\
&\propto \frac{1}{2}\, \mathcal{N}(6 \mid 5.855, \sqrt{.0350})\, \mathcal{N}(130 \mid 176.25, \sqrt{122.9})\, \mathcal{N}(8 \mid 11.25, \sqrt{.9167}) \\
&= .5 \times 1.579 \times 5.988 \cdot 10^{-6} \times 1.311 \cdot 10^{-3} \\
&= 6.120 \cdot 10^{-9}
\end{aligned}
$$

$$
\begin{aligned}
p(female \mid height, weight, shoe) &\propto p(female)\, p(height \mid female)\, p(weight \mid female)\, p(shoe \mid female) \\
&\propto p(female)\, \mathcal{N}(height \mid \mu_{hf}, \sigma_{hf})\, \mathcal{N}(weight \mid \mu_{wf}, \sigma_{wf})\, \mathcal{N}(shoe \mid \mu_{sf}, \sigma_{sf}) \\
&\propto \frac{1}{2}\, \mathcal{N}(6 \mid 5.418, \sqrt{.0972})\, \mathcal{N}(130 \mid 132.5, \sqrt{558.3})\, \mathcal{N}(8 \mid 7.5, \sqrt{1.667}) \\
&= .5 \times 2.235 \cdot 10^{-1} \times 1.679 \cdot 10^{-2} \times 2.867 \cdot 10^{-1} \\
&= 5.378 \cdot 10^{-4}
\end{aligned}
$$

Support Vector Machine

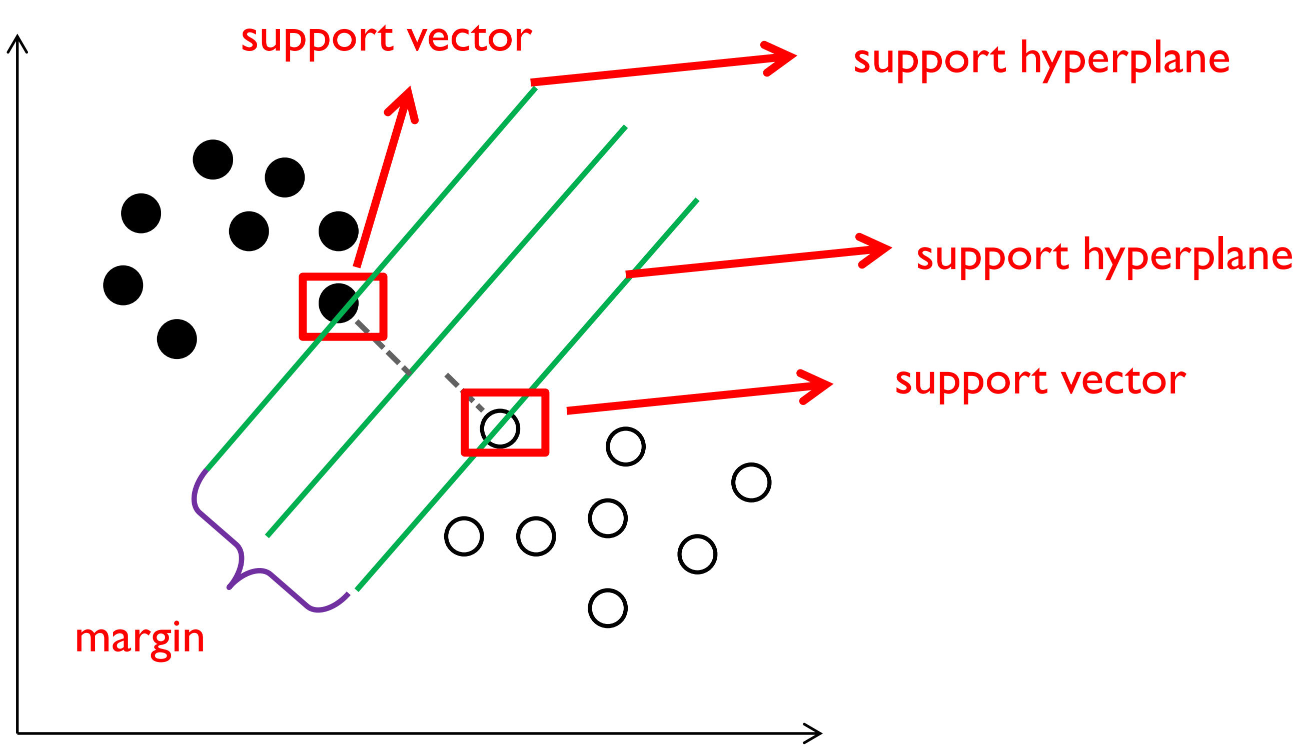
What's Support Vector Machine

- support vector machines (SVM) are supervised learning models
- Linear SVM find a hyperplane that separate data with maximum margin



- H_1 does not separate the classes
- H_2 does, but only with a small margin
- H_3 separates them with the maximum margin

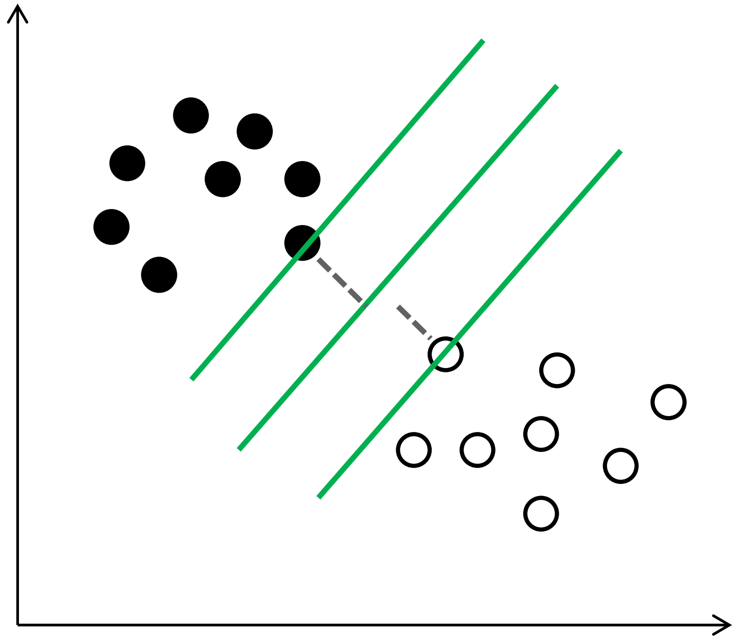
What's Support Vector

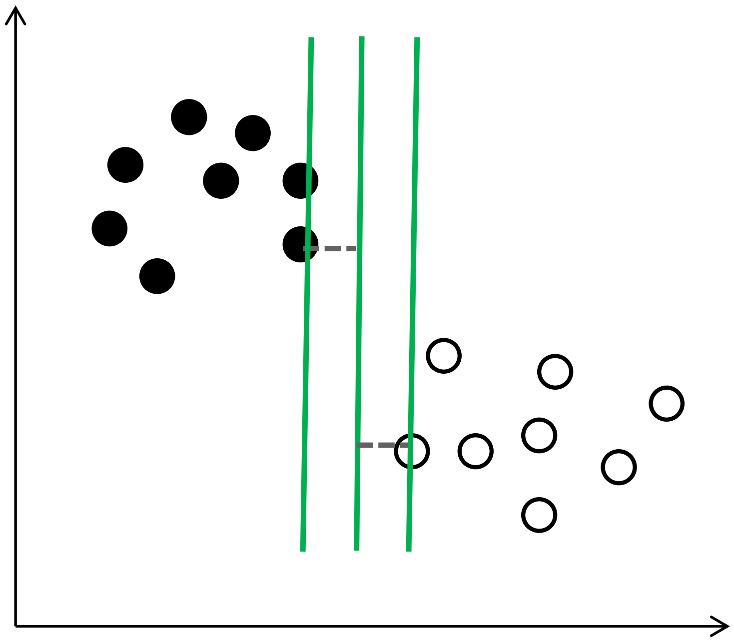


support vectors are points that affect hyperplane with maximum margin

What SVM do

Choose the one with large margin





How to calculate margin/distance?

- What's margin/distance between hyperplane $2x - y + 2z = -5$ and point $(2, 0, 0)$

change all stuffs on one side

$$2x - y + 2z + 5 = 0$$

calculate distance

$$\frac{|2 * 2 - 0 - 2 * 0 + 5|}{\sqrt{2^2 + (-1)^2 + 2^2}} = 3$$

How to calculate margin/distance?

- Any Hyperplane in N-dimension can model

$$w^T x - b = 0$$

$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Example

- What's distance between the following 5-D hyperplane and point (2, 3, 4,1,1)

$$H\colon w^T x - b = 0 \; where \; w = \begin{bmatrix} 1 \\ -1 \\ 2 \\ 3 \\ 1 \end{bmatrix} and \; b = -5$$

Example

change all stuffs on one side $w^T x - b = 0$

$$w = \begin{bmatrix} 1 \\ -1 \\ 2 \\ 3 \\ 1 \end{bmatrix} and\ b = -5$$

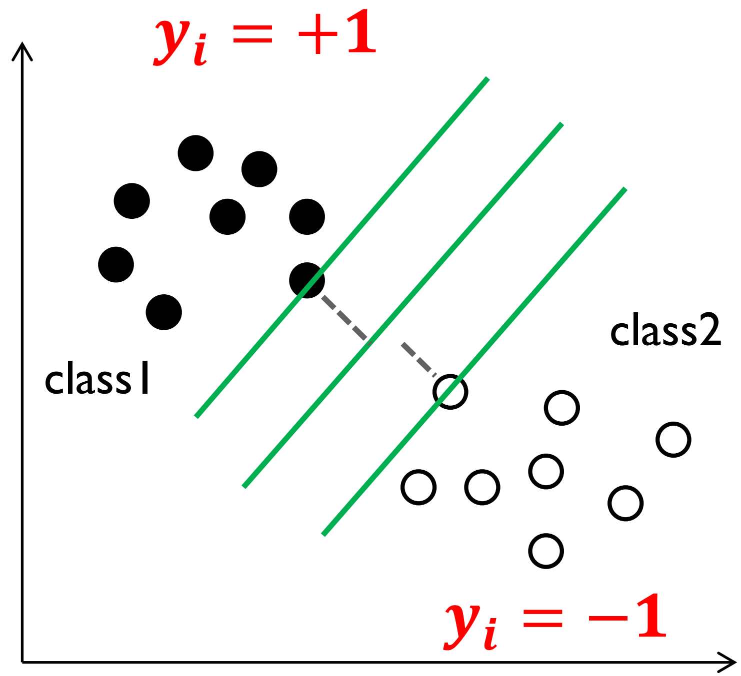
calculate distance

$$\frac{|1 * 2 + (-1) * 3 + 2 * 4 + 3 * 1 + 1 * 1 + 5|}{\sqrt{1^2 + (-1)^2 + 2^2 + 3^2 + 1^2}} = 4$$

SVM

$$\{x_i, y_i\}, i = 1, \dots \dots, n$$
$$x_i \in R^d, y^i \in \{+1, -1\}$$

label

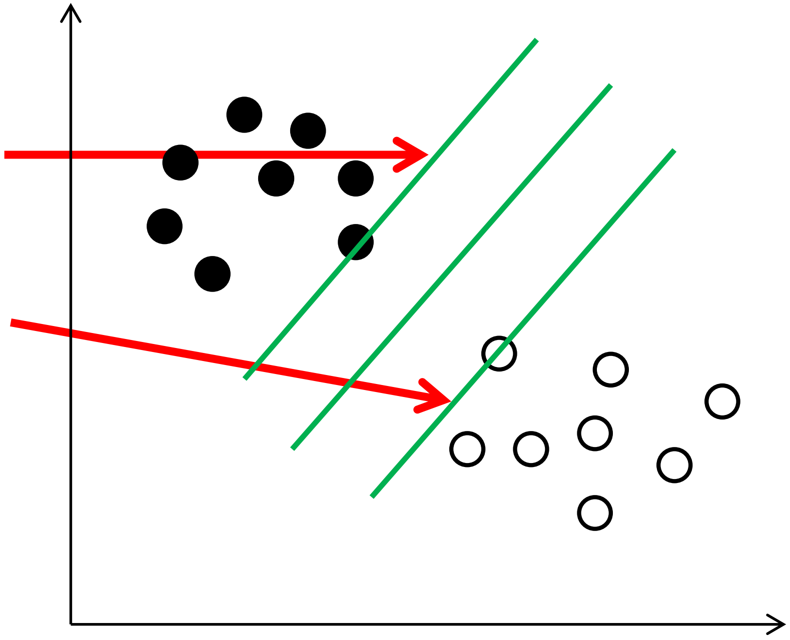


SVM

Assume

$$support\ hyperplane(black)\ is$$
$$f(x_{black}) = w^T x_{black} - b = 1$$

$$support\ hyperplane(white)\ is$$
$$f(x_{white}) = w^T x_{white} - b = -1$$



SVM

$$f(x) = w^T x - b = 0$$

$if\ f(x_i) = w^T x_i - b < 0$

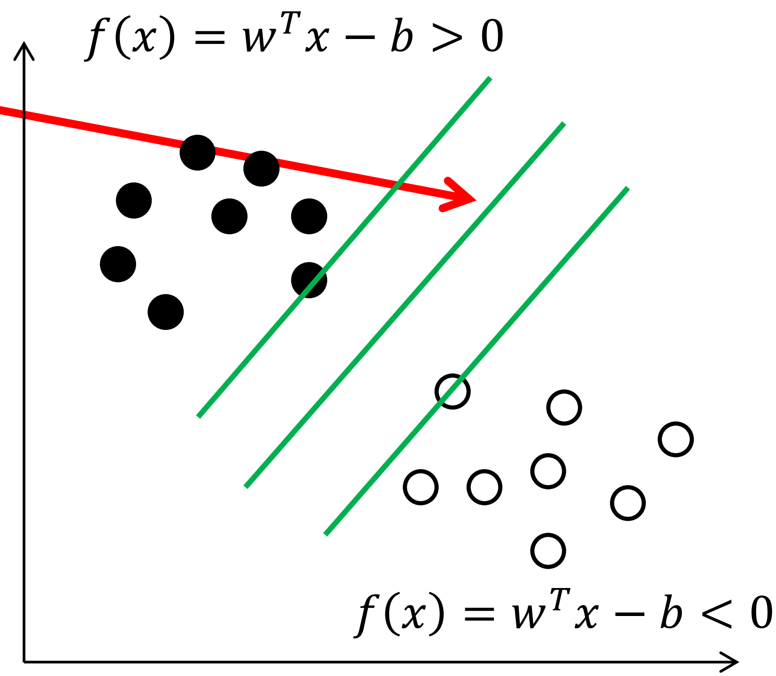
Class 1

$y_i = -1$

$if\ f(x_i) = w^T x_i - b > 0$

Class 2

$y_i = +1$



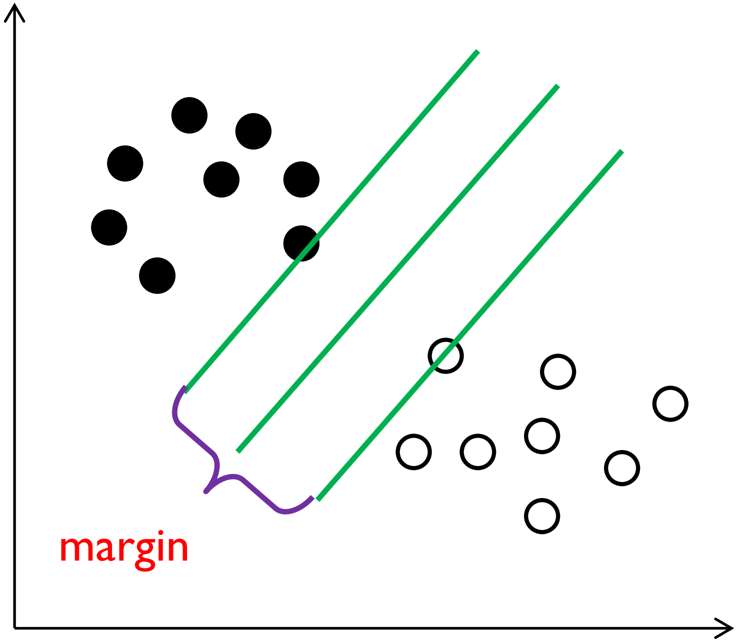
SVM

Goal(what we want):

$$margin = \frac{|w^T x_{black} - b|}{\|w\|} + \frac{|w^T x_{white} - b|}{\|w\|} = \frac{2}{\|w\|}$$

Note :

$$f(x_{black}) = w^T x_{black} - b = 1$$
$$f(x_{white}) = w^T x_{white} - b = -1$$



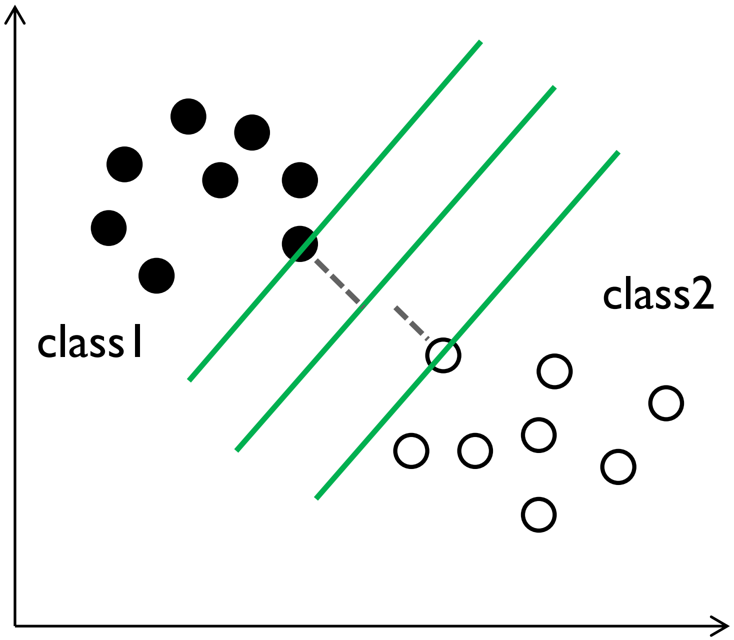
SVM

Constrain:

$$w^T x_i - b \leq -1 \;\; \forall y_i = -1$$
$$w^T x_i - b \geq -1 \;\; \forall y_i = +1$$

combine

$$y_i(w^T x_i - b) - 1 \geq 0$$



SVM

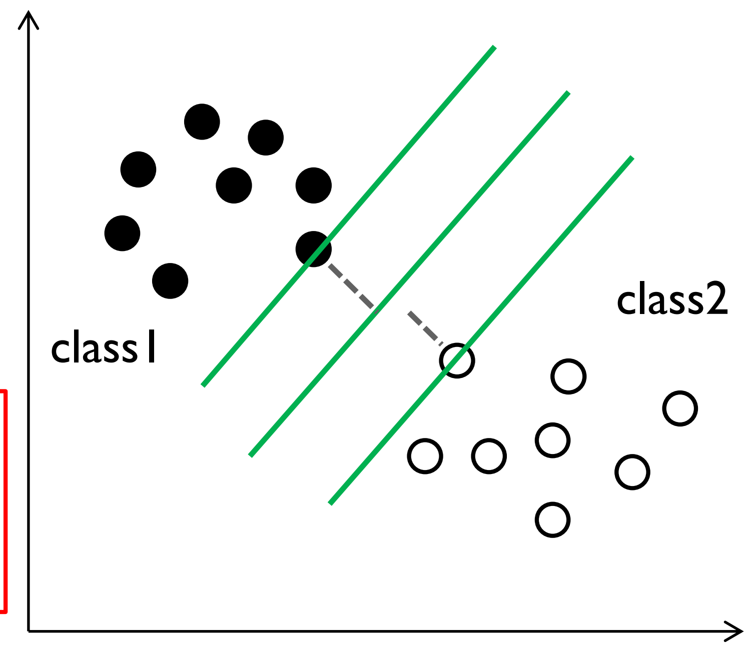
$$max \frac{2}{\|w\|}$$

$$subject\ to\ y_i(w^T x_i - b) - 1 \geq 0\ \forall i$$

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

$$subject\ to\ y_i(w^T x_i - b) \geq 1\ \forall i$$

What SVM solve in math



SVM

$$\boxed{\begin{aligned} & min \frac{1}{2} \|w\|^2 \\ & subject\ to\ y_i(w^T x_i - b) \geq 1\ \forall i \end{aligned}}$$

How to solve actually?

Please reference:
http://www.cmlab.csie.ntu.edu.tw/~cyy/learning/tutorials/SVM2.pdf

Hard Cost V.S. Soft Cost

Hard Cost

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

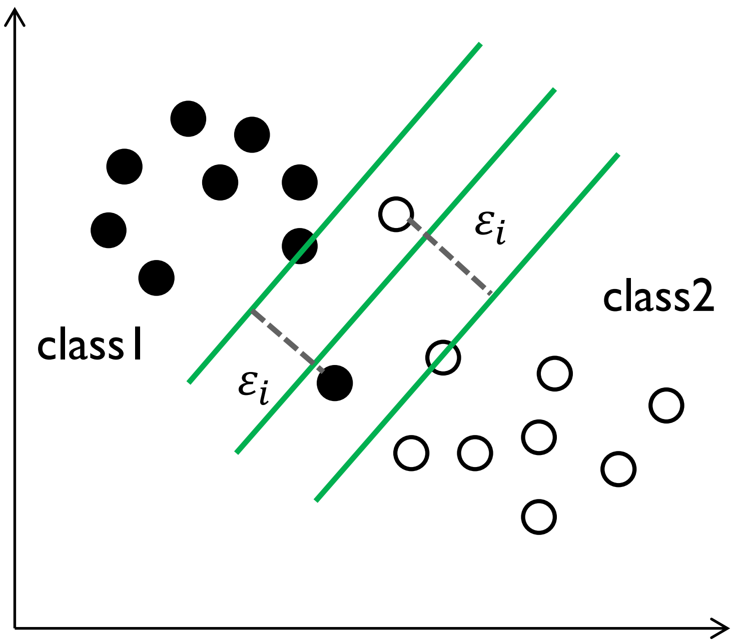
$$subject\ to\ y_i(w^T x_i - b) \geq 1\ \forall i$$

Soft Cost

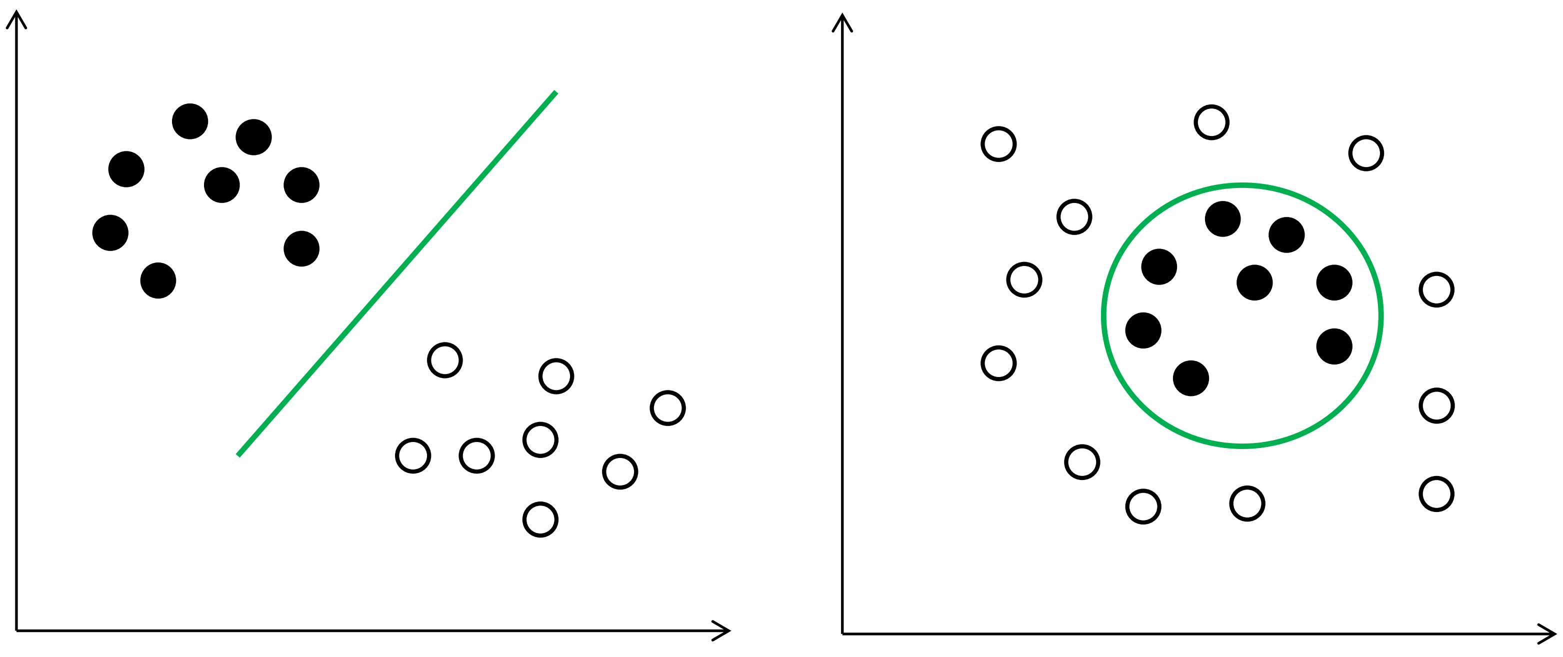
$$min \frac{1}{2} \|w\|^2 + C \sum \varepsilon_i$$

$$subject\ to\ y_i(w^T x_i - b) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0\ \forall i$$

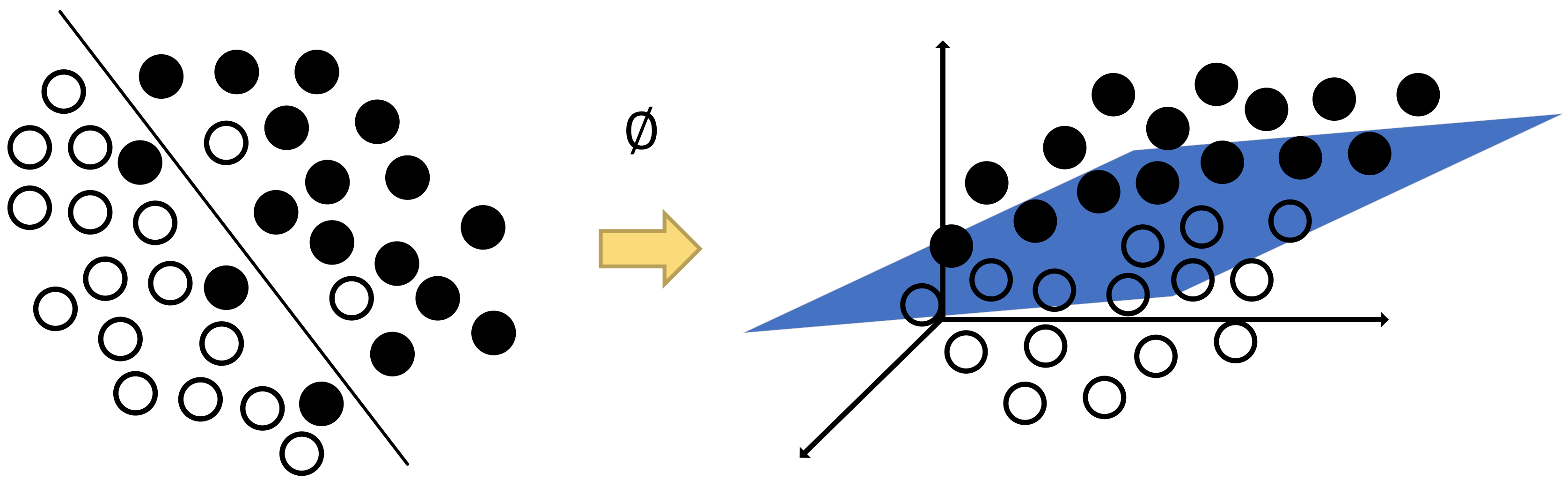


Linear VS nonlinear problems



SVM Kernel Trick

- Usually, data can't be linear separable
  - map data to higher dimension
  - https://www.youtube.com/watch?v=3liCbRZPrZA



SVM Kernel Trick

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \to \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1 x_2 \end{pmatrix} \qquad \mathbb{R}^2 \to \mathbb{R}^3$$

$$\begin{aligned} \boxed{\Phi(\mathbf{x})^\top \Phi(\mathbf{z})} &= \left(x_1^2, x_2^2, \sqrt{2}x_1 x_2\right) \begin{pmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2}z_1 z_2 \end{pmatrix} \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 x_2 z_1 z_2 \\ &= (x_1 z_1 + x_2 z_2)^2 \\ &= (\mathbf{x}^\top \mathbf{z})^2 \end{aligned}$$

kernel

SVM Kernel Trick

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

$$subject\ to\ y_i(w^T x_i - b) \geq 1\ \forall i$$

primal problem

$$\max \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i)^T x_j$$

$$subject\ to\ \alpha_i \geq 0\ \forall i$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

dual problem

SVM Kernel Trick

$$\emptyset(x_i)^T \emptyset(x_j)$$

$$max \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \boxed{(x_i)^T x_j}$$

$$subject\ to\ \alpha_i \geq 0\ \forall i$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

dual problem

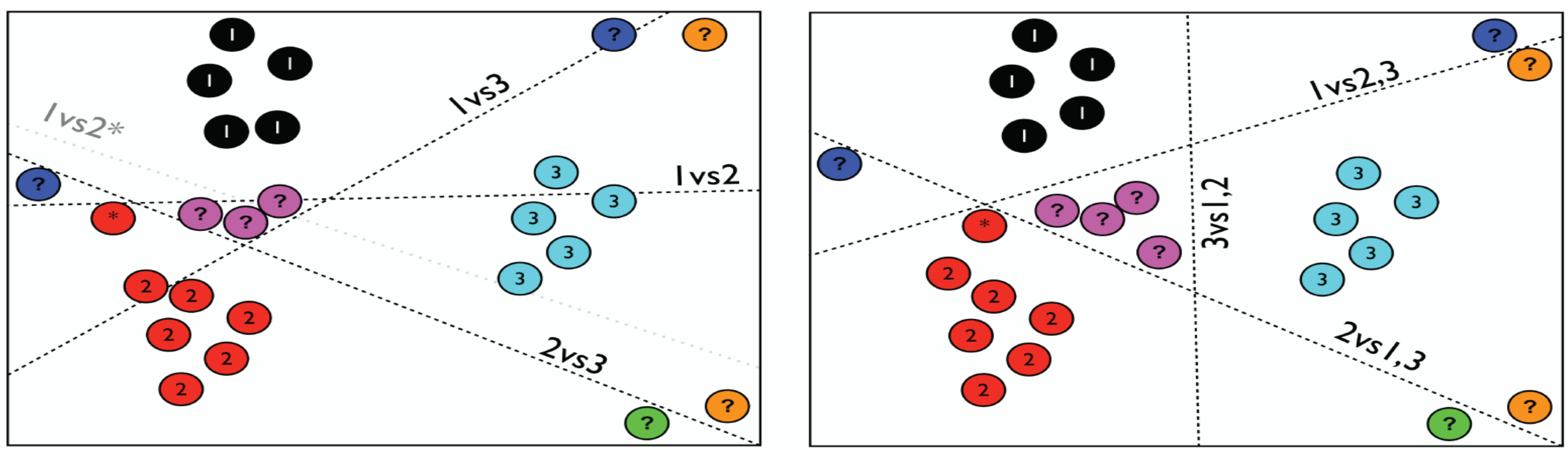
Common Kernel in SVM

Kernel name	Kernel function
Linear kernel	$K(x, y) = x \times y$
Polynomial kernel	$K(x, y) = (x \times y + 1)^d$
RBF kernel	$K(x, y) = e^{-\gamma \|x-y\|^2}$

Multi-class in SVM

- If there are k class
  - Method 1: one-against-rest(One-vs-All)
    - Make k SVM binary classifier and use m-th of binary SVM predict if the data belong to m-th class
  - Method 2: one-against-one(OvO)
    - Make $\frac{n(n-1)}{2}$ binary classifier (n is # of class) and each of binary SVM predict if the data belong to one of any two class

Multi-class in SVM



(a) 1-vs-1 (b) 1-vs-All