USC Viterbi
School of Engineering
Daniel J. Epstein Department of Industrial and Systems Engineering

Predictive Analytics (ISE529)

Support Vector Machines (II)

Dr. Tao Ma
ma.tao@usc.edu

Tue/Thu, Aug 26 - Dec 6, 2024, Fall

USC University of Southern California

UNDERSTAND VECTOR

Vector Norm

Definition: A vector is an object that has both a magnitude and a direction.

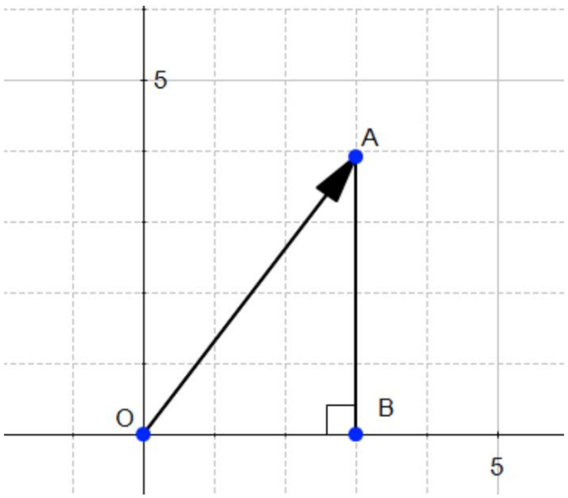
The magnitude or length of a vector *x* is written ‖*x*‖ and is called its norm.

For vector $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

The l^2-norm is calculated by

$$\|x\| = \sqrt{\sum_{k=1}^{n} x_k^2}$$

Example:
For vector $\overrightarrow{OA}$, ‖OA‖ is the length of the segment OA.



$$OA^2 = 3^2 + 4^2$$

$$OA^2 = 25$$

$$OA = \sqrt{25}$$

$$\|OA\| = OA = 5$$

Vector Direction

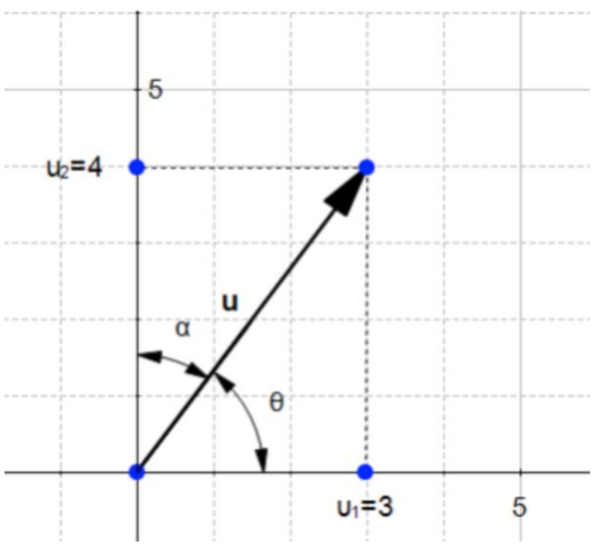
Definition: The direction of a vector **u**(u_1, u_2) is the vector $\mathbf{w}\left(\frac{u_1}{\|u\|}, \frac{u_2}{\|u\|}\right)$.

The direction of the vector **u** is defined by the *cosine* of the angle θ and the *cosine* of the angle α.

$$cos(\theta) = \frac{u_1}{\|u\|}$$

$$cos(\alpha) = \frac{u_2}{\|u\|}$$

i.e., the original definition of the vector **w**.



$$cos(\theta) = \frac{u_1}{\|u\|} = \frac{3}{5} = 0.6 \qquad cos(\alpha) = \frac{u_2}{\|u\|} = \frac{4}{5} = 0.8$$

The direction of **u**(3,4) is the vector **w**(0.6,0.8) that its norm is equal to 1 and is called **unit vector**.

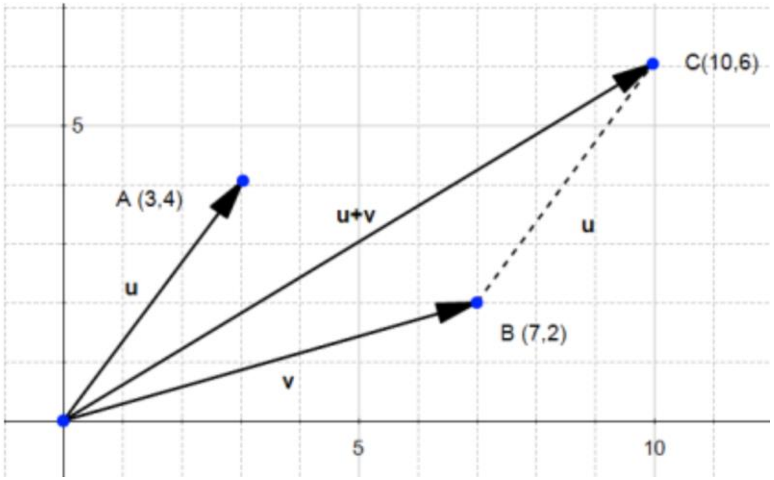
Add and Subtract Vectors

- The sum of two vectors

Given two vectors $\mathbf{u}(u_1,u_2)$ and $\mathbf{v}(v_1,v_2)$ then :

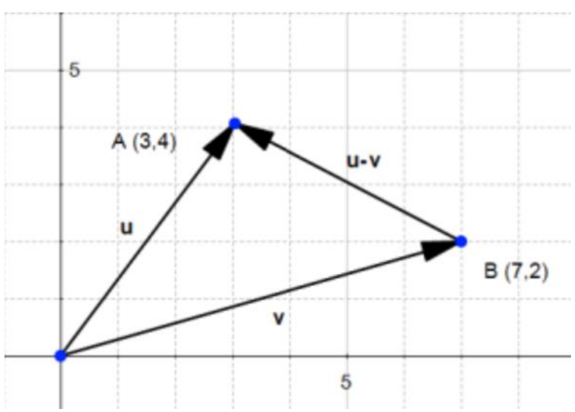
$$\mathbf{u} + \mathbf{v} = (u_1 + v_1, u_2 + v_2)$$

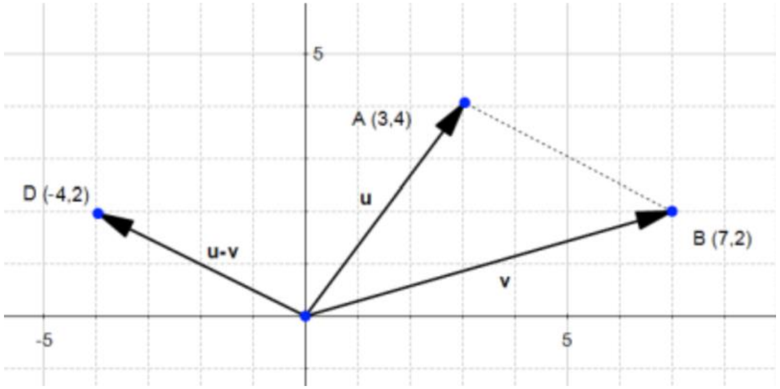
adding two vectors gives us a **third vector** whose coordinate are the sum of the coordinates of the original vectors.



- The difference between two vectors

$$\mathbf{u} - \mathbf{v} = (u_1 - v_1, u_2 - v_2)$$





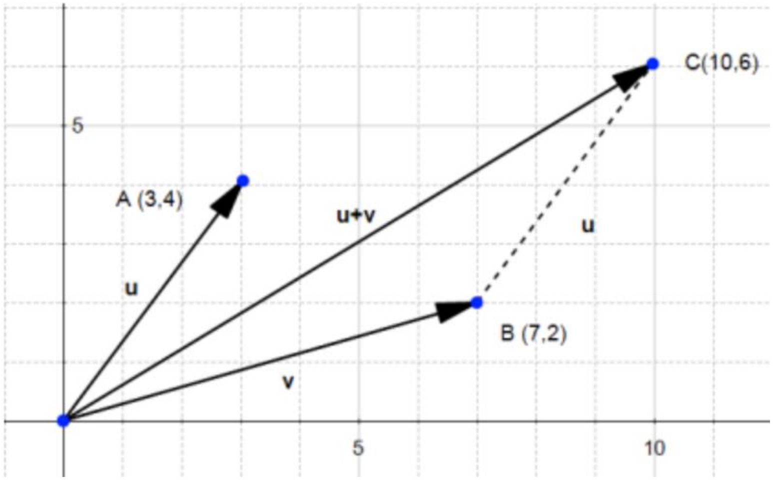
vectors with the same magnitude and direction but with a different origin are the same vector.

Add and Subtract Vectors

- The sum of two vectors

Given two vectors **u**(u_1,u_2) and **v**(v_1,v_2) then :

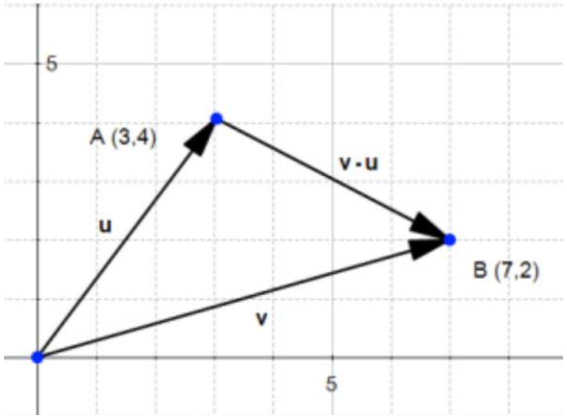
$$\mathbf{u}+\mathbf{v}=(u_1+v_1,u_2+v_2)$$

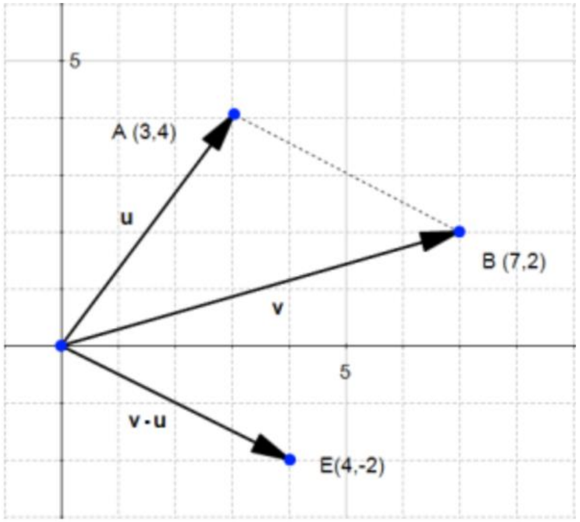


adding two vectors gives us a **third vector** whose coordinate are the sum of the coordinates of the original vectors.

- The difference between two vectors

$$\mathbf{v}-\mathbf{u}=(v_1-u_1,v_2-u_2)$$





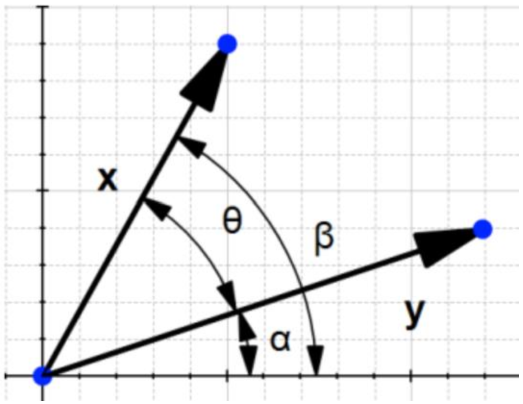
Dot Product/Inner Product

USC Viterbi School of Engineering

Definition: **Geometrically**, it is the product of the Euclidian magnitudes of the two vectors and the *cosine* of the angle between them.

If we have two vectors x and y and there is an angle θ between them, their dot product is:

$$\mathbf{x} \cdot \mathbf{y} = \|x\|\|y\|cos(\theta)$$



Because,

$$\cos\theta = \cos(\beta - \alpha) = \cos\beta\cos\alpha + \sin\beta\sin\alpha$$

$$cos(\beta) = \frac{adjacent}{hypotenuse} = \frac{x_1}{\|x\|}$$

$$sin(\beta) = \frac{opposite}{hypotenuse} = \frac{x_2}{\|x\|}$$

$$cos(\alpha) = \frac{adjacent}{hypotenuse} = \frac{y_1}{\|y\|}$$

$$sin(\alpha) = \frac{opposite}{hypotenuse} = \frac{y_2}{\|y\|}$$

$$cos(\theta) = \frac{x_1}{\|x\|}\frac{y_1}{\|y\|} + \frac{x_2}{\|x\|}\frac{y_2}{\|y\|}$$

$$\|x\|\|y\|cos(\theta) = x_1y_1 + x_2y_2$$

The algebraic definition of the dot product is:

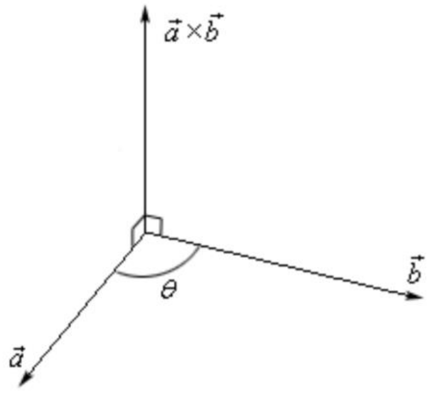
$$\mathbf{x} \cdot \mathbf{y} = x_1y_1 + x_2y_2 = \sum_{i=1}^{2}(x_iy_i)$$

Also called the inner product $\langle x, y \rangle$

Cross Product

Definition: Geometrically, let θ be the angle between the two vectors and assume that $0 \leq \theta \leq \pi$, then we have the following fact.

$$\left\|\vec{a} \times \vec{b}\right\| = \|\vec{a}\| \; \left\|\vec{b}\right\| \sin\theta$$



The cross product is **orthogonal to both original vectors**.

$$\vec{a} \times \vec{b} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} \begin{matrix} \vec{i} & \vec{j} \\ a_1 & a_2 \\ b_1 & b_2 \end{matrix}$$

We multiply along each diagonal and add those that move from left to right and subtract those that move from right to left.

Example If $\vec{a} = \langle 2, 1, -1 \rangle$ and $\vec{b} = \langle -3, 4, 1 \rangle$ compute

$$\begin{aligned} \vec{a} \times \vec{b} &= \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 2 & 1 & -1 \\ -3 & 4 & 1 \end{vmatrix} \begin{matrix} \vec{i} & \vec{j} \\ 2 & 1 \\ -3 & 4 \end{matrix} \\ &= \vec{i}\,(1)\,(1) + \vec{j}\,(-1)\,(-3) + \vec{k}\,(2)\,(4) - \vec{j}\,(2)\,(1) - \vec{i}\,(-1)\,(4) - \vec{k}\,(1)\,(-3) \\ &= 5\vec{i} + \vec{j} + 11\vec{k} \end{aligned}$$

The Orthogonal Projection of a Vector

Given two vectors ***x*** and ***y***, we would like to find the orthogonal projection of ***x*** onto ***y***. This give us the vector **z.**

By definition: $cos(\theta) = \frac{||z||}{||x||}$ $\quad ||z|| = ||x||cos(\theta)$

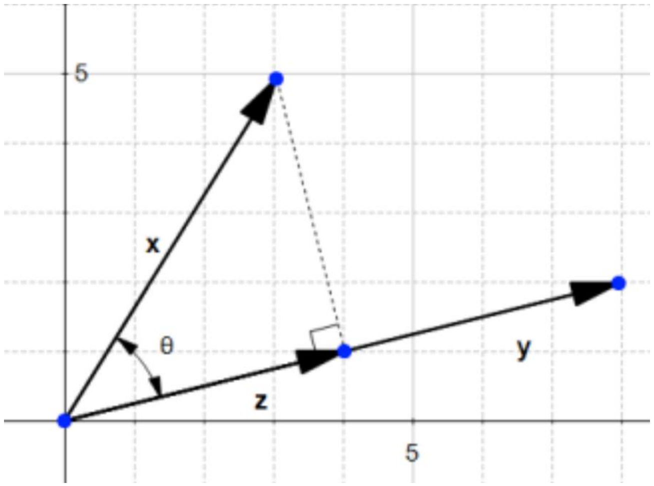
By the dot product:

$$cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{||x||\,||y||}$$

Thus,

$$||z|| = ||x|| \frac{\mathbf{x} \cdot \mathbf{y}}{||x||\,||y||}$$

$$||z|| = \frac{\mathbf{x} \cdot \mathbf{y}}{||y||}$$



Define the vector **u** as the direction of **y**, then $\mathbf{u} = \frac{\mathbf{y}}{||y||}$

We now have a simple way to compute the norm of the vector **z**: $||z|| = \mathbf{u} \cdot \mathbf{x}$

Since this vector is in the same direction as **y** it has the direction **u,** then $\mathbf{z} = ||z||\mathbf{u}$

The vector **z** = (**u**·**x**)**u** is the orthogonal projection of x onto y.

EQUATIONS OF PLANES

Equations of Planes

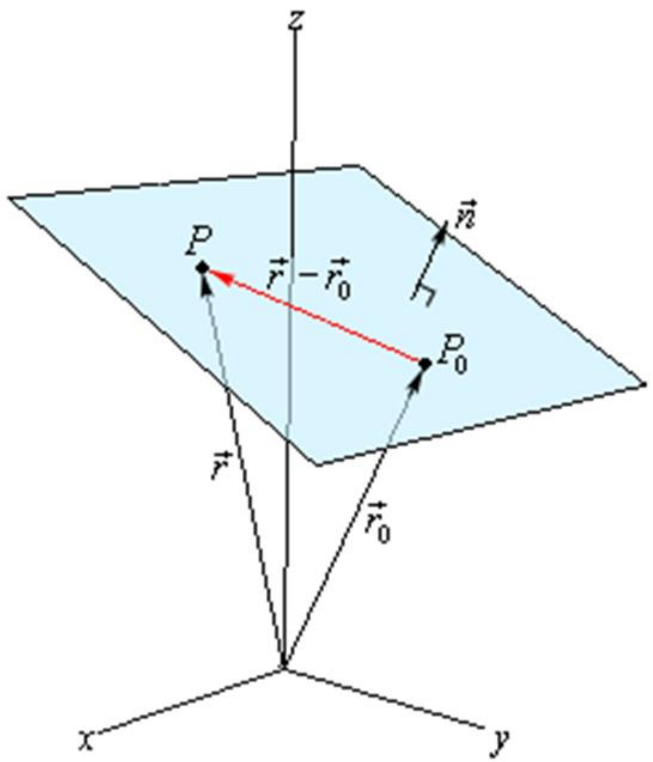
Let assume that
$P_0 = (x_0, y_0, z_0)$ is a point that is on the plane.

$P = (x, y, z)$ is any point in the plane.

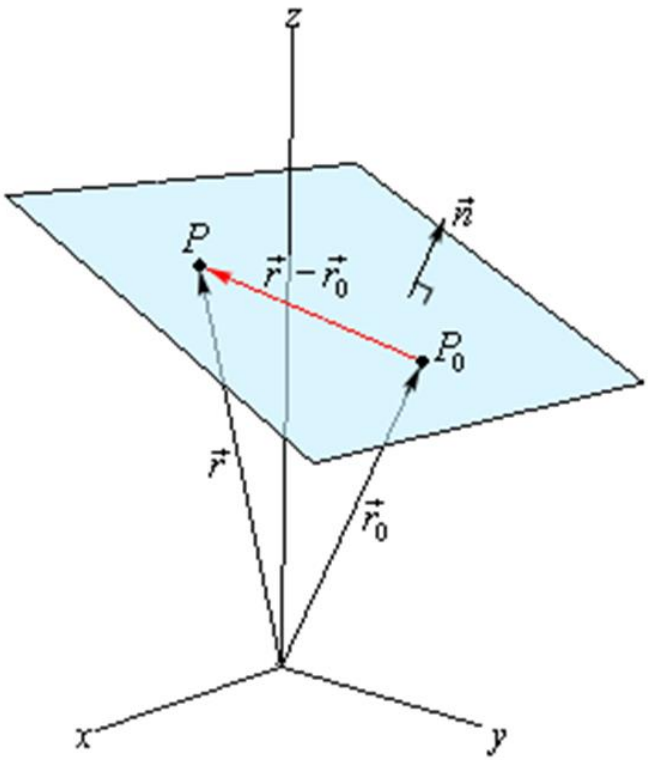
$\overrightarrow{r_0}$ and $\vec{r}$ be the position vectors for
P_0 and P respectively.

$\vec{n} = \langle a, b, c \rangle$ is a vector that is orthogonal (perpendicular) to the plane, called the **normal vector**.

The vector $\vec{r} - \overrightarrow{r_0}$ will lie completely in the plane.



Equations of Planes



Recall that two orthogonal vectors will have a dot product of zero.

$$\vec{n}\cdot\left(\vec{r}-\overrightarrow{r_0}\right)=0$$

This is called the **vector equation of the plane.**

$$\langle a,b,c\rangle\cdot(\langle x,y,z\rangle-\langle x_0,y_0,z_0\rangle)=0$$
$$\langle a,b,c\rangle\cdot\langle x-x_0,y-y_0,z-z_0\rangle=0$$

$$a\left(x-x_0\right)+b\left(y-y_0\right)+c\left(z-z_0\right)=0$$

This is called the **scalar equation of plane**. Often this will be written as

$$ax+by+cz=d$$

where

$$d=ax_0+by_0+cz_0$$

Example

Determine the equation of the plane that contains the points

$$P = (1, -2, 0),\ Q = (3, 1, 4) \text{ and } R = (0, -1, 2).$$

Solution:
To write down the equation of plane we need a point and a normal vector. We get two vectors from the given points. These two vectors will lie completely in the plane.

$$\overrightarrow{PQ} = \langle 2, 3, 4 \rangle \qquad \overrightarrow{PR} = \langle -1, 1, 2 \rangle$$

The cross product of two vectors will be **orthogonal** to both of these vectors and will also be orthogonal to the plane.

$$\vec{n} = \overrightarrow{PQ} \times \overrightarrow{PR} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 2 & 3 & 4 \\ -1 & 1 & 2 \end{vmatrix} \begin{matrix} \vec{i} & \vec{j} \\ 2 & 3 \\ -1 & 1 \end{matrix} = 2\vec{i} - 8\vec{j} + 5\vec{k}$$

The equation of the plane is then using point *P*,

$$\begin{aligned} 2(x-1) - 8(y+2) + 5(z-0) &= 0 \\ 2x - 8y + 5z &= 18 \end{aligned}$$

THE SVM HYPERPLANE

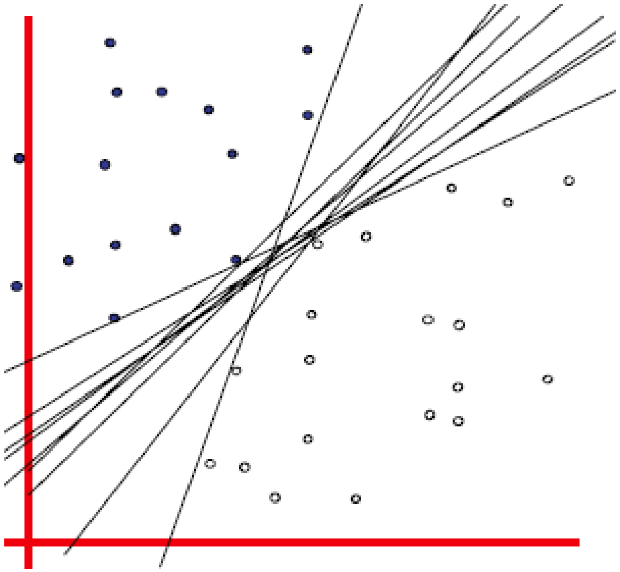
SVM Hyperplane

What is the **goal** of the Support Vector Machine (SVM)?
The goal of a support vector machine is to find the optimal separating hyperplane which maximizes the margin of the training data.

What is the **optimal** separating hyperplane?
There are many ways to draw the stick. We will try to select a hyperplane **as far as possible from data points from each class**. We use "**margin**" to define the optimality of a hyperplane.

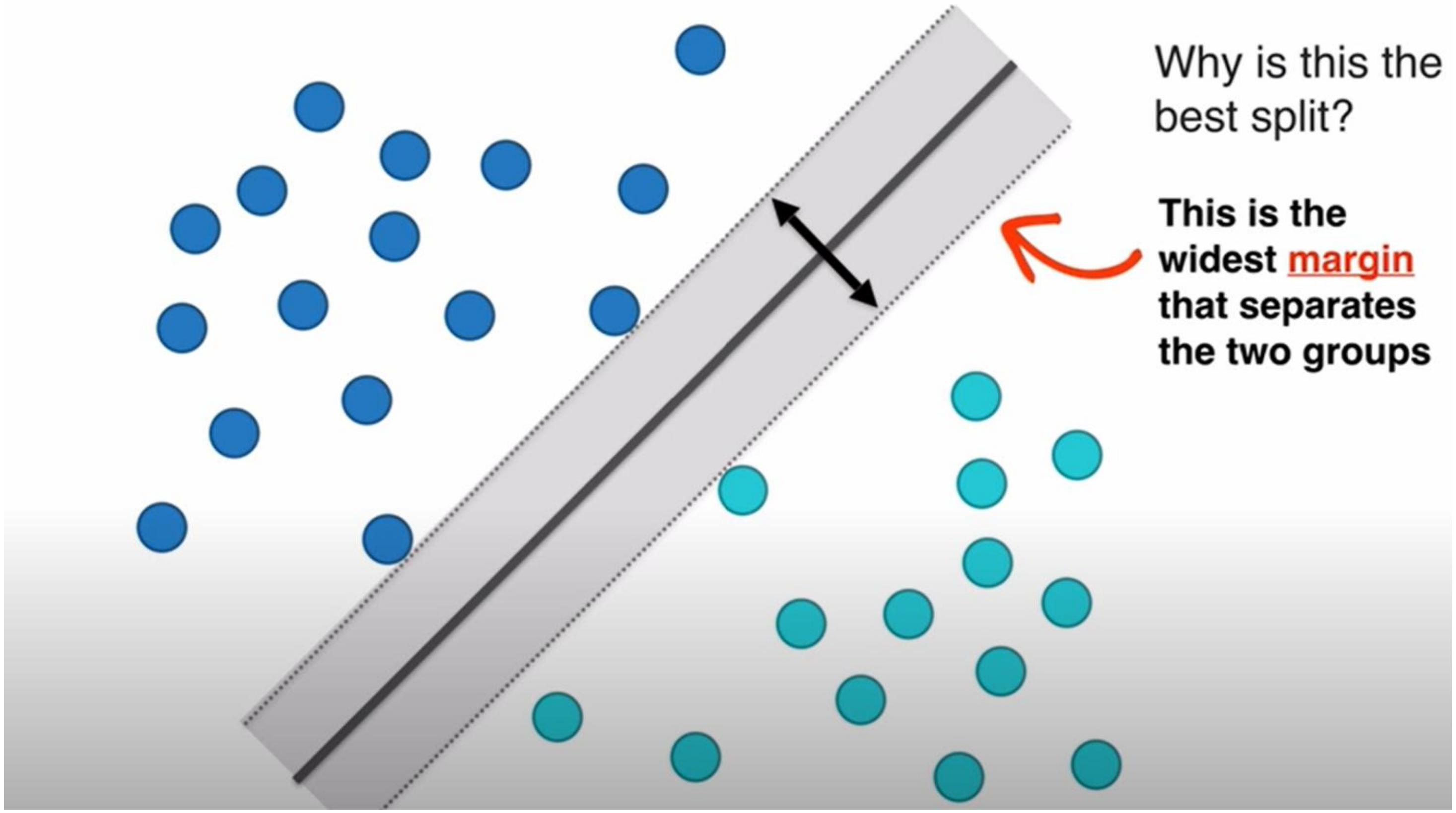
What is the margin and how does it help choosing the optimal hyperplane?

Basically, the **margin** is a no man's land. There will never be any data point inside the margin.



SVM Hyperplane

We need to select two hyperplanes separating the data with no points between them.



Equation of Hyperplane

A hyperplane is a generalization of a plane.

- in one dimension, a hyperplane is called a point
- in two dimensions, it is a line
- in three dimensions, it is a plane
- in more dimensions you can call it a hyperplane

Use vector **w** to represent a hyperplane, the equation of a hyperplane is usually defined by:

$$\mathbf{w}^T\mathbf{x} = 0$$

the inner product of two vectors, where **w** is normal vector orthogonal to the hyperplane.
Example: Given two vectors

$$\mathbf{w} \begin{pmatrix} -b \\ -a \\ 1 \end{pmatrix} \text{ and } \mathbf{x} \begin{pmatrix} 1 \\ x \\ y \end{pmatrix}$$

$$\mathbf{w}^T\mathbf{x} = -b \times (1) + (-a) \times x + 1 \times y$$

$$\mathbf{w}^T\mathbf{x} = y - ax - b$$

It is the same thing as

$$y - ax - b = 0$$

$$y = ax + b$$

Equation of Hyperplane

Why do we use the hyperplane equation $\mathbf{w}^T\boldsymbol{x}$ instead of $y = ax + b$, an equation of a line, or the scalar equation of plane $ax + by + cz = d$.

For two reasons:

- it is easier to work in more than two dimensions with this notation,
- the vector **w** will always be normal to the hyperplane (we use this vector to define the hyperplane, so it will be normal by definition.)

Example

Compute the distance from a point A(3, 4) to the hyperplane.

Given two vectors to define the hyperplane:

$\mathbf{w}\begin{pmatrix}2\\1\end{pmatrix}$ and $\mathbf{x}\begin{pmatrix}x_1\\x_2\end{pmatrix}$ $\quad \mathbf{w}^T\mathbf{x}=0$ which is equivalent to $x_2=-2x_1$

a is a vector from the origin to A. If we project it onto the normal vector **w**. We get the vector **P**.

$$\|w\| = \sqrt{2^2+1^2} = \sqrt{5}$$

Let the vector **u** be the direction of **w**

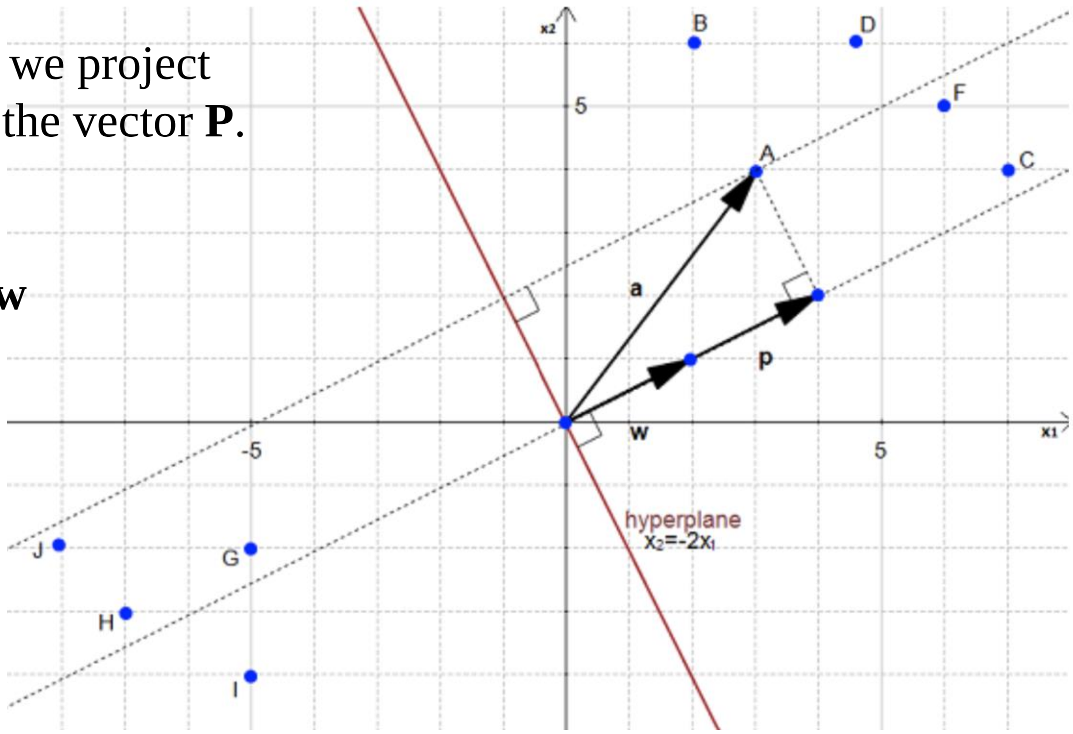
$$\mathbf{u} = \left(\frac{2}{\sqrt{5}}, \frac{1}{\sqrt{5}}\right)$$

P is the orthogonal projection of **a** onto **w** so :

$$\mathbf{p} = (\mathbf{u}\cdot\mathbf{a})\mathbf{u}$$

$$\mathbf{p} = \left(3\times\frac{2}{\sqrt{5}} + 4\times\frac{1}{\sqrt{5}}\right)\mathbf{u}$$

$$\|p\| = \sqrt{4^2+2^2} = 2\sqrt{5}$$



Example

We computed the distance ‖p‖ between a point A and a hyperplane. We then computed the margin which was equal to 2‖p‖.

$$margin = 2||p|| = 4\sqrt{5}$$

However, it was not the optimal hyperplane.

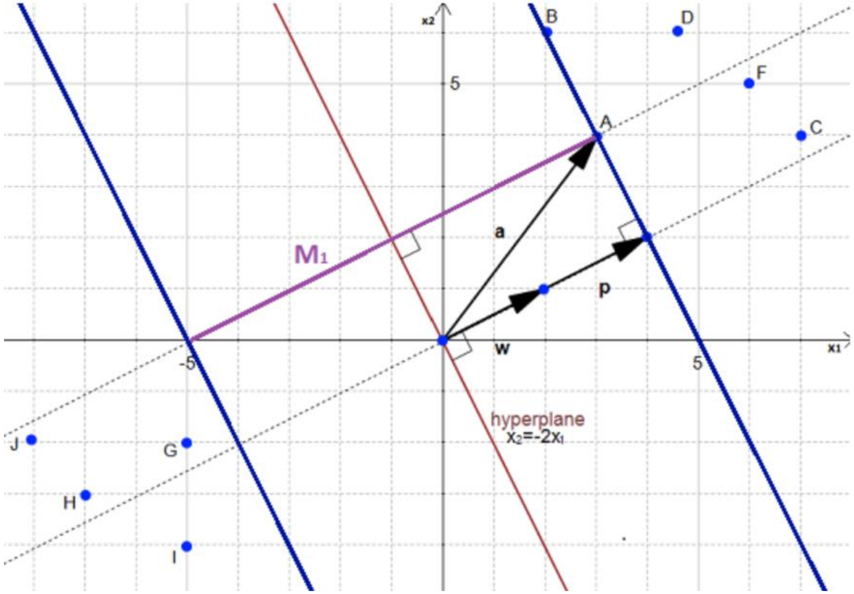


Figure 1

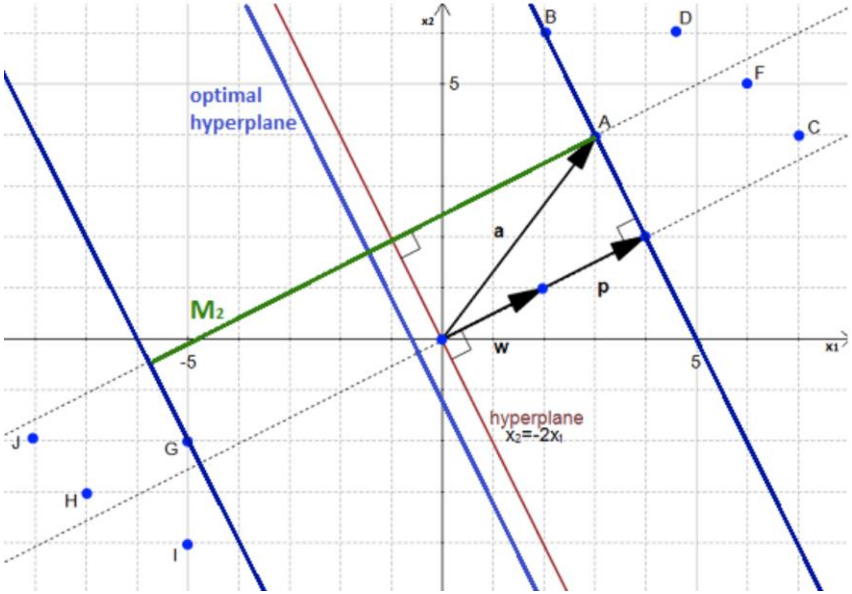


Figure 2

The optimal hyperplane is the one which maximizes the margin of the training data. See the optimal hyperplane on Figure 2.

Data Set

USC Viterbi School of Engineering

How do we calculate this max margin?

1. You have a dataset *D* and you want to classify it

Most of the time your data will be composed of *n* vectors x_i. Each x_i will also be associated with a value y_i indicating if the element belongs to the class (+1) or not (-1). Note that y_i can only have two possible values -1 or +1.

We can say that x_i is a *p*-dimensional vector if it has *p* dimensions.

So your dataset *D* is the set of *n* couples of element (x_i, y_i).

The more formal definition of an initial dataset *D* is

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p,\ y_i \in \{-1, 1\}\}_{i=1}^{n}$$

Select Hyperplanes

2. Select two hyperplanes separating the data with no points between them. Let's assume that our dataset D is linearly separable. If we slightly change the definition of **w** for leaving the intercept out, any hyperplane can be written as the set of points *x* satisfying

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

We can select two other hyperplanes H_0 and H_1 which also separate the data and satisfy the following equations:

$$\mathbf{w} \cdot \mathbf{x} + b = 1$$
$$\mathbf{w} \cdot \mathbf{x} + b = -1$$

so that H is **equidistant** from H_0 and H_1.

Define Constraints

We want to be sure that they have no points between them. We will only select those who meet the two following constraints:

For each vector x_i either :

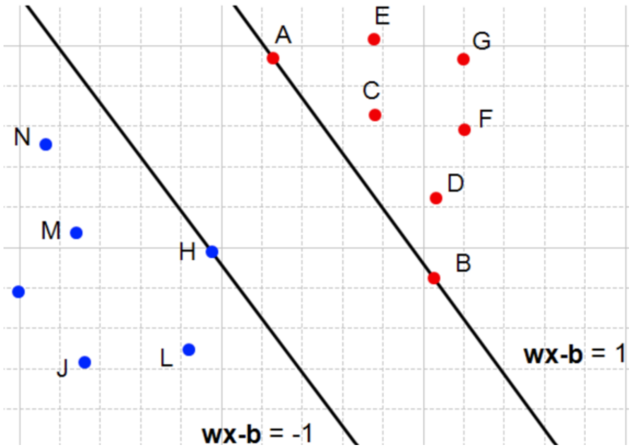
$$\mathbf{w} \cdot \mathbf{x_i} + b \geq 1 \text{ for } \mathbf{x_i} \text{ having the class } 1$$

or

$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 \text{ for } \mathbf{x_i} \text{ having the class } -1$$

Combining both constraints

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1 \text{ for all } 1 \leq i \leq n$$



Compute the Margin

3. Maximize the distance between the two hyperplanes

Compute the distance between **two** hyperplanes.

Let:
H_0 be the hyperplane having the equation $\mathbf{w} \cdot \boldsymbol{x} + b = -1$
H_1 be the hyperplane having the equation $\mathbf{w} \cdot \boldsymbol{x} + b = 1$
x_0 be a point on the hyperplane H_0.

We will call m the perpendicular distance from x_0 to the hyperplane H_1 . As x_0 is in H_0, m is the distance between hyperplanes H_0 and H_1. By definition, m is what we are used to call the "**margin**".

Compute the Margin

We know the vector **w** perpendicular to H_1 (because H_1: $\mathbf{w} \cdot \boldsymbol{x} + b = 1$)

Let's construct a vector:

$$\mathbf{k} = m\mathbf{u} = m\frac{\mathbf{w}}{||\mathbf{w}||}$$

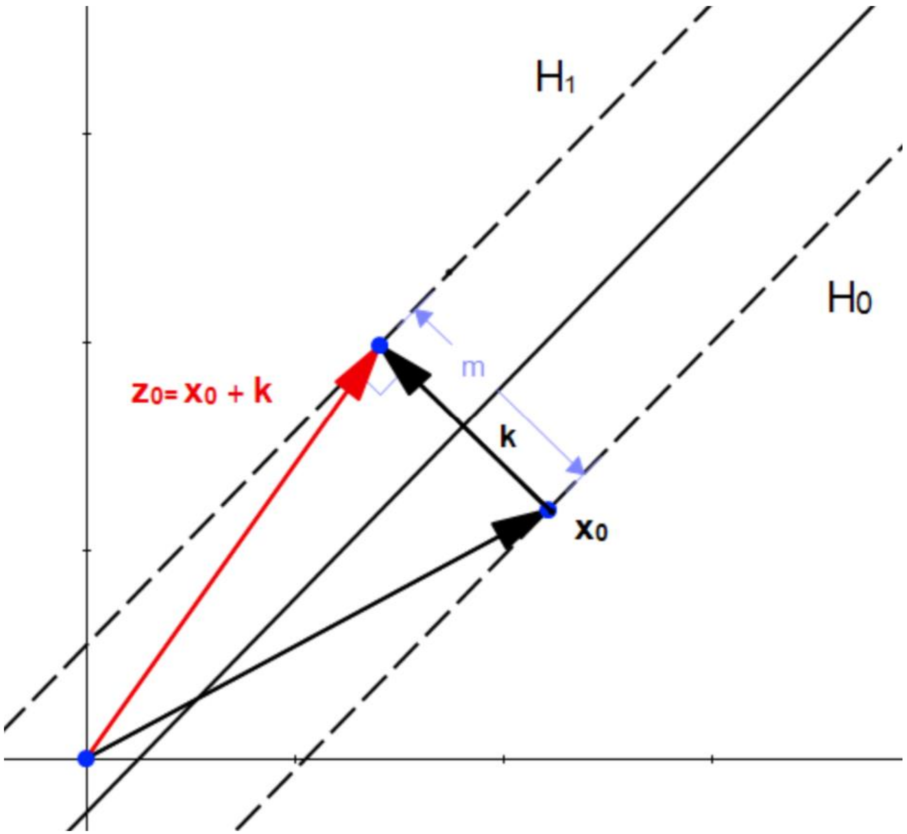
where

$$\mathbf{u} = \frac{\mathbf{w}}{||\mathbf{w}||}$$ the unit vector of **w**

that has the same direction as **w**, so it is also perpendicular to the hyperplane.

$$||\mathbf{k}|| = m$$

k is perpendicular to H_1 (because it has the same direction as **u**), which is the vector we were looking for.



Compute the Margin

From vector **k**, we find the vector:

$$\mathbf{z}_0 = \mathbf{x}_0 + \mathbf{k}$$

corresponding to the point on the hyperplane H_1.

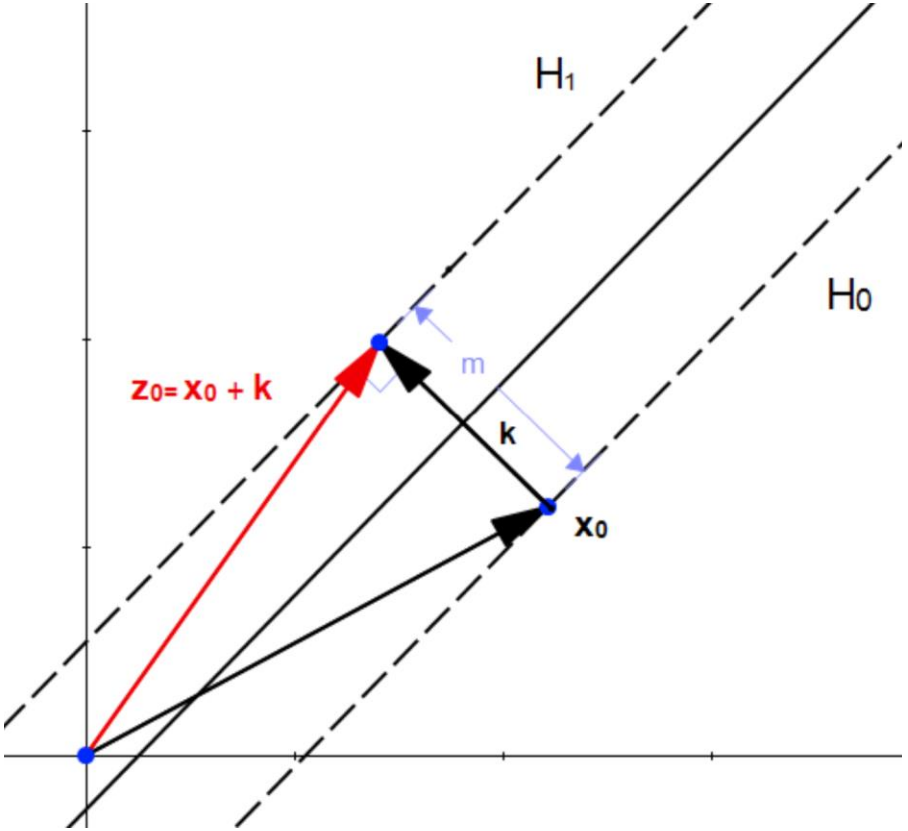
Use the fact that $\mathbf{z}_0$ is on H_1 and $\boldsymbol{x}_0$ is on H_0, We can show that

$$\mathbf{w} \cdot \mathbf{z}_0 + b = 1$$

$$\mathbf{w} \cdot (\mathbf{x}_0 + \mathbf{k}) + b = 1$$

$$\mathbf{w} \cdot (\mathbf{x}_0 + m\frac{\mathbf{w}}{\|\mathbf{w}\|}) + b = 1$$

$$\mathbf{w} \cdot \mathbf{x}_0 + m\frac{\mathbf{w} \cdot \mathbf{w}}{\|\mathbf{w}\|} + b = 1$$



Compute the Margin

The dot product of a vector with itself is the square of its norm so:

$$\mathbf{w} \cdot \mathbf{x}_0 + m\frac{\|\mathbf{w}\|^2}{\|\mathbf{w}\|} + b = 1$$

$$\mathbf{w} \cdot \mathbf{x}_0 + m\|\mathbf{w}\| + b = 1$$

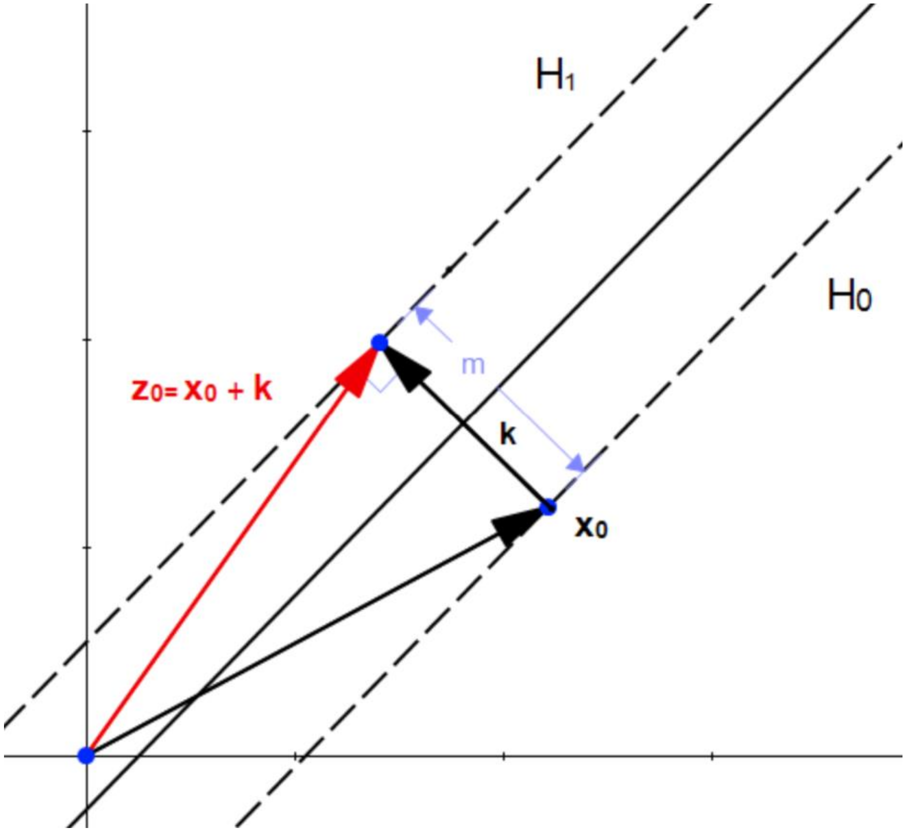
$$\mathbf{w} \cdot \mathbf{x}_0 + b = 1 - m\|\mathbf{w}\|$$

As x_0 is in H_0 then $\mathbf{w} \cdot \boldsymbol{x}_0 + b = -1$.

$$-1 = 1 - m\|\mathbf{w}\|$$

$$m\|\mathbf{w}\| = 2$$

$$m = \frac{2}{\|\mathbf{w}\|}$$



OPTIMIZATION MODEL

Optimization for Margin

We now have a formula to compute the margin:

$$m = \frac{2}{\|\mathbf{w}\|}$$

This give us the following optimization problem, maximizing the margin *m* is the same thing as minimizing the norm of **w**:

Minimize in $(\mathbf{w}, b)$

$$\|\mathbf{w}\|$$

subject to $y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1$

(for any $i = 1, \ldots, n$)

Once we have solved it, we will have found the couple (w,b) for which ||w|| is the smallest possible. We will have the equation of the optimal hyperplane.

Optimization for Margin

We now have a formula to compute the margin:

$$m = \frac{2}{||\mathbf{w}||}$$

The optimization problem can be formulated in different ways but share one thing in common, minimizing the norm of **w**:

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{minimize}} && \frac{1}{2}||\mathbf{w}||^2 \\ &\text{subject to} && y_i(\mathbf{w}\cdot\mathbf{x}_i)+b \geq 1, \quad i=1,\ldots,m \end{aligned}$$

The factor ½ has been added for later convenience, when we use quadratic programing (QP) solver to solve the problem and squaring the norm has the advantage of removing the square root.

Lagrange Method

The Lagrange multiplier method

Introduce the Lagrangian function:

$$\begin{aligned}\mathcal{L}(\mathbf{w}, b, \alpha) &= \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \\ &= \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1]\end{aligned}$$

We introduced one Lagrange multiplier α_i for each constraint function.

Lagrange Method

The Lagrange multiplier method

Taking the partial derivatives of L with respect to **w** and b.

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = \mathbf{0}$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{m} \alpha_i y_i = 0$$

We find that

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

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

Lagrange Method

Let us substitute by these value into L:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

This is the Wolfe **dual** Lagrangian function. The optimization problem is now called the Wolfe dual problem:

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to} \quad & \alpha_i \geq 0, \text{ for any } i = 1, \ldots, m \\ & \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned}$$

The main advantage of the Wolfe dual problem over the Lagrangian problem is that the objective function now depends only on the Lagrange multipliers. Support vectors are examples having a positive Lagrange multiplier.

Wolfe Dual Problem

Change to minimize:

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i=1}^{m}\alpha_i \\ \text{subject to} \quad & -\alpha_i \leq 0, \ \text{for any } i = 1, \ldots, m \\ & \sum_{i=1}^{m}\alpha_i y_i = 0 \end{aligned}$$

Construct a vectorized version of the Wolfe dual problem

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & \frac{1}{2}\alpha^T(\mathbf{y}\mathbf{y}^T K)\alpha - \alpha \\ \text{subject to} \quad & -\alpha \leq 0, \\ & \mathbf{y} \cdot \alpha = 0 \end{aligned}$$

A QP solver can be used to solve this quadratic programming problem.

Wolfe Dual Problem

Compute w

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

Compute b

$$b = y_i - \mathbf{w} \cdot \mathbf{x}_i$$

Two versions of hypothesis functions

$$h(\mathbf{x}_i) = \operatorname{sign}(\mathbf{w} \cdot \mathbf{x}_i + b)$$

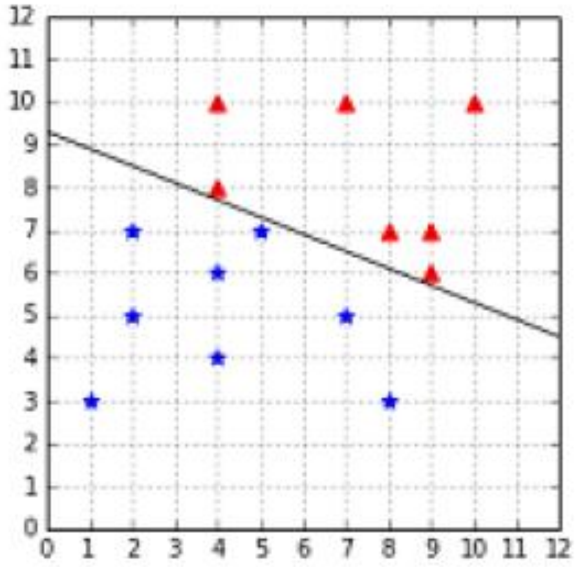
$$h(\mathbf{x}_i) = \operatorname{sign}\left(\sum_{j=1}^{S} \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}_i) + b \right)$$

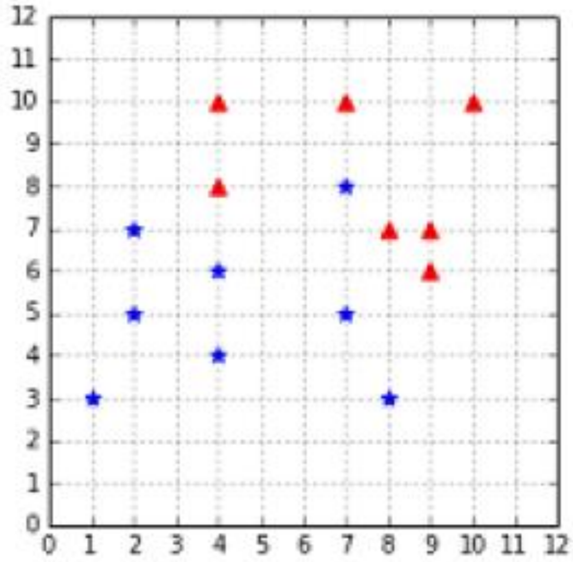
This formulation of the SVM is called the **hard margin SVM**. It cannot work when the data is not linearly separable.

Soft Margin SVM

The **soft margin SVM is** able to work when data is non-linearly separable. There are two cases:

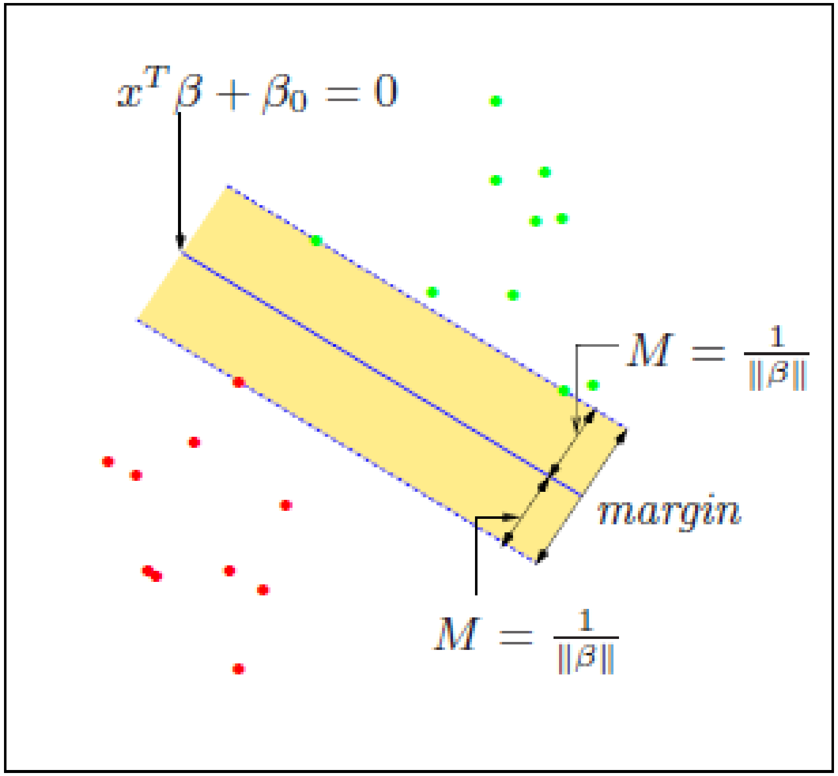
- the outlier can be closer to the other points than most of the points of its class, thus reducing the margin,
- it can be among the other points and break linear separability

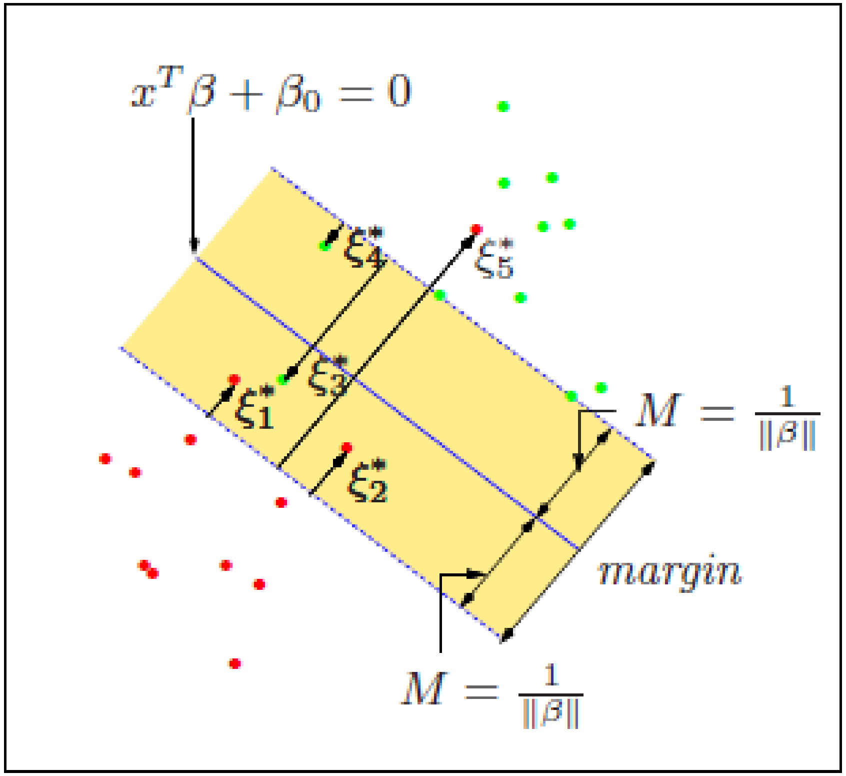




Slack Variables

Introduce Slack Variables ε_i to the optimization model.





Soft Margin SVM Model

Introduce **slack variables** to penalize outliers, this leads us to the **soft margin formulation**:

$$\begin{aligned} \underset{\mathbf{w},b,\zeta}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\zeta_i \\ \text{subject to} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \zeta_i \\ & \zeta_i \geq 0 \quad \text{for any } i = 1,\dots,m \end{aligned}$$

We need to maximize the same **Wolfe dual** as before, under a slightly different constraint:

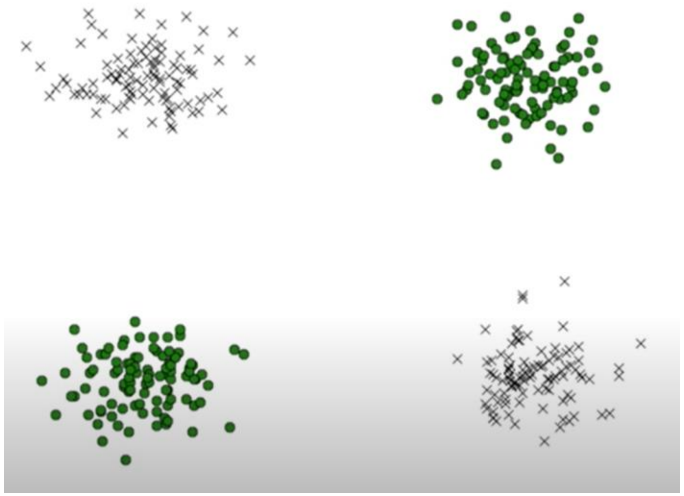
$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, \text{ for any } i = 1,\dots,m \\ & \sum_{i=1}^{m}\alpha_i y_i = 0 \end{aligned}$$

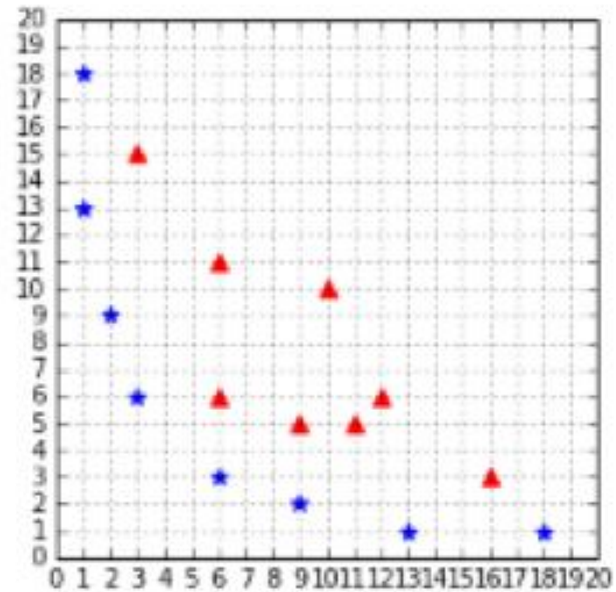
This constraint is often called the box constraint because the vector α is constrained to lie inside the box with side length C.

KERNEL TRICK

Enlarge Feature Space

Can we classify non-linearly separable data?

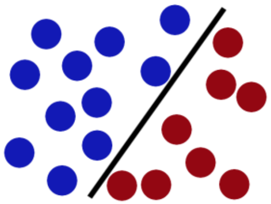




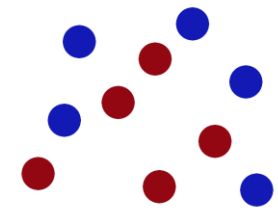
Kernel Function:
$K(x, y) = <f(x), f(y)>$. where K is the kernel function, x, y are p dimensional inputs. f is a map from p-dimension to m-dimension space. $<x, y>$ denotes the dot product, usually m is much larger than n.

Feature space transformations, i.e., called the **kernel trick**

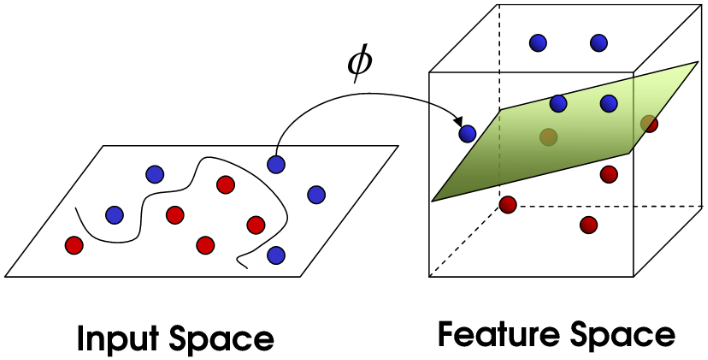
Enlarge Feature Space



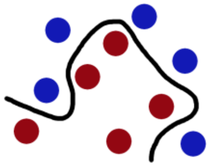
We have 2 colors of balls on the table that we want to separate. We get a stick and put it on the table, this works pretty well, right?



A villain has seen how good you are with a stick so he gives you a new challenge.

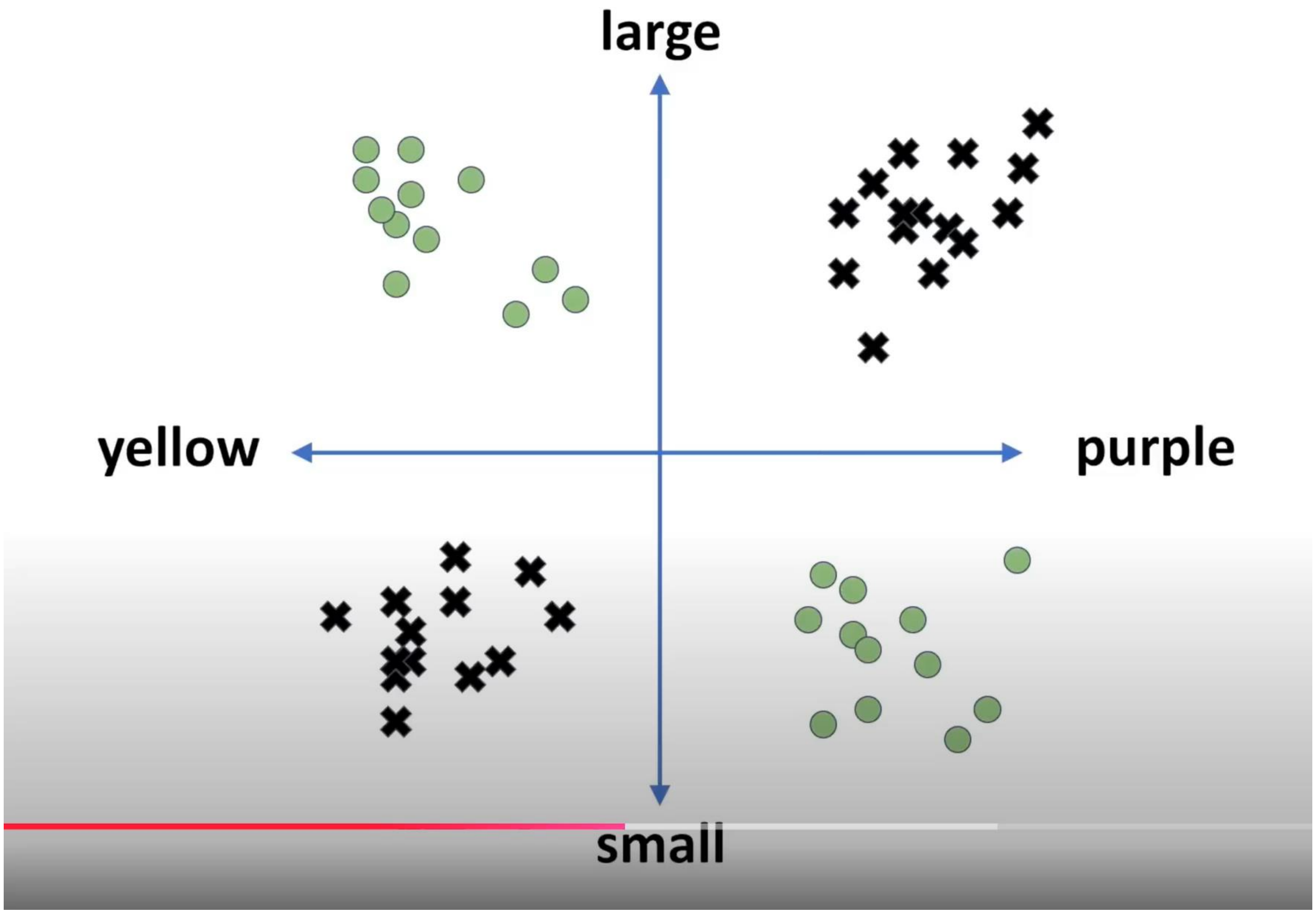


There's no stick in the world that will let you split those balls well, so what do you do? You flip the table of course! Throwing the balls into the air. Then, with your pro ninja skills, you grab a sheet of paper and slip it between the balls.



Now, looking at the balls from where the villain is standing, the balls will look split by some curvy line.

Kernel Trick Enlarging Feature Space



Soft-margin Wolfe Dual Model

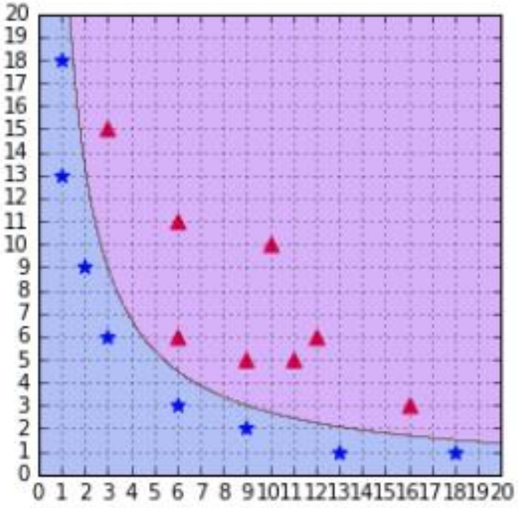
We can then rewrite the soft-margin dual problem with kernel function:

$$\begin{aligned}
\underset{\alpha}{\text{minimize}} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{m}\alpha_i \\
\text{subject to} \quad & 0 \le \alpha_i \le C, \ \text{for any } i = 1,\ldots,m \\
& \sum_{i=1}^{m}\alpha_i y_i = 0
\end{aligned}$$

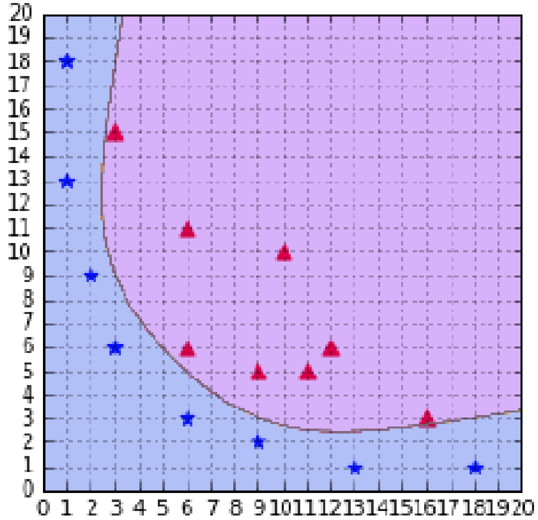
The hypothesis function to use the kernel function:

$$h(\mathbf{x}_i) = \text{sign}\left(\sum_{j=1}^{S}\alpha_j y_j K(\mathbf{x}_j,\mathbf{x}_i) + b\right)$$

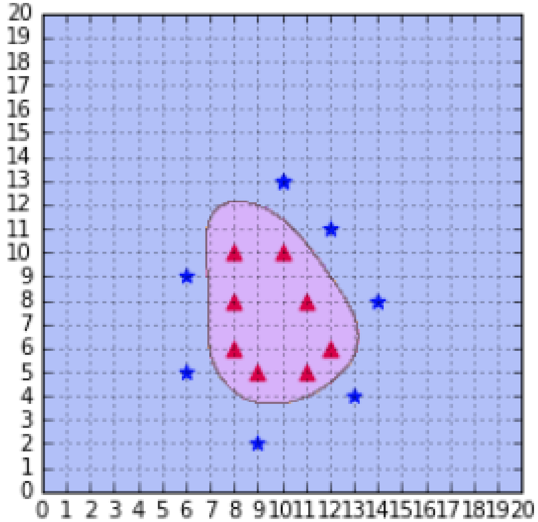
SVM with Kernel Functions



a polynomial kernel
(degree=2)



polynomial kernel
(degree = 6)



RBF kernel