BITS Pilani
Pilani Campus

# Support Vector Machines

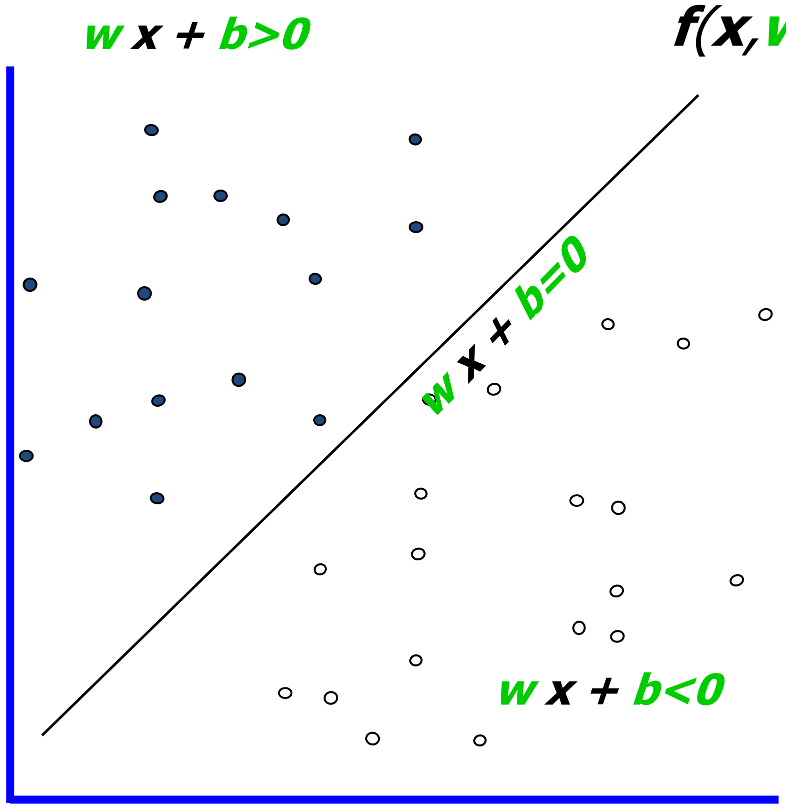
MFDS Team

# Topics to be covered

- Linear Classifiers
- Maximum Margin Classification
- Linear SVM
- SVM optimization problem
- Soft Margin SVM

# Linear Classifiers

- denotes +1
- denotes -1



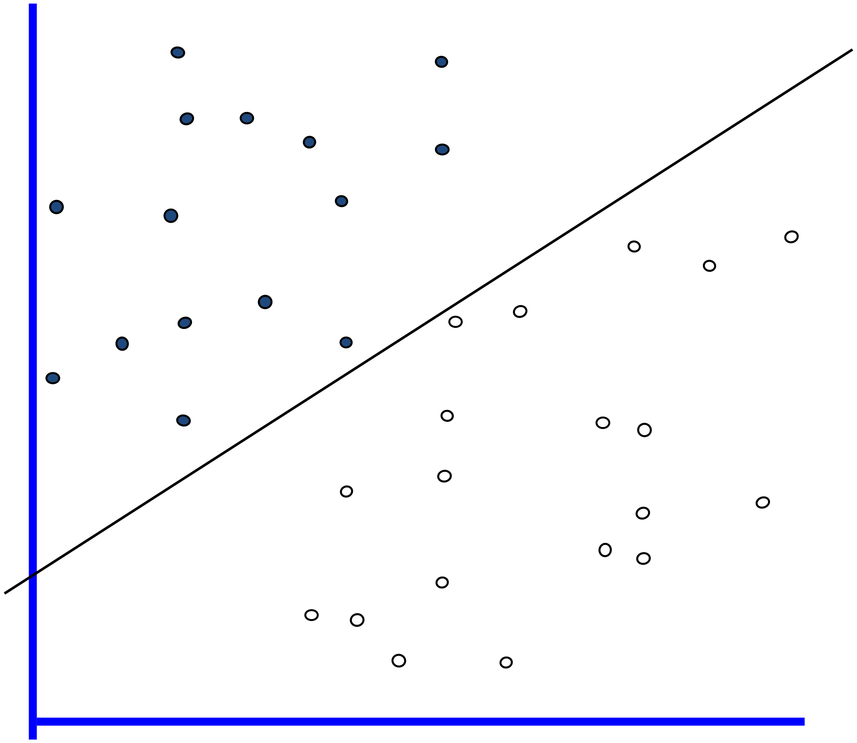

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

How would you classify this data?

# Linear Classifiers

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

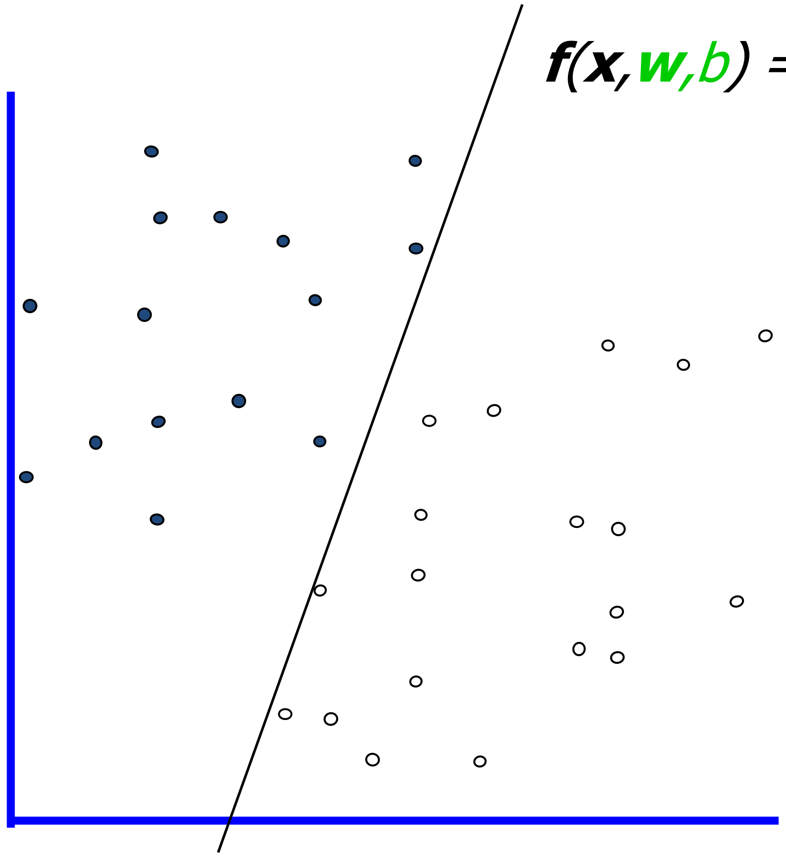
- denotes +1
- denotes -1



How would you classify this data?

# Linear Classifiers

- denotes +1
- denotes -1



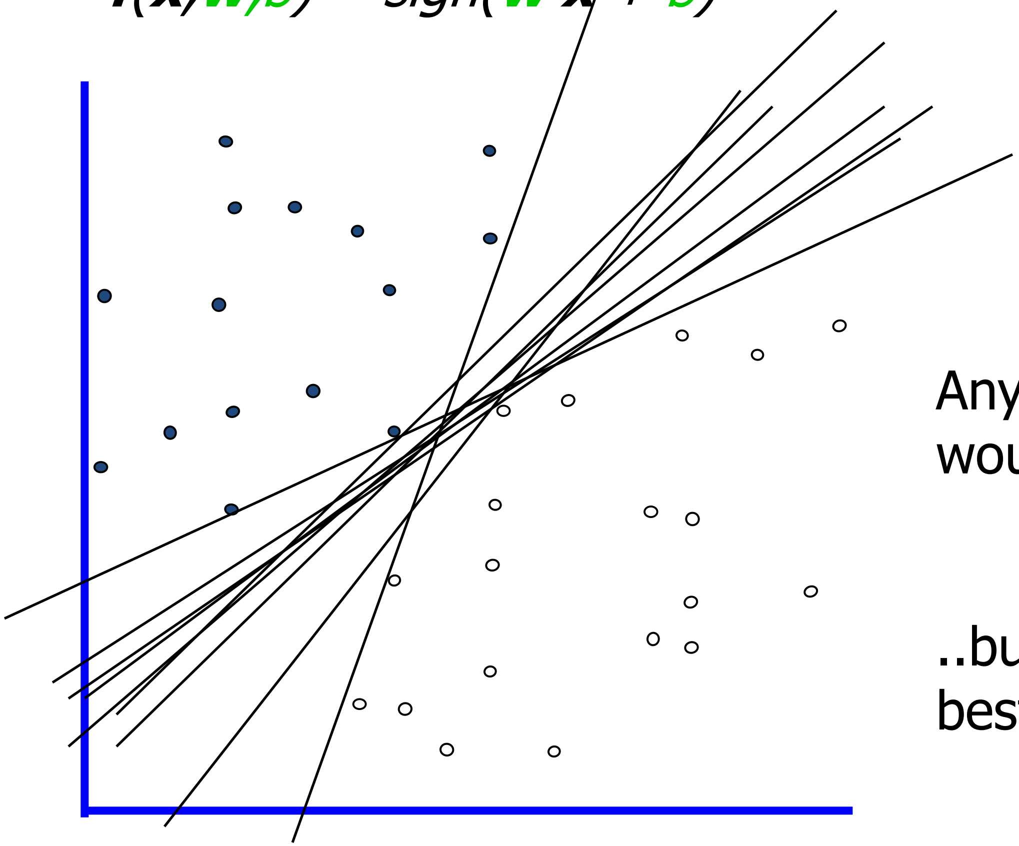
$$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}\ \mathbf{x} + b)$$

How would you classify this data?

# Linear Classifiers

$$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\ \mathbf{x} + b)$$

- denotes +1
- denotes -1



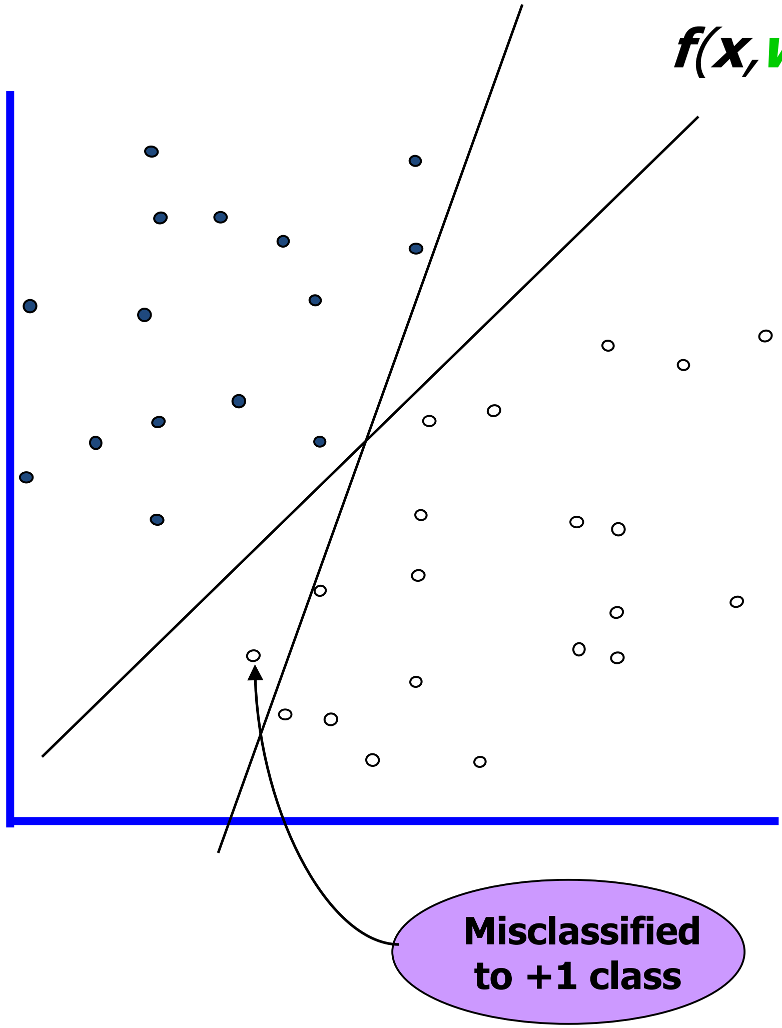
Any of these would be fine..

..but which is best?

# Linear Classifiers

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

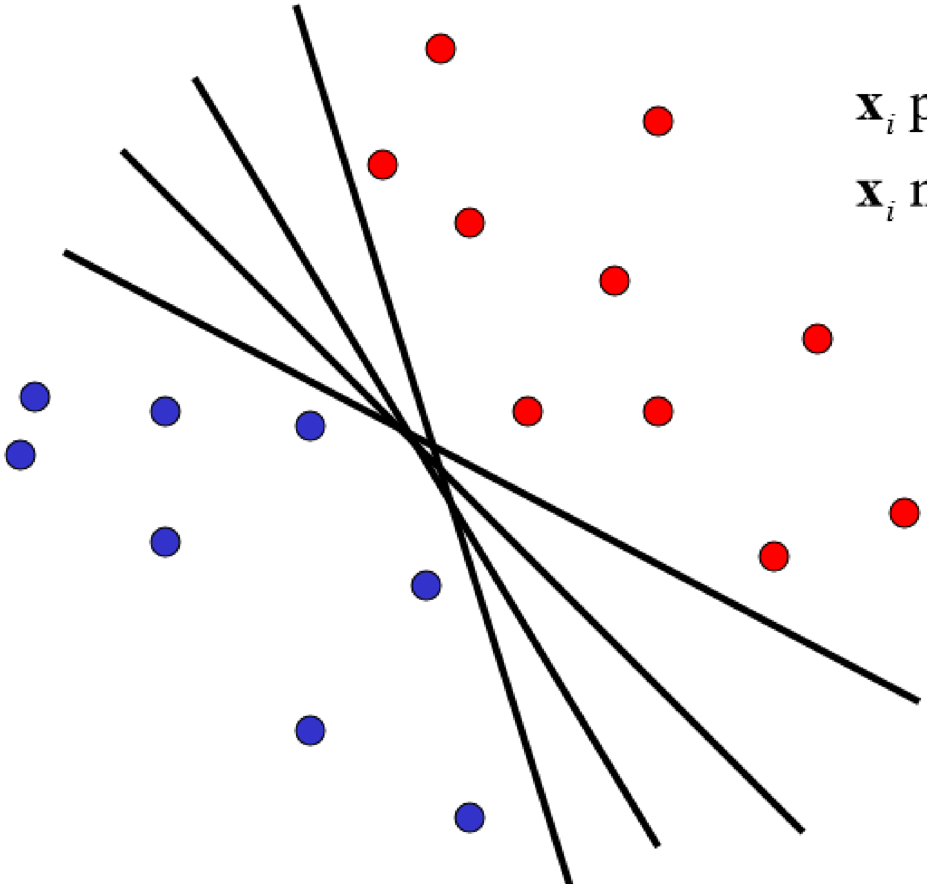
- denotes +1
- denotes -1




How would you classify this data?

# Linear Classifier

- Find linear function to separate positive and negative examples



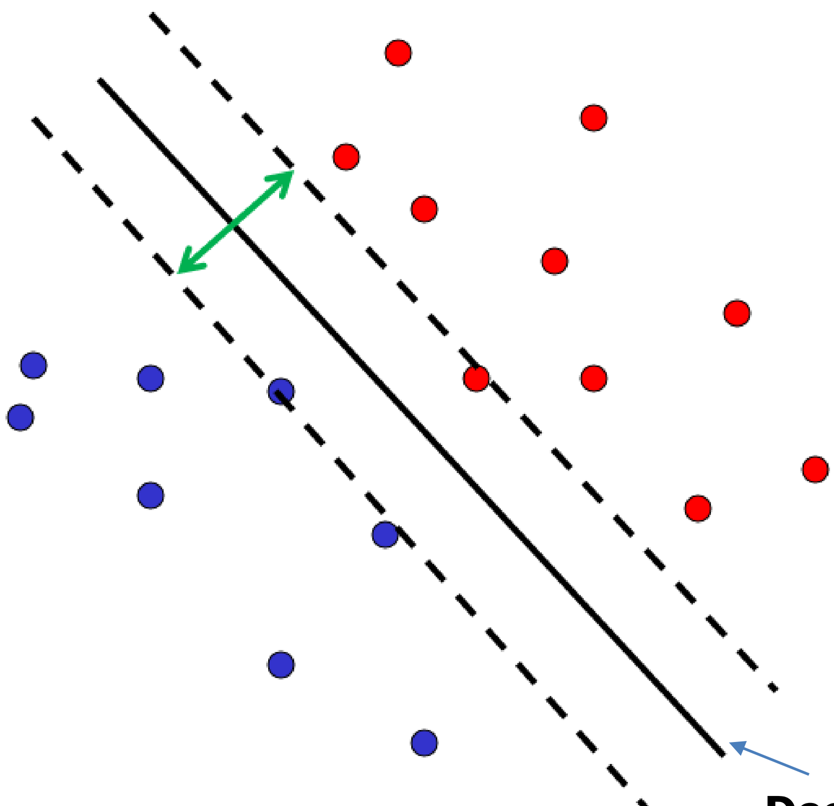
$$\mathbf{x}_i \text{ positive}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \text{ negative}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

Which line is best?

. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

# Linear Classifier

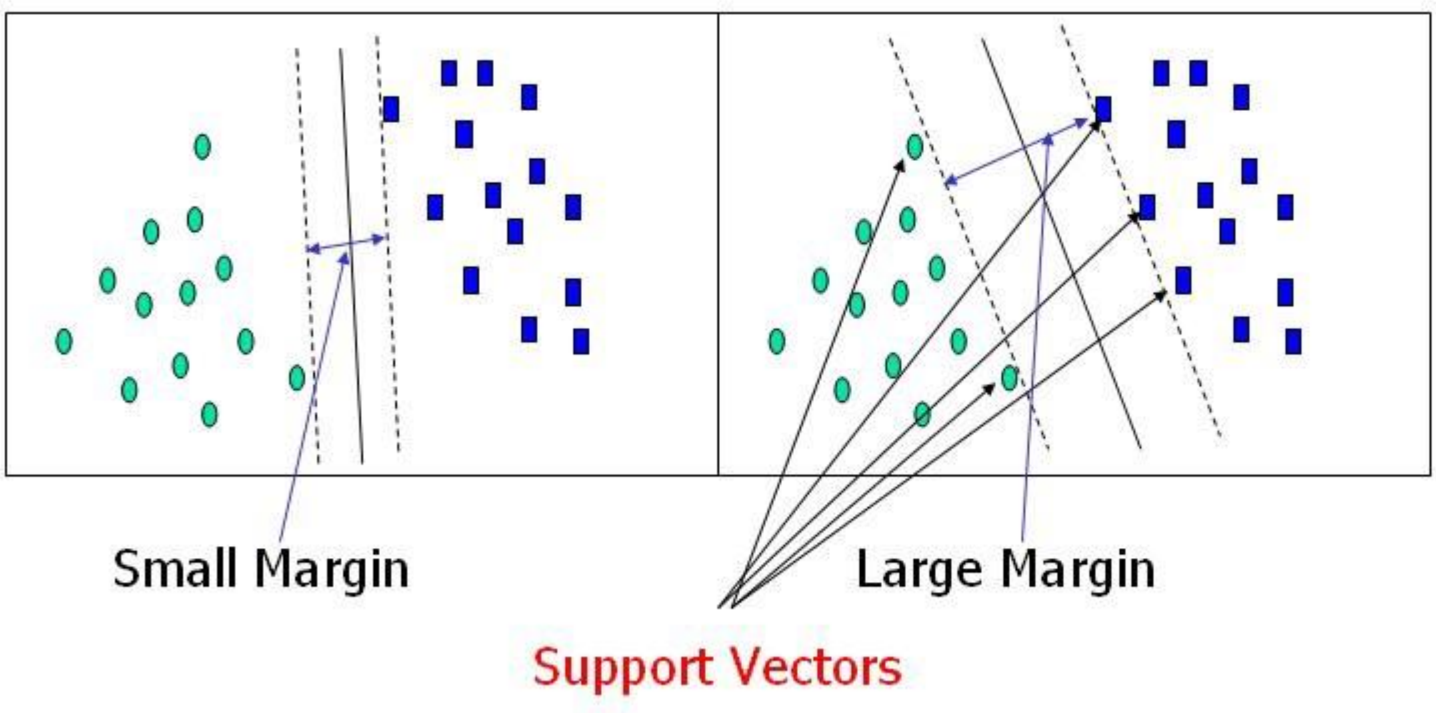


- Discriminative classifier based on *optimal separating line (for 2d case)*
- Maximize the *margin* between the positive and negative training examples

**Decision Boundary**

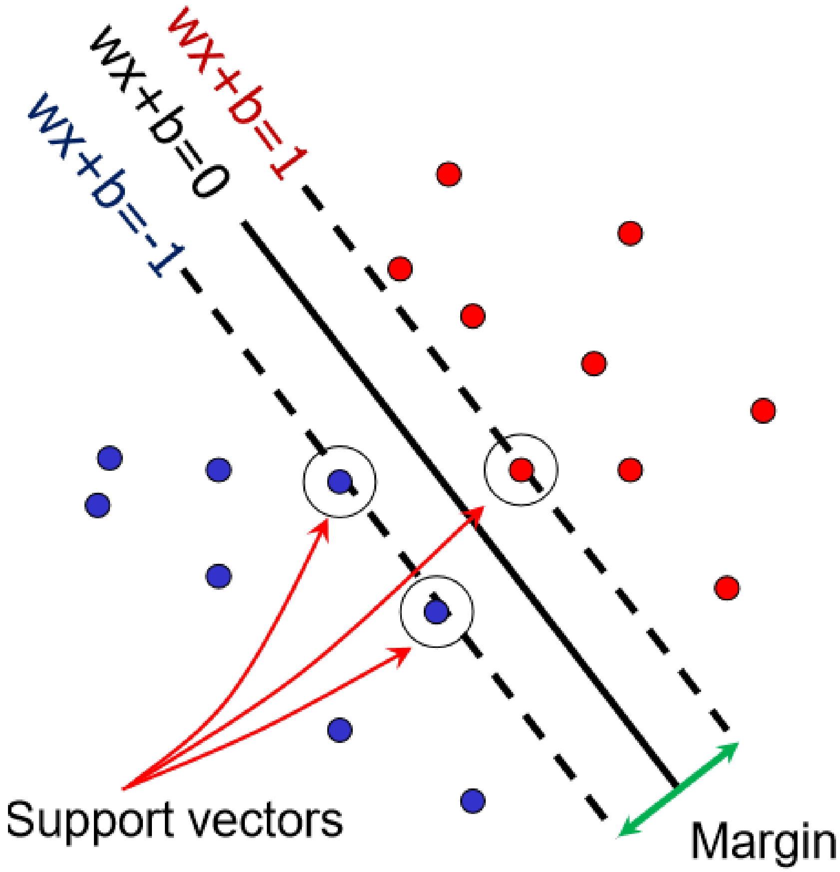
C. Burges, [A Tutorial on Support Vector Machines for Pattern Recognition](), Data Mining and Knowledge Discovery, 1998

# Large margin and support vectors



# Support Vector Machines

- Want line that maximizes the margin.




$\mathbf{x}_i$ positive $(y_i = 1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

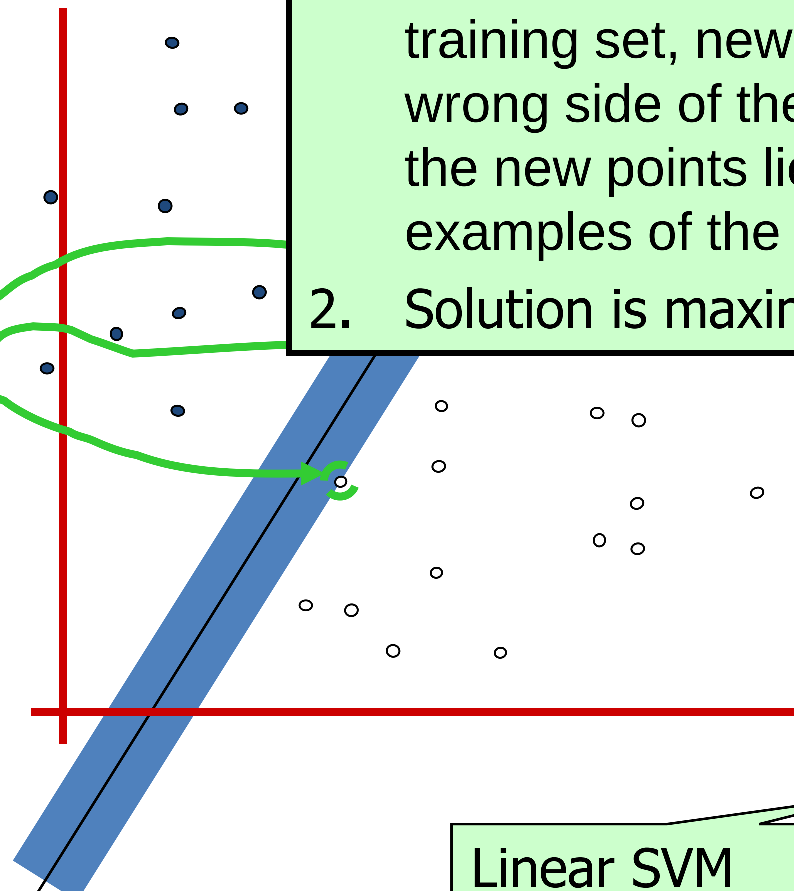
$\mathbf{x}_i$ negative $(y_i = -1)$: $\quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support vectors, $\quad \mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

# Maximum Margin

denotes +1

denotes -1



1. If hyperplane is oriented such that it is close to some of the points in your training set, new data may lie on the wrong side of the hyperplane, even if the new points lie close to training examples of the correct class.
2. Solution is maximizing the margin

with the, maximum margin.

Support Vectors are those datapoints that the margin pushes up against

This is the simplest kind of SVM (Called an LSVM)

Linear SVM

# Support Vectors

- Geometric description of SVM is that the max-margin hyperplane is completely determined by those points that lie nearest to it.
- Points that lie on this margin are the support vectors.
- The points of our data set which if removed, would alter the position of the dividing hyperplane

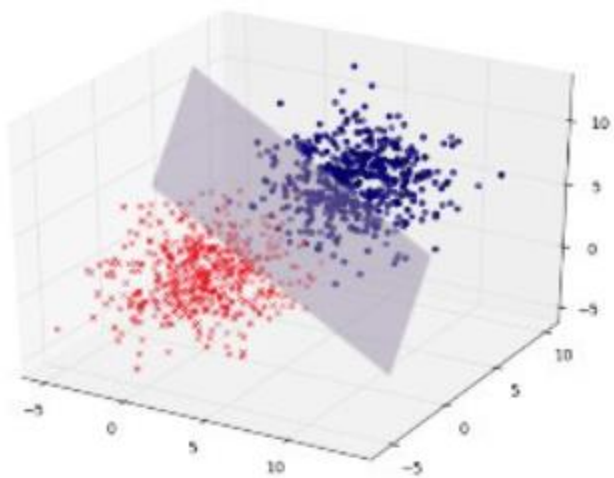
# Example

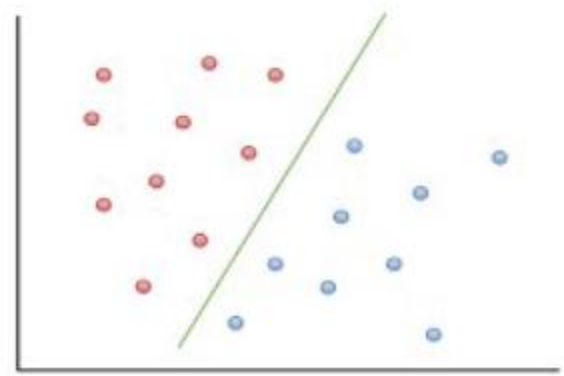
$$\mathbf{w}^T\mathbf{x} = 0$$

Hyperplane

$$y = ax + b$$

Line





# Weight vector is perpendicular to the hyperplane

Consider the points $x_a$ and $x_b$,
which lie on the decision boundary.
This gives us two equations:

$w^T x_a + b = 0$

$w^T x_b + b = 0$

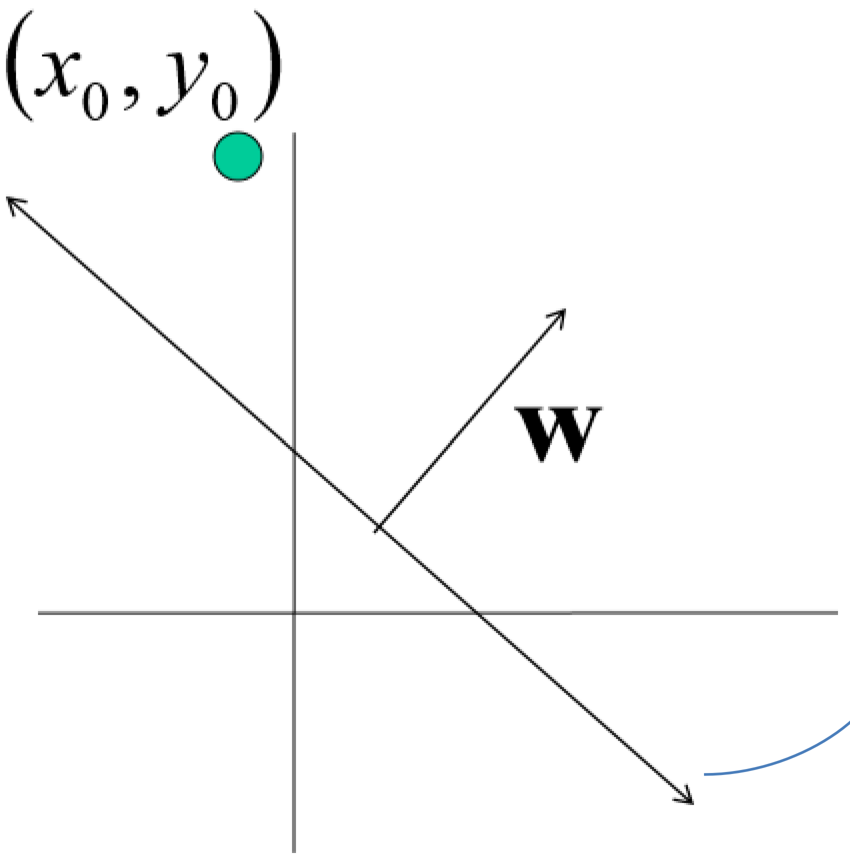
Subtracting these two equations gives us

$w^T.(x_a - x_b) = 0$

Note that the vector $x_a - x_b$ lies on the decision boundary, and it is directed from $x_b$ to $x_a$.

Since the dot product $w^T.(x_a - x_b)$ is zero, $w^T$ must be orthogonal to $x_a - x_b$ and in turn, to the decision boundary.

# Line with 2 features: R2



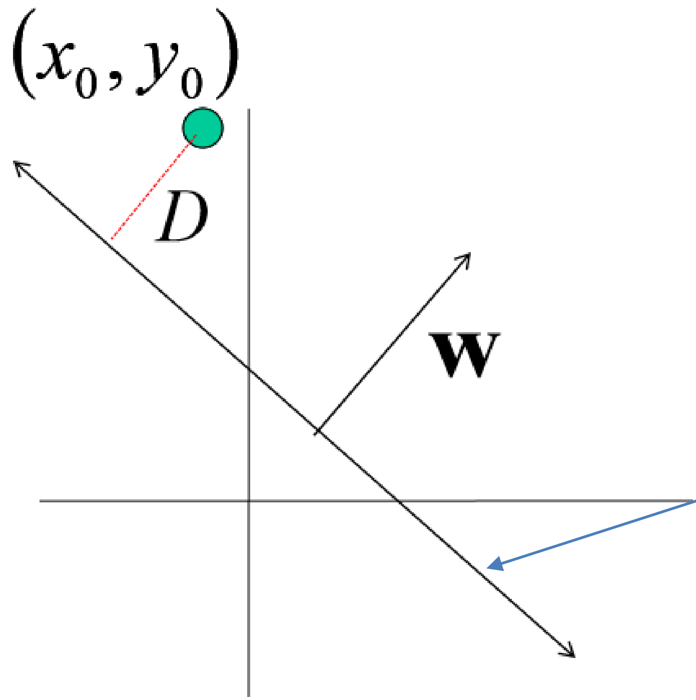

Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

$$\updownarrow$$

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

# Line with 2 features: R2




Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\quad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

$$\Updownarrow$$

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

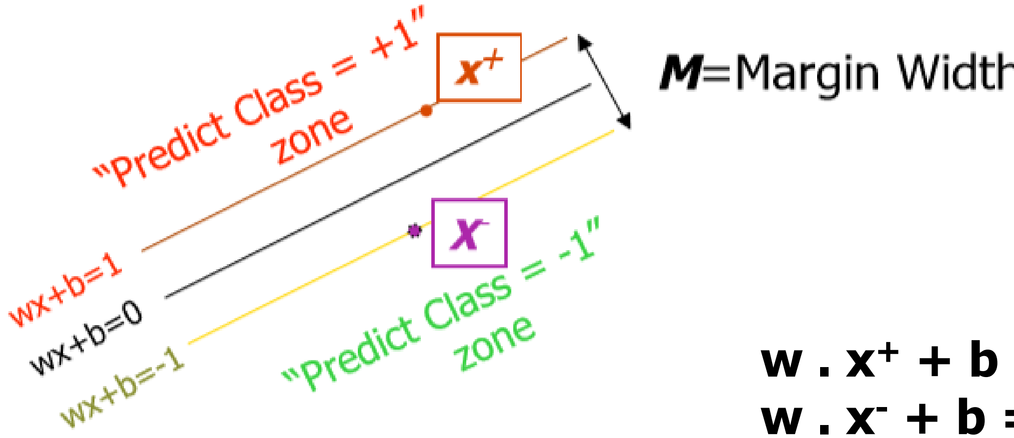
$$D = \frac{|ax_0 + cy_0 + b|}{\sqrt{a^2 + c^2}} = \frac{|\mathbf{w}^{\mathrm{T}}\mathbf{x} + b|}{\|\mathbf{w}\|}$$

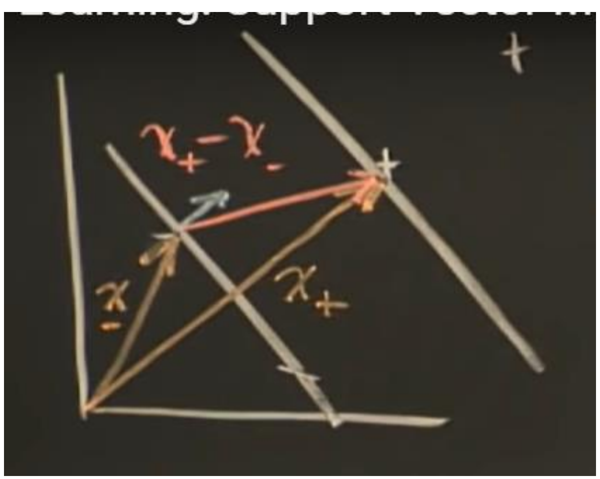
distance from point to line

Kristen Grauman

https://brilliant.org/wiki/dot-product-distance-between-point-and-a-line/

# Linear SVM Mathematically



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

**Margin width**

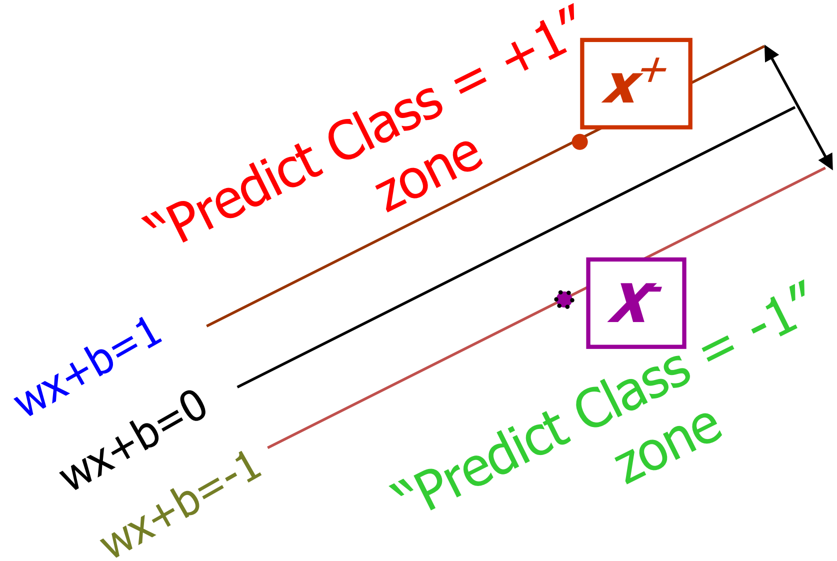
$$= x^+ - x^- \; . \frac{w}{||w||}$$

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

$$= (1-b) - (-1-b) \; / \; ||w||$$

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

# Linear SVM Mathematically




***M***=Margin Width

What we know:

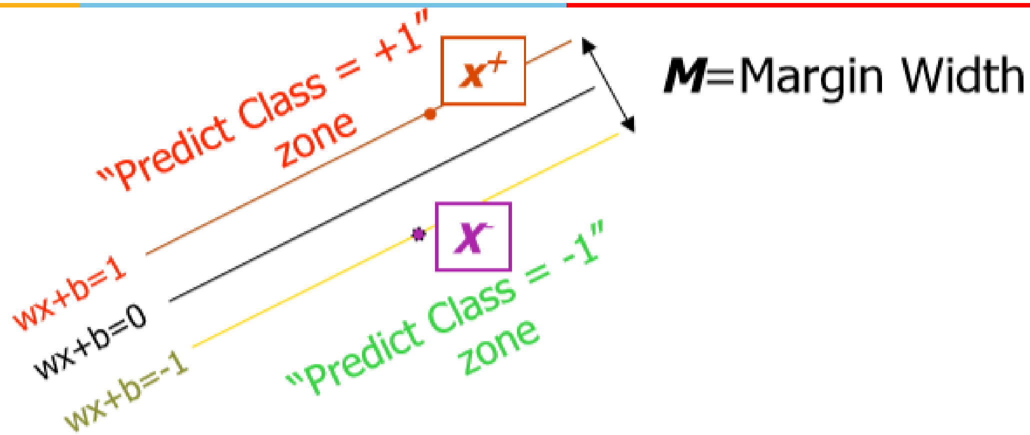
- $\mathbf{w} \cdot \mathbf{x}^{+} + b = +1$
- $\mathbf{w} \cdot \mathbf{x}^{-} + b = -1$

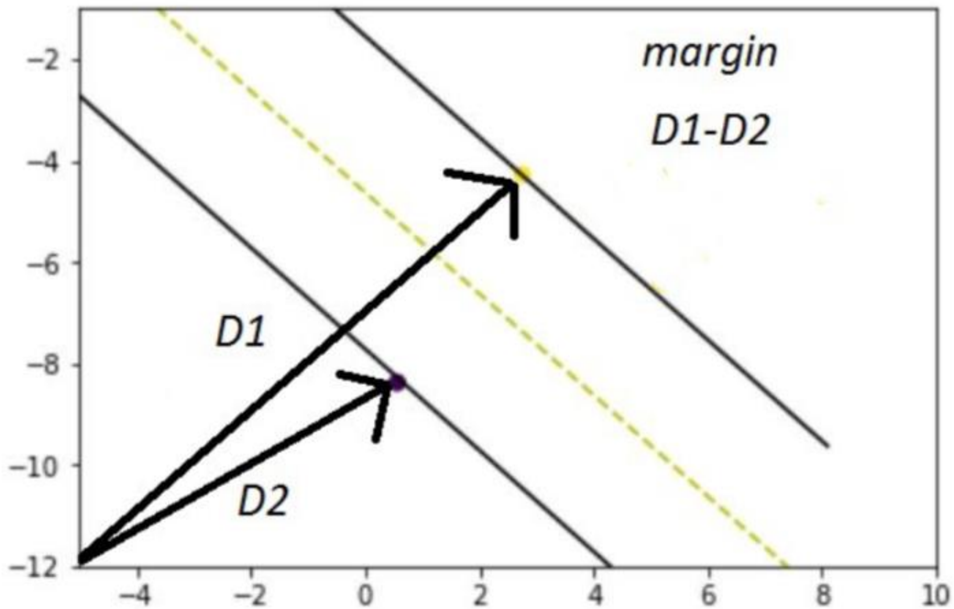
Distance between lines given by solving linear equation:

Maximize margin: $M = \dfrac{2}{||w||}$

Equivalent to minimize: $\frac{1}{2}||w||^2$

# Linear SVM Mathematically




$D1 = w^T x+b=1 \qquad w^T x+b-1=0$

$D2= w^T x+b=-1 \qquad w^T x+b+1=0$

$w^T x+b-1 \; - \; w^T x+b+1$

$\Downarrow$ *Solve algebraically*

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

# Solving the Optimization Problem

1. Maximize margin $2/||\mathbf{w}||$
2. Correctly classify all training data points:

$$\mathbf{x}_i \text{ positive } (y_i = 1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$$

$$\mathbf{x}_i \text{ negative } (y_i = -1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$$

*Quadratic optimization problem*:

Find $\mathbf{w}$ and b such that

$\boldsymbol{\Phi}(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2$ is minimized;

and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i (\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

$y_i\ (\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

$+1(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

$-1(\mathbf{w}^T\mathbf{x}_i + b) \leq 1$

same as $(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

# Solving the Optimization Problem

Find **w** and b such that
$\mathbf{\Phi}(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2$ is minimized; Type equation here.
and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

← Primal

- **Need to optimize a *quadratic* function subject to *linear inequality* constraints.**
- **All constraints in SVM are linear**
- **Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them.**
- **The solution involves constructing a *unconstrained problem* where a *Lagrange multiplier* $\alpha_i$ is associated with every constraint in the primary problem:**

# Optimization Problem

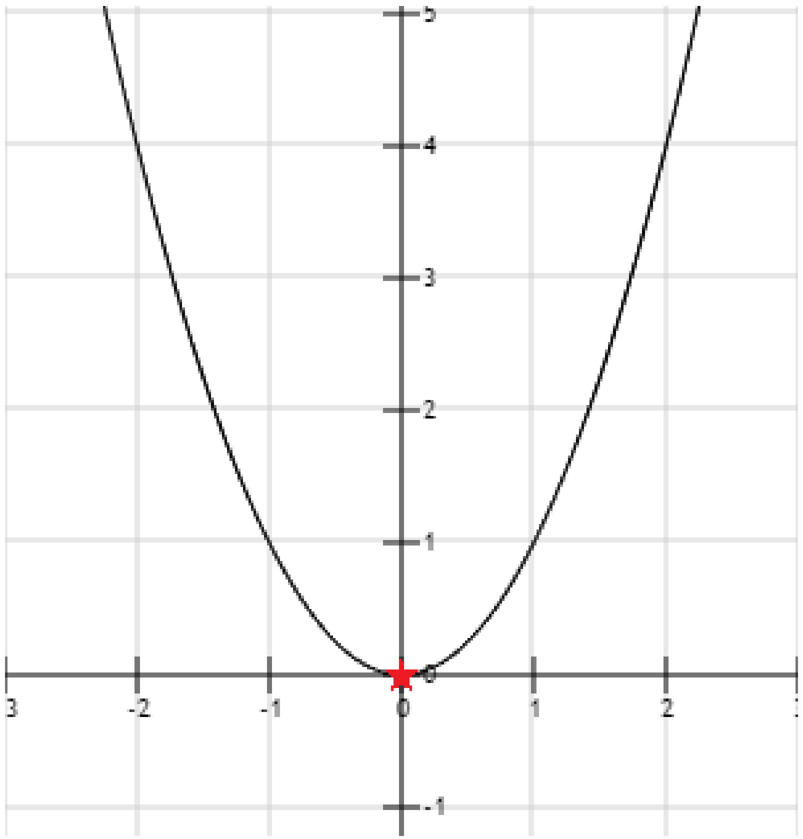
- Optimization problem is typically written:

$$\text{Minimize } f(x)$$
$$\text{subject to}$$
$$g_i(x) = 0, \quad i=1,\dots,p$$
$$h_i(x) <= 0, \quad i=1,\dots,m$$

- $f(x)$ is called the objective function
- By changing $x$ (the optimization variable) we wish to find a value $x*$ for which $f(x)$ is at its minimum.
- p functions of $g_i$ define equality constraints and
- m functions $h_i$ define inequality constraints.
- The value we find MUST respect these constraints!

# Unconstrained Optimization

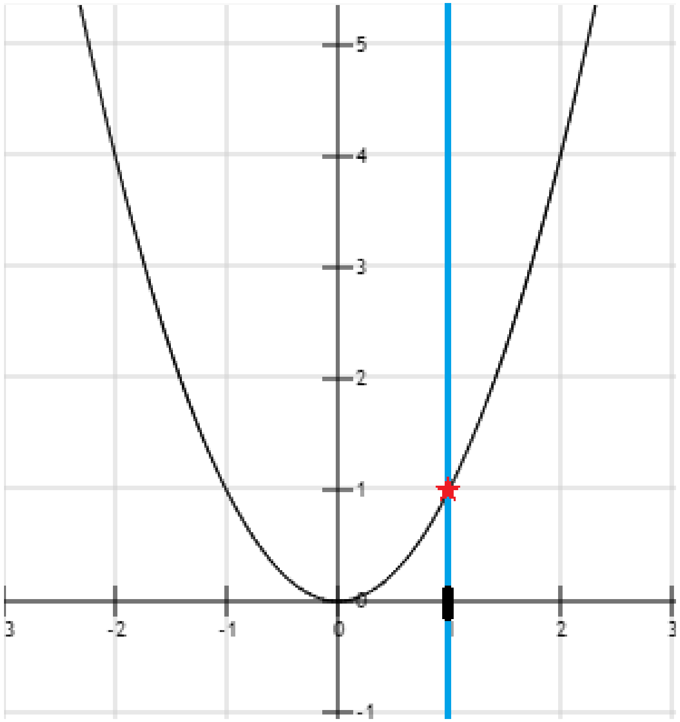
- Minimize $x^2$



# Constrained Optimization -Equality Constraint

Minimize $x^2$

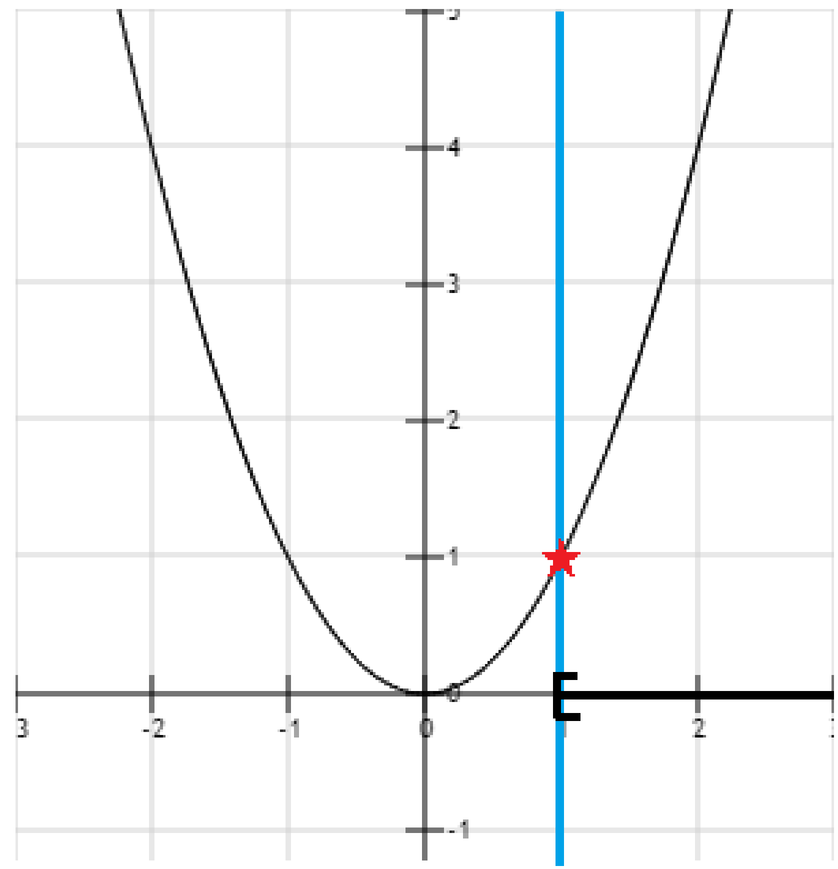
Subject to $x = 1$



# Constrained Optimization -Inequality Constraint

Minimize $x^2$

Subject to x >= 1



# Constrained optimization

- We can also have mix equality and inequality constraints together.
- Only restriction is that if we use contradictory constraints, we can end up with a problem which does not have a feasible set

Minimize $x^2$

Subject to

$x = 1$

$x < 0$

Impossible for x to be equal 1 and less than zero at the same

# Constrained optimization

- A solution is an assignment of values to variables.
- A feasible solution is an assignment of values to variables such that all the constraints are satisfied.
- The objective function value of a solution is obtained by evaluating the objective function at the given solution.
- An optimal solution (assuming minimization) is one whose objective function value is less than or equal to that of all other feasible solutions.

# Lagrange Multipliers

- **How do we find the solution to an optimization problem with constraints?**
- Constrained maximization (minimization) problem is rewritten as a Lagrange function whose optimal point is a saddle point, i.e. a global maximum (minimum)
- *Lagrange function use Lagrange multipliers is a strategy for finding the local maxima and minima of a function subject to constraints*

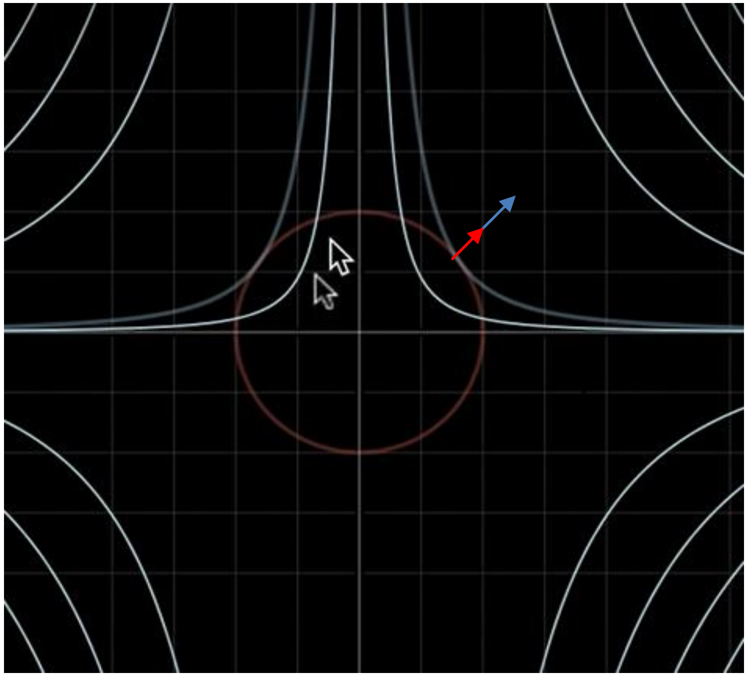
# Constrained to Unconstrained Optimization: Lagrange Multiplier

**Maximize**

$$f(x,y) = x^2 y$$

**Subject to**

$$g(x, y) : x^2 + y^2 = 1$$



- Maximum of f(x,y) under constraint g(x, y) is obtained when their gradients point to same direction (when they are tangent to each other).
- Introduce a Lagrange multiplier $\lambda$ for the equality constraint
- Mathematically,

$$\nabla f(x,y) = \lambda \nabla g(x,y)$$

# Example:

$$\max_{x,y} xy \text{ subject to } x + y = 6$$

- Introduce a Lagrange multiplier λ for constraint
- Construct the Lagrangian

$$L(x, y) = xy - \lambda(x + y - 6)$$

- Stationary points

$$\frac{\partial L(x,y)}{\partial \lambda} = x + y - 6 = 0$$

$$\left.\begin{aligned}\frac{\partial L(x,y)}{\partial x} &= y - \lambda = 0 \\ \frac{\partial L(x,y)}{\partial y} &= x - \lambda = 0\end{aligned}\right\} \Rightarrow x = y = \lambda$$

$$\Rightarrow x = y = 3$$

x and y values remain same even if you take +λ or -λ for equality constraint

2 x = 6
x = y = 3
λ = 3

# Karush–Kuhn–Tucker (KKT) theorem

- KKT approach to nonlinear programming (quadratic) generalizes the method of Lagrange multipliers, which allows only equality constraints.
- KKT allows inequality constraints

# Karush-Kuhn-Tucker (KKT) conditions



- Start with

  max f(x) subject to

  $$g_i(x) = 0 \text{ and } h_j(x) \geq 0 \text{ for all } i, j$$

- Make the Lagrangian function

  $$\mathcal{L} = f(x) - \sum_i \lambda_i g_i(x) - \sum_j \mu_j h_j(x)$$

- Take gradient and set to 0 – but other conditions also.

# KKT conditions

- Make the Lagrangian function

$$\mathcal{L} = f(x) - \sum_i \lambda_i g_i(x) - \sum_j \mu_j h_j(x)$$

- Necessary conditions to have a minimum are

$$\begin{aligned} \nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) &= 0 \\ g_i(x^*) &= 0 \text{ for all } i \\ h_j(x^*) &\geq 0 \text{ for all } j \\ \mu_j &\geq 0 \text{ for all } j \\ \mu_j^* h_j(x^*) &= 0 \text{ for all } j \end{aligned}$$

# Solving the Optimization Problem

Find $\mathbf{w}$ and b such that

$\mathbf{\Phi}(\mathbf{w}) = \frac{\mathbf{1}}{\mathbf{2}}||\mathbf{w}||^2$ is minimized;

and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- **Need to optimize a *quadratic* function subject to *linear inequality* constraints.**
- **The solution involves constructing a *dual problem* where a *Lagrange multiplier* $\alpha_i$ is associated with every constraint in the primal problem**

# Solving the Optimization Problem

- **The solution involves constructing a *dual problem* where a *Lagrange multiplier* $\boldsymbol{\alpha_i}$ is associated with every constraint in the primary problem:**

$$\boxed{L(w, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \Sigma\, \alpha_i\, [y_i\, (\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) - 1]}$$

- **Taking partial derivative with respect to w , $\frac{\partial L}{\partial w}$ = 0**
  - w - $\Sigma\, \alpha_i y_i\, \mathbf{x_i} = \mathbf{0}$
  - w $= \Sigma\, \alpha_i y_i\, \mathbf{x_i}$
- **Taking partial derivative with respect to b, $\frac{\partial L}{\partial b}$ = 0**
  - $-\,\Sigma\, \alpha_i\, y_i = 0$
  - $\Sigma\, \alpha_i\, y_i = 0$

# Solving the Optimization Problem

$$\boxed{L(w, b, \alpha_i) = \frac{1}{2}||\mathbf{w}||^2 - \Sigma\, \alpha_i\, [y_i\, (\mathbf{w} \,.\, \mathbf{x_i} + b) - 1]}$$

- Expanding above equation:

$$L(w, b, \alpha_i) = \frac{1}{2}\, ||\mathbf{w}||^2 - \Sigma\, \alpha_i y_i\, \mathbf{w} \,.\, \mathbf{x_i} - \Sigma\, \alpha_i y_i\, b \;+ \Sigma\, \alpha_i$$

- Substituting w $= \Sigma\, \alpha_i y_i\, \mathbf{x_i}$ and $\Sigma\, \alpha_i\, y_i = 0$ in above equation

$$L(w, b, \alpha_i) = \frac{1}{2}\, (\textstyle\sum_i \alpha_i y_i\, \mathbf{x_i})(\sum_j \alpha_j y_\mathbf{j}\, \mathbf{x_j}) - \; (\sum_i \alpha_i y_i\, \mathbf{x_i})(\sum_j \alpha_j y_\mathbf{j}\, \mathbf{x_j}) + \Sigma\, \alpha_i$$

$$L(w, b, \alpha_i) = \Sigma\, \alpha_i - \frac{1}{2}\, (\textstyle\sum_i \alpha_i y_i\, \mathbf{x_i})(\sum_j \alpha_j y_\mathbf{j}\, \mathbf{x_j})$$

$$L(w, b, \alpha_i) = \Sigma\, \alpha_i - \frac{1}{2}\, (\textstyle\sum_i \sum_j \alpha_i\, \alpha_j y_i\, y_\mathbf{j}\, \mathbf{x_i} \,.\, \mathbf{x_j}\, )$$

# Support Vectors

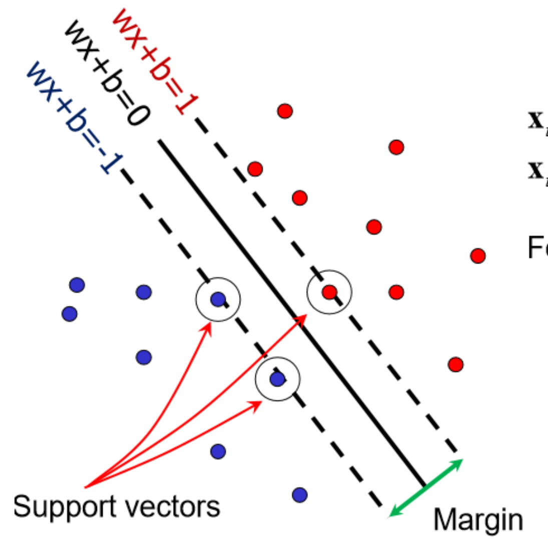
Using KKT conditions :
$\alpha_i [y_i (\mathbf{w^T x_i} + b) -1]=0$

For this condition to be satisfied
either $\alpha_i = 0$ and $y_i (\mathbf{w^T x_i} + b) -1 > 0$
OR
$y_i (\mathbf{w^T x_i} + b) -1=0$ and $\alpha_i > 0$

For support vectors:
$y_i (\mathbf{w^T x_i} + b) -1=0$

For all points other than support vectors:
$\alpha_i = 0$

- Want line that maximizes the margin.




$\mathbf{x}_i$ positive $(y_i = 1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ negative $(y_i = -1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support vectors, $\mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

$$L(w, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \Sigma\, \alpha_i [y_i (\mathbf{w^T x_i} + b) -1]$$

# Solving the Optimization Problem

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

Learned weight

Support vector

# Solving the Optimization Problem

- Solution: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

  $b = y_i - \mathbf{w} \cdot \mathbf{x}_i$ (for any support vector)

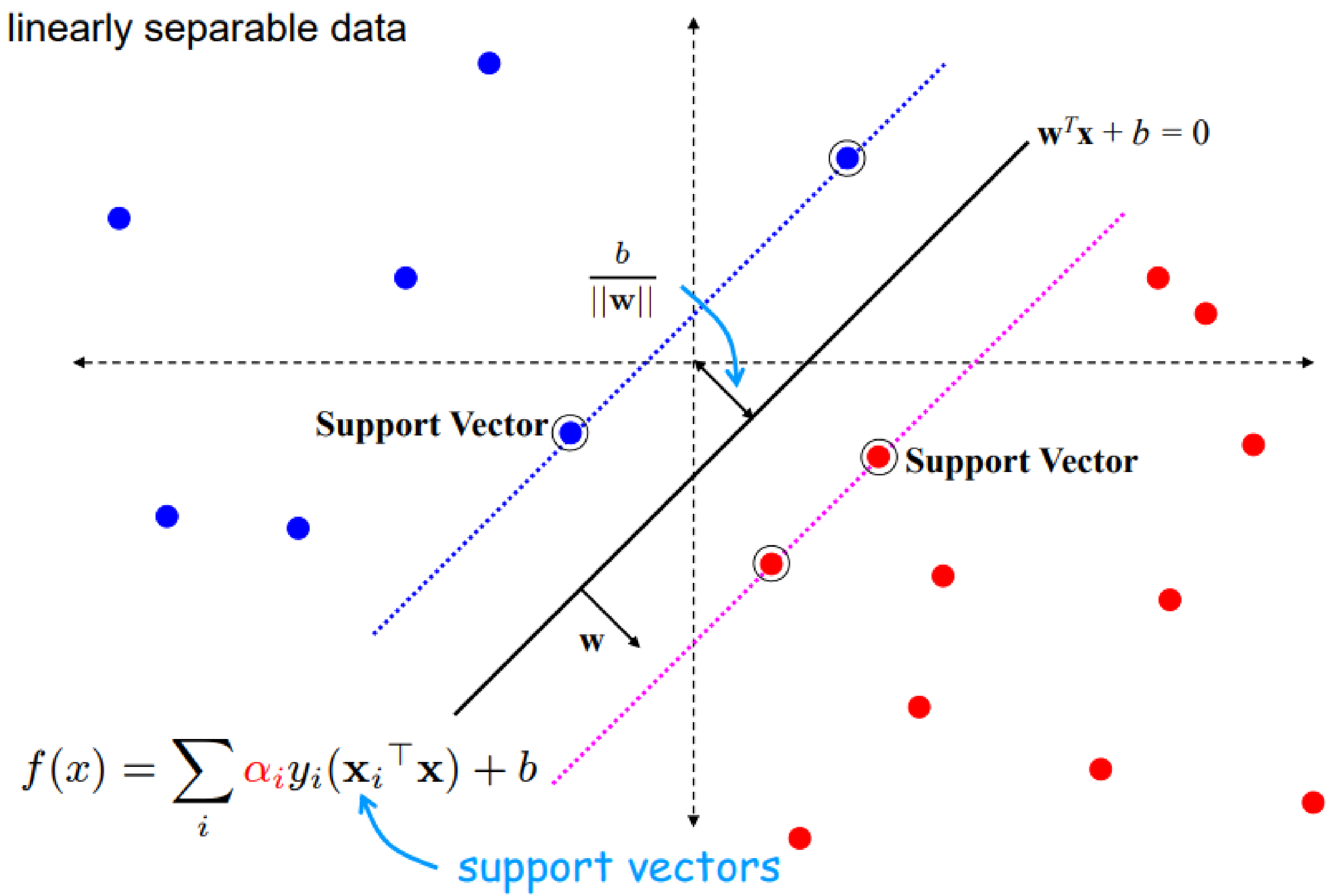
- Classification function:

$$f(x) = \text{sign}\,(\mathbf{w} \cdot \mathbf{x} + \text{b})$$
$$= \text{sign}\left(\sum_{\text{i}} \alpha_{\text{i}} y_i \boxed{\mathbf{x}_i \cdot \mathbf{x}} + b\right)$$

*If f(x) < 0, classify as negative, otherwise classify as positive.*

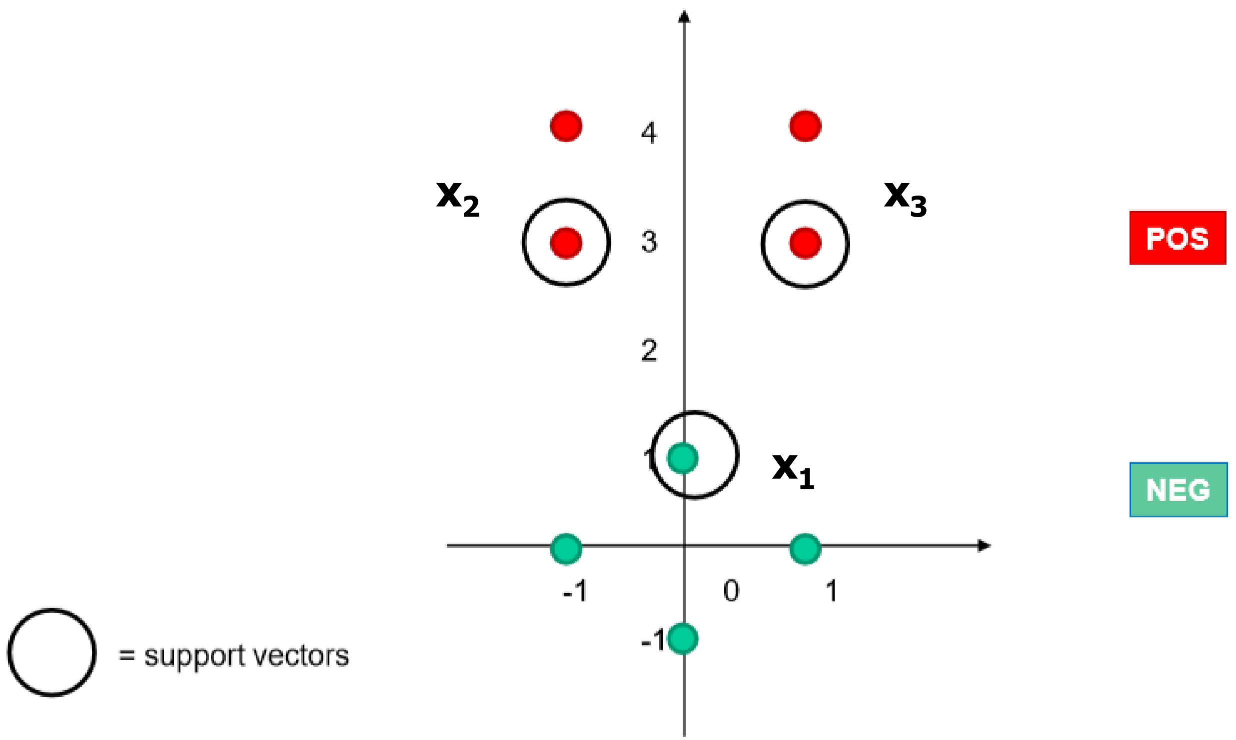
- Notice that it relies on an *inner product* between the test point $\boldsymbol{x}$ and the support vectors $\boldsymbol{x}_i$
- (Solving the optimization problem also involves computing the inner products $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ between all pairs of training points)

# Substituting w in support vectors function




$$f(x) = \sum_i \alpha_i y_i (\mathbf{x}_i{}^\top \mathbf{x}) + b$$

# Example




Example adapted from Dan Ventura

# Solving for α

- We know that for the support vectors, $f(x) = 1$ or $-1$ exactly
- Add a 1 in the feature representation for the bias
- The support vectors have coordinates and labels:
  - $x1 = [0\ 1\ 1]$, $y1 = -1$
  - $x2 = [-1\ 3\ 1]$, $y2 = +1$
  - $x3 = [1\ 3\ 1]$, $y3 = +1$
- Thus we can form the following system of linear equations:

# Solving for α

- System of linear equations:

$\alpha 1\ y1\ \mathrm{dot}(x1, x1) + \alpha 2\ y2\ \mathrm{dot}(x1, x2) + \alpha 3\ y3\ \mathrm{dot}(x1, x3) = y1$

$\alpha 1\ y1\ \mathrm{dot}(x2, x1) + \alpha 2\ y2\ \mathrm{dot}(x2, x2) + \alpha 3\ y3\ \mathrm{dot}(x2, x3) = y2$

$\alpha 1\ y1\ \mathrm{dot}(x3, x1) + \alpha 2\ y2\ \mathrm{dot}(x3, x2) + \alpha 3\ y3\ \mathrm{dot}(x3, x3) = y3$

$-2 * \alpha 1 + 4 * \alpha 2 + 4 * \alpha 3 = -1$

$-4 * \alpha 1 + 11 * \alpha 2 + 9 * \alpha 3 = +1$

$-4 * \alpha 1 + 9 * \alpha 2 + 11 * \alpha 3 = +1$

$\alpha_i\,[-1\,(\mathbf{w} \cdot \mathbf{x_i} + b)] = -1$

$\alpha_i\,[+1\,(\mathbf{w} \cdot \mathbf{x_i} + b)] = 1$

- Solution: α1 = 3.5, α2 = 0.75, α3 = 0.75

# Solving for w and b

We know $w = \alpha_1\, y_1\, x_1 + \ldots + \alpha_N\, y_N\, x_N$ where N = # SVs

Thus w = -3.5 * [0 1 1] + 0.75 [-1 3 1] + 0.75 [1 3 1] = [0 1 -2]

Separating out weights and bias, we have: w = [0 1] and b = -2

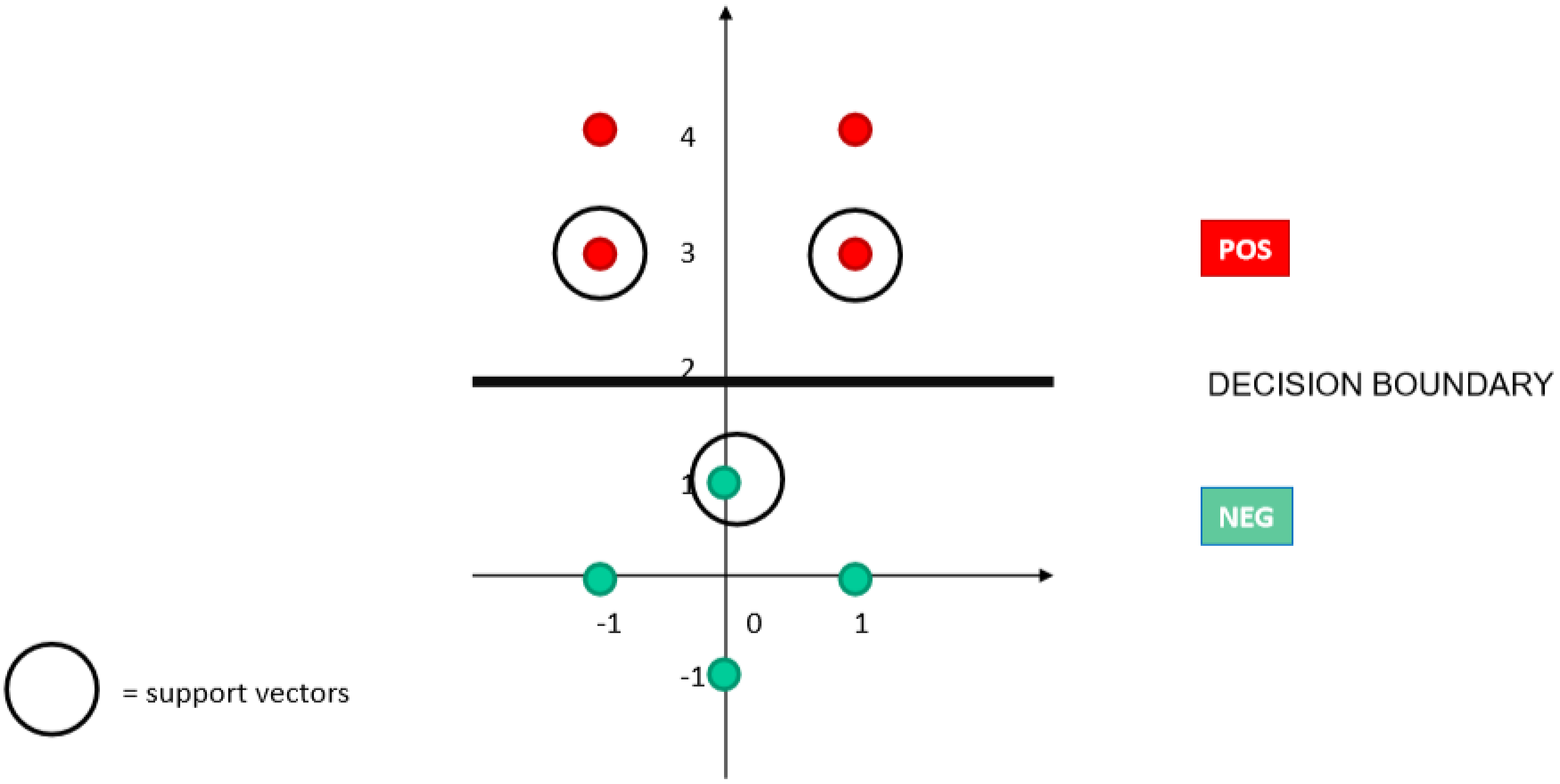
For SVMs, we used this eq for a line: ax + cy + b = 0 where w = [a c]

Thus ax + b = -cy ➔ y = (-a/c) x + (-b/c)

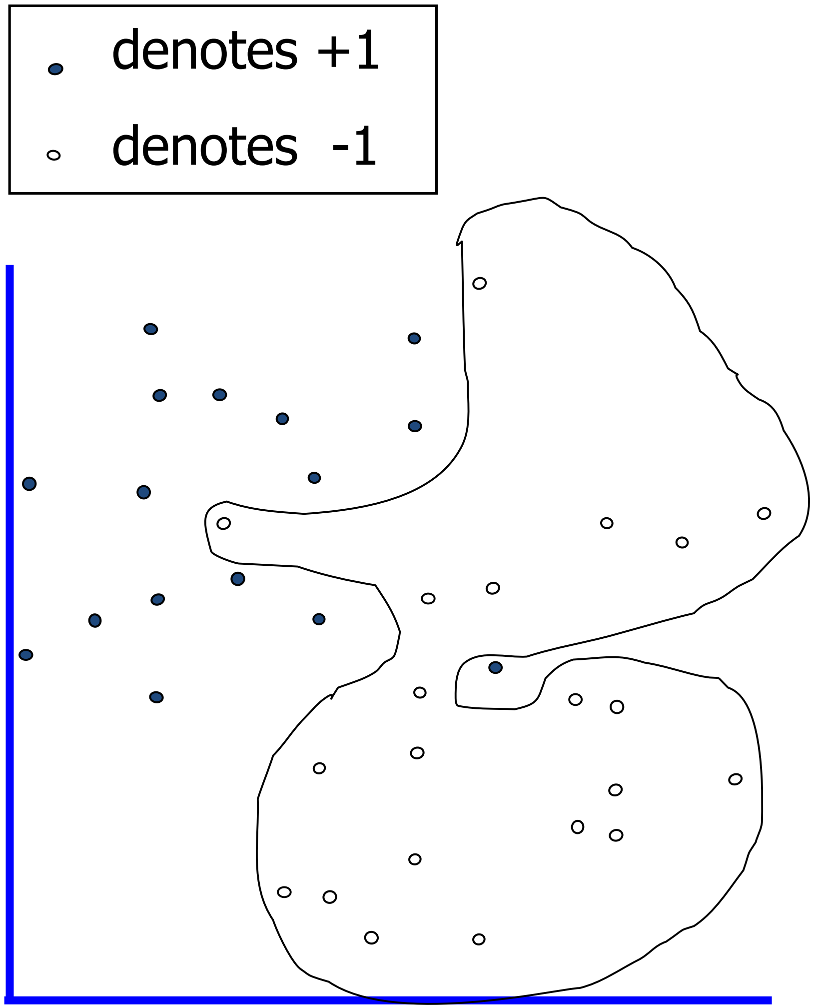
Thus y-intercept is -(-2)/1 = 2

The decision boundary is perpendicular to w and it has slope -0/1 = 0

# Decision boundary



# Dataset with noise




- **Hard Margin:** **So far we require all data points be classified correctly**

  **- No training error**

- **What if the training set is noisy?**

# Soft Margin Classification

***Slack variables*** $\xi_i$ **can be added to allow misclassification of difficult or noisy examples.**

What should our quadratic optimization criterion be?

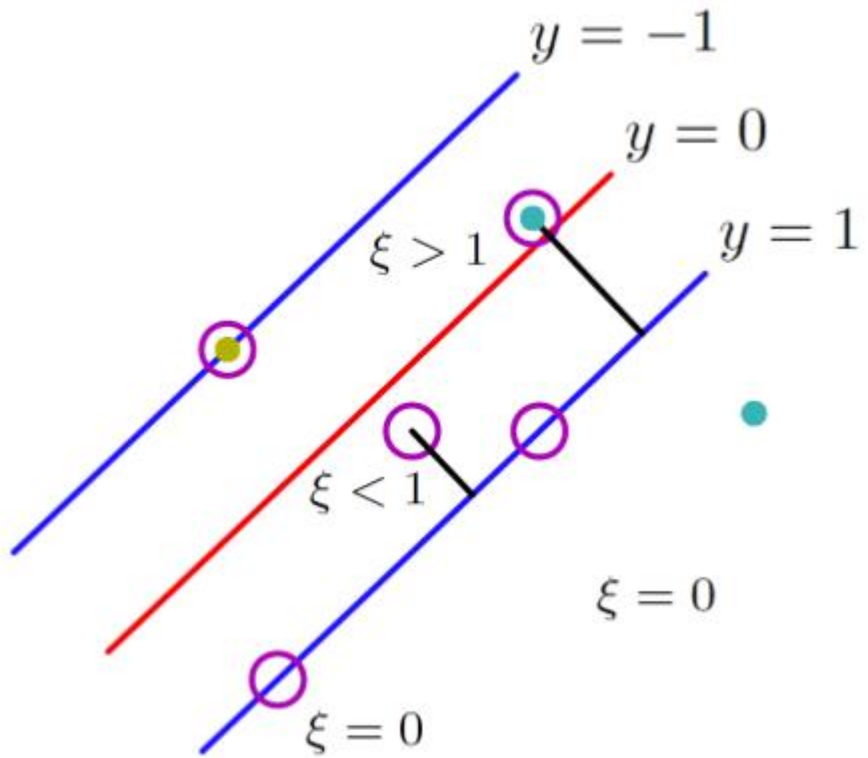
Minimize

$$\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{R}\varepsilon_k$$

# Slack Variable

- **Slack variable** as giving the classifier some leniency when it comes to moving around points near the **margin**.
- When C is large, larger slacks penalize the objective function of SVM's more than when C is small.

# Soft margin example



# Soft Margin

# data samples

Misclassification cost

Slack variable

$$\min_{\boldsymbol{w}} \underbrace{\frac{1}{2}\|\boldsymbol{w}\|^2}_{\text{Maximize margin}} + \underbrace{C\sum_{i=1}^{N}\xi_i}_{\text{Minimize misclassification}}$$

The *w* that minimizes…

$$\text{subject to} \quad y_i \boldsymbol{w}^T \boldsymbol{x}_i \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \quad \forall i = 1, \ldots, N$$

# Hard Margin versus Soft Margin

- **Hard Margin:**

Find $\mathbf{w}$ and $b$ such that
$$\boldsymbol{\Phi}(\mathbf{w}) = \tfrac{1}{2}\,\mathbf{w}^{\mathrm{T}}\mathbf{w} \text{ is minimized and for all } \{(\mathbf{x_i}, y_i)\}$$
$$y_i\,(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + \mathrm{b}) \geq 1$$

- **Soft Margin incorporating slack variables:**

Find $\mathbf{w}$ and $b$ such that
$$\boldsymbol{\Phi}(\mathbf{w}) = \tfrac{1}{2}\,\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum \xi_i \quad \text{is minimized and for all } \{(\mathbf{x_i}, y_i)\}$$
$$y_i\,(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0 \text{ for all } i$$

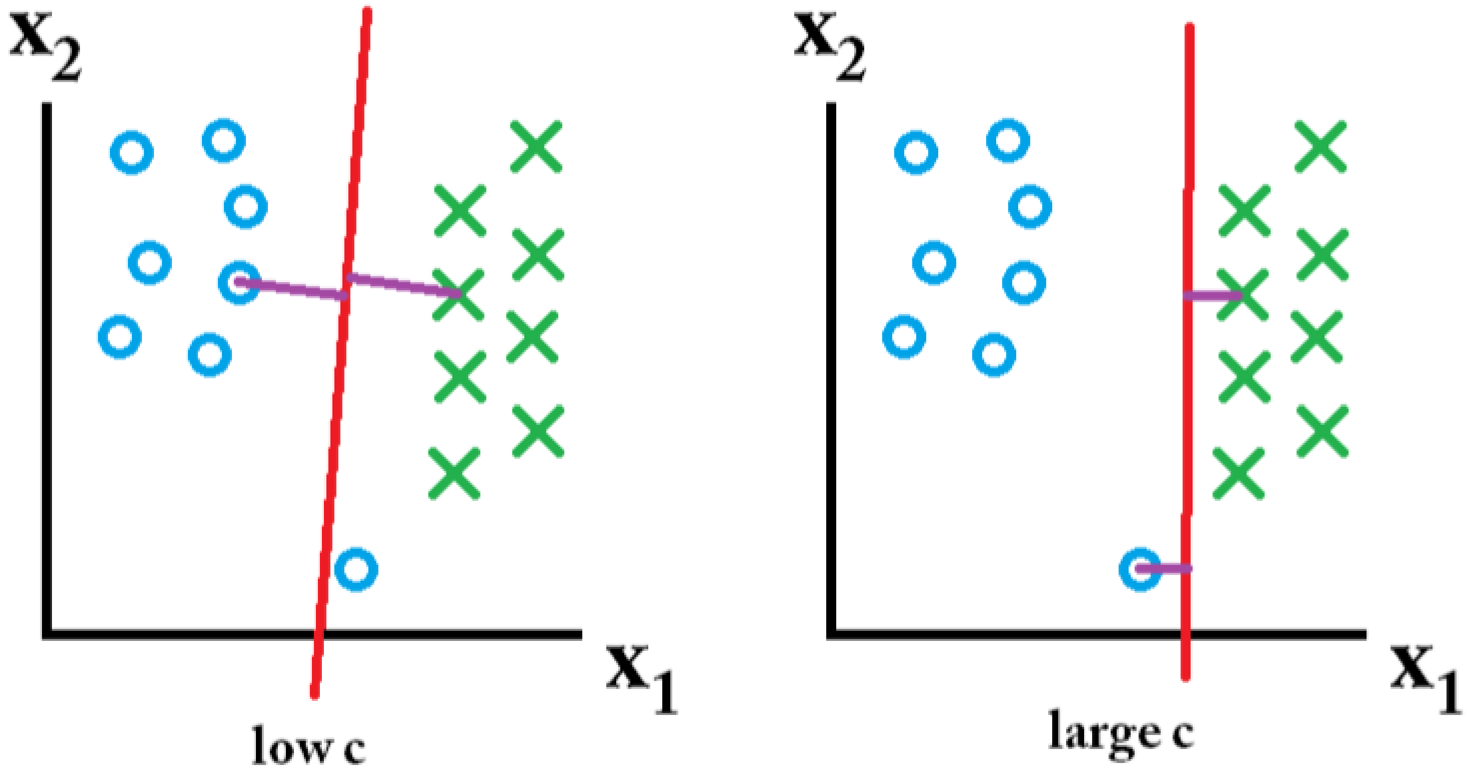
- **Parameter *C* can be viewed as a way to control overfitting.**

# Value of C parameter

- C parameter tells the SVM optimization how much you want to avoid misclassifying each training example.
- For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly.
- Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

# Effect of Margin size v/s misclassification cost

Training set

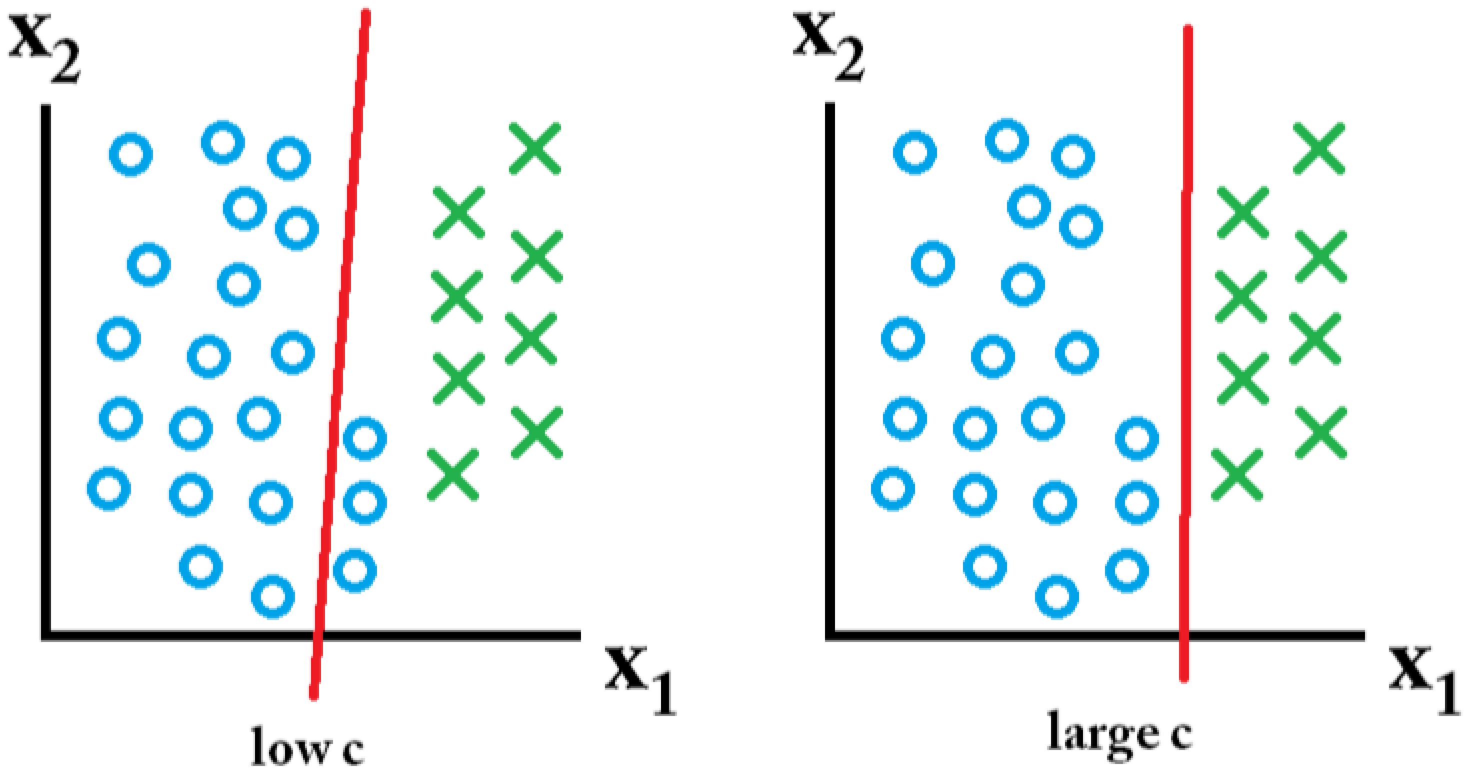



Misclassification ok, want large margin

Misclassification not ok

# Effect of Margin size v/s misclassification cost

Including test set A

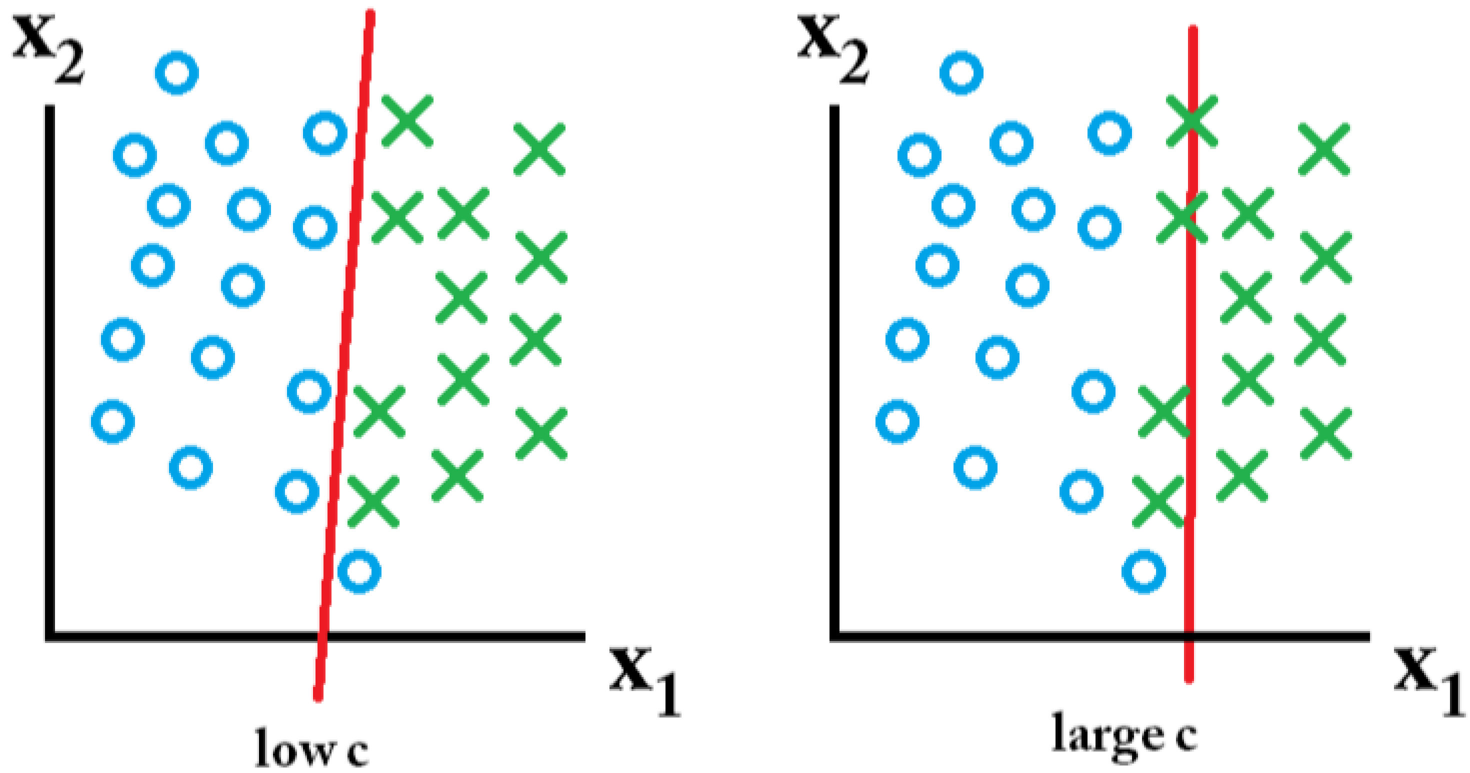



Misclassification ok, want large margin

Misclassification not ok

# Effect of Margin size v/s misclassification cost

Including test set B




Misclassification ok, want large margin

Misclassification not ok

# Linear SVMs: Overview

innovate achieve lead

- **The classifier is a *separating hyperplane.***
- **Most "important" training points are support vectors; they define the hyperplane.**
- **Quadratic optimization algorithms can identify which training points $\mathbf{x}_i$ are support vectors with non-zero Lagrangian multipliers $\alpha_i$.**

$$\boxed{f(\mathbf{x}) = \Sigma \alpha_i y_i \mathbf{x}_i^{T}\mathbf{x} + b}$$

# Good Web References for SVM

- **Text categorization with Support Vector Machines: learning with many relevant features** - T. Joachims, ECML
- **A Tutorial on Support Vector Machines for Pattern Recognition**, Kluwer Academic Publishers - Christopher J.C. Burges
- http://www.cs.utexas.edu/users/mooney/cs391L/
- https://www.coursera.org/learn/machine-learning/home/week/7
- https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
- https://data-flair.training/blogs/svm-kernel-functions/
- MIT 6.034 Artificial Intelligence, Fall 2010
- https://stats.stackexchange.com/questions/30042/neural-networks-vs-support-vector-machines-are-the-second-definitely-superior
- https://www.sciencedirect.com/science/article/abs/pii/S0893608006002796
- https://medium.com/deep-math-machine-learning-ai/chapter-3-support-vector-machine-with-math-47d6193c82be
- Radial basis kernel