BITS Pilani
Pilani Campus

Classification: Support Vector Machines

Dr. Chetana Gavankar, Ph.D,
IIT Bombay-Monash University Australia
Chetana.gavankar@pilani.bits-pilani.ac.in

innovate achieve lead

BITS Pilani
Pilani Campus

Session 7
Date – 20/02/2022
Time – 10 to 12.30

Text Book(s)

T1	Christopher Bishop: Pattern Recognition and Machine Learning, Springer International Edition
T2	Tom M. Mitchell: Machine Learning, The McGraw-Hill Companies, Inc..

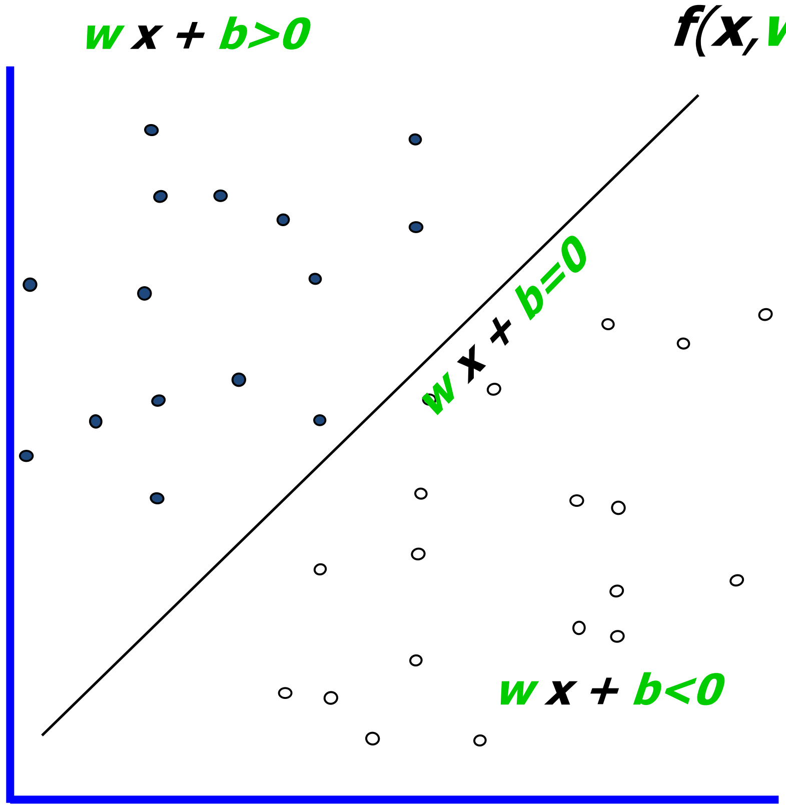
These slides are prepared by the instructor, with grateful acknowledgement of Prof. Tom Mitchell, Prof. Andrew Moore and many others who made their course materials freely available online.

Topics to be covered

- Understanding the spirit and significance of maximum margin classifier
- Posing an optimization problem for SVM in non-overlapping class scenario
- Converting the constrained optimization problem into unconstrained using Lagrange multipliers
- Dual of the optimization problem
- Appreciation of sparse kernel machine and support vectors in the solution of the optimization problem
- Implementation of SVM in python

Linear Classifiers

- denotes +1
- denotes -1



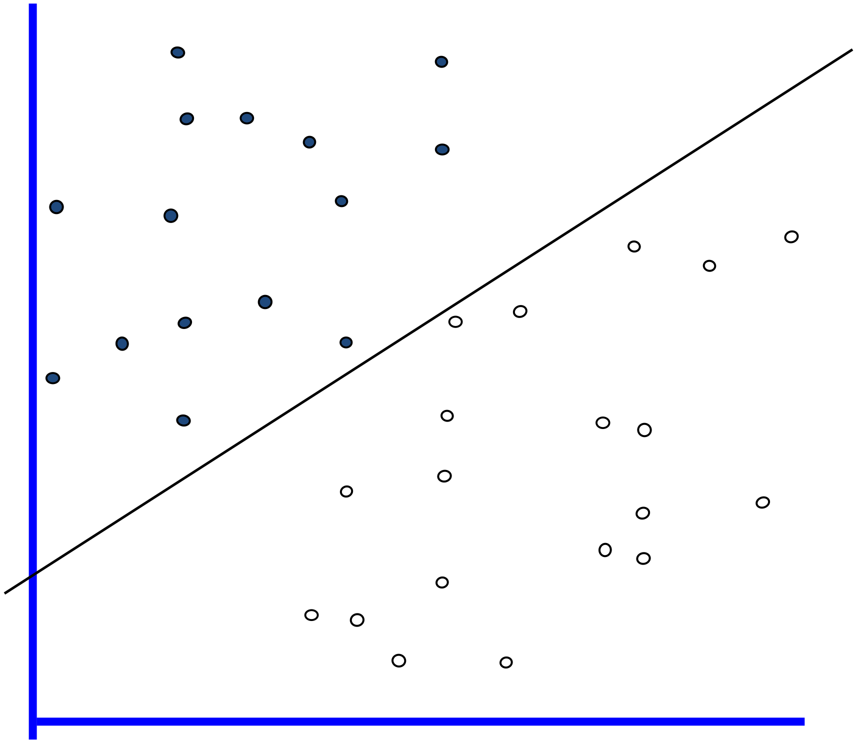
$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}\ \mathbf{x} + b)$

How would you classify this data?

Linear Classifiers

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

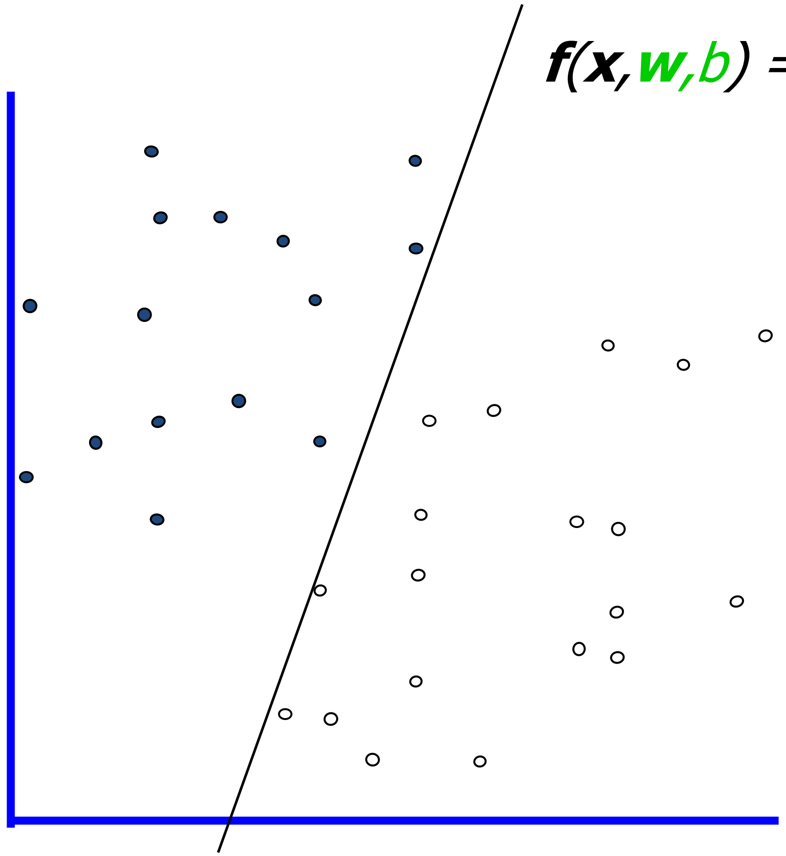
- denotes +1
- denotes -1



How would you classify this data?

Linear Classifiers

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



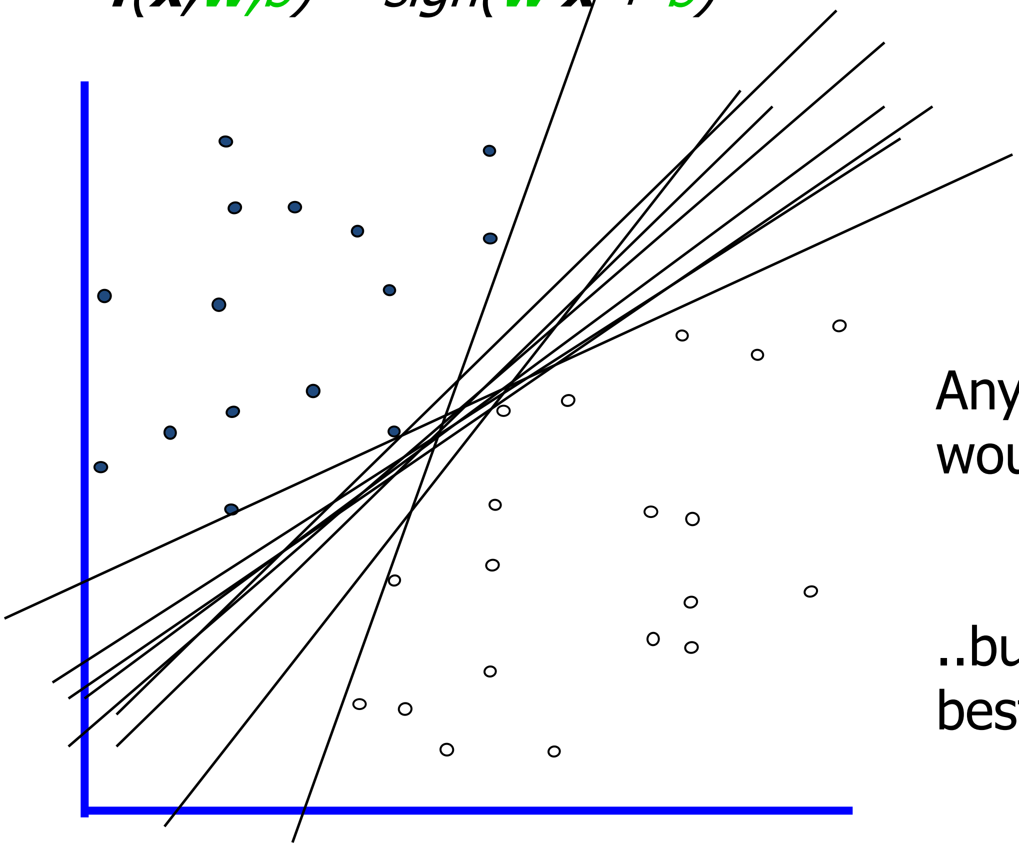
- denotes +1
- denotes -1

How would you classify this data?

Linear Classifiers

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

- denotes +1
- denotes -1



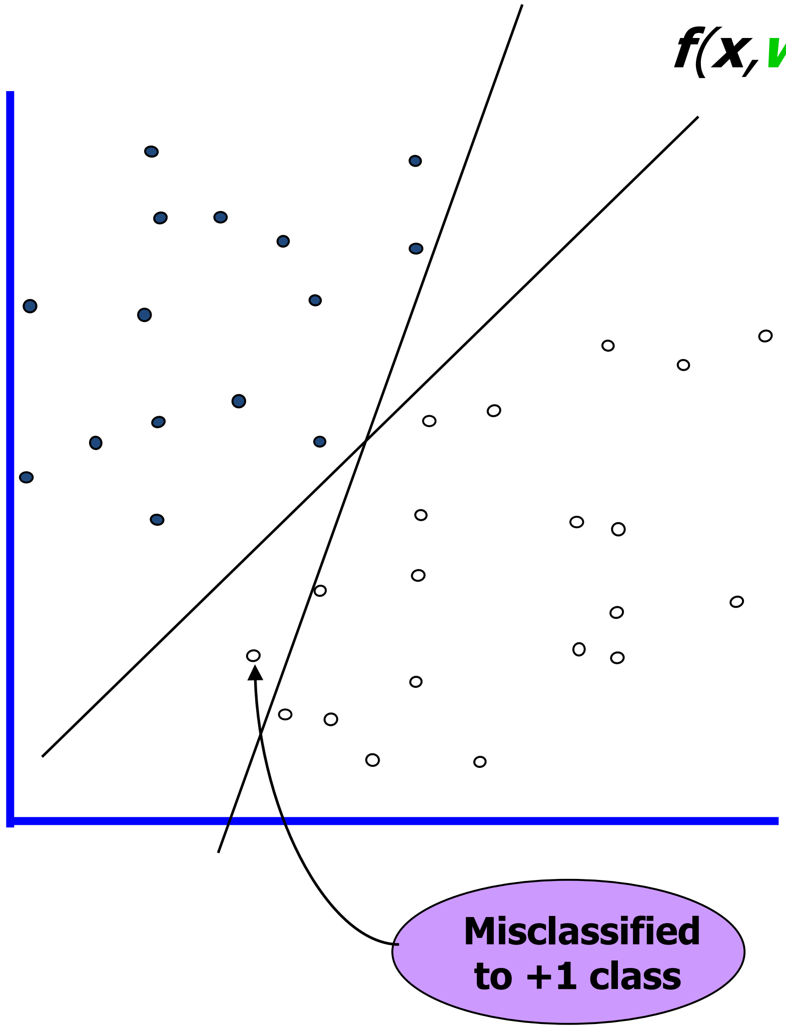
Any of these would be fine..

..but which is best?

Linear Classifiers

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

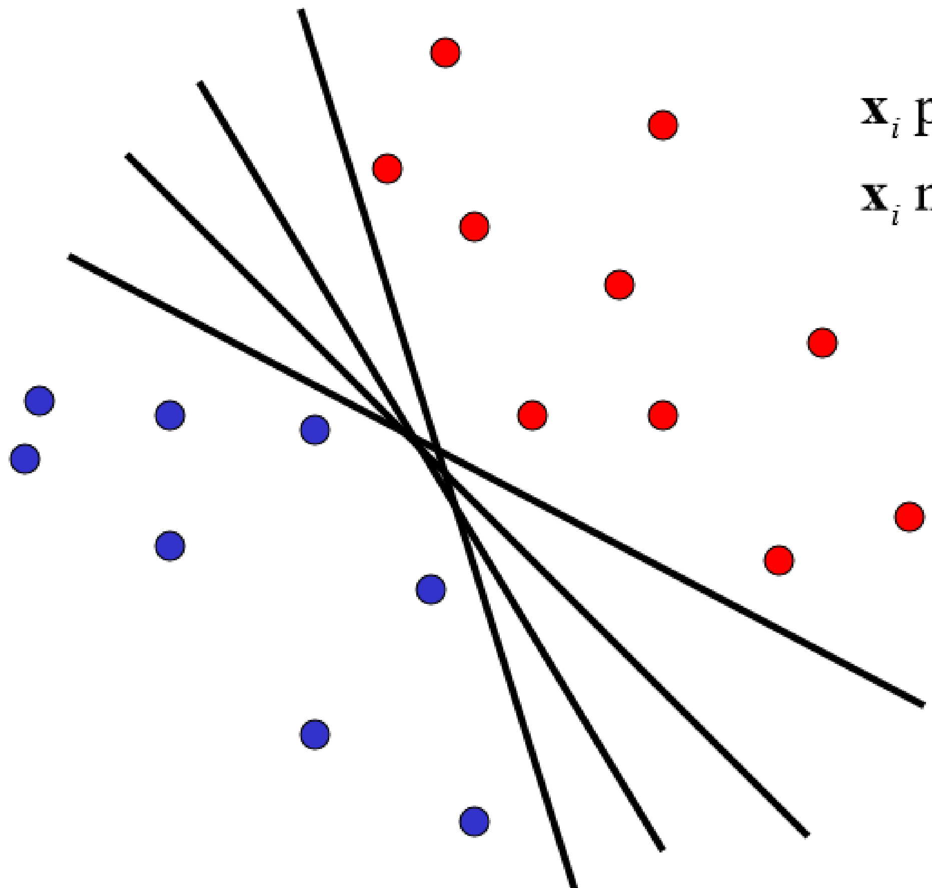
- denotes +1
- denotes -1



How would you classify this data?

Linear Classifier

- Find linear function to separate positive and negative examples



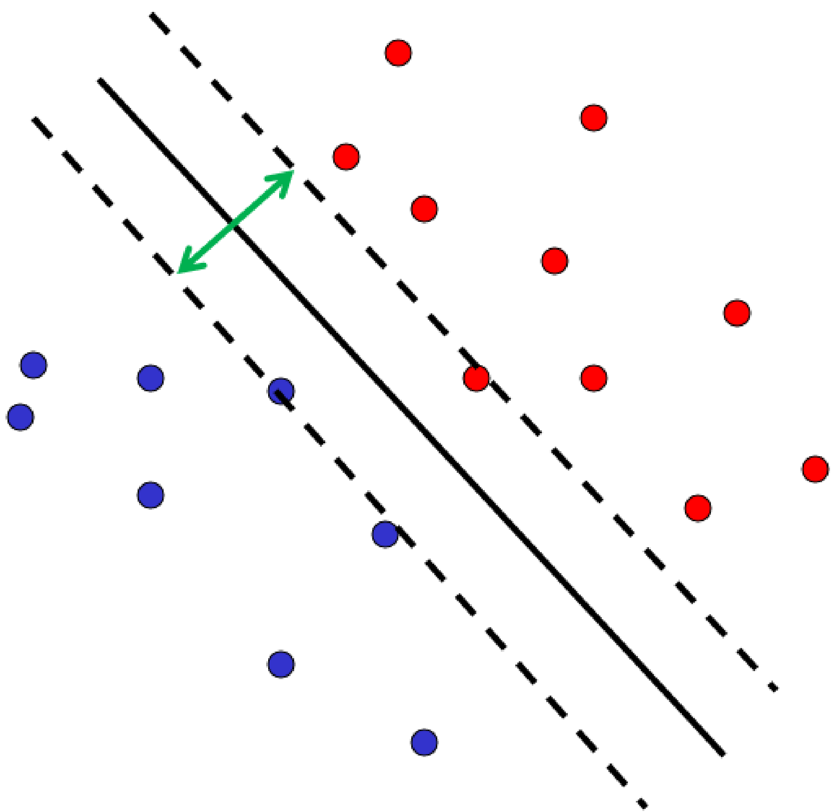
$\mathbf{x}_i$ positive : $\quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$

$\mathbf{x}_i$ 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

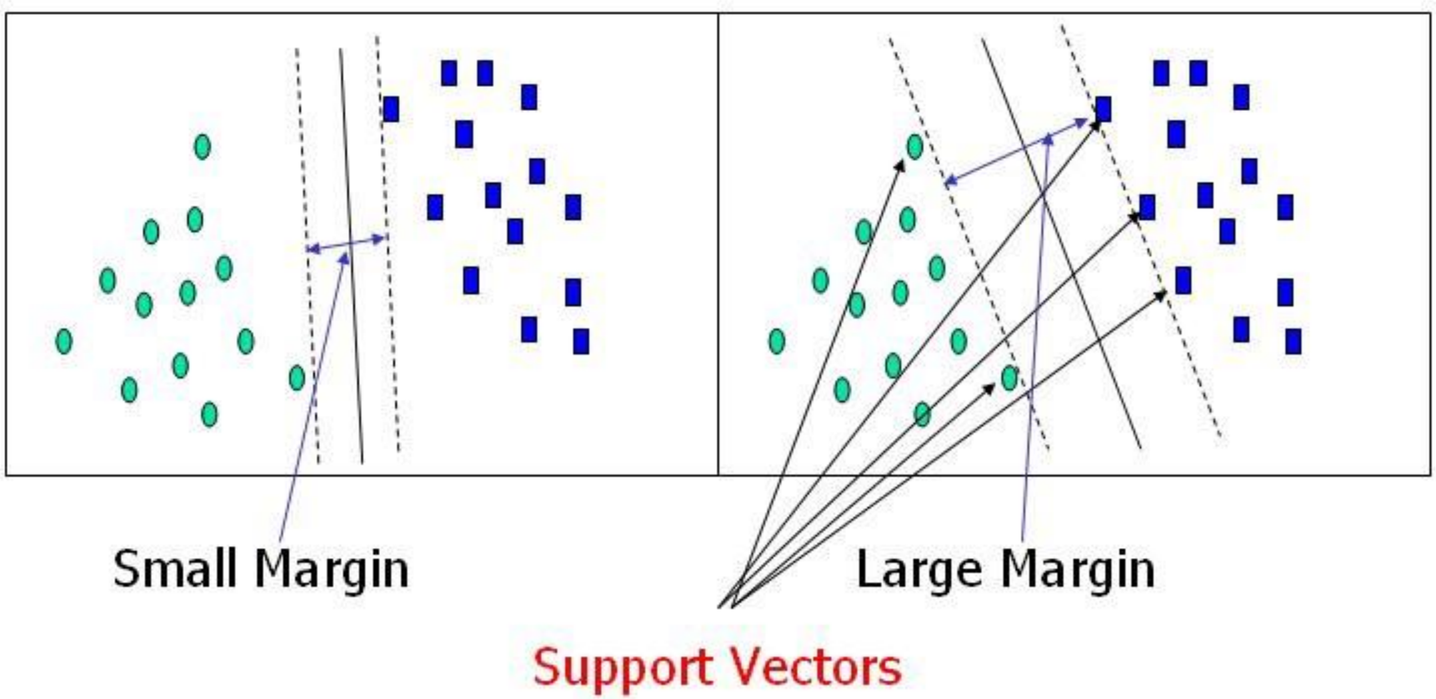
Linear Classifier



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

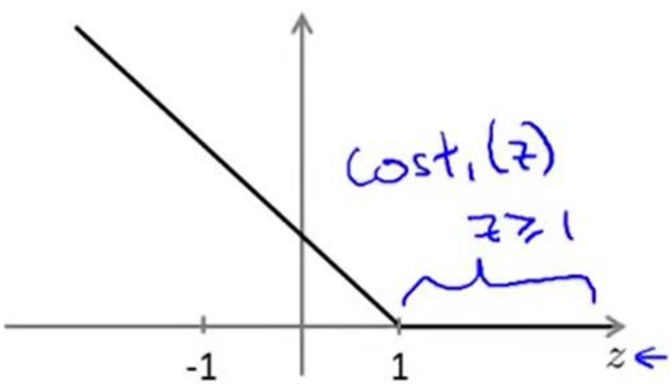
C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

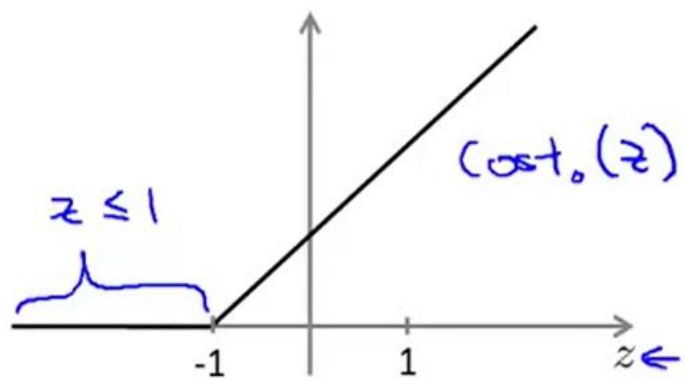
Large margin and support vectors



Support Vector Machine

$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$



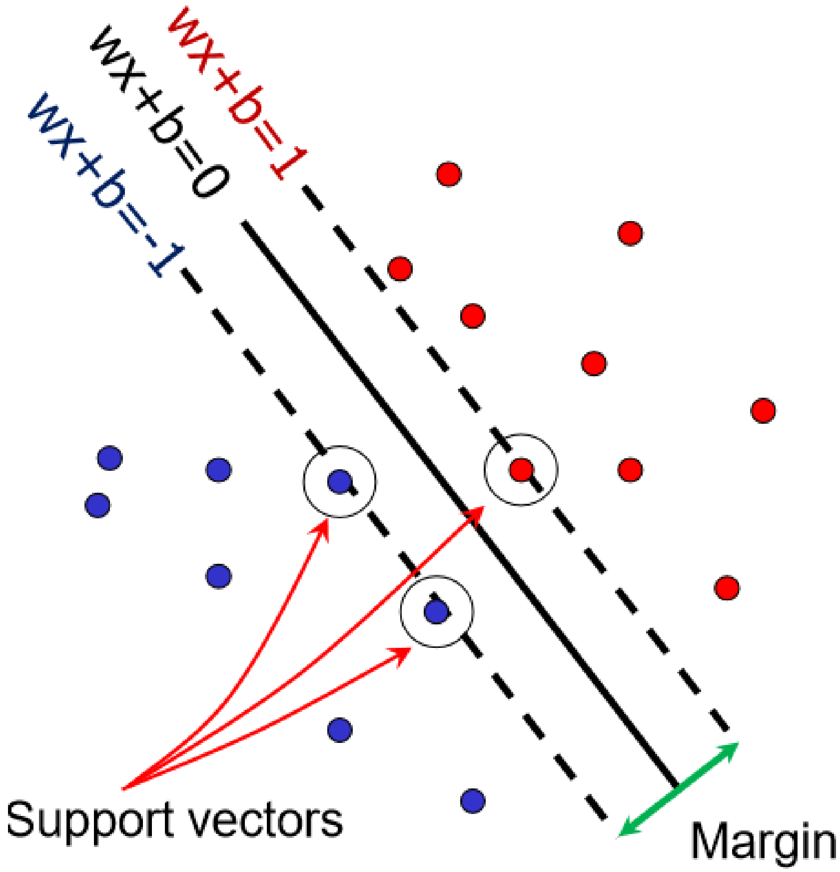


- If $y = 1$, we want $\theta^T x \geq 1$ (not just ≥ 0) $\theta^T x \geq \cancel{0}\ 1$
- If $y = 0$, we want $\theta^T x \leq -1$ (not just < 0) $\theta^T x \leq \cancel{0}\ -1$

Andrew Ng

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

Define the hyperplanes H such that:

$w \bullet x_i + b \geq +1$ when $y_i = +1$

$w \bullet x_i + b \leq -1$ when $y_i = -1$

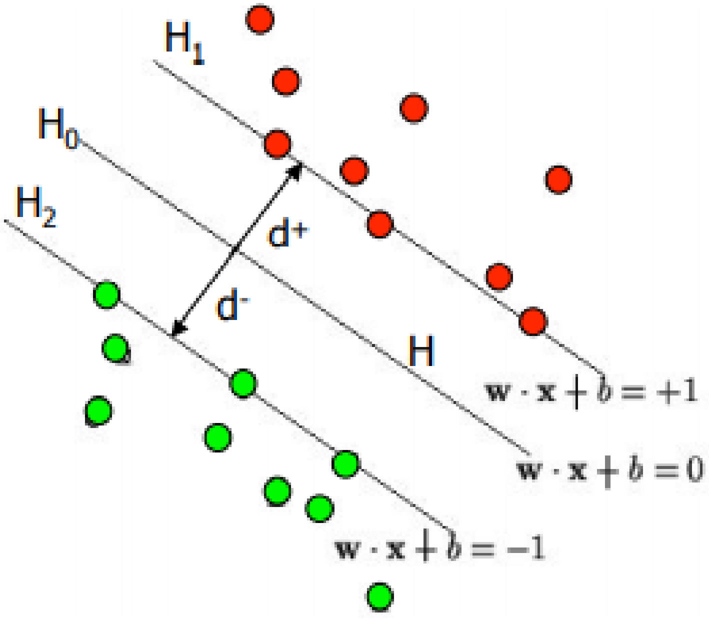
H_1 and H_2 are the planes:

H_1: $w \bullet x_i + b = +1$

H_2: $w \bullet x_i + b = -1$

The points on the planes H_1 and H_2 are the tips of the Support Vectors

The plane H_0 is the median in between, where $w \bullet x_i + b = 0$



d+ = the shortest distance to the closest positive point

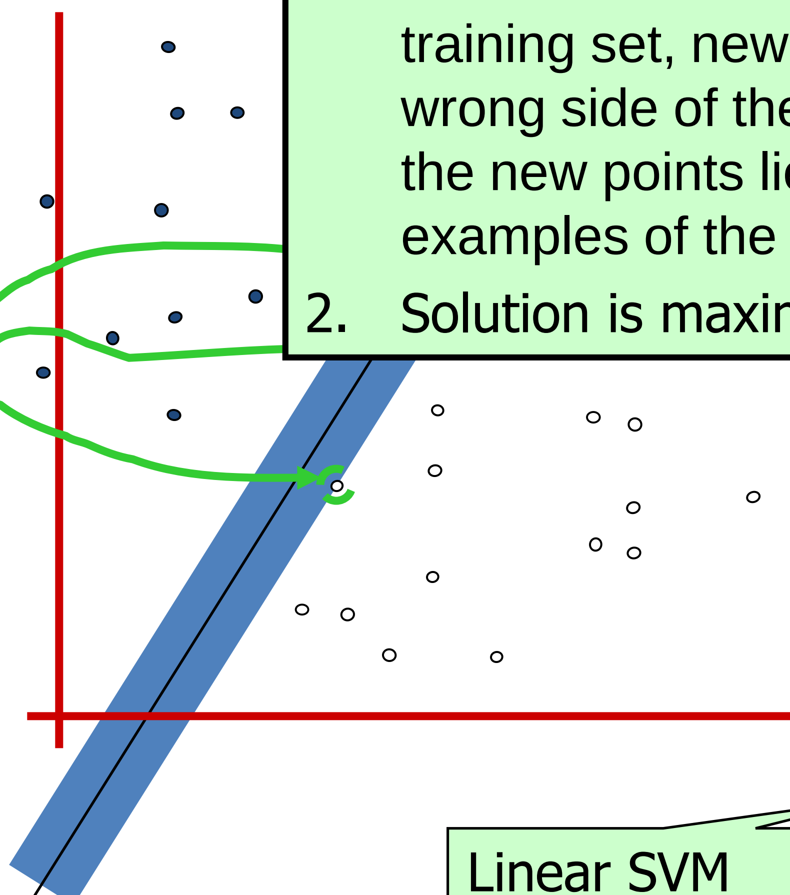
d- = the shortest distance to the closest negative point

The margin (gutter) of a separating hyperplane is d+ + d–.

Maximum Margin

- denotes +1
- denotes -1

Support Vectors are those datapoints that the margin pushes up against



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.

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

Linear SVM

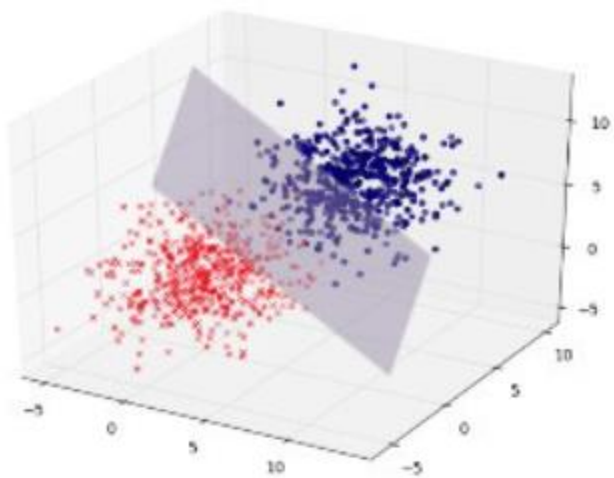
Support Vectors

- 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

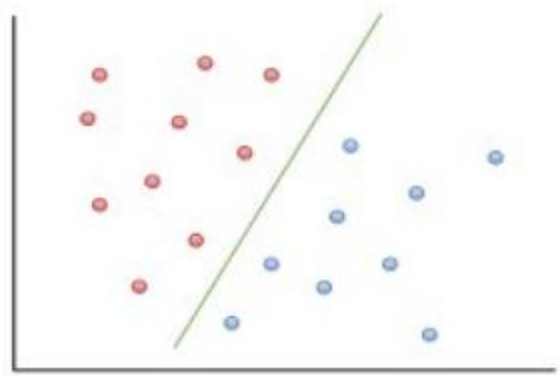
$$\mathbf{w}^T\mathbf{x} = 0$$

Hyperplane



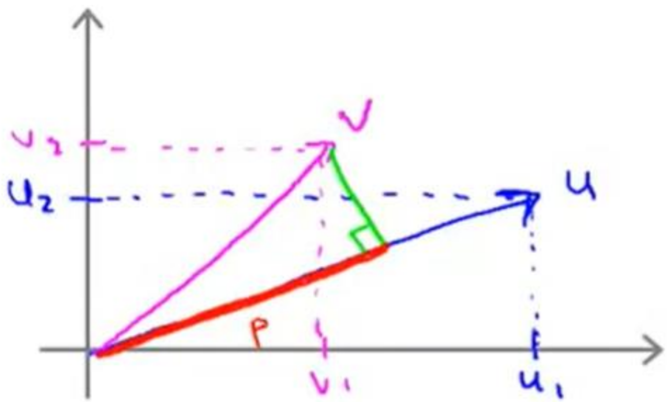
$$y = ax + b$$

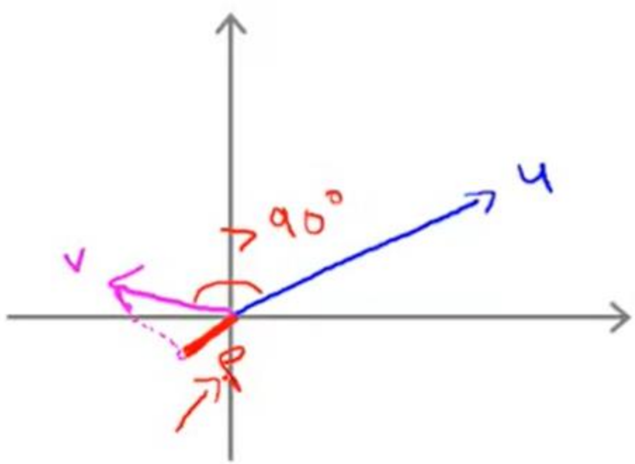
Line



Norm of vector

Vector Inner Product





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

$u^T v = ?$ $\quad [u_1 \;\; u_2] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$

$\|u\|$ = length of vector u

$= \sqrt{u_1^2 + u_2^2} \in \mathbb{R}$

p = length of projection of v onto u.

Signed

$u^T v = \underline{p} \cdot \underline{\|u\|} \leftarrow \quad = v^T u$

$= u_1 v_1 + u_2 v_2 \leftarrow \quad p \in \mathbb{R}$

$u^T v = p \cdot \|u\|$

$p < 0$

Andrew Ng

SVM Decision Boundary intuition

SVM Decision Boundary

$\omega = \left(\sqrt{\omega}\right)^2$

$$\min_{\theta} \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 = \frac{1}{2}\left(\theta_1^2+\theta_2^2\right) = \frac{1}{2}\left(\sqrt{\theta_1^2+\theta_2^2}\right)^2 = \frac{1}{2}\|\theta\|^2$$

$\sqrt{\theta_1^2+\theta_2^2} = \|\theta\|$

$$\text{s.t.}\quad \boxed{\theta^T x^{(i)} \geq 1} \quad \text{if } y^{(i)} = 1$$

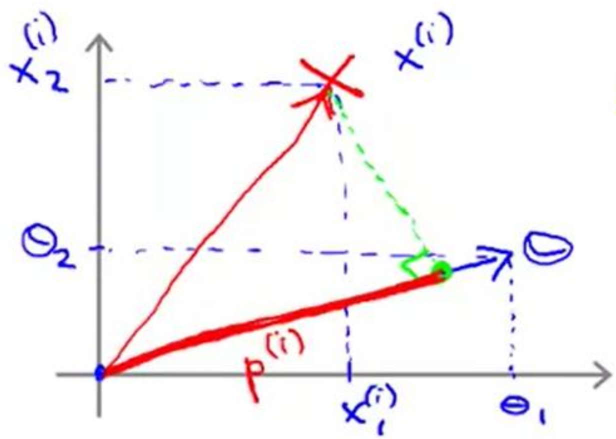
$$\theta^T x^{(i)} \leq -1 \quad \text{if } y^{(i)} = 0$$

Simplification: $\theta_0 = 0$. $n = 2$

$\begin{bmatrix} \cancel{\theta_0} \\ \theta_1 \\ \theta_2 \end{bmatrix}$ $\theta_0 = 0$

$\theta^T x^{(i)} = ?$

$u^T v$



$$\theta^T x^{(i)} = \boxed{p^{(i)} \cdot \|\theta\|}$$

$$= \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}$$

Andrew Ng

SVM Decision Boundary intuition

SVM Decision Boundary

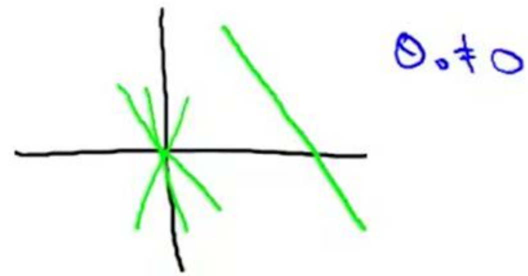
$$\min_{\theta} \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 = \frac{1}{2}\|\theta\|^2$$

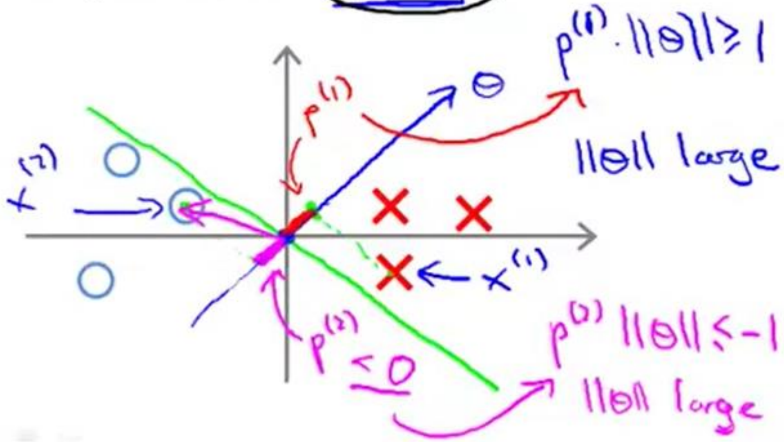
s.t. $\boxed{p^{(i)} \cdot \|\theta\| \geq 1}$ if $y^{(i)} = 1$

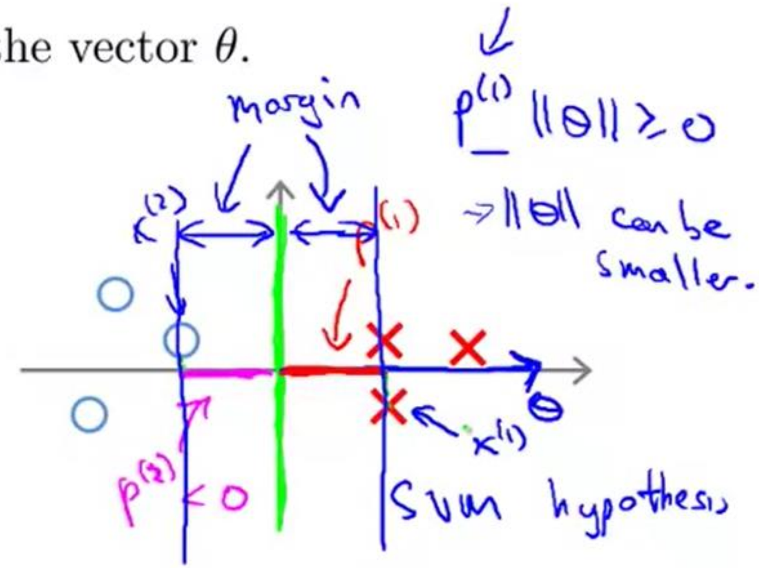
$p^{(i)} \cdot \|\theta\| \leq -1$ if $y^{(i)} = 1$

where $p^{(i)}$ is the projection of $x^{(i)}$ onto the vector θ.

Simplification: $\theta_0 = 0$

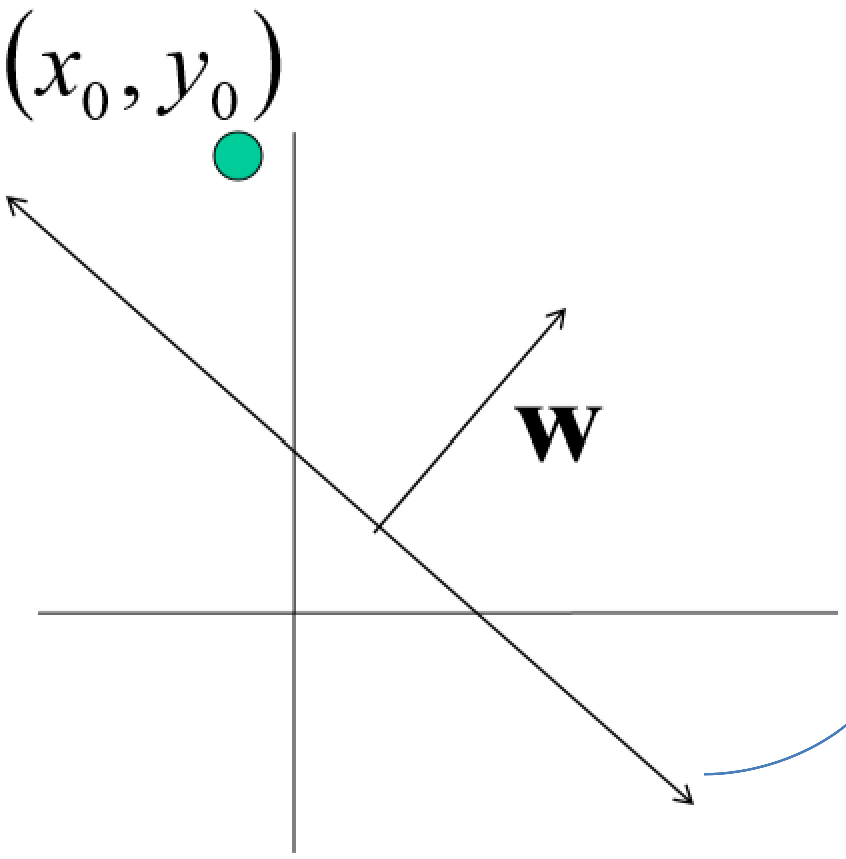






Andrew Ng

Line with 2 features: R2



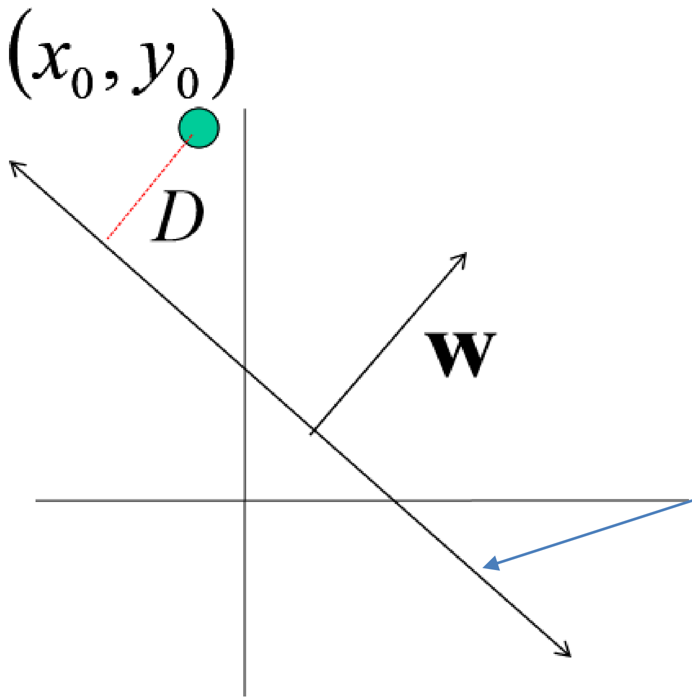
Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

$$\Updownarrow$$

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

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

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:

$w^T x_a + b = 0$

$w^T x_b + b = 0$

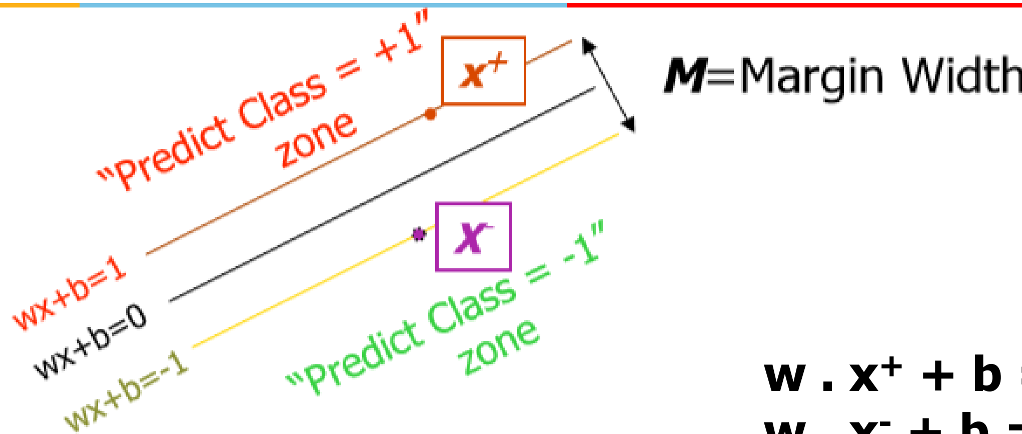
Subtracting these two equations gives us

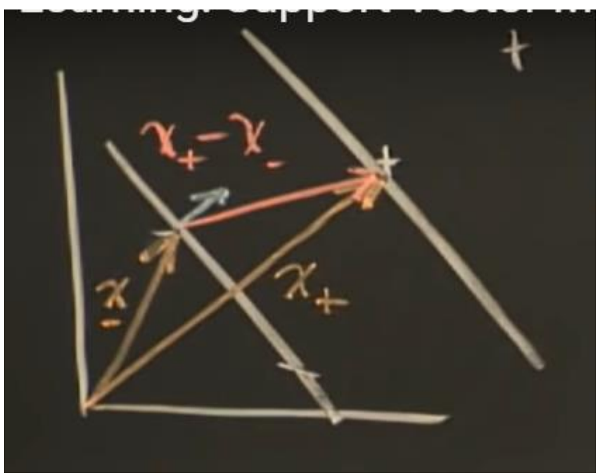
$w^T.(x_a - x_b) = 0$

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.

Linear SVM Mathematically





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

Margin width

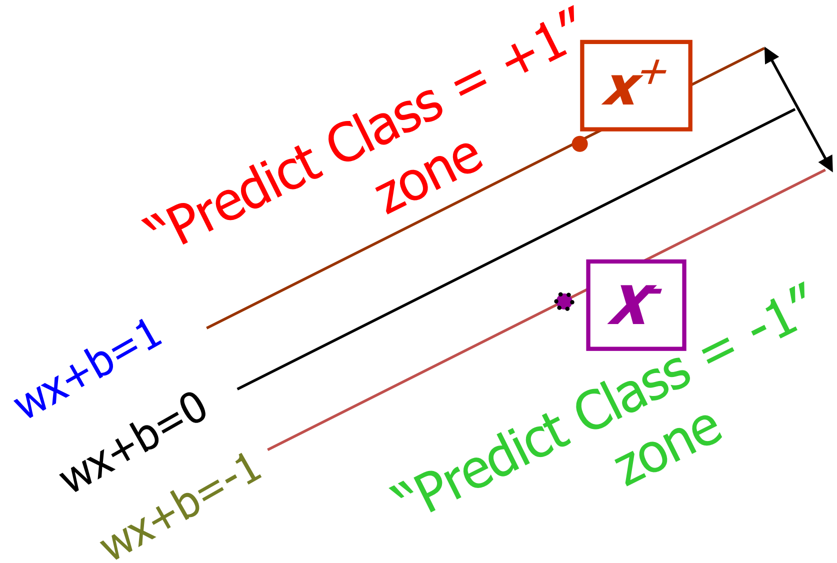
$$= x^+ - x^- \cdot \frac{w}{||w||}$$

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

$$= (1\text{-b}) - (-1\text{-b}) \, / \, ||w||$$

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

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

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

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

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

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

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

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

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

Solving the Optimization Problem

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

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

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

- **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.**
- **Because it is quadratic, the surface is a paraboloid, with just a single global minimum (thus avoiding a problem we had with neural nets!)**

Solving the Optimization Problem

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

← 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* α_i is associated with every constraint in the primary problem:**

Solving the Optimization Problem

- **The solution involves constructing a *unconstrained problem* where a *Lagrange multiplier* α_i is associated with every constraint in the primary problem:**

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

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

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

- Expanding above equation:

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

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

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

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

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

Optimization Problem

Find **w** and b such that

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

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

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

Find $\alpha_1 \ldots \alpha_N$ such that

$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\left(\sum_i \sum_j \alpha_i\, \alpha_j\, y_i\, y_{\mathbf{j}}\, \mathbf{x_i} \cdot \mathbf{x_j} \right)$ is maximized and

(1) $\Sigma\alpha_i y_i = 0$

(2) $\alpha_i \geq 0$ for all α_i

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

  min 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 – Equality and Inequality constraint

- Make the Lagrangian function for minimization:

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

- Necessary conditions to have a minimum are

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

Support Vectors

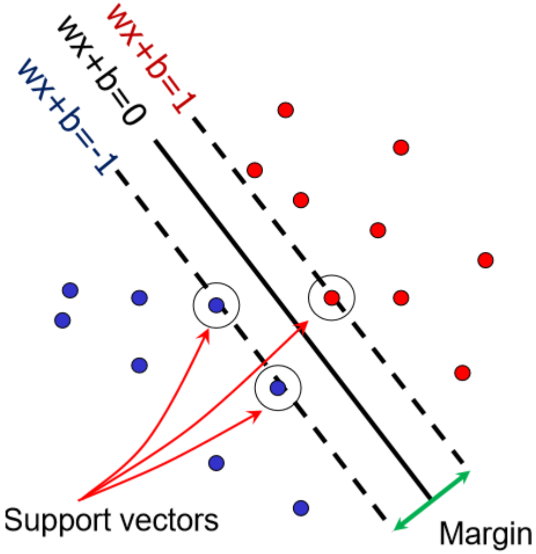
Using KKT conditions :
$\alpha_i [y_i (\mathbf{w}^T\mathbf{x_i} + b) -1]=0$

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

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

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

- Want line that maximizes the margin.



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

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

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

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

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)

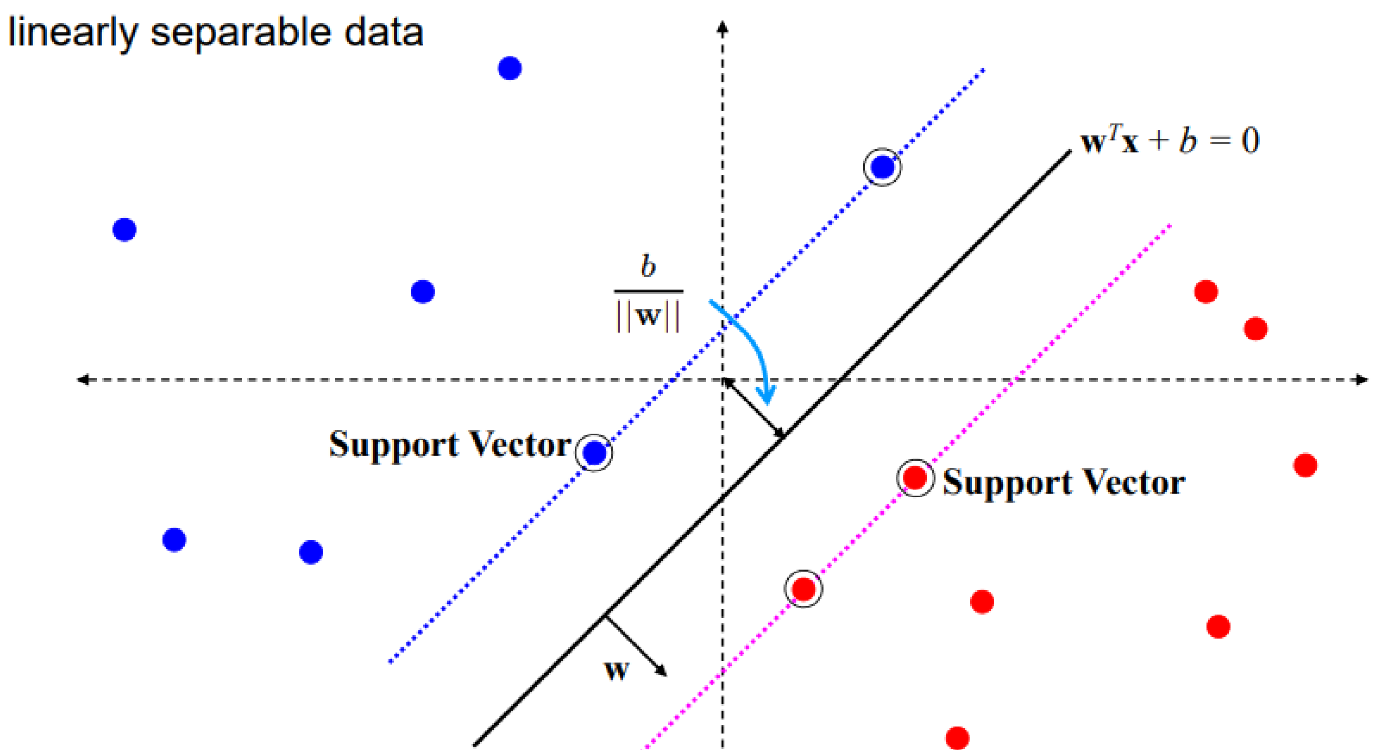
- Classification function:

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

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)

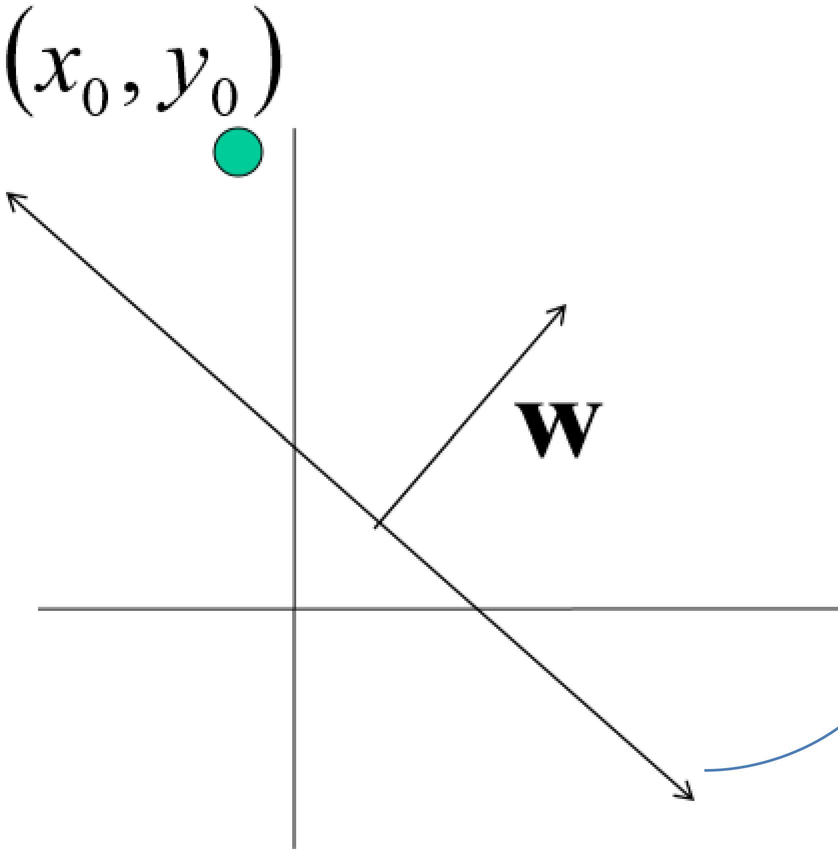
Substituting w in support vectors function



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

support vectors

Line with 2 features: R2



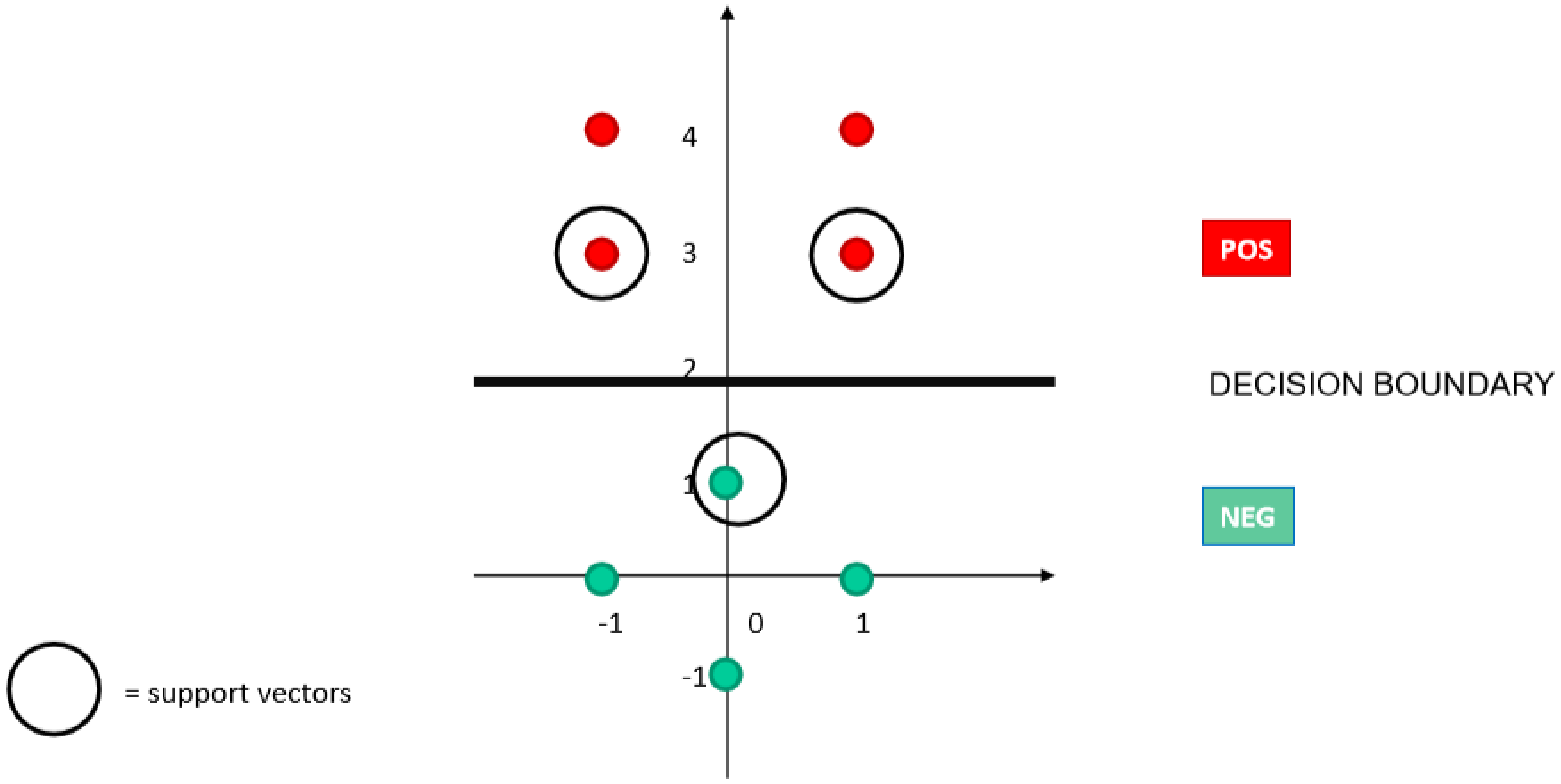
Let $\mathbf{w} = \begin{bmatrix} a \\ c \end{bmatrix}$ $\quad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

$$ax + cy + b = 0$$

$$\Updownarrow$$

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

Decision boundary



SVM optimization example

- *Let 2 points for classification be x_1=(2,1)* and x_2=(1,2).

With class labels y_1=-1 and y_2=1

$$
\begin{aligned}
\max_{\alpha} L_D &= \sum_{i=1}^{L} \alpha_i - \tfrac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\
&= \alpha_1 + \alpha_2 - \tfrac{1}{2} \Big(\alpha_1 \alpha_1 \cdot 1 \cdot 1 \cdot \langle \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix} \rangle + \\
&\quad +2 \cdot \alpha_1 \alpha_2 \cdot 1 \cdot (-1) \cdot \langle \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix} \rangle + \\
&\quad +\alpha_2 \alpha_2 \cdot (-1) \cdot (-1) \cdot \langle \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix} \rangle \Big) \\
&= \alpha_1 + \alpha_2 - \tfrac{1}{2} \left(5\alpha_1^2 - 8\alpha_1 \alpha_2 + 5\alpha_2^2 \right) \\
\text{subject to} \quad & \alpha_1 y_1 + \alpha_2 y_2 = 0 \\
& \alpha_1 \geq 0, \ \alpha_2 \geq 0
\end{aligned}
$$

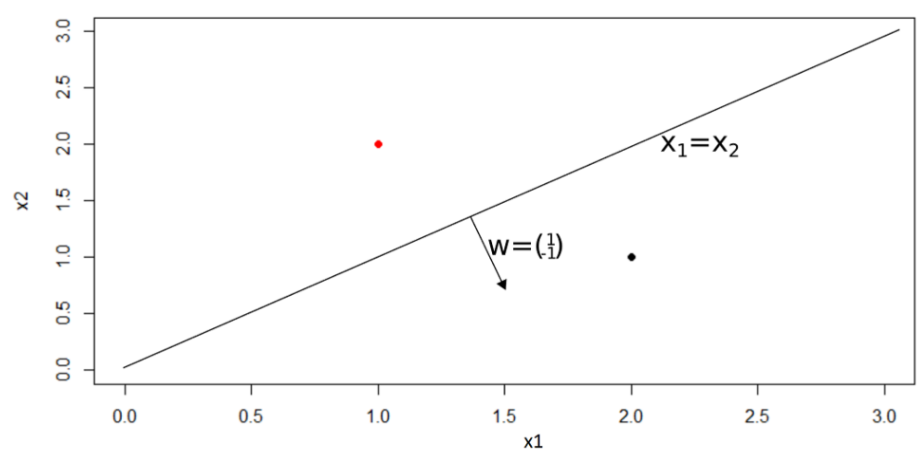
- *Solve* Quadratic Programming Problem using wolfram or any library, we get α1=1 and α2=1

SVM optimization example

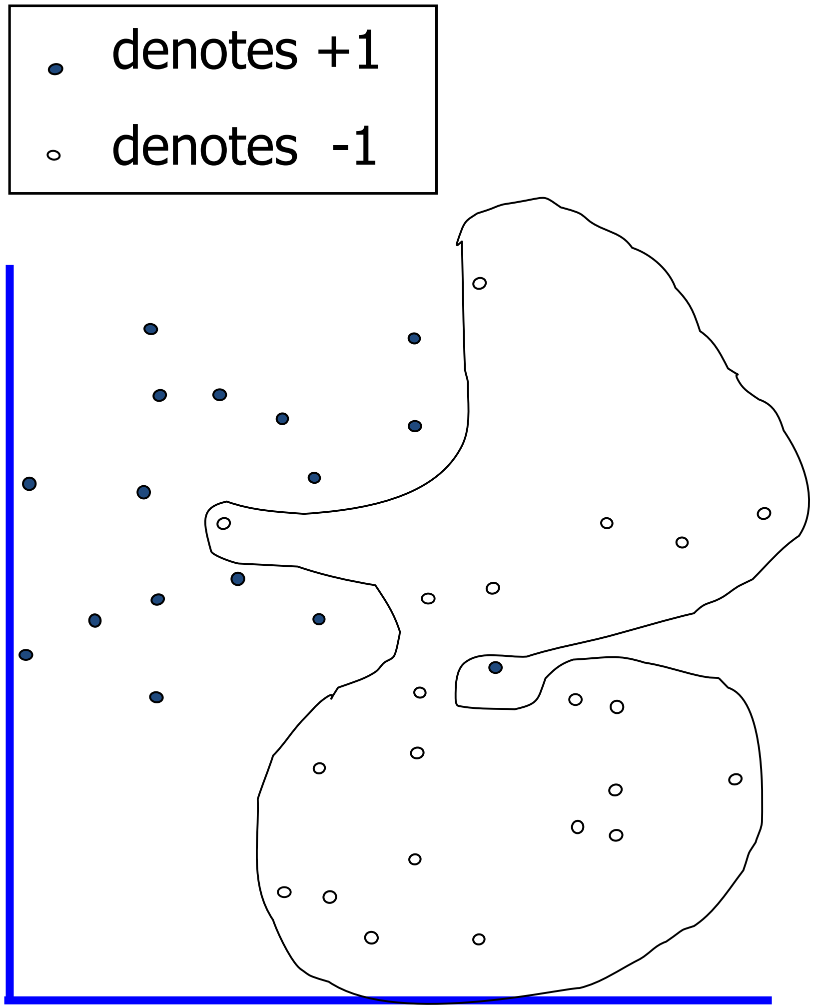
- Solve to get w:

$$w = \sum_{i=1}^{L} \alpha_i y_i x_i = 1 \cdot 1 \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix} + 1 \cdot (-1) \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$
$$= \begin{pmatrix} 2 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$
$$= \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Hyperplane can be represented by
wx+b=0
=> x1-x2=0
=> x1=x2



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 ξ_i **can be added to allow misclassification of difficult or noisy examples.**

What should our quadratic optimization criterion be?

Minimize

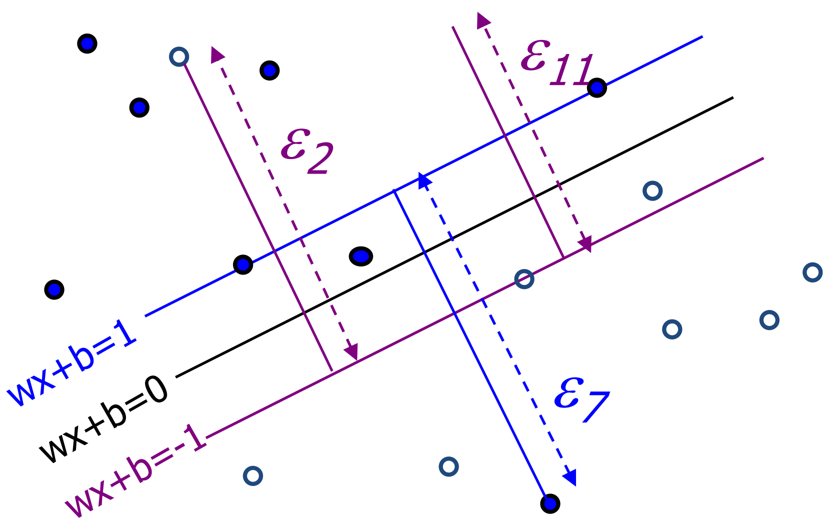
$$\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{R}\varepsilon_k$$

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 Classification

Slack variables ξ_i **can be added to allow misclassification of difficult or noisy examples.**

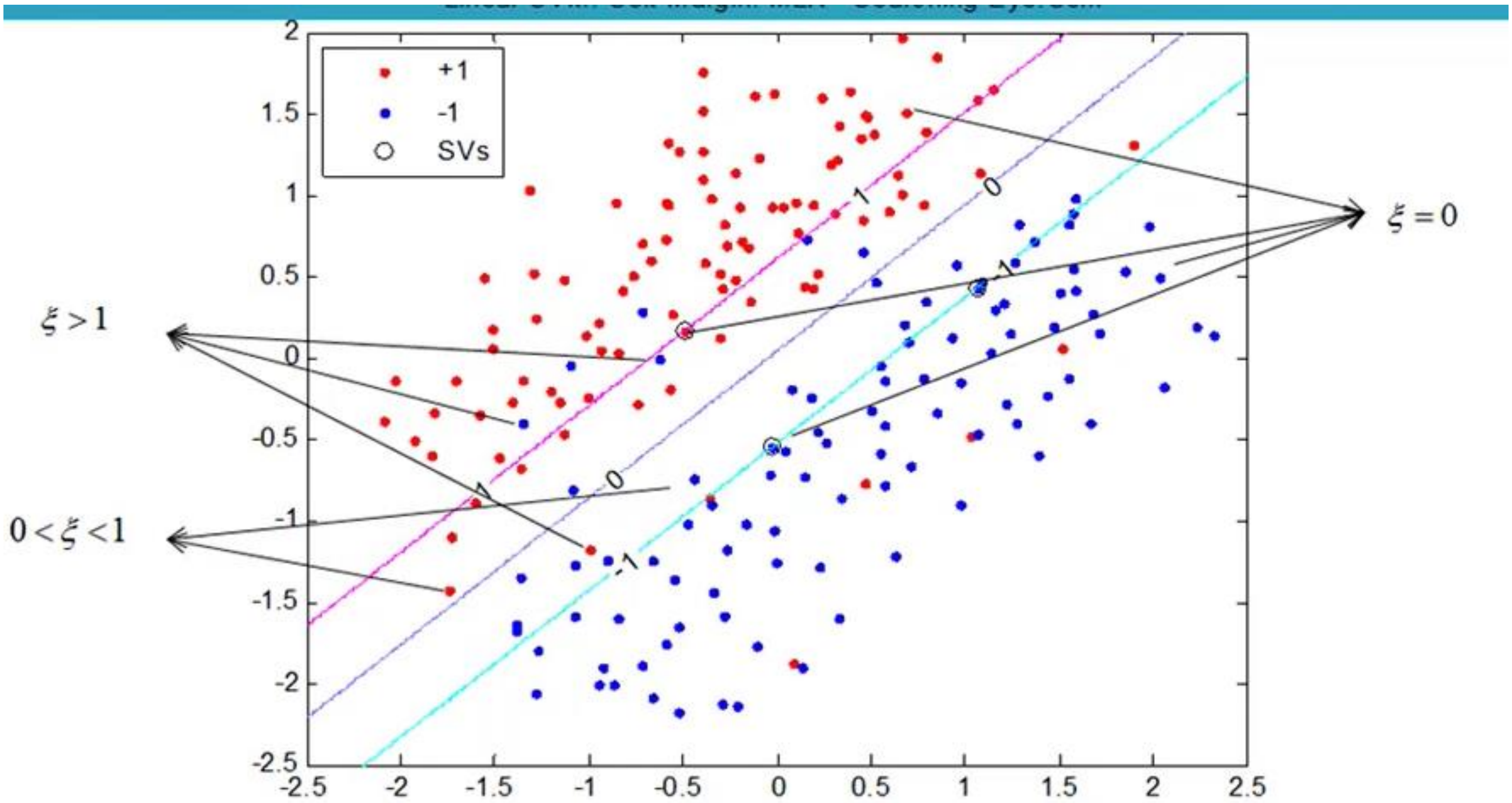


What should our quadratic optimization criterion be?

Minimize

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

Soft Margin Classification



Soft Margin

$$\underbrace{\min_{\boldsymbol{w}}}_{\text{The } w \text{ that minimizes...}} \underbrace{\frac{1}{2}\|\boldsymbol{w}\|^2}_{\text{Maximize margin}} + \underbrace{\overset{\text{Misclassification cost}}{C} \sum_{i=1}^{\overset{\#\text{ data samples}}{N}} \overset{\text{Slack variable}}{\xi_i}}_{\text{Minimize misclassification}}$$

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

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

Hard Margin versus Soft Margin

- **Hard Margin:**

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

$\mathbf{\Phi}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$ 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

$\mathbf{\Phi}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum \xi_i$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$

$y_i(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

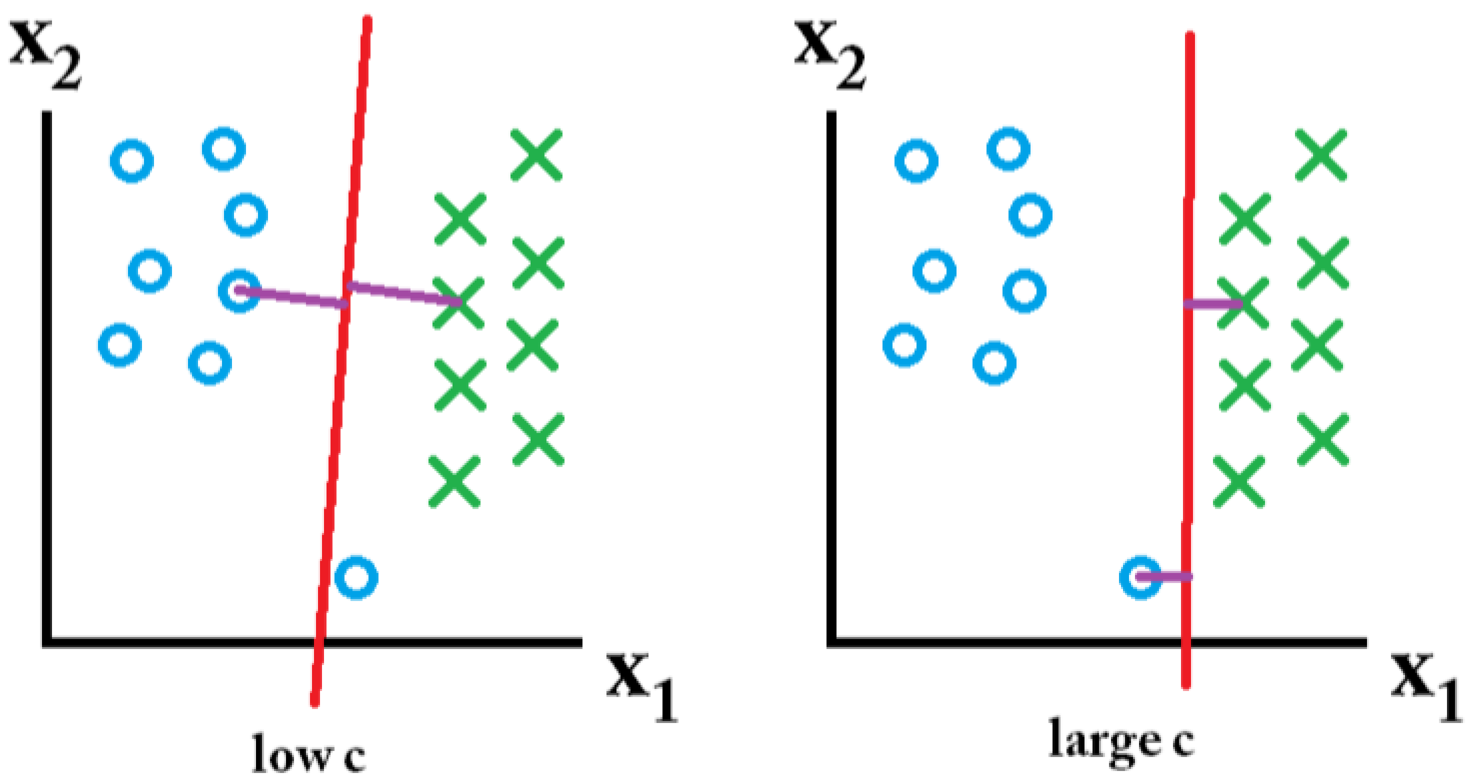
- **Parameter *C* can be viewed as a way to control overfitting.**

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



overfitting

Misclassification ok, want large margin

Misclassification not ok

Linear SVMs: Overview

- 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 α_i.
- Both in the dual formulation of the problem and in the solution training points appear only inside dot products:

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

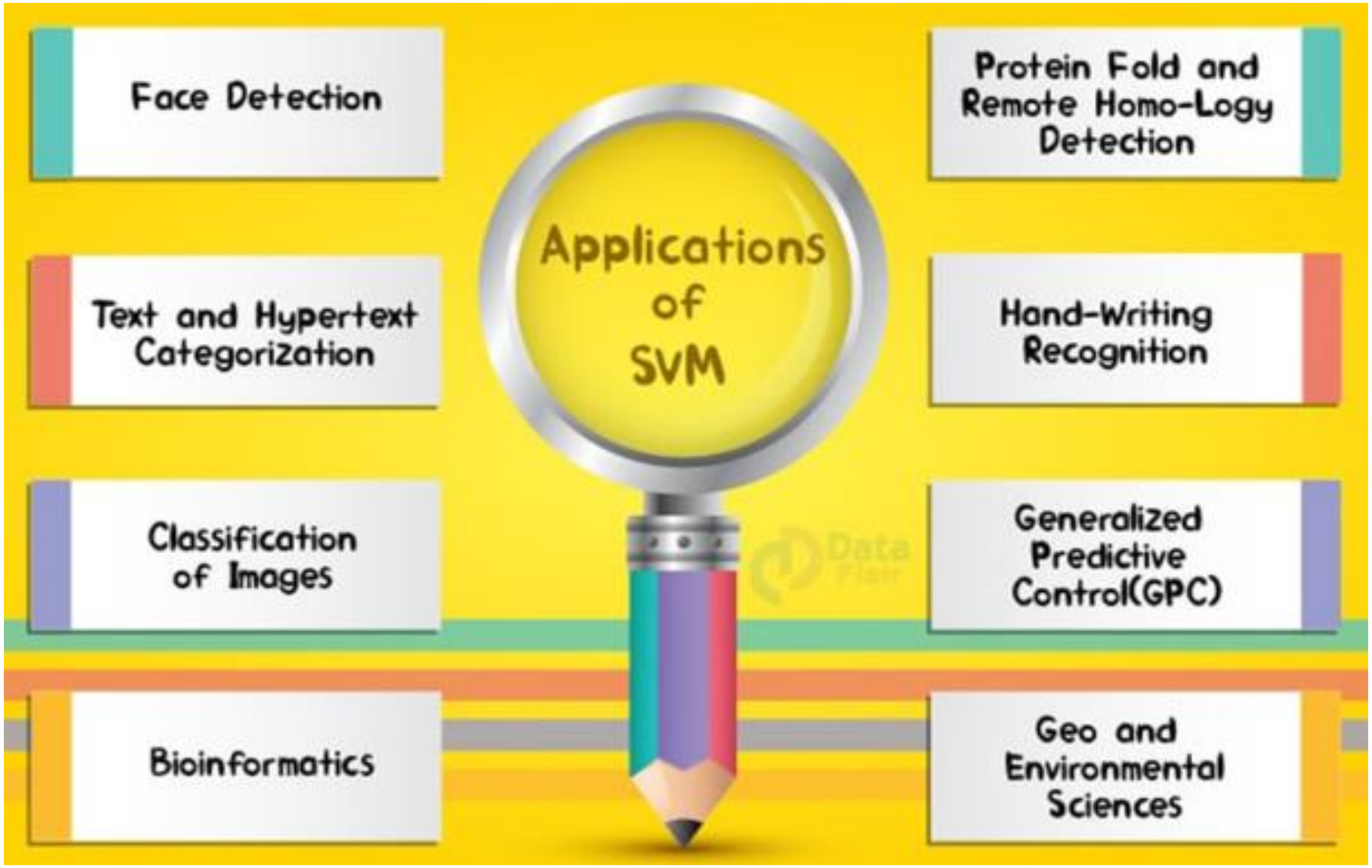
$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^T\mathbf{x} + b$$

Properties of SVM

- Flexibility in choosing a similarity function
- Sparseness of solution when dealing with large data sets
  - Only support vectors are used to specify the separating hyperplane
  - Therefore SVM also called sparse kernel machine.
- Ability to handle large feature spaces
  - complexity does not depend on the dimensionality of the feature space
- Overfitting can be controlled by soft margin approach
- Nice math property: a simple convex optimization problem which is guaranteed to converge to a single global solution
- Feature Selection

SVM Applications

SVM has been used successfully in many real-world problems



Application : Text Categorization

- Task: The classification of natural text (or hypertext) documents into a fixed number of predefined categories based on their content.

  A document can be assigned to more than one category, so this can be viewed as a series of binary classification problems, one for each category

Text Categorization using SVM

- **The distance between two documents is $\phi(x)\cdot\phi(z)$**

- **$K(x,z) = \phi(x)\cdot\phi(z)$ is a valid kernel, SVM can be used with $K(x,z)$ for discrimination.**

- **Why SVM?**

-High dimensional input space

-Few irrelevant features (dense concept)

-Sparse document vectors (sparse instances)

-Text categorization problems are linearly separable

Reference

- **Support Vector Machine Classification of Microarray Gene Expression Data**, Michael P. S. Brown William Noble Grundy, David Lin, Nello Cristianini, Charles Sugnet, Manuel Ares, Jr., David Haussler
- **Text categorization with Support Vector Machines: learning with many relevant features**

  T. Joachims, ECML - 98
- Christopher Bishop: Pattern Recognition and Machine Learning, Springer International Edition
- **A Tutorial on Support Vector Machines for Pattern Recognition**, Kluwer Academic Publishers - Christopher J.C. Burges

Good Web References for SVM

- http://www.cs.utexas.edu/users/mooney/cs391L/
- https://www.coursera.org/learn/machine-learning/home/week/7
- 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

Good Web References for SVM

- https://www.coursera.org/learn/machine-learning/home/week/7
- https://www.youtube.com/watch?time_continue=1&v=_PwhiWxHK8o
- https://www.youtube.com/watch?v=eh3sM4-3heo
- https://www.youtube.com/watch?time_continue=138&v=s8B4A5ubw6c
- https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
- https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
- https://data-flair.training/blogs/svm-kernel-functions/

Complementary slackness:

- https://www.youtube.com/watch?time_continue=722&v=Nbnd8KxRHGU&feature=emb_logo

SVM code:

- https://www.youtube.com/watch?v=TtKF996oEl8
- https://stackabuse.com/implementing-svm-and-kernel-svm-with-pythons-scikit-learn/

Thank You

Dr. Chetana Gavankar has over 24 years of Teaching, Research and Industry experience. She has published papers in peer reviewed international conferences and journals. She is also reviewer for multiple conferences and journals. She has worked on different projects with multiple industries and received awards for her research work. Her areas of research interests include Natural Language Processing, Information Retrieval, Web Mining and Semantic Web, Ontology, Big Data Analytics, Machine learning, Deep learning and Artificial Intelligence.