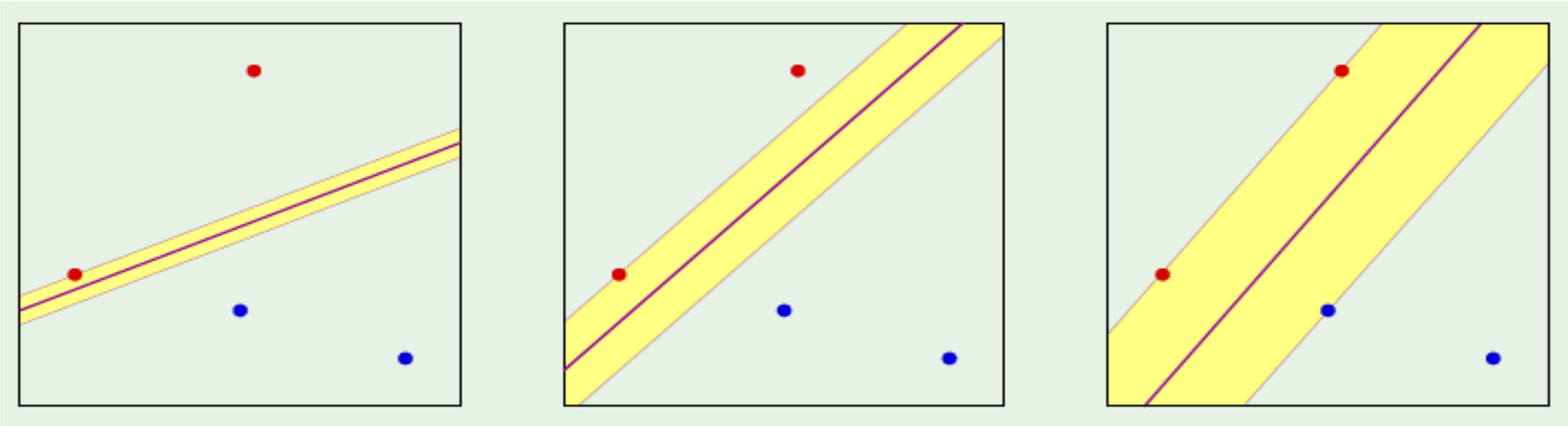
CS 60050
Machine Learning

Support Vector Machines

Some slides taken from course materials of Abu Mostafa

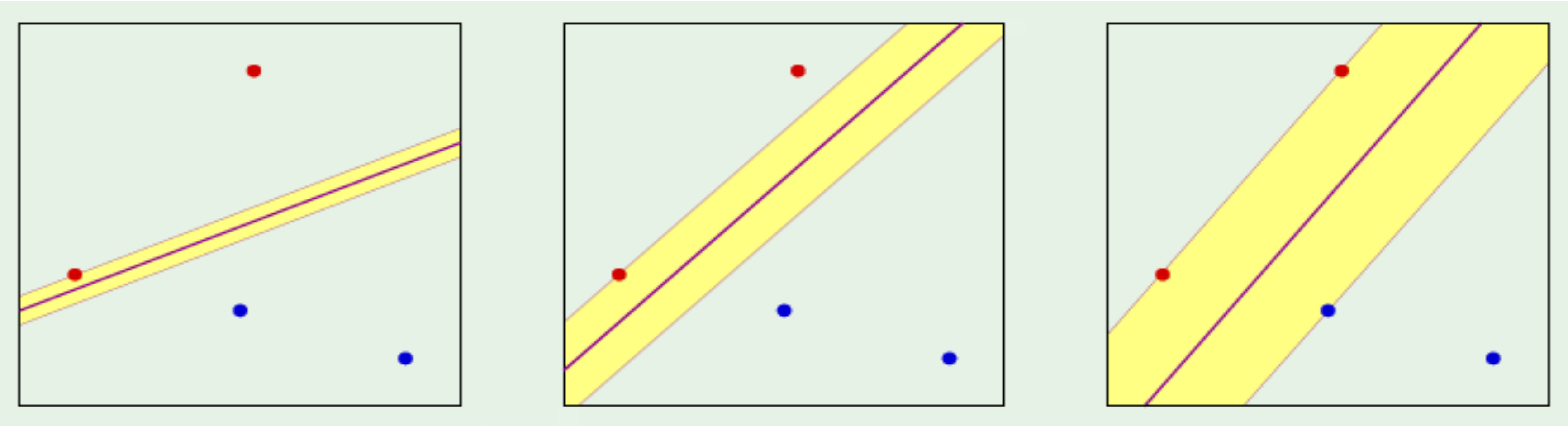
Intuition



- Consider a linearly separable dataset with 2 features
- Many possible separators. Each of the separators shown has 100% accuracy

- Which is the best?
  - In terms of generalization to unseen data?

Intuition



- Many possible separators. Which is the best?
- That one is best which is farthest away from all training points

- Margin: distance from the nearest data point to the separator
- Bigger margin is better → better generalization to unknown data
- SVMs guarantee to find the separator with the biggest margin

Finding the decision boundary

- We want to find the decision boundary that not only classifies all the points correctly but also maximizes the margin
- Assume d-dimensional feature space
- Decision boundary in d-dimensional feature space: a (hyper)plane
- We assume data is linearly separable; the separating hyperplane will not touch any point

Notations

- Training set: ($x^{(j)}$, $y^{(j)}$), j = 1, 2, …, N,
  - Each $x^{(j)}$ is a vector of d dimensions
  - Each $y^{(j)}$ = +1 or -1
- Separating plane: $w^T x = 0$ (vector notation)
  - Vector w = (w_0, w_1, …, w_d)
  - w_j are the parameters to learn
- Question: Which w maximizes the margin?

Two preliminary technicalities (to simplify the math)

- Let x_n be the nearest data point to the plane $w^T x = 0$
- (1) Multiplying all w's by any constant factor still gives the same plane. Hence we normalize w such that $| w^T x_n | = 1$
  - This normalization does not reduce generality – we are not missing any planes

Two preliminary technicalities (to simplify the math)

- Let x_n be the nearest data point to the plane $w^T x = 0$
- (1) Normalize w such that $| w^T x_n | = 1$
- (2) Pull out w_0, so that $w = (w_1, \ldots, w_d)$. Insert constant $b = w_0\, x_0$.
  - Remember: data points are of d dimensions $x_1, x_2, \ldots, x_d$. x_0 is a dummy dimension added by us
- Plane is now $w^T x + b = 0$, normalized such that $| w^T x_n + b | = 1$

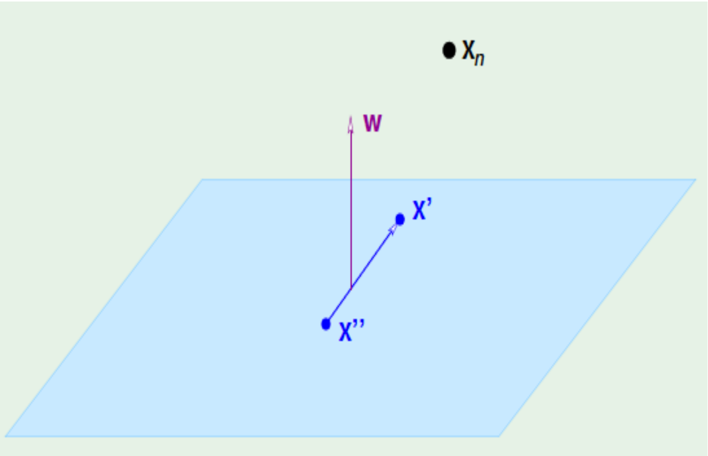
Computing the margin

The distance between $\mathbf{x}_n$ and the plane $\mathbf{w}^\intercal \mathbf{x} + b = 0$

where $|\mathbf{w}^\intercal \mathbf{x}_n + b| = 1$

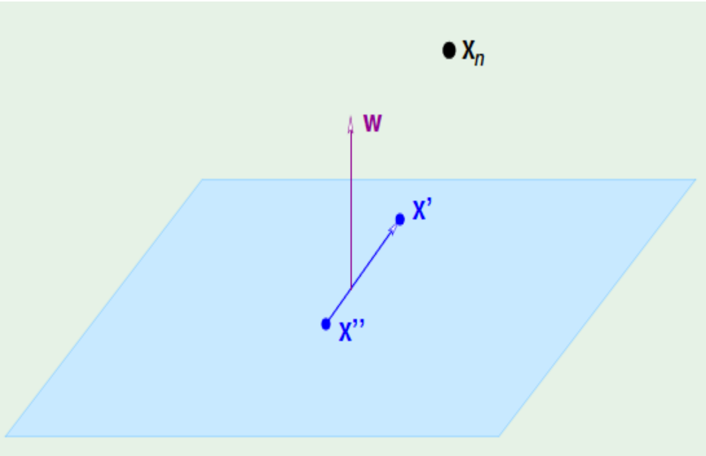
Computing the margin

Proposition:
The vector w is orthogonal to the plane in the *X* space



Computing the margin

Proposition:
The vector w is orthogonal to the plane in the *X* space



Take any two points x' and x'' on the plane.

$w^T x' + b = 0$ and $w^T x'' + b = 0$
=> $w^T (x' - x'') = 0$

Hence w is orthogonal to any vector that lies on the plane => w is orthogonal to the plane

Margin: distance between x_n and the plane

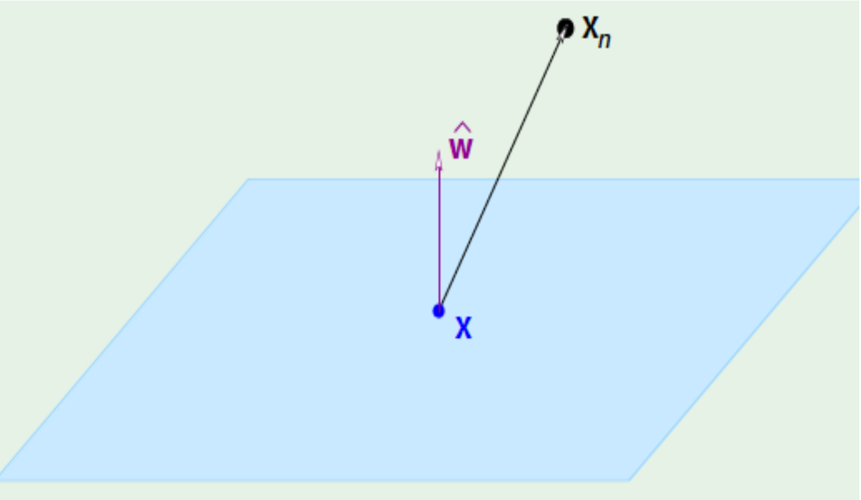
Take any point $\mathbf{x}$ on the plane

Projection of $\mathbf{x}_n - \mathbf{x}$ on $\mathbf{w}$ (direction orthogonal to the plane)

$$\hat{\mathbf{w}} = \frac{\mathbf{w}}{\|\mathbf{w}\|} \implies \text{distance} = \left|\hat{\mathbf{w}}^{\mathsf{T}}(\mathbf{x}_n - \mathbf{x})\right|$$

Projection of the vector $x_n - x$ along w computed by taking the vector product of $x_n - x$ with the unit vector in the direction of w

||w|| is the norm of w



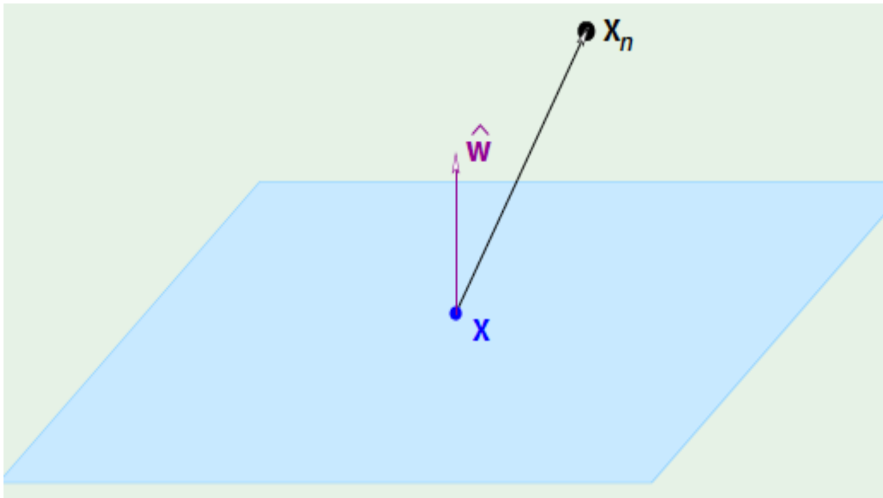
Margin: distance between x_n and the plane

$$\text{distance} = \frac{1}{\|\mathbf{w}\|}\left|\mathbf{w}^{\mathsf{T}}\mathbf{x}_n - \mathbf{w}^{\mathsf{T}}\mathbf{x}\right| =$$

$$\frac{1}{\|\mathbf{w}\|}\left|\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b - \mathbf{w}^{\mathsf{T}}\mathbf{x} - b\right| = \frac{1}{\|\mathbf{w}\|}$$

$w^T x + b$ is the equation of the plane at a point x on the plane. Hence 0.

$| w^T x_n + b | = 1$ for the nearest point x_n (due to our normalization)



The optimization problem

Maximize $\dfrac{1}{\|\mathbf{w}\|}$

subject to $\min\limits_{n=1,2,\ldots,N} |\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b| = 1$

The optimization problem

$$\text{Maximize } \frac{1}{\|\mathbf{w}\|}$$

$$\text{subject to } \min_{n=1,2,\ldots,N} |\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b| = 1$$

This optimization problem is too complex, because of

(i) the norm in the objective function, and

(ii) the minimum term in the constraints

Can we find an equivalent optimization problem that is easier to tackle?

Simplifying the optimization problem

$$\text{Maximize } \frac{1}{\|\mathbf{w}\|}$$

$$\text{subject to } \min_{n=1,2,\ldots,N} \left|\mathbf{w}^{\intercal}\mathbf{x}_n + b\right| = 1$$

Maximizing 1 / ||w||

Equivalent to

Minimizing (w^T w)

Simplifying the optimization problem

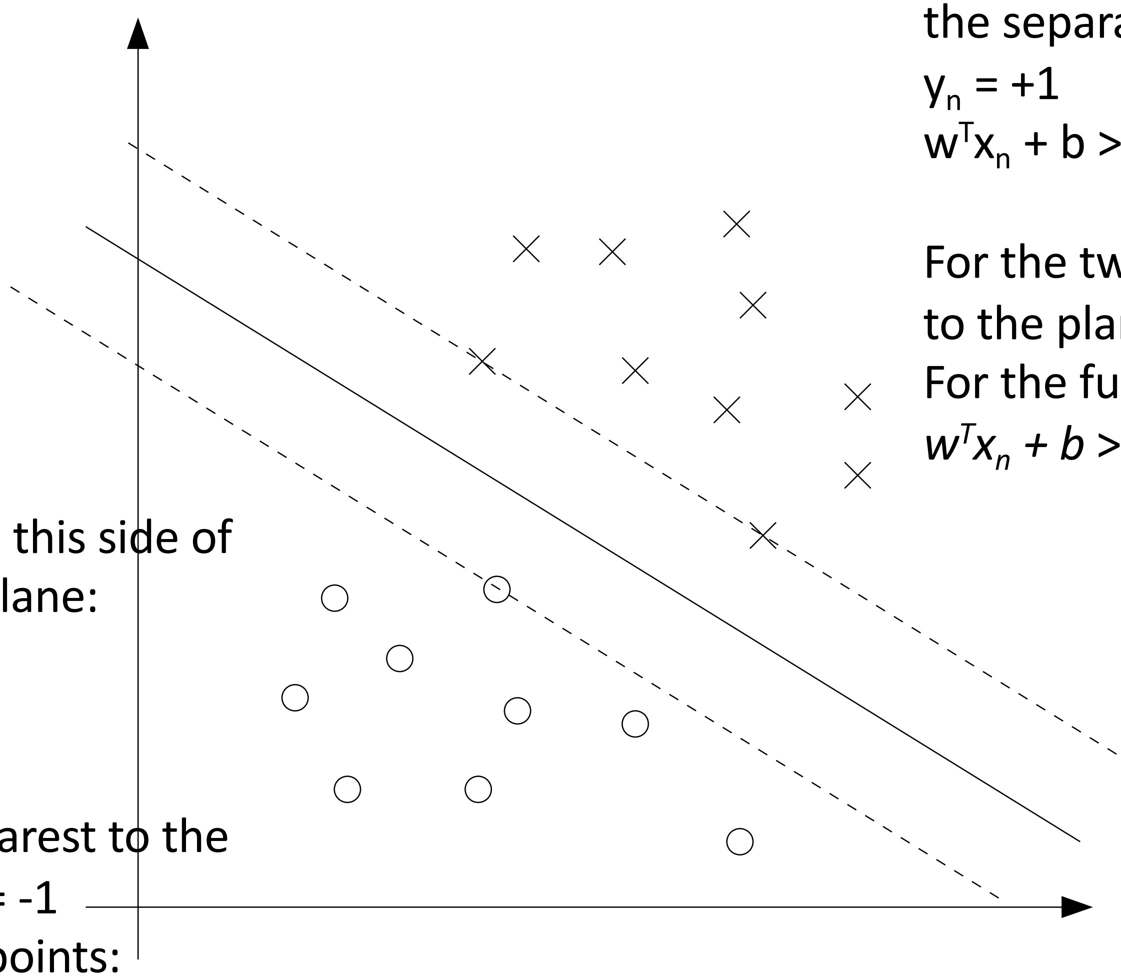
$$\text{Maximize } \frac{1}{\|\mathbf{w}\|}$$

$$\text{subject to } \min_{n=1,2,\ldots,N} |\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b| \;=\; 1$$

$$\text{Notice: } |\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b| \;=\; y_n\left(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b\right)$$

(assuming all points are classified correctly)

The geometry



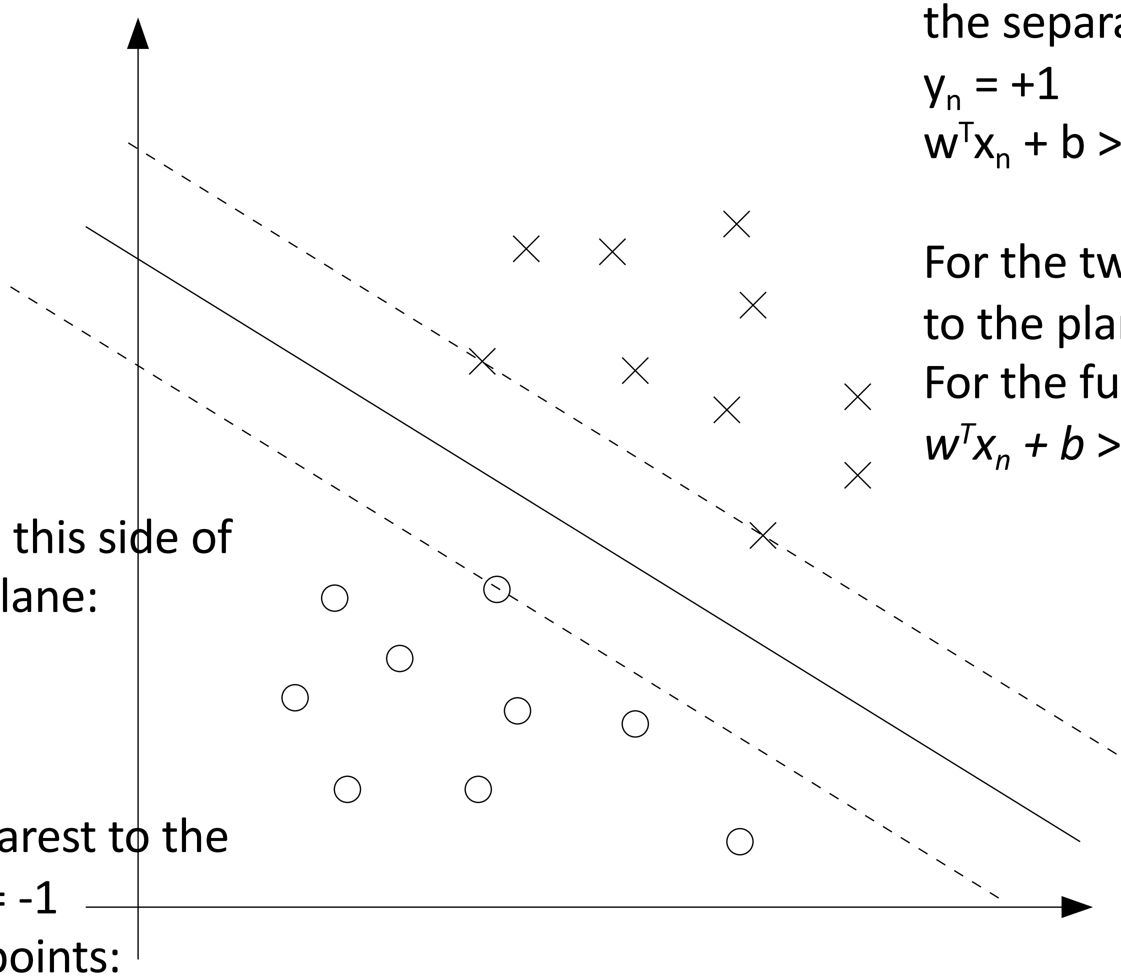
For any point on this side of the separating plane:
$y_n = +1$
$w^T x_n + b > 0$

For the two points nearest to the plane: $w^T x_n + b = 1$
For the further points:
$w^T x_n + b > 1$

For any point on this side of the separating plane:
$y_n = -1$
$w^T x_n + b < 0$

For the point nearest to the plane: $w^T x_n + b = -1$
For the further points:
$w^T x_n + b < -1$

The geometry



For any point on this side of the separating plane:
$y_n = +1$
$w^T x_n + b > 0$

For the two points nearest to the plane: $w^T x_n + b = 1$
For the further points:
$w^T x_n + b > 1$

For any point on this side of the separating plane:
$y_n = -1$
$w^T x_n + b < 0$

For the point nearest to the plane: $w^T x_n + b = -1$
For the further points:
$w^T x_n + b < -1$

Notice: $|\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b| = y_n(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b)$

Equivalent optimization problem

Maximize $\dfrac{1}{\|\mathbf{w}\|}$

subject to $\min\limits_{n=1,2,\ldots,N} |\mathbf{w}^\intercal \mathbf{x}_n + b| = 1$

Notice: $|\mathbf{w}^\intercal \mathbf{x}_n + b| = y_n \left(\mathbf{w}^\intercal \mathbf{x}_n + b\right)$

Minimize $\dfrac{1}{2}\,\mathbf{w}^\intercal \mathbf{w}$

subject to $y_n \left(\mathbf{w}^\intercal \mathbf{x}_n + b\right) \geq 1 \quad$ for $\quad n = 1, 2, \ldots, N$

Final optimization problem

Minimize $\quad \frac{1}{2}\,\mathbf{w}^{\intercal}\mathbf{w}$

subject to $\quad y_n\left(\mathbf{w}^{\intercal}\mathbf{x}_n + b\right) \geq 1 \quad$ for $\quad n = 1, 2, \ldots, N$

$\mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R}$

Solving the optimization problem

Solving the optimization

Minimize $\frac{1}{2}\,\mathbf{w}^{\mathsf{T}}\mathbf{w}$

subject to $y_n\left(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b\right) \geq 1 \quad$ for $\quad n = 1, 2, \ldots, N$

$\mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R}$

A way of solving constrained optimization problems: take the Lagrangian formulation of the problem

One issue: constraints are inequality constraints - handled by KKT conditions (due to Karush and Kuhn-Tucker)

Details out of scope of this course

Towards Lagrange formulation

Minimize $\frac{1}{2}\mathbf{w}^{\intercal}\mathbf{w}$

subject to $y_n\left(\mathbf{w}^{\intercal}\mathbf{x}_n + b\right) \geq 1$ for $n = 1, 2, \ldots, N$

$\mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R}$

For each constraint, consider a 'slack' quantity: difference between the left hand side and right hand side of the constraint

The slack quantities will be multiplied by 'Lagrange multipliers' α_n and will be made part of the objective function

Details out of scope of this course

Lagrange formulation

Minimize $\quad \frac{1}{2}\,\mathbf{w}^{\mathsf{T}}\mathbf{w}$

subject to $\quad y_n\left(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b\right) \geq 1 \quad$ for $\quad n = 1, 2, \ldots, N$

$\mathbf{w} \in \mathbb{R}^d, \;\; b \in \mathbb{R}$

Minimize $\quad \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\,\mathbf{w}^{\mathsf{T}}\mathbf{w} - \sum_{n=1}^{N} \alpha_n \overbrace{\left(y_n\left(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b\right) - 1\right)}^{\text{slack}}$

w.r.t. $\mathbf{w}$ and b and maximize w.r.t. each $\alpha_n \geq 0$

Note: we have one Lagrange multiplier for each of the n data points

Lagrange formulation

Minimize $\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^{\intercal}\mathbf{w} - \sum_{n=1}^{N} \alpha_n (y_n (\mathbf{w}^{\intercal}\mathbf{x}_n + b) - 1)$

w.r.t. $\mathbf{w}$ and b and maximize w.r.t. each $\alpha_n \geq 0$

Let us consider the unconstrained case:

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n = \mathbf{0}$$

Vector differentiation

$$\frac{\partial \mathcal{L}}{\partial b} = - \sum_{n=1}^{N} \alpha_n y_n = 0$$

Scalar differentiation

Lagrange formulation

$$\text{Minimize} \quad \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\,\mathbf{w}^{\intercal}\mathbf{w} - \sum_{n=1}^{N} \alpha_n (y_n\,(\mathbf{w}^{\intercal}\mathbf{x}_n + b) - 1)$$

w.r.t. $\mathbf{w}$ and b and maximize w.r.t. each $\alpha_n \geq 0$

Substituting

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n \quad \text{and} \quad \sum_{n=1}^{N} \alpha_n y_n = 0$$

We get

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{x}_n^{\intercal} \mathbf{x}_m$$

Explaining the Lagrange formulation

$$L(w,b,\alpha) = \frac{1}{2} w^T w - \sum_{n=1}^{N} \alpha_n \left(y_n (w^T x_n + b) - 1 \right)$$

$$= \frac{1}{2} w^T w + \sum_{n=1}^{N} \alpha_n - \sum_{n=1}^{N} \alpha_n y_n x_n w^T - \sum_{n=1}^{N} b \alpha_n y_n$$

$$= \sum_{n=1}^{N} \alpha_n - b \sum_{n=1}^{N} \alpha_n y_n + \frac{1}{2} w^T w - \sum_{n=1}^{N} \alpha_n y_n x_n w^T$$

Since $\sum_{n=1}^{N} \alpha_n y_n = \emptyset$, second term vanishes

Since $w = \sum_{n=1}^{N} \alpha_n y_n x_n$, the last term

is actually equivalent to $w^T w$

Hence we get

$$= \sum_{n=1}^{N} \alpha_n - \frac{1}{2} w^T w$$

Again substituting $w = \sum_{n=1}^{N} \alpha_n y_n x_n$,

$$= \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m x_n^T x_m$$

Final constrained optimization

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{x}_n^{\mathsf{T}} \mathbf{x}_m$$

Maximize w.r.t. to $\boldsymbol{\alpha}$ <u>subject to</u>

$\alpha_n \geq 0$ for $n = 1, \cdots, N$ **and** $\sum_{n=1}^{N} \alpha_n y_n = 0$

Can be solved by Quadratic Programming, which gives us

$$\boldsymbol{\alpha} = \alpha_1, \cdots, \alpha_N$$

Details out of scope of this course

The solution

Solution: $\boldsymbol{\alpha} = \alpha_1, \cdots, \alpha_N$

$$\Longrightarrow \quad \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

KKT condition: For $n = 1, \cdots, N$

$$\alpha_n \underbrace{\left(y_n \left(\mathbf{w}^\top \mathbf{x}_n + b\right) - 1\right)}_{\text{slack}} = 0$$

For each data point x_n :
Either the slack is zero, or the Lagrange multiplier α_n is zero

α's for most points will be zero, only for few points α will be positive

$\alpha_n > 0 \Longrightarrow \mathbf{x}_n$ is a $\boxed{\text{support vector}}$

Support vectors

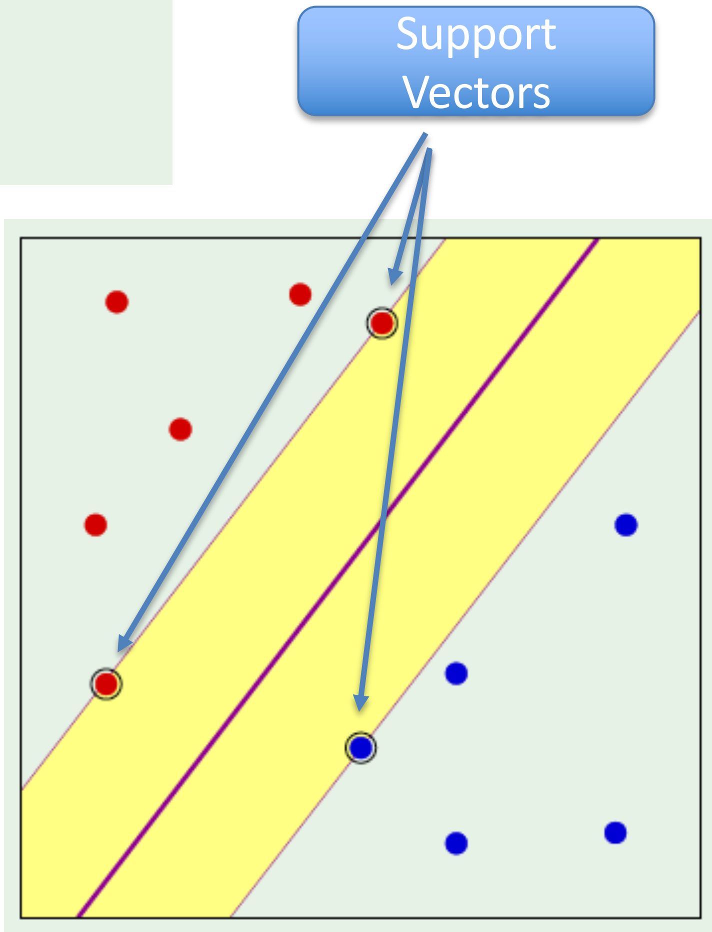
Closest $\mathbf{x}_n$'s to the plane: achieve the margin

$$\Longrightarrow \quad y_n\left(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b\right) = 1$$

$$\mathbf{w} = \sum_{\mathbf{x}_n \text{ is SV}} \alpha_n y_n \mathbf{x}_n$$

Solve for b using any SV:

$$y_n\left(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b\right) = 1$$



Hypothesis $g(x) = \text{sign}(\ w^T x + b\ )$

Advantage of SVM

- When we started, the number of parameters was the number of components of w vector
- Now, we see - the effective number of parameters is the number of SVs, which is much smaller (since most α's are zero)

- SVMs known to perform well over many types of data

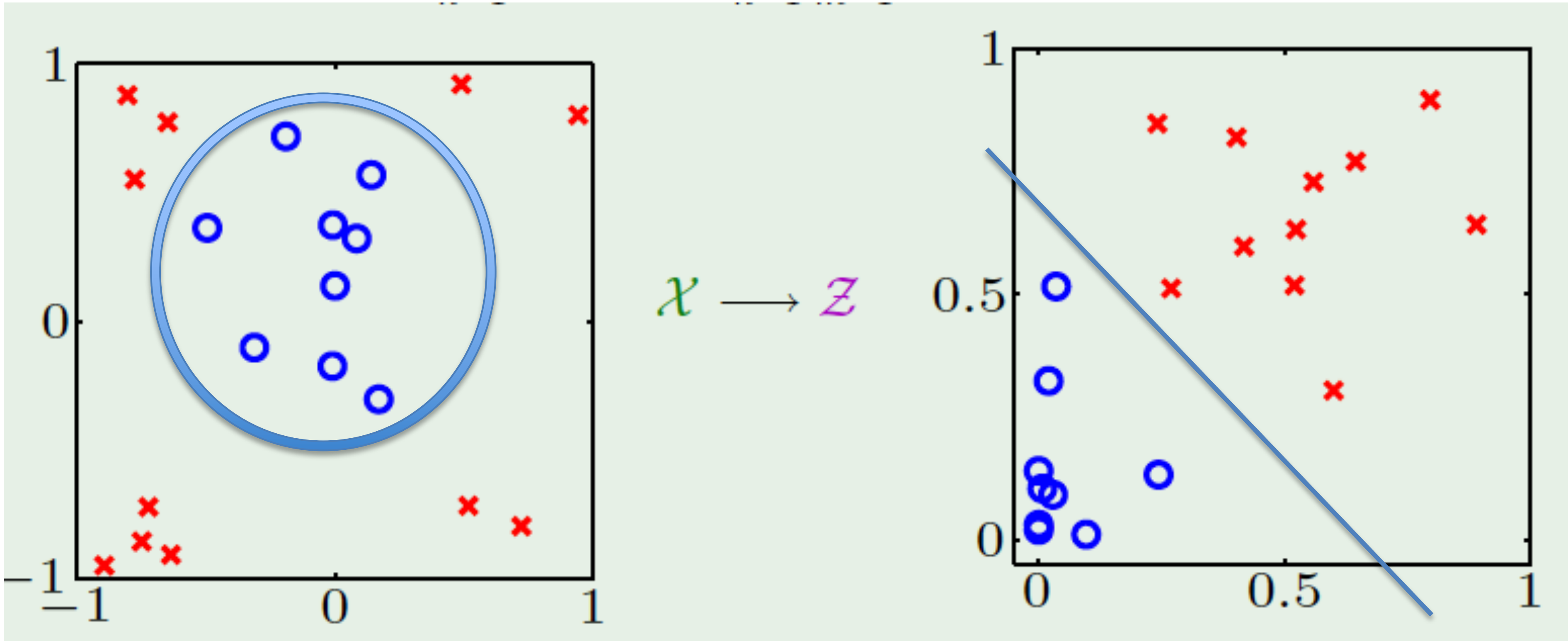
Extension of SVMs

- Till now, we considered linearly separable data
  - What we discussed is called “Hard margin SVM”
- What if the data is slightly non-linearly separable?
  - A variant called “Soft margin SVM”
  - Allows for few misclassifications (suitably penalized) in order to achieve large margin
- What if the data is highly non-linearly separable (complex decision boundary)?
  - We go for non-linear transforms

Non-linear transforms

Used when the data is non-linearly separable in the feature space

Nonlinear transforms



Non-linearly separable in original feature space

Linearly separable in some other space (usually higher dimensional)

Nonlinear transforms

- Points transformed from X-space to Z-space
- Optimization problem formulated in Z-space

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{z}_n^{\mathsf{T}} \mathbf{z}_m$$

- SVs found in Z-space (different Z-spaces can give different SVs)
- Complexity of optimization problem is independent of dimension of Z-space, only depends on number of points (N)

What do we need from the Z-space?

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{z}_n^{\intercal} \mathbf{z}_m$$

Constraints: $\alpha_n \geq 0$ for $n = 1, \cdots, N$ and $\sum_{n=1}^{N} \alpha_n y_n = 0$

$$\boxed{g(\mathbf{x}) = \operatorname{sign}\left(\mathbf{w}^{\intercal}\mathbf{z} + b\right)}$$

where $\mathbf{w} = \sum_{\mathbf{z}_n \text{ is SV}} \alpha_n y_n \mathbf{z}_n$

and b: $y_m \left(\mathbf{w}^{\intercal}\mathbf{z}_m + b\right) = 1$

What do we need from the Z-space?

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{z}_n^{\intercal} \mathbf{z}_m$$

Constraints: $\alpha_n \geq 0$ for $n = 1, \cdots, N$ and $\sum_{n=1}^{N} \alpha_n y_n = 0$

$$\boxed{g(\mathbf{x}) = \operatorname{sign}\left(\mathbf{w}^{\intercal}\mathbf{z} + b\right)}$$

where $\mathbf{w} = \sum_{\mathbf{z}_n \text{ is SV}} \alpha_n y_n \mathbf{z}_n$

and b: $y_m \left(\mathbf{w}^{\intercal}\mathbf{z}_m + b\right) = 1$

need $\mathbf{z}_n^{\intercal}\mathbf{z}$

What do we need from the Z-space?

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \, \alpha_n \alpha_m \, \mathbf{z}_n^{\intercal} \mathbf{z}_m$$

Constraints: $\alpha_n \geq 0$ for $n = 1, \cdots, N$ and $\sum_{n=1}^{N} \alpha_n y_n = 0$

$$\boxed{g(\mathbf{x}) = \operatorname{sign}\left(\mathbf{w}^{\intercal} \mathbf{z} + b\right)}$$

need $\mathbf{z}_n^{\intercal} \mathbf{z}$

where $\mathbf{w} = \sum_{\mathbf{z}_n \text{ is SV}} \alpha_n y_n \mathbf{z}_n$

and b: $y_m \left(\mathbf{w}^{\intercal} \mathbf{z}_m + b\right) = 1$

need $\mathbf{z}_n^{\intercal} \mathbf{z}_m$

What do we need from the Z-space?

$$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{z}_n^{\intercal} \mathbf{z}_m$$

Constraints: $\alpha_n \geq 0$ for $n = 1, \cdots, N$ and $\sum_{n=1}^{N} \alpha_n y_n = 0$

$$\boxed{g(\mathbf{x}) = \mathrm{sign}\left(\mathbf{w}^{\intercal}\mathbf{z} + b\right)}$$

need $\mathbf{z}_n^{\intercal}\mathbf{z}$

where $\mathbf{w} = \sum_{\mathbf{z}_n \text{ is SV}} \alpha_n y_n \mathbf{z}_n$

and b: $y_m\left(\mathbf{w}^{\intercal}\mathbf{z}_m + b\right) = 1$

need $\mathbf{z}_n^{\intercal}\mathbf{z}_m$

Need only inner products of vectors in the Z-space

Inner products in Z-space

- Given two vectors x and x’ (in original feature space)
- Which is easier:
  – Getting the transformed vectors z and z’ in Z-space
  – Getting the inner product of z and z’

- Can we compute inner products in Z-space without transforming vectors to Z-space?

Kernel function

- A kernel function is a function of x and x', such that the value K(x, x') is an inner product of two vectors in some Z-space

- Given two points $x, x' \varepsilon X$, $z^T z' = K(x, x')$

- Allows computation of the inner product of transformed vectors in the Z-space, without needing to transform the vectors to the Z-space

Kernel function: an example

Assume original feature space X has two dimensions

$x = (x_1, x_2)$

$x' = (x_1' , x_2')$

Consider the following function:

Consider $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\intercal \mathbf{x}')^2 = (1 + x_1 x'_1 + x_2 x'_2)^2$

$$= 1 + x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2 x_1 x'_1 + 2 x_2 x'_2 + 2 x_1 x'_1 x_2 x'_2$$

Is K a kernel function?

Yes, K is a kernel function

Consider $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^{\intercal}\mathbf{x}')^2 = (1 + x_1 x'_1 + x_2 x'_2)^2$

$$= 1 + x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2x_1 x'_1 + 2x_2 x'_2 + 2x_1 x'_1 x_2 x'_2$$

This is an inner product!

x → z = $(\ 1\ ,\ x_1^2\ ,\ x_2^2\ ,\ \sqrt{2}\,x_1\ ,\ \sqrt{2}\,x_2\ ,\ \sqrt{2}\ x_1 x_2\ )$

x' → z' = $(\ 1\ ,\ x_1'^2\ ,\ x_2'^2\ ,\ \sqrt{2}\,x'_1\ ,\ \sqrt{2}\,x'_2\ ,\ \sqrt{2}\,x'_1 x'_2\ )$

What functions are valid kernel functions?

- For a function to be a valid kernel function, it has to obey several properties
  - Be continuous
  - Be symmetric
  - Obey Mercer’s condition

- You can design your own kernel, provided it satisfies the conditions

Details out of scope of this course

Several well-known kernels exist

- Polynomial kernel: $K(x, z) = (1 + x^T z)^d$
  - d=1 gives linear kernel
  - d=2 gives quadratic kernel

- Radial Basis Function (RBF) kernel

$$K(\vec{x}, \vec{z}) = e^{-(\vec{x}-\vec{z})^2 / (2\sigma^2)}$$

Note: In this particular slide, x and z are vectors in the original feature space (this is different from the rest of the slides, where the symbol z has been used to denote the transformation of x to the Z-space)

Summary: The kernel trick

- Helps to perform the classification in a high-dimensional space (as compared to original feature space)
  - Advantage: data may be linearly separable (or at least, easier to separate) in a high-dimensional space
  - Need not pay much of a price in terms of computational complexity, since we do not have to actually transform the vectors to the high-dimensional space

- Z-space can be very high dimensional, even of infinite dimensions (e.g., for the RBF kernels)

THANK YOU

Questions can be mailed to Dr. S. Ghosh (saptarshi@cse.iitkgp.ac.in)