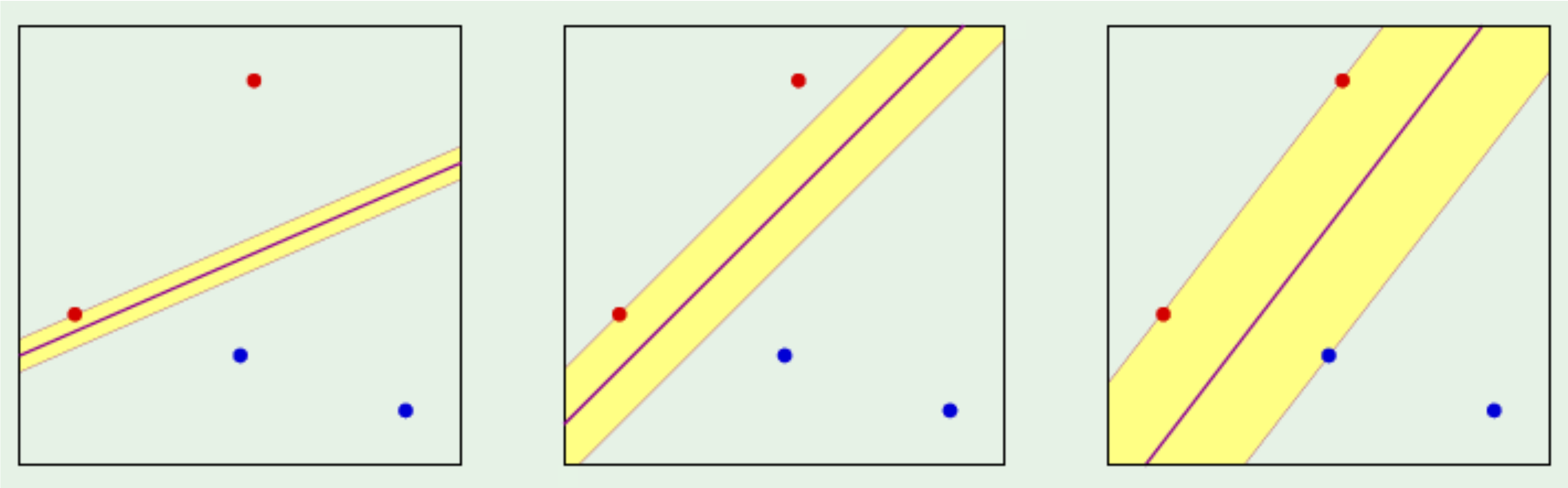
CS 60050
Machine Learning

Support Vector Machines

Some slides taken from course materials of Abu Mostafa

Intuition



- Many possible separating lines. Which separating line is the best?
- Margin: distance from the nearest example to the separating line
- Bigger margin is better → better generalization

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 plane will not touch any point

Notations

- Training set: (x_j, y_j), $j = 1, 2, \ldots, N$,
  - Each x_j is a vector of d dimensions
  - Each $y_j = +1$ or -1
- Separating plane: $\Sigma\, w_j\, x_j = 0$ where w_j are the parameters to learn
- Vector notation for the plane: $w^T x = 0$
  - Vector $w = (w_0, w_1, \ldots, w_d)$
- 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) Normalize w such that $| w^T x_n | = 1$
  - Multiplying all w's by a constant factor still gives the same plane
  - 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. Plane is now $w^T x + b = 0$, normalized such that $| w^T x_n + b| = 1$
  - Remember: data points are of d dimensions 1, …, d

Computing the margin

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

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

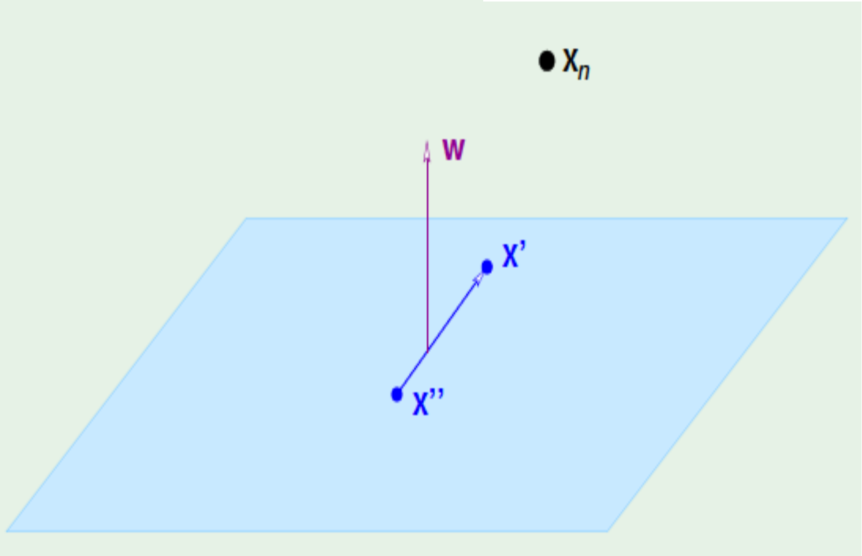
Computing the margin

The vector $\mathbf{w}$ is $\perp$ to the plane in the $\mathcal{X}$ space:

Take $\mathbf{x}'$ and $\mathbf{x}''$ on the plane

$$\mathbf{w}^{\mathsf{T}}\mathbf{x}' + b = 0 \quad \text{and} \quad \mathbf{w}^{\mathsf{T}}\mathbf{x}'' + b = 0$$

$$\Longrightarrow \quad \mathbf{w}^{\mathsf{T}}(\mathbf{x}' - \mathbf{x}'') = 0$$



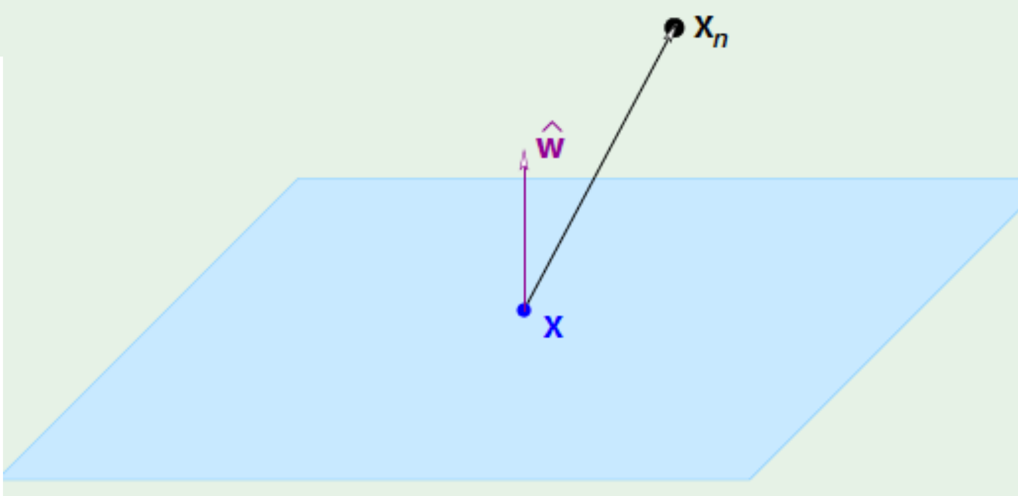
The vector w is orthogonal to a vector that lies on the plane

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)

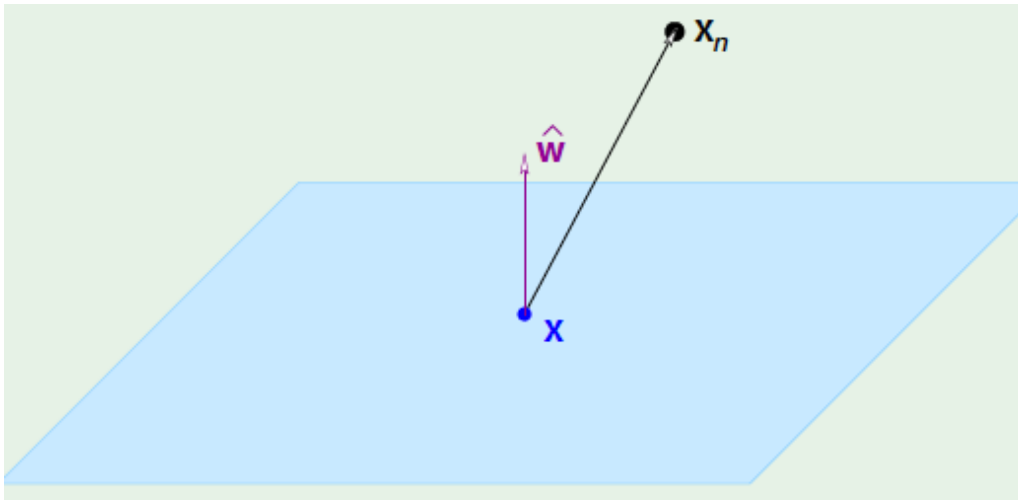
$$\hat{\mathbf{w}} = \frac{\mathbf{w}}{\|\mathbf{w}\|} \implies \text{distance} = \left|\hat{\mathbf{w}}^{\mathsf{T}}(\mathbf{x}_n - \mathbf{x})\right|$$



Distance between x_n and the plane

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

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



The optimization problem

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

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

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 is not a 'friendly' optimization problem, 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 more friendly?

Simplifying the optimization problem

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

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

Maximizing 1 / ||w||

Equivalent to

Minimizing (w^T w)

Simplifying the optimization problem

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

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

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

(assuming all points are classified correctly)

Equivalent optimization problem

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

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

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

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

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

Final optimization problem

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 $n = 1, 2, \ldots, N$

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

Solving the optimization problem

Solving the optimization

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

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

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

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)

Towards Lagrange formulation

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

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

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

For each equality constraint, consider a 'slack' (difference between the left hand side and right hand side)

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

Lagrange formulation

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

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

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

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

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

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

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

Substituting

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

We get

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

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

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

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 $n = 1, \cdots, N$

$$\alpha_n \left(y_n \left(\mathbf{w}^\intercal \mathbf{x}_n + b \right) - 1 \right) = 0$$

For all points:
Either the slack is zero, or the Lagrange multiplier α is zero

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

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

Support vectors

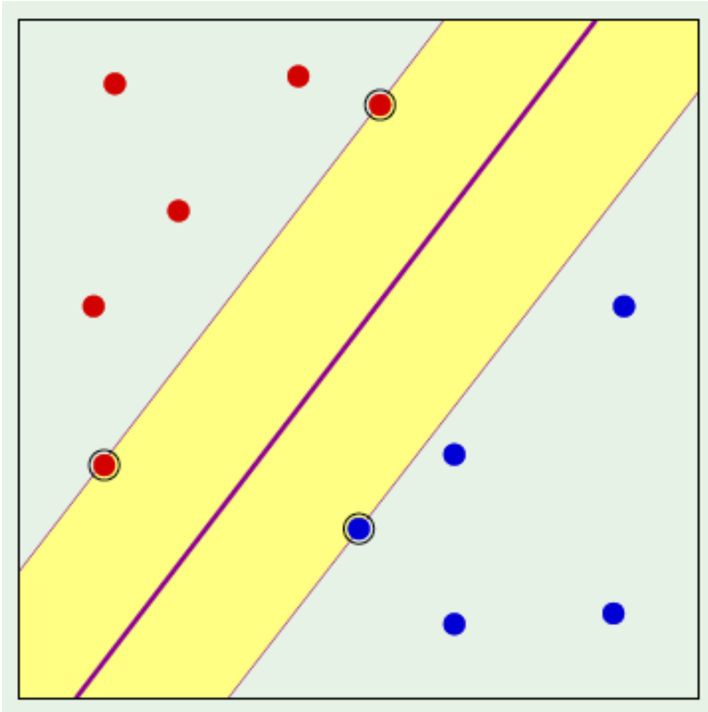
Closest $\mathbf{x}_n$'s to the plane: achieve the margin

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

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

Solve for b using any SV:

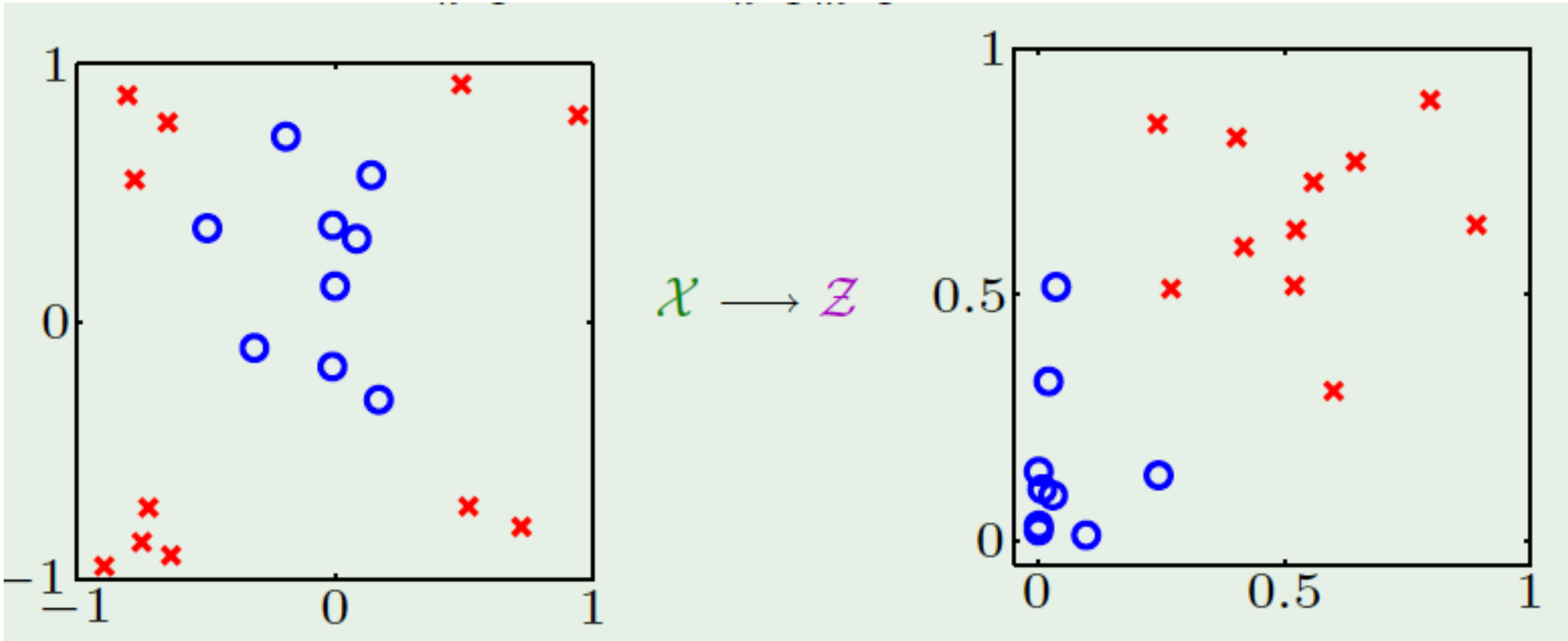
$$y_n\left(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b\right) = 1$$



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

Non-linear transforms

Nonlinear transforms



Non-linearly separable in original feature space

Linearly separable in some other space

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?

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

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

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

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

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

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 $\sum_{n=1}^{N} \alpha_n y_n = 0$

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

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

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

- Given two points *x, x' ε X*, let $z^T z' = K(x, x')$
- 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
- Allows computation of the inner product 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

We apply a 2nd order non-linear transformation ϕ

Example: $\mathbf{x} = (x_1, x_2) \longrightarrow$ 2nd-order Φ

$$\mathbf{z} = \Phi(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_2^2, x_1 x_2)$$

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

Can we compute K(x, x') without transforming x and x'?

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

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

This is an inner product!

$$(1 , x_1^2 , x_2^2 , \sqrt{2} x_1 , \sqrt{2} x_2 , \sqrt{2}\, x_1 x_2)$$

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

The kernel trick

- Get the classification done in a high-dimensional space, without paying 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 dimension

Several well-known kernels exist

- Polynomial kernel
- Exponential kernel
- Radial Basis Function (RBF) kernel

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