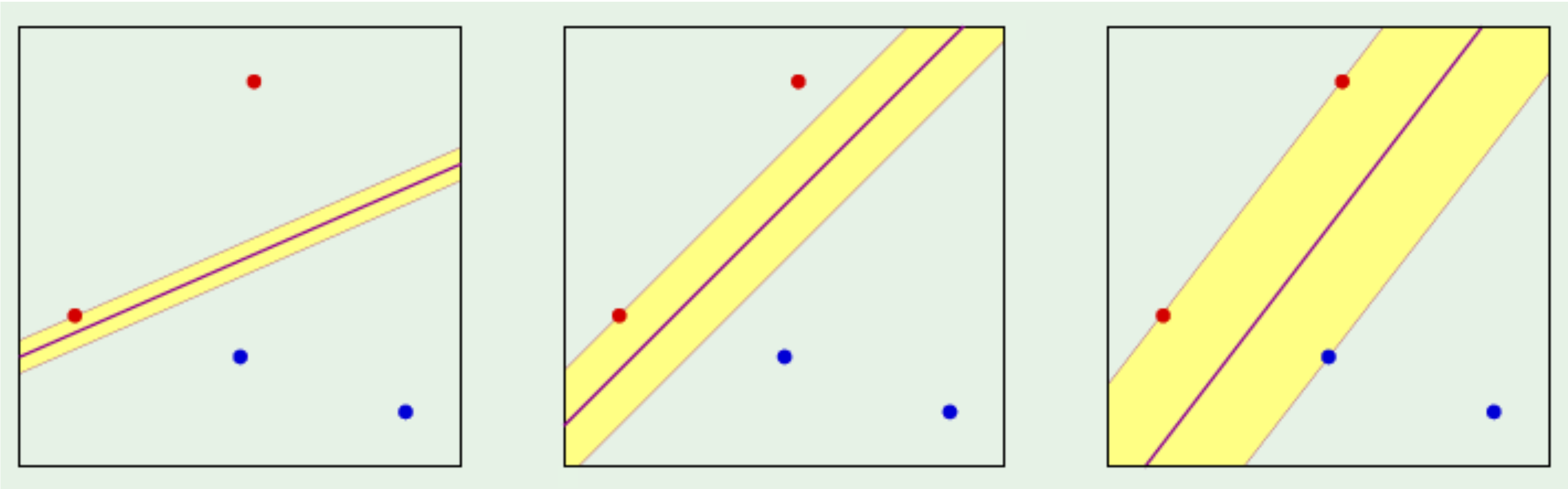
CS 60050
Machine Learning

Support Vector Machines

Intuition



- Many possible separating lines. Which separating line is the best?
- Margin: distance from the nearest example to the separating line

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?

Technicalities

- Let x_n be the nearest data point to the plane $w^T x = 0$
- Normalizing w such that $| w^T x_n | = 1$
- 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 $| w^T x_n + b| = 1$

Computing the margin

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

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

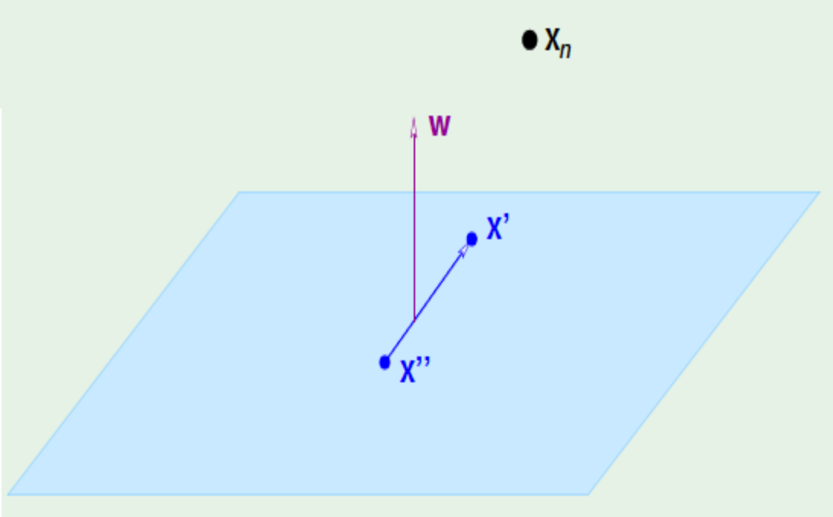
Computing the margin

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

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

$$\mathbf{w}^{\intercal}\mathbf{x}' + b = 0 \quad \text{and} \quad \mathbf{w}^{\intercal}\mathbf{x}'' + b = 0$$

$$\Longrightarrow \quad \mathbf{w}^{\intercal}(\mathbf{x}' - \mathbf{x}'') = 0$$

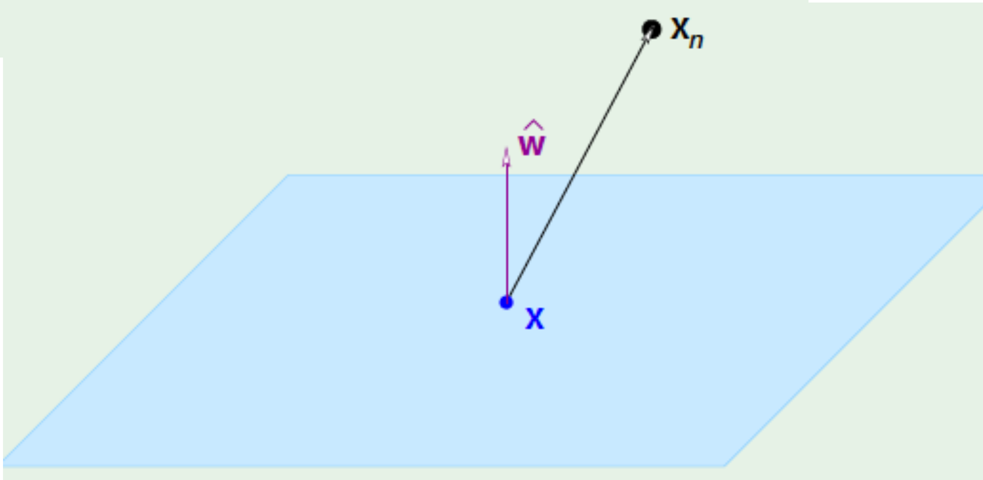


Distance between x_n and the plane

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

Projection of $\mathbf{x}_n - \mathbf{x}$ on $\mathbf{w}$

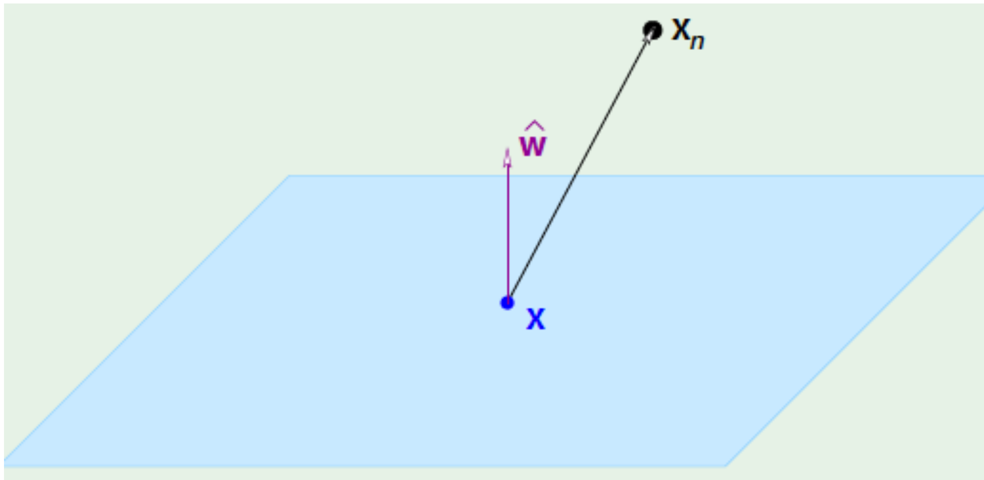
$$\hat{\mathbf{w}} = \frac{\mathbf{w}}{\|\mathbf{w}\|} \implies \text{distance} = \left|\hat{\mathbf{w}}^{\mathsf{T}}(\mathbf{x}_n - \mathbf{x})\right|$$



Distance between x_n and the plane

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

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



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

Equivalent optimization problem

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

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

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

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

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

Solving the optimization

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

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

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

Inequality constraints handled by KKT conditions
(due to Karush and Kuhn-Tucker)

Lagrange formulation

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

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

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

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

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 $\alpha_n \geq 0$

For the unconstrained case:

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

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

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 $\alpha_n \geq 0$

Substituting

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

We get

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

Final constrained optimization

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

Maximize w.r.t. to $\boldsymbol{\alpha}$ subject to

$\alpha_n \geq 0$ 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 $n = 1, \cdots, N$

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

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

Support vectors

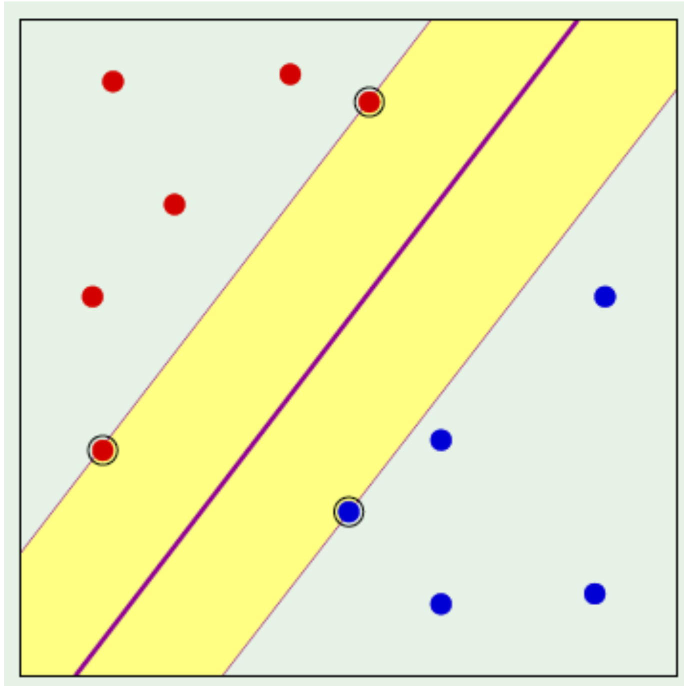
Closest $\mathbf{x}_n$'s to the plane: achieve the margin

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

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

Solve for b using any SV:

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



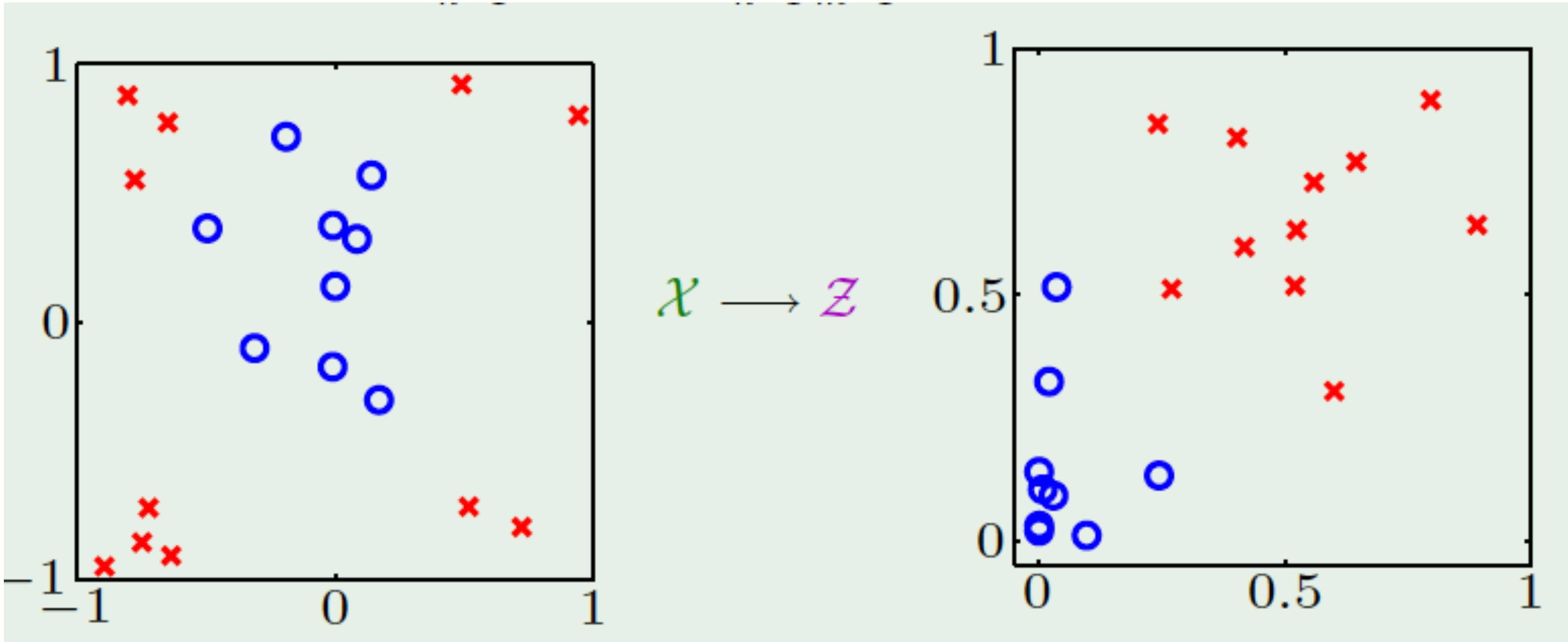
Hypothesis $g(x) = \text{sign}(\ w^T x + b\ )$

#SV's = #effective parameters

Generalization result

$$\mathbb{E}[E_{\text{out}}] \leq \frac{\mathbb{E}[\text{\# of SV's}]}{N-1}$$

Nonlinear transforms



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?

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

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

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

need $\mathbf{z}_n^{\mathsf{T}} \mathbf{z}$

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

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

need $\mathbf{z}_n^{\mathsf{T}} \mathbf{z}_m$

Need only inner products of vectors in the Z-space

Inner products in Z-space

- Given two vectors x and x’
- 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 an inner product of two vectors in some Z-space

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

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

$$K(\mathbf{x}, \mathbf{x}') = \mathbf{z}^\mathsf{T}\mathbf{z}' = 1 + x_1 x'_1 + x_2 x'_2 + x_1^2 x'^2_1 + x_2^2 x'^2_2 + x_1 x'_1 x_2 x'_2$$

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

Consider $K(\mathbf{x},\mathbf{x}') = (1+\mathbf{x}^{\mathsf{T}}\mathbf{x}')^2 = (1 + x_1 x'_1 + x_2 x'_2)^2$

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

This is an inner product!

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

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

The kernel trick

- Get the classification done in a high-dimensional space, without paying much of a price in terms of computational complexity
- Z-space can be very high dimensional, even of infinite dimension

$$K(x, x') = \exp\left(-(x - x')^2\right)$$

$$= \exp\left(-x^2\right) \exp\left(-x'^2\right) \underbrace{\sum_{k=0}^{\infty} \frac{2^k (x)^k (x')^k}{k!}}_{\exp(2xx')}$$

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

Designing your own kernel

$K(\mathbf{x}, \mathbf{x}')$ is a valid kernel iff

1. It is symmetric and

2. The matrix:

$$\begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & K(\mathbf{x}_1, \mathbf{x}_2) & \dots & K(\mathbf{x}_1, \mathbf{x}_N) \\ K(\mathbf{x}_2, \mathbf{x}_1) & K(\mathbf{x}_2, \mathbf{x}_2) & \dots & K(\mathbf{x}_2, \mathbf{x}_N) \\ \dots & \dots & \dots & \dots \\ K(\mathbf{x}_N, \mathbf{x}_1) & K(\mathbf{x}_N, \mathbf{x}_2) & \dots & K(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

is **positive semi-definite**

for any $\mathbf{x}_1, \cdots, \mathbf{x}_N$ (Mercer's condition)