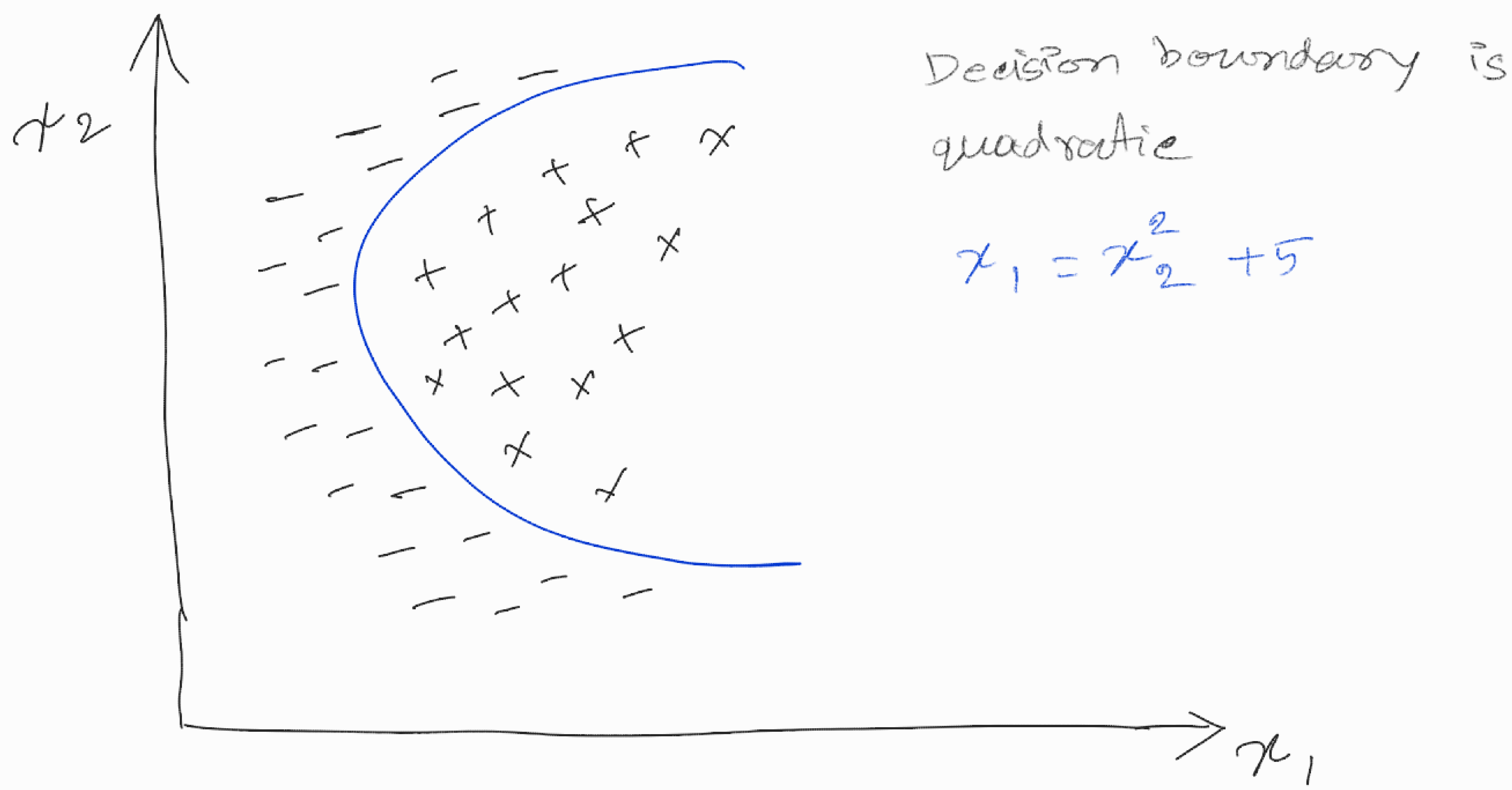
Kernel let us use linear classifier (Perceptron, linear regr) & get a decision boundary that is non linear.

SVM → kernel trick + optimum margin classifier

We take low dimensional data & embed it in higher dimensional space.



Decision boundary is quadratic in $X = (x_1, x_2) \in \mathbb{R}^2$
But it is linear in $\phi(x) = (x_1, x_2, x_1^2, x_2^2, x_1 x_2) \in \mathbb{R}^5$

$$x_1 = x_2^2 + 5$$
$$\Rightarrow x_1 - x_2^2 - 5 = 0$$

in $\phi(x)$ this equation will be,

$$\left.\begin{array}{l} w \cdot \phi(x) + b = 0 \\ w = [1, 0, 0, -1, 0] \\ \phi(x) = (x_1, x_2, x_1^2, x_2^2, x_1 x_2) \end{array}\right\} \text{this is linear.}$$

$b = -5$

What if input feature have d dimension, $x \in \mathbb{R}^d$ and decision boundary is quadratic.

$$\phi(x) = (x_1, x_2 \cdots\cdots\cdots x_d \longrightarrow d$$
$$x_1^2, x_2^2 \cdots\cdots\cdots x_d^2 \longrightarrow d$$
$$x_1x_2, x_1x_3 \cdots\cdots\cdots x_{d-1}x_d \longrightarrow {}^dC_2$$

$$\therefore \text{dim of } \phi(x) = d + d + {}^dC_2$$
$$= 2d + \frac{d(d-1)}{2}$$
$$\approx O(d^2)$$

This is called the basis expansion.

Let's say we want to expand the basis for MNIST dataset

$$x \in \mathbb{R}^{784}$$
$$\phi(x) \in \mathbb{R}^{308,504}$$

This is a problem our input feature now becomes 3,08,504. To solve this we will use kernel trick.

The kernel trick :- implement this without even write down a vector in the higher dimensional space ($\phi(x)$)

Let's do that using perceptron algorithm:-

Perceptron algorithm.

initialize $w=0,\ b=0$ $\qquad\qquad\qquad$ $\alpha=0$

while some training point is misclassified

$w = w + y^{(i)}\phi(x^{(i)})$

$b = b + y^{(i)}$

$\alpha_i = \alpha_i + 1$

$b = b + y^{(i)}$

$$w = \sum_i \alpha_i y^{(i)} \phi(x^{(i)})$$

α_i = # number of times an update occured at point i.

So w in dual form: $\alpha = (\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n)$

Now we will compute $w\cdot\phi(x)$ without writing $\phi(x)$

$$w\cdot\phi(x) = \sum_{i=1}^{m} \alpha_i y^{(i)} \left(\phi(x^{(i)}) \cdot \phi(x)\right)$$

We can compute $\phi(x)\cdot\phi(z)$ without ever writting $\phi(x)$ or $\phi(z)$.

suppose,

$x = (x_1, x_2)$ $\qquad$ $\phi(x) = (x_1, x_2, x_1^2, x_2^2, x_1 x_2)$

Let's tweak, $\phi(x) = (1 + \sqrt{2}x_1 + \sqrt{2}x_2 + \sqrt{2}x_1^2 + \sqrt{2}x_2^2 + \sqrt{2}x_1 x_2)$

twiking this doesnot change our decision boundary in $\phi(x)$ space.

$\phi(x)\cdot\phi(z) = (1 + \sqrt{2}x_1 + \sqrt{2}x_2 + \sqrt{2}x_1^2 + \sqrt{2}x_2^2 + \sqrt{2}x_1 x_2).$

$$\left(1+\sqrt{2}z_1+\sqrt{2}z_2+\sqrt{2}z_1^2+\sqrt{2}z_2^2+\sqrt{2}z_1z_2\right)$$

$$=\left(1+2x_1z_1+2x_2z_2+x_1^2z_1^2+x_2^2z_2^2+2x_1z_1x_2z_2\right)$$

$$=(1+x_1z_1+x_2z_2)^2$$

$$=(1+x\cdot z)^2$$

this is the original input feature which is smaller dimension.

to classify a new point x :-

$$\operatorname{sign}\left(w\cdot\phi(x)+b\right)$$

$$\Rightarrow \operatorname{sign}\left(\sum_{i=1}^{m}\alpha_i y^{(i)}\,\phi(x^{(j)})\cdot\phi(x)+b\right)$$

$$(1+x^{(j)}\cdot x)^2$$

so in training process we need to find $\alpha_i \in \mathbb{R}^m$

Lets implement kernel trick in SVM:-

From SVM we get equation-9 of SVM explanation

$$\max_{\alpha\in\mathbb{R}^m}\sum_{i=1}^{m}\alpha_i-\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}\left(x^{(i)}\cdot x^{(j)}\right)$$

$$\left(\phi(x^{(i)})\cdot\phi(x^{(j)})\right)$$

$$\text{s.t. } \sum\alpha_i y^{(i)}=0$$

$$0\le\alpha_i\le C$$

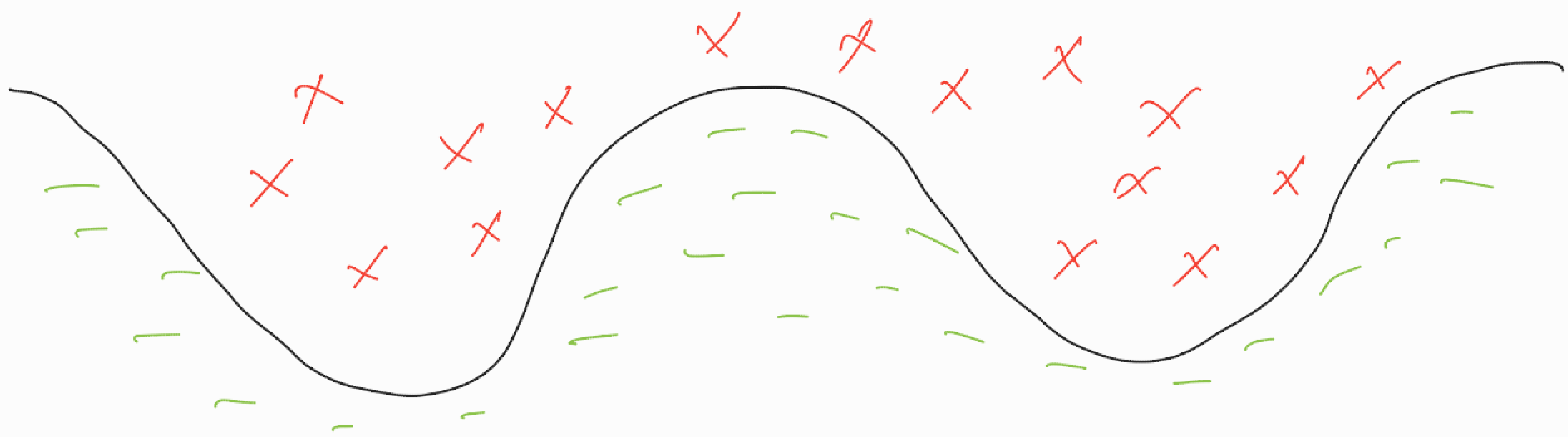
kernel trick for SVM

$$w=\sum_i \alpha_i y^{(i)}\phi(x^{(i)})$$

Given a new point x.

$$\text{sign}\left(\sum_i \alpha_i y^{i}\left(\phi(x^{i})\cdot\phi(x)\right)+b\right)$$

Polynomial decision boundary :-



this is a degree four decision boundary.

Let $\phi(x)$ consist of all terms of order $\leq p$
such as $x_1 x_2^2 x_3^{p-3}$

Dim of $\phi(x) = O(d^p)$ $x \in \mathbb{R}^d$, $p \Rightarrow$ degree of decision boundary

Same trick as before.

$$\phi(x)\cdot\phi(z) = (1 + x\cdot z)^p$$

Kernel function :-

Kernel function, $K(x,z) = \phi(x)\cdot\phi(z)$

we can of $K(x,z)$ as a measure of similarity between x & z.

Kernel perceptron :-

$\alpha = 0, \quad b = 0$

while some i has $y^{(i)} \left(\sum_j \alpha_j y^{(j)} K(x^{(j)}, x^{(i)}) \right) + b \le 0$

$$\alpha_i = \alpha_i + 1.$$
$$b = b + y^{(i)}$$

to classify new point x : $\text{sign}\left(\underbrace{\sum_j^m \alpha_j y^{(j)} K(x^{(j)}, x) + b}_{} \right)$

$$F(x) = \alpha_1 y^{(1)} K(x^{(1)}, x) + \cdots\cdots + \alpha_m y^{(m)} K(x^{(m)}, x) + b$$

suppose x is very similar to training point $x^{(i)}$ & very dissimilar to other traing example. so value of $K(x^{(i)}, x)$ will be higest. So value $y^{(i)}$ will get higest priority during prediction. & $y^{(i)}$'s value might be the prediction of x.

In the above example our similarity function was

$$K(x, z) = (x^T z)^2$$
$$= \sum_i^n x_i z_i \sum_j^n x_j z_j.$$
$$= \sum_{i,j=1}^n (x_i x_j)(z_i z_j)$$

Can we choose K to be any similarity function?

— No, need $K(x, z) = \phi(x) \cdot \phi(z)$ for some embedding ϕ

similarity function needs to correspond to dot product in some high dimensional extended feature space.

In other words given some similarity matrix k there should be some feature mapping ϕ so that $K(x,z) = \phi(x) \cdot \phi(z)$ for x, z; to satisfy this condition kernel matrix 'K' need to be a positive semi definite matrix.

$$K_{ij} = K(x^{(i)}, x^{(j)})$$

$$z^T K z \geq 0$$

this is also called Mercer theorem.

For example,

$$K(x,z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$$

is a valid kernel this is called gaussian kernel or RBF kernel.