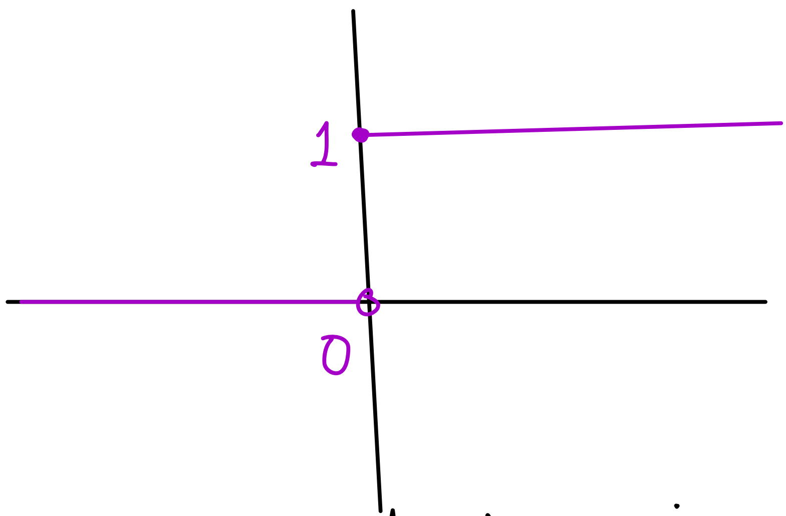
LOGISTIC PERCEPTRON

Perceptron

$x^{(i)} \in \mathbb{R}^d \quad y^{(i)} \in \{0,1\}$

$$h_\theta(x) = g(\theta^T x)$$

$$g(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

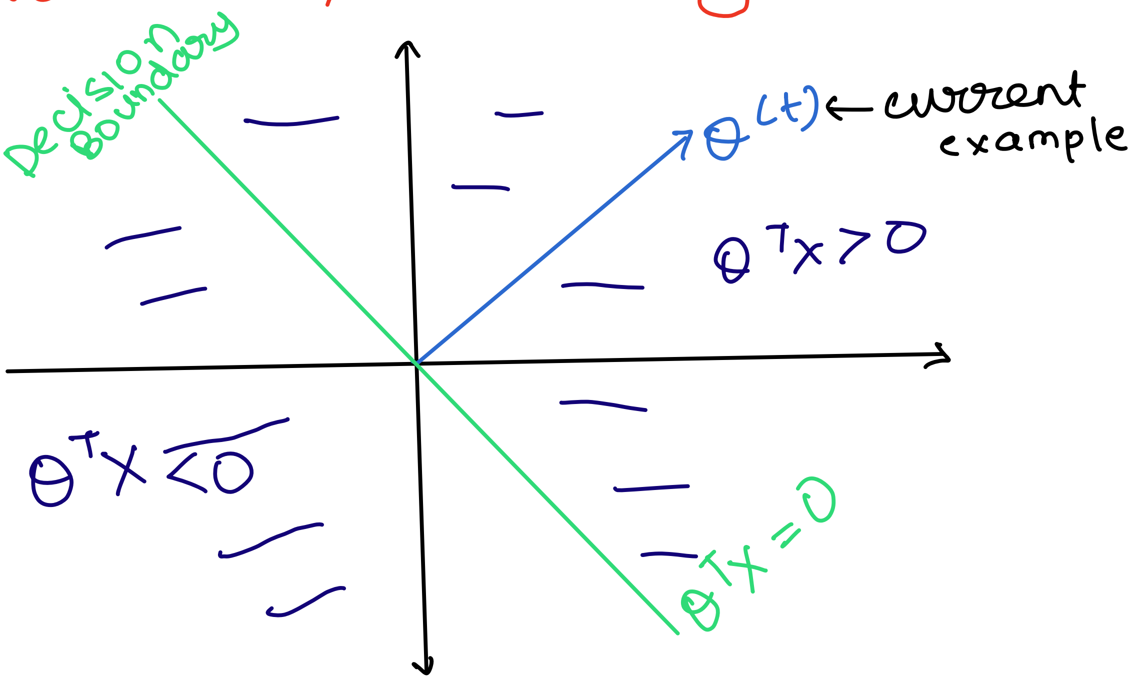


Also called streaming algorithm
You don't have access to all examples.
You take 1 eg. update then wait for next

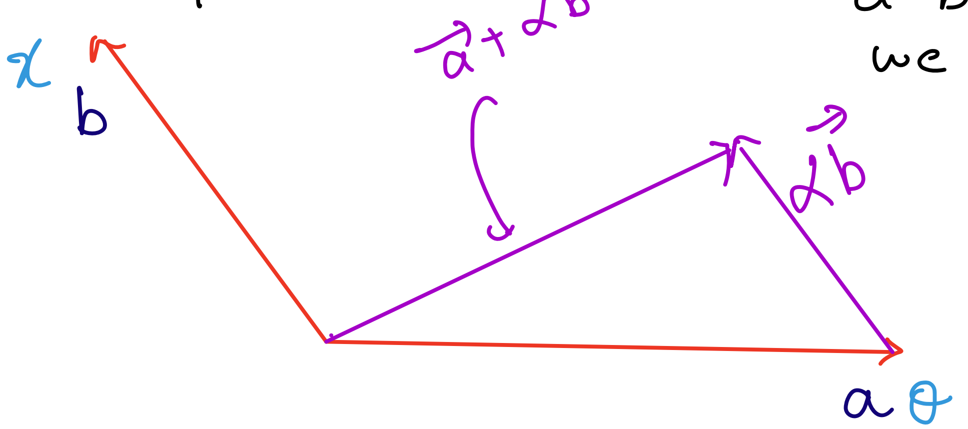
$\vec{\theta} := \text{Init}(\vec{0})$
for i in 1, 2 . . .

$$\theta := \theta + \alpha\left(y^{(i)} - h_\theta(x^{(i)})\right) \cdot x^{(i)}$$

[looks superficially similar to linear but in linear hypothesis was $\theta^T x$, here it is $g(\theta^T x)$



Example:



$a^T b < 0$

we want to make > 0

How??

$$(\vec{a}+\alpha\vec{b})^T b = (a^T+\alpha b^T)b$$
$$= a^T b + \alpha \underbrace{b^T b}_{\geq 0}$$
$$\geq a^T b$$

If we want to decrease, then subtract

$$\underbrace{\theta}_{\mathbb{R}^d} := \overset{\mathbb{R}^d}{\theta} + \alpha \overbrace{\left(y^{(i)} - h_\theta(x^{(i)})\right)}^{\mathbb{R}} . \overset{\mathbb{R}^d}{x^{(i)}}$$

→ If example is 0 and $h_\theta(x^{(i)})$ is 0, then don't update θ.
Similar for 1 (both)

→ If example is 1, output is 0, you want to increase value of dot product by adding some x in θ.

There are ∞ solutions for the seperating hyperplane, the model gives one. It changes as we change order of data passing.

LOGISTIC REGRESSION

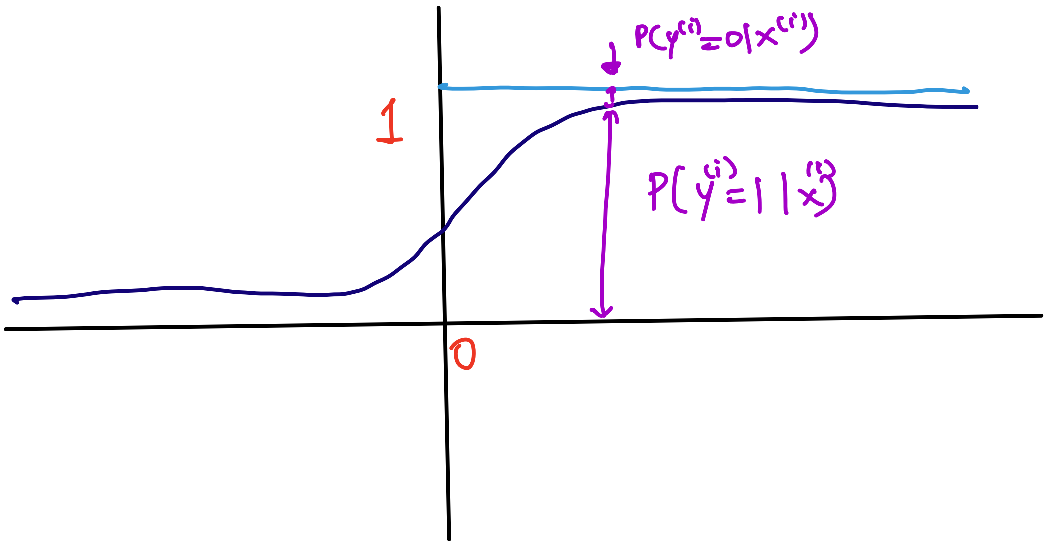
$x^{(i)} \in \mathbb{R}^d \qquad y^{(i)} \in \{0,1\}$

$y^{(i)} = 1$ positive example

$y^{(i)} = 0$ negative example

$$h_\theta(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{1+e^{-z}} \quad \text{(logistic function)}$$



$$P(y^{(i)}=1 | x^{(i)}; \theta) = h_\theta(x)$$

$$P(y^{(i)}=0 | x^{(i)}; \theta) = 1 - h_\theta(x)$$

$\theta^T x = 0 \qquad z = \theta^T x$

$g(z) = 0.5$

$$P(y|x;\theta) = [h_\theta(x)]^y \times [1-h_\theta(x)]^{1-y}$$

$$L(\theta) = \prod_{i=1}^{n} p(y^{(i)}|x^{(i)};\theta) \quad [\text{IID}]$$

$$\log L(\theta) = \ell(\theta) = \sum_{i=1}^{n} y\, h_\theta(x) + (1-y)(1-h_\theta(x))$$

Generative : $p(y,x)$

Discriminative: $p(y|x)$

$$g(z) = \frac{1}{1+e^{-z}}$$

$$g'(z) = g(z)(1-g(z))$$

$$\ell(\theta) = y \log g(\theta^T x) + (1-y)\log(1-g(\theta^T x))$$

$$\nabla_\theta \ell(\theta) = \frac{y}{g(\theta^T x)} \cdot g'(\theta^T x)\, x + \frac{(1-y)(-1)}{(1-g(\theta^T x))} \cdot g'(\theta^T x)\, x$$

$$\nabla_\theta \ell(\theta) = [y - h_\theta(x)] \cdot x$$

$$\theta := \theta + \Sigma \alpha [y - h_\theta(x)] x$$

Logistic → soft version of perceptron

We always perform updates as $h_\theta(x) \neq 1$ or 0

so even if it is predicting 1 for 0.7 still 0.3 difference makes update.

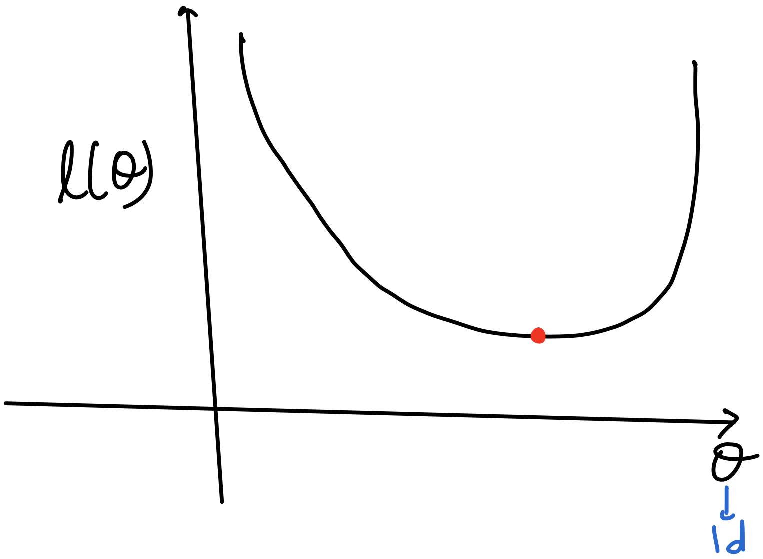
Logistic gives out probability

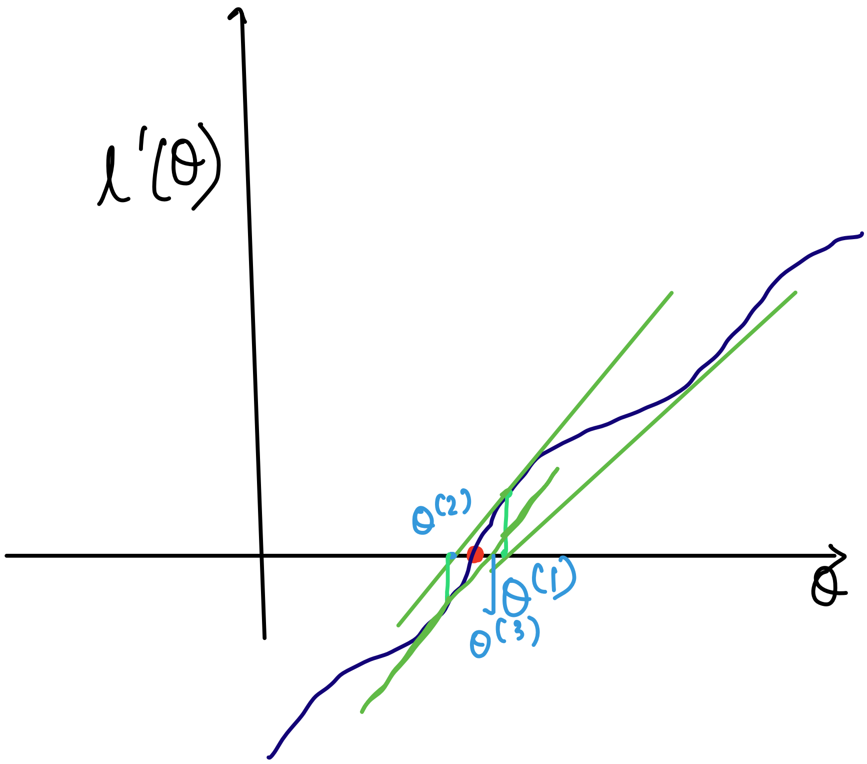
Perceptron → update for misclassified only

Logistic → updates for all x.

Newton's Method

It is alternative for S.G.D.





Root finding method $\rightarrow f(x)=0$

Apply Newton's Method to first gradient

Do linear approximation & then from

then one repeat.

Newton's Method converges pretty quickly as compared to G.D.

$$\theta^{(t+1)} = \theta^{(t)} - \frac{f(\theta)}{f'(\theta)}$$

We find roots of $l'(\theta)$

$$\theta^{(t+1)} = \theta^{(t)} - \frac{l'(\theta^{(t)})}{l''(\theta^{(t)})} \leftarrow \text{Scalar } \theta$$

$$\theta^{(t+1)} = \theta^{(t)} - \alpha H^{-1} \nabla_\theta l(\theta^{(t)})$$

H = hessian of loss function

Newton - Ralphson's Method

Similar to G.D. but except(H^{-1})

H^{-1} accounts for curvature.

G.D → steepest descent but curvature can be unusual.

Much faster in no. of steps required

However

$O(d) \rightarrow$ G.D.

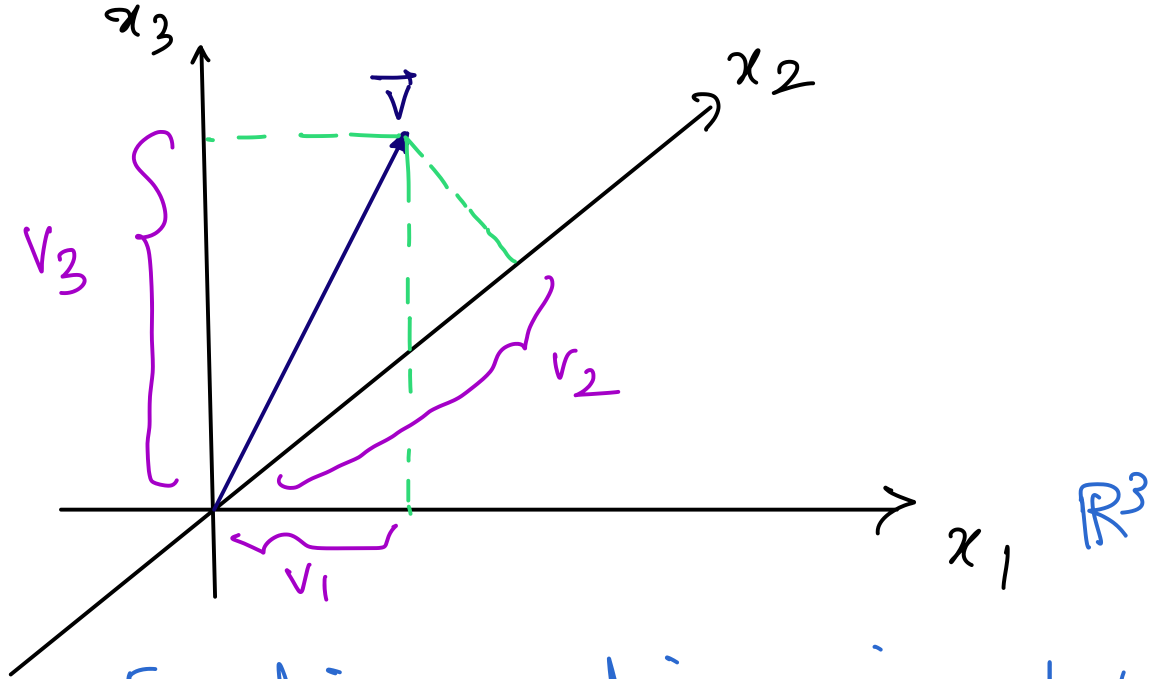
$O(d \times \underset{\substack{\uparrow \\ \text{Hessian}}}{d} \times \underset{\substack{\uparrow \\ \text{Inverse of Hessian}}}{d}) = O(d^3) \Rightarrow$ Newton Method

Newton's Method takes you to nearest stationary point (maxima/minima)

Fisher Scoring: When optimization using Newton's Method.

Functional Analysis

Study of functions → L.A. in ∞ dimensions



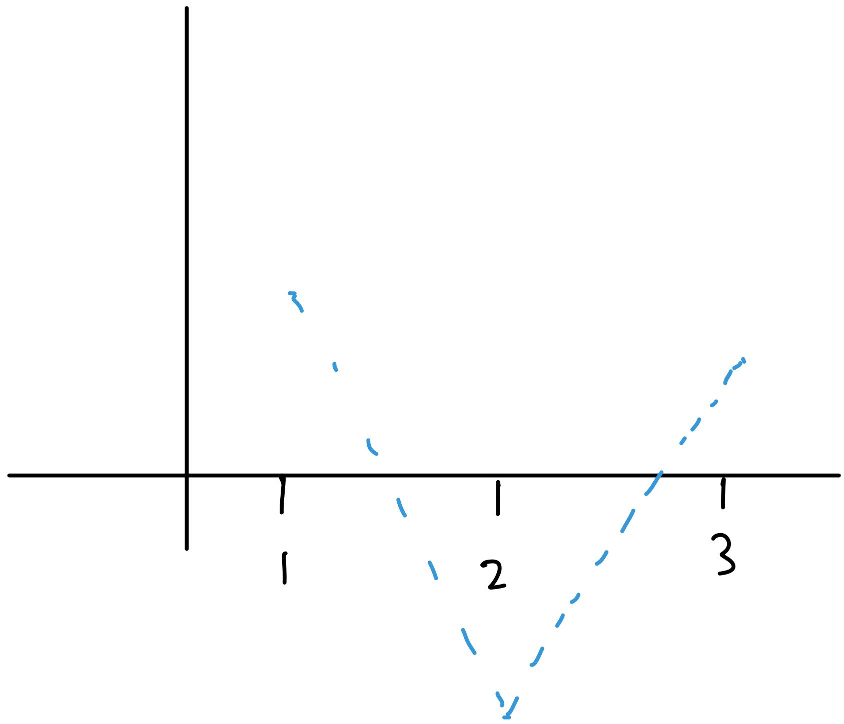
Function → ∞ dimension vector

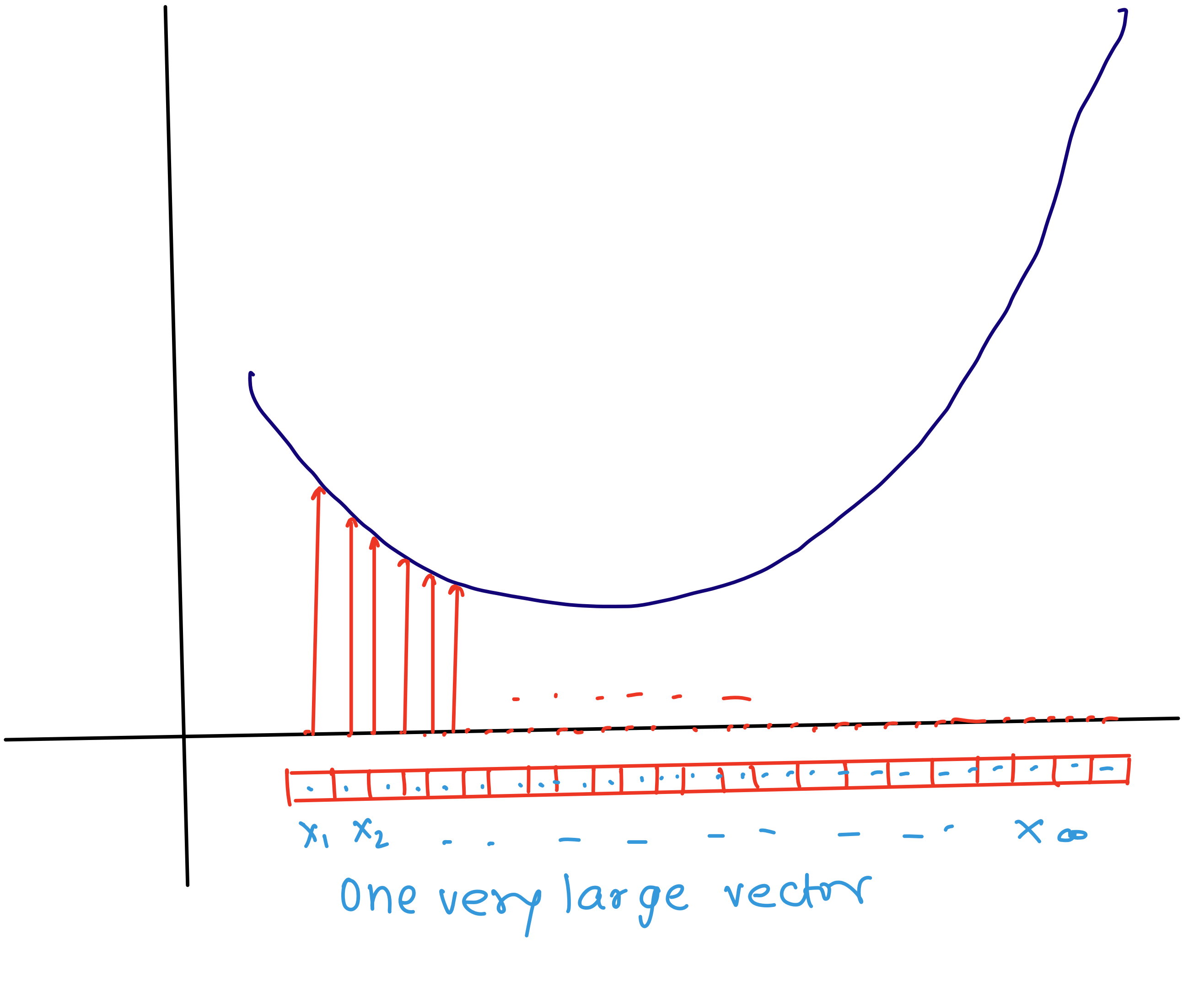
v → function

$$V : \{1,2,3\} \to \mathbb{R}$$

index of space is domain

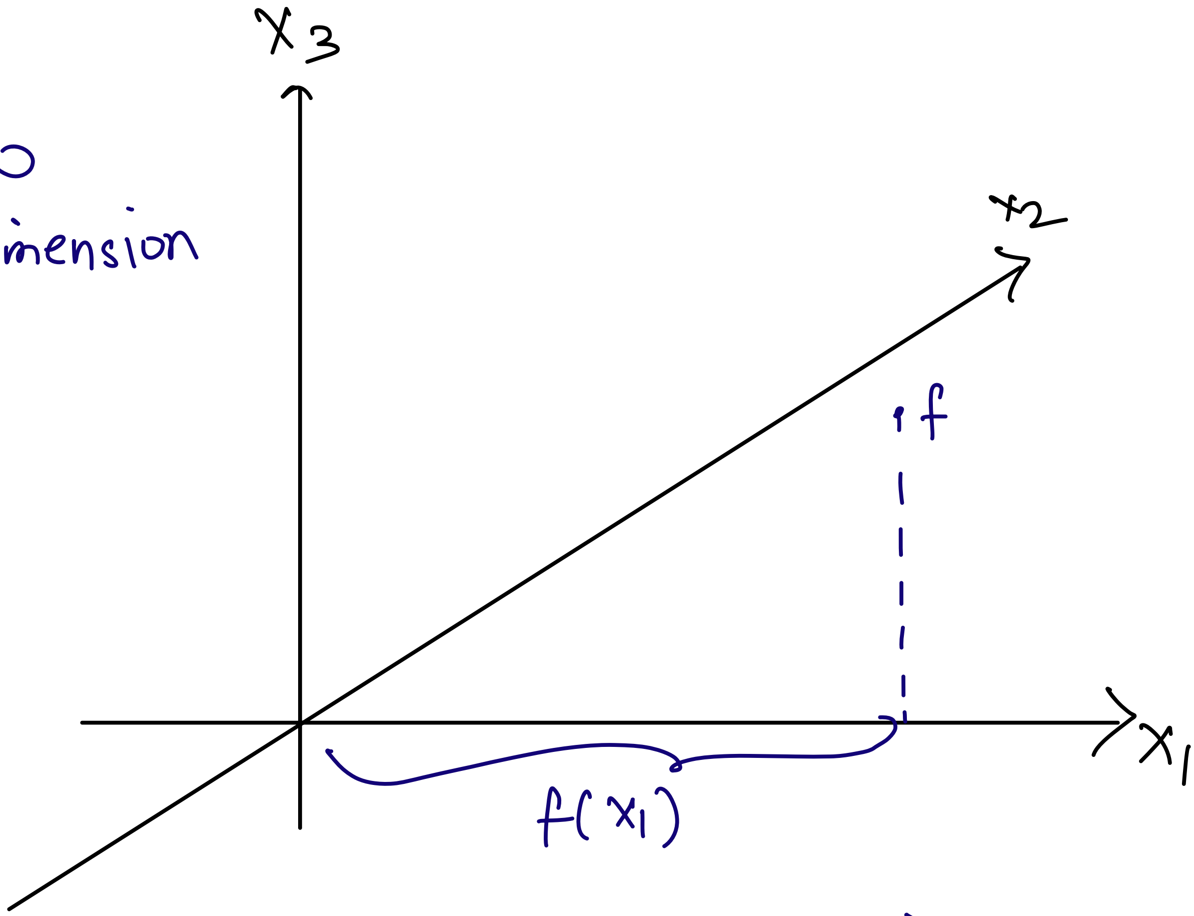
$$V(1) = v_1$$
$$V(2) = v_2$$
$$V(3) = v_3$$


1
2
3


x_1 x_2 . . . — — — — — — . x_∞
One very large vector

$$f = \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_\infty) \end{bmatrix}$$

∞ dimension



Value of function at a given value is projection value

Finite

* Vector $\vec{v}$
* Index $\to \{1, 2, 3\}$
  axes
* Components
* Explicit Rep
  $v = [v_1 \dots v_d]$
* Dot product
  $\langle u, v \rangle = \sum_i u_i v_i$

Matrix A A_{ij}

$y = Ax$

Eigen vector

$Ax = \lambda x$

Infinite

function $f(t)$

Domain $\mathbb{R}$

Values

$f(t)$ = Symbolic Representation

Inner Product

$\langle f, g \rangle = \int f(t) g(t) dt$

$= \underset{X \sim f(x)}{E} [g(x)]$

$\underset{x \sim p}{E} [g(x)]$

$= \langle p, g \rangle$

$K(s, t)$

$f' = D[f]$

$f' = D[f] = \lambda f$

$D[e^{kt}] = k e^{kt}$

Exponent is eigen function

$\vec{u} = A\vec{v}$

$u_i = \sum_j A_{ij} V_j$

$g(s) = \int k(s,t) f(t) dt$

When $k(s,t) = e^{-st}$

$k(s,t) = e^{-ist}$