Principal Component Analysis

$$X = (\underbrace{N}_{\text{examples}} \times \underbrace{D}_{\text{features.}})$$

Each $x_i \in \mathbb{R}^D$

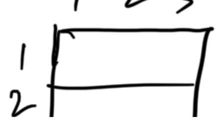
$$X \longrightarrow Z_{N \times M}$$

$$M \ll D$$

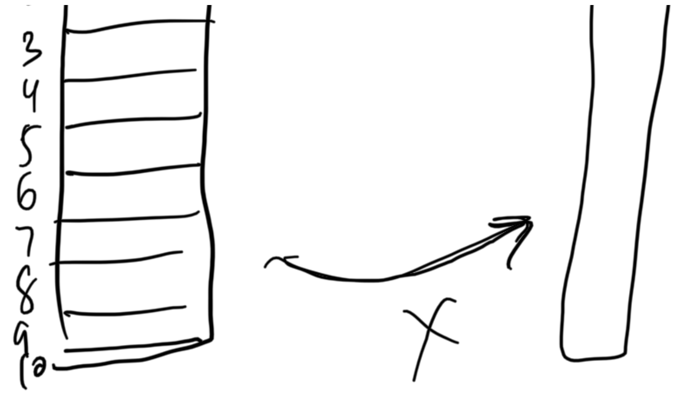
Why DR?

1. Reduces data size
   - → Storage
   - → Compute $O(f(D))$
   - → Transfer
2. Visualization
3. Improves data

How? 1 2 3 $D=1$

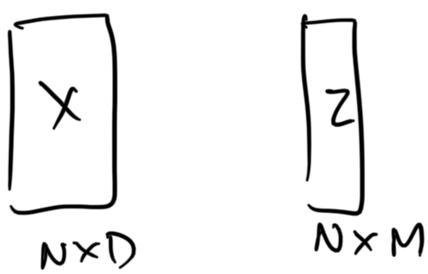


$M=1$?



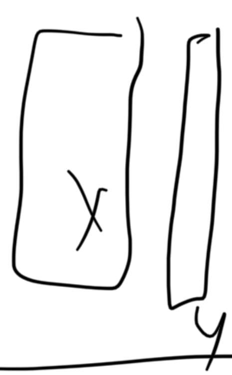
DR $\Rightarrow$ Loss in information

<u>Property to be preserved</u>

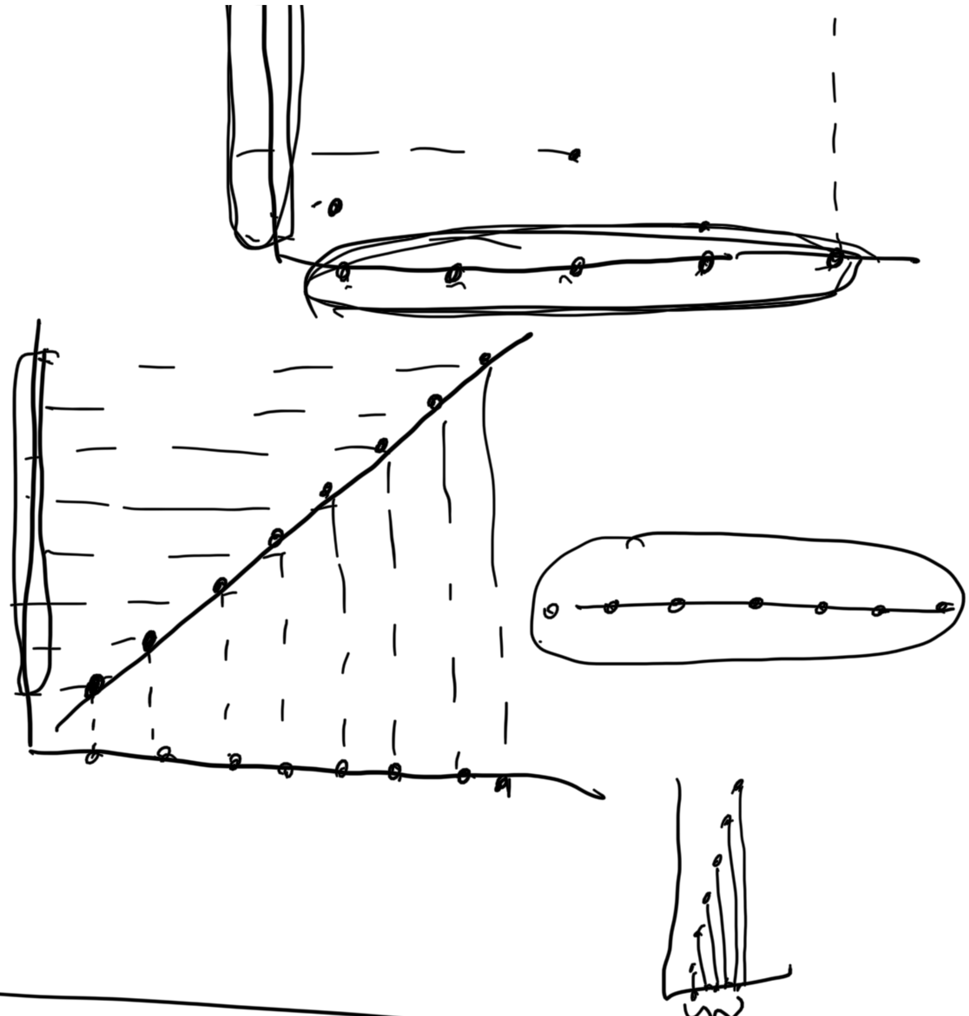


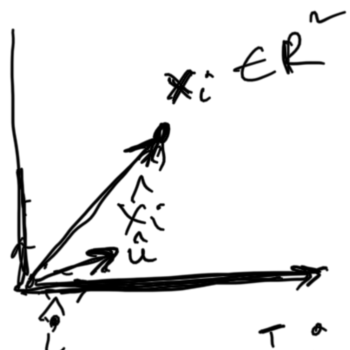
$M << D$

$$\|x_i - x_j\|_2 \approx \|z_i - z_j\|_2$$



<u>reduce</u> $\quad \underline{\mathbb{R}^D \longrightarrow \mathbb{R}^M}$





$\vec{x}_i \cdot \hat{u}$

$\hat{u} \rightarrow$ unit vecter

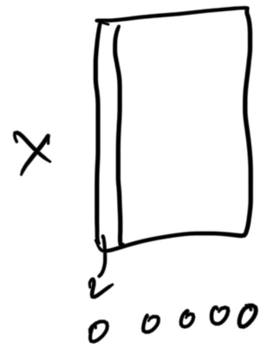
$\|\hat{u}\|_2 = 1$

$x_i^T \hat{i}$

$\vec{x}_i \cdot \hat{i}$

$\hat{u}$

We will assume that our data is mean-centered



$$\hat{u} \quad \frac{1}{N}\sum_{i=1}^{N}\left(x_i^T \hat{u}\right)^2$$

→ variance of the new data along $\hat{u}$

$$\text{Variance along } \hat{u} = \hat{u}^T\boxed{\left(\frac{1}{N}\sum x_i x_i^T\right)}\hat{u}$$

Sample covariance matrix of X $\quad \frac{1}{N}\sum (x_i-\mu)(x_i-\mu)^T$

$\mu = 0$

$$S = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T$$

$$\boxed{\begin{aligned} &\max_{\hat{u}} \ \hat{u}^T S \hat{u} \\ &\text{s.t. } \hat{u}^T u = 1 \end{aligned}}$$

$$\boxed{\begin{aligned} &\|u\|_2 = 1 \\ &(u^T u)^2 = 1 \end{aligned}}$$

$$\max_{\hat{u}, \lambda} \quad \hat{u}^T S \hat{u} - \lambda(\hat{u}^T u - 1)$$

$\frac{d}{d\hat{u}}$

$$2S\hat{u} - 2\lambda \hat{u} = 0$$

$$\frac{d}{du} u^T S u = 2Su$$

$$\boxed{S\hat{u} = \lambda \hat{u}}$$

$\frac{d}{d\lambda}$

$$\boxed{\hat{u}^T \hat{u} = 1}$$

Eigen-vector analysis

$\hat{u}$ will be the eigen vector of S.

$$\hat{U}_{max} : \hat{u}^T S \hat{u}$$

$$= \cancel{\hat{u}^T S \lambda u}$$

$$= \hat{u}^T \lambda \hat{u} = \lambda \hat{u}^T u$$

$$= \lambda$$

Variance along the eigen vector $\hat{u}$ is λ.

S is a $D \times D$ matrix.

It has D eigen vectors

and D eigen values.

$$\begin{bmatrix} \underline{\underline{\hat{u}_1}} & \underline{\underline{\hat{u}_2}} & \cdots & \hat{u}_D \\ \underline{\underline{\lambda_1}} & \underline{\underline{\lambda_2}} & \cdots & \lambda_D \end{bmatrix}$$

$$\boxed{\begin{array}{l} \hat{u}_i^T \hat{u}_i = 1 \\ \hat{u}_i^T u_j = 0 \end{array}}$$

The eigen vector corresponding to the largest eigen-value gives the direction of maximal variance.

1st Principal Components

$$\boxed{\begin{array}{l} \hat{u} = D \times 1 \\ \underset{N \times D}{X} \hat{u} = N \times 1 \end{array}}$$

λ_1 variance

λ_2 variance

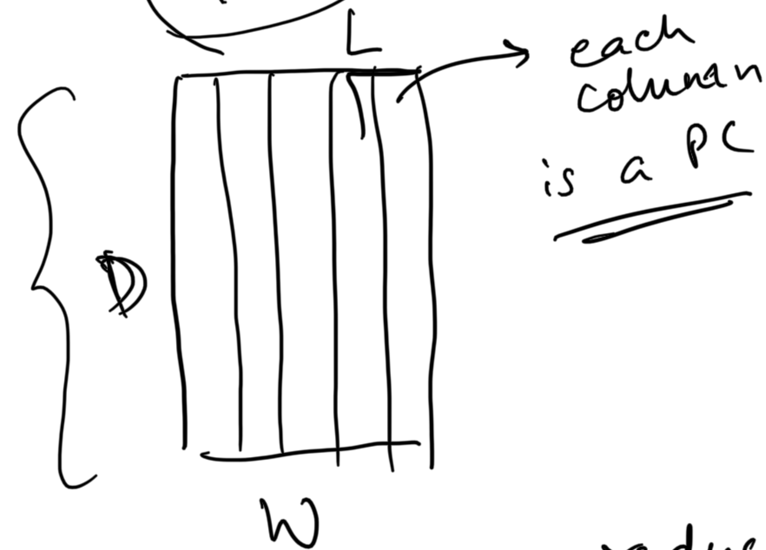
$\vdots$

λ_D variance

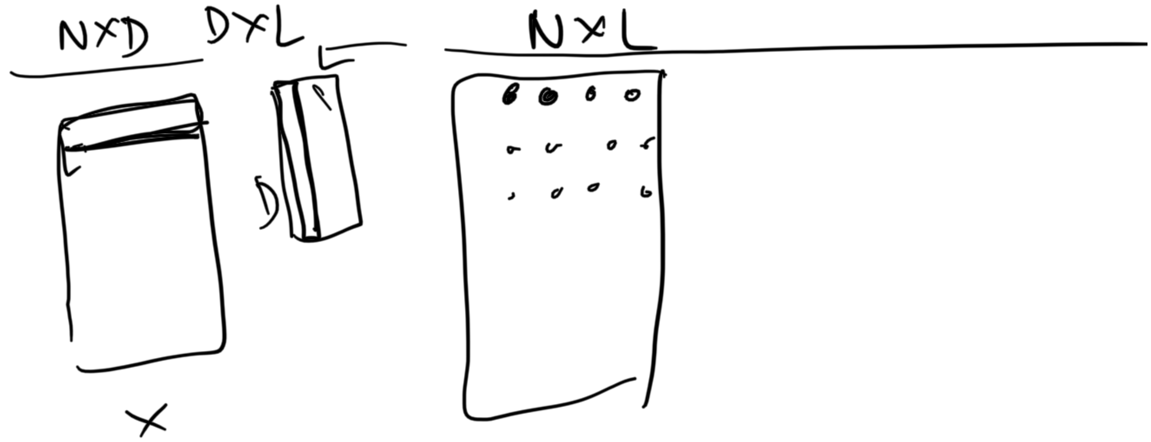
$$\sum_{i=1}^{D} \lambda_i = \text{Total variance}$$

first L PCs

$$\frac{\sum_{i=1}^{L} \lambda_i}{\sum_{i=1}^{D} \lambda_i} \times 100$$



$$XW = Z$$ ← reduced data set.



PCA is a linear algorithm.

$$\boxed{z_i = W^T x_i}$$

$L \times 1 \qquad L \times D \qquad D \times 1$

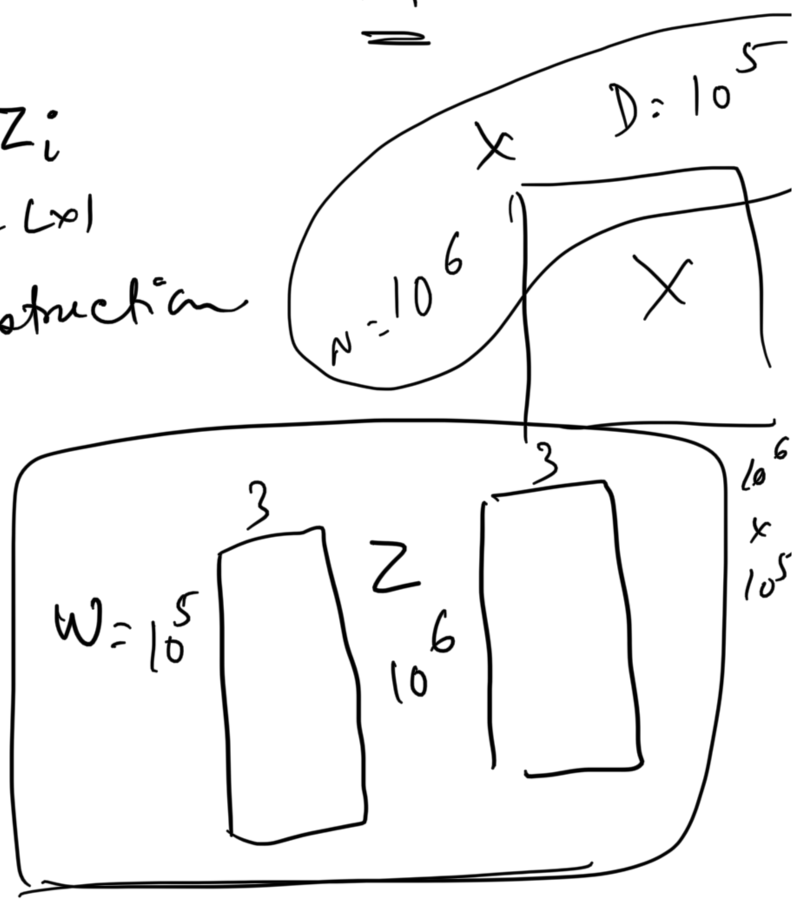
$X \qquad x_i \longrightarrow z_i$

$z_i \longrightarrow \hat{x}_i$

$\hat{x}_i = W z_i$

$D \times 1 \qquad D \times L \quad L \times 1$

reconstruction



$10^5 \times 3 + 10^6 \times 3$

$\hat{X} = Z W^T$

Reconstruction loss from PCA

$\hat{x}_i = W z_i$

$$\boxed{J = \frac{1}{N}\sum_{i=1}^{N} \| x_i - \hat{x}_i \|_2^2}$$

Non-linear PCA

$$\Phi = \{ \phi_1 \; \phi_2 \; \phi_3 \; \ldots \}$$

$$\phi_1(x_i) \to \mathbb{R}$$

$$\phi_2(x_i) \to \mathbb{R}$$

$$\phi_3(x_i) \to \mathbb{R}$$

$$\vdots$$

$$\boxed{\underset{D\times 1}{\Phi(x_i)} \longrightarrow \mathbb{R}^M}$$

$$k(x_i, x_j) \equiv \Phi(x_i)^T \Phi(x_j)$$

$$X \quad \begin{matrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{matrix} \longrightarrow \boxed{\begin{matrix} \phi(x_1) \\ \phi(x_2) \\ \vdots \\ \phi(x_N) \end{matrix}}$$

$$C = \frac{1}{N} \sum_{i=1}^{N} \phi(x_i)\,\phi(x_i)^T$$

$(M \times M)$

$$C v_k = \lambda_k v_k$$

$(M \times 1)$

$$\frac{1}{N} \sum \phi(x_i)\,\phi(x_i)^T v_k = \lambda_k v_k$$

$$\sum \phi(x_i) \left(\frac{\phi(x_i)^T v_k}{N} \right) = \lambda_k v_k$$

let $a_{ik} = \dfrac{\phi(x_i)^T v_k}{N \lambda_k}$

$$v_k = \sum a_{ki}\, \phi(x_i)$$

$$\sum_{i=1}^{N} k(x_l, x_i) \sum_{j=1}^{N} a_{kj}\, k(x_i, x_j)$$

$$= N \lambda_k \sum^{N} k(x_l, x_i)$$

$i=1$

$$K^2 a_k = \lambda_k N K a_k$$

$$K a_k = \lambda_k N a_k$$

K - kernel matrix : $N \times N$

---

we Directly calculate K

How can we ensure that implicity mapped dataset is mean-centered?