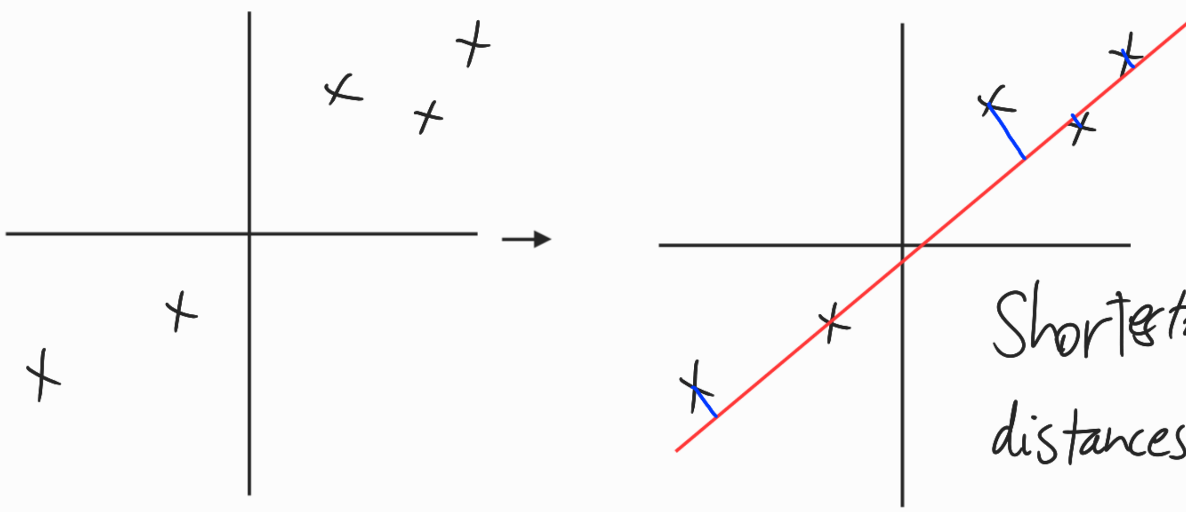
- Reduce memory
- Reduce time of algorithms

Example:



Shortest orthogonal distances

PCA finds the surface in which to project data, minimizing the overall error (having done before feature scaling).

↳ Find K vectors where to project data, minimum error.

<u>Apply PCA</u>:

1^{st}: Preprocessing on unlabelled data ⇒ feature scaling

2^{nd}: Compute "covariance matrix" ⇒ $\Sigma = \frac{1}{m} \sum_{i=1}^{n} (x^{(i)})(x^{(i)})^T \in \mathbb{R}^{n \times n}$

3^{rd}: Compute eigenvectors of Σ, $U \in \mathbb{R}^{n \times n}$

4^{th}: Take first K columns of U, to make $x \in \mathbb{R}^n \to z \in \mathbb{R}^K$

$$z = U_{reduced}^T \cdot x \quad \in \mathbb{R}^{K \times 1}$$

Reconstruction from compression:

$z = U_{red}^T \cdot x \; ; \quad x = U_{red} \cdot z \quad \rightarrow$ 100% Variance of data

Choose n principal components: Retain 99% of variance

Increase iteratively K, compute PCA and check if variance retained is $\geq 99\%$. Then stop.

$$\frac{\frac{1}{m}\sum_{i=1}^{m} \| x^{(i)} - x^{(i)}_{approx} \|^2}{\frac{1}{m}\sum_{i=1}^{m} \| x^{(i)} \|^2} \leq 0'01 \equiv 1 - \frac{\sum_{i=1}^{K} S_{ii}}{\sum_{i=1}^{n} S_{ii}} \leq 0'01$$

$$S = \begin{bmatrix} S_{11} & & & \\ & S_{22} & & \\ & & S_{33} & \\ & & & \ddots & \\ & & & & S_{nn} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Bad use of PCA

↳ Reduce overfitting: it does not, because we reduce number of features. Better use regularization.

↳ First try with original data.