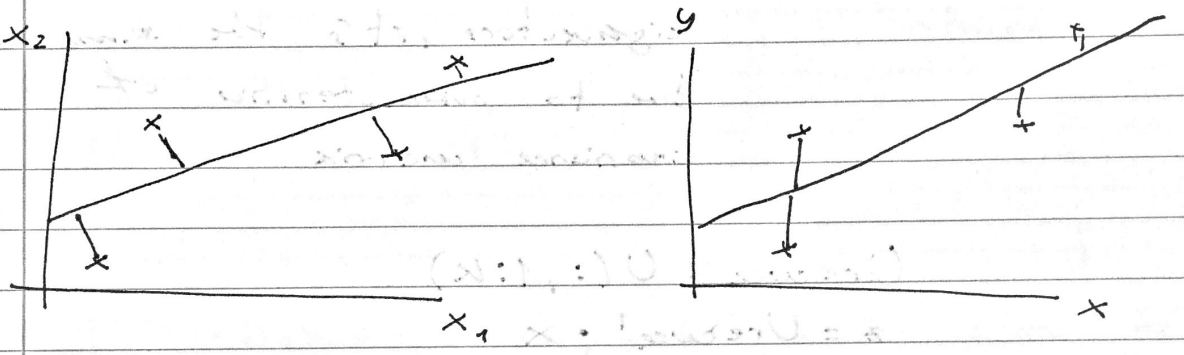
Dimensionality reduction

- PCA → principal component analysis

It's to make the common shortest projection in ~~a~~ a lower dimension to all your points

PCA	Linear regression
There's no label	There's a label



The distances to the projection are orthogonal	Parallell to the label

- Data preprocessing

Training set: $x^1, x^2 \dots x^m$

Preprocessing (feature scaling / mean normalization)

$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^i$ ← replace x_j^i with $x_j - \mu_j$

Scale features to have comparable range of values

• PCA algorithm

Reduce data from n-dimensions to k-dimensions

- Covariance matrix

$$\Sigma = \frac{1}{m} \sum_{i=1}^{n} \underbrace{x^{(i)}}_{n \times 1} \underbrace{(x^{(i)})^T}_{1 \times n} \rightarrow \bullet$$

- In Matlab → $[U, S, V] = svd(\Sigma)$

$eig(\Sigma)$

eigenvectors, it's the same here due to characteristics of covariance matrix

$$U_{reduce} = U(:, 1:k)$$

$$z = U_{reduce}^T \cdot x$$

$$\Sigma = (1/m) \cdot X^T \cdot X$$

- Reconstruction for compressed representation

$$z = \underbrace{U_{reduce}^T}_{n \times k} \cdot \underbrace{x}_{k \times 1}$$

- Choosing k (nº of principal components)

$$k \text{ to } \frac{\frac{1}{m} \sum_{i=1}^{m} \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^{m} \|x^{(i)}\|^2} \leq 0.01$$

avg squared projection error

total variation in the data

In Matlab, $[U, S, V] = svd(\Sigma)$

$$S = \begin{bmatrix} S_{11} & & 0 \\ & S_{22} & \\ 0 & & S_{nn} \end{bmatrix}$$

For given k

$$\frac{1 - \sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}} \leq 0.01 \iff \frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}} \geq 0.99$$

99% of variance retained

- Application of PCA

- Compression
  - Reduce memory/disk needed to store data
  - Speed up learning algorithm

- Visualization

- Don't use to prevent overfitting! For that you have other tools like regularization.