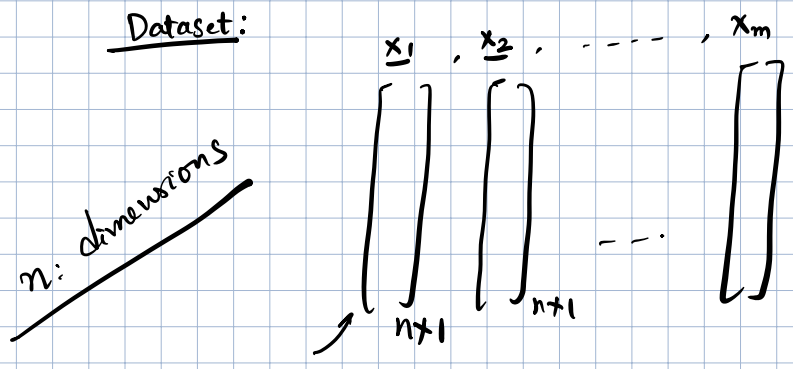
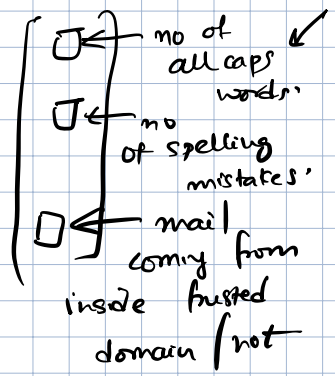
Dimensionality Reduction

"Dimension" → degrees of freedom.

Dataset:



email: spam / not spam



"Feature engineering"

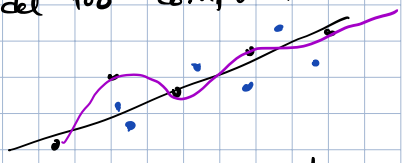
64×64

$n = 64 \times 64 \times 3$

Why reduce dimensions?

1) Reduced space requirement. Reduce the training complexity, complexity for classification.

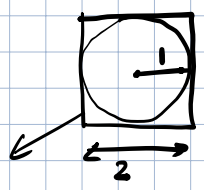
2) Avoiding overfitting → model has too many parameters, model too complicated.



3) Lower dimensions may have better interpretation / visualization.

4) High dimensional geometry is non-intuitive.

2-d



$|x_1| \le 1, \ |x_2| \le 1$

What fraction of space inside the box is taken up by the circle

$$= \frac{\text{area of circle}}{\text{area of box}} = \frac{\pi}{4}$$

3-d

$$\frac{\text{volume of sphere}}{\text{volume of box}} = \frac{4\pi}{3 \times 8} = \frac{\pi}{6}$$

wikipedia volume of n-dimensional ball.

n-d

$$\frac{\text{volume of hypersphere}}{\text{volume of hypercube}} \to \frac{\pi^{n/2}}{\left(\frac{n}{2}\right)! \, 2^n} \to 0$$

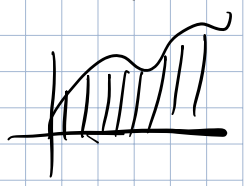
hypersphere → set of all points with distance 1 from origin

hypercube → set of all points whose co-ordinates are between −1 and 1

$$\frac{1}{\left(\frac{n}{2}\right)!} \left(\frac{\sqrt{3.14}}{2}\right)^n$$

$$\pi^{n/2} = \left(\pi^{1/2}\right)^n$$

Integration:



$\sqrt{3.14} \approx 1.8$

$\frac{1.8}{2} \approx 0.9$

$(0.9)^n$

Most of the volume in the hypercube is present on the corners, not the center.

In 2-D : How many vectors can I pick such that they are all to each other

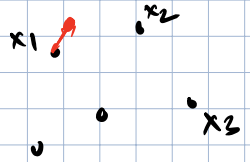
perpendicular → 90°

: How many vectors can I pick such that the angles between any two vectors is between 89° and 91°.

→ In n-d, there are exponentially many vectors that can be picked while satisfying this! $\varepsilon^{(\text{const})^n}$

$x_1, x_2, \dots, x_m$

$\|x_1\|_2 = \|x_2\|_2 = \dots = \|x_m\|_2 = 1$



K-NN

given a new data point find the point in the dataset which is closest to the new data point

$\|x - x_1\|_2^2 = 1 + 1 - 2\,x^T x_1$

$= 2 - 2\cos(\text{angle between } x \text{ and } x_1)$

2°

$1.5 - 2.5^\circ$

On high dim spaces, there may be too many (exponential) points at roughly the same distance -

5) Understand the dependencies in the <u>engineered features</u>.

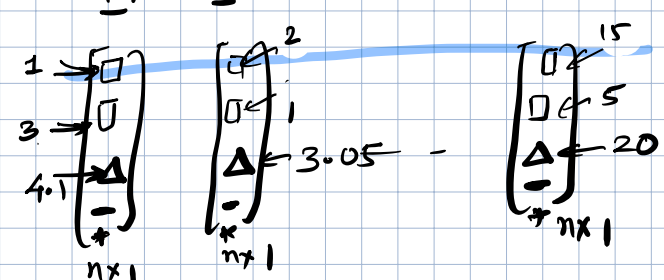
<u>Unsupervised Learning</u> : PCA principal component analysis ↓ dimensionality reduction technique.

x_1 x_2 - - - x_m

m-vectors mean μ, std dev

4-7

40-100



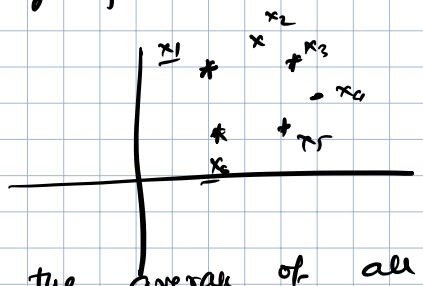
$\Delta \approx \square + \square$

$2\Delta + 3\square = 20$ —

+30 *

Linear dependence among features.

$n = 2$



Feature scaling

→ ensure that the average of all features is zero and variance is 1

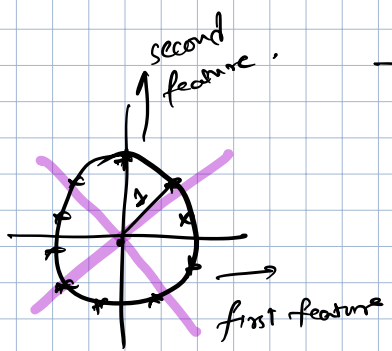
height → $\frac{\text{height} - \text{mean } h}{\text{std } h}$ → normalised heights

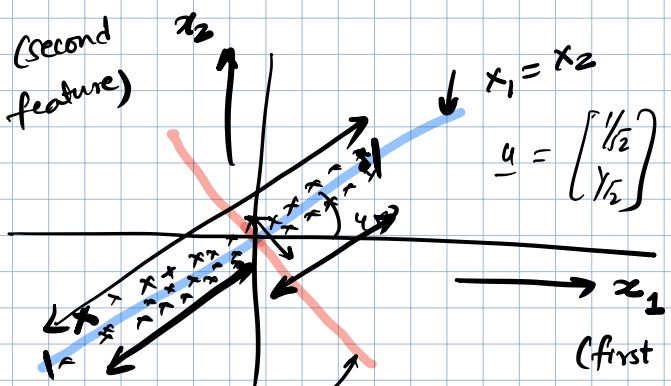
z-scores

set of normalised heights: mean = 1, std = $\sqrt{2}$

$\begin{bmatrix}1\end{bmatrix}$ $\begin{bmatrix}0\end{bmatrix}$ $\begin{bmatrix}2\end{bmatrix}$ $\begin{bmatrix}1\end{bmatrix}$

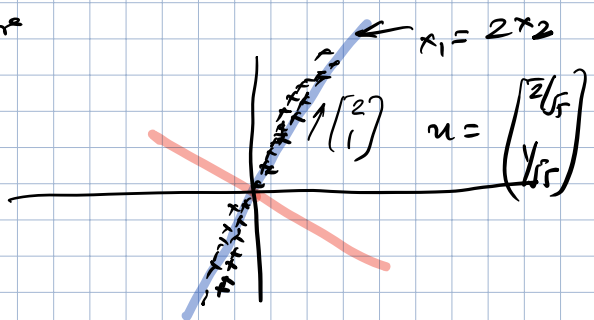
$n = 2$

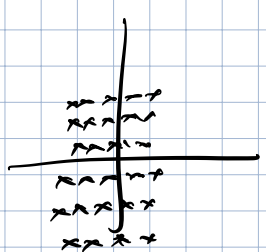




$x_1 \approx x_2$

$u = \begin{bmatrix}1/\sqrt{2} \\ 1/\sqrt{2}\end{bmatrix}$





Suppose I want to reduce dimensions from n to 1.

$\underline{x}_1 \quad \underline{x}_2 \quad \underline{x}_3 \quad - - - \quad \underline{x}_m$

$\begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \end{pmatrix}_{n\times 1} \quad \begin{pmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2n} \end{pmatrix}_{n\times 1} \quad - - - -$

give a single number for each data point.

the dimensionality-reduced dataset is given by

$\underline{x}_1$: $u_1 x_{11} + u_2 x_{12} + \cdots + u_n x_{1n} = \underline{u}^T \underline{x}_1$

$\underline{x}_2$: $u_1 x_{21} + u_2 x_{22} + \cdots + u_n x_{2n} = \underline{u}^T \underline{x}_2$ $\quad - - -$

$\underline{u}^T \underline{x}_3 \quad - - -$

Pick $\underline{u}$ such that

$\rightarrow \{ \underline{u}^T \underline{x}_1, \underline{u}^T \underline{x}_2, - - -, \underline{u}^T \underline{x}_m \}$

$\hookrightarrow$ have the highest variance possible

$$\frac{1}{m}\left(\underline{u}^T \underline{x}_1 + \underline{u}^T \underline{x}_2 + \cdots + \underline{u}^T \underline{x}_m\right)$$

$$= \underline{u}^T \left(\frac{\underline{x}_1 + \underline{x}_2 + \cdots + \underline{x}_m}{m}\right)$$

$$= 0$$

$$\text{Variance}(u) = \frac{1}{m}\left[\underbrace{(\underline{u}^T \underline{x}_1)^2}_{(\underline{u}^T \underline{x}_1)(\underline{x}_1^T \underline{u})} + (\underline{u}^T \underline{x}_2)^2 + \cdots + (\underline{u}^T \underline{x}_m)^2\right]$$

$\underline{x}_1 : n \times 1$

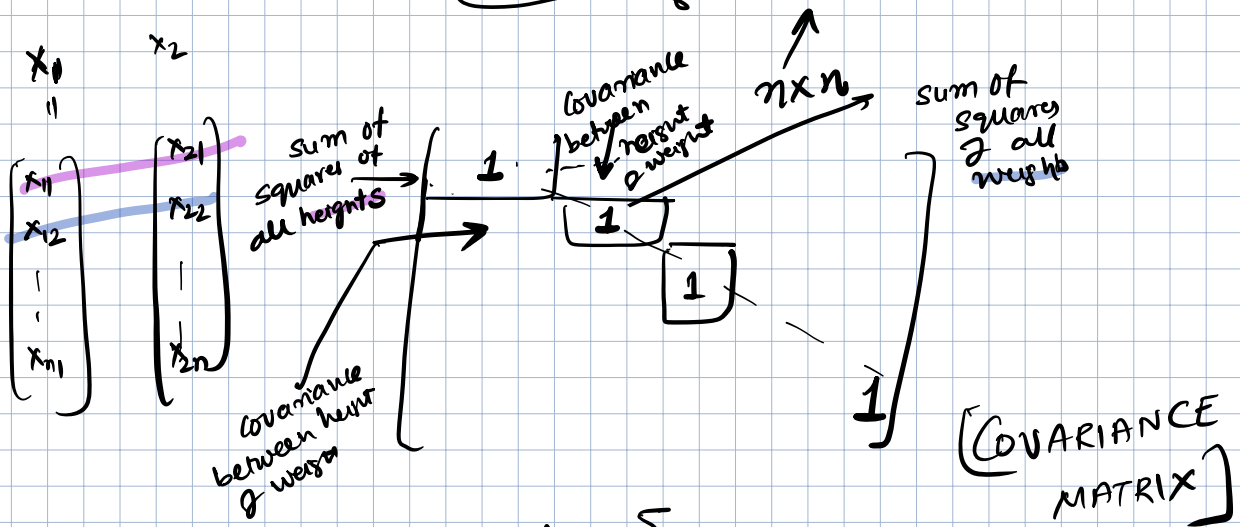
$$\frac{1}{m}\left[\underline{u}^T (\underline{x}_1 \underline{x}_1^T) \underline{u} + \underline{u}^T \underline{x}_2 \underline{x}_2^T \underline{u} + \cdots + \underline{u}^T \underline{x}_m \underline{x}_m^T \underline{u}\right]$$

$n \times n$ matrix

$$= u^T \left[\frac{x_1 x_1^T + x_2 x_2^T + \cdots + x_m x_m^T}{m} \right] u$$

$1 \times n$ $\quad$ $n \times 1$

Σ



[COVARIANCE MATRIX]

On the $n \times n$ matrix Σ

1) All diagonal entries are 1

2) The off diagonal entries are covariances between different features.

How to pick a unit vector $\underline{u}$ such that

$\underline{u}^T \Sigma \underline{u}$ is the largest posible ?

$$\underline{y} = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right]$$

6 pm

what if Σ were a diagonal matrix?

$n = 2$

Suppose, $\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 5 \end{bmatrix}$ $\quad \underline{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$

$$\underline{u}^T \Sigma \underline{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$= \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} 2u_1 \\ 5u_2 \end{bmatrix} = 2u_1^2 + 5u_2^2$$

$u_1^2 + u_2^2 = 1$

we should pick $u_2 = 1, \; u_1 = 0.$

$\underline{u} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Suppose Σ is a diagonal matrix with diagonal entries $d_1, d_2, d_3, \ldots$

then $\underline{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \end{bmatrix}$ $\quad \underline{u}^T \Sigma \underline{u} = d_1 u_1^2 + d_2 u_2^2 + d_3 u_3^2 + \cdots$

$u_1^2 + u_2^2 + u_3^2 + \cdots = 1$

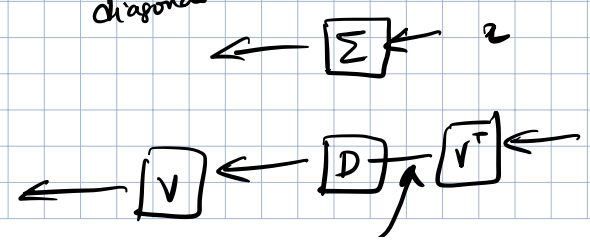
$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \qquad \underline{u} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

what if the matrix Σ is not diagonal?

$\Sigma = V D V^T$ (symmetric; V orthogonal, D diagonal)

$\underline{u}^T \Sigma \underline{u}$

$\underline{u}^T V D V^T \underline{u}$



$\underline{w}^T \underline{w}$

$\underline{w} = V^T \underline{u}$

$\underline{w}$: w-ordinates after rotation by V.

$w^T = u^T V$

$\sqrt{w^T w} =$ $\underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$

$w_1^2 + w_2^2 + \cdots + w_n^2$

$= u_1^2 + u_2^2 + \cdots + u_n^2$

$= 1$

$\underline{w} = V^T \underline{u} \rightarrow \boxed{\underline{u} = V \underline{w}}$

$\underline{w}$ is also a unit vector.

$\text{Variance}(\underline{u}) = \underline{u}^T \Sigma \underline{u}$

$= \underline{w}^T D \underline{w}$

For simplicity, assume that the Diagonal entries in D are in sorted order $\begin{bmatrix} 10 & & & \\ & 5 & & \\ & & 3 & \\ & & & 2 \end{bmatrix}$

$\underline{w} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$\underline{u} = \underbrace{\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}}_{V} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

eigenvector for largest eval

eigen vector for 2nd largest eval

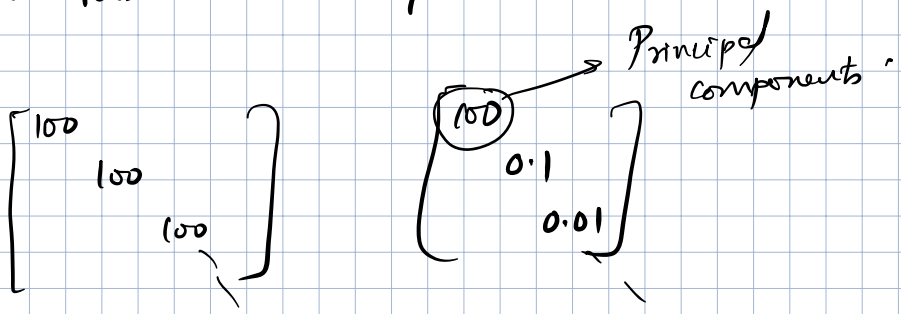
$\underbrace{\begin{bmatrix} | & | & | \\ \cdots & \cdots & \cdots \\ | & | & | \end{bmatrix}}_{V}$

$= \begin{bmatrix} \cos\theta \\ -\sin\theta \end{bmatrix}$ first column of the matrix V.

we should pick $\underline{u}$ to be the first column of the (sorted) eigen vector matrix V.

$\underline{u}$ = eigenvector corresponding to largest eigenvalue of Σ.

How good will this dimensionality reduction be?



Principal components are the directions along which the data is "most expressive" or "highly informative" or "most varying".

→ obtained by taking eigenvectors corresponding to the top- eigenvalues of the covariance matrix.

Algorithm (PCA: Principal component analysis)

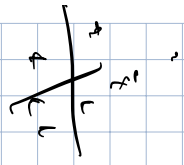
Input: Total m vectors, $\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_m$ with each vector having n-coordinates.

K : target dimension $(K << n)$.

1. feature scaling
2. Find the covariance matrix $\Sigma = \frac{x_1 x_1^T + x_2 x_2^T + \cdots}{m}$
3. Find the eigen vectors for the K- largest eigen values of Σ.

   $\underline{u}_1 \quad \underline{u}_2 \quad \cdots \quad \underline{u}_K$
4. the output (dimensionality reduced dataset) is given by

   x_1 $n \times 1$ $\quad x_2 \quad \cdots \quad x_m$



$$\begin{bmatrix} u_1^T x_1 \\ u_2^T x_1 \\ u_3^T x_1 \\ \vdots \\ u_k^T x_1 \end{bmatrix}_{k \times 1} \quad \begin{bmatrix} u_1^T x_2 \\ u_2^T x_2 \\ u_3^T x_2 \\ \vdots \\ u_k^T x_2 \end{bmatrix} \quad \begin{bmatrix} u_1^T x_m \\ u_2^T x_m \\ \vdots \\ u_k^T x_m \end{bmatrix}$$

Sort all e-vals

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$$

Construct

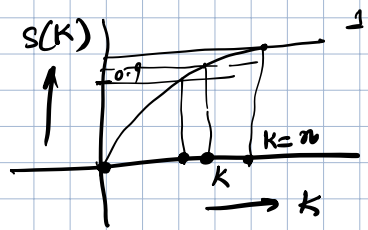
$$S(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$$

$\sum_{i=1}^{k} \lambda_i$ ← top k evals

$\sum_{i=1}^{n} \lambda_i$ ↑ all evals.

$$\begin{bmatrix} 100 & & & & \\ & 50 & & & \\ & & 10 & & \\ & & & 0.001 & \\ & & & & 0.0001 \end{bmatrix}$$

$S(k)$ to be close to 1.



$$\begin{bmatrix} 10 & & & \\ & 10 & & \\ & & 1 & \\ & & & 0 \end{bmatrix}$$

① Ethem Alpaydin

Intro to machine learning

(entire chapter on dimensionality reduction)

② Bishop, Murphy

PRML

Probabilistic perspective Machine learning.

Convex optimisation: → KKT conditions

$f(x)$

- Duality
- Gradient descent.

$\underline{x}$ $\quad \max \underbrace{\underline{y}^T \Sigma \underline{y}}_{f(\underline{y})}$

$$\sum_{i=1}^{n} (y_i - w^T x_i)^2$$

Textbook: Convex optimization — Stephen Boyd

email: stadityaa@ee.iith.ac.in