Curse of Dimensionality

x1	t
3.77	RED
14.40	GREEN
16.20	RED
9.66	GREEN
18.99	RED
15.59	GREEN
4.58	RED
19.30	GREEN
11.55	RED
4.05	GREEN
1.11	RED
11.76	GREEN
18.36	RED
13.12	GREEN
11.53	RED
14.63	GREEN
6.51	RED
7.52	GREEN
10.19	RED
17.92	GREEN

$\{x_1, x_2, \ldots, x_N\}$ $N=20$ $x \in [0,20]$

$\{t_1, t_2, \ldots, t_N\}$ $t \in \{r, g\}$

$p(t = g|x)$

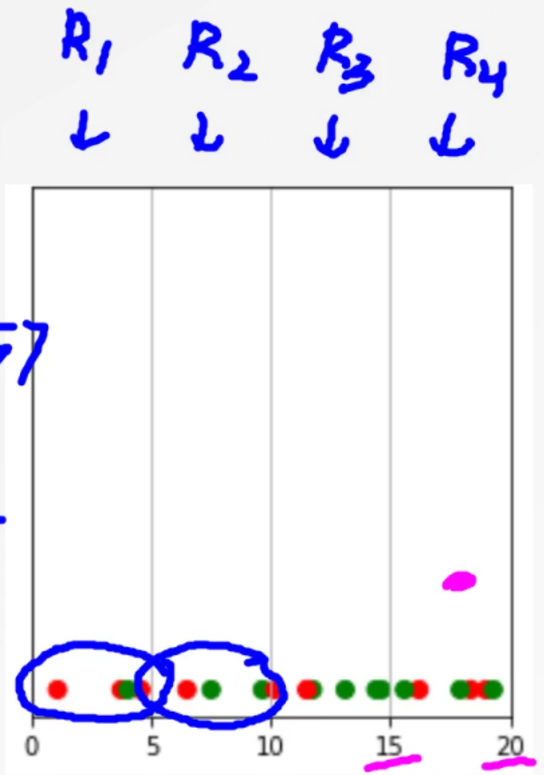
x1	t
3.77	RED
14.40	GREEN
16.20	RED
9.66	GREEN
18.99	RED
15.59	GREEN
4.58	RED
19.30	GREEN
11.55	RED
4.05	GREEN
1.11	RED
11.76	GREEN
18.36	RED
13.12	GREEN
11.53	RED
14.63	GREEN
6.51	RED
7.52	GREEN
10.19	RED
17.92	GREEN

$$p(t=g|R_1) = \frac{1}{4} = 0.25$$

$$p(t=g|R_2) = \frac{2}{3} \approx 0.67$$

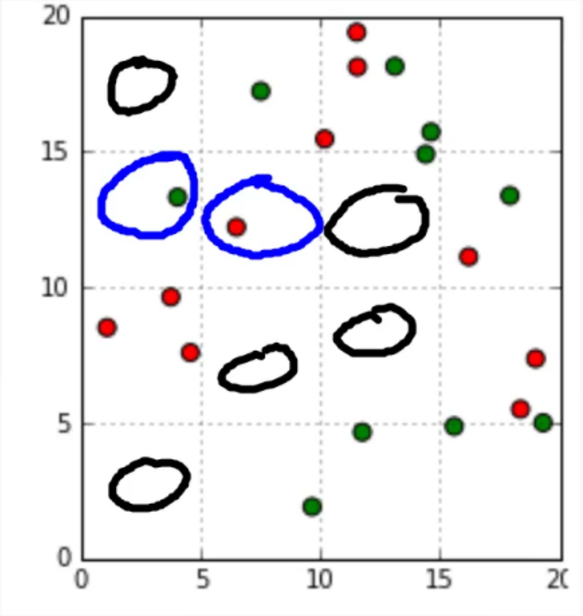
$$p(t=g|R_3) = \frac{4}{7} = 0.57$$

$$p(t=g|R_4) = \frac{3}{6} = 0.5$$



x1	x2	t
3.77	9.64	RED
14.40	14.91	GREEN
16.20	11.13	RED
9.66	1.91	GREEN
18.99	7.37	RED
15.59	4.87	GREEN
4.58	7.59	RED
19.30	5.01	GREEN
11.55	18.12	RED
4.05	13.32	GREEN
1.11	8.51	RED
11.76	4.66	GREEN
18.36	5.51	RED
13.12	18.14	GREEN
11.53	19.40	RED
14.63	15.72	GREEN
6.51	12.22	RED
7.52	17.23	GREEN
10.19	15.47	RED
17.92	13.38	GREEN

4x4 = 16



x1	x2	x3	t
3.77	9.64	15.64	RED
14.40	14.91	9.24	GREEN
16.20	11.13	2.52	RED
9.66	1.91	18.73	GREEN
18.99	7.37	6.88	RED
15.59	4.87	4.39	GREEN
4.58	7.59	12.90	RED
19.30	5.01	19.17	GREEN
11.55	18.12	16.54	RED
4.05	13.32	3.36	GREEN
1.11	8.51	15.70	RED
11.76	4.66	16.98	GREEN
18.36	5.51	9.52	RED
13.12	18.14	3.41	GREEN
11.53	19.40	12.57	RED
14.63	15.72	10.64	GREEN
6.51	12.22	11.04	RED
7.52	17.23	7.90	GREEN
10.19	15.47	16.46	RED
17.92	13.38	5.80	GREEN

$4 \times 4 \times 4 = 64$

1D

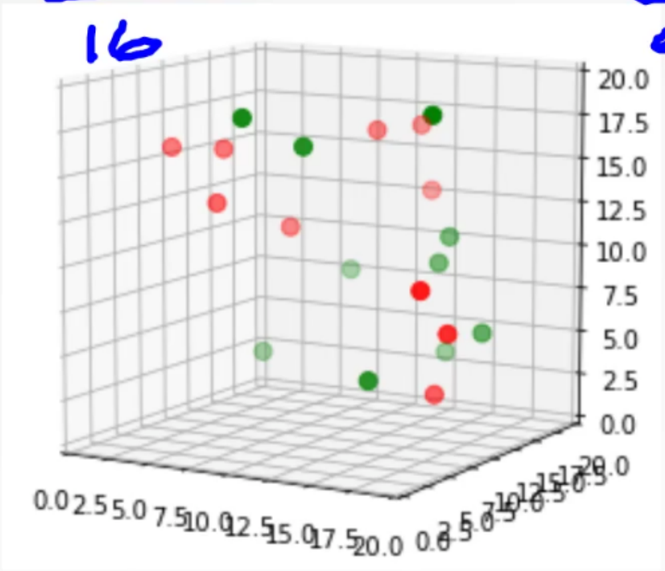
$\frac{20}{4} = 5$

2D

$\frac{20}{16} = 1.25$

3D

$\frac{20}{64} \approx 0.31$



Can we increase N?

$N^{\frac{1}{D}}$

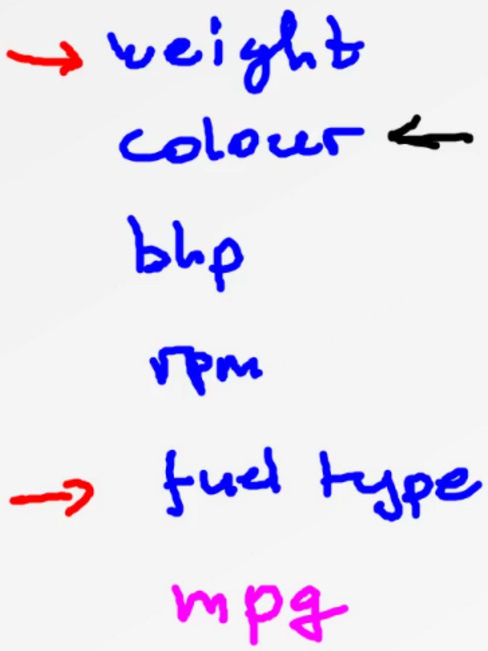
$20^{\frac{1}{1}} = x^{\frac{1}{3}}$

$x = 8'000$

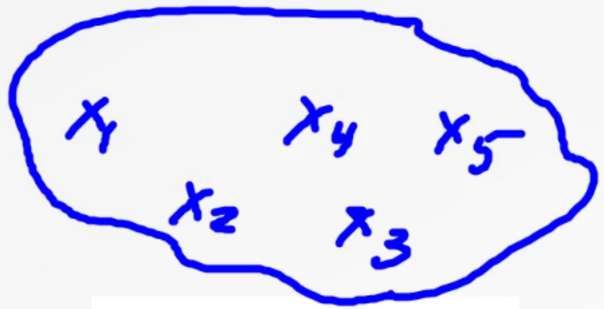
- Data becomes sparse as we add dimensions
- Distances lose meaning in high dimensions

Dimensionality Reduction

- Feature selection



- Feature reduction
  - Use all features from the dataset
  - Generate artificial features
    - smaller than the original set
    - retain as much information as possible

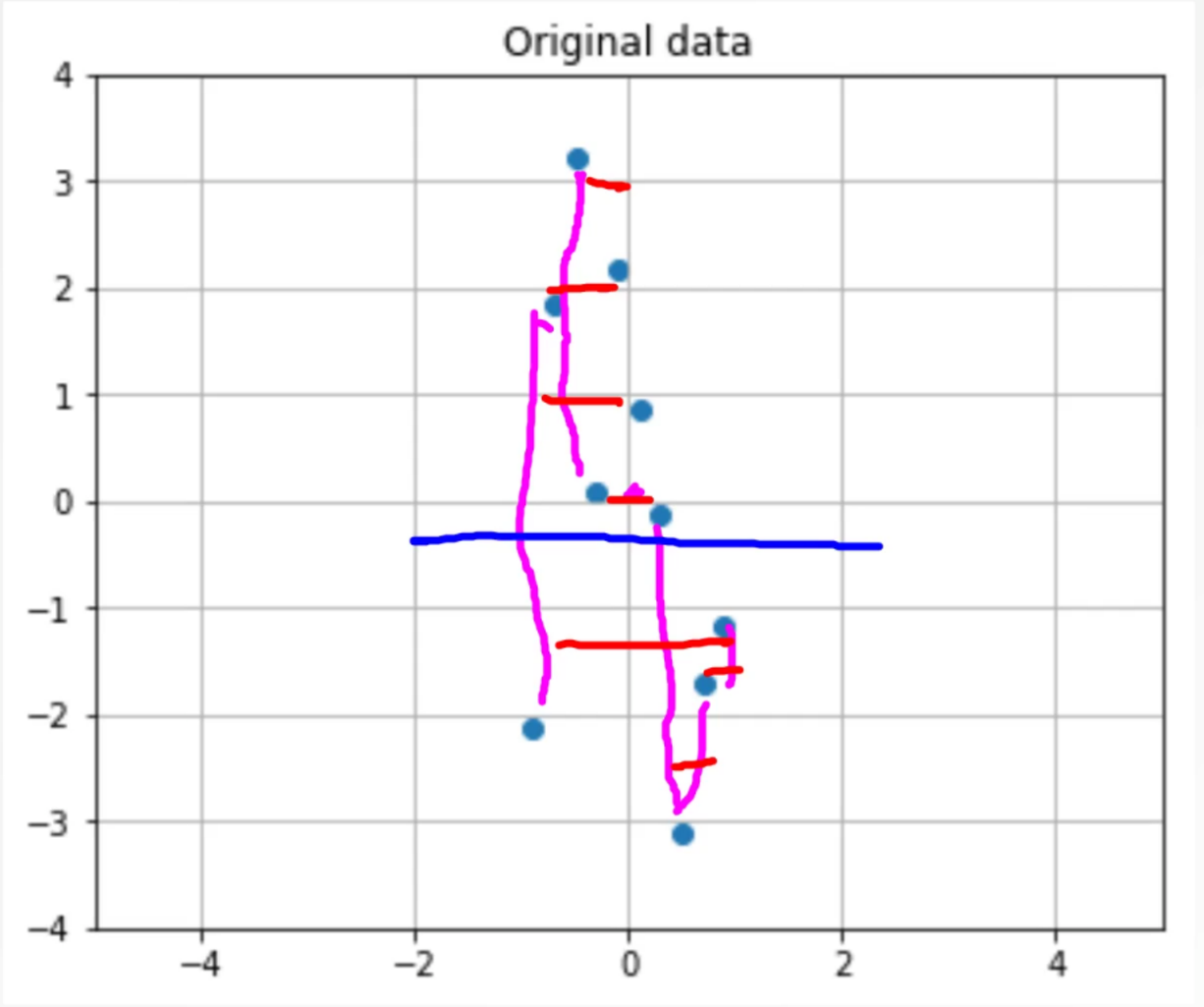


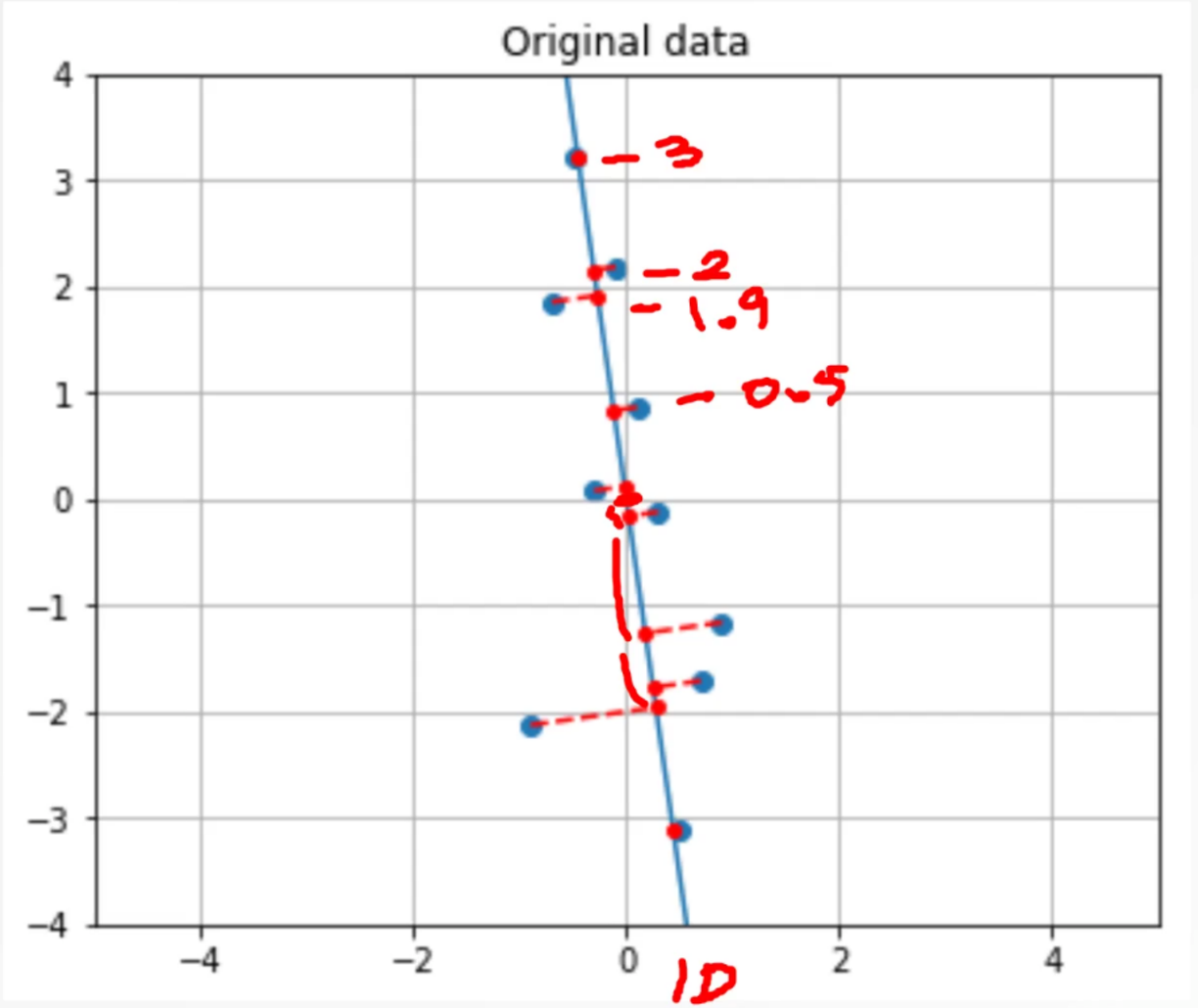
x_1 x_2

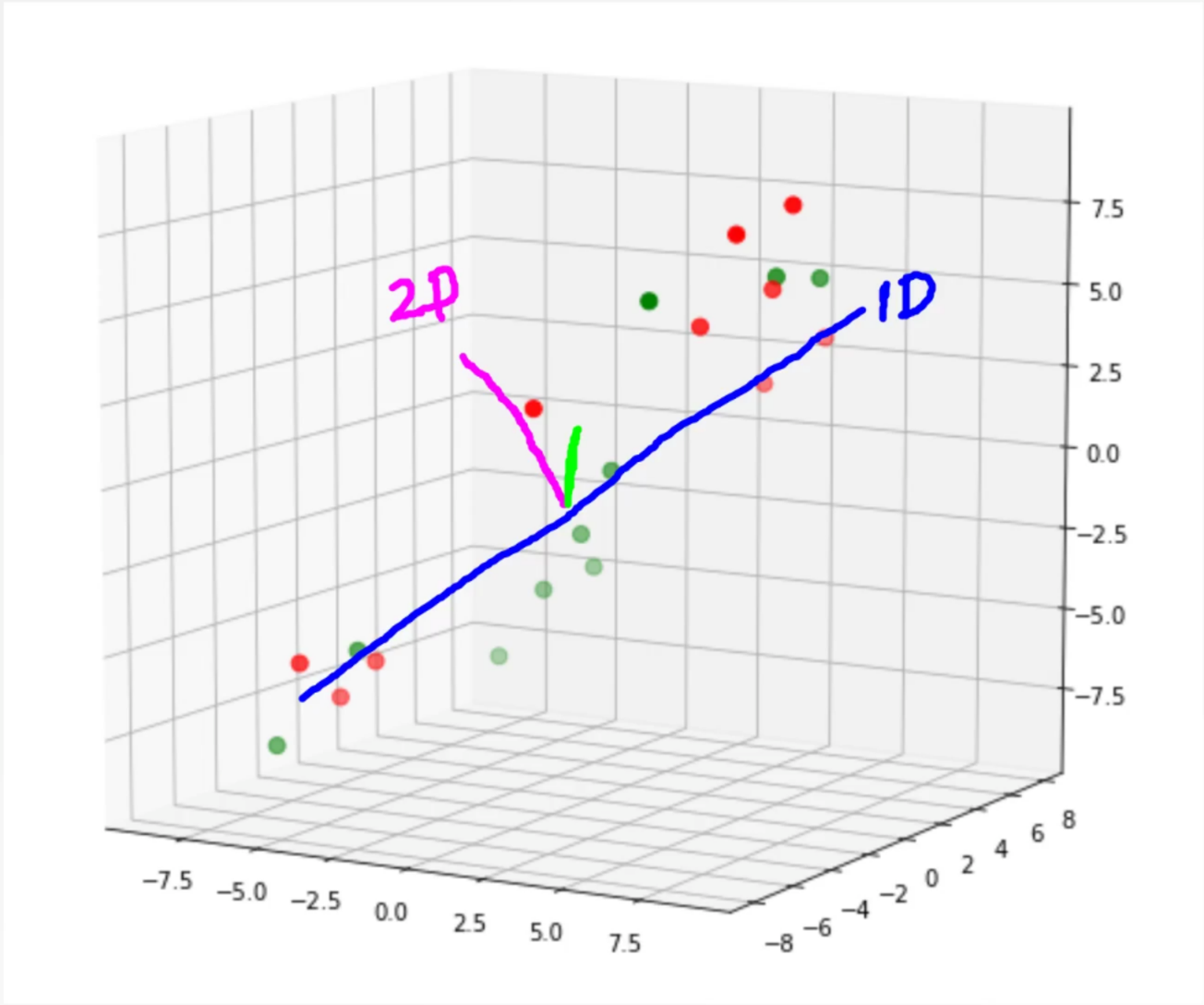
- Methods
  - Linear
    - Principal Component Analysis (PCA)
    - Linear Discriminant Analysis (LDA)
    - Canonical correlation analysis
    - Multi-dimensional scaling
  - Non-linear
    - Manifold learning (e.g. SOM, autoencoders etc.)

Principal Component Analysis

$var(x) = E[(x-\bar{x})^2]$




Original data
4
3
2
1
0
−1
−2
−3
−4
−4
−2
0
2
4
−3
−2
−1.9
−0.5
1D


2D
1D
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
−8
−6
−4
−2
0
2
4
6
8

$x - \bar{x}$

- Calculating PCA
  1. Centre the data
     - Compute covariance matrix Σ
     - Find the eigenvectors and eigenvalues of Σ
  1. The eigenvectors become the principal components
  2. The eigenvalues provide the explained variance
  3. Select new dimensions and project the data

$$cov(x_1, x_2) = \frac{\sum_{i=1}^{N} (x_{1i} - \bar{x}_1)(x_{2i} - \bar{x}_2)}{n-1}$$

$$\Sigma = \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}$$

i, j

$$\Sigma = \frac{1}{n-1} X^T X$$

$\Sigma \cdot e$

1) eigenvalues

$$\det(\Sigma - \lambda I) = 0$$

$$\lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \quad \det \begin{pmatrix} s_{11} - \lambda & s_{12} \\ s_{21} & s_{22} - \lambda \end{pmatrix} = 0$$

2) eigenvectors

$$(s_{11} - \lambda)(s_{22} - \lambda) - s_{21} s_{12} = 0$$

$$\lambda^2 \text{ ---- } =$$

$\bar{\lambda}_1$ $\bar{\lambda}_2$

$$\Sigma \underline{e} = \lambda \underline{e}$$

$\lambda_1 \rightarrow e_1$

$\lambda_2 \rightarrow e_2$

$$\lambda = \frac{tr(\Sigma) \pm \sqrt{tr^2(\Sigma) - 4 \det(\Sigma)}}{2}$$

$x - \bar{x}$

- Calculating PCA
  1. Centre the data
     - Compute covariance matrix Σ
     - Find the eigenvectors and eigenvalues of Σ
  1. The eigenvectors become the principal components
  2. The eigenvalues provide the explained variance
  3. Select new dimensions and project the data