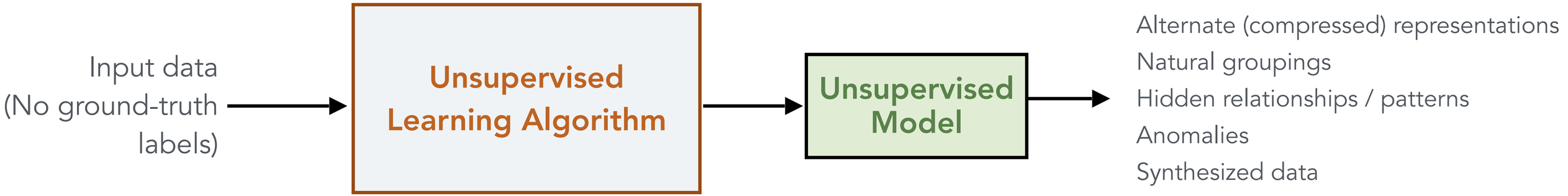
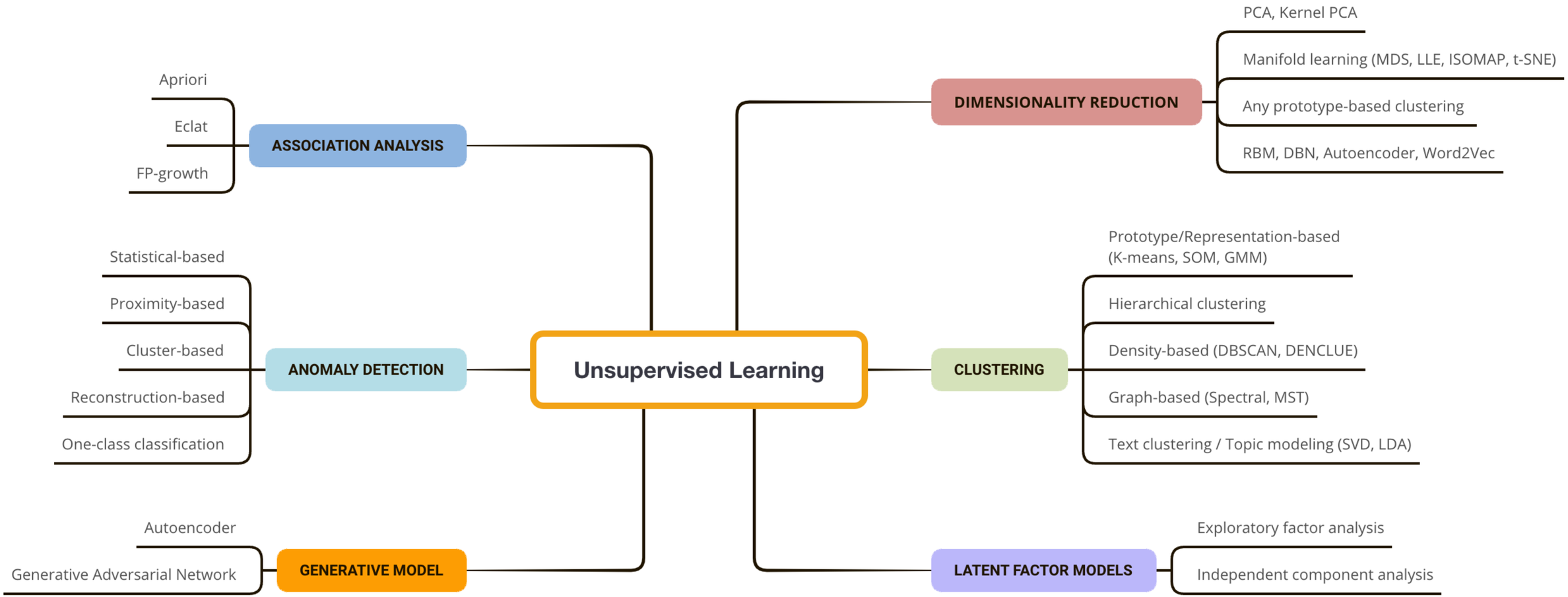
Unsupervised Learning




Unsupervised Learning
ASSOCIATION ANALYSIS
Apriori
Eclat
FP-growth
ANOMALY DETECTION
Statistical-based
Proximity-based
Cluster-based
Reconstruction-based
One-class classification
GENERATIVE MODEL
Autoencoder
Generative Adversarial Network
DIMENSIONALITY REDUCTION
PCA, Kernel PCA
Manifold learning (MDS, LLE, ISOMAP, t-SNE)
Any prototype-based clustering
RBM, DBN, Autoencoder, Word2Vec
CLUSTERING
Prototype/Representation-based (K-means, SOM, GMM)
Hierarchical clustering
Density-based (DBSCAN, DENCLUE)
Graph-based (Spectral, MST)
Text clustering / Topic modeling (SVD, LDA)
LATENT FACTOR MODELS
Exploratory factor analysis
Independent component analysis

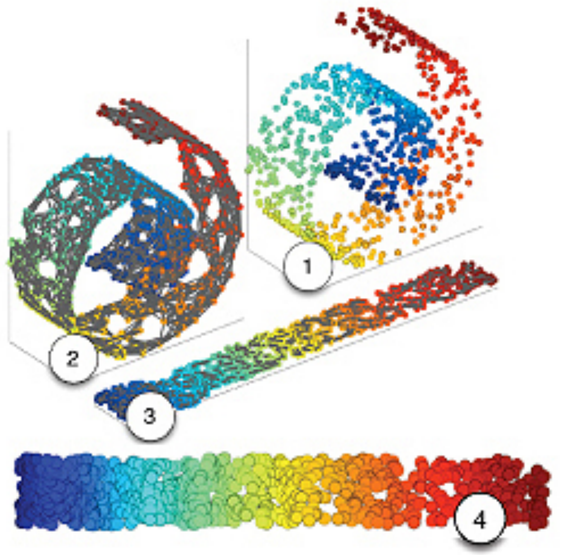
Dimensionality Reduction

Peerapon S.

Machine Learning

Topics

- ☐ Principal component analysis (PCA)
- ☐ Manifold learning



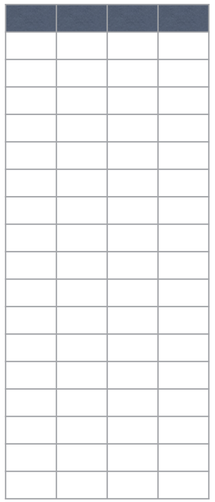
Dimensionality Reduction

Lots of variables

Row ID	Order ID	Order Date	Order Priority	Order Quantity	Sales	Discount	Ship Mode	Profit	Unit Price	Shipping Cost
1	3	10-13-10	Low	6	261.54	0.04	Regular Air	-213.25	38.94	35
49	293	10-1-12	High	49	10123.02	0.07	Delivery Truck	457.81	208.16	68.02
50	293	10-1-12	High	27	244.57	0.01	Regular Air	46.71	8.69	2.99
80	483	07-10-11	High	30	4965.7595	0.08	Regular Air	1198.97	195.99	3.99
85	515	08-28-10	Not Specified	19	394.27	0.08	Regular Air	30.94	21.78	5.94
86	515	08-28-10	Not Specified	21	146.69	0.05	Regular Air	4.43	6.64	4.95
97	613	06-17-11	High	12	93.54	0.03	Regular Air	-54.04	7.3	7.72
98	613	06-17-11	High	22	905.08	0.09	Regular Air	127.70	42.76	6.22
103	643	03-24-11	High	21	2781.82	0.07	Express Air	-695.26	138.14	35
107	678	02-26-10	Low	44	228.41	0.07	Regular Air	-226.36	4.98	8.33
127	807	11-23-10	Medium	45	196.85	0.01	Regular Air	-166.85	4.28	6.18
128	807	11-23-10	Medium	32	124.56	0.04	Regular Air	-14.33	3.95	2
134	868	06-8-12	Not Specified	32	716.84	0	Regular Air	134.72	21.78	5.94
135	868	06-8-12	Not Specified	31	1474.33	0.04	Regular Air	114.46	47.98	3.61
149	933	08-4-12	Not Specified	15	80.61	0.02	Regular Air	-4.72	5.28	2.99
160	995	05-30-11	Medium	46	1815.49	0.03	Regular Air	782.91	39.89	3.04
161	998	11-25-09	Not Specified	16	248.26	0.07	Regular Air	93.80	15.74	1.39
175	1154	02-14-12	Critical	44	4462.23	0.04	Delivery Truck	440.72	100.98	26.22
176	1154	02-14-12	Critical	11	663.784	0.25	Regular Air	-481.04	71.37	69
203	1344	04-15-12	Low	15	834.904	0.06	Regular Air	-11.68	65.99	5.26
204	1344	04-15-12	Low	18	2480.9205	0.01	Regular Air	313.58	155.99	8.99
213	1412	03-12-10	Not Specified	13	59.03	0.1	Express Air	26.92	3.69	0.5
214	1412	03-12-10	Not Specified	21	97.48	0.05	Regular Air	-5.77	4.71	0.7
229	1539	03-9-11	Low	33	511.83	0.1	Regular Air	-172.88	15.99	13.18
230	1539	03-9-11	Low	38	184.99	0.05	Regular Air	-144.55	4.89	4.93
231	1540	08-4-12	High	30	80.9	0.09	Regular Air	5.76	2.88	0.7

Few ***transformed features*** *or* ***alternate representation*** that captures most input information.

Principal component analysis
Manifold learning
Factor analysis
Autoencoder



Input data

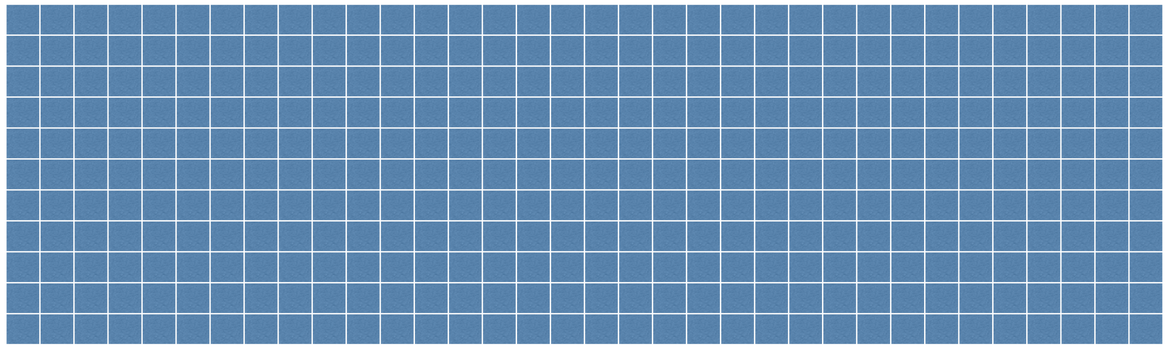
$$\boldsymbol{X}_{n\times d} = \begin{bmatrix} & \boldsymbol{X}_1 & \boldsymbol{X}_2 & \cdots & \boldsymbol{X}_d \\ \boldsymbol{x}_1 & x_{11} & x_{12} & \cdots & x_{1d} \\ \boldsymbol{x}_2 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_n & x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}$$

Notations

Row = vector of random variables

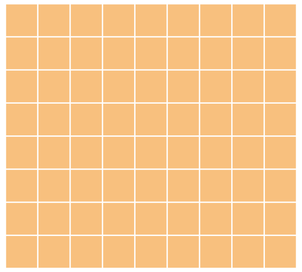
(Redundant/correlated) Original features

n rows of observations/ data points/ Users



Dimension = d

n rows of observations/ data points/ Users

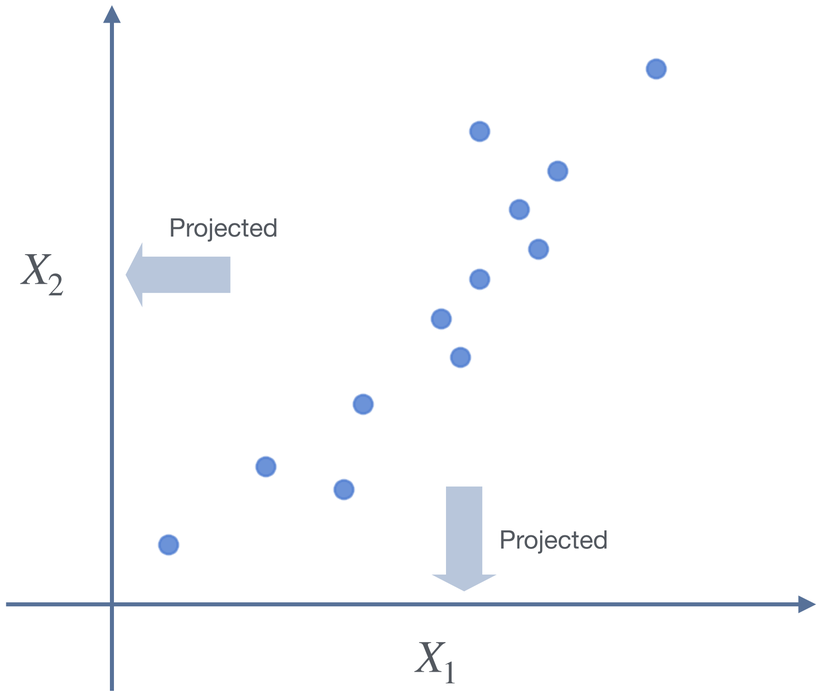


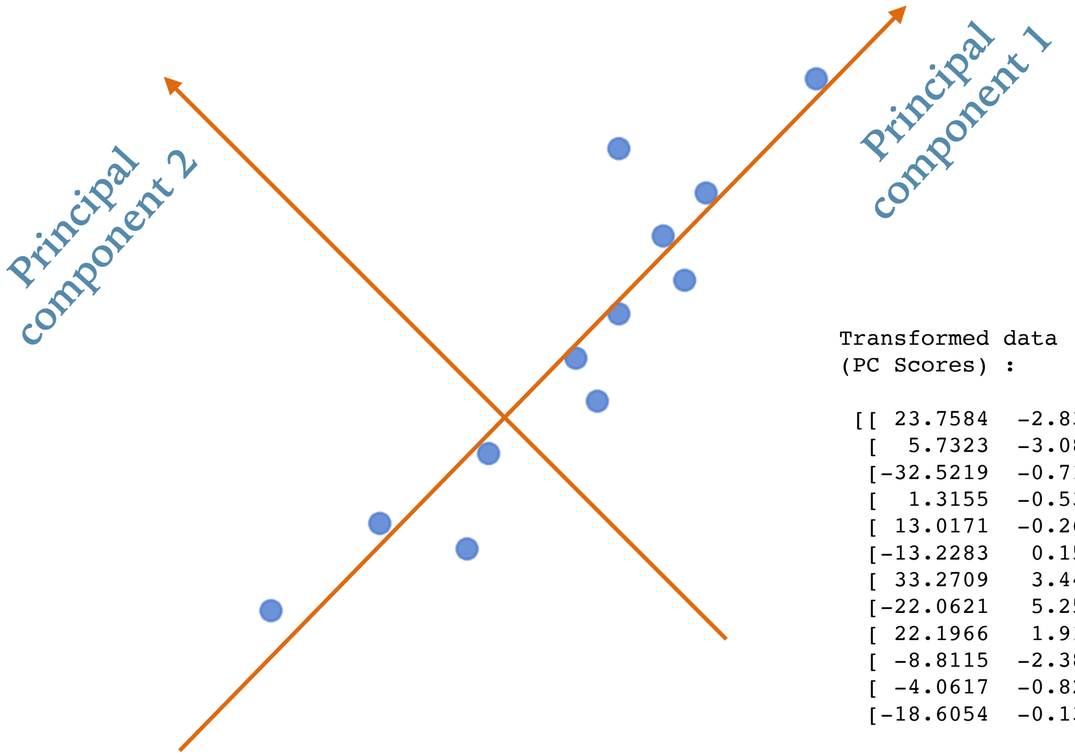
Dimension $= r \ll d$

Visualization
Clustering
Supervised learning
Anomaly detection

Principal Component Analysis (PCA) : Basic Concept

```
Original data

[[ 9 39]
 [15 56]
 [25 93]
 [14 61]
 [10 50]
 [18 75]
 [ 0 32]
 [16 85]
 [ 5 42]
 [19 70]
 [16 66]
 [20 80]]
```






```
Transformed data
(PC Scores) :

[[ 23.7584  -2.8373]
 [  5.7323  -3.0802]
 [-32.5219  -0.711 ]
 [  1.3155  -0.5324]
 [ 13.0171  -0.2637]
 [-13.2283   0.1592]
 [ 33.2709   3.4487]
 [-22.0621   5.2548]
 [ 22.1966   1.9125]
 [ -8.8115  -2.3886]
 [ -4.0617  -0.8268]
 [-18.6054  -0.1352]]
```

Principal Component (PC) = Unit vector representing an axis direction.

PC 1 = Axis with PC scores having highest variance

PC 2 = Axis with PC scores having 2nd highest variance

$\mathbf{X}$ $\mathbf{X-\mu}$ $\mathbf{P}$ $\mathbf{Z}$

Original data:

```
[[ 9 39]
 [15 56]
 [25 93]
 [14 61]
 [10 50]
 [18 75]
 [ 0 32]
 [16 85]
 [ 5 42]
 [19 70]
 [16 66]
 [20 80]]
```

Zero-mean data:

```
[[ -4.92 -23.42]
 [  1.08  -6.42]
 [ 11.08  30.58]
 [  0.08  -1.42]
 [ -3.92 -12.42]
 [  4.08  12.58]
 [-13.92 -30.42]
 [  2.08  22.58]
 [ -8.92 -20.42]
 [  5.08   7.58]
 [  2.08   3.58]
 [  6.08  17.58]]
```

Principal components:

```
[[0.3201  0.9474]
 [0.9474 -0.3201]]
```

PC1 PC2

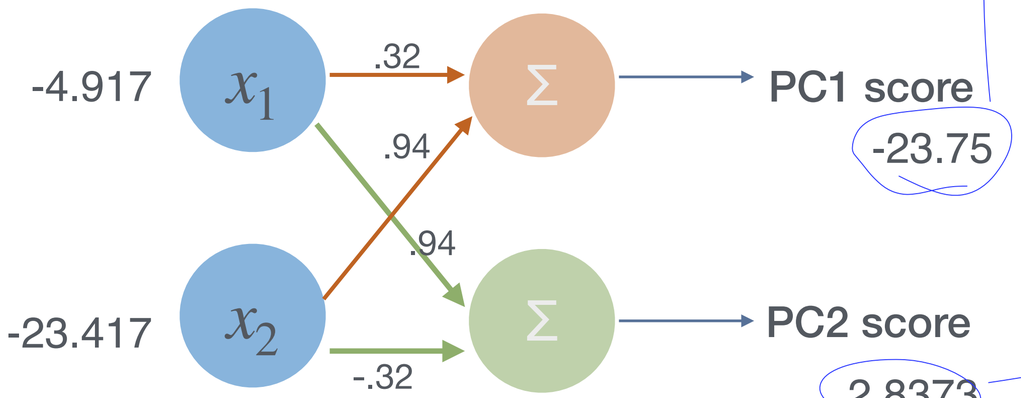
x_2 gets more weight (***loading***) in PC1

x_1 gets more weight in PC2

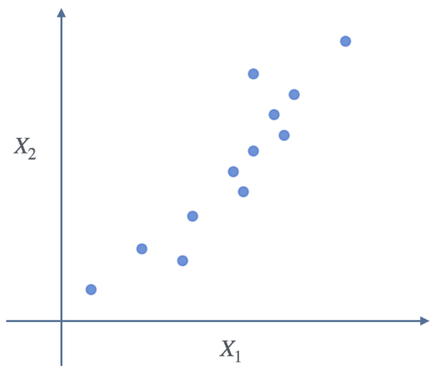
Transformed data (PC scores):

```
[[-23.7584   2.8373]
 [ -5.7323   3.0802]
 [ 32.5219   0.711 ]
 [ -1.3155   0.5324]
 [-13.0171   0.2637]
 [ 13.2283  -0.1592]
 [-33.2709  -3.4487]
 [ 22.0621  -5.2548]
 [-22.1966  -1.9125]
 [  8.8115   2.3886]
 [  4.0617   0.8268]
 [ 18.6054   0.1352]]
```

Zero column means

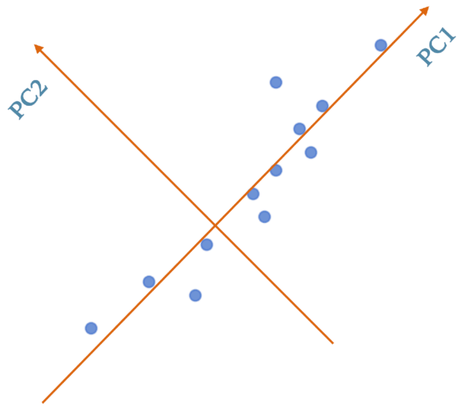


Proportion of Variance Explained (PVE)




```
[[ 9 39]
 [15 56]
 [25 93]
 [14 61]
 [10 50]
 [18 75]
 [ 0 32]
 [16 85]
 [ 5 42]
 [19 70]
 [16 66]
 [20 80]]
```

Total variance = 47.7 + 370 = 417.7



PC1 scores	PC2 scores
23.7584	-2.8373
5.7323	-3.0802
-32.5219	-0.711
1.3155	-0.5324
13.0171	-0.2637
-13.2283	0.1592
33.2709	3.4487
-22.0621	5.2548
22.1966	1.9125
-8.8115	-2.3886
-4.0617	-0.8268
-18.6054	-0.1352

Var(PC score 1)

All PC scores are uncorrelated and have zero means

Total variance = 411.6 + 6.18 = 417.7

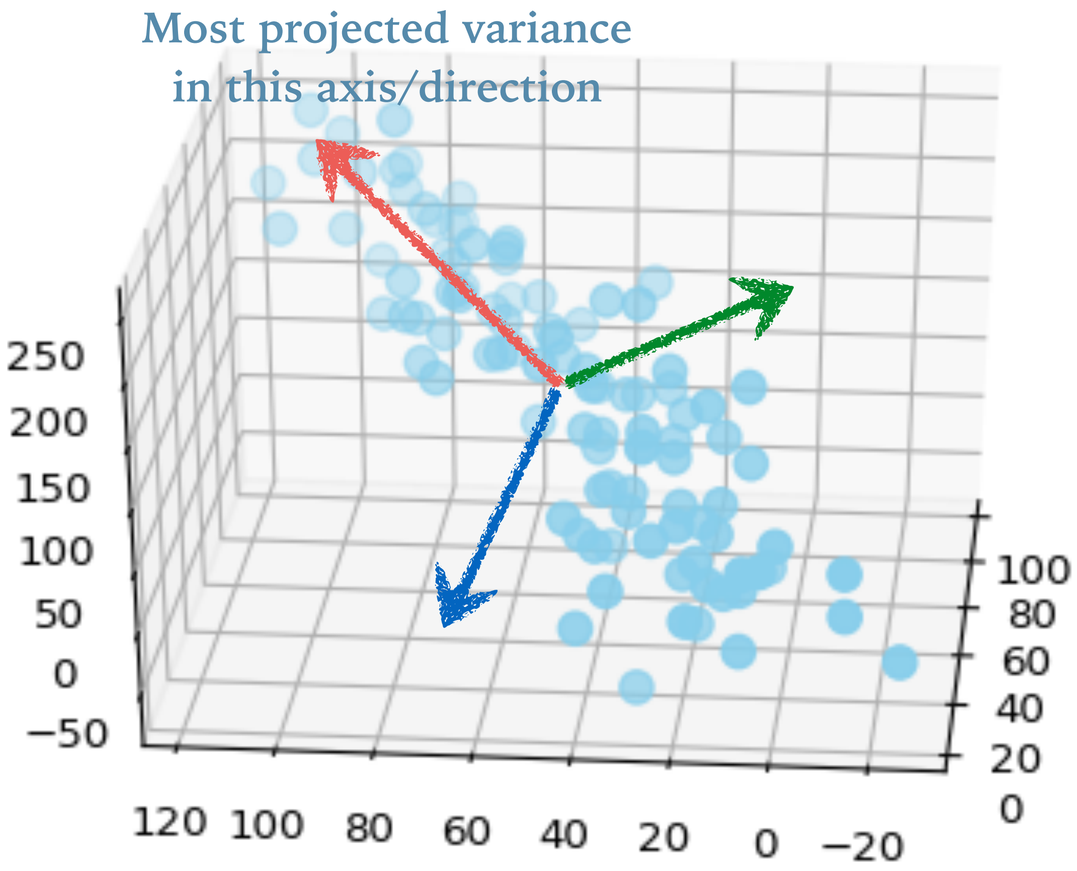
Var(PC score 2)

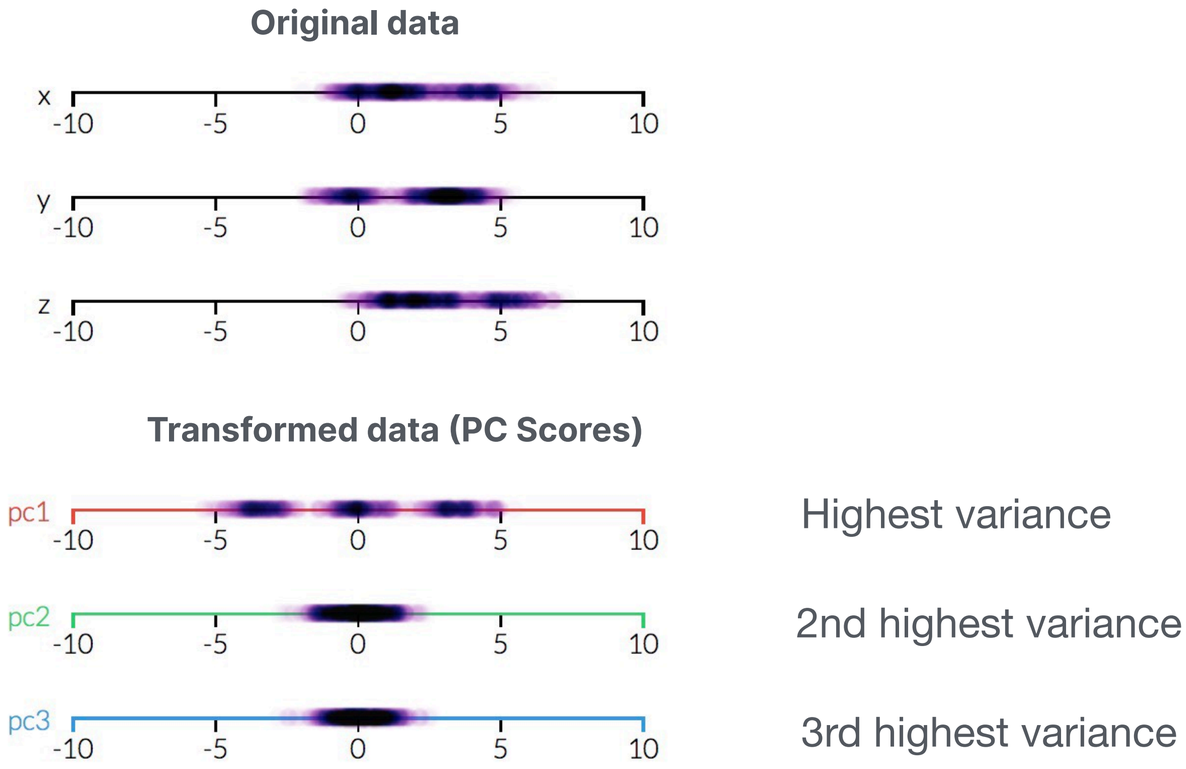
PVE by PC1 ~ 98.52%

PVE by PC2 ~ 1.47%

Total variance of original data = Total variance of **all PC scores**.

Since PC1 scores capture most information in the original data, the dimension thus can be reduced from 2 to 1.

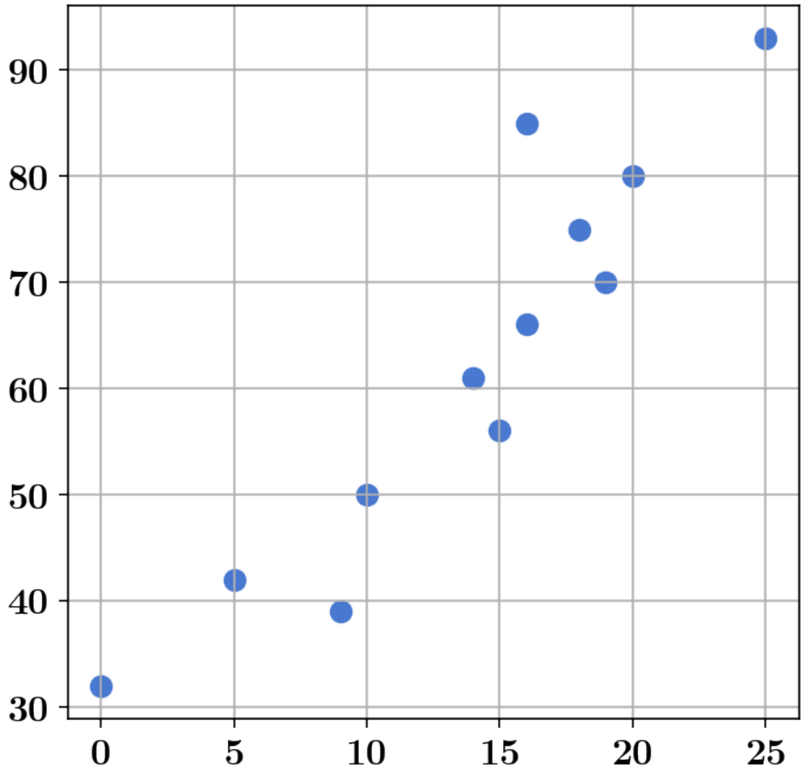




Data dimension could be reduced from 3 to 2.

Sample Covariance of Two Centered Variables

	Raw data before centering		$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_1 \cdot \mathbf{X}_2$
$\mathbf{x}_1$	9	39	-4.92	-23.42	115.23
$\mathbf{x}_2$	15	56	1.08	-6.42	-6.93
	25	93	11.08	30.58	338.83
	14	61	0.08	-1.42	-0.11
	10	50	-3.92	-12.42	48.69
	18	75	4.08	12.58	51.33
	0	32	-13.92	-30.42	423.45
	16	85	2.08	22.58	46.97
	5	42	-8.92	-20.42	182.15
	19	70	5.08	7.58	38.51
	16	66	2.08	3.58	7.45
$\mathbf{x}_n$	20	80	6.08	17.58	106.89



$$\text{(Sample) } \mathbb{C}\text{ov}(X_1, X_2) = \frac{\sum_{i=1}^{n}(x_{i1} - \overline{X}_1)(x_{i2} - \overline{X}_2)}{n-1}$$

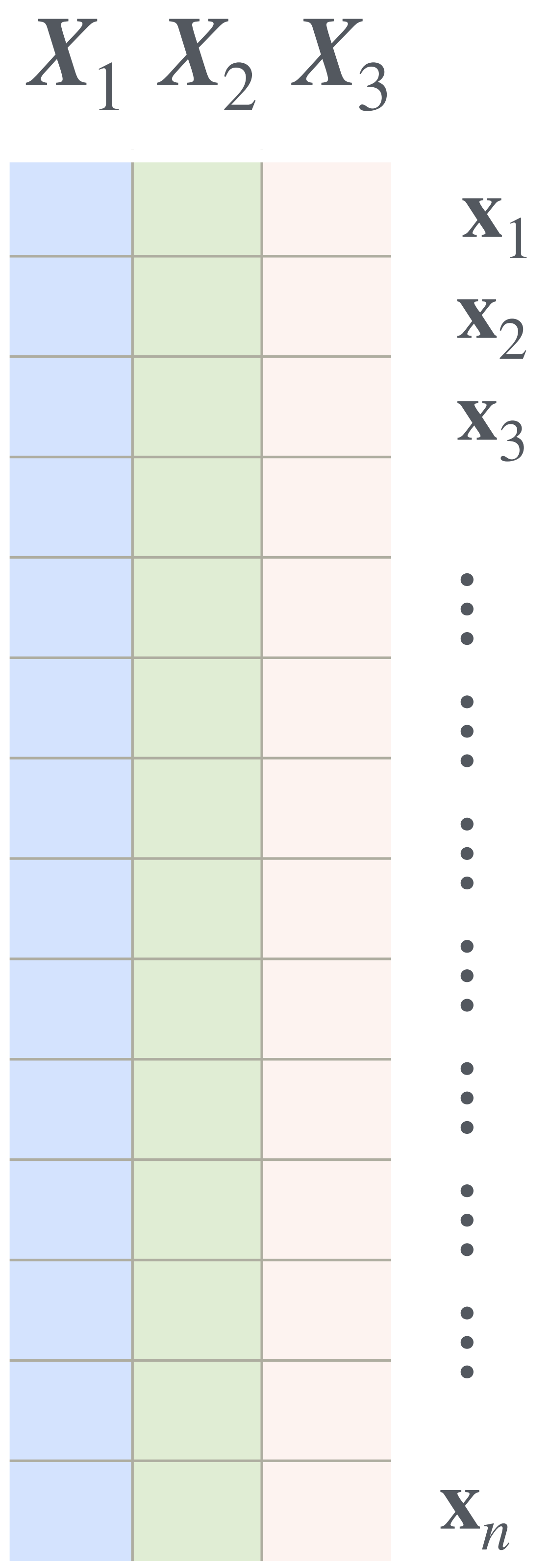
$$= \frac{\boldsymbol{X}_1 \cdot \boldsymbol{X}_2}{n-1} \quad \text{if } \boldsymbol{X}_1 \text{ and } \boldsymbol{X}_2 \text{ have zero mean}$$

$$\mathbb{C}\text{ov}(X_1, X_1) = \mathbb{V}\text{ar}(X_1)$$

Sample Covariance Matrix of a Random Vector

- ☐ Represent all pair-wise linear relationships in dataset.
- ☐ Let $\mathbf{X}$ be an $(n \times d)$ data matrix with ***zero-mean*** columns $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_d\}$ and rows $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

$\boldsymbol{X} =$



$$\boldsymbol{\Sigma} = \begin{bmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) & \mathrm{Cov}(X_1, X_3) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) & \mathrm{Cov}(X_2, X_2) \\ \mathrm{Cov}(X_3, X_1) & \mathrm{Cov}(X_3, X_2) & \mathrm{Cov}(X_3, X_3) \end{bmatrix}$$

$$= \frac{1}{n-1} \begin{bmatrix} \boldsymbol{X}_1 \cdot \boldsymbol{X}_1 & \boldsymbol{X}_1 \cdot \boldsymbol{X}_2 & \boldsymbol{X}_1 \cdot \boldsymbol{X}_3 \\ \boldsymbol{X}_2 \cdot \boldsymbol{X}_1 & \boldsymbol{X}_2 \cdot \boldsymbol{X}_2 & \boldsymbol{X}_2 \cdot \boldsymbol{X}_3 \\ \boldsymbol{X}_3 \cdot \boldsymbol{X}_1 & \boldsymbol{X}_3 \cdot \boldsymbol{X}_2 & \boldsymbol{X}_3 \cdot \boldsymbol{X}_3 \end{bmatrix}$$

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

$$= \frac{1}{n-1} \sum_{i=1}^{n} \boldsymbol{x}_i^T \boldsymbol{x}_i = \frac{1}{n-1} \sum_{i=1}^{n} \begin{bmatrix} x_{i1}^2 & x_{i1}x_{i2} & \cdots & x_{i1}x_{id} \\ x_{i2}x_{i1} & x_{i2}^2 & \cdots & x_{i2}x_{i3} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i3}x_{i1} & x_{i3}x_{i2} & \cdots & x_{id}^2 \end{bmatrix}$$

$$A = \begin{bmatrix} \sigma_{(1,1)} & \sigma_{(1,2)} \\ \sigma_{(2,1)} & \sigma_{(2,2)} \end{bmatrix}$$

$\det(A - \lambda I_n) = \vec{0}$
↓
ได้ eigenvalue
↓
แทน λ คืน + ทำ row echelon
↓
eigen $\vec{v}$

=0.

X_1	X_2	$X_1 \cdot X_2$
-4.92	-23.42	115.23
1.08	-6.42	-6.93
11.08	30.58	338.83
0.08	-1.42	-0.11
-3.92	-12.42	48.69
4.08	12.58	51.33
-13.92	-30.42	423.45
2.08	22.58	46.97
-8.92	-20.42	182.15
5.08	7.58	38.51
2.08	3.58	7.45
6.08	17.58	106.89
954	**7400**	**1352.4**
47.7	**370**	**122.9**

(Squared Total)

Average

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

$$\begin{vmatrix} 47.7-\lambda & 122.9 \\ 122.9 & 370-\lambda \end{vmatrix} = 0$$

$$(47.7-\lambda)(370-\lambda) - 122.9^2 = 0$$

$$\lambda^2 - 417.7\lambda + \underbrace{47.7(370)}_{2544.59} - 122.9^2 = 0$$

$$\therefore \lambda = 6.18,\ 411.52 \quad (*)$$

$$\Sigma \approx \begin{bmatrix} 47.7 & 122.9 \\ 122.9 & 370 \end{bmatrix}$$

eigen $\vec{v}$

$\lambda = 411.6$;

$$\begin{bmatrix} 47.7-411.6 & 122.9 \\ 122.9 & 370-411.6 \end{bmatrix} = \begin{bmatrix} -363.9 & 122.9 \\ 122.9 & -41.6 \end{bmatrix}$$

$$\times \underbrace{\frac{1}{122.9}} \begin{bmatrix} -2.96 & 1 \\ 1 & 0.34 \end{bmatrix}$$

$$\underbrace{R_1 \times -\frac{1}{2.96}}_{R_2 - R_1} \begin{bmatrix} 1 & \overbrace{-\frac{1}{2.96}}^{-0.34} \\ 0 & 0.68 \end{bmatrix}$$

$$\underbrace{R_2/2}_{R_2+R_1} \begin{bmatrix} 1 & -0.34 \\ 0 & 0 \end{bmatrix}$$

$p_1 - 0.34 p_2 = 0$

$p_1 = 0.34 p_2$

$$\begin{bmatrix} 1 \\ 0.34 \end{bmatrix}$$

$$\text{Variance of each variable} = \text{var}(X_i) = \sum_j (X_{ij} - \overline{X}_i)^2/(n-1)$$

$$\text{Total variance in the dataset} = \sum_i \text{var}(X_i) = 417.7$$

PCA under the hood

Given the sample covariance matrix $\mathbf{\Sigma}$ of the input data, project the data on the new axes (PCs) so that

$$\mathbf{\Sigma} \quad \Rightarrow \quad \mathbf{\Sigma}_Z$$

Original covariance matrix

$$\begin{bmatrix} 47.7 & 122.9 \\ 122.9 & 370 \end{bmatrix}$$

$$\begin{bmatrix} 411.62 & 0 \\ 0 & 6.18 \end{bmatrix}$$

New covariance matrix with only diagonal entries (All PCs uncorrelated)

☐ Let $\mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2\ \ldots\ \mathbf{p}_d]$ and $\{\lambda_1, \lambda_2, \ldots, \lambda_d\}$ respectively be eigenvectors and eigenvalues of $\boldsymbol{\Sigma}$ (largest ones first).

$$\boldsymbol{\Sigma} = \begin{bmatrix} 47.7 & 122.9 \\ 122.9 & 370 \end{bmatrix}$$

eigenvector 1 — eigenvalue 1

$$\boldsymbol{\Sigma}\,\boldsymbol{p}_1 = \lambda_1 \boldsymbol{p}_1$$

$$\boldsymbol{\Sigma}\,\boldsymbol{p}_2 = \lambda_2 \boldsymbol{p}_2$$

$$\boldsymbol{P} = [\boldsymbol{p}_1\ \boldsymbol{p}_2] = \begin{bmatrix} -0.32 & -0.947 \\ -0.947 & 0.32 \end{bmatrix}$$

Eigenvector(Σ) with largest eigenvalues first

$\lambda_1 = 411.6$ $\quad$ $\lambda_1 = 6.18$

☐ Then, the following transformation on zero-mean $\mathbf{X}$ results in the data with a diagonal covariance matrix:

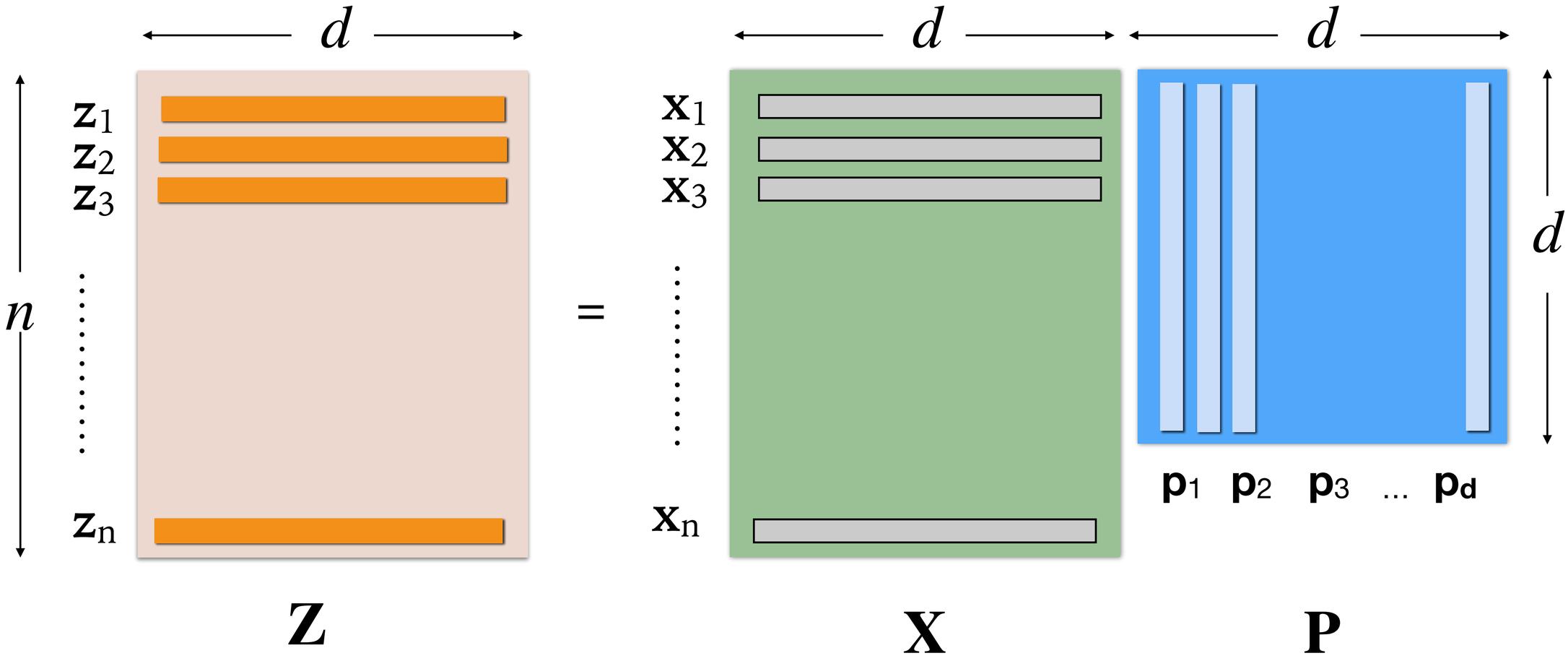
$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{P}$$

zero-mean — transformed — eigenvector

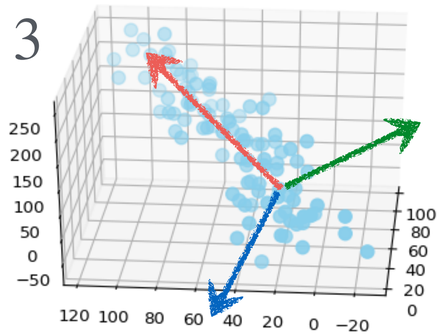
```
[[ -4.917 -23.417]
 [  1.083  -6.417]
 [ 11.083  30.583]
 [  0.083  -1.417]
 [ -3.917 -12.417]
 [  4.083  12.583]
 [-13.917 -30.417]
 [  2.083  22.583]
 [ -8.917 -20.417]
 [  5.083   7.583]
 [  2.083   3.583]
 [  6.083  17.583]]
```

```
[[ -0.320 -0.947]
 [ -0.947  0.320]]
```

Transformed data



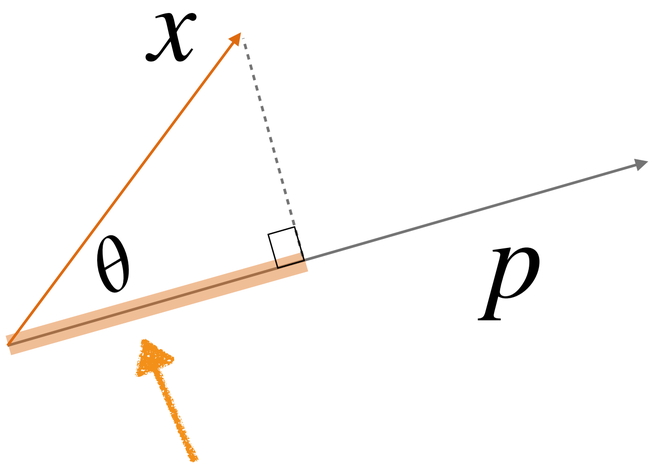
$d = 3$



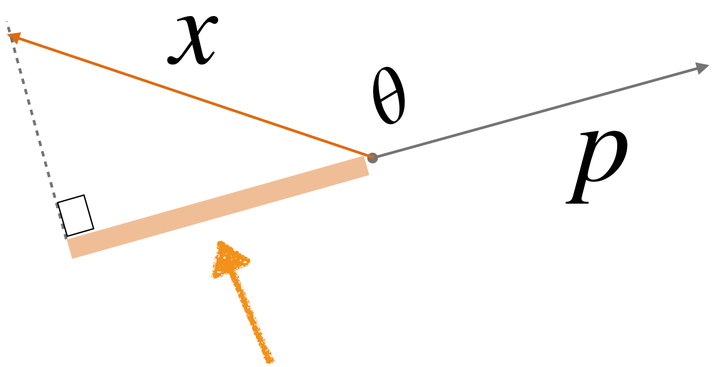
$$\mathbf{z}_i = \mathbf{x}_i\mathbf{P}$$
$$= [\mathbf{x}_i \cdot \mathbf{p}_1 \ \mathbf{x}_i \cdot \mathbf{p}_2 \ \cdots \ \mathbf{x}_i \cdot \mathbf{p}_d]$$

Scalar Projection

- Project "size" of one vector on another vector direction.

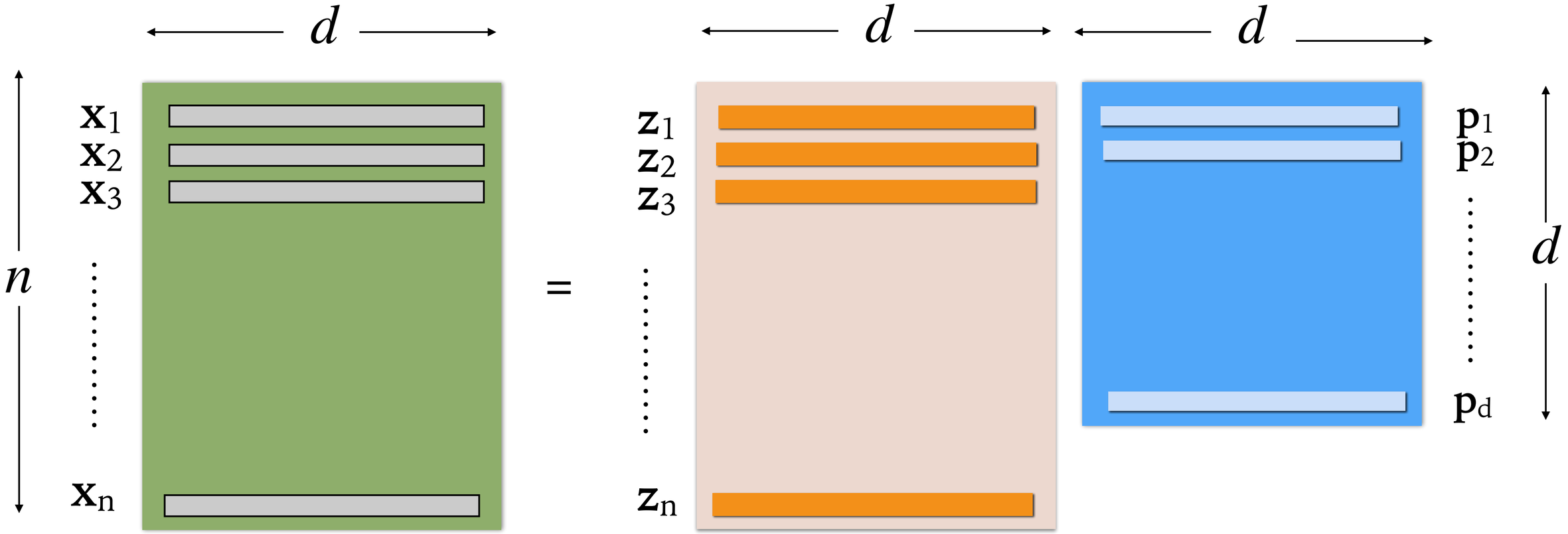


$$\text{Proj}_{\boldsymbol{p}}\boldsymbol{x} = \boldsymbol{x} \cdot \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|}$$



$$\text{Proj}_{\boldsymbol{p}}\boldsymbol{x} = \boldsymbol{x} \cdot \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|}$$

Inverse transformed data



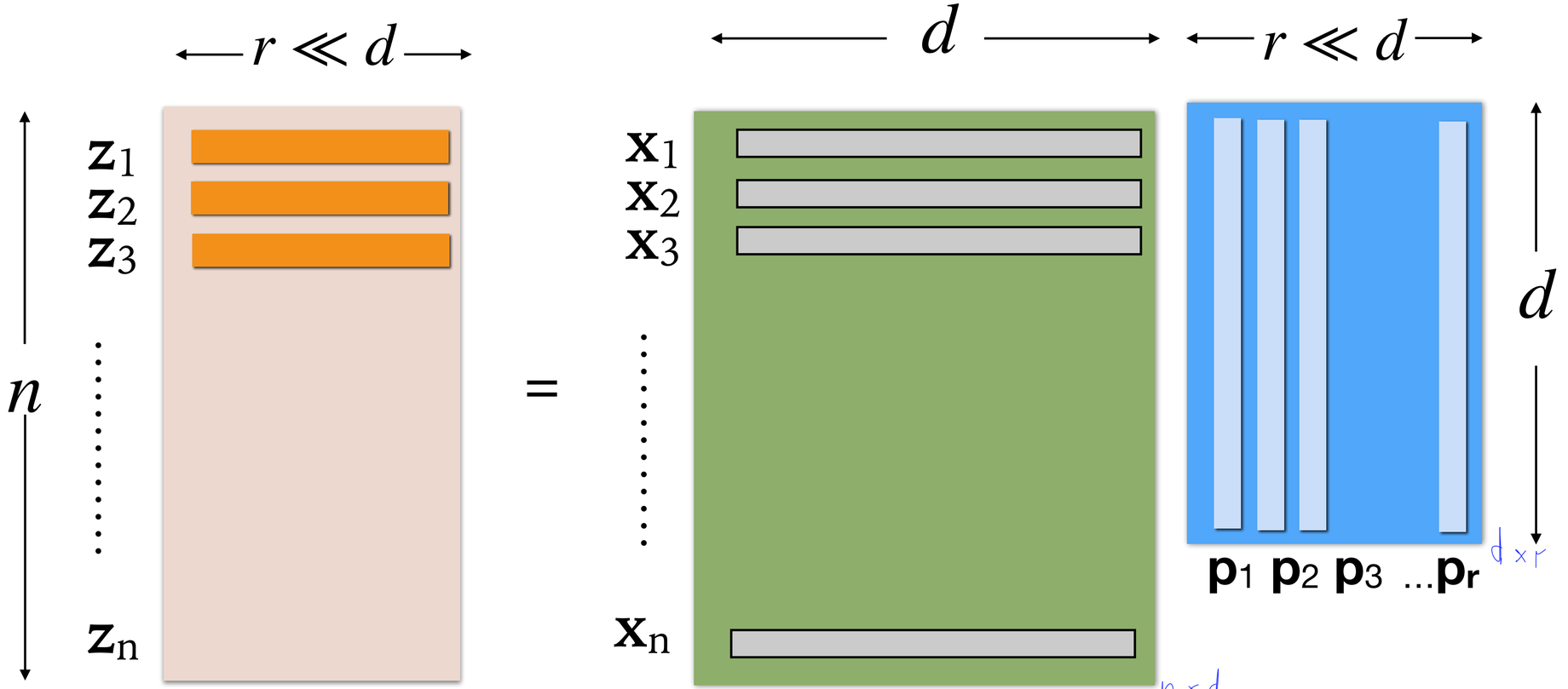
$$X_{n\times d} = \begin{bmatrix} & \boldsymbol{X}_1 & \boldsymbol{X}_2 & \cdots & \boldsymbol{X}_d \\ \hline \boldsymbol{x}_1 & x_{11} & x_{12} & \cdots & x_{1d} \\ \boldsymbol{x}_2 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_n & x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}$$

$$\mathbf{z}_i = [z_{i1}, z_{i2}, \cdots, z_{id}]$$

$$\mathbf{x}_i = z_{i1}\mathbf{p}_1^T + z_{i2}\mathbf{p}_2^T + \cdots + z_{id}\mathbf{p}_d^T$$

$$\boldsymbol{X} = \boldsymbol{Z}\,\boldsymbol{P}^T \quad (+\text{subtracted mean})$$

Reducing Dimension



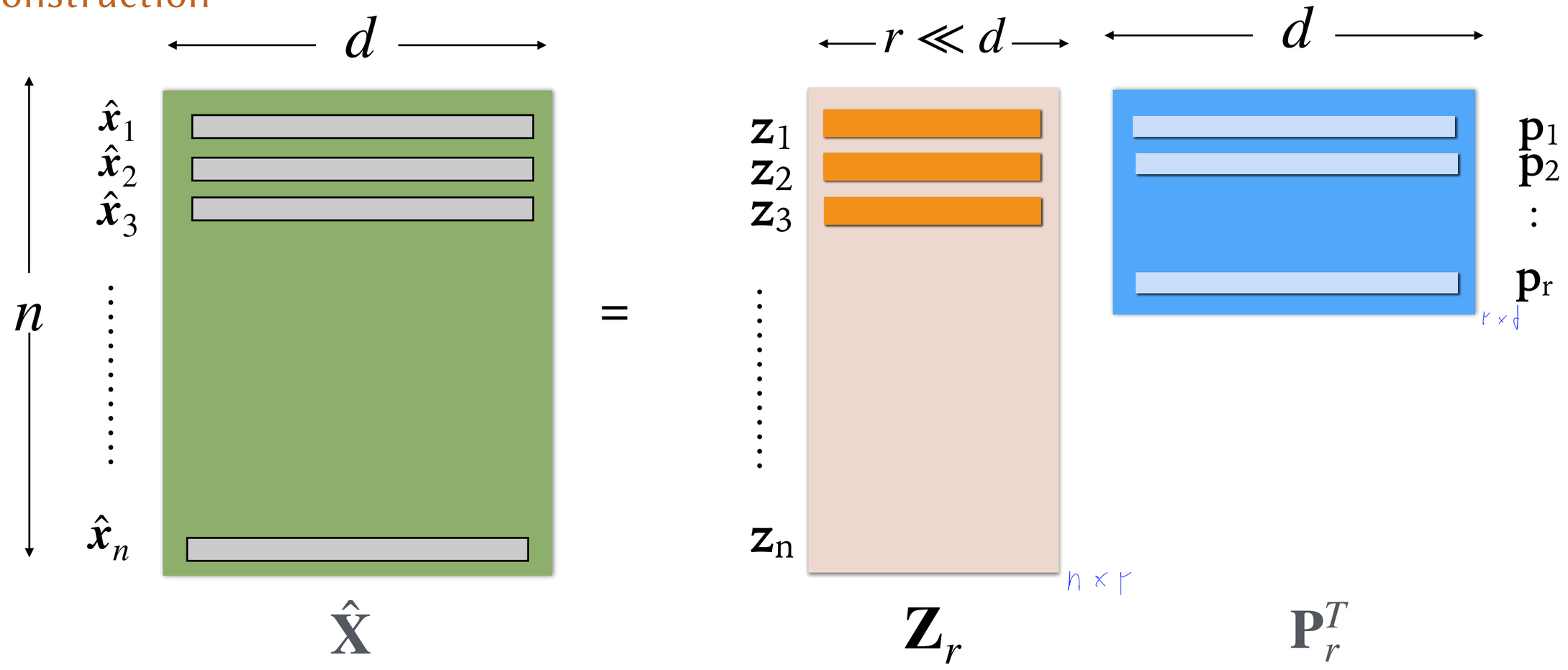
$\mathbf{Z}_r$ $\mathbf{X}$ $\mathbf{P}_r$

$$\mathbf{z}_i = \mathbf{x}_i \mathbf{P}_r$$

Entries = First r Eigenvectors of Σ with largest eigenvalues first

* เรียงตาม eigenvalues ก่อน เลือก

Reconstruction



$$\hat{X}_{n\times d} = \begin{bmatrix} & X_1 & X_2 & \cdots & X_d \\ \hat{x}_1 & x_{11} & x_{12} & \cdots & x_{1d} \\ \hat{x}_2 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{x}_n & x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}$$

$$\mathbf{z}_i = [z_{i1}, z_{i2}, \cdots, z_{ir}]$$

$$\hat{\mathbf{x}}_i = z_{i1}\mathbf{p}_1^T + z_{i2}\mathbf{p}_2^T + \cdots + z_{ir}\mathbf{p}_r^T$$

$$\hat{X} = Z_r P_r^T \quad (+\text{subtracted mean})$$

Variance Properties of Projected Data Set in Reduced Dimension

$$\boldsymbol{\Sigma}_Z = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \lambda_r \end{bmatrix}$$

$\lambda_i =$ Eigval($\boldsymbol{\Sigma}$) = Variance of PC i

Total variance of original data $= \lambda_1 + \lambda_2 + \cdots + \lambda_d$

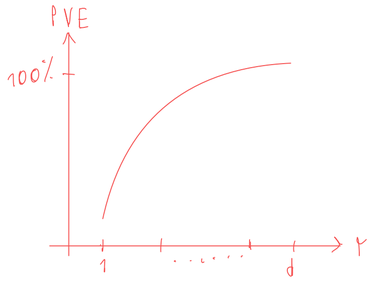
Total variance of PC scores $= \lambda_1 + \lambda_2 + \cdots + \lambda_r$

Proportion of variance explained (PVE) by first r PCs:

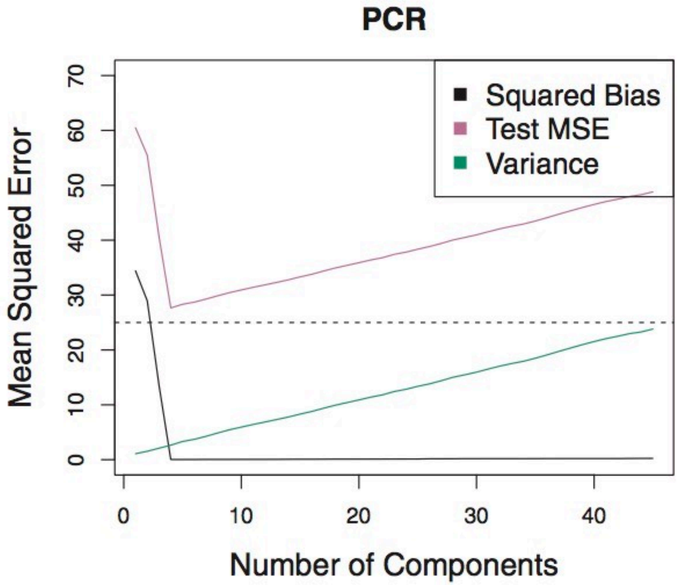
$$f(r) = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_r}{\lambda_1 + \lambda_2 + \cdots + \lambda_d} = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$$

reduce / original

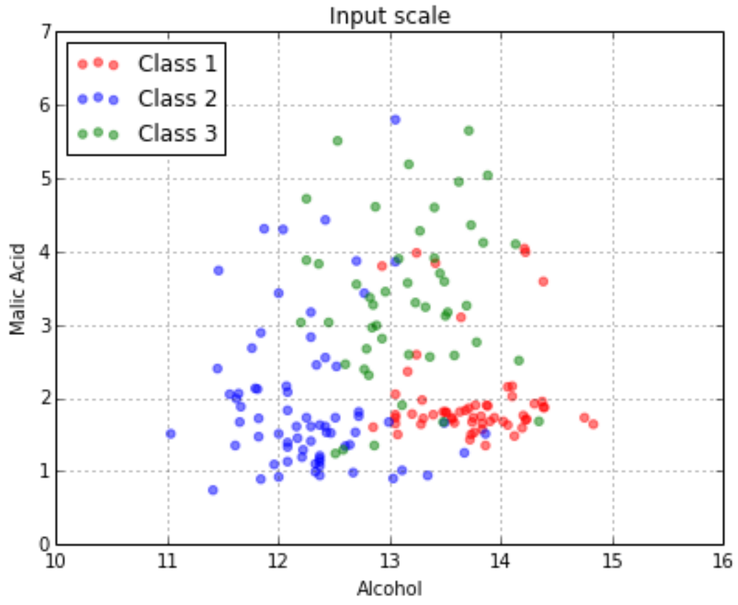
$$\begin{bmatrix} 411.62 & 0 \\ 0 & 6.18 \end{bmatrix}$$

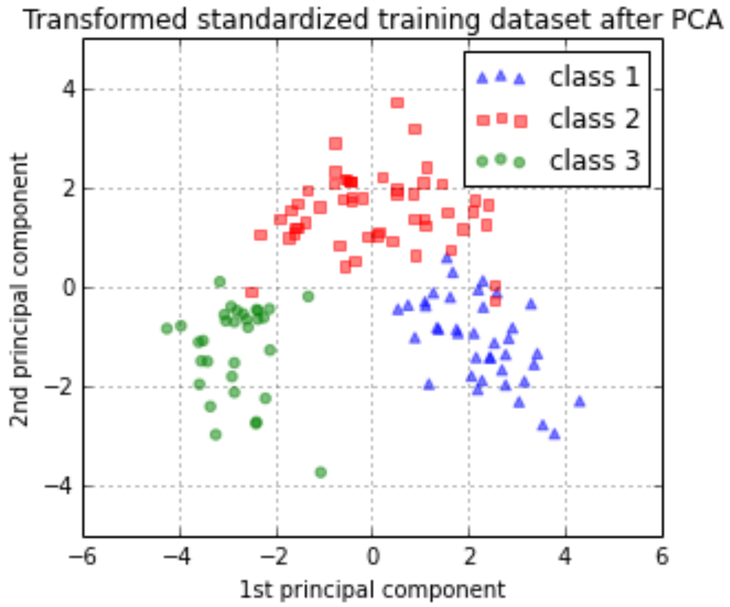


Applying PCA to Supervised Learning



Principal component regression (PCR)

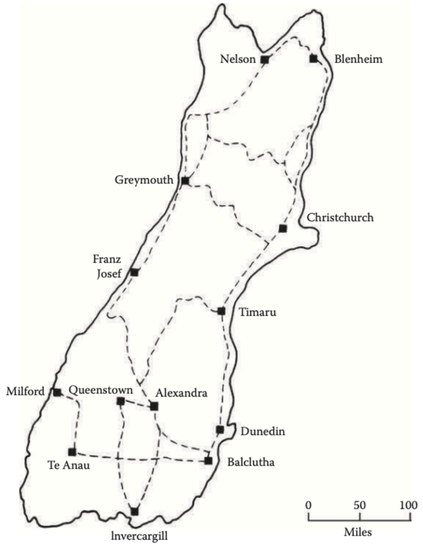




Could yield bettter class separability and guide the choices of classification algorithm.

- Given a map, can you generate some intercity distances ?
- Given a set of distances, can you recreate the (2D) map itself ?

	Berlin	Dresden	Hamburg	Koblenz	Munich	Rostock
Berlin	0	214	279	610	596	237
Dresden		0	492	533	496	444
Hamburg			0	520	772	140
Koblenz				0	521	687
Munich					0	771
Rostock						0



- Applications in political science, psychology, marketing, etc.

Table 11.5 The distances between 15 congressmen from New Jersey in the United States House of Representatives

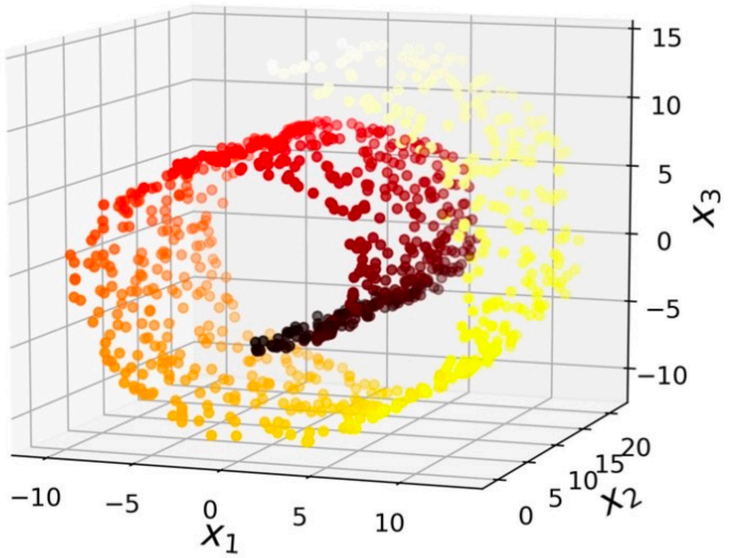
	Hunt	Sandman	Howard	Thompson	Frelinghuysen	Forsythe	Widnall	Roe	Helstoski	Rodino	Minish	Rinaldo	Maraziti	Daniels	Pattern
Hunt (R)	0														
Sandman (R)	8	0													
Howard (D)	15	17	0												
Thompson (D)	15	12	9	0											
Frelinghuysen (R)	10	13	16	14	0										
Forsythe (R)	9	13	12	12	8	0									
Widnall (R)	7	12	15	13	9	7	0								
Roe (D)	15	16	5	10	13	12	17	0							
Helstoski (D)	16	17	5	8	14	11	16	4	0						
Rodino (D)	14	15	6	8	12	10	15	5	3	0					
Minish (D)	15	16	5	8	12	9	14	5	2	1	0				
Rinaldo (R)	16	17	4	6	12	10	15	3	1	2	1	0			
Maraziti (R)	7	13	11	15	10	6	10	12	13	11	12	12	0		
Daniels (D)	11	12	10	10	11	6	11	7	7	4	5	6	9	0	
Pattern (D)	13	16	7	7	11	10	13	6	5	6	5	4	13	9	0

Note: The numbers shown are the number of times that the congressmen voted differently on 19 environmental bills. R = Republican party, D = Democratic party.

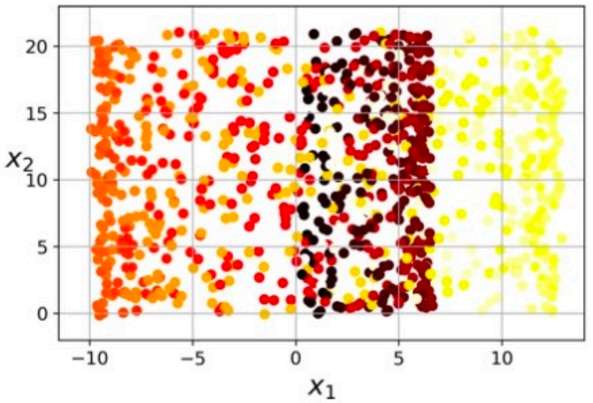
Manifold Learning

- Discover low-dimensional structure (manifold) in R^q on which the data actually lies from a higher dimensional input space R^d.
- Learning types are either ***linear*** or ***non-linear***.

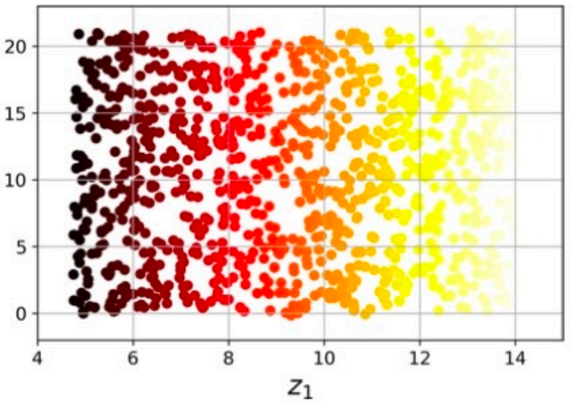
MDS : pairwise-distance (linear)
LLE : local structure (non-linear)
ISOMAP : global structure (non-linear)



Original Space



Low-dimension space
by using projection



Low-dimension space
by unrolling (2D manifold)

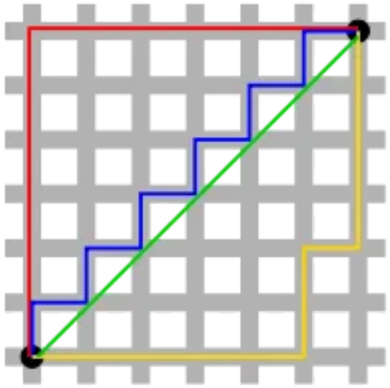
Proximity Matrix

- Only need pair-wise dissimilarity of objects as the input -- Interval scale or Ordinal scale
- Let $\mathbf{\Delta}$ be the proximity matrix with entries δ_{ij} representing dissimilarity between two objects i and j.
- Ex: Common dissimilarity metrics:

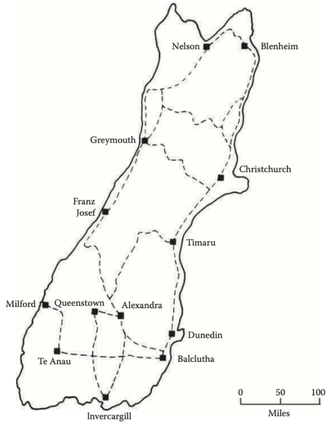
$$\text{Euclidean Distance:} \quad \delta_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| = \left[\sum_k (x_{ik} - x_{jk})^2\right]^{\frac{1}{2}}$$

$$\text{Manhattan Distance:} \quad \delta_{ij} = \sum_k |x_{ik} - x_{jk}|$$

$$\text{Minowski Distance:} \quad \delta_{ij} = \left[\sum_k |x_{ik} - x_{jk}|^m\right]^{\frac{1}{m}}$$



- Geographical distance or ranking also possible.



Multi-Dimensional Scaling (MDS)

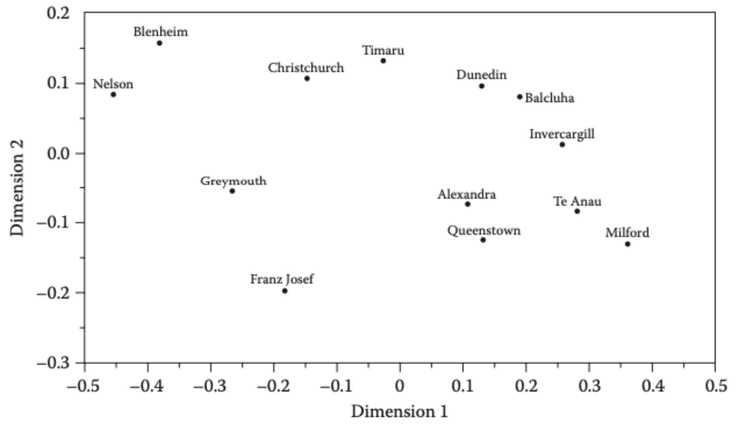
- Project p-dimensional data to q-dimensional data points $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n \in R^q$ (*principal coordinates* or *configuration*) such that the ***monotonicity*** of the pairwise distances in the original data are preserved in q-space.
- For $M = \frac{n(n-1)}{2}$ pair-wise distances descendingly sorted:

$$\delta_{i1,k1} > \delta_{i2,k2} > \cdots > \delta_{in,kn}$$

- Find q-dimensional data points $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$, $\mathbf{y}_i \in R^q$, $q \ll p$ such that

$$d_{i1,k1} > d_{i2,k2} > \cdots > d_{in,kn}$$

```
class sklearn.manifold.MDS(n_components=2, *, metric=True, n_init=4,
max_iter=300, verbose=0, eps=0.001, n_jobs=None, random_state=None,
dissimilarity='euclidean', normalized_stress='auto')
```



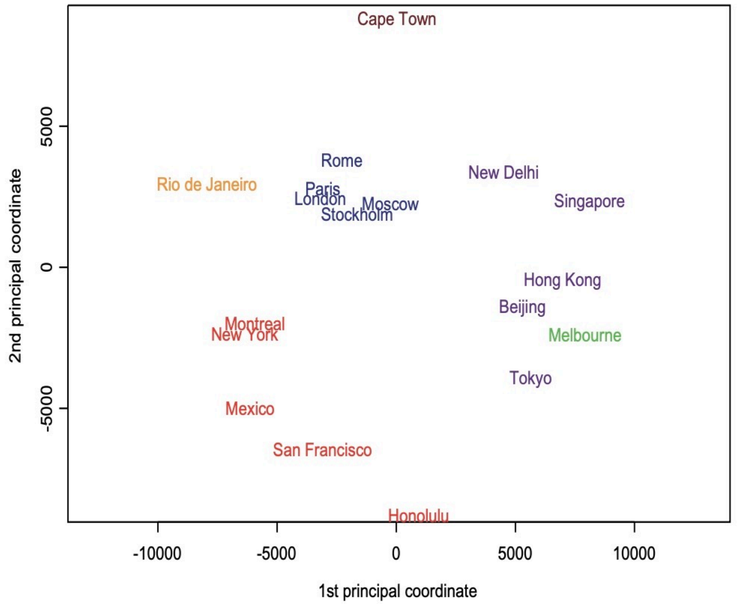
Example: Airline Distance

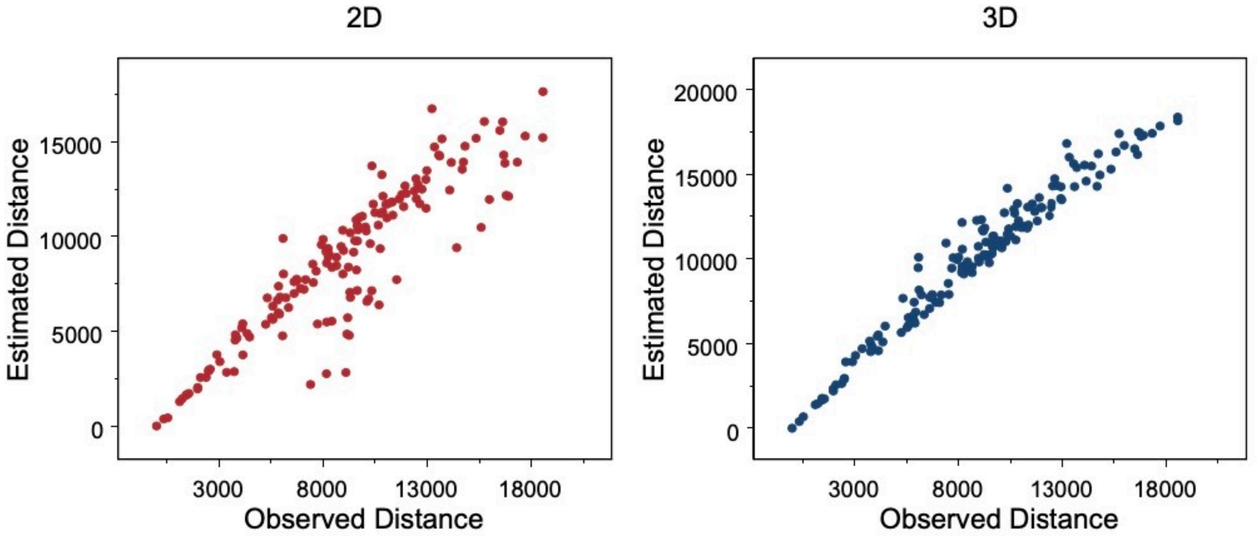
18 cities

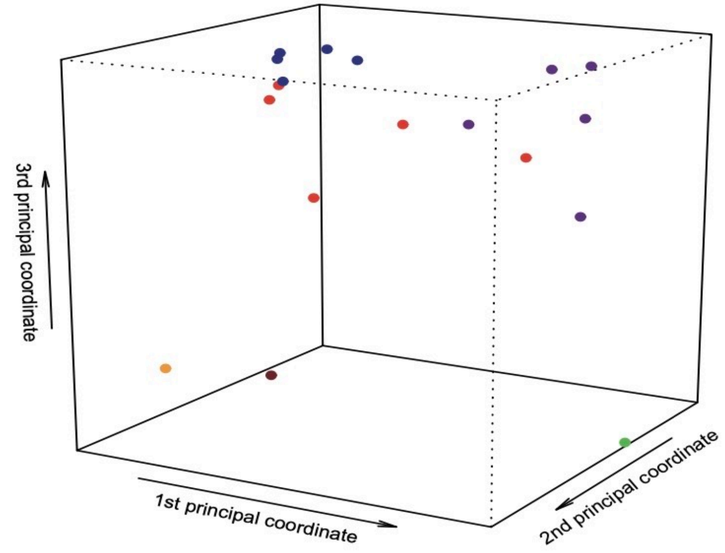
	Beijing	Cape Town	Hong Kong	Honolulu	London	Melbourne
Cape Town	12947					
Hong Kong	1972	11867				
Honolulu	8171	18562	8945			
London	8160	9635	9646	11653		
Melbourne	9093	10338	7392	8862	16902	
Mexico	12478	13703	14155	6098	8947	13557
Montreal	10490	12744	12462	7915	5240	16730
Moscow	5809	10101	7158	11342	2506	14418
New Delhi	3788	9284	3770	11930	6724	10192
New York	11012	12551	12984	7996	5586	16671
Paris	8236	9307	9650	11988	341	16793
Rio de Janeiro	17325	6075	17710	13343	9254	13227
Rome	8144	8417	9300	12936	1434	15987
San Francisco	9524	16487	11121	3857	8640	12644
Singapore	4465	9671	2575	10824	10860	6050
Stockholm	6725	10334	8243	11059	1436	15593
Tokyo	2104	14737	2893	6208	9585	8159

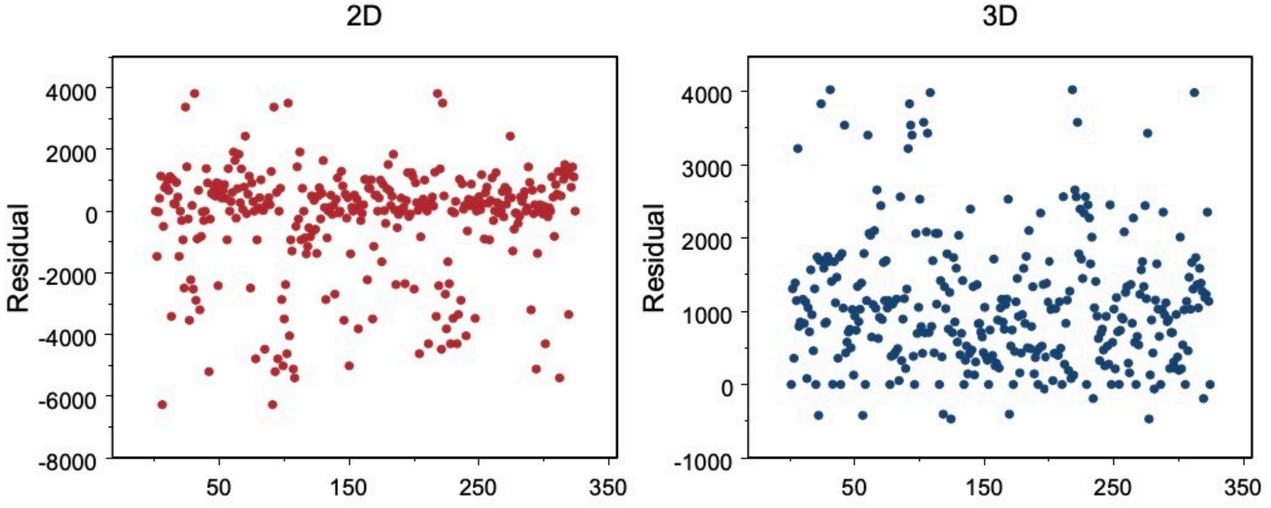
	Mexico	Montreal	Moscow	New Delhi	New York	Paris
Montreal	3728					
Moscow	10740	7077				
New Delhi	14679	11286	4349			
New York	3362	533	7530	11779		
Paris	9213	5522	2492	6601	5851	
Rio	7669	8175	11529	14080	7729	9146
Rome	10260	6601	2378	5929	6907	1108
S.F.	3038	4092	9469	12380	4140	8975
Singapore	16623	14816	8426	4142	15349	10743
Stockholm	9603	5900	1231	5579	6336	1546
Tokyo	11319	10409	7502	5857	10870	9738

	Rio	Rome	S.F.	Singapore	Stockholm
Rome	9181				
S.F.	10647	10071			
Singapore	15740	10030	13598		
Stockholm	10682	1977	8644	9646	
Tokyo	18557	9881	8284	5317	8193


Cape Town
Rome
New Delhi
Rio de Janeiro
Paris
London
Moscow
Stockholm
Singapore
Hong Kong
Beijing
Montreal
New York
Melbourne
Tokyo
Mexico
San Francisco
Honolulu
2nd principal coordinate
1st principal coordinate
5000
0
-5000
-10000
-5000
0
5000
10000


2D
3D
Estimated Distance
Observed Distance
15000
10000
5000
0
20000
3000
8000
13000
18000

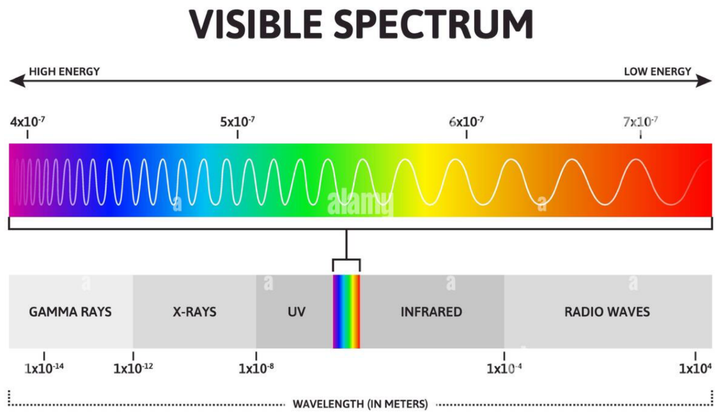

3rd principal coordinate
1st principal coordinate
2nd principal coordinate


2D
3D
Residual
4000
2000
0
-2000
-4000
-6000
-8000
3000
1000
-1000
50
150
250
350

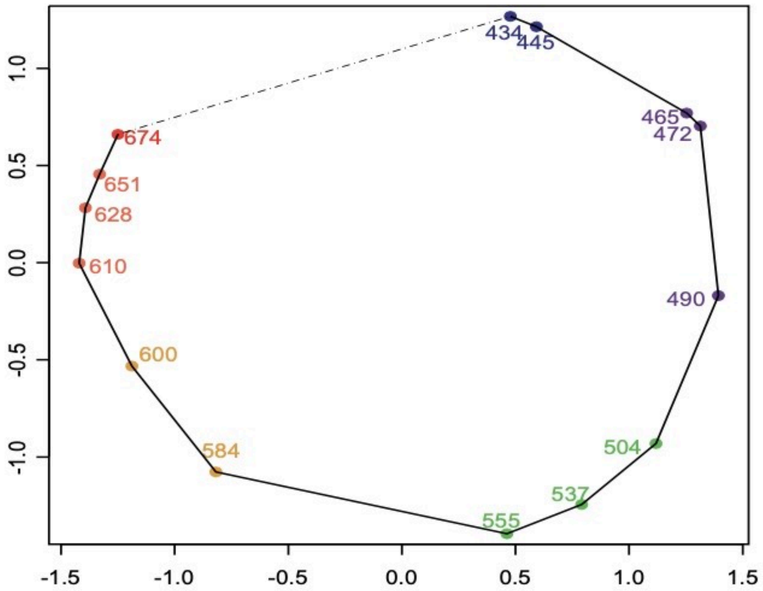
Example: Perceptions of Color Human Vision

	434	445	465	472	490	504	537	555	584	600	610	628	651
445	0.14												
465	0.58	0.50											
472	0.58	0.56	0.19										
490	0.82	0.78	0.53	0.46									
504	0.94	0.91	0.83	0.75	0.39								
537	0.93	0.93	0.90	0.90	0.69	0.38							
555	0.96	0.93	0.92	0.91	0.74	0.55	0.27						
584	0.98	0.98	0.98	0.98	0.93	0.86	0.78	0.67					
600	0.93	0.96	0.99	0.99	0.98	0.92	0.86	0.81	0.42				
610	0.91	0.93	0.98	1.00	0.98	0.98	0.95	0.96	0.63	0.26			
628	0.88	0.89	0.99	0.99	0.99	0.98	0.98	0.97	0.73	0.50	0.24		
651	0.87	0.87	0.95	0.98	0.98	0.98	0.98	0.98	0.80	0.59	0.38	0.15	
674	0.84	0.86	0.97	0.96	1.00	0.99	1.00	0.98	0.77	0.72	0.45	0.32	0.24

Dissimilarity matrix of color wavelengths



Meaningful visual representation of human perception of colors



Color circle

Locally Linear Embedding (LLE)

Analyzing overlapping local neighborhoods in order to determine the local structure.

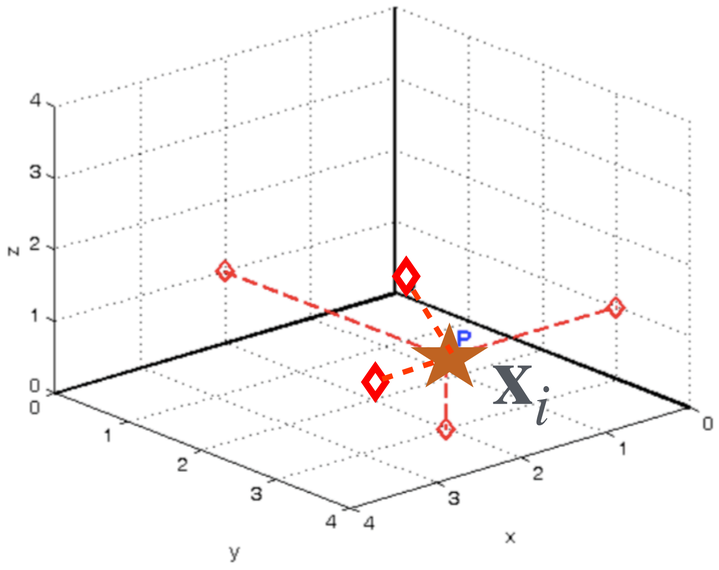
Construct a "neighborhood preserving embedding" of the data into the lower-dimensional space.

1: Find the nearest neighbors of each data point.
2: Express each point $\mathbf{x}_i$ as a linear combination of the other points, i.e., $\mathbf{x}_i = \sum_j w_{ij}\mathbf{x}_j$, where $\sum_j w_{ij} = 1$ and $w_{ij} = 0$ if $\mathbf{x}_j$ is not a near neighbor of $\mathbf{x}_i$.
3: Find the coordinates of each point in lower-dimensional space of specified dimension p by using the weights found in step 2.

Define the weight matrix $\mathbf{W} = [w_{ij}]$, which is obtained by minimizing

weight sum of neighbor x_i

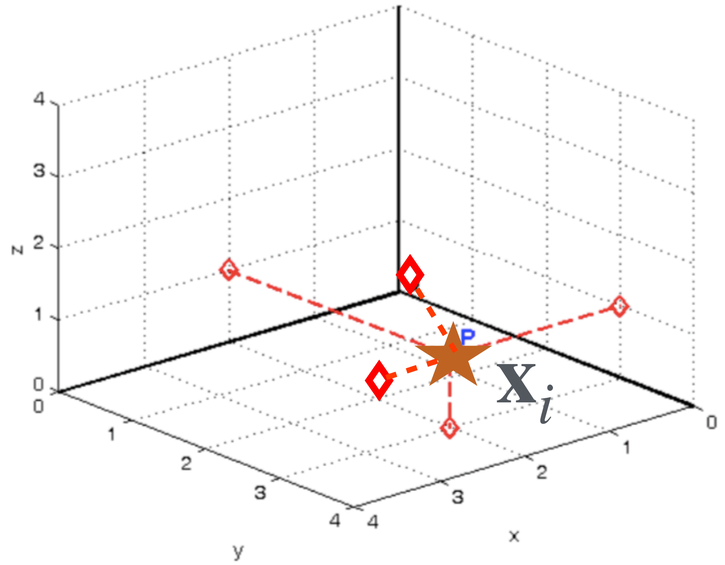
$$\text{Error}(\boldsymbol{W}) = \sum_{i=1}^{n} \left(\boldsymbol{x}_i - \sum_{j \in N(\boldsymbol{x}_i)} w_{ij}\boldsymbol{x}_j \right)^2$$

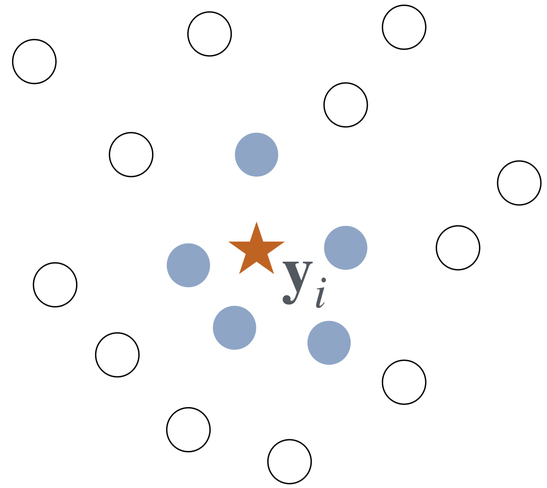


Let $\mathbf{y}_i$ be a new vector in the reduced dimension r and $\mathbf{Y}$ be the matrix whose i^{th} row is $\mathbf{y}_i$

Given the weight matrix $\mathbf{W}$ and low-dimension r specified by the user, find $\mathbf{Y}$ that minimizes

$$\text{Error}(\boldsymbol{Y}) = \sum_i \left(\boldsymbol{y}_i - \sum_{j \in N(\boldsymbol{x}_i)} w_{ij} \boldsymbol{y}_j \right)^2$$



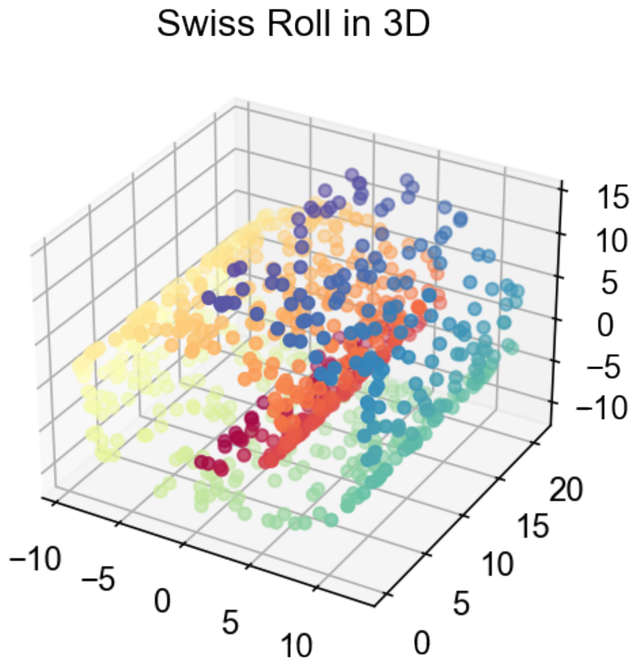


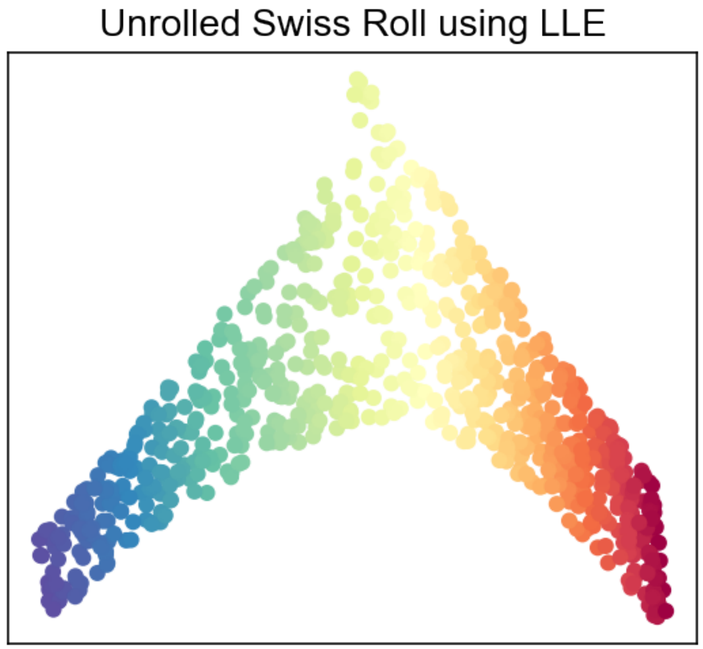
```python
from sklearn.datasets import make_swiss_roll
from sklearn.manifold import LocallyLinearEmbedding

X, color = make_swiss_roll(n_samples=800, random_state=123)
lle = LocallyLinearEmbedding(n_components=2,
                             n_neighbors=10)
X_reduced = lle.fit_transform(X)
```

LocallyLinearEmbedding

```
class sklearn.manifold.LocallyLinearEmbedding(*, n_neighbors=5,
n_components=2, reg=0.001, eigen_solver='auto', tol=1e-06, max_iter=100,
method='standard', hessian_tol=0.0001, modified_tol=1e-12,
neighbors_algorithm='auto', random_state=None, n_jobs=None) #
```

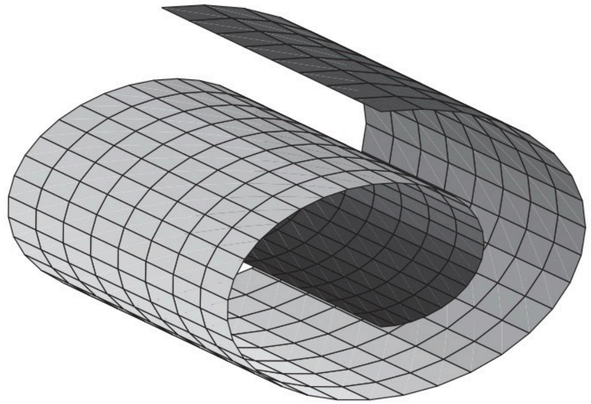
[source]





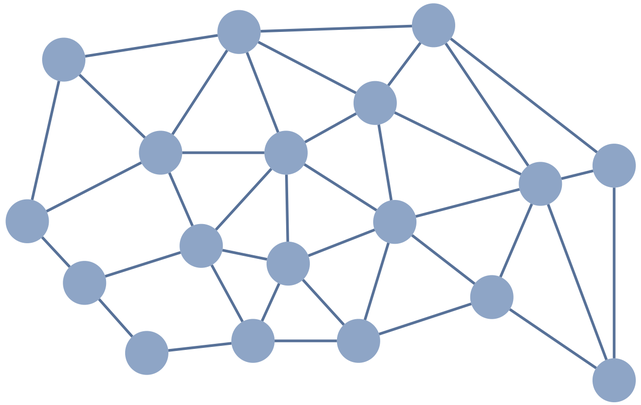
Isometric Feature Mapping (ISOMAP)

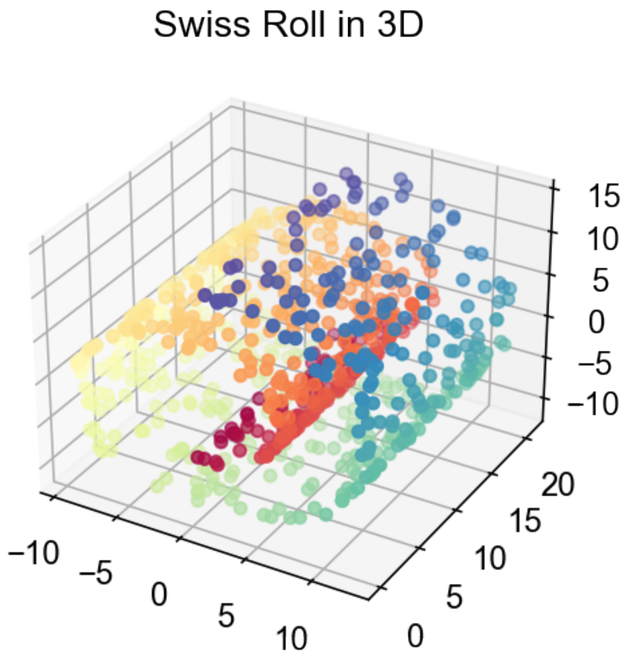
- Extend MDS to handle non-linear relationship in data
- Use ***geodesic distances*** in the original space.

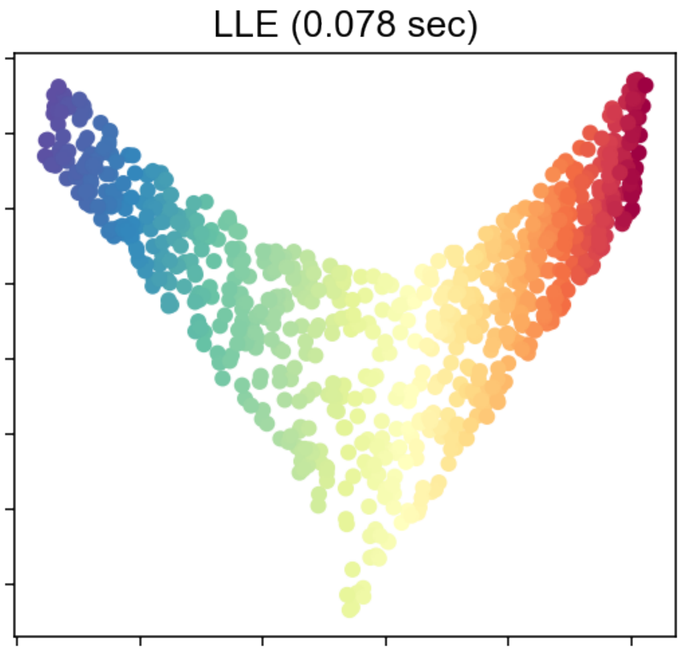


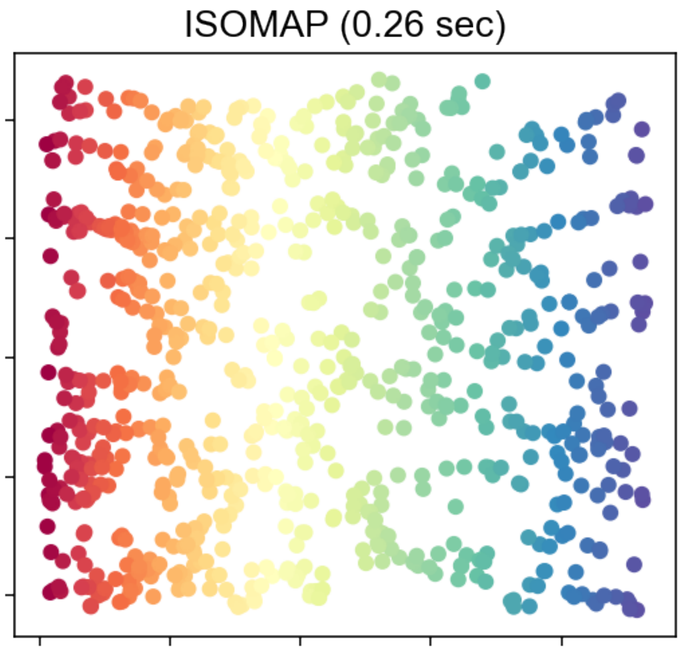
1: Find the nearest neighbors of each data point and create a weighted graph by connecting a point to its nearest neighbors. The nodes are the data points and the weights of the links are the distances between points.
2: Redefine the distances between points to be the length of the shortest path between the two points in the neighborhood graph.
3: Apply classical MDS to the new distance matrix.

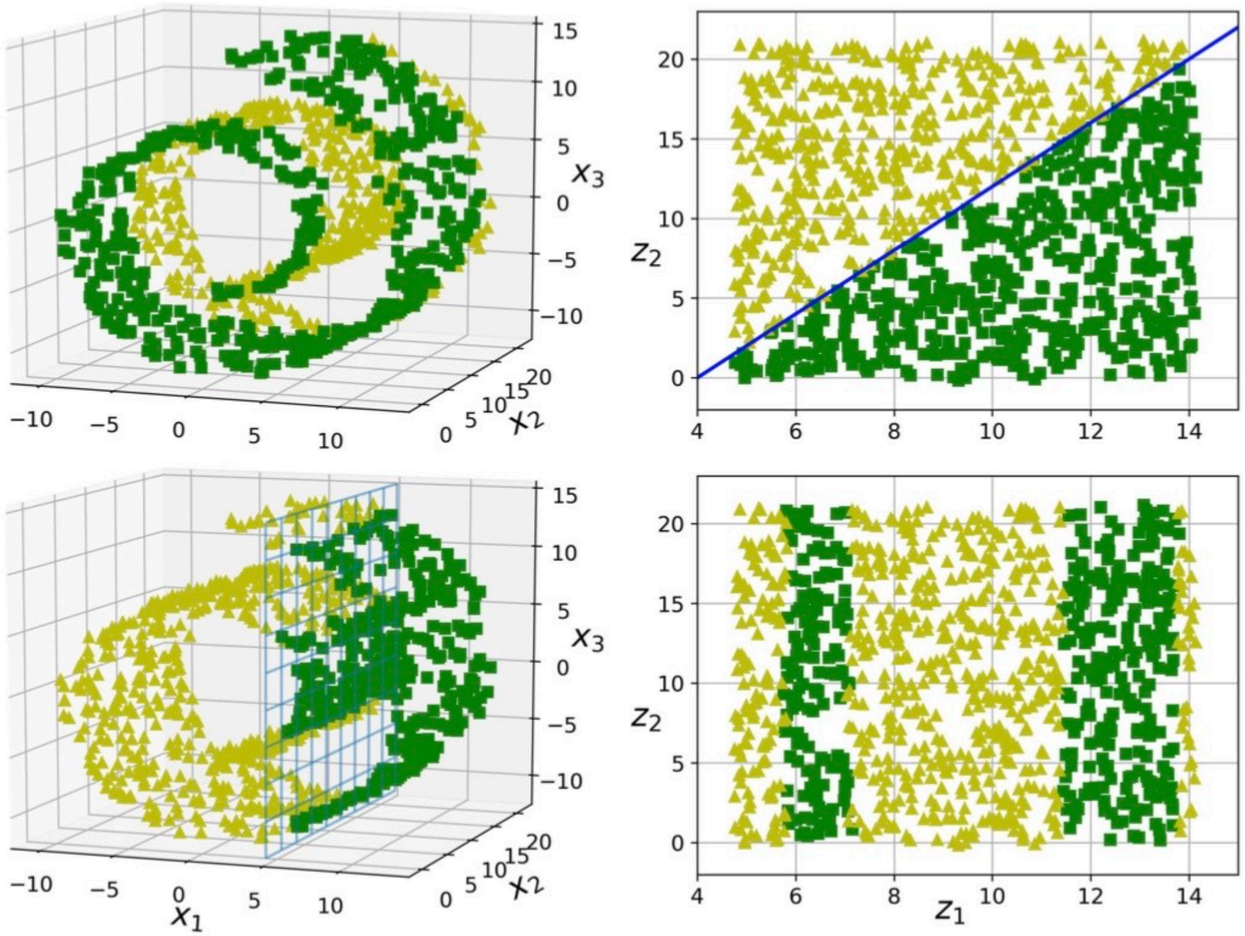
Neighborhood graph




Swiss Roll in 3D
15
10
5
0
−5
−10
20
15
10
5
0
−10
−5
0
5
10


LLE (0.078 sec)


ISOMAP (0.26 sec)



Simplified class separation

More complex class separation

Summary

- ☐ Dimensionality reduction transforms the dataset into lower dimensions to capture most information of the data.
- ☐ PCA projects data into uncorrelated axes with most variances first.
- ☐ Manifold learning maps the data into a lower dimension that preserves the distance relationship among points in the original space.
  - Linear manifold learning uses direct pairwise distances in original space.
  - Non-linear manifold learning uses geodesic distance or gives more weights to closer neighbors in original space.

PCA Lab

(Group of 4 - 6)