Principle Component Analysis
For Dimension Reduction

Prof. Mingkui Tan

SCUT Machine Intelligence Laboratory (SMIL)

Contents

Contents

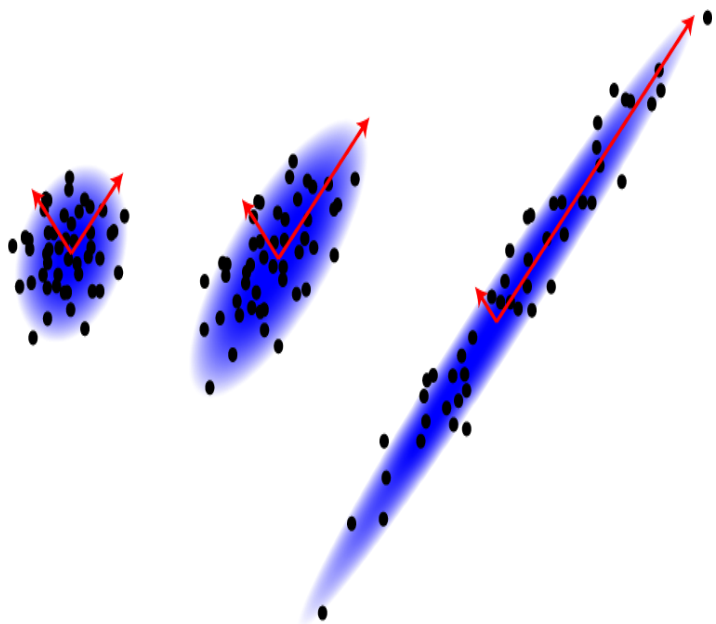
Motivation: Data Redundancy

- Data may contain very similar or even the same columns

Highly Correlated Data!

Curse of Dimensionality for Big Data!

A	B	C	D	E	F	G	H
线性代数	数学分析1	数学分析2	概率论	机器学习	人工智能	离散数学	计算机网络
91	91	89	88	88	84	86	76
73	89	90	66	80	82	90	82
71	62	60	71	60	84	66	63
85	93	85	72	82	83	80	89
78	66	94	69	80	81	86	65
69	73	73	64	90	80	87	90
83	97	96	70	86	85	87	77
95	100	100	97	88	84	88	76
69	68	60	72	76	78	73	79
78	68	84	62	76	80	80	63
84	87	79	73	86	83	81	71
80	91	88	80	81	79	87	72
85	92	87	85	92	86	83	81
71	65	100	75	86	80	86	85
68	79	66	60	71	83	60	84
82	92	81	78	89	81	95	94
96	88	89	76	80	74	87	64
85	82	94	71	88	85	83	82
81	78	91	70	78	79	85	80

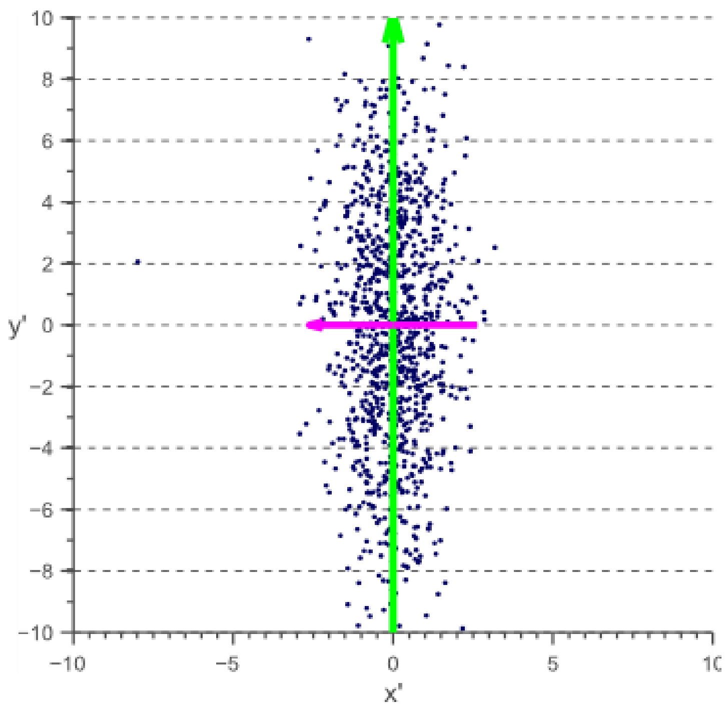


Motivation: Noise

- Some columns are random noises

Highly contaminated!

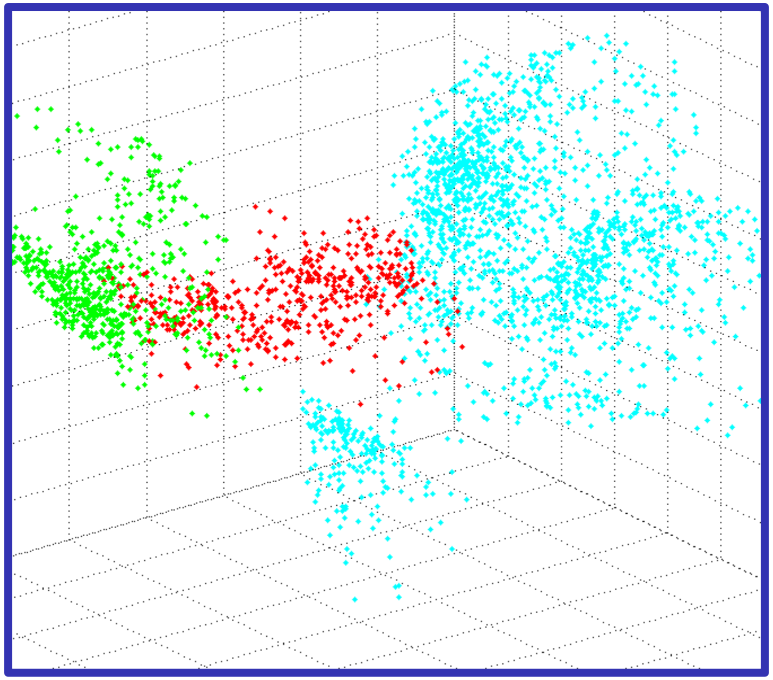
高级语言编程I	高级语言编程II	java程序设计	编程语言训练	大学英语	大学物理 III (2)
72	76	72	78	79	79
84	86	82	88	84	63
61	64	54	80	66	61
87	80	82	82	81	83
74	73	63	87	82	53
73	72	69	85	82	93
81	77	68	78	73	72
93	85	84	90	78	75
73	76	73	82	78	65
67	81	65	84	71	48
84	89	74	84	82	79
84	83	80	85	86	73
91	89	86	94	76	84
72	75	71	80	92	79
79	64	60	0	74	60
90	89	93	100	85	92
88	87	77	89	74	80
80	80	78	88	82	83
80	81	61	88	88	69



Motivation: Data Visualization

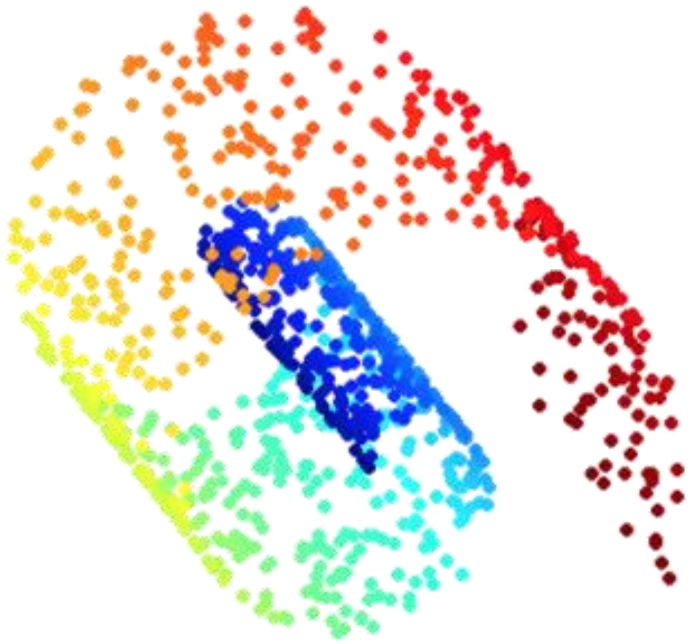
- We are only interested in some useful column

A	B	C	D	E	F	G	H
线性代数	数学分析1	数学分析2	概率论	机器学习	人工智能	离散数学	计算机网络
91	91	89	88	88	84	86	76
73	89	90	66	80	82	90	82
71	62	60	71	60	84	66	63
85	93	85	72	82	83	80	89
78	66	94	69	80	81	86	65
69	73	73	64	90	80	87	90
83	97	96	70	86	85	87	77
95	100	100	97	88	84	88	76
69	68	60	72	76	78	73	79
78	68	84	62	76	80	80	63
84	87	79	73	86	83	81	71
80	91	88	80	81	79	87	72
85	92	87	85	92	86	83	81
71	65	100	75	86	80	86	85
68	79	66	60	71	83	60	84
82	92	81	78	89	81	95	94
96	88	89	76	80	74	87	64
85	82	94	71	88	85	83	82
81	78	91	70	78	79	85	80

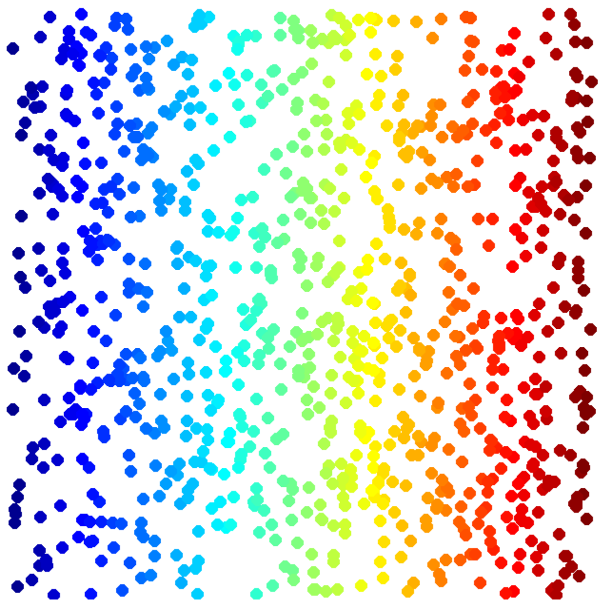


Motivation: Dimension Reduction

High dimension vector → **Low dimension vector**



Looks like 3-D

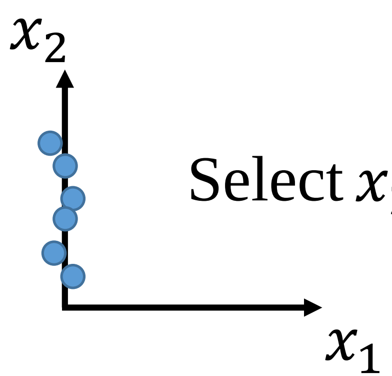


Actually 2-D

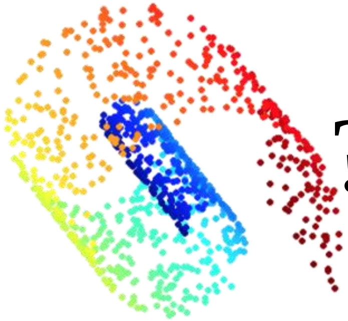
Motivation: Distributed Representation

High dimension vector x → **Low dimension vector z**

- Feature selection



Select x_2



?

- Principle component analysis (PCA) $\mathbf{z} = \mathbf{W}\mathbf{x}$

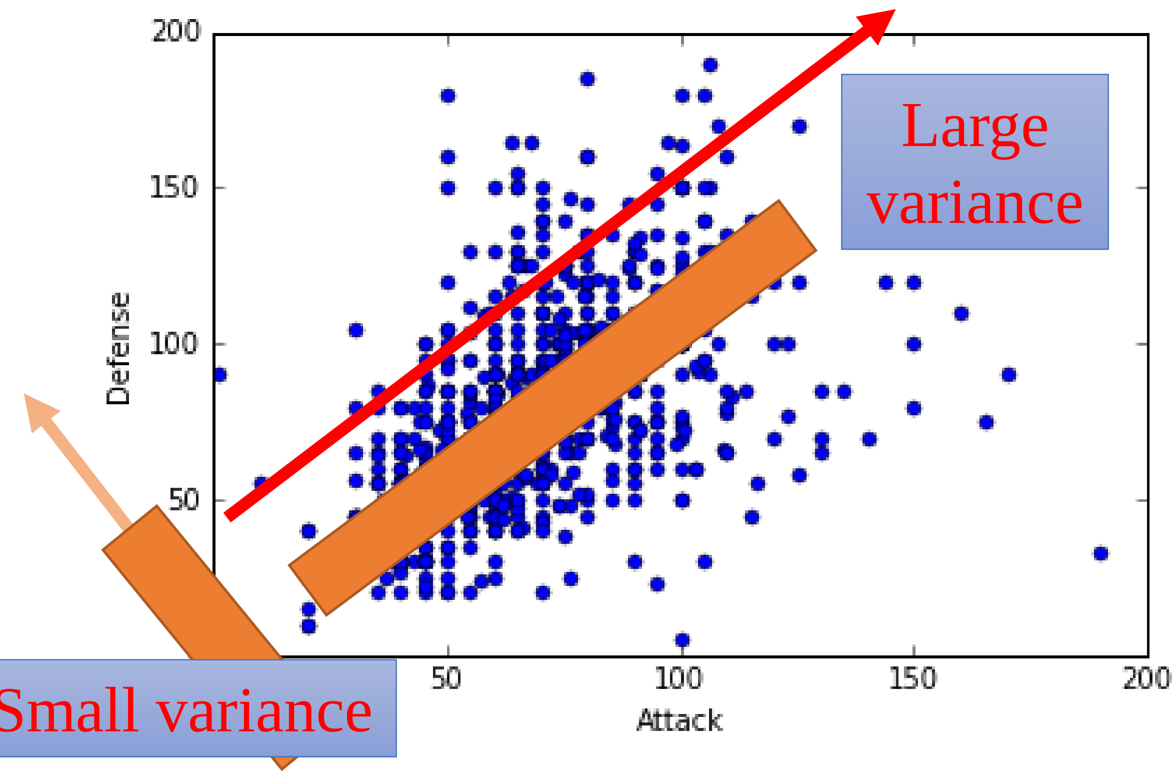
Contents

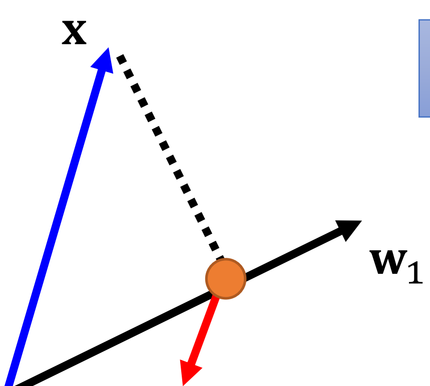
Maximum Variance Formulation

$$\mathbf{z} = \mathbf{W}\mathbf{x}$$

Reduce to 1-D:

$$z_1 = \mathbf{w}_1 \cdot \mathbf{x}$$





$$z_1 = \mathbf{w}_1 \cdot \mathbf{x}$$

Project data $\mathbf{x}$ onto $\mathbf{w}_1$, and obtain $\mathbf{z}_1$

We want the variance of $\mathbf{z}_1$ as large as possible

$$\underset{\mathbf{w}_1}{\operatorname{argmax}}\, var(z_1) = \frac{1}{N}\sum(z_1 - \bar{z}_1)^2$$

$$s.t.\ \|\mathbf{w}_1\|_2 = 1$$

Where N is the number of samples

Maximum Variance Formulation

$\mathbf{z} = \mathbf{W}\mathbf{x}$

Reduce to 1-D:

$z_1 = \mathbf{w}_1 \cdot \mathbf{x}$

$z_2 = \mathbf{w}_2 \cdot \mathbf{x}$

$$\mathbf{W} = \begin{bmatrix} (\mathbf{w}_1)^{\mathrm{T}} \\ (\mathbf{w}_2)^{\mathrm{T}} \\ \vdots \end{bmatrix}$$

Orthogonal matrix

Project data $\mathbf{x}$ onto $\mathbf{w}_1$ and obtain z_1

We want the variance of z_1 as large as possible

$$\underset{\mathbf{w}_1}{\operatorname{argmax}}\, var(z_1) = \frac{1}{N}\sum (z_1 - \bar{z}_1)^2$$

$$s.t.\ \|\mathbf{w}_1\|_2 = 1$$

Project data $\mathbf{x}$ onto $\mathbf{w}_2$ and obtain z_2

We want the variance of z_2 as large as possible

$$\underset{\mathbf{w}_2}{\operatorname{argmax}}\, var(z_2) = \frac{1}{N}\sum (z_2 - \bar{z}_2)^2$$

$$s.t.\ \|\mathbf{w}_2\|_2 = 1 \quad \mathbf{w}_1 \cdot \mathbf{w}_2 = 0$$

Formula Derivation

$$Var(\mathbf{Z}_1) = \frac{1}{N}\sum(z_1 - \bar{z}_1)^2$$

$\mathbf{w}_1 \cdot \mathbf{x}$

$$\bar{z}_1 = \frac{1}{N}\sum z_1 = \frac{1}{N}\sum \mathbf{w}_1 \cdot \mathbf{x}$$

$$= \mathbf{w}_1 \cdot \frac{1}{N}\sum \mathbf{x} = \mathbf{w}_1 \cdot \bar{\mathbf{x}}$$

$$= \frac{1}{N}\sum(\mathbf{w}_1 \cdot \mathbf{x} - \mathbf{w}_1 \cdot \bar{\mathbf{x}})^2$$

$$= \frac{1}{N}\sum(\mathbf{w}_1 \cdot (\mathbf{x} - \bar{\mathbf{x}}))^2$$

$$= \frac{1}{N}\sum((\mathbf{w}_1)^{\mathrm{T}}(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^{\mathrm{T}}\mathbf{w}_1)$$

$$= (\mathbf{w}_1)^{\mathrm{T}}\boxed{\frac{1}{N}\sum(\,(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^{\mathrm{T}})}\mathbf{w}_1$$

Find $\mathbf{w}_1$ maximizing $(\mathbf{w}_1)^{\mathrm{T}}\mathbf{S}\mathbf{w}_1$

where $\|\mathbf{w}_1\|_2^2 = (\mathbf{w}_1)^{\mathrm{T}}\mathbf{w}_1 = 1$

$$= (\mathbf{w}_1)^{\mathrm{T}}\boxed{Cov(\mathbf{X})}\mathbf{w}_1$$

$$= (\mathbf{w}_1)^{\mathrm{T}}\mathbf{S}\mathbf{w}_1$$

$S = Cov(\mathbf{X})$

Formula Derivation

$$\underset{\mathbf{w_1}}{\operatorname{argmax}} (\mathbf{w}_1)^{\mathrm{T}} \mathbf{S} \mathbf{w}_1 \qquad s.t. \quad (\mathbf{w}_1)^{\mathrm{T}} \mathbf{w}_1 = 1$$

$\mathbf{S} = Cov(\mathbf{X})$ Symmetric Positive-semidefinite (non-negative eigenvalues)

Using Lagrange multiplier:

$$g(\mathbf{w}_1) = (\mathbf{w}_1)^{\mathrm{T}} \mathbf{S} \mathbf{w}_1 - \alpha((\mathbf{w}_1)^{\mathrm{T}} \mathbf{w}_1 - 1)$$

$$\left.\begin{aligned} &\partial g(\mathbf{w}_1) / \partial w_{11} = 0 \\ &\partial g(\mathbf{w}_1) / \partial w_{12} = 0 \\ &\vdots \end{aligned}\right\}$$

$$\mathbf{S}\mathbf{w}_1 - \alpha \mathbf{w}_1 = 0$$

$$\mathbf{S}\mathbf{w}_1 = \alpha \mathbf{w}_1$$

$\mathbf{w}_1$: eigenvector

$$(\mathbf{w}_1)^{\mathrm{T}} \mathbf{S} \mathbf{w}_1 = \alpha (\mathbf{w}_1)^{\mathrm{T}} \mathbf{w}_1 = 0$$

Choose the maximum one

$\mathbf{w}_1$ is the eigenvector of the covariance $\mathbf{S}$ matrix, corresponding to the largest eigenvalue λ_1

Formula Derivation

$$\underset{\mathbf{w}_2}{\operatorname{argmax}}(\mathbf{w}_2)^{\mathrm{T}}\mathbf{S}\mathbf{w}_2 \quad s.t. \quad (\mathbf{w}_2)^{\mathrm{T}}\mathbf{w}_2 = 1 \quad (\mathbf{w}_2)^{\mathrm{T}}\mathbf{w}_1 = \mathbf{0}$$

$$g(\mathbf{w}_2) = (\mathbf{w}_2)^{\mathrm{T}}\mathbf{S}\mathbf{w}_2 - \alpha((\mathbf{w}_2)^{\mathrm{T}}\mathbf{w}_2 - 1) - \beta\,((\mathbf{w}_2)^{\mathrm{T}}\mathbf{w}_1 - 0)$$

$$\left.\begin{array}{l} \partial g(\mathbf{w}_2)/\,\partial w_{21} = 0 \\ \partial g(\mathbf{w}_2)/\,\partial w_{22} = 0 \\ \quad\vdots \end{array}\right\} \quad \mathbf{S}\mathbf{w}_2 - \alpha\mathbf{w}_2 - \beta\mathbf{w}_1 = 0$$

$$\underbrace{(\mathbf{w}_1)^{\mathrm{T}}\mathbf{S}\mathbf{w}_2}_{0} - \alpha\underbrace{(\mathbf{w}_1)^{\mathrm{T}}\mathbf{w}_2}_{0} - \beta\underbrace{(\mathbf{w}_1)^{\mathrm{T}}\mathbf{w}_1}_{1} = 0$$

$$= (\mathbf{w}_1)^{\mathrm{T}}\mathbf{S}\mathbf{w}_2 = (\mathbf{w}_2)^{\mathrm{T}}\mathbf{S}\mathbf{w}_1$$

$$\mathbf{S}\mathbf{w}_1 = \lambda_1\mathbf{w}_1$$

$$= \lambda_1(\mathbf{w}_2)^{\mathrm{T}}\mathbf{w}_1 = 0$$

$$\beta = 0: \Longrightarrow \mathbf{S}\mathbf{w}_2 - \alpha\mathbf{w}_2 = 0 \Longrightarrow \mathbf{S}\mathbf{w}_2 = \alpha\mathbf{w}_2$$

$\mathbf{w}_2$ is the eigenvector of the covariance matrix $\mathbf{S}$, corresponding to the 2nd largest eigenvalue λ_2

How to Reduce Dimension?

To reduce dimension of data $\mathbf{X}$ from d to k ($k < d$), we perform:

- **Step1:** Calculate the covariance matrix $\mathbf{S} = Cov(\mathbf{X})$
- **Step2:** Select a set of orthonormal eigenvectors corresponding to the k largest eigenvalues, resulting in the projection matrix $\mathbf{W}$
- **Step3:** Reduce the dimension by calculating:

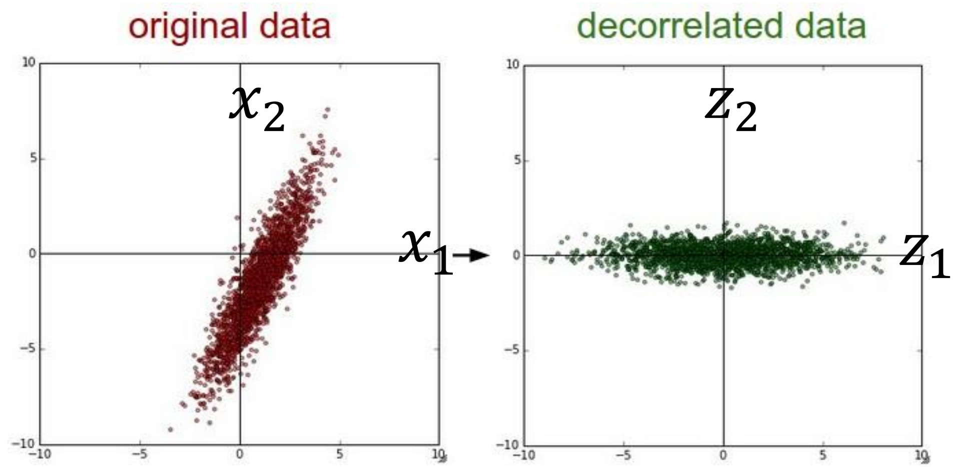
$$\mathbf{Z} = \mathbf{W}\mathbf{X} = \begin{bmatrix} (\mathbf{w}_1)^{\mathrm{T}} \\ (\mathbf{w}_2)^{\mathrm{T}} \\ \vdots \\ (\mathbf{w}_k)^{\mathrm{T}} \end{bmatrix} \mathbf{X}$$

Example: Decorrelation

$\mathbf{Z} = \mathbf{WX}$

$Cov(\mathbf{Z}) = D$

Diagonal matrix



$$Cov(\mathbf{Z}) = \frac{1}{n}\sum(\mathbf{z}-\bar{\mathbf{z}})(\mathbf{z}-\bar{\mathbf{z}})^{\mathrm{T}} = \mathbf{W}\mathbf{S}\mathbf{W}^{\mathrm{T}}$$

$Cov(\mathbf{X})$

$$= \mathbf{W}[\mathbf{S}\mathbf{w}_1 \ \cdots \mathbf{S}\mathbf{w}_k]$$

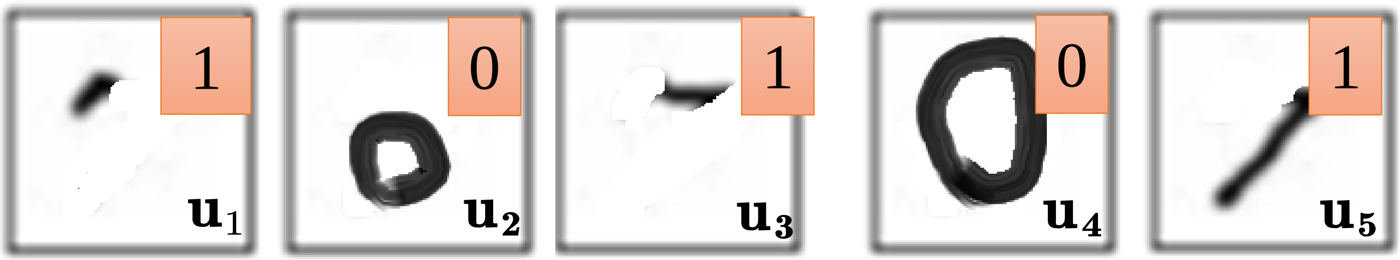
$$= \mathbf{W}[\ \lambda_1\mathbf{S}\mathbf{w}_1 \ \cdots \ \lambda_k\mathbf{S}\mathbf{w}_k]$$

$$= [\ \lambda_1\mathbf{e}_1 \ \cdots \lambda_k\,\mathbf{e}_k] = D$$

Diagonal matrix

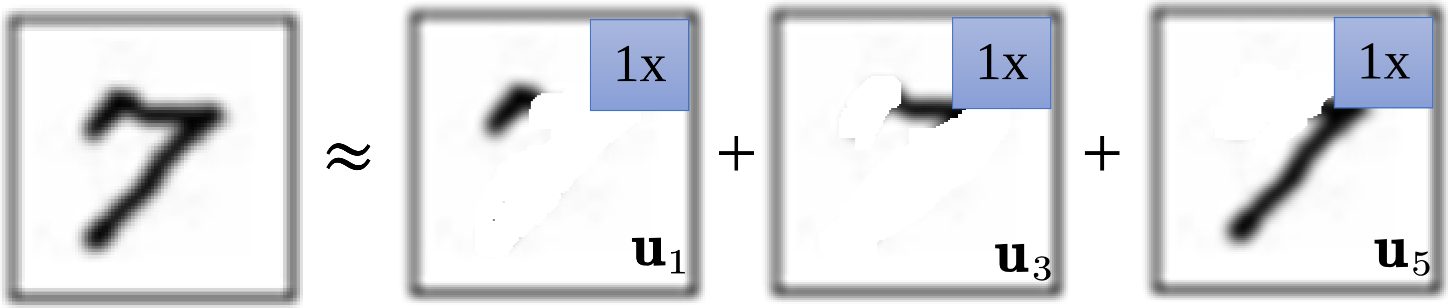
Minimum Error Formulation

Basic Component:



....

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



$$\mathbf{x} \approx c_1 \mathbf{u}_1 + c_2 \mathbf{u}_2 + \cdots + c_k \mathbf{u}_k + \tilde{\mathbf{x}}$$

Pixels in a digit image

component

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix}$$ Represent a digit image

Minimum Error Formulation

$$\mathbf{x}-\tilde{\mathbf{x}} \approx c_1\mathbf{u}_1 + c_2\mathbf{u}_2 + \cdots + c_k\,\mathbf{u}_k = \hat{\mathbf{x}}$$

Find $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ to minimize the following reconstruction error:

$$\boldsymbol{L} = \underset{\{\mathbf{u}_1,\mathbf{u}_2,\cdots,\mathbf{u}_k\}}{\operatorname{argmax}} \left\| (\mathbf{x}-\tilde{\mathbf{x}}) - \underbrace{\textstyle\sum_{k=1}^{K} c_k\mathbf{u}_k}_{\hat{\mathbf{x}}} \right\|_2$$

PCA: $\mathbf{z} = \mathbf{W}\mathbf{x}$

$$\mathbf{z} = \begin{bmatrix} (\mathbf{w}_1)^{\mathrm{T}} \\ (\mathbf{w}_2)^{\mathrm{T}} \\ \vdots \\ (\mathbf{w}_k)^{\mathrm{T}} \end{bmatrix} \mathbf{x}$$

$\{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k\}$ (from PCA) is the component $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\}$ (minimizing $\boldsymbol{L}$)

Proof in [Bishop, Chapter 12.1.2]

Minimum Error Formulation

$$\mathbf{x} - \tilde{\mathbf{x}} \approx c_1\mathbf{u}_1 + c_2\mathbf{u}_2 + \cdots + c_k\,\mathbf{u}_k = \hat{\mathbf{x}}$$

Find $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ to minimize the following reconstruction error:

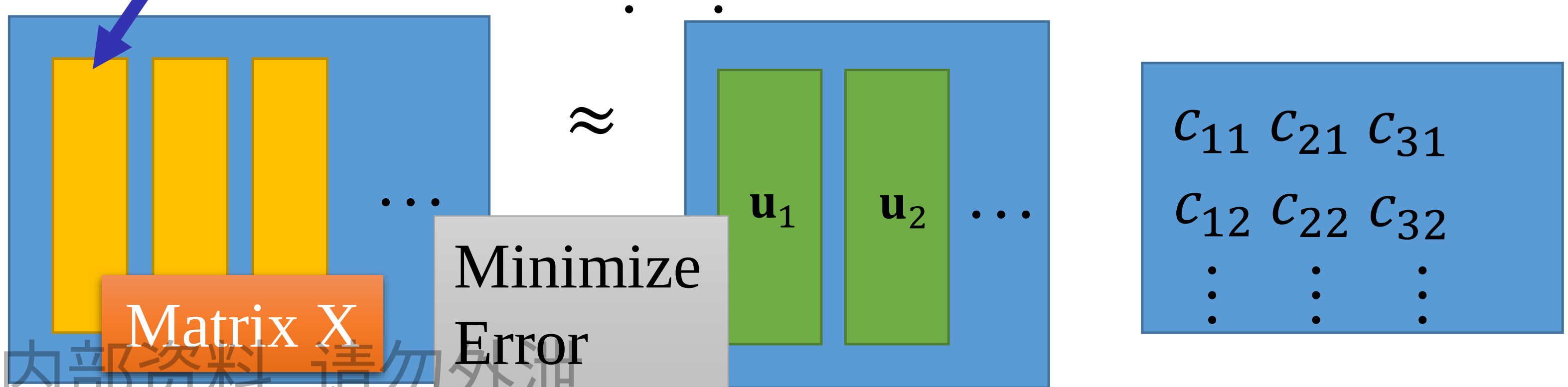
$$\|(\mathbf{x} - \tilde{\mathbf{x}}) - \hat{\mathbf{x}}\|_2$$

$$\mathbf{x}_1 - \tilde{\mathbf{x}} \approx c_{11}\mathbf{u}_1 + c_{12}\mathbf{u}_2 + \cdots$$

$$\mathbf{x}_2 - \tilde{\mathbf{x}} \approx c_{21}\mathbf{u}_1 + c_{22}\mathbf{u}_2 + \cdots$$

$$\mathbf{x}_3 - \tilde{\mathbf{x}} \approx c_{31}\mathbf{u}_1 + c_{32}\mathbf{u}_2 + \cdots$$

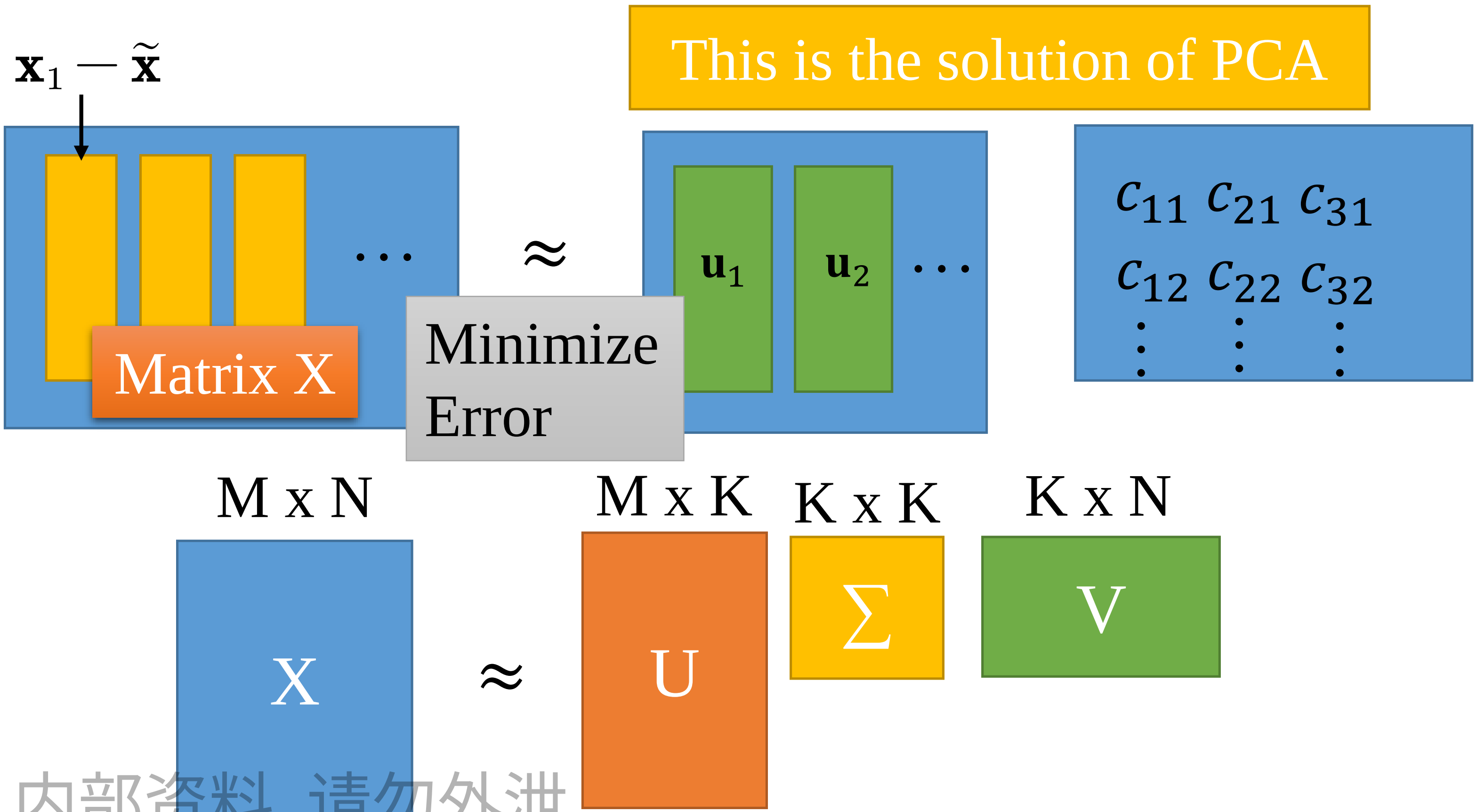
$$\vdots \quad \vdots$$



Minimum Error Formulation

- **K columns of U:**

a set of orthonormal eigenvectors corresponding to the K largest eigenvalues of $\mathbf{X}\mathbf{X}^T$



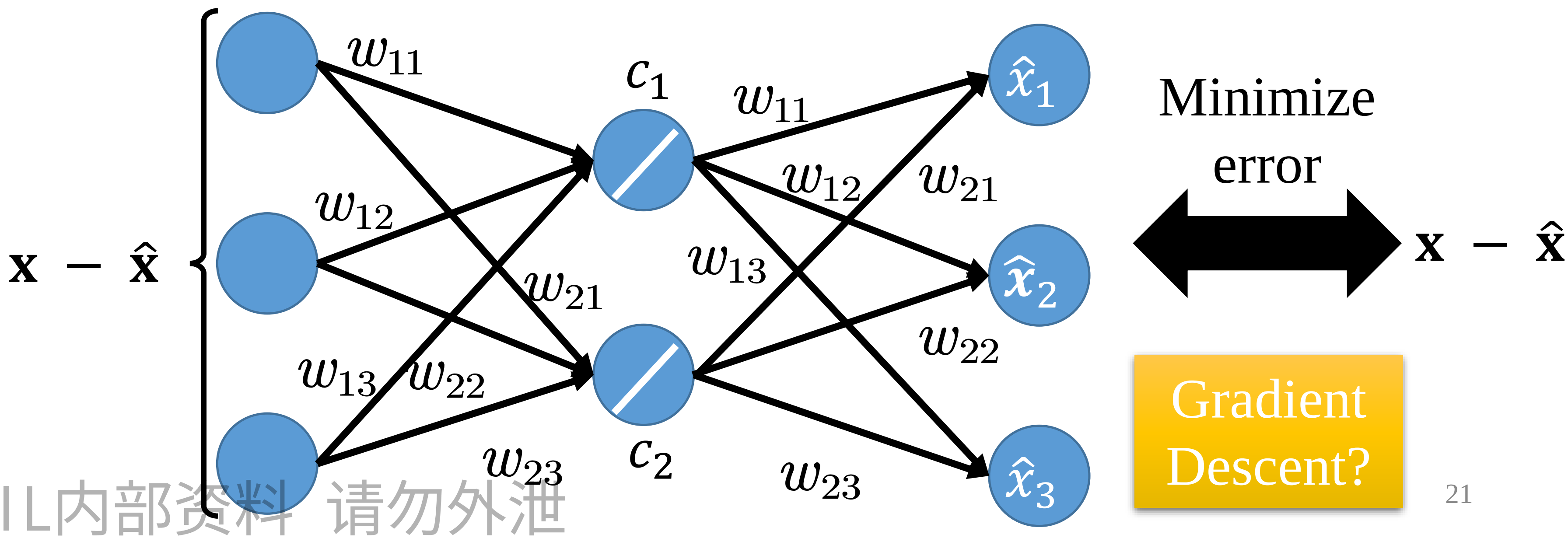
Autoencoder

PCA looks like a neural network with one hidden layer (linear activation function)

If $\{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k\}$ is the component $\{\mathbf{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_k\}$, then we have

$$\hat{\mathbf{x}} = \sum_{k=1}^{K} c_k \mathbf{w}_k \Leftrightarrow \mathbf{x} - \hat{\mathbf{x}}$$

For the case where $K = 2$:



Contents

Example: Pokemon

- Inspired from: https://www.kaggle.com/strakul5/d/abcsds/pokemon/principal-component-analysis-of-pokemon-data
- 800 Pokemons with 6 features:

  HP, Atk, Def, Sp Atk, Sp Def, Speed
- How many principle components?

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6}$$

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
ratio	0.45	0.18	0.13	0.12	0.07	0.04

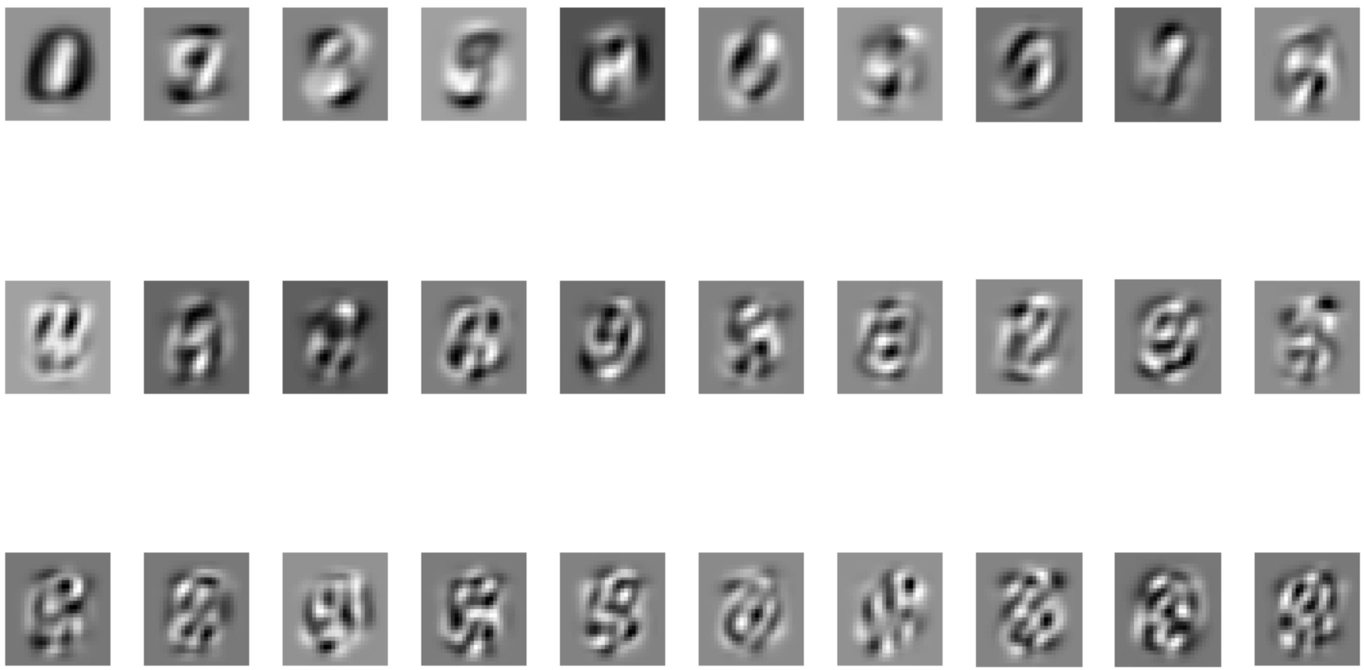
Using 4 components is good enough

Example: MNIST

$$= \mathbf{a}_1 \mathbf{w}_1 + \mathbf{a}_2 \mathbf{w}_2 + \cdots$$

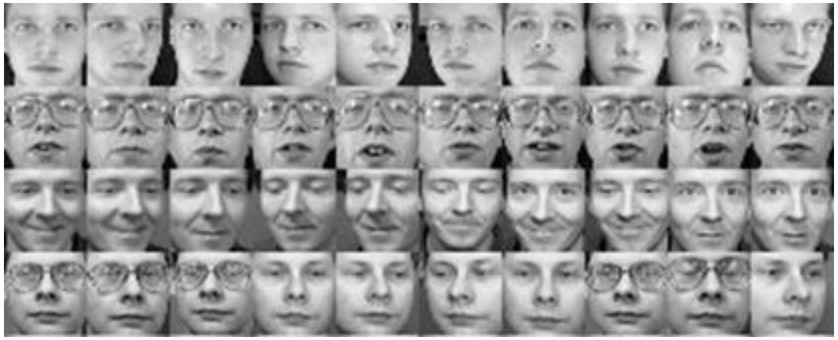
images

30 components:



Example: Face

Eigen-face



30 components:

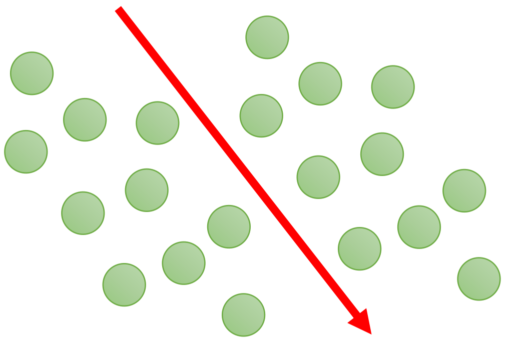
http://www.cs.unc.edu/~lazebnik/research/spring08/assignment3.html

Contents

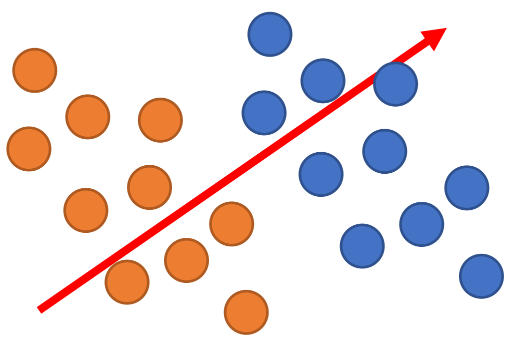
Conclusion

- Unsupervised

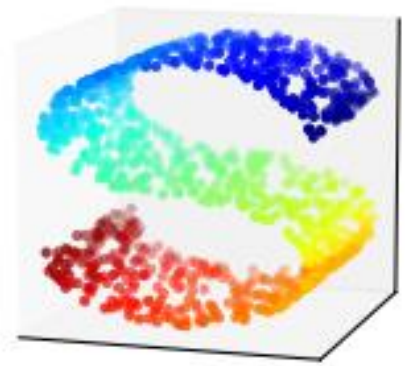
PCA

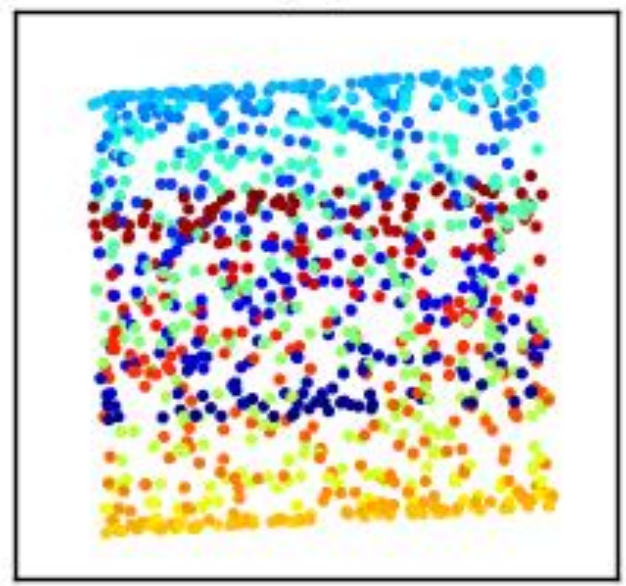


LDA



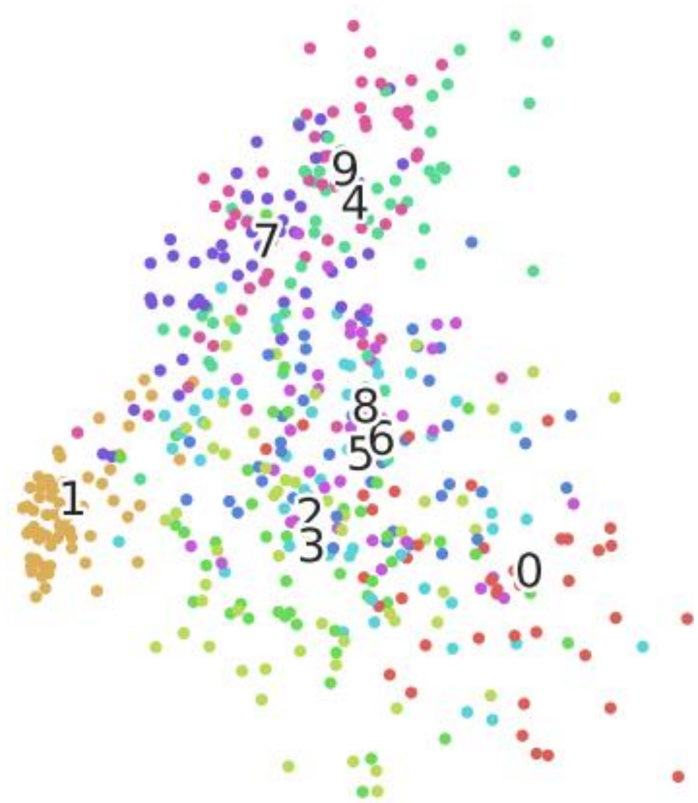
- Linear



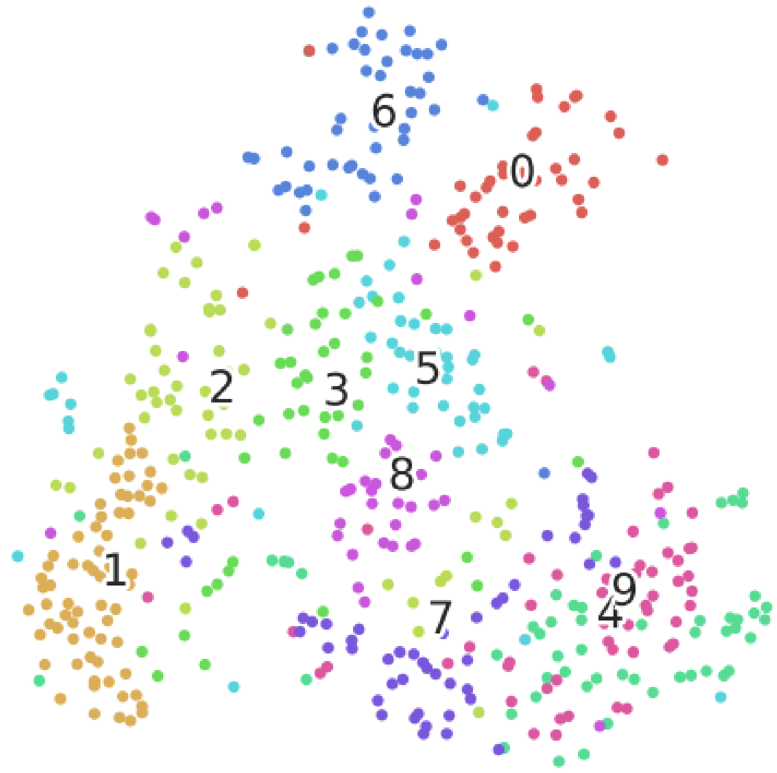


http://www.astroml.org/book_figures/chapter7/fig_S_manifold_PCA.html

Conclusion



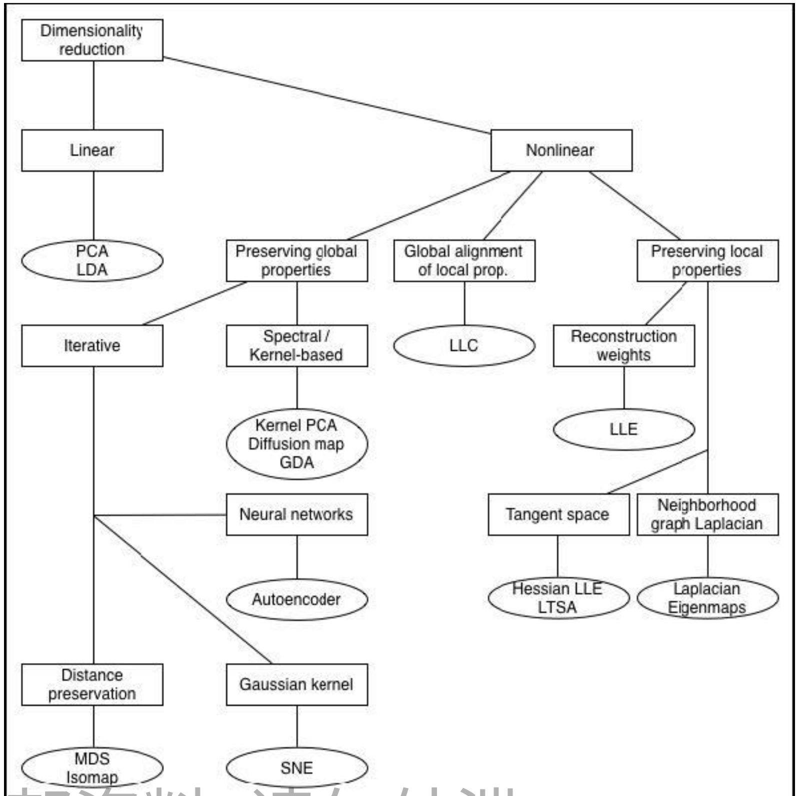
Pixel (28x28) -> PCA (2)



Pixel (28x28) -> tSNE (2)

Appendix

- http://4.bp.blogspot.com/_sHcZHRnxlLE/S9EpFXYjfvI/AAAAAAAABZ0/_oEQiaR3WVM/s640/dimensionality+reduction.jpg
- https://lvdmaaten.github.io/publications/papers/TR_Dimensionality_Reduction_Review_2009.pdf



Thank You