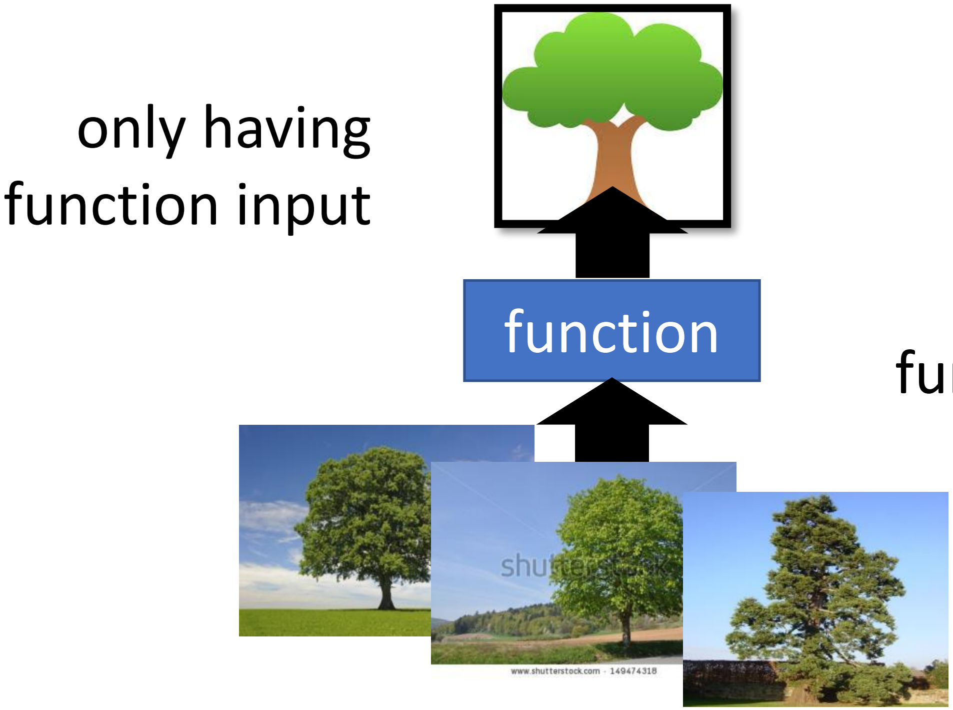
Unsupervised Learning: Principle Component Analysis

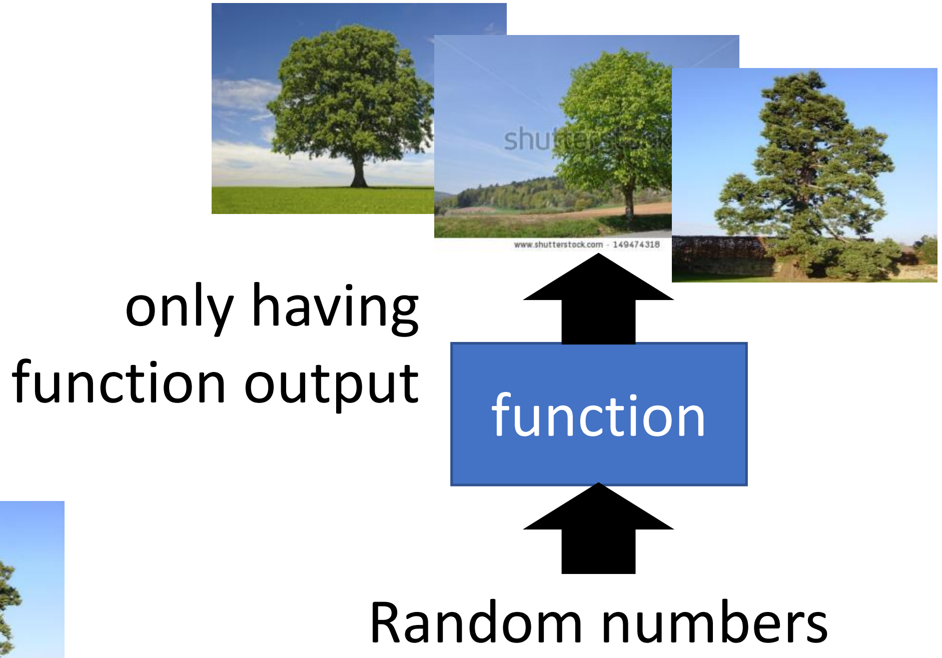
Prof. Wu, Pei-Yuan
Prof. Lee, Hung-Yi
National Taiwan University
Electrical Engineering Department

Unsupervised Learning

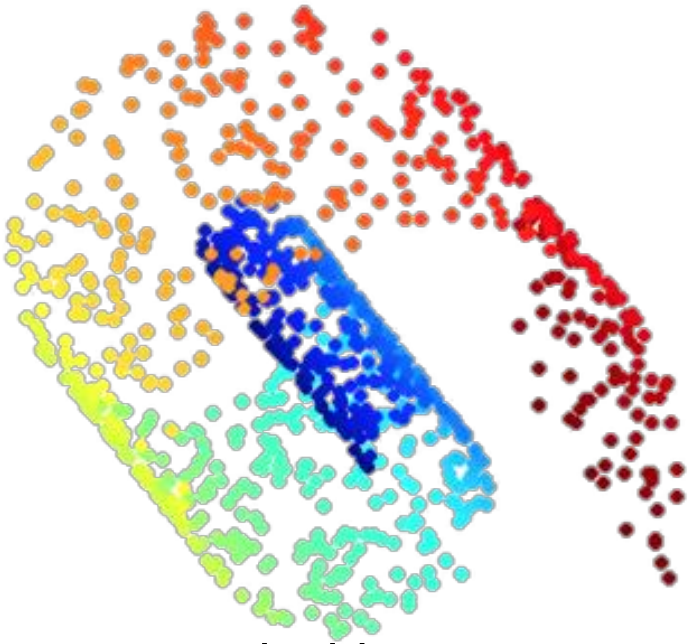
- Dimension Reduction (化繁為簡)



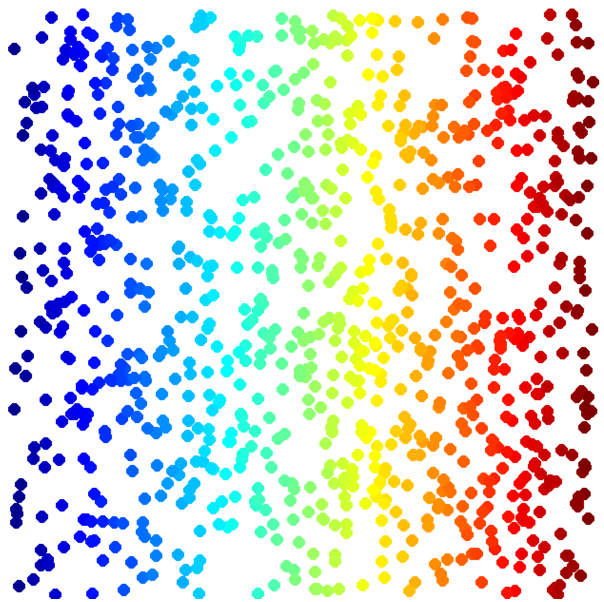
- Generation (無中生有)



Dimension Reduction

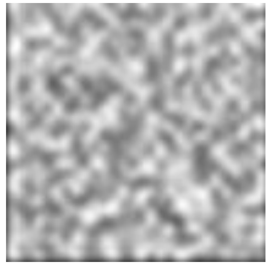


Looks like 3-D

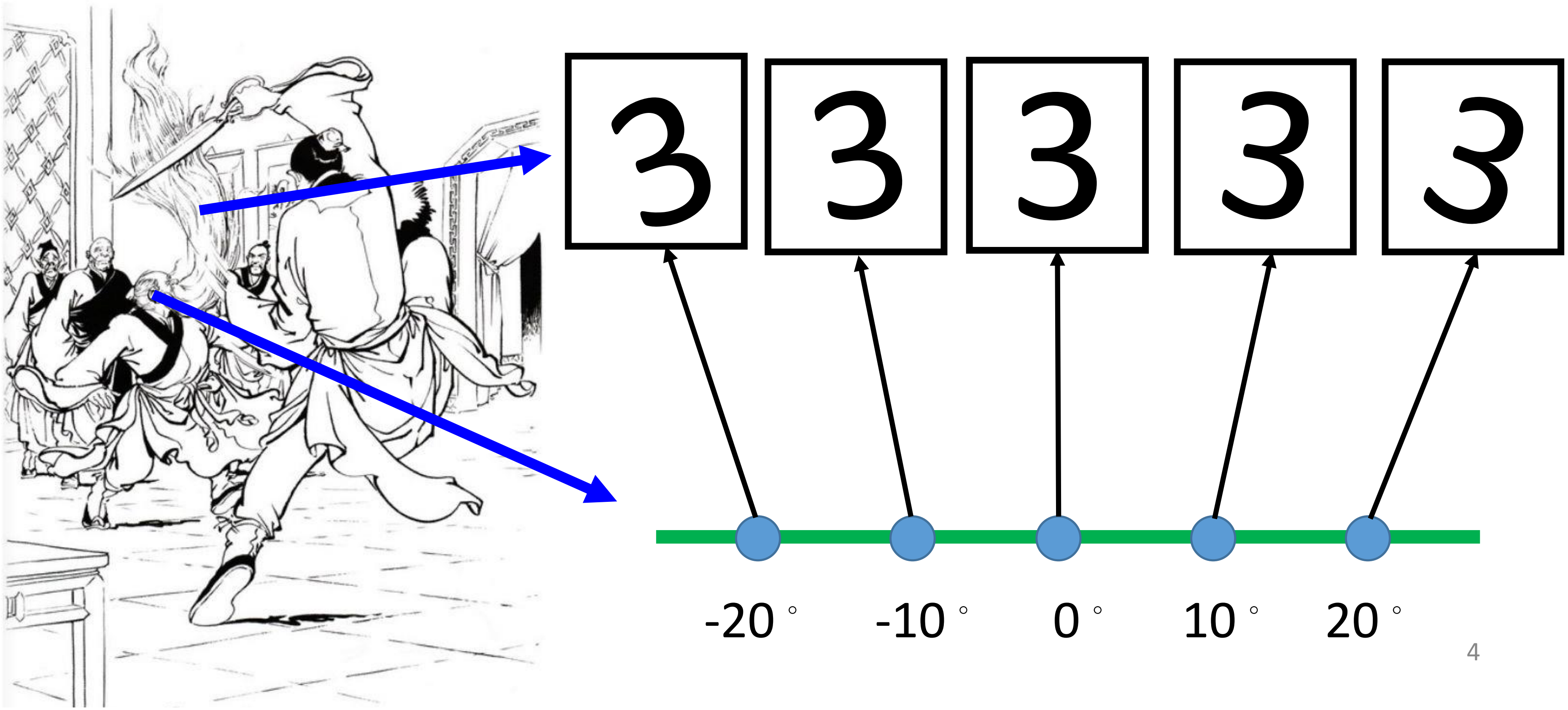


Actually, 2-D

Dimension Reduction



- In MNIST, a digit is 28 x 28 dims.
    - Most 28 x 28 dim vectors are not digits



Clustering

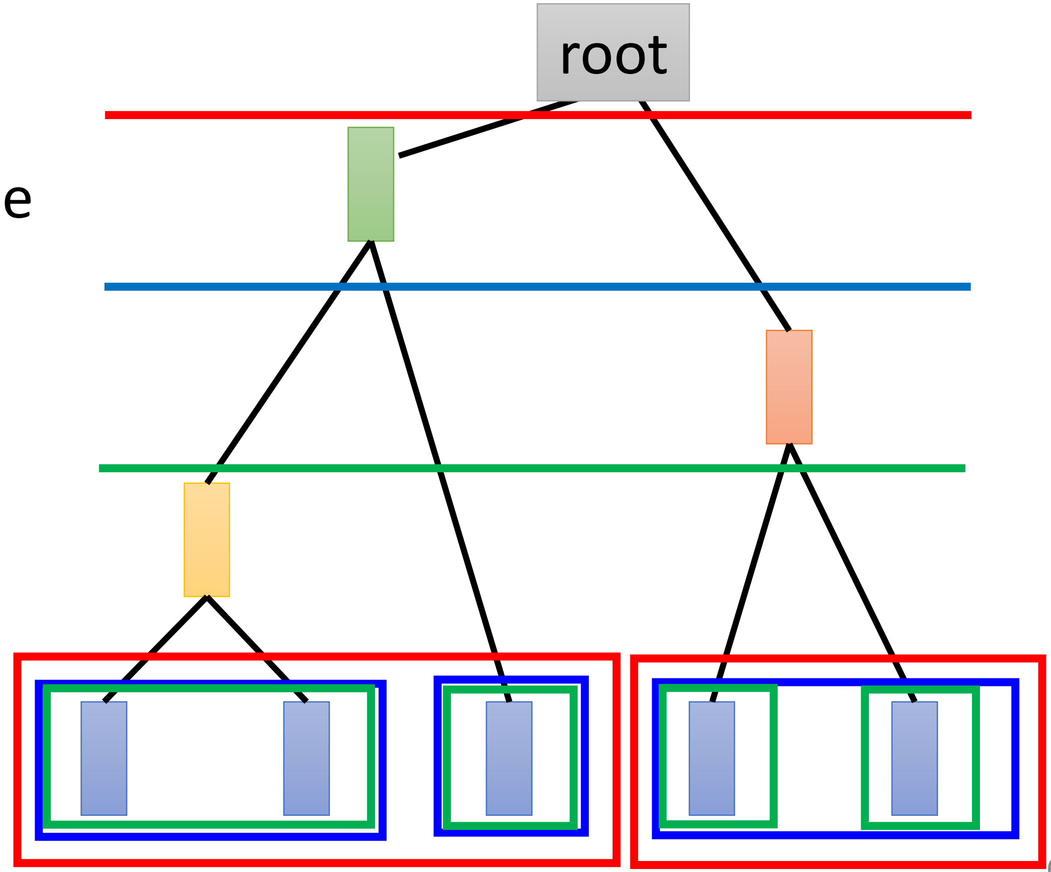
- K-means
  - Clustering $X = \{x^1, \cdots, x^n, \cdots, x^N\}$ into K clusters
  - Initialize cluster center c^i, i=1,2, … K (K random x^n from X)
  - Repeat
    - For all x^n in X: $b_i^n \begin{cases} 1 & x^n \text{ is most “}\textbf{\textit{close}}\text{” to } c^i \\ 0 & \text{Otherwise} \end{cases}$
    - Updating all c^i: $c^i = \sum_n b_i^n x^n \Big/ \sum_n b_i^n$

Clustering

- Hierarchical Agglomerative Clustering (HAC)

Step 1: build a tree

Step 2: pick a threshold



Distributed Representation

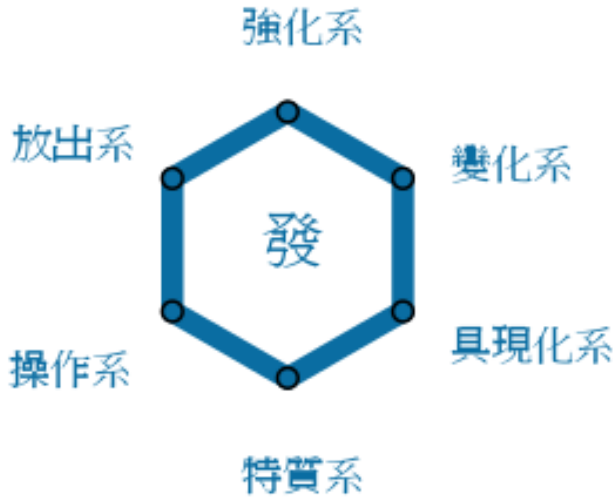
- Clustering: an object must belong to one cluster

小傑是強化系

- Distributed representation

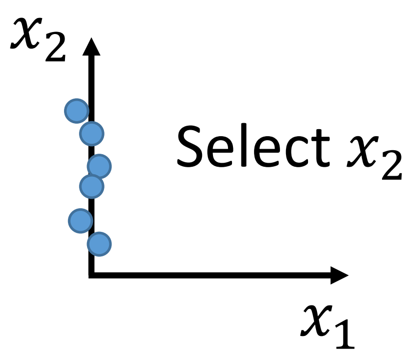
小傑是

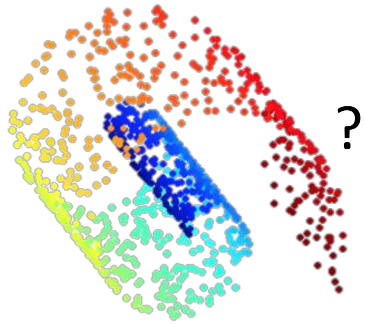
強化系	0.70
放出系	0.25
變化系	0.05
操作系	0.00
具現化系	0.00
特質系	0.00



Distributed Representation

- Feature selection



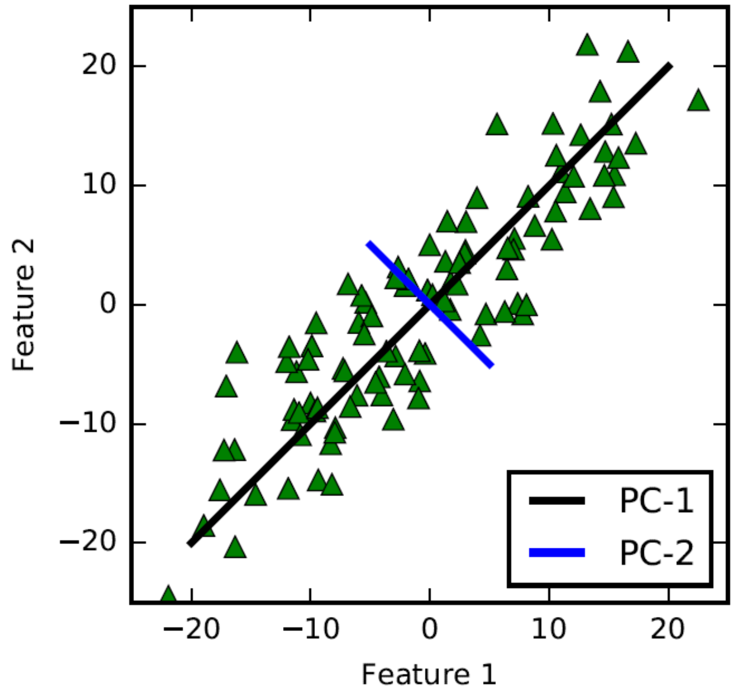


- Principle component analysis (PCA)
  [Bishop, Chapter 12]

$$z = Wx$$

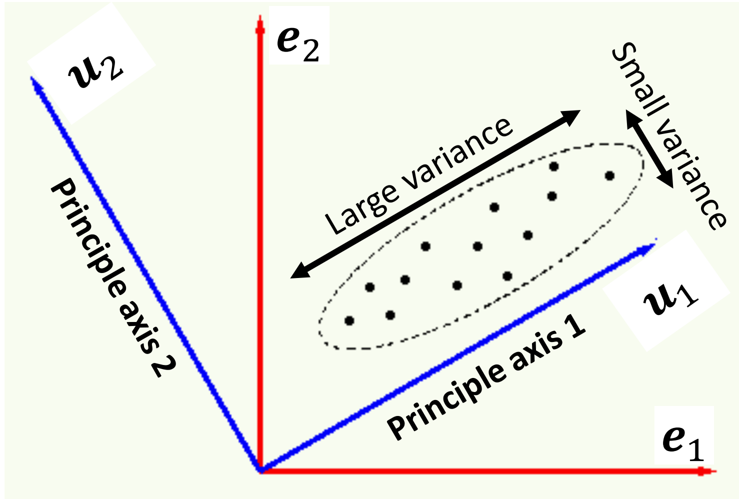
Principal Component Analysis (PCA)

- PCA's target: finding the best lower dimensional sub-space that conveys most of the variance in the original data
- Example: If we were to compress 2-D data to 1-D subspace, then PCA prefers projecting to the ***black*** line, since it preserves more variance comparing to ***blue*** line.



Principle Axes

- Objective of PCA: Given data in $\mathbb{R}^M$, want to *rigidly rotate* the axes to new positions (principle axes) with the following properties:
  - ➢ *Ordered such that principle axis 1 has the highest variance, axis 2 has the next highest variance, …, and axis M has the lowest variance.*
  - ➢ *Covariance among each pair of the principal axes is zero.*
- The k'th ***principle component*** is the projection to the k'th principle axis.
- Keep the first $m < M$ principle components for dimensionality reduction.



Principle Component Computation

- Given N data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathbb{R}^M$, PCA first computes the covariance matrix for the data

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T = \boldsymbol{U\Lambda U}^T$$

where $\boldsymbol{\mu} \in \mathbb{R}^M$ is the data mean.

- Since $\boldsymbol{\Sigma}$ is symmetric, $\boldsymbol{\Sigma}$ can be written as $\boldsymbol{\Sigma} = \boldsymbol{U\Lambda U}^T$, where $\boldsymbol{U} = [\boldsymbol{u}_1 \dots \boldsymbol{u}_{\mathrm{M}}]$ is ***orthogonal*** matrix of eigenvectors (of $\boldsymbol{\Sigma}$), $\boldsymbol{\Lambda} = diag(\lambda_1, \dots, \lambda_M)$ is diagonal matrix of the associated eigenvalues arranged in non-ascending order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M \geq 0$. (Note that all eigenvalues are non-negative real scalars since $\boldsymbol{\Sigma}$ is ***semi-positive definite***.)
- For data $\boldsymbol{x} \in \mathbb{R}^M$, compute its 1st principle component as $\boldsymbol{u}_1^T\boldsymbol{x}$, 2nd principle component as $\boldsymbol{u}_2^T\boldsymbol{x}$,…, M'th principle component as $\boldsymbol{u}_M^T\boldsymbol{x}$

Orthogonal matrix:

$\boldsymbol{U} = [\boldsymbol{u}_1 \dots \boldsymbol{u}_{\mathrm{M}}] \in \mathbb{R}^{M\times M}$ is an orthogonal matrix if $\boldsymbol{u}_1, \dots, \boldsymbol{u}_{\mathrm{M}}$ are orthogonal and have unit length

$$\boldsymbol{u}_{\mathrm{i}}^T\boldsymbol{u}_{\mathrm{j}} = \begin{cases} 1 & \text{if i} = j \\ 0 & \text{if } i \neq \text{j} \end{cases}$$

That is, $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, namely, $\boldsymbol{U}^{-1} = \boldsymbol{U}^T$.

Positive definite:

$\boldsymbol{\Sigma} \in \mathbb{R}^{M\times M}$ is positive semi-definite if $\boldsymbol{x}^T\boldsymbol{\Sigma x} \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^M$. If the equality holds only when $\boldsymbol{x} = \boldsymbol{0}$, then $\boldsymbol{\Sigma}$ is positive definite.

Principle Components are Uncorrelated

- The covariance of the k'th and ℓ'th principle components of data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ is

$$\frac{1}{N}\sum_{i=1}^{N}\left[\boldsymbol{u}_k^T(\boldsymbol{x}_i - \boldsymbol{\mu})\right]\left[\boldsymbol{u}_\ell^T(\boldsymbol{x}_i - \boldsymbol{\mu})\right] = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{u}_k^T(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T\boldsymbol{u}_\ell$$

$$= \boldsymbol{u}_k^T\boldsymbol{\Sigma}\boldsymbol{u}_\ell = \boldsymbol{u}_k^T\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{u}_\ell = \boldsymbol{e}_k^T\boldsymbol{\Lambda}\boldsymbol{e}_\ell = \begin{cases}\lambda_k & \text{if k} = \ell \\ 0 & \text{if k} \neq \ell\end{cases}$$

Therefore

- The variance of the k'th principle components is λ_k.

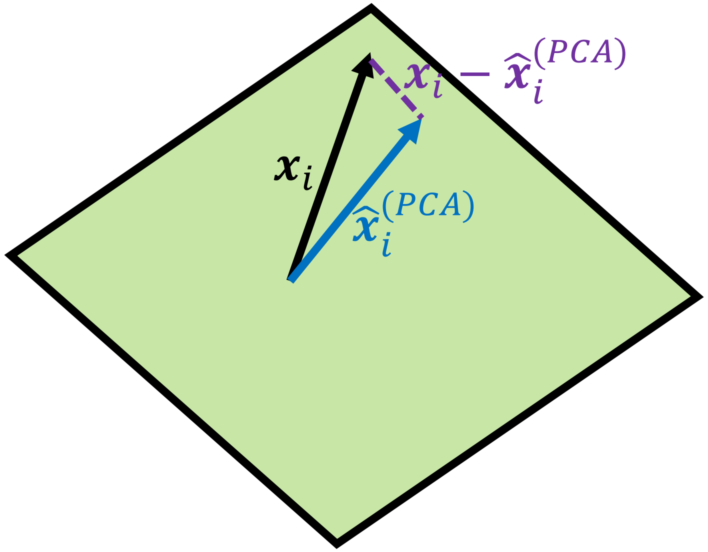
  ⇒ *principle axis 1 has the highest variance, axis 2 has the next highest variance, …, and axis M has the lowest variance.*

- The covariance of different principle components is zero.
- ⇒ *Covariance among each pair of the principal axes is zero.*

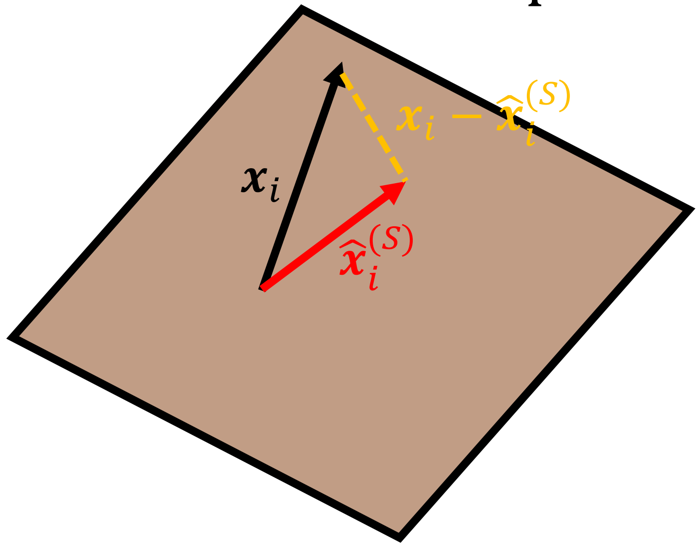
PCA and Reconstruction Error

WLOG assume zero mean $\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i = \mathbf{0}$

$S_{PCA} = Span(\boldsymbol{u}_1, \dots, \boldsymbol{u}_m)$



S: Arbitrary m-dimensional subspace



Variance after projection:

$$\sum_{i=1}^{N} \left\| \widehat{\boldsymbol{x}}_i^{(PCA)} \right\|^2 \geq \sum_{i=1}^{N} \left\| \widehat{\boldsymbol{x}}_i^{(S)} \right\|^2$$

Mean square error after projection:

$$\sum_{i=1}^{N} \left\| \boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i^{(PCA)} \right\|^2 \leq \sum_{i=1}^{N} \left\| \boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i^{(S)} \right\|^2$$

Projecting to S_{PCA} yields the minimum mean squared error among all possible m-dimensional subspaces. ***Why???***

Low Rank Approximation

Eckart-Young-Mirsky Theorem:
Let $\boldsymbol{X} \in \mathbb{R}^{M\times N}$ be a matrix with singular value decomposition $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, where $\boldsymbol{U} \in \mathbb{R}^{M\times M}, \boldsymbol{V} \in \mathbb{R}^{N\times N}$ are orthogonal matrices of left- and right-eigenvectors (of $\boldsymbol{X}$), and $\boldsymbol{D} \in \mathbb{R}^{M\times N}$ is a diagonal matrix of singular values $\sigma_i = D_{ii}$, arranged by their magnitude

$$|\sigma_1| \geq |\sigma_2| \geq \cdots \geq \left|\sigma_{\min(M,N)}\right|$$

Let $m \leq \min(\mathrm{M},\mathrm{N})$, then both low rank approximation problems

$$\min_{\widehat{\boldsymbol{X}}} \left\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\right\|_2 \qquad \text{subject to } rank(\widehat{\boldsymbol{X}}) \leq m$$

$$\min_{\widehat{\boldsymbol{X}}} \left\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\right\|_F \qquad \text{subject to } rank(\widehat{\boldsymbol{X}}) \leq m$$

Has optimal solution $\widehat{\boldsymbol{X}} = \sum_{i=1}^{m} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$. Here $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ denotes the i’th column in matrices $\boldsymbol{U}, \boldsymbol{V}$, respectively.

WLOG assume zero mean $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i = \boldsymbol{0}$

$$\boldsymbol{X} = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \dots \boldsymbol{x}_N] = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$$

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T = \frac{1}{N}\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{D}^T\boldsymbol{U}^T = \boldsymbol{U}\left(\frac{1}{N}\boldsymbol{D}\boldsymbol{D}^T\right)\boldsymbol{U}^T$$

$$\boldsymbol{\Lambda} = diag(\lambda_1, \dots, \lambda_M) = \frac{1}{N}\boldsymbol{D}\boldsymbol{D}^T = diag\left(\frac{\sigma_1^2}{N}, \dots, \frac{\sigma_{\min(M,N)}^2}{N}, 0, \dots, 0\right)$$

$|\sigma_1| \geq |\sigma_2| \geq \cdots$ implies $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$

Projection by PCA: $\hat{\boldsymbol{x}}_n^{(PCA)} = \sum_{i=1}^{m} \boldsymbol{u}_i\boldsymbol{u}_i^T\boldsymbol{x}_n$

$$\hat{\boldsymbol{X}}^{(PCA)} = \left[\hat{\boldsymbol{x}}_1^{(PCA)}\hat{\boldsymbol{x}}_2^{(PCA)} \dots \hat{\boldsymbol{x}}_N^{(PCA)}\right] = \sum_{i=1}^{m}\boldsymbol{u}_i\boldsymbol{u}_i^T\boldsymbol{X} = \sum_{i=1}^{m}\boldsymbol{u}_i\boldsymbol{u}_i^T\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T = \sum_{i=1}^{m}\sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^T$$

Projection to S: $\hat{\boldsymbol{x}}_n^{(S)} \in \boldsymbol{S}$

$$\hat{\boldsymbol{X}}^{(S)} = \left[\hat{\boldsymbol{x}}_1^{(S)}\ \hat{\boldsymbol{x}}_2^{(S)} \dots \hat{\boldsymbol{x}}_N^{(S)}\right] \Rightarrow \boldsymbol{rank}\left(\hat{\boldsymbol{X}}^{(S)}\right) \leq \boldsymbol{dim}(\boldsymbol{S}) = \boldsymbol{m}$$

Hence by Eckart-Young-Mirsky Theorem,

$$\left\|\boldsymbol{X} - \hat{\boldsymbol{X}}^{(PCA)}\right\|_F \leq \left\|\boldsymbol{X} - \hat{\boldsymbol{X}}^{(S)}\right\|_F, \quad \textit{for all m-dimensional subspace S}$$

That is,

$$\sum_{i=1}^{N}\left\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i^{(PCA)}\right\|^2 \leq \sum_{i=1}^{N}\left\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i^{(S)}\right\|^2, \quad \textit{for all m-dimensional subspace S}$$

Question: Why it suffices to assume data points have zero mean?

Given data points $\boldsymbol{\mathfrak{x}}_1, \dots, \boldsymbol{\mathfrak{x}}_N \in \mathbb{R}^M$ with mean $\widetilde{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\mathfrak{x}}_i$ (not necessarily zero). Let S be a m-dimensional subspace, and denote $S_{\xi} = \boldsymbol{\xi} + S$ as the m-dimensional affine set by shifting S with vector $\boldsymbol{\xi} \in \mathbb{R}^M$.

Let $\widetilde{\boldsymbol{u}}_1, \dots, \widetilde{\boldsymbol{u}}_m$ be an orthonormal basis of S. Then the projection of $\widetilde{\boldsymbol{x}}_n$ to S_{ξ} is

$$\widehat{\boldsymbol{\mathfrak{x}}}_n^{(S_{\xi})} = \boldsymbol{\xi} + \sum_{i=1}^{m} \widetilde{\boldsymbol{u}}_i \widetilde{\boldsymbol{u}}_i^T (\boldsymbol{\mathfrak{x}}_n - \boldsymbol{\xi}),$$

and the projection error becomes

$$\boldsymbol{\mathfrak{x}}_n - \widehat{\boldsymbol{\mathfrak{x}}}_n^{(S_{\xi})} = \boldsymbol{\mathfrak{x}}_n - \boldsymbol{\xi} - \sum_{i=1}^{m} \widetilde{\boldsymbol{u}}_i \widetilde{\boldsymbol{u}}_i^T (\boldsymbol{\mathfrak{x}}_n - \boldsymbol{\xi}) = \underbrace{\left(\boldsymbol{I} - \sum_{i=1}^{m} \widetilde{\boldsymbol{u}}_i \widetilde{\boldsymbol{u}}_i^T\right)}_{\text{Denote as } A} (\boldsymbol{\mathfrak{x}}_n - \boldsymbol{\xi})$$

The sum of squared projection error becomes

$$\sum_{n=1}^{N} \left\| \boldsymbol{x}_n - \widehat{\boldsymbol{x}}_n^{(S_{\xi})} \right\|^2 = \sum_{n=1}^{N} \|\mathrm{A}(\boldsymbol{\mathfrak{x}}_n - \widetilde{\boldsymbol{\mu}}) + \mathrm{A}(\widetilde{\boldsymbol{\mu}} - \boldsymbol{\xi})\|^2$$

$$= \sum_{n=1}^{N} \left(\|\mathrm{A}(\boldsymbol{\mathfrak{x}}_n - \widetilde{\boldsymbol{\mu}})\|^2 + \underbrace{2\langle \mathrm{A}(\boldsymbol{\mathfrak{x}}_n - \widetilde{\boldsymbol{\mu}}), \mathrm{A}(\widetilde{\boldsymbol{\mu}} - \boldsymbol{\xi})\rangle}_{=0} + \underbrace{\|\mathrm{A}(\widetilde{\boldsymbol{\mu}} - \boldsymbol{\xi})\|^2}_{\text{Minimized when } \boldsymbol{\xi} = \widetilde{\boldsymbol{\mu}}}\right)$$

Hence, to minimize sum of squared projection error, one may take $\boldsymbol{\xi} = \widetilde{\boldsymbol{\mu}}$, in which case

$$\boldsymbol{\mathfrak{x}}_n - \widehat{\boldsymbol{\mathfrak{x}}}_n^{(S_{\widetilde{\mu}})} = \boldsymbol{x}_n - \widehat{\boldsymbol{x}}_n^{(S)}$$

where the shifted data $\boldsymbol{x}_n = \boldsymbol{\mathfrak{x}}_n - \widetilde{\boldsymbol{\mu}}$ $(n = 1, \dots, N)$ have zero mean.

Theorem 0.1. Eckart-Young-Mirsky theorem
Let $m \leq n$, $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a matrix with singular value decomposition $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$, where $\mathbf{U}, \mathbf{V}$ are unitary matrices, $\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_m)$ is a diagonal matrix with eigenvalues $|\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_m|$. Let $k \leq m$, then both low rank approximation problems

$$\begin{aligned}
&\underset{\mathbf{A}_k \in \mathbb{R}^{m\times n}}{minimize} \quad \|\mathbf{A} - \mathbf{A}_k\|_2 \quad subject\ to \quad rank(\mathbf{A}_k) \leq k \\
&\underset{\mathbf{A}_k \in \mathbb{R}^{m\times n}}{minimize} \quad \|\mathbf{A} - \mathbf{A}_k\|_F \quad subject\ to \quad rank(\mathbf{A}_k) \leq k
\end{aligned}$$

have optimal solution $\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$. Here $\mathbf{u}_i$ and $\mathbf{v}_i$ denote the i'th column in matrices $\mathbf{U}$ and $\mathbf{V}$, respectively.

Proof. • Low rank approximation under 2-norm: Prove by contradiction. Suppose there exists low rank matrix $\mathbf{B} \in \mathbb{R}^{m\times n}$ with $\text{rank}(\mathbf{B}) \le k$ such that

$$\|\mathbf{A} - \mathbf{B}\|_2 < \|\mathbf{A} - \mathbf{A}_k\|_2 = |\sigma_{k+1}|.$$

Note that each nonzero vector $\mathbf{w} \in \text{Null}(\mathbf{B})$ satisfies

$$\frac{\|\mathbf{A}\mathbf{w}\|_2}{\|\mathbf{w}\|_2} = \frac{\|(\mathbf{A} - \mathbf{B})\mathbf{w}\|_2}{\|\mathbf{w}\|_2} \le \|\mathbf{A} - \mathbf{B}\|_2 < |\sigma_{k+1}|$$

On the other hand, each nonzero vector $\mathbf{x} \in \text{Span}(\mathbf{v}_1, \cdots, \mathbf{v}_{k+1})$ satisfies

$$\frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \ge |\sigma_{k+1}|$$

Hence $\text{Null}(\mathbf{B})$ and $\text{Span}(\mathbf{v}_1, \cdots, \mathbf{v}_{k+1})$ are linear independent subspaces in $\mathbb{R}^n$. However

$$\dim(\text{Null}(\mathbf{B})) + \dim(\text{Span}(\mathbf{v}_1, \cdots, \mathbf{v}_{k+1})) = (n - \text{rank}(B)) + (k+1) \ge n+1$$

which is greater than the dimension of $\mathbb{R}^n$, leading to a contradiction.

- Low rank approximation under Frobenius norm: For arbitrary $\mathbf{B} \in \mathbb{R}^{m\times n}$ with $\operatorname{rank}(\mathbf{B}) \le k$, denote $\mathbf{N} = \mathbf{U}^T\mathbf{B}\mathbf{V}$, then

$$\|\mathbf{A}-\mathbf{B}\|_F^2 = \|\mathbf{U}^T(\mathbf{A}-\mathbf{B})\mathbf{V}\|_F^2 = \|\Sigma - \mathbf{N}\|_F^2$$

Since $\dim(\mathrm{Null}(\mathbf{N})) \ge n-k$, let $\xi_1, \cdots, \xi_{n-k}$ be orthonormal vectors in $\mathrm{Null}(\mathbf{N})$, and let $\Xi = [\xi_1 \ \cdots \ \xi_n] \in \mathbb{R}^{n\times n}$ be a unitary matrix. Then

$$\begin{aligned}
\|\Sigma-\mathbf{N}\|_F^2 &= \|(\Sigma-\mathbf{N})\Xi\|_F^2 = \sum_{i=1}^{n} \|(\Sigma-\mathbf{N})\xi_i\|^2 \\
&\ge \sum_{i=1}^{n-k} \|(\Sigma-\mathbf{N})\xi_i\|^2 = \sum_{i=1}^{n-k} \|\Sigma\xi_i\|^2 = \sum_{j=1}^{m} \sigma_j^2 \overset{w_j}{\boxed{\sum_{i=1}^{n-k} \left(\xi_i^{(j)}\right)^2}}
\end{aligned}$$

Denote $w_j = \sum_{i=1}^{n-k} \left(\xi_i^{(j)}\right)^2$, then $0 \le w_j \le 1, \forall 1 \le j \le n$, and

$$\sum_{j=1}^{n} w_j = \sum_{j=1}^{n}\sum_{i=1}^{n-k} \left(\xi_i^{(j)}\right)^2 = \sum_{i=1}^{n-k}\sum_{j=1}^{n} \left(\xi_i^{(j)}\right)^2 = \sum_{i=1}^{n-k} \|\xi_i\|^2 = n-k$$

Therefore

$$\|\Sigma-\mathbf{N}\|_F^2 \ge \sum_{j=1}^{m} w_j\sigma_j^2 \ge \sum_{j=1}^{k} 0\cdot\sigma_j^2 + \sum_{j=k+1}^{m} 1\cdot\sigma_j^2 = \|\mathbf{A}-\mathbf{A}_k\|_F^2$$

Take $\boldsymbol{X} = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \ldots \boldsymbol{x}_N]$ s.t. $\boldsymbol{\Sigma} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, then

$$\mathrm{Trace}(\boldsymbol{\Phi}^T\boldsymbol{\Sigma}\boldsymbol{\Phi}) = \frac{1}{N}\mathrm{Trace}(\boldsymbol{\Phi}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{\Phi}) = \frac{1}{N}\|\boldsymbol{\Phi}^T\boldsymbol{X}\|_F^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{\Phi}^T\boldsymbol{x}_i\|^2 = \frac{1}{N}\sum_{i=1}^{N}\left\|\widehat{\boldsymbol{x}}_i^{(S)}\right\|^2 \le \frac{1}{N}\sum_{i=1}^{N}\left\|\widehat{\boldsymbol{x}}_i^{(PCA)}\right\|^2$$

Optimization problem:

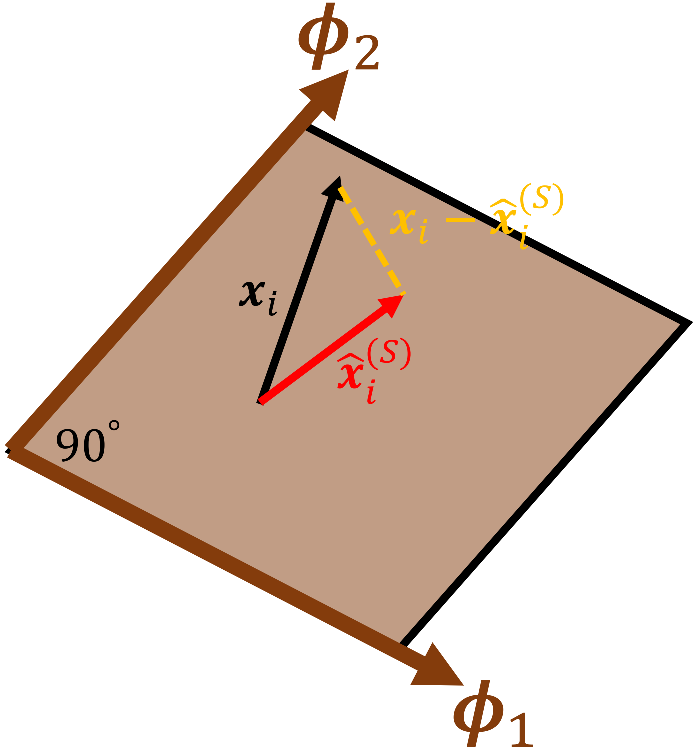
maximize $\mathrm{Trace}(\boldsymbol{\Phi}^T\boldsymbol{\Sigma}\boldsymbol{\Phi})$

subject to $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{I}_m$

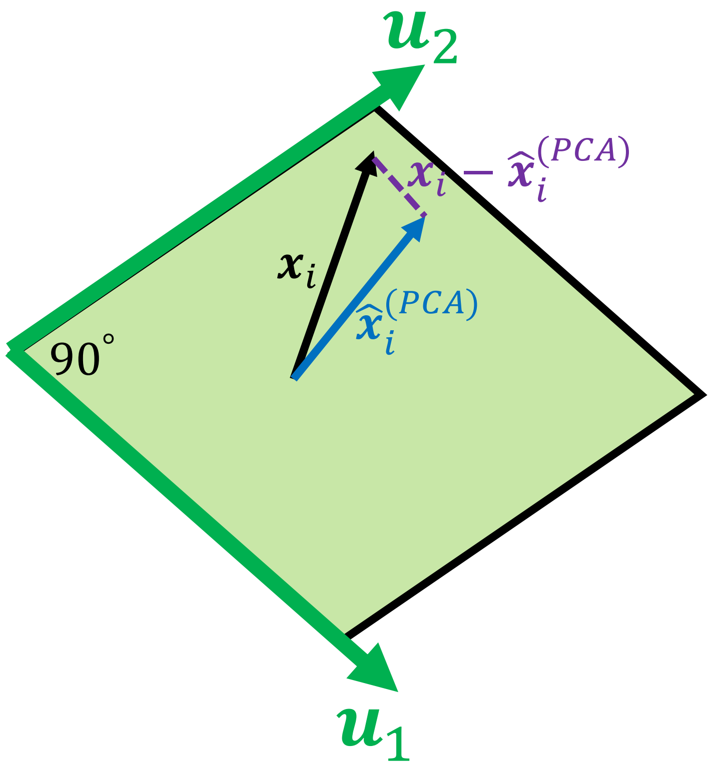
variables $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_m] \in \mathbb{R}^{M\times m}$

Optimal solution: **PCA axes**

$$\boldsymbol{\Phi}_{opt} = [\boldsymbol{u}_1\ \boldsymbol{u}_2\ \ldots \boldsymbol{u}_m]$$



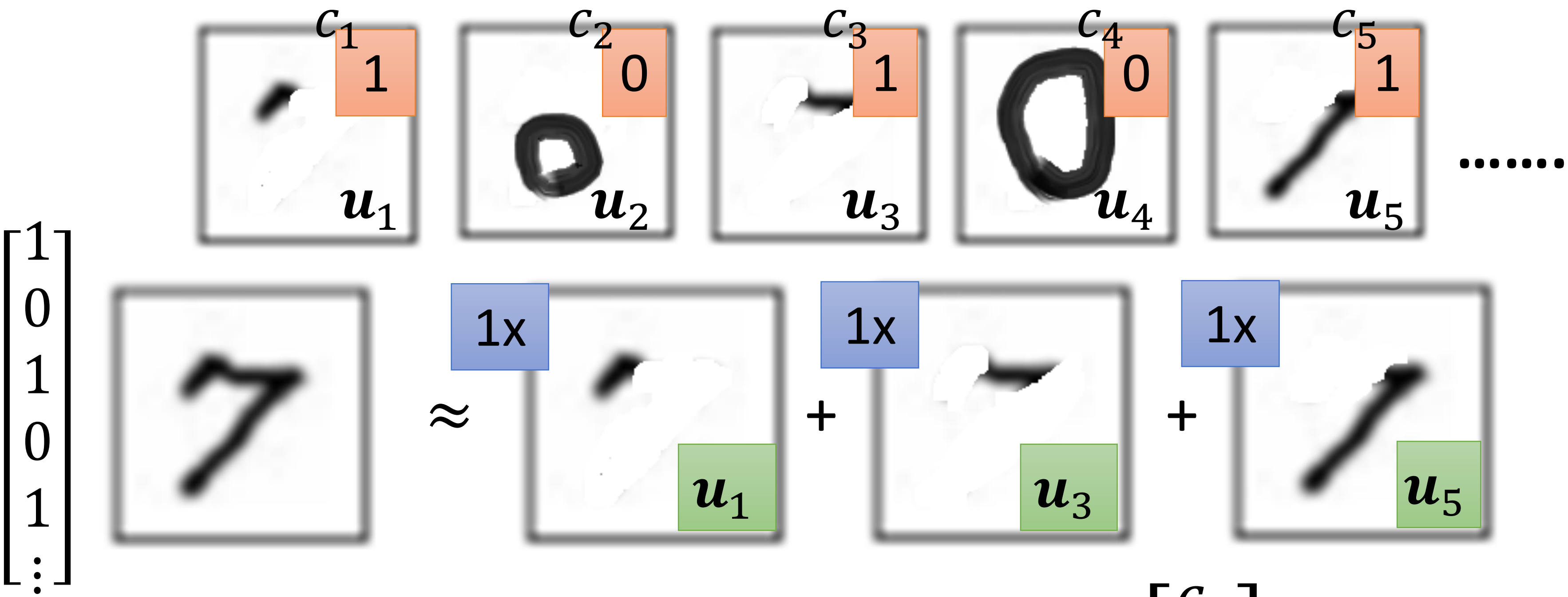
$S = span(\boldsymbol{\Phi})$ is a m-dimensional subspace



$S_{PCA} = Span(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m)$

PCA – Another Point of View

Basic Component:



$$x \approx c_1 \boldsymbol{u}_1 + c_2 \boldsymbol{u}_2 + \cdots + c_{\mathrm{K}} \boldsymbol{u}_K + \boldsymbol{\mu}$$

Pixels in a digit image

component

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{\mathrm{K}} \end{bmatrix}$$

Represent a digit image

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

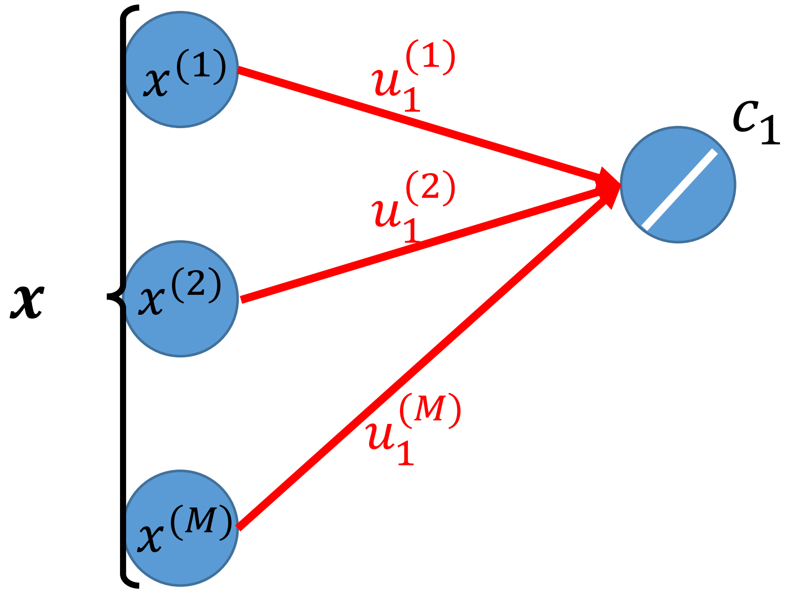
Autoencoder

$$\hat{\boldsymbol{x}} = \sum_{k=1}^{K} c_k \boldsymbol{u}_k + \boldsymbol{\mu}$$

To minimize reconstruction error:

$$c_k = (\boldsymbol{x} - \boldsymbol{\mu}) \cdot \boldsymbol{u}_k$$

$K = 2$:



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

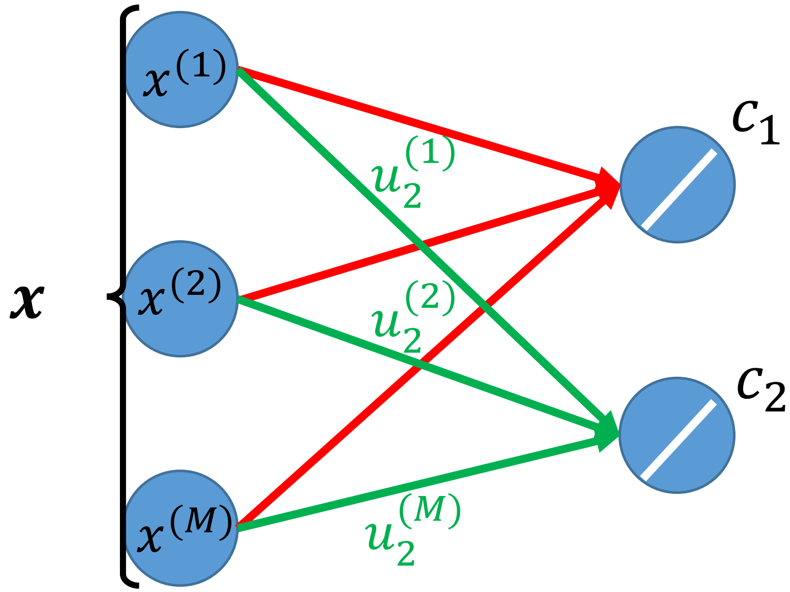
Autoencoder

$$\hat{\boldsymbol{x}} = \sum_{k=1}^{K} c_k \boldsymbol{u}_k + \boldsymbol{\mu}$$

To minimize reconstruction error:

$$c_k = (\boldsymbol{x} - \boldsymbol{\mu}) \cdot \boldsymbol{u}_k$$

$K = 2$:



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

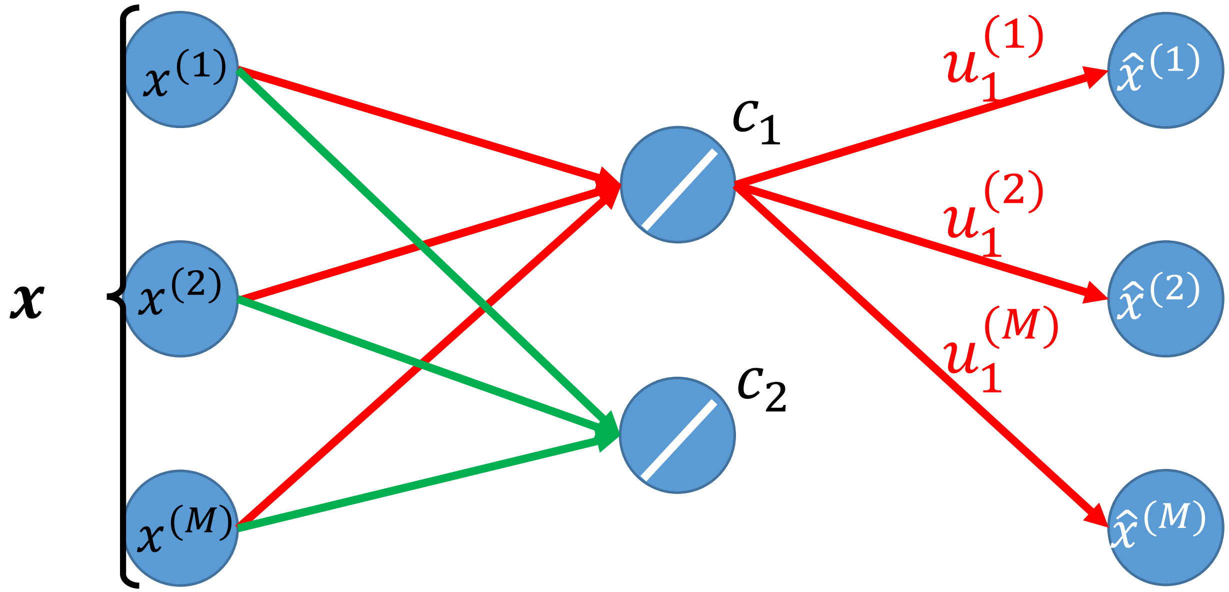
Autoencoder

$$\hat{\boldsymbol{x}} = \sum_{k=1}^{K} c_k \boldsymbol{u}_k + \boldsymbol{\mu}$$

To minimize reconstruction error:

$$c_k = (\boldsymbol{x} - \boldsymbol{\mu}) \cdot \boldsymbol{u}_k$$

$K = 2$:



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

Autoencoder

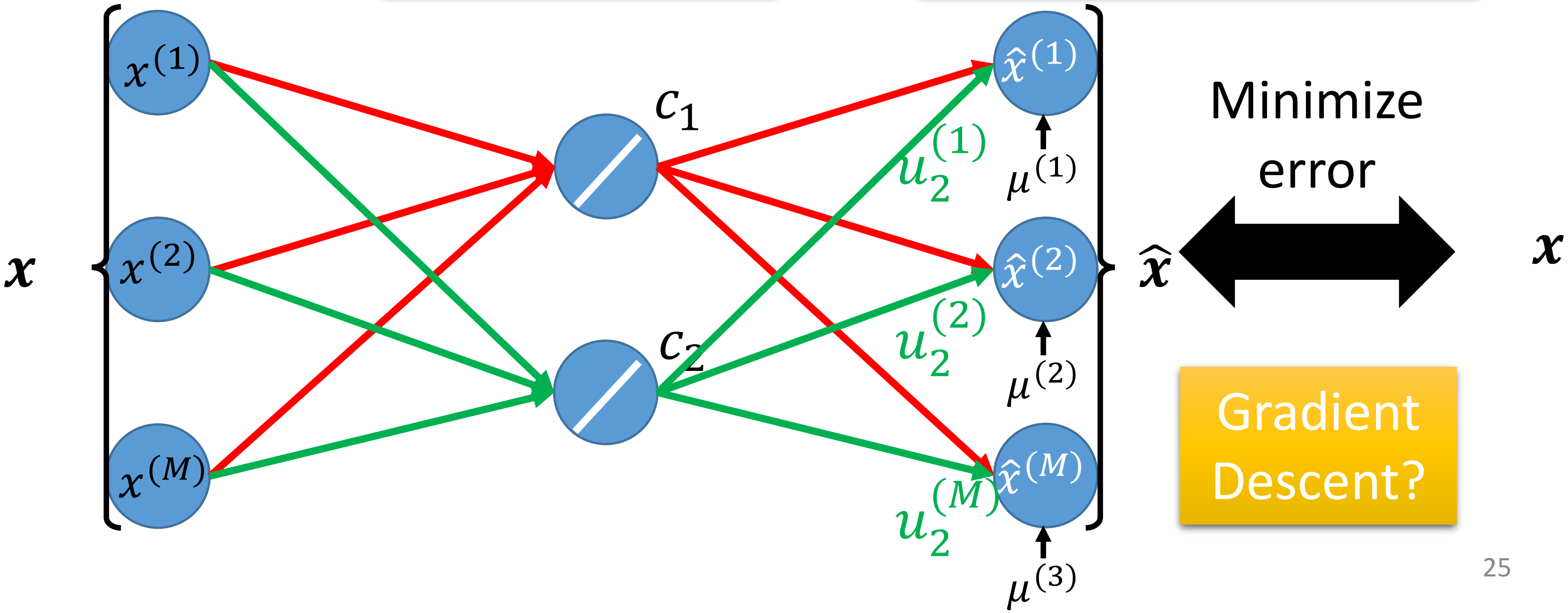
$$\hat{\boldsymbol{x}} = \sum_{k=1}^{K} c_k \boldsymbol{u}_k + \boldsymbol{\mu}$$

To minimize reconstruction error:

$$c_k = (\boldsymbol{x} - \boldsymbol{\mu}) \cdot \boldsymbol{u}_k$$

$K = 2$:

It can be deep. → Deep Autoencoder



PCA - Pokémon

- Inspired from: https://www.kaggle.com/strakul5/d/abcsds/pokemon/principal-component-analysis-of-pokemon-data
- 800 Pokemons, 6 features for each (HP, Atk, Def, Sp Atk, Sp Def, Speed)
- How many principle components? $\dfrac{\lambda_i}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6}$

	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

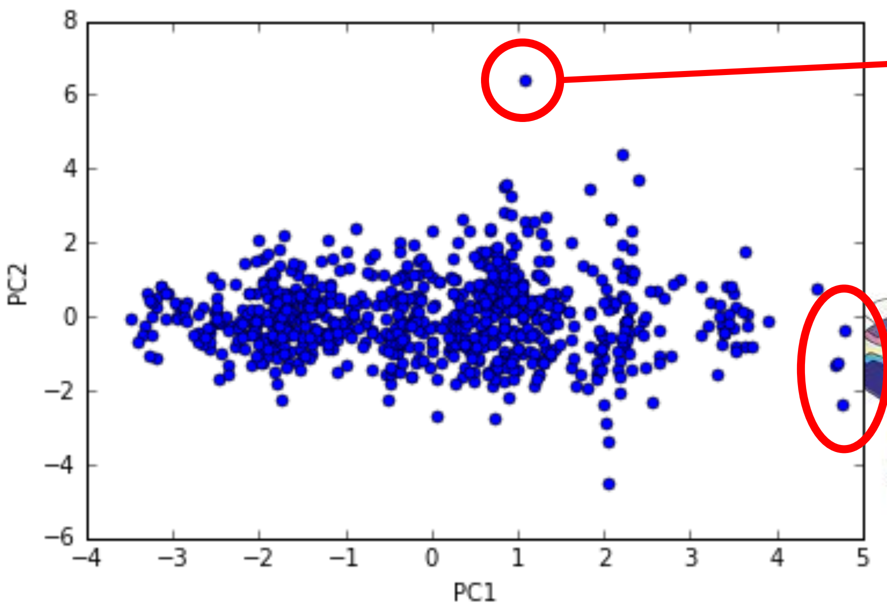
Using 4 components is good enough

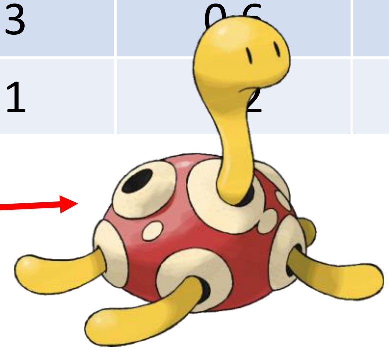
PCA - Pokémon

	HP	Atk	Def	Sp Atk	Sp Def	Speed
PC1	0.4	0.4	0.4	0.5	0.4	0.3
PC2	0.1	0.0	0.6	-0.3	0.2	-0.7
PC3	-0.5	-0.6	0.1	0.3	[illegible]	[illegible]
PC4	0.7	-0.4	-0.4	0.1	[illegible]	-0.3

強度

防禦(犧牲速度)



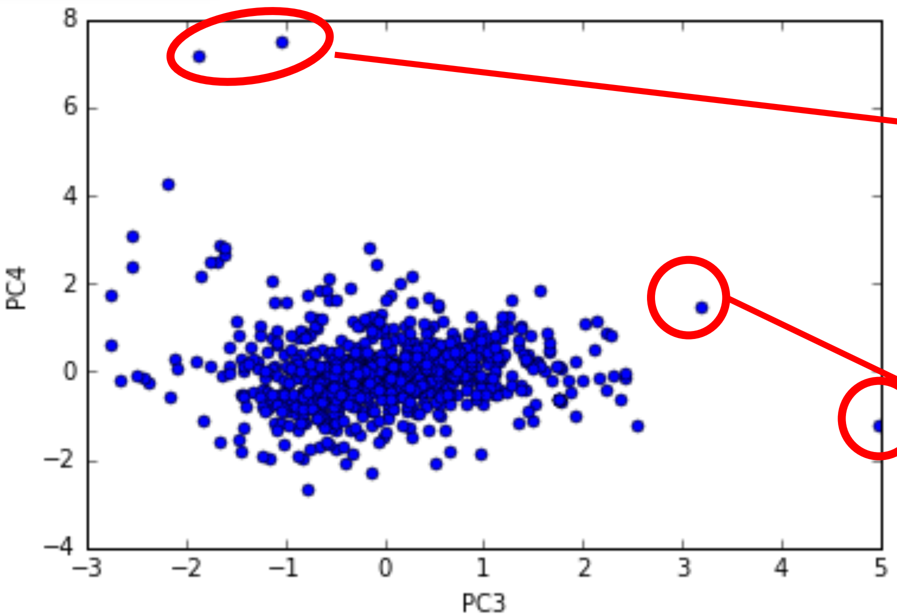


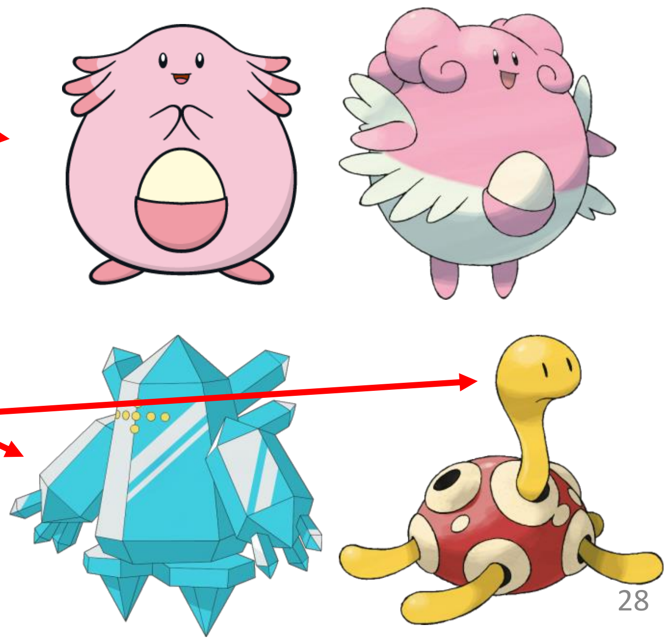
PCA - Pokémon

	HP	Atk	Def	Sp Atk	Sp Def	Speed
PC1	0.4	0.4	0.4	0.5	0.4	0.3
PC2	0.1	0.0	0.6	-0.3	0.2	-0.7
PC3	-0.5	-0.6	0.1	0.3	0.6	[illegible]
PC4	0.7	-0.4	-0.4	0.1	0.2	[illegible]

特殊防禦(犧牲攻擊和生命)

生命力強



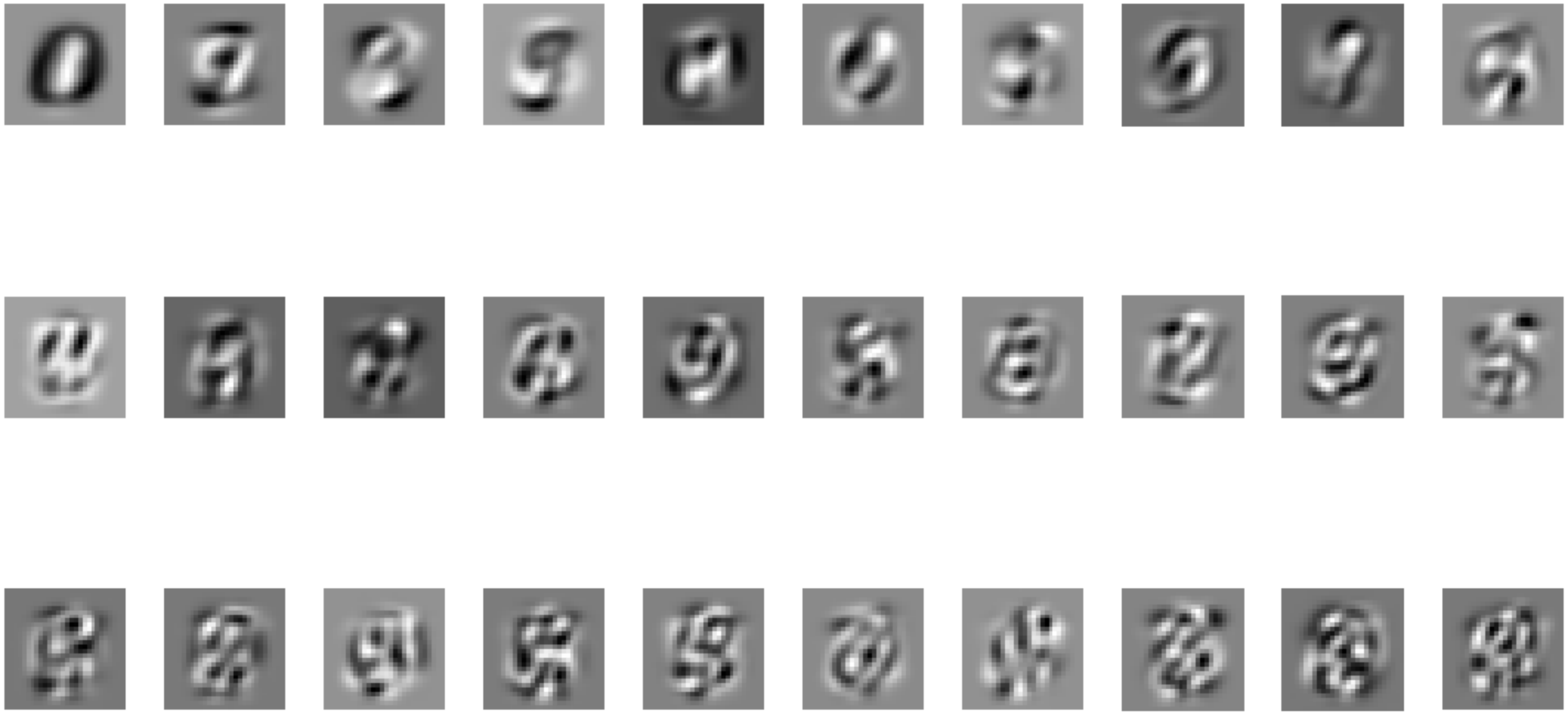


PCA - MNIST

$$= a_1 w^1 + a_2 w^2 + \cdots$$

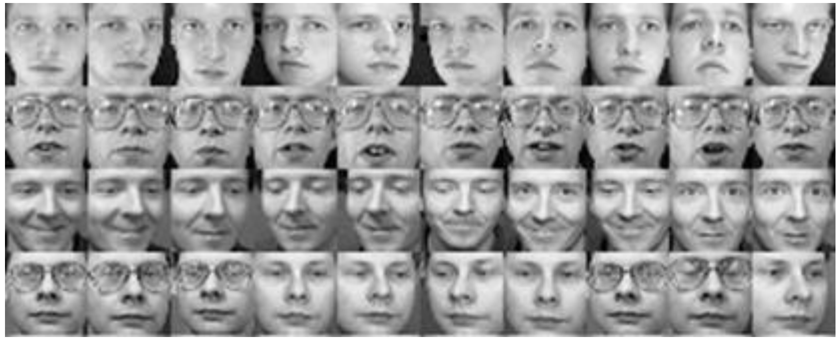
images

30 components:



Eigen-digits

PCA - Face



30 components:

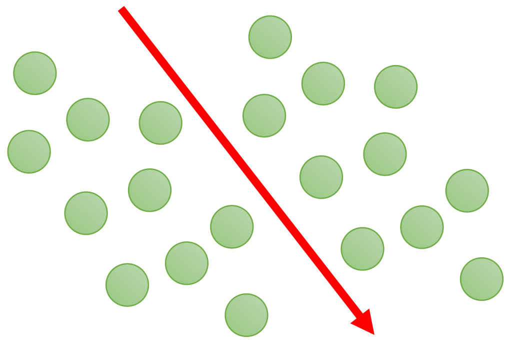
http://www.cs.unc.edu/~lazebnik/research/spring08/assignment3.html

Eigen-face

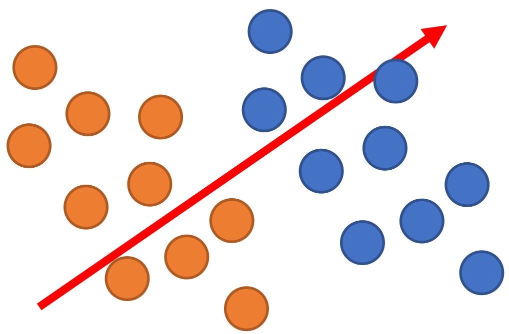
Weakness of PCA

- Unsupervised

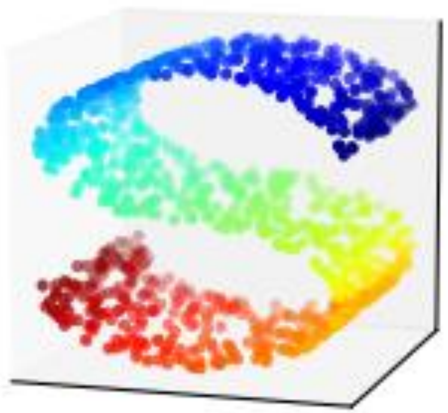
PCA

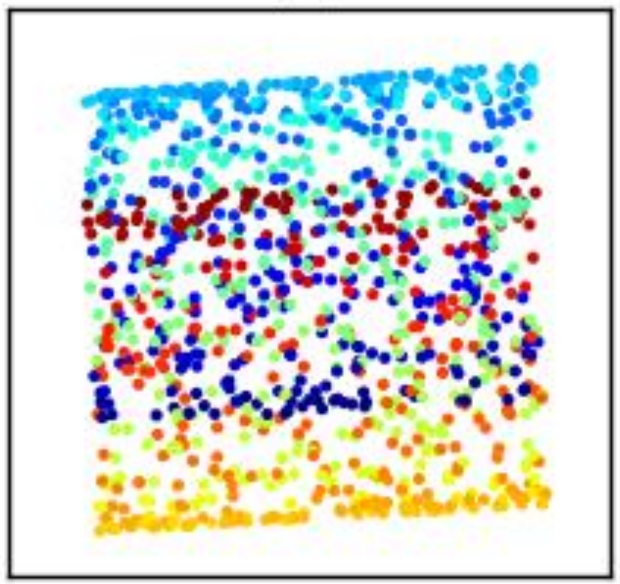


LDA



- Linear





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