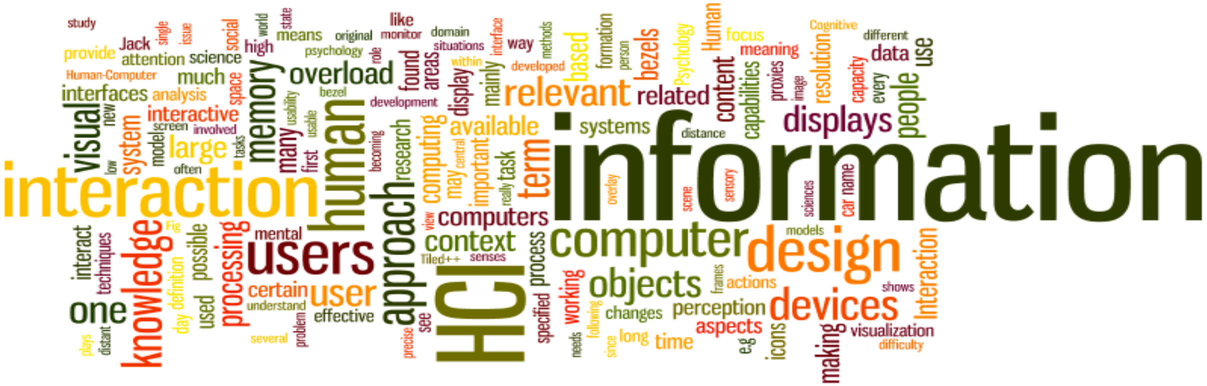
机器学习

—— 第11章 降维——



倪张凯
zkni@tongji.edu.cn
https://eezkni.github.io/

维度灾难问题 Curse of dimensionality

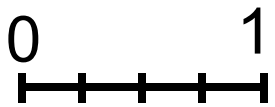
- Many explored domains have hundreds to tens of thousands of **variables/features** with many **irrelevant** and **redundant** ones!
- In domains with many features the underlying probability distribution can be very complex and very hard to estimate (e.g. dependencies between variables) !
- Irrelevant and redundant features can confuse learners!

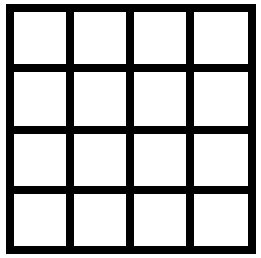
- Limited training data!
- Limited computational resources!
- **Curse of dimensionality**!

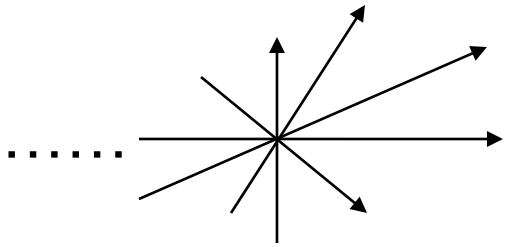
维度灾难问题 Curse of dimensionality

- Spatial sampling

4 units for each dimension, 10 samples per unit







total sample：

1-D：4 2-D：4*4=16 10-D：4^10=1048576

~40 ~160 ~10M

– Sample sparsity

total sample：1000

partition of each dimension ：4

Samples per unit ：

1D：1000/4 = 250

2D：1000/(4*4)= 62.5

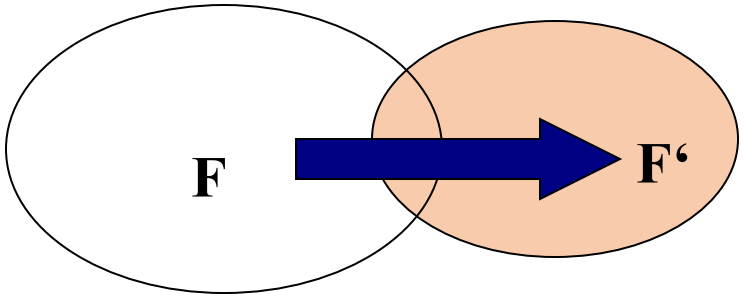
10D：1000/(4^10)= 0.001

噪声影响 Noise influence

- Feature space ： 101D
- The distance between the positive and negative samples at the first dimension: 1
- The noise of the sample in its residual dimension ： 10%
- “Noise distance” ： $\sqrt{100 \times 0.1^2} = 1$
- Even if the noise is only 10%, the noise distance in high dimensional space is enough to mask the essential difference between positive and negative samples.

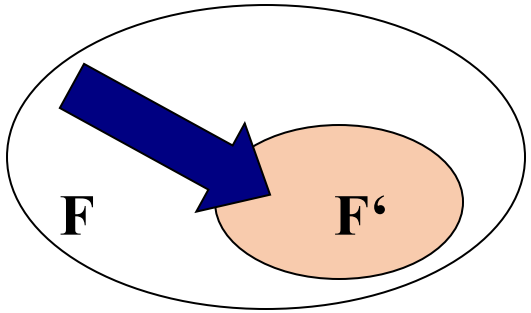
降维与特征选择 Dimensionality reduction& Feature selection

- Dimensionality reduction:creating a subset of new features by combinations of the existing features



$$\{f_1,\ldots,f_i,\ldots,f_n\} \xrightarrow[f.extraction]{} \{g_1(f_1,\ldots,f_n),\ldots,g_j(f_1,\ldots,f_n),\ldots,g_m(f_1,\ldots,f_n)\}$$

- Feature Selection: choosing a subset of all the features



$$\{f_1,\ldots,f_i,\ldots,f_n\} \xrightarrow[f.selection]{} \{f_{i_1},\ldots,f_{i_j},\ldots,f_{i_m}\}$$

$$i_j \in \{1,\ldots,n\};\ j=1,\ldots,m$$

$$i_a = i_b \Rightarrow a=b;\ \ a,b \in \{1,\ldots,m\}$$

Machine Learning Problems

	Supervised Learning	*Unsupervised Learning*
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

降维方法 Dimensionality reduction Method

- Principal Component Analysis (PCA)
- Probabilistic PCA
- Kernel PCA
- Multidimensional Scaling (MDS)
- Locally Linear Embedding (LLE)
- Laplacian Eigenmaps (LE）
- Isometric Mapping(Isomap)
- T-distributed Stochastic Neighbor Embedding (t-SNE)
- Auto-encoders
- Non-negative Matrix Factorization(NMF)
- Canonical Correlation Analysis(CCA)
- Independent Component Analysis (ICA)
- (Linear Discriminant Analysis(LDA)-supervised)
- …

举例 Dimensionality Reduction Example

- Given 53 blood and urine samples (features) from 65 people.
- How can we visualize the measurements?

Matrix format (65x53)

	H-WBC	H-RBC	H-Hgb	H-Hct	H-MCV	H-MCH	H-MCHC
A1	8.0000	4.8200	14.1000	41.0000	85.0000	29.0000	34.0000
A2	7.3000	5.0200	14.7000	43.0000	86.0000	29.0000	34.0000
A3	4.3000	4.4800	14.1000	41.0000	91.0000	32.0000	35.0000
A4	7.5000	4.4700	14.9000	45.0000	101.0000	33.0000	33.0000
A5	7.3000	5.5200	15.4000	46.0000	84.0000	28.0000	33.0000
A6	6.9000	4.8600	16.0000	47.0000	97.0000	33.0000	34.0000
A7	7.8000	4.6800	14.7000	43.0000	92.0000	31.0000	34.0000
A8	8.6000	4.8200	15.8000	42.0000	88.0000	33.0000	37.0000
A9	5.1000	4.7100	14.0000	43.0000	92.0000	30.0000	32.0000

Instances

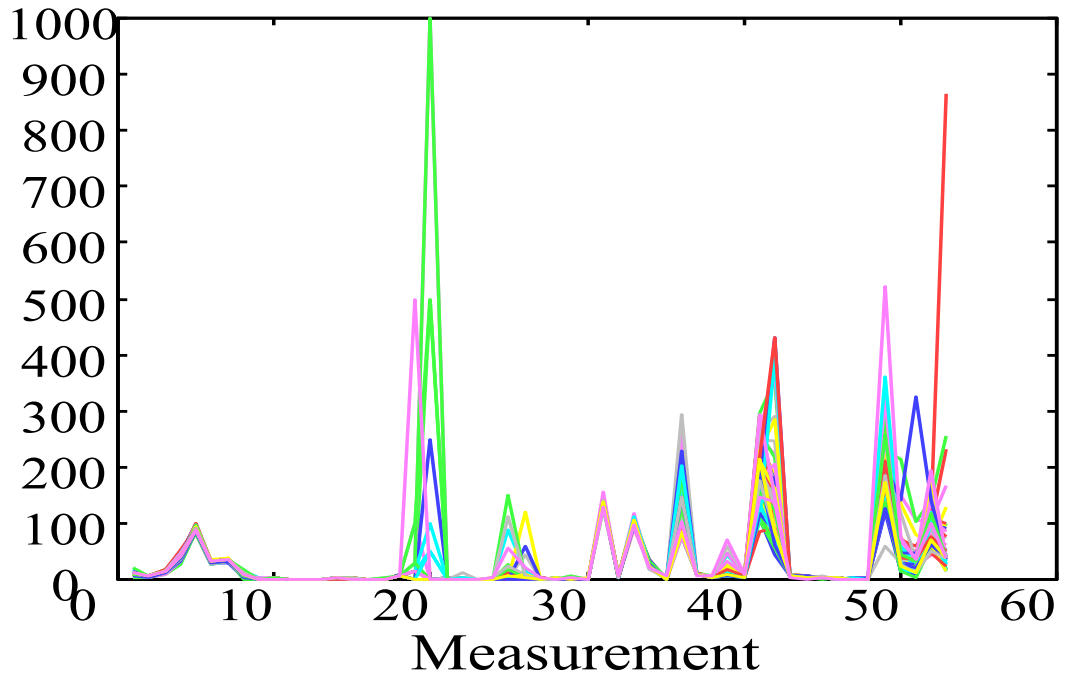
Features

Difficult to see the correlations between the features...

举例 Dimensionality Reduction Example

- Given 53 blood and urine samples (features) from 65 people.
- How can we visualize the measurements?

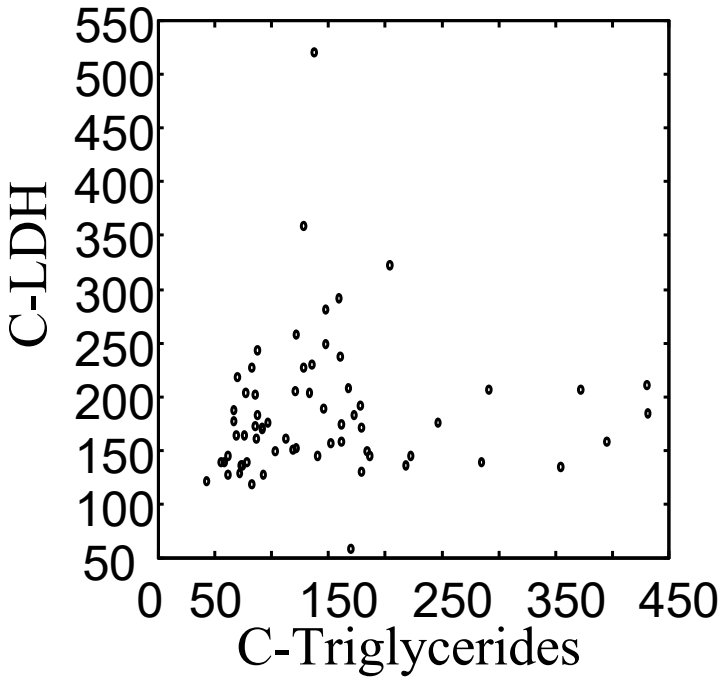
Matrix format (65x53)



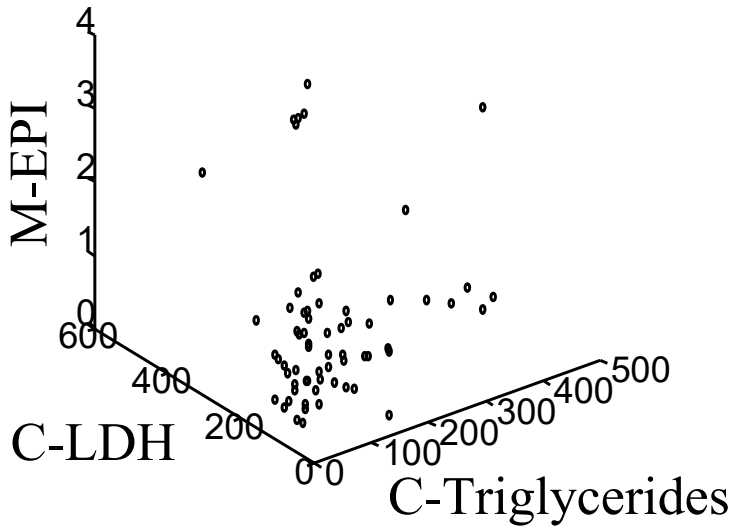
Difficult to compare the different patients...

举例 Dimensionality Reduction Example

Bi-variate



Tri-variate



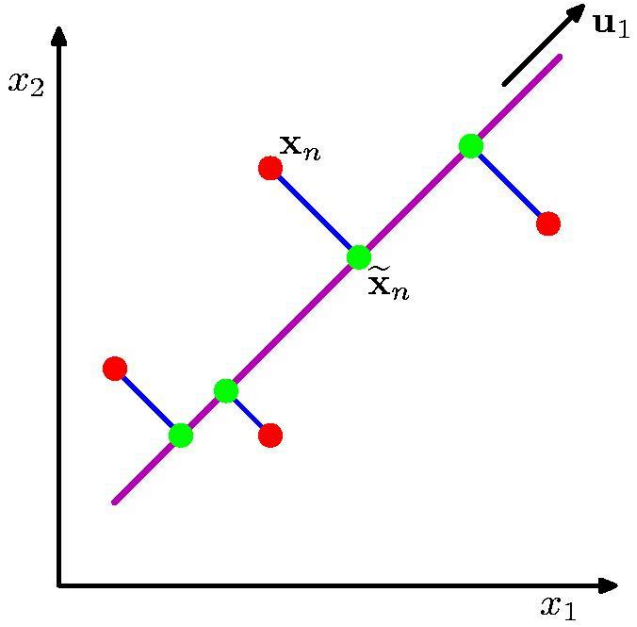
How can we visualize the other variables???

… difficult to see in 4 or higher dimensional spaces...

举例 Dimensionality Reduction Example

- Is there a representation better than the coordinate axes?
- Is it really necessary to show all the 53 dimensions?
  - discard low significance dimensions
  - Get compact description (what if there are strong correlations between the features？)
- How could we find
  the *smallest* subspace of the 53-D space that
  keeps the *most information* about the original data?
- A solution: Principal Component Analysis (PCA)

主成分分析 Principle Component Analysis



PCA:

Orthogonal projection of data onto lower-dimension linear space that...

- maximizes **variance** of projected data (purple line)
- minimizes **mean squared distance** between data point and projections (sum of blue lines)

主成分分析 Principle Component Analysis

- **Data Preparation**:
  - Standardize the data if the features have different units or scales. (centering, scaling)

- **Compute and Select Principal Components**:
  - Iterative Algorithms (For very large datasets, can find the main components without computing the full covariance matrix)
  - Covariance Matrix (data dimensions are not excessively high)
  - Singular Value Decomposition (can handle matrices of any size)

- **Transform the Original Dataset**:
  - Project the original dataset into the new space formed by the selected principal components to obtain the reduced-dimensionality data representation.

主成分分析算法一（顺序求解）
PCA algorithm I (sequential)

- **Vectors** originating from the center of mass

- Principal component #1 points in the direction of the **largest variance**.

- Each subsequent principal component…
  - is **orthogonal** to the previous ones, and
  - points in the directions of the **largest variance of the residual subspace**

主成分分析算法一（顺序求解）

PCA algorithm I (sequential)

Given the **centered** data $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, compute the principal vectors:

$$\mathbf{w}_1 = \arg\max_{\|\mathbf{w}\|=1} \frac{1}{m}\sum_{i=1}^{m}\{(\mathbf{w}^T\mathbf{x}_i)^2\}$$

1st PCA vector

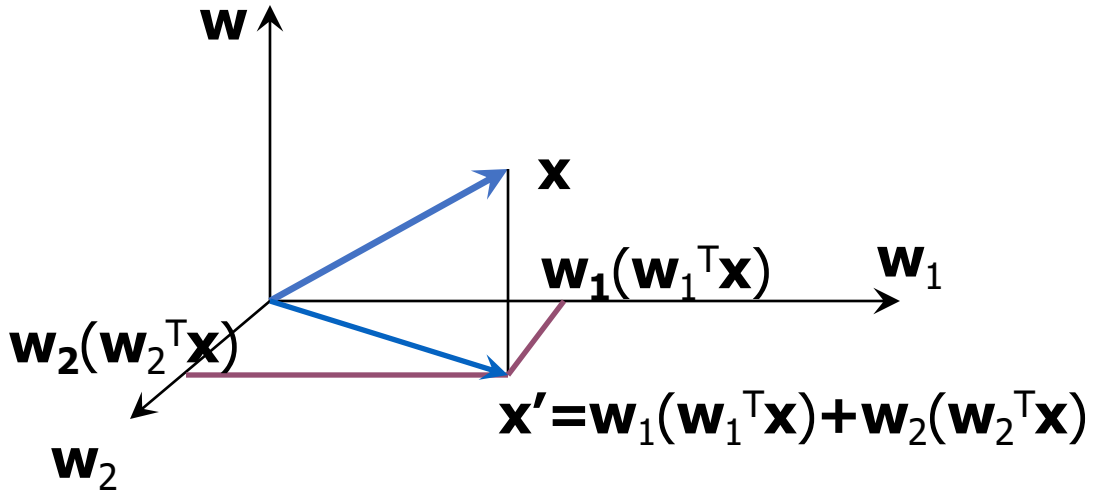
can be obtained by gradient ascent

$$\mathbf{w}_k = \arg\max_{\|\mathbf{w}\|=1} \frac{1}{m}\sum_{i=1}^{m}\{[\mathbf{w}^T(\mathbf{x}_i - \sum_{j=1}^{k-1}\mathbf{w}_j\mathbf{w}_j^T\mathbf{x}_i)]^2\}$$

k^{th} PCA vector

$\mathbf{x}'$ PCA reconstruction

We maximize the variance of the projection in the residual subspace



主成分分析算法二（特征值分解）

PCA algorithm II(sample covariance matrix)

- Given data $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, compute covariance matrix Σ

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T \quad \text{where} \quad \bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i$$

(Optional) Standardization

- Eigenvalue decomposition $\Sigma = U \wedge U^T$
- PCA basis vectors = the eigenvectors of Σ
- Larger eigenvalues $\Rightarrow$ more important eigenvectors

主成分分析算法二（特征值分解）

PCA algorithm II(sample covariance matrix)

PCA algorithm(**X**, *k*): top *k* eigenvalues/eigenvectors

% **X** = N × m data matrix, each data point $\mathbf{x}_i$ = column vector, i=1..m

- **X** ← subtract mean $\overline{X}$ from each column vector $\mathbf{x}_i$ in **X**
- $\Sigma \leftarrow \mathbf{X}\mathbf{X}^T$ …compute covariance matrix of **X**
- $\{ \lambda_i, \mathbf{u}_i \}_{i=1..N}$ = eigenvectors/eigenvalues of Σ
  ... Sort the eigenvalues in descending order $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$,
- Return $\{ \lambda_i, \mathbf{u}_i \}_{i=1..k}$
  % top *k* principle components

主成分分析算法三（奇异值分解）

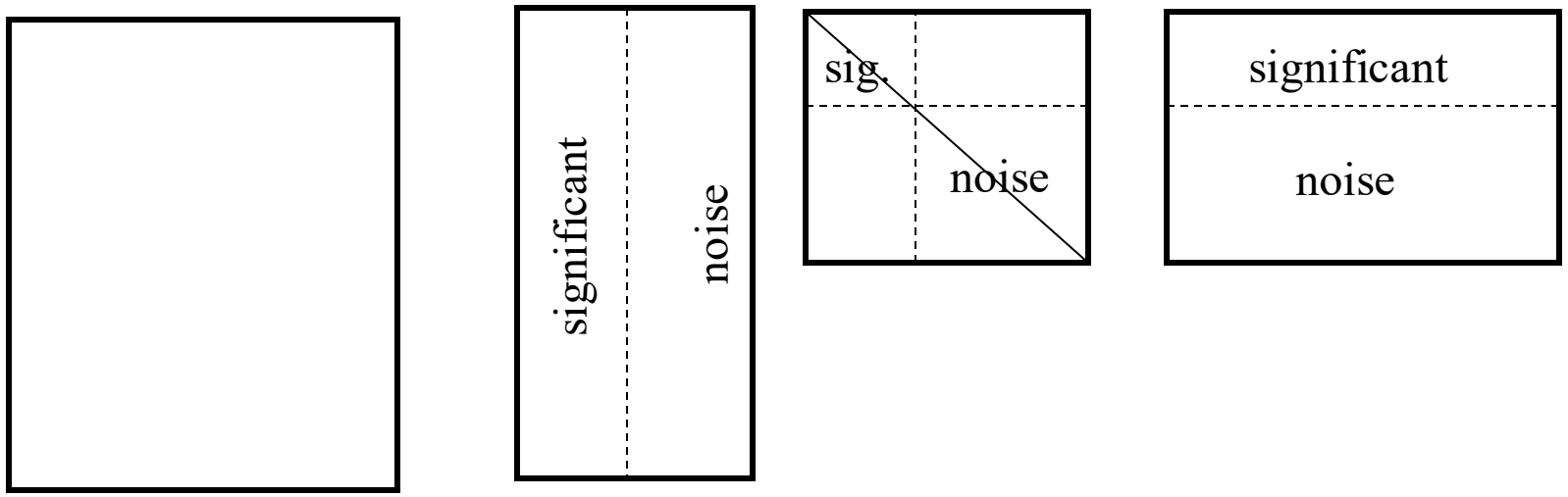
PCA algorithm III (SVD of the data matrix)

Singular Value Decomposition of the **centered** data matrix **X**.

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \in \mathbb{R}^{N \times m},$$

m: number of instances,
N: dimension

$$X_{N\times m} = U_{N\times N} S_{N\times m} V^T_{m\times m} \approx U_{N\times k} S_{k\times k} V^T_{k\times m}$$



主成分分析算法三（奇异值分解）

PCA algorithm III (SVD of the data matrix)

- **Columns of U**
  - the principal vectors, $\{ \mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \}$
  - orthogonal and has unit norm – so $U^T U = I$
  - Can reconstruct the data using linear combinations of $\{ \mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \}$
- **Matrix S**
  - Diagonal
  - Shows importance of each eigenvector
- **Columns of V^T**
  - The coefficients for reconstructing the samples

PCA-Another Point of View

$$x - \bar{x} \approx c_1 u^1 + c_2 u^2 + \cdots + c_K u^K = \hat{x}$$

Reconstruction error:
$\| (x - \bar{x}) - \hat{x} \|_2$

Find $\{u^1, \ldots, u^K\}$ minimizing the error

$$L = \min_{\{u^1,\ldots,u^K\}} \sum \left\| (x - \bar{x}) - \underbrace{\left(\sum_{k=1}^{K} c_k u^k \right)}_{\hat{x}} \right\|_2$$

PCA: $z = Wx$

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_K \end{bmatrix} = \begin{bmatrix} (w_1)^T \\ (w_2)^T \\ \vdots \\ (w_K)^T \end{bmatrix} x$$

$\{w^1, w^2, \ldots w^K\}$ is the component

$\{u^1, u^2, \ldots u^K\}$ minimizing L

Proof in [Bishop, Chapter 12.1.2]

主成分个数 How many PCA components

Problem: How many to keep?

Many criteria.

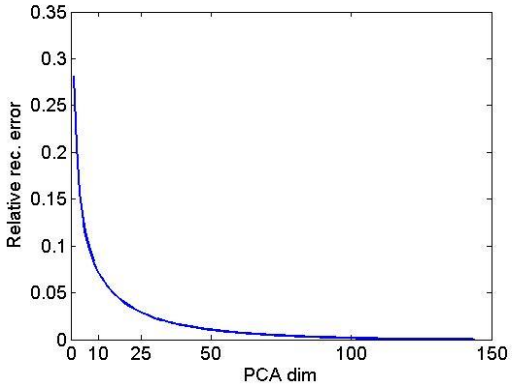
e.g. % total data variance:

$$\max(m) \ni \frac{\sum_{i=(m+1):n} \lambda_i}{\sum_{i=1:n} \lambda_i} < \varepsilon$$

	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

L_2 error and PCA dim



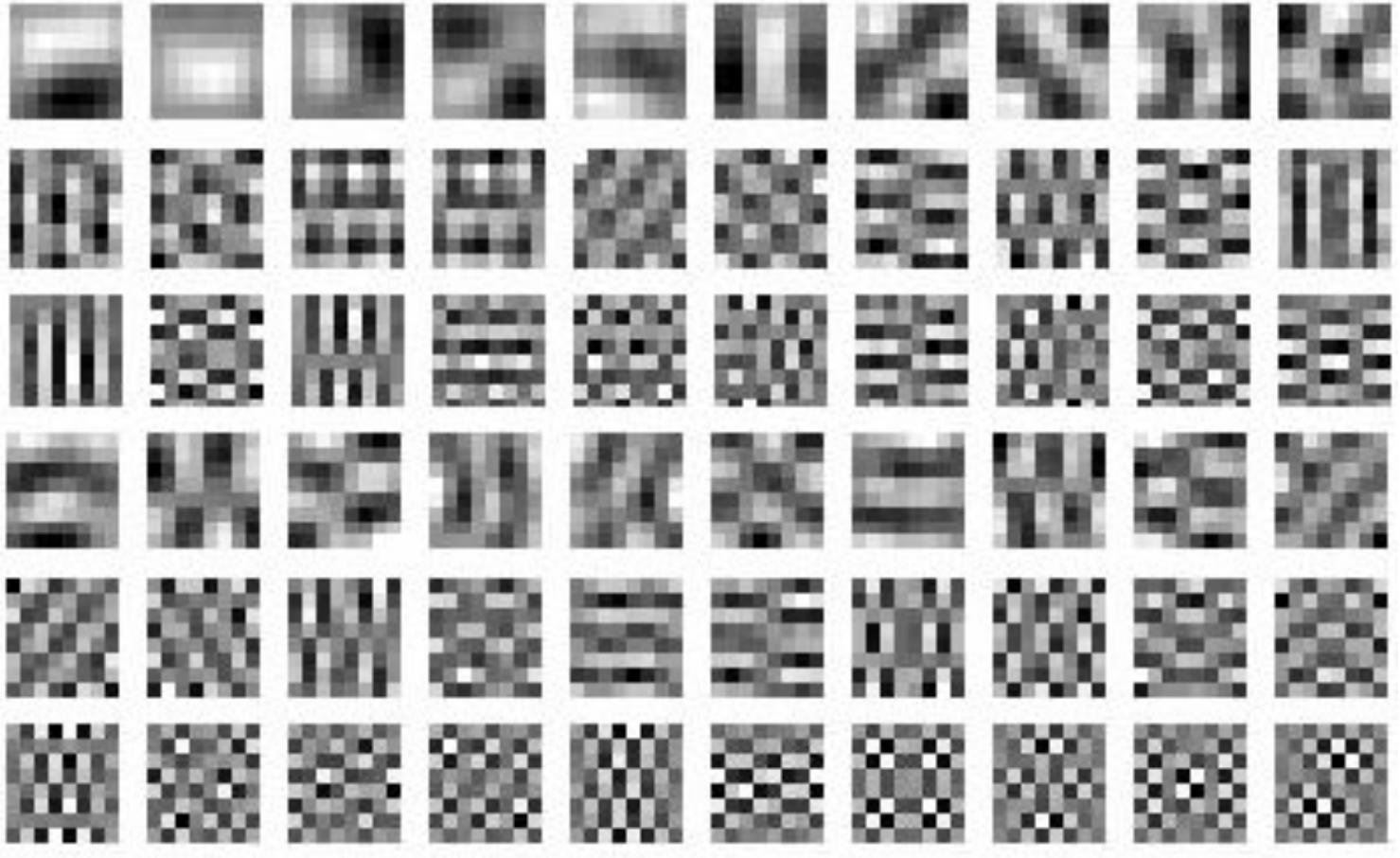
主成分分析用于图像压缩 PCA application: Image Compression

Original Image

- Divide the original 372x492 image into patches:
  - Each patch is an instance that contains 12x12 pixels on a grid (31*41=1271)
- View each as a 144-D vector

主成分分析用于图像压缩 PCA application: Image Compression

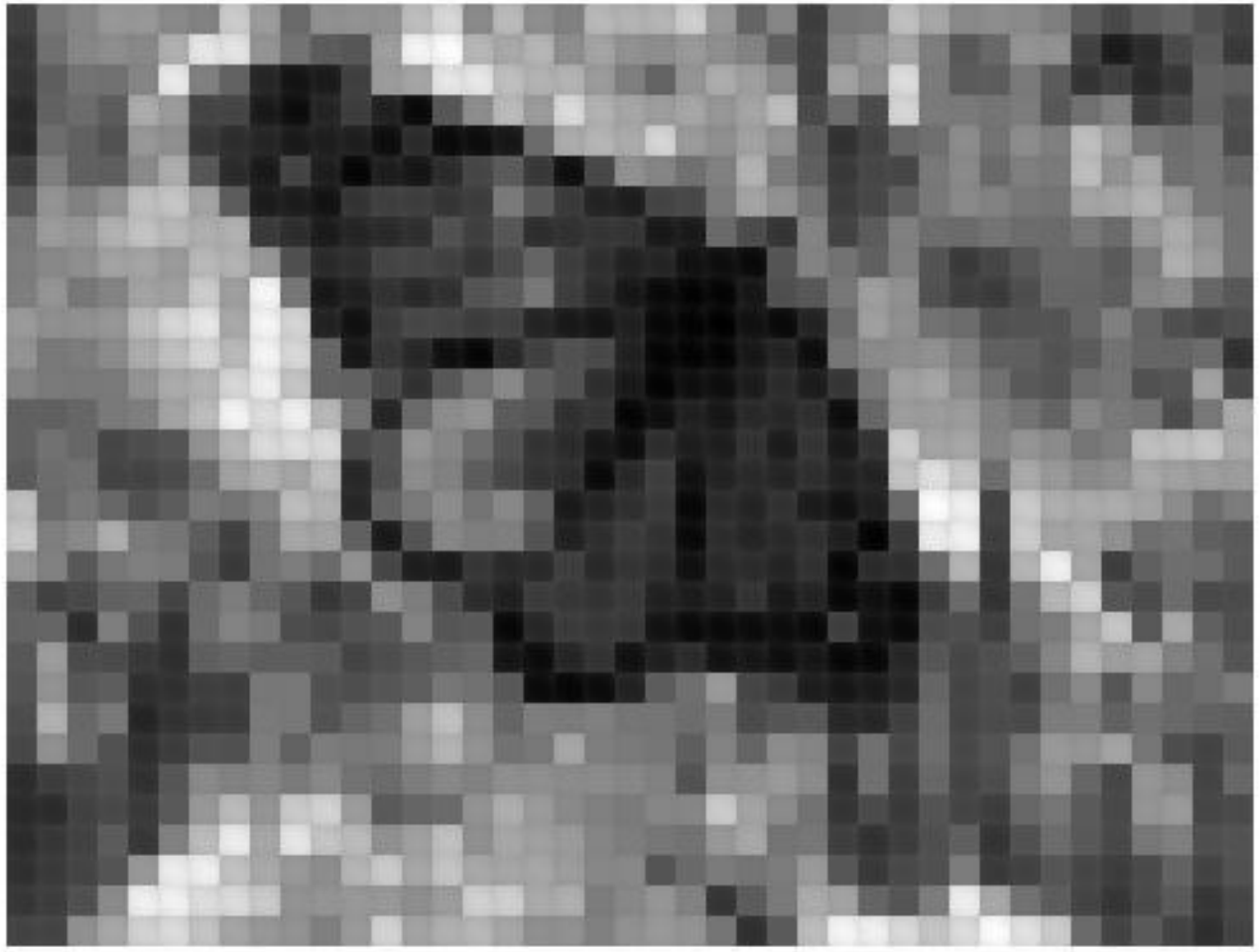
60 most important eigenvectors：



Looks like the discrete cosine bases of JPG!...

主成分分析用于图像压缩 PCA application: Image Compression

PCA compression: 144D) 1D



主成分分析用于滤除噪声 PCA application: Noise Filtering

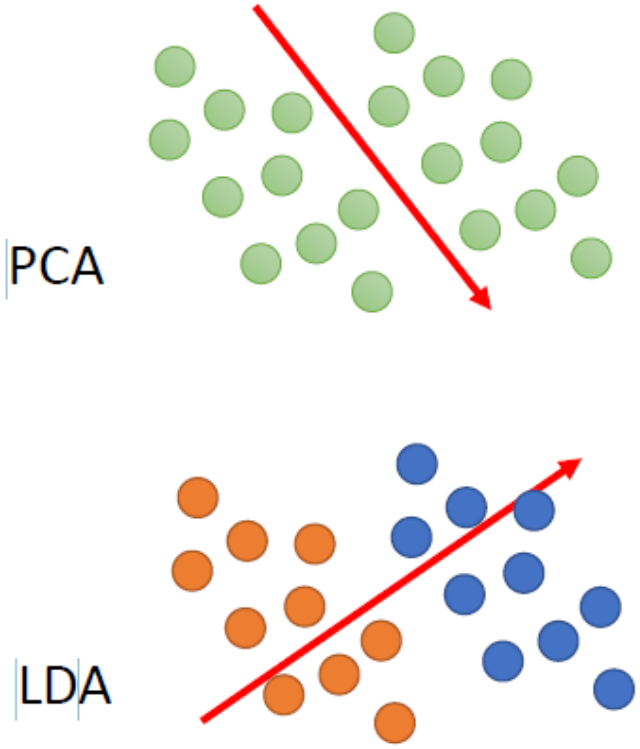
Noisy image

主成分分析用于滤除噪声 PCA application: Noise Filtering

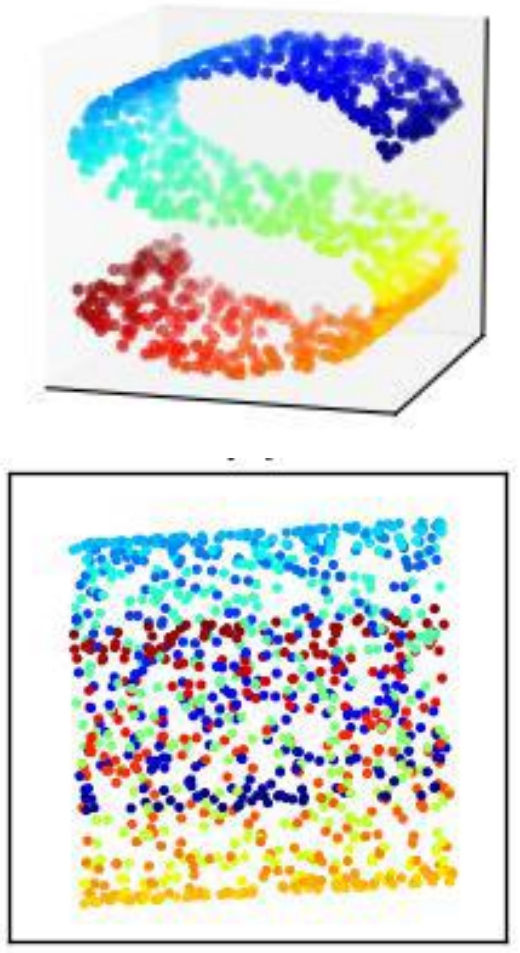
Denoised image using 15 PCA components

主成分分析弱点 Weakness of PCA

Unsupervised



Linear:



t-分布领域嵌入算法(t-SNE)

T-distributed Stochastic Neighbor Embedding

Compute similarity between all pairs of x: $S(x^i, x^j)$

Compute similarity between all pairs of z: $S'(z^i, z^j)$

$$P(x^j|x^i) = \frac{S(x^i, x^j)}{\sum_{k \neq i} S(x^i, x^k)}$$

$$Q(z^j|z^i) = \frac{S'(z^i, z^j)}{\sum_{k \neq i} S'(z^i, z^k)}$$

Find a set of z making the two distributions as close as possible

$$L = \sum_i KL\left(P(*|x^i)||Q(*|z^i)\right)$$

$$= \sum_i \sum_j P(x^j|x^i) log \frac{P(x^j|x^i)}{Q(z^j|z^i)}$$

t-分布领域嵌入算法(t-SNE)

T-distributed Stochastic Neighbor Embedding

X → Z

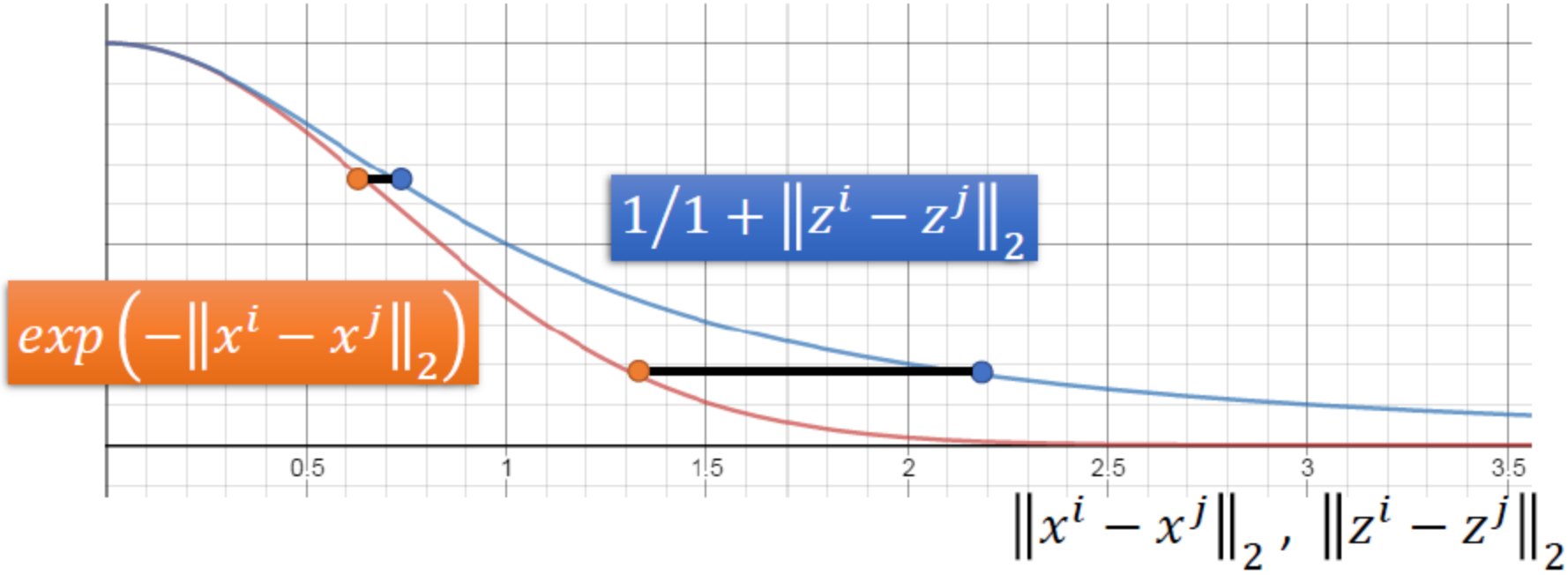
SNE:

$$S(x^i, x^j) = exp\left(-\left\|x^i - x^j\right\|_2\right)$$

$$S'(z^i, z^j) = exp\left(-\left\|z^i - z^j\right\|_2\right)$$

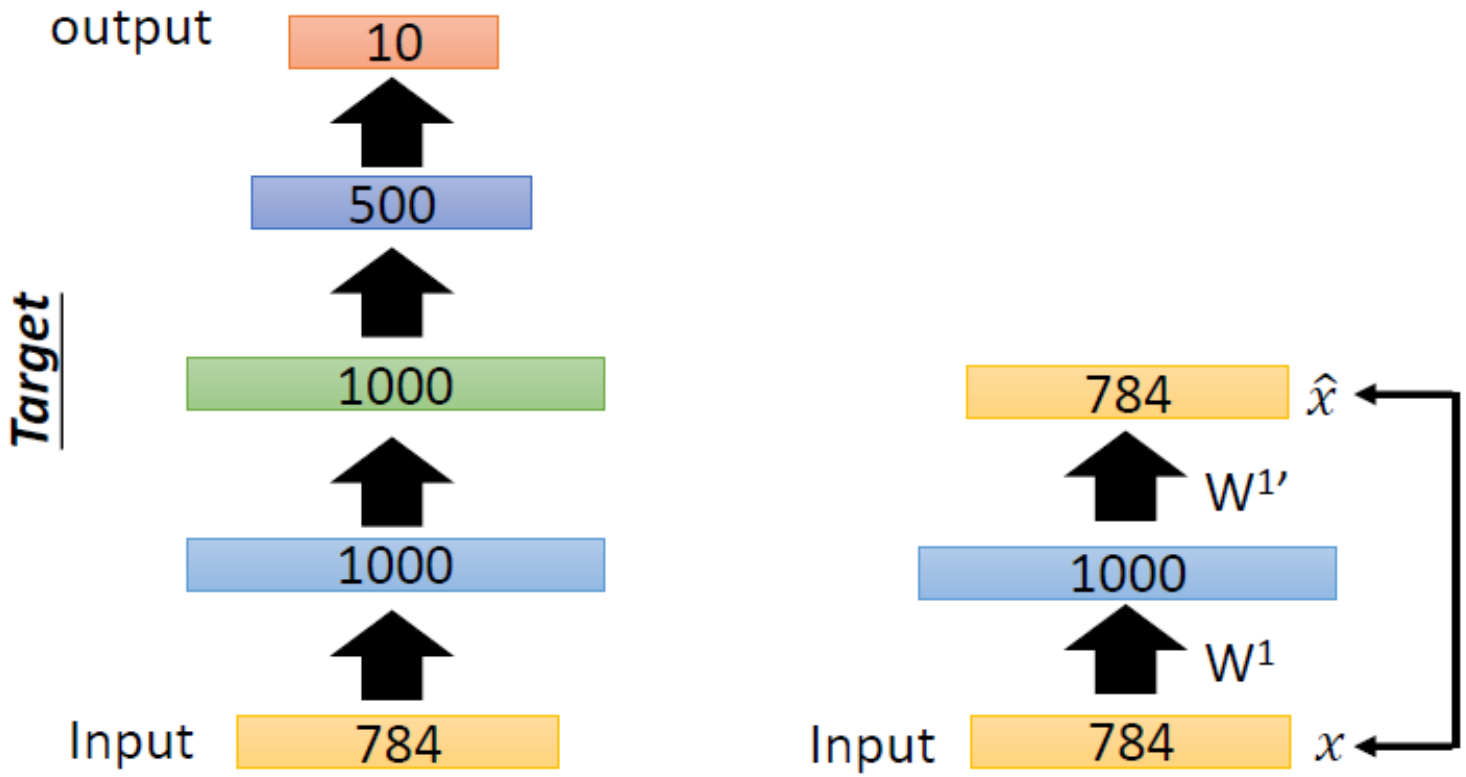
t-SNE:

$$S'(z^i, z^j) = 1/1 + \left\|z^i - z^j\right\|_2$$



自编码器 Auto-encoder

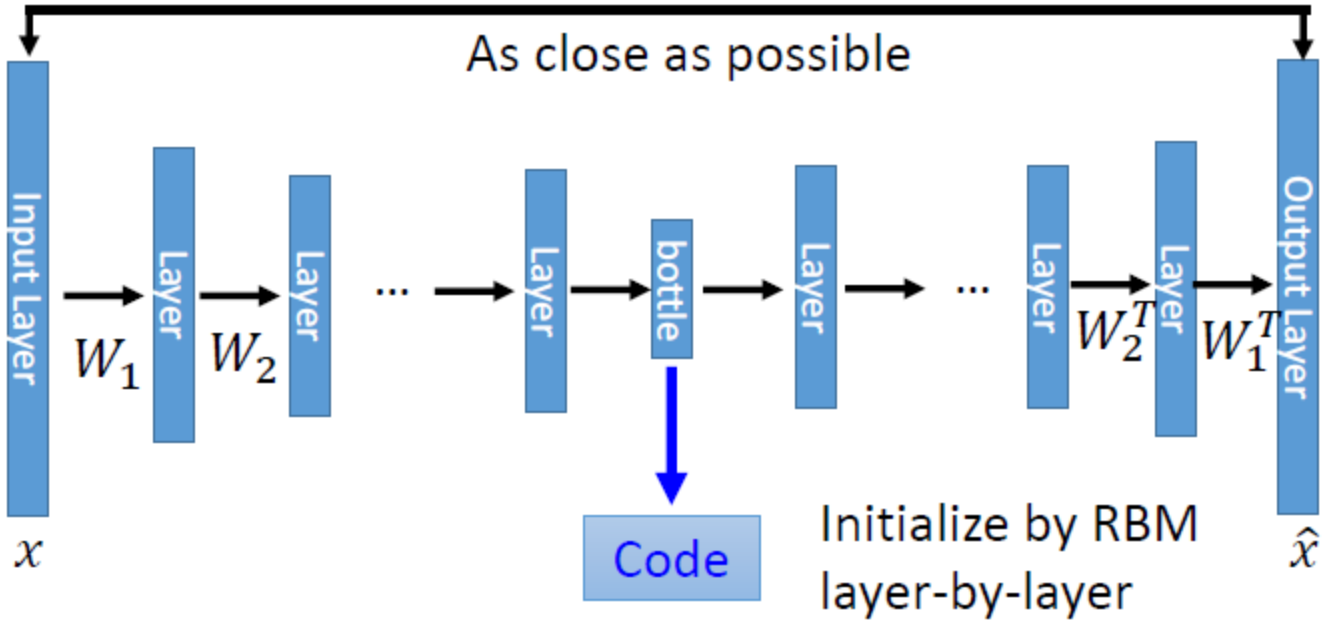
– Greedy Layer-wise Pre-training again



Reference: Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." *Science* 313.5786 (2006): 504-507

自编码器 Auto-encoder

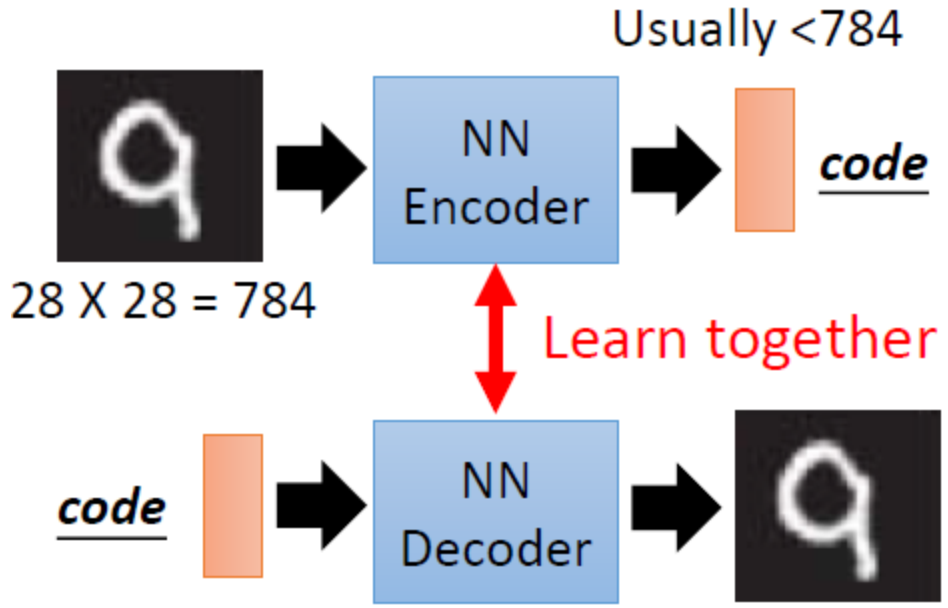
- An auto-encoder is an artificial neural net used for unsupervised learning of efficient codings.
- Greedy Layer-wise Pre-training again



use neural networks to recover the data

Reference: Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." *Science* 313.5786 (2006): 504-507

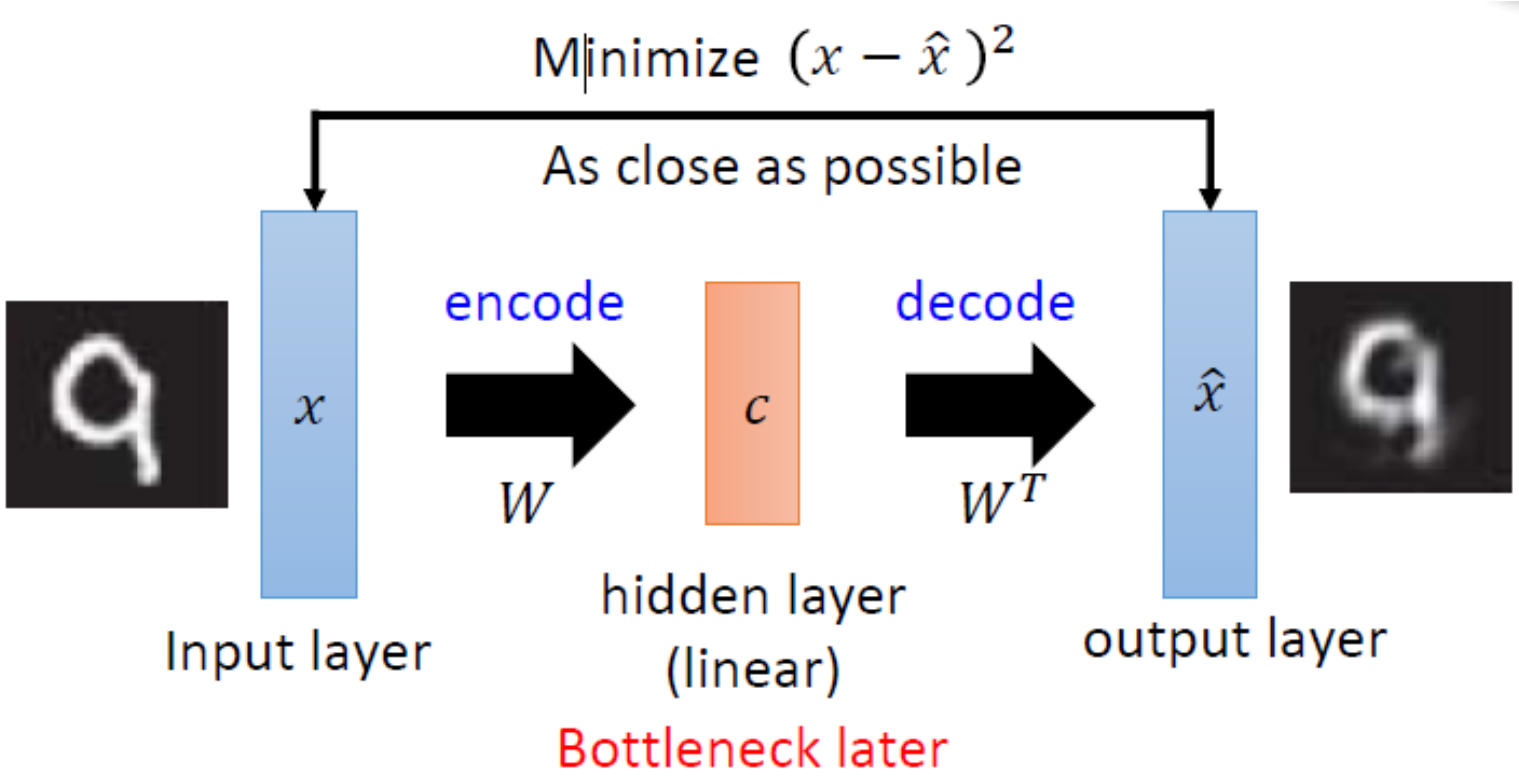
学习自编码器 Learning Auto-encoder



Compact representation of the input object

Can reconstruct the original object

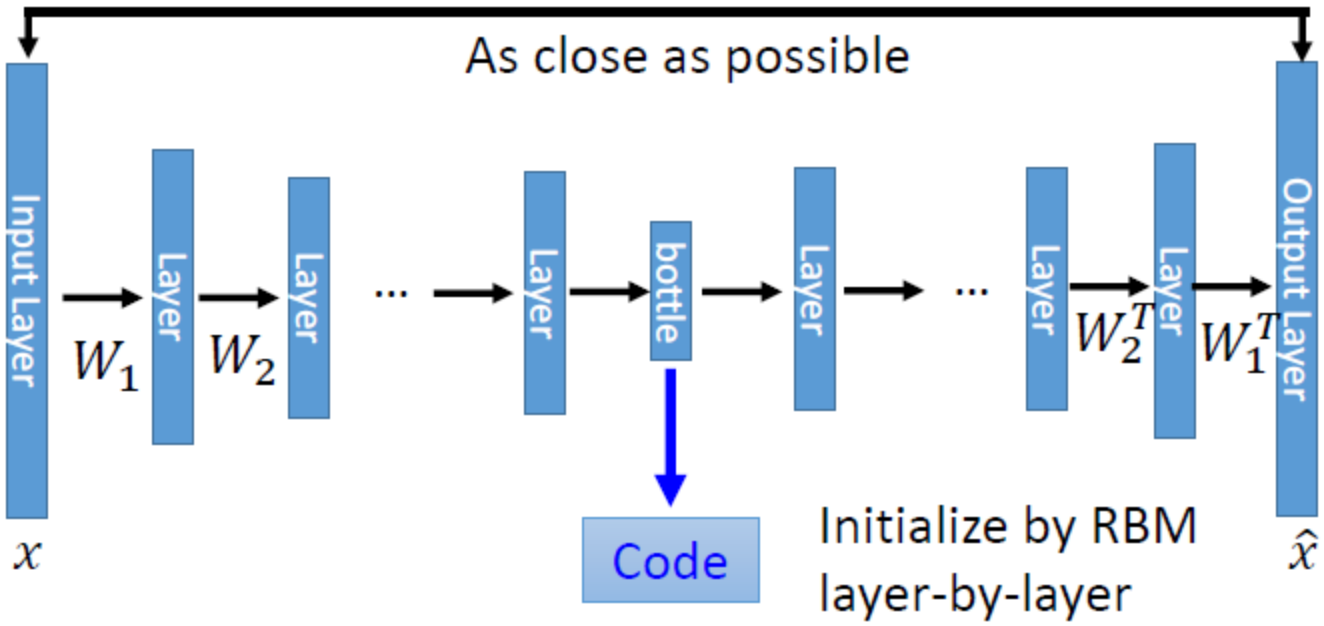
学习PCA Learning PCA



Output of the hidden layer is the code

深度自编码器 Deep Auto-encoder

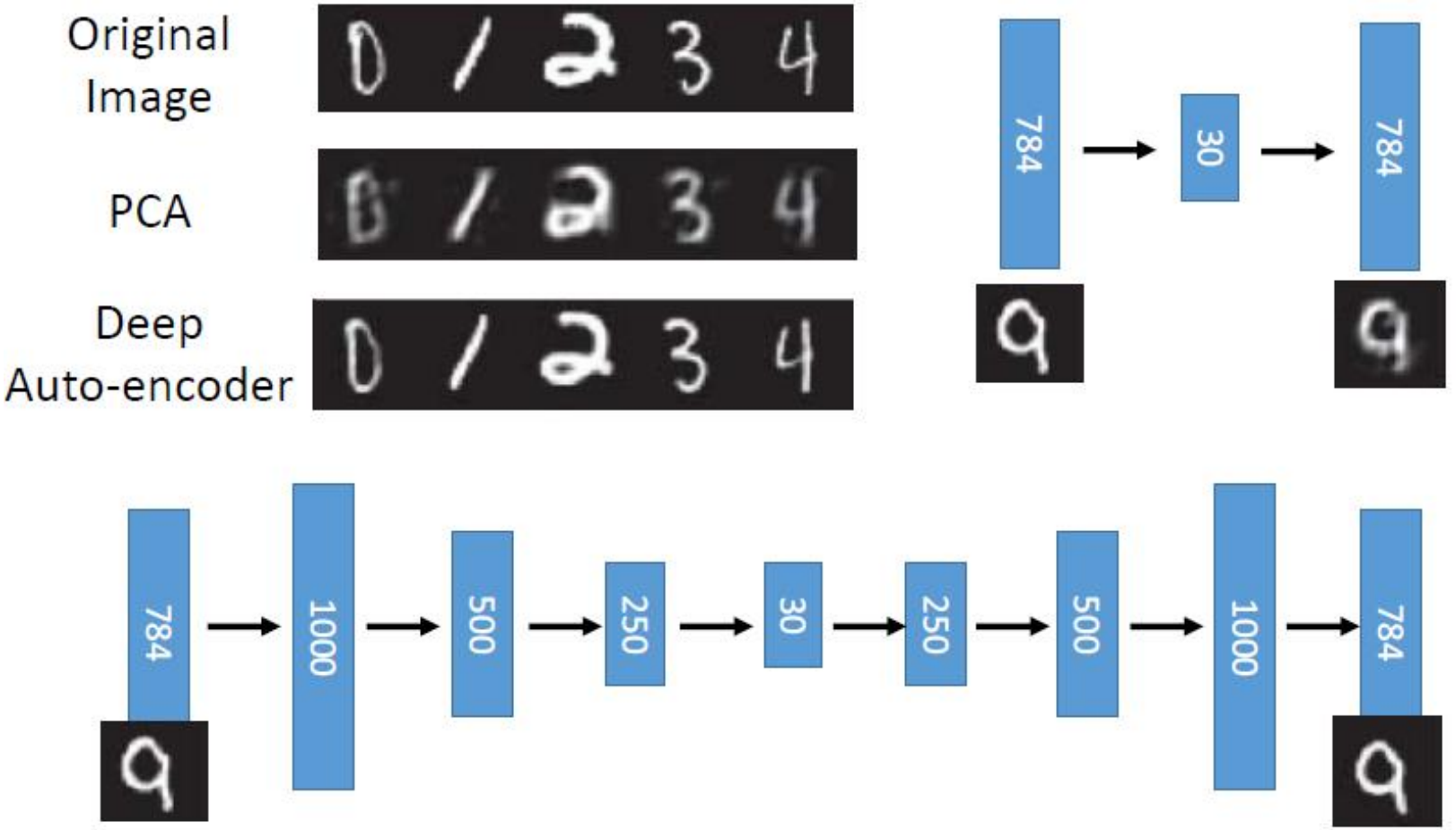
The auto-encoder can be deep



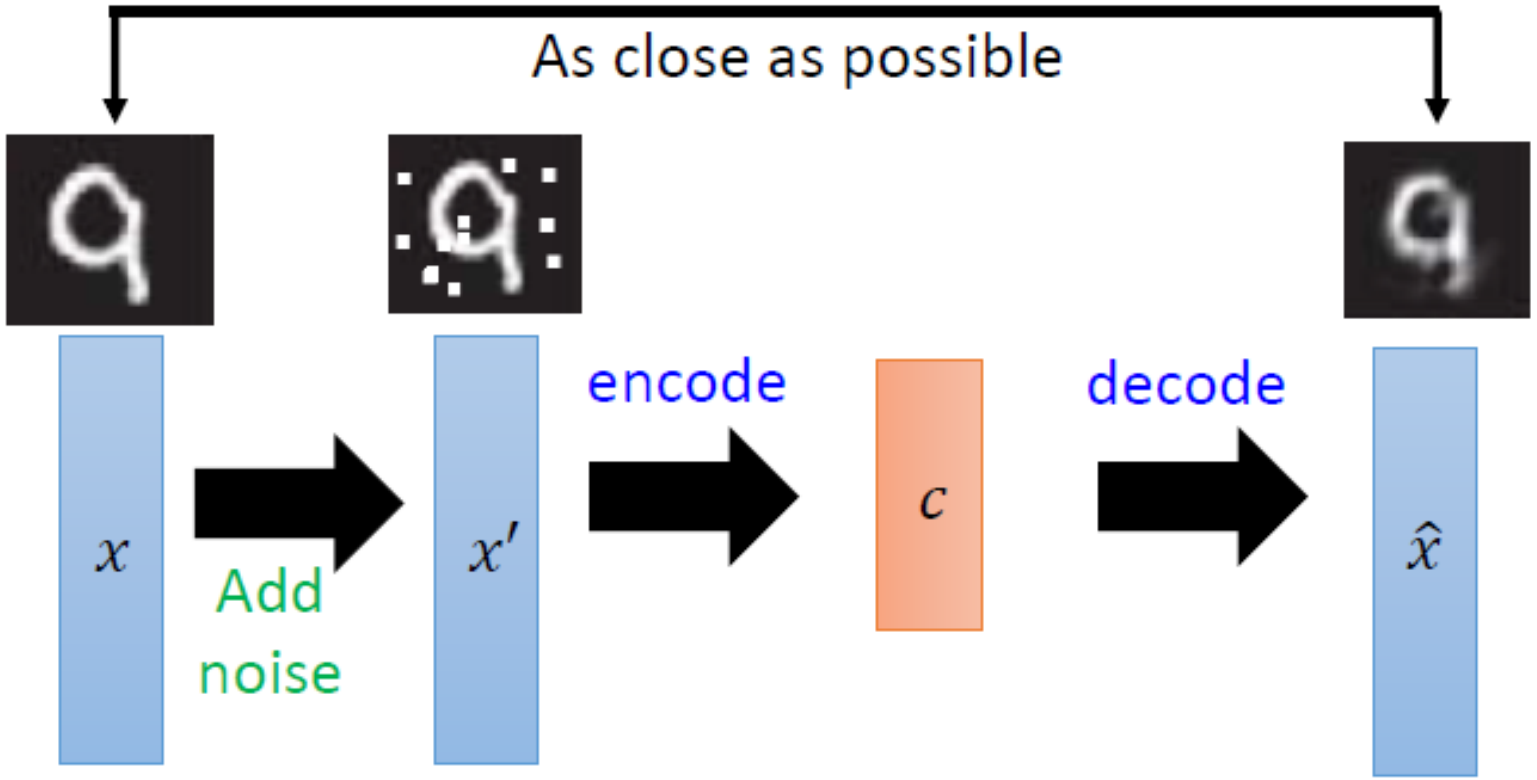
use neural networks to recover the data

Reference: Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." *Science* 313.5786 (2006): 504-507

深度自编码器 Deep Auto-encoder

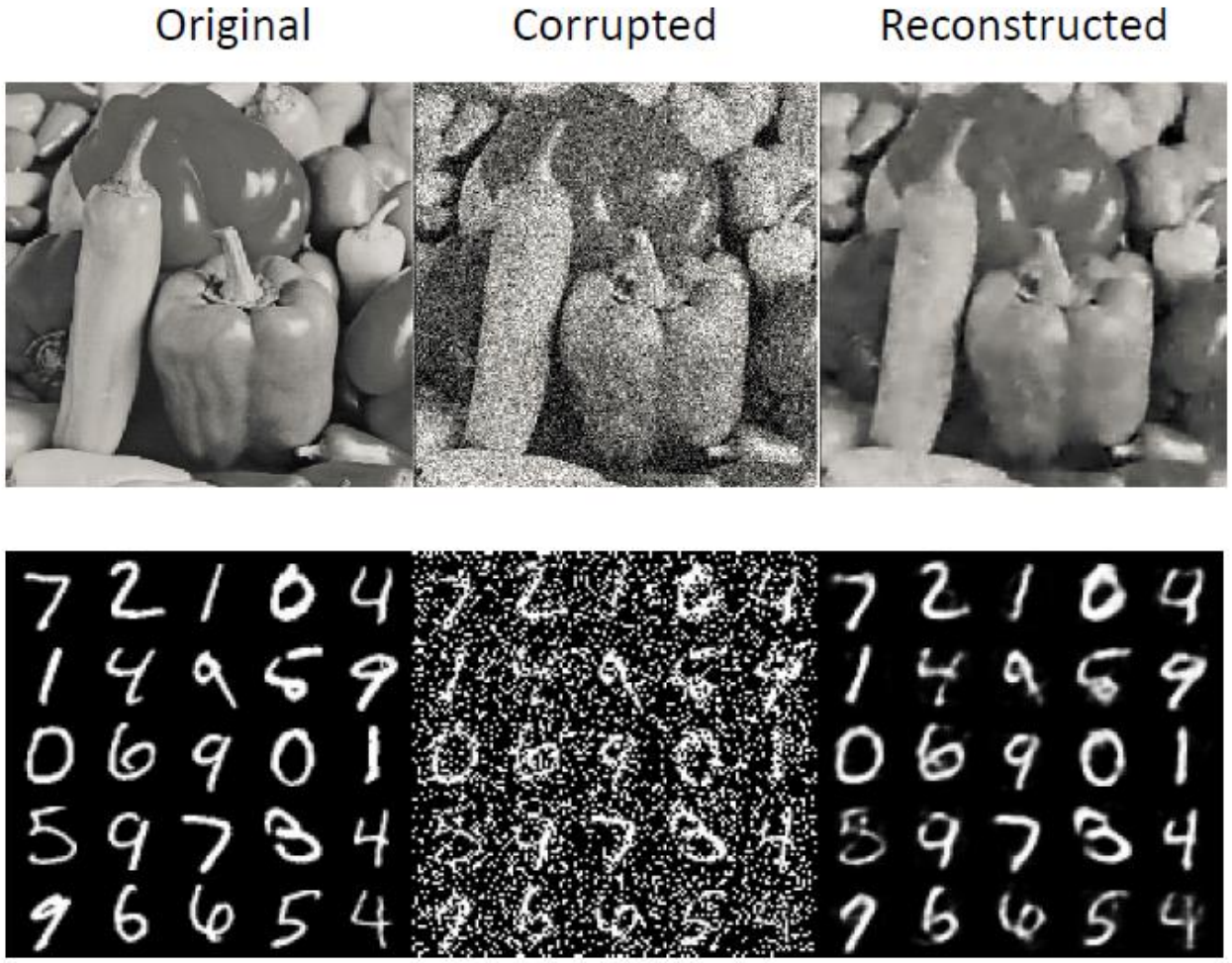


深度自编码器用于去噪声 Denoising Auto-Encoder Examples



Vincent, Pascal, et al. "Extracting and composing robust features with denoising autoencoders." *ICML*, 2008.

深度自编码器用于去噪声 Denoising Auto-Encoder Examples

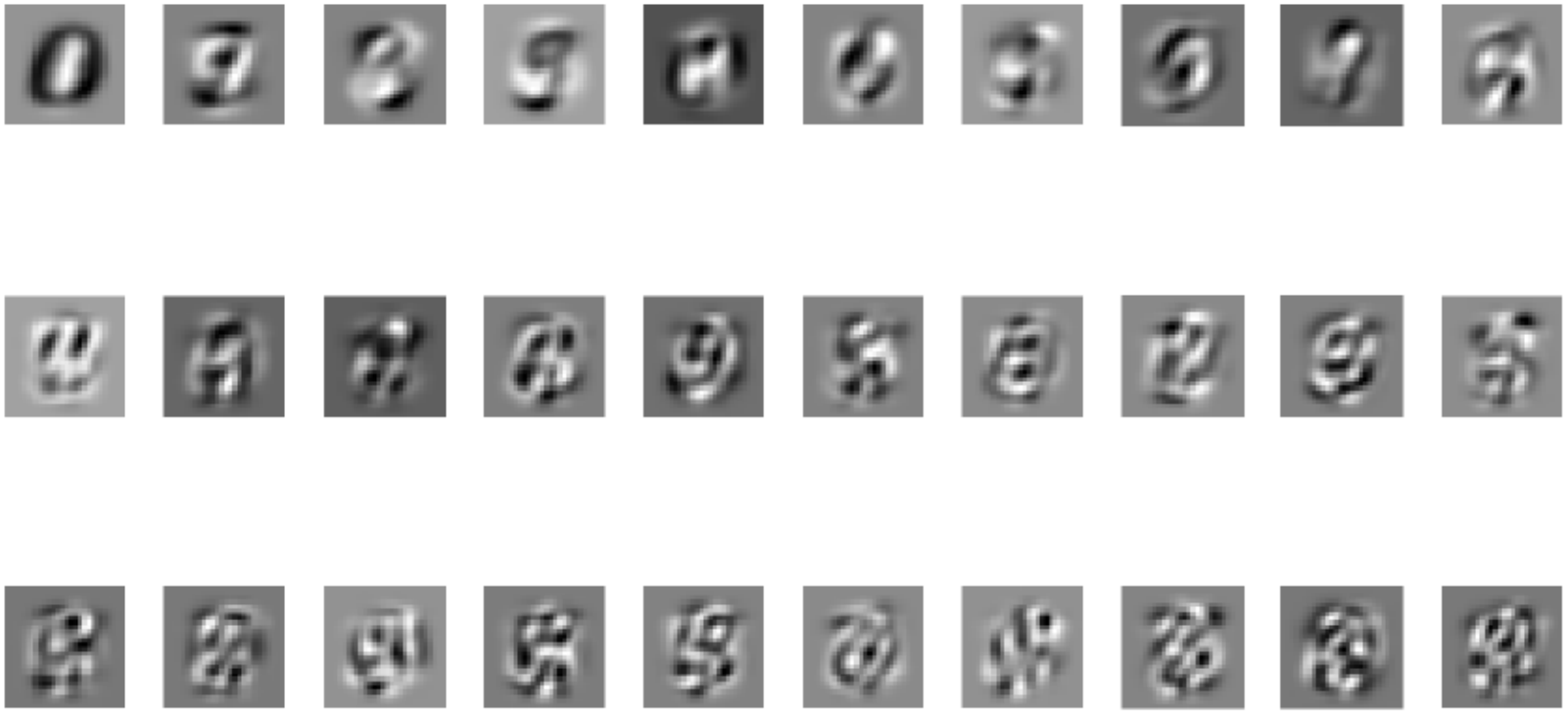


PCA人脸分析 PCA on Face

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

images

30 components:



Eigen-digits

PCA人脸分析 PCA on Face

30 components:

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

Eigen-face

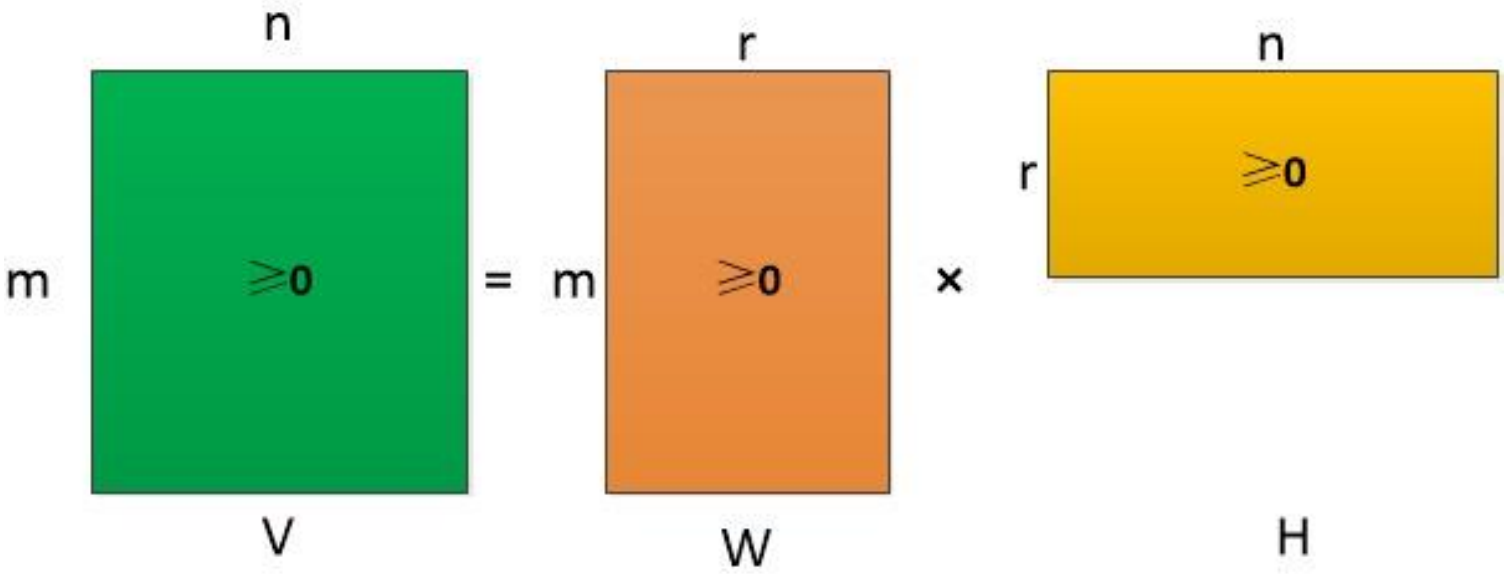
PCA结果 What happens to PCA

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

Can be any real number

- PCA involves adding up and subtracting some components (images)
  - Then the components may not be “parts of digits”
- Non-negative matrix factorization (NMF)
  - Forcing a_1, a_2 …… be non-negative
    - additive combination
  - Forcing w^1, w^2…… be non-negative
    - More like “parts of digits”
- Ref: Daniel D. Lee and H. Sebastian Seung. "Algorithms for non-negative matrix factorization." *Advances in neural information processing systems*. 2001.

非负矩阵分解（NMF）Non-negative Matrix Factorization

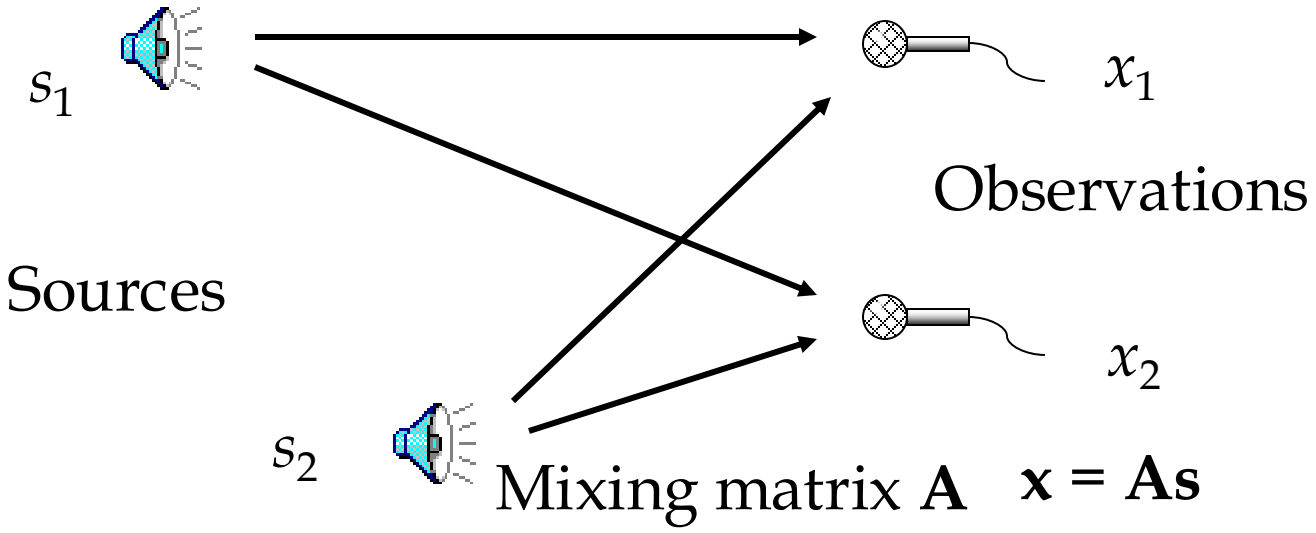


Non-negativity constraint offers physical and real-world interpretability!

$$argmin\frac{1}{2}\|X-WH\|^2=\frac{1}{2}\sum_{i,j}(X_{ij}-WH_{ij})^2$$

$$argmin\,J(W,H)=\sum_{ij}\left(X_{ij}ln\frac{X_{ij}}{WH_{ij}}-X_{ij}+WH_{ij}\right)$$

鸡尾酒会问题 Cocktail-party problem



鸡尾酒会问题 Cocktail-party problem

Model $$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) \\ x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t) \end{aligned}$$

We observe

$$\begin{pmatrix} x_1(1) \\ x_2(1) \end{pmatrix}, \begin{pmatrix} x_1(2) \\ x_2(2) \end{pmatrix}, \ldots, \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

We want

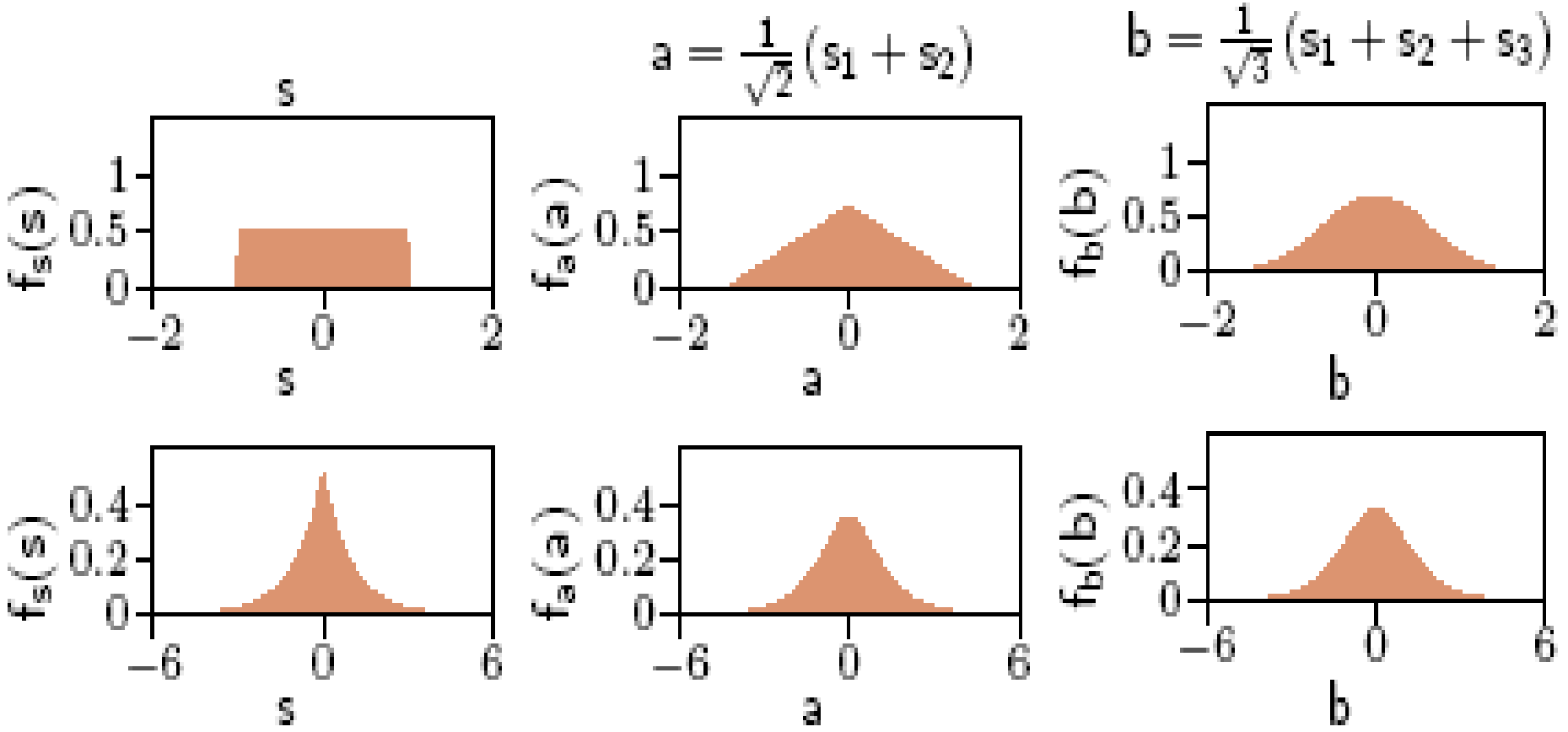
$$\begin{pmatrix} s_1(1) \\ s_2(1) \end{pmatrix}, \begin{pmatrix} s_1(2) \\ s_2(2) \end{pmatrix}, \ldots, \begin{pmatrix} s_1(t) \\ s_2(t) \end{pmatrix}$$

But we don't know $\{a_{ij}\}$, nor $\{s_i(t)\}$

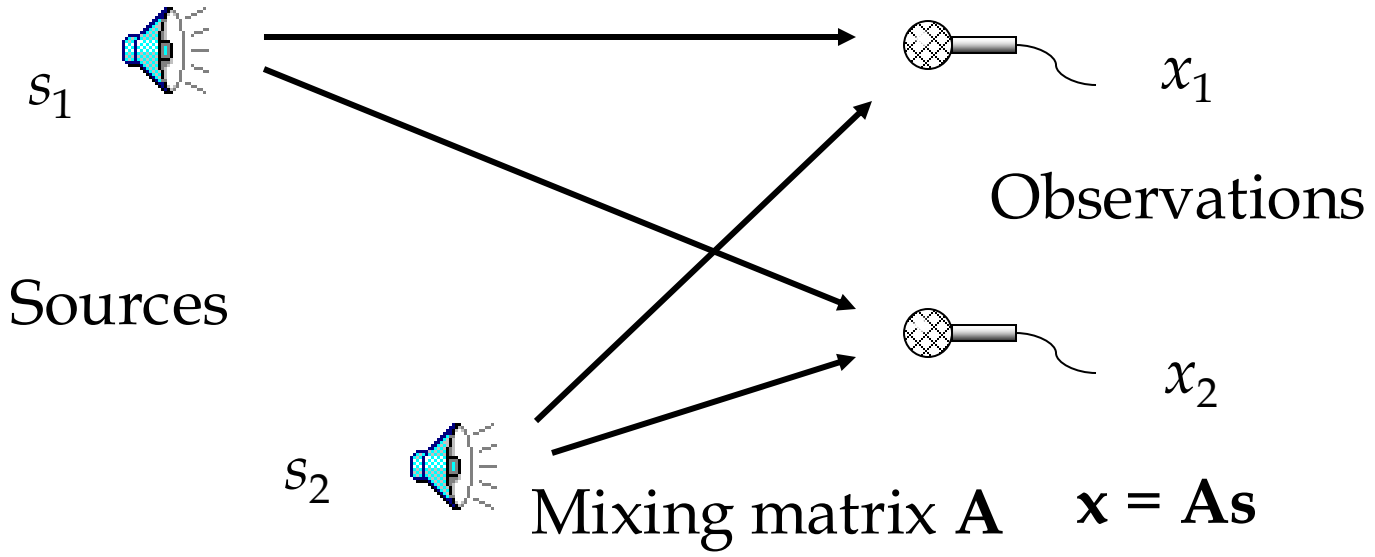
Goal: Estimate $\{s_i(t)\}$, (and also $\{a_{ij}\}$)

中心极限定理 Central Limit Theorem

The sum of independent variables converges to the normal distribution



ICA 假设 ICA Assumptions



1. $\mathbf{s}_1,\ldots\mathbf{s}_K$ statistical Independence
2. Nongaussianity (at most one component is Gaussian)
3. A is column full-rank

ICA 原则 ICA Principal (Non-Gaussian is Independent)

$$y = w^T x = w^T As = z^T s$$

- Key to estimating A is non-gaussianity
- y is a linear combination of s_i, with weights given by z_i.
- $z^T s$ is more gaussian than either of s_i. AND becomes least gaussian when its equal to one of s_i.
- So we could take ***w*** as a vector which maximizes the non-gaussianity of $\boldsymbol{w^T x}$.
- Such a ***w*** would correspond to a ***z*** with only one non zero comp. So we get back the s_i.

非高斯性衡量 Measures of Non-Gaussianity

- Kurtotis : gauss=0 (sensitive to outliers)

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2$$

Subgaussian: negative
Supergaussian: positive

- Entropy : gauss=largest

$$H(y) = -\int f(y) \log f(y) dy$$

- Neg-entropy : gauss = 0 (difficult to estimate)

$$J(y) = H(y_{gauss}) - H(y)$$

- Approximations

$$J(y) = \frac{1}{12} E\{y^2\}^2 + \frac{1}{48} kurt(y)^2$$

$$J(y) \approx [E\{G(y)\} - E\{G(v)\}]^2$$

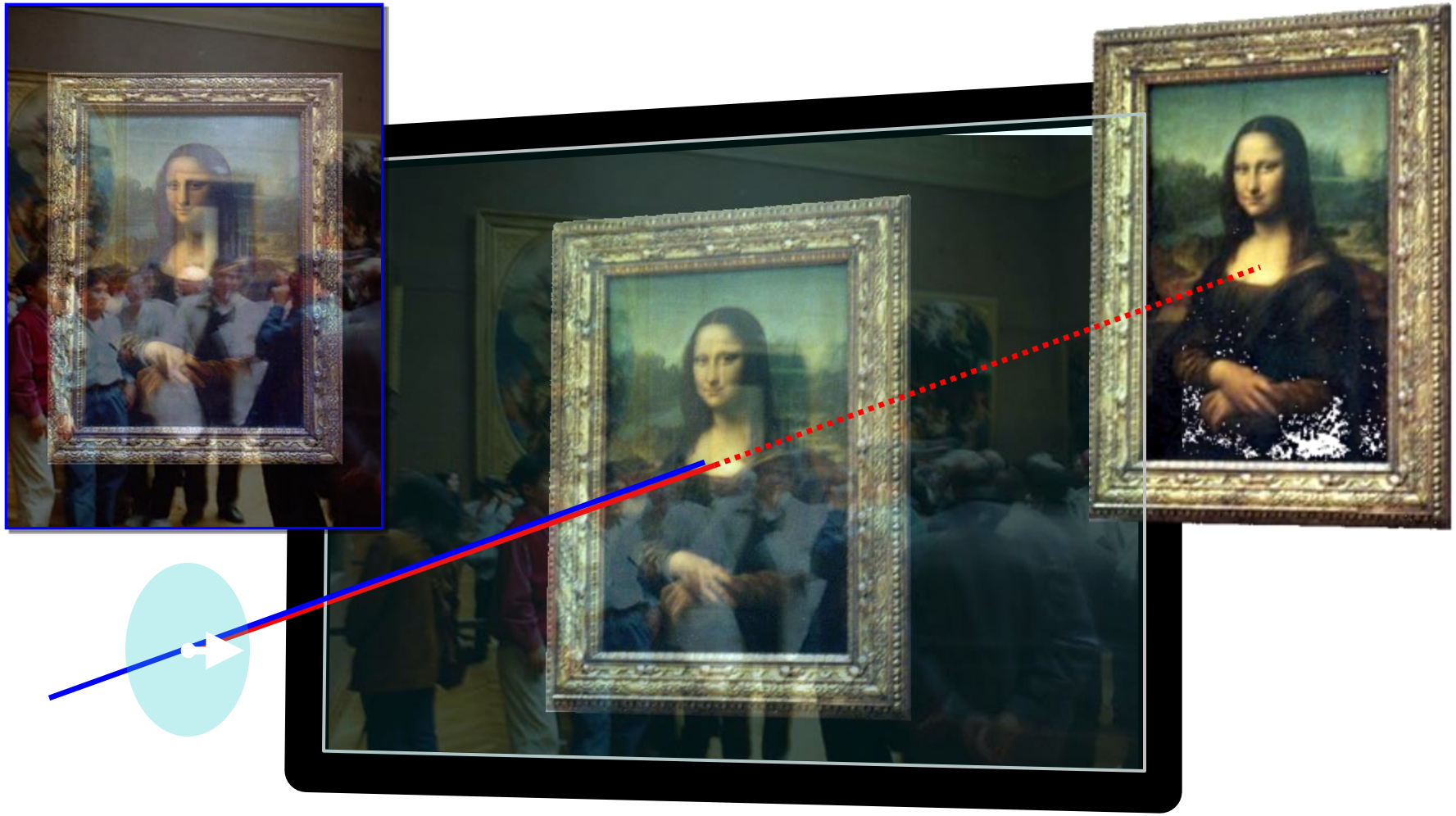
where *v* is a standard gaussian random variable and :

$$G(y) = \frac{1}{a} \log \cosh(a.y)$$

$$G(y) = -\exp(-a.u^2/2)$$

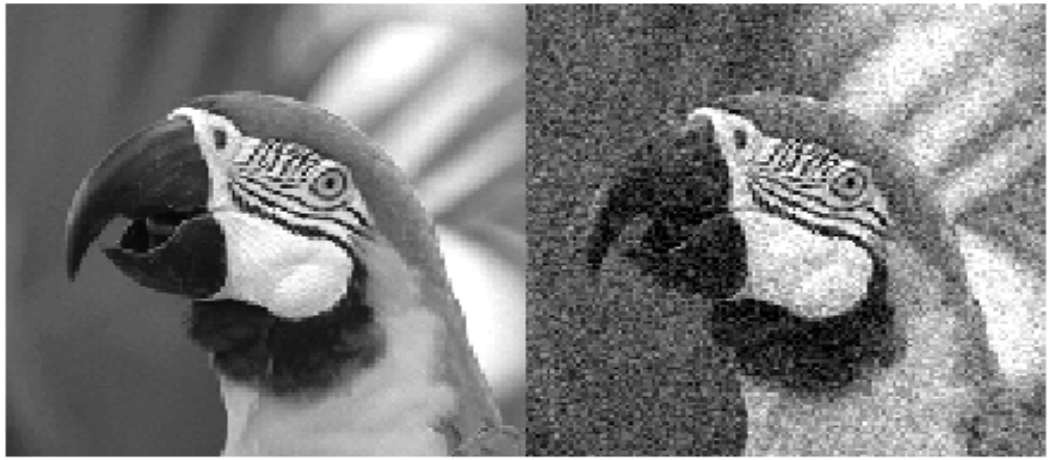
同济大学 TONGJI UNIVERSITY

图像问题ICA求解 ICA Application on image



图像问题ICA求解 ICA Application on image

Original image



Noisy image

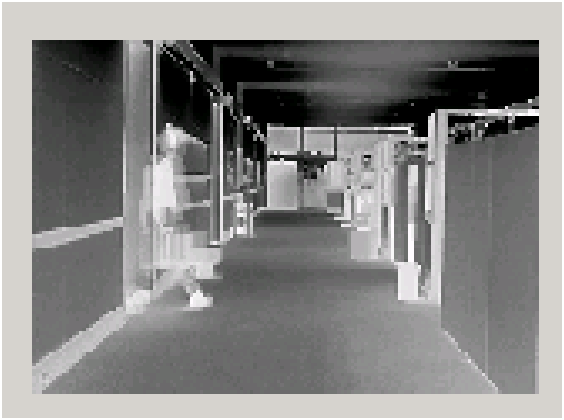
Wiener filtering

ICA filtering

视频问题ICA求解 ICA Application on Video

Source images

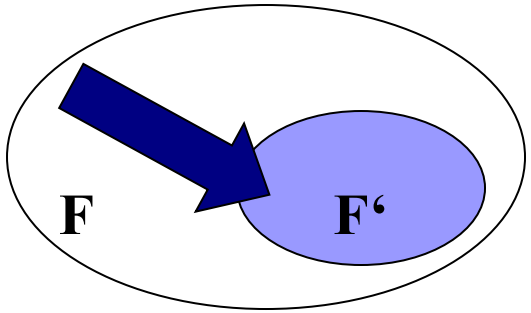
视频问题ICA求解 ICA Application on Video



ICs

特征选择 Feature selection

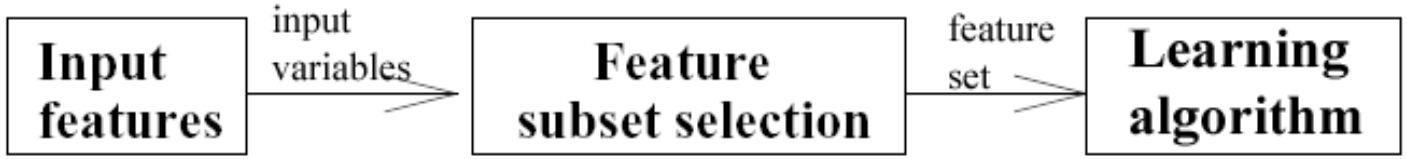
- Find the optimal feature subset.



- Filter Methods
- Wrapper Methods
- Embedded Methods

过滤法 Filter Methods

Select subsets of variables as a pre-processing step, independently of the used classifier!!



Features could be evaluated by

- Distance
- Information
- Dependency (correlation)
- …

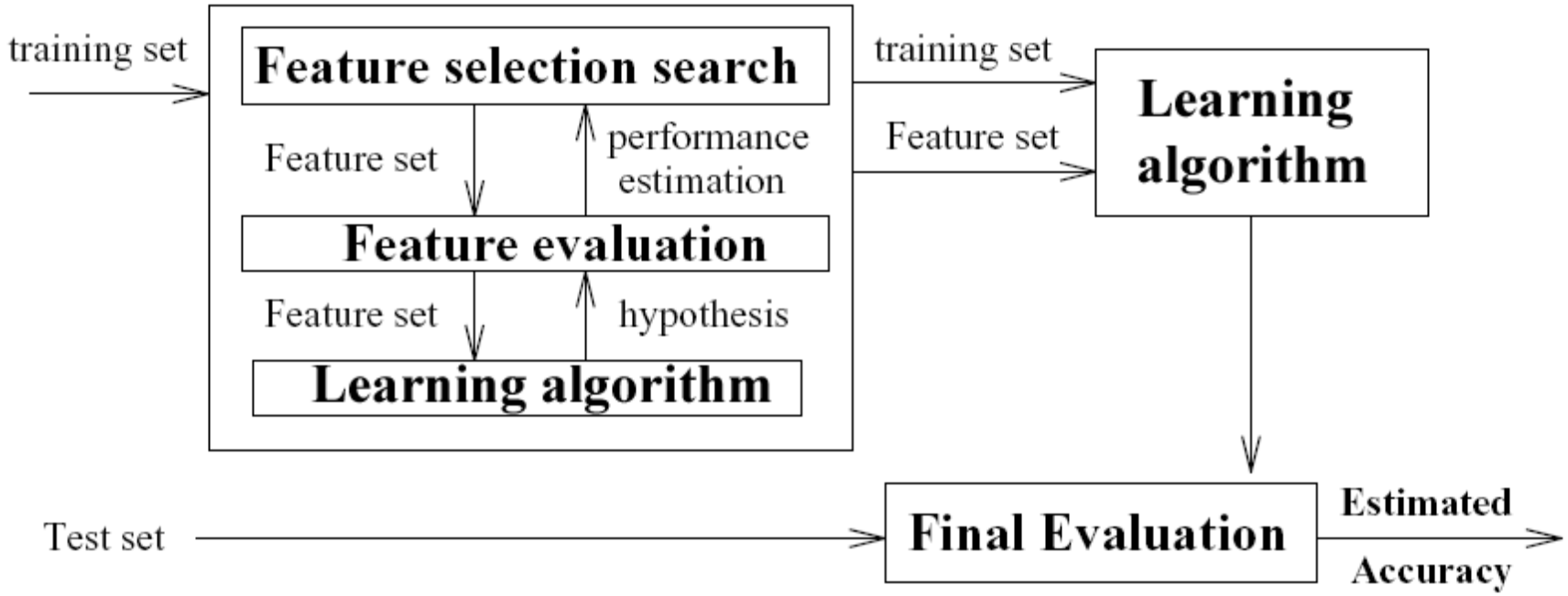
过滤法 Filter Methods

- Relief (Relevant Features)

$$\delta^j = \sum_i -\mathrm{diff}(x_i^j, x_{i,\mathrm{nh}}^j)^2 + \mathrm{diff}(x_i^j, x_{i,\mathrm{nm}}^j)^2$$

- Fischer's ratio. $\dfrac{|S_b|}{|S_w|} = \dfrac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$
- Information gain $\mathrm{Gain}(A) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \dfrac{|D^v|}{|D|}\mathrm{Ent}(D^v)$
- Coefficient scores $\Re(i) = \dfrac{\mathrm{cov}(X_i, Y)}{\sigma(X_i) \times \sigma(Y)}$
- …

包裹法 Wrapper Methods



Features could be evaluated by classifier error rate (the classifier themselves).

包裹法 Wrapper Methods

- The problem of finding the optimal subset is NP-hard!

- A wide range of heuristic search strategies can be used.
  Two different classes:
  - **Forward selection**
    (start with empty feature set and add features at each step)
  - **Backward elimination**
    (start with full feature set and discard features at each step)

- Predictive power is usually measured on a validation set or by cross-validation
- By using the learner as a black box wrappers are universal and simple!
- Criticism: a large amount of computation is required.

嵌入法 Embedded Methods

- Specific to a given learning machine!
- Performs feature selection (implicitly) in the process of training
- E.g. LASSO [Tibshirani, 1996]

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{w}\|_1$$

- Random Forest and Gradient Boosting