CS540 Introduction to Artificial Intelligence
Lecture 12

Young Wu
Based on lecture slides by Jerry Zhu and Yingyu Liang

June 27, 2019

High Dimensional Data

Motivation

- High dimensional data are training set with a lot of features.

1. Document classification.
2. MEG brain imaging.
3. Handwritten digits (or images in general).

Low Dimension Representation

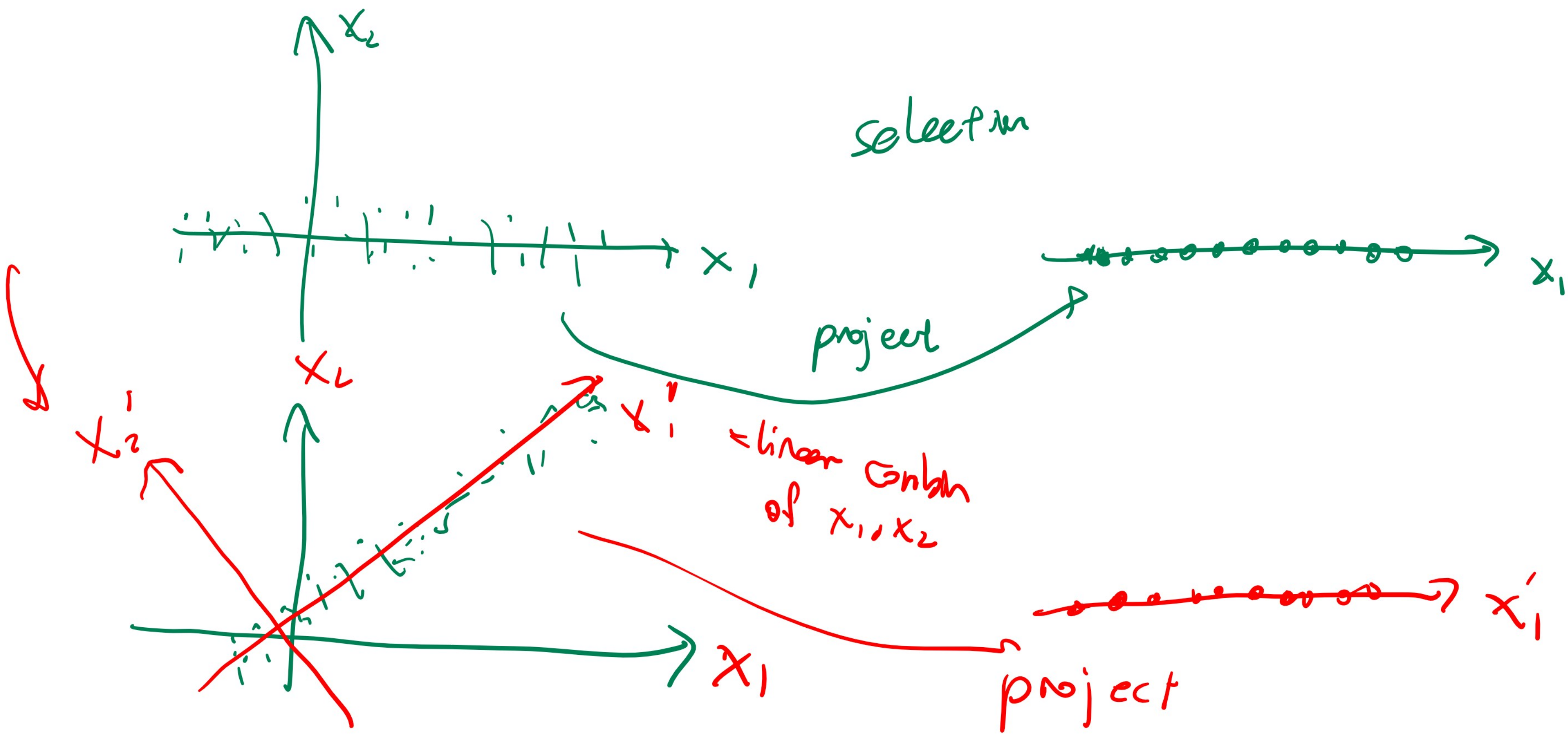
Motivation

- Unsupervised learning techniques are used to find low dimensional representation.

1. Visualization. 2 or 3 features
2. Efficient storage.
3. Better generalization.
4. Noise removal.

Dimension Reduction Diagram

Motivation



Dimension Reduction

Principal Components Variance

- Rotate the axes so that they capture the directions of the greatest variability of data.
- The new axes (orthogonal directions) are principal components.

Principal Component Analysis

Description

- Find the direction of the greatest variability in data, call it $u_{1.}$
- Find the next direction orthogonal to u_1 of the greatest variability, call it $u_{2.}$
- Repeat until there are $u_1, u_2, ..., u_K$.

Orthogonal Directions

Definition

- In Euclidean space (L_2 norm), a unit vector u_k has length 1.

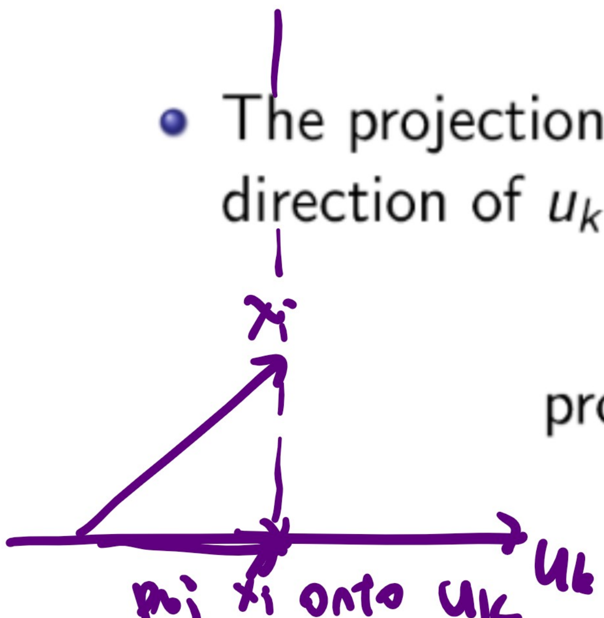
$$\|u_k\|_2 = u_k^T u_k = 1$$

- Two vectors $u_k, u_{k'}$ are orthogonal (or uncorrelated) if the dot product is 0.

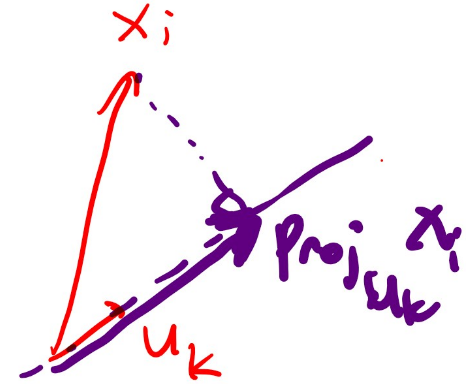
$$u_k \cdot u_{k'} = u_k^T u_{k'} = 0$$

Projection

Definition



- The projection of x_i onto a unit vector u_k is the vector in the direction of u_k that is the closest to x_i.



length

$$\text{proj}_{u_k} x_i = \boxed{\left(\frac{u_k^T x_i}{u_k^T u_k}\right) u_k} = u_k^T x_i u_k$$

unit

unit direction

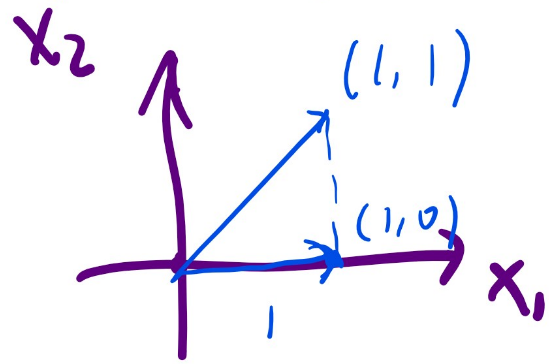
- The length of the projection of x_i onto a unit vector u_k is $u_k^T x_i$.

$$\|\text{proj}_{u_k} x_i\|_2 = u_k^T x_i$$

Project Example, Part I

Quiz (Graded)

- What is the projection of $\begin{bmatrix}1\\1\end{bmatrix}$ onto $\begin{bmatrix}1\\0\end{bmatrix}$?
- A: 1
- B: $\dfrac{1}{\sqrt{2}}$
- C: $\begin{bmatrix}1\\0\end{bmatrix}$
- D: $\begin{bmatrix}\frac{1}{\sqrt{2}}\\0\end{bmatrix}$
- E: $\begin{bmatrix}\sqrt{2}\\0\end{bmatrix}$



$u_k^T x_i \, u_k$

$$(1, 0)\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = (1\cdot1+0\cdot1)\cdot\begin{pmatrix}1\\0\end{pmatrix}$$

$$= \underbrace{1}_{\text{length}}\underbrace{\begin{pmatrix}1\\0\end{pmatrix}}_{\text{direction}} = \begin{pmatrix}1\\0\end{pmatrix}$$

Project Example, Part II

Quiz (Graded)

- What is the projection of $\begin{bmatrix}1\\0\end{bmatrix}$ onto $\begin{bmatrix}1\\1\end{bmatrix}$?
- ~~A: 1~~
- ~~B: $\frac{1}{\sqrt{2}}$~~
- ~~C: $\begin{bmatrix}1\\1\end{bmatrix}$~~
- D: $\begin{bmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{bmatrix}$
- E: $\begin{bmatrix}\frac{1}{2}\\\frac{1}{2}\end{bmatrix}$

$\begin{pmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{pmatrix}$

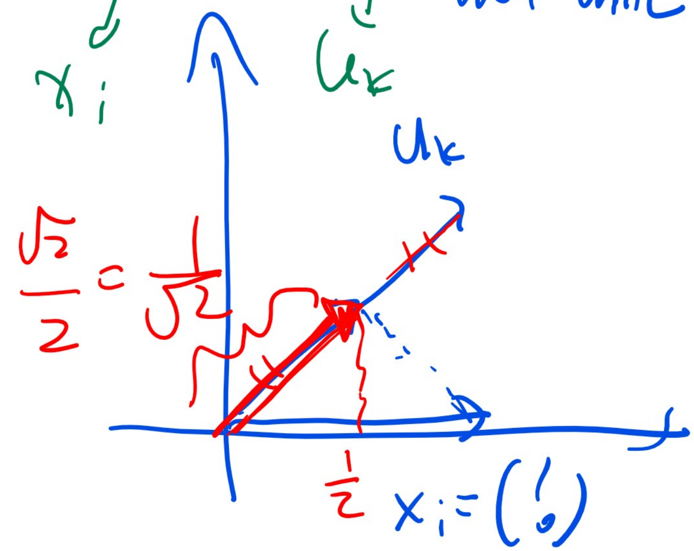
x_i u_k not unit

$$\frac{\begin{pmatrix}1\\1\end{pmatrix}}{\left\|\begin{pmatrix}1\\1\end{pmatrix}\right\|_2} = \frac{\begin{pmatrix}1\\1\end{pmatrix}}{\sqrt{(1,1)\begin{pmatrix}1\\1\end{pmatrix}}}$$

$\sqrt{u_k^T u_k}$

① convert $\begin{pmatrix}1\\1\end{pmatrix}$ to unit vector

$\hat{u}_k^T x_i \hat{u}_k^T$



② $\frac{u_k^T x_i}{u_k^T u_k} u_k$

$$\frac{(1,1)\begin{pmatrix}1\\0\end{pmatrix}}{(1,1)\begin{pmatrix}1\\1\end{pmatrix}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}\frac{1}{2}\\\frac{1}{2}\end{pmatrix}$$

$\hat{u}_k = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ → unit vector of u_k

$\boxed{\hat{u}_k^T x_i \, \hat{u}_k} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)\begin{pmatrix} 1 \\ 0 \end{pmatrix}\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$

only works for unit $\hat{u}_k$

Variance

Definition

- The sample variance of a data set $\{x_1, x_2, ..., x_n\}$ is the sum of the squared distance from the mean.

$$\begin{pmatrix} x_{i1}^2 & x_{i1}x_{i2} & x_{i1}x_{i3} \\ x_{i2}x_{i1} & x_{i2}^2 & x_{i2}x_{i3} \\ & \cdots & \end{pmatrix}$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}$$

$m \times 1$ vector →

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{im} \end{pmatrix}$$

$m \times m$ matrix →

$$\hat{\Sigma} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu})(x_i - \hat{\mu})^T$$

$$\begin{pmatrix} x_{i1} \\ \vdots \\ x_{im} \end{pmatrix}(x_{i1} \ldots x_{im})$$

Covariance matrix.

~~***~~ NOT $(x_i - \hat{\mu})^T(x_i - \hat{\mu})$

Normalization

Definition

- Normalize the data by subtracting the mean, then the variance expression can be simplified.

$$x_i = x_i - \mu$$

$$\hat{\Sigma} = \frac{1}{n-1}\sum_{i=1}^{n} x_i x_i^T = \frac{1}{n-1} X^T X$$

Maximum Variance Directions

Definition

After a lot of math

compute variance along direction u_k.

- The goal is to find the direction that maximizes the projected variance.

$u_k^T \lambda u_k$

$\lambda \cancel{u_k^T u_k}$

$$\max_{u_k} u_k^T \hat{\Sigma} u_k \text{ such that } u_k^T u_k = 1$$

$$\Rightarrow \max_{u_k} u_k^T \hat{\Sigma} u_k - \lambda u_k^T u_k$$

$$\boxed{\Rightarrow \hat{\Sigma} u_k = \lambda u_k}$$

a lot of math again.

$\Longrightarrow$ max λ

find eigen values eigenvectors

PC are eigenvectors of covariance matrix.

Eigenvalue

Definition

- The λ represents the projected variance.

$$u_k^T \hat{\Sigma} u_k = u_k^T \lambda u_k = \lambda$$

$\lambda_k \uparrow$ variance in $u_k \uparrow$

- The larger the variance, the larger the variability in direction u_k. There are m eigenvalues for a symmetric positive semidefinite matrix (for example, $X^T X$ is always symmetric PSD). Order the eigenvectors u_k by the size of their corresponding eigenvalues λ_k.

$k \ll m$

$$\lambda_1 \geqslant \lambda_2 \geqslant \dots \overset{\lambda_k}{} \geqslant \lambda_m$$

$$\begin{array}{ccc} \downarrow & \downarrow & \downarrow \\ u_1 & u_2 & u_k \end{array}$$

k principal components $\longrightarrow$

Eigenvalue Algorithm

Definition

- Solving eigenvalue using the definition (characteristic polynomial) is computationally inefficient.

$$\left(\hat{\Sigma} - \lambda_k I\right) u_k = 0 \Rightarrow \det \left(\hat{\Sigma} - \lambda_k I\right) = 0$$

- There are many fast eigenvalue algorithms that computes the spectral (eigen) decomposition for real symmetric matrices. Columns of Q are unit eigenvectors and diagonal elements of D are eigenvalues.

$$\begin{aligned} \hat{\Sigma} &= PDP^{-1}, D \text{ is diagonal} \\ &= QDQ^T, \text{ if } Q \text{ is orthogonal, i.e. } Q^TQ = I \end{aligned}$$

Spectral Decomposition Example

Quiz (Participation)

- Given the following spectral decomposition of $\hat{\Sigma}$, what is the first principal component?

v_1 $\quad$ λ_1 λ_2 λ_1 $\quad$ largest λ

$$\hat{\Sigma} = \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}^{-1}}_{P^{-1}}$$

- A: $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$, B: $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$, C: $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, D: $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$, E: $\begin{bmatrix} -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

unit vector

Principal Component Analysis

Algorithm

- Input: instances: $\{x_i\}_{i=1}^{n}$, the number of dimensions after reduction $K < m$.
- Output: K principal components.
- Find the largest K eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant ... \geqslant \lambda_K$.
- Return the corresponding unit orthogonal eigenvectors $u_1, u_2 ... u_K$.

Reduced Feature Space

Discussion

- The original feature space is m dimensional.

$$(x_{i1}, x_{i2}, \ldots, x_{im})^T$$

m

- The new feature space is K dimensional.

$$\left(u_1^T x_i, u_2^T x_i, \ldots, u_K^T x_i\right)^T$$

K

- Other supervised learning algorithms can be applied on the new features.

Reduced Space Example

Quiz (Graded)

Q6

$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$ $\quad$ $\begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$

- 2017 Fall Final Q10
- If $u_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}^T$ and $u_2 = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}^T$. If one original data is $x = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T$. What is the new representation?
- A: $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$, B: $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$, C: $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{2}} \end{bmatrix}$, D: $\begin{bmatrix} \frac{3}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$, E: $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} \end{bmatrix}$

$(u_1^T x, \quad u_2^T x)$

Number of Dimensions

Discussion

- There are a few ways to choose the number of principal components K.
- K can be selected given prior knowledge or requirement.
- K can be the number of non-zero eigenvalues.
- K can be the number of eigenvalues that are large (larger than some threshold).

λ_k = variance in u_k direction

< 0.1

remove k: $\lambda_k < 0.1$

Reconstruction Error

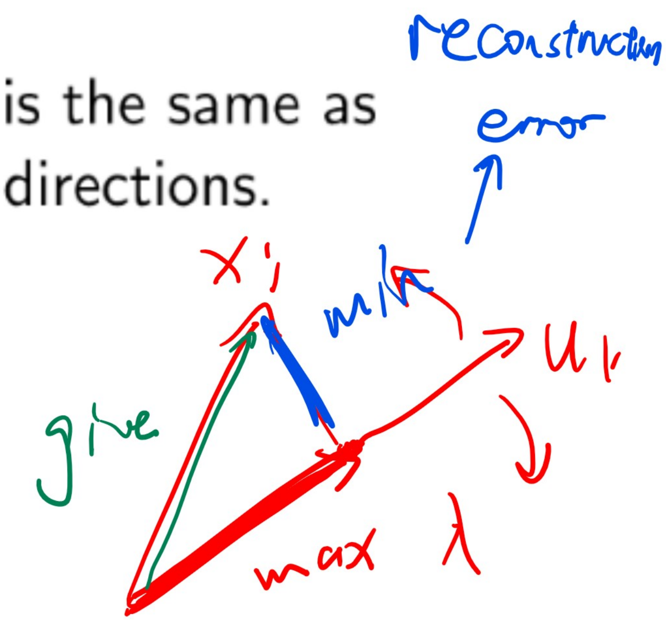
Discussion

- Reconstruction error is the squared error (distance) between the original data and its projection onto u_k.

$$\left\| x_i - \left(u_k^T x_i \right) u_k \right\|^2$$

- Finding the variance maximizing directions is the same as finding the reconstruction error minimizing directions.

$$\frac{1}{n} \sum_{i=1}^{n} \left\| x_i - \left(u_k^T x_i \right) u_k \right\|^2$$



Reconstruction Error Diagram

Discussion

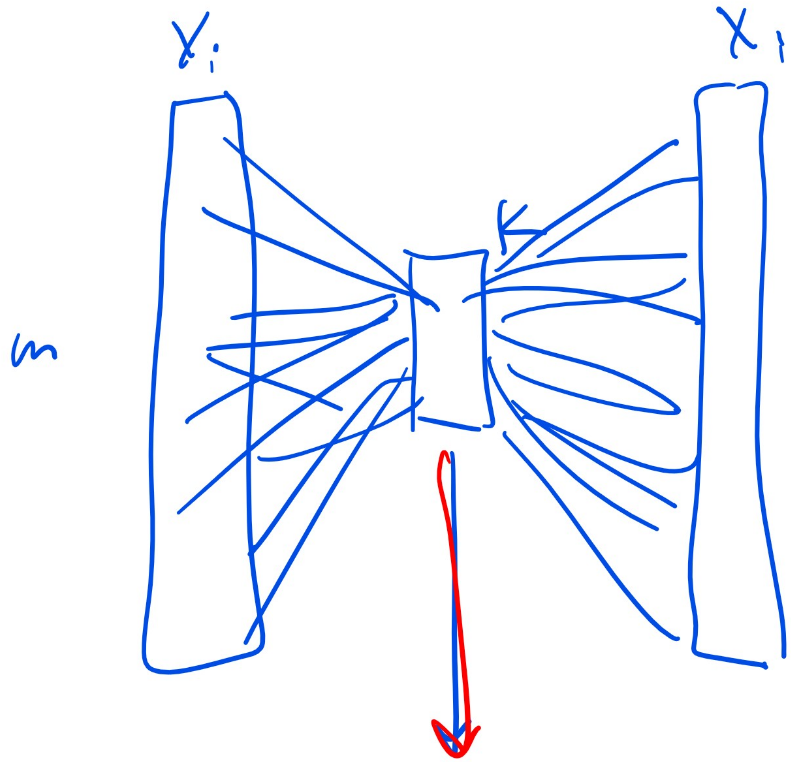
Autoencoder

Discussion

- A multi-layer neural network with the same input and output $y_i = x_i$ is called an autoencoder.
- The hidden layers have fewer units than the dimension of the input m.
- The hidden units form an encoding of the input with reduced dimensionality.

Autoencoder Diagram

Discussion



if $a_i = w^T x + b$ approximate PCA

NO $g(w^T x + b)$ non linear Logistic

non linear PCA

Eigenface

Discussion

- Eigenfaces are eigenvectors of face images (pixel intensities or HOG features).
- Every face can be written as a linear combination of eigenfaces. The coefficients determine specific faces.

#

$$x_i = \sum_{k=1}^{m} \left(u_k^T x_i\right) u_k \approx \sum_{k=1}^{K} \left(u_k^T x_i\right) u_k$$

new face

Unique identify a face

$(u_1^T x_i) u_1 + (u_2^T x_i) u_2 + \dots$

features

- Eigenfaces and SVM can be combined to detect or recognize faces.

T-Distributed Stochastic Neighbor Embedding

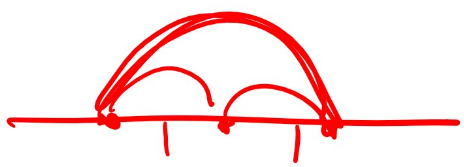
Discussion

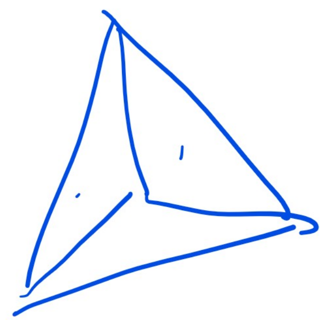
- t-distributed stochastic neighbor embedding is another non-linear dimensionality reduction method used mainly for visualization.
- Points in high dimensional spaces are embedded in 2 or 3-dimensional spaces to preserve the distance (neighbor) relationship between points.

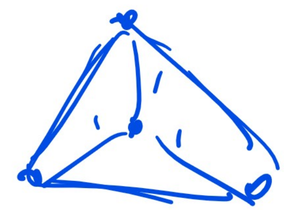
Embedding Diagram

Discussion

2D 1D 3D 2D







embed

⇒ 2D → 1D keep distance relation