CS540 Introduction to Artificial Intelligence
Lecture 12

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

June 26, 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

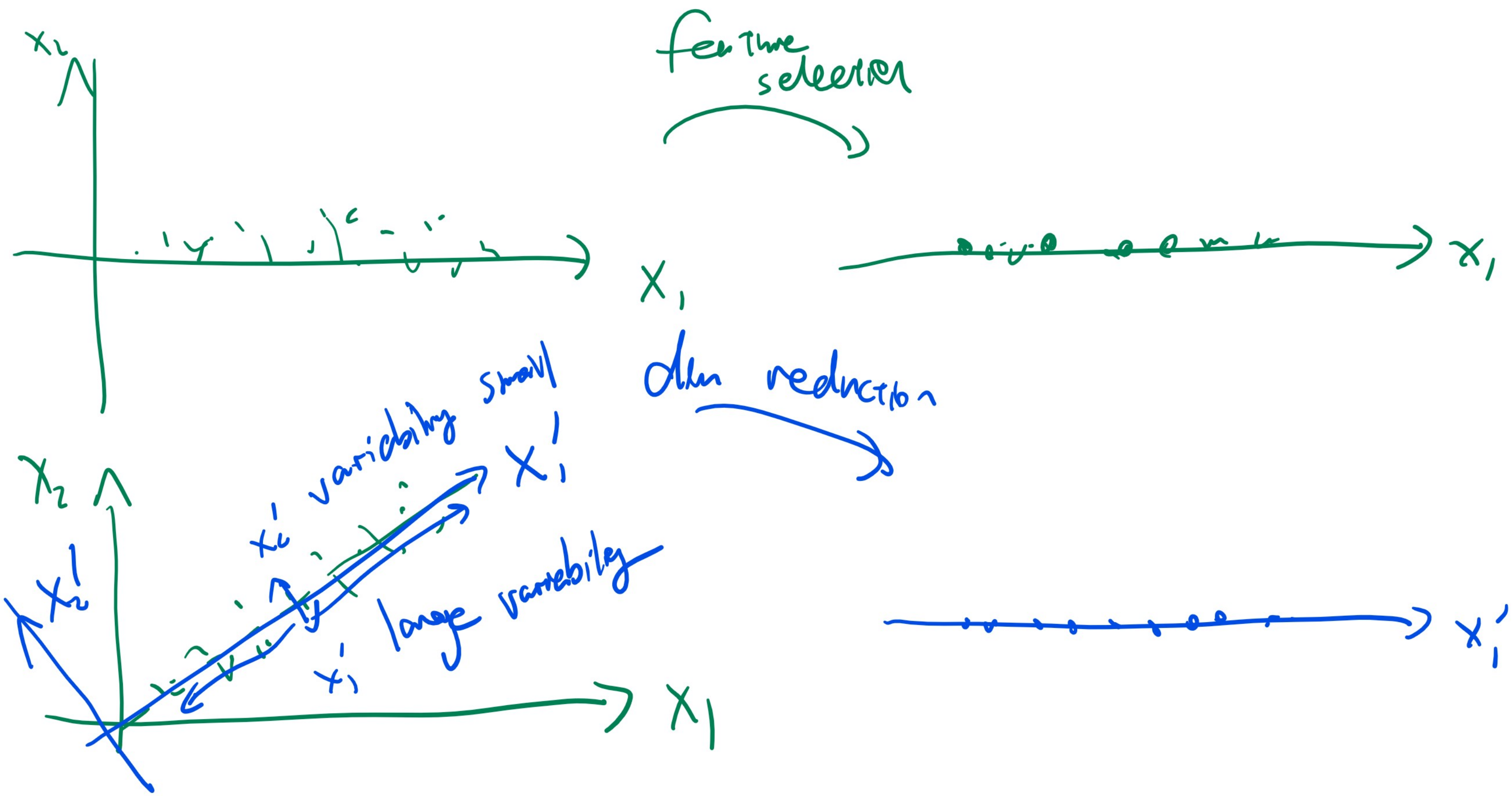
Motivation

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

1. Visualization.
2. 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

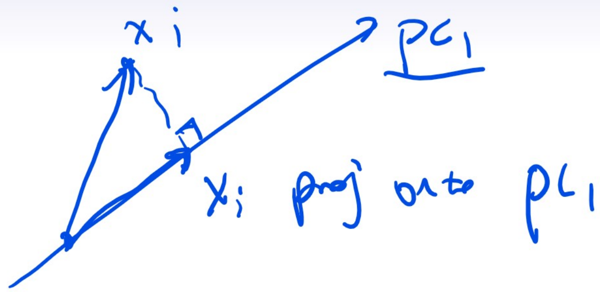
- In Euclidean space (L_2 norm), a unit vector u_k has length 1.

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

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

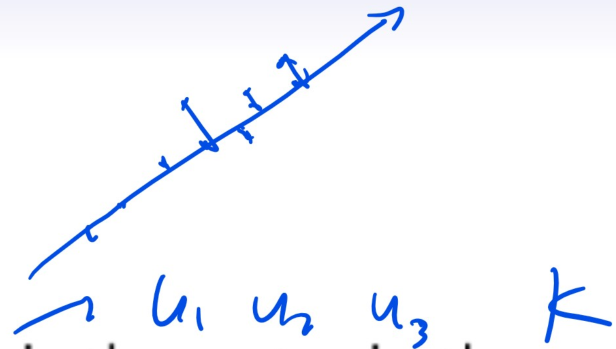
$$\begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix}\begin{pmatrix}0\\1\\0\\0\\0\end{pmatrix}\begin{pmatrix}0\\0\\1\\0\\0\end{pmatrix}\cdots$$

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



Projection

Definition

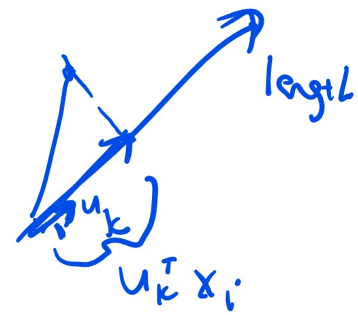


- The projection of x_i onto a unit vector u_k is the vector in the direction of u_k that is the closest to x_i.

u_k

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

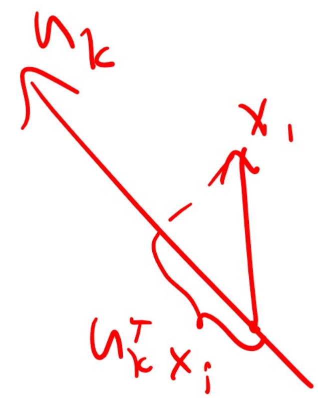
→ length of proj



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

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

$u_k^T u_k = 1$



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: $\begin{bmatrix}\sqrt{2}\\0\end{bmatrix}$

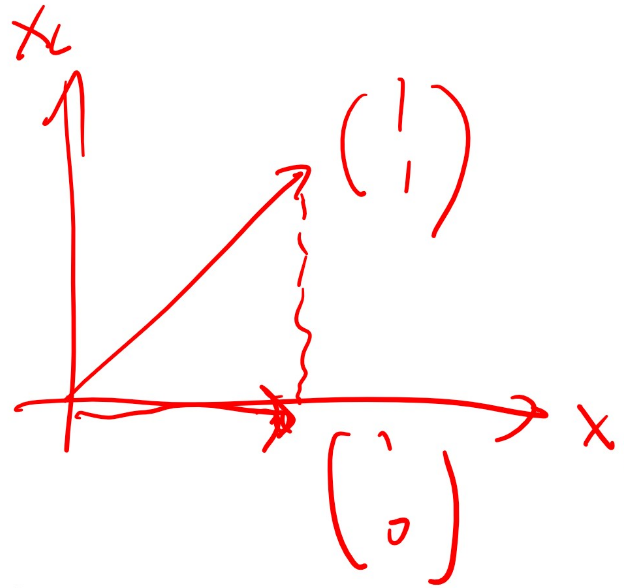
length

$(\underline{1}, \underline{1})\left(\frac{1}{2}\right) = 1$

direction

$\begin{pmatrix}1\\0\end{pmatrix} \leftarrow$ unit vec

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



$\text{Proj} = 1 \cdot \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}$

Project Example, Part II

Quiz (Graded)

ignore Q1

- What is the projection of $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ onto $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$?

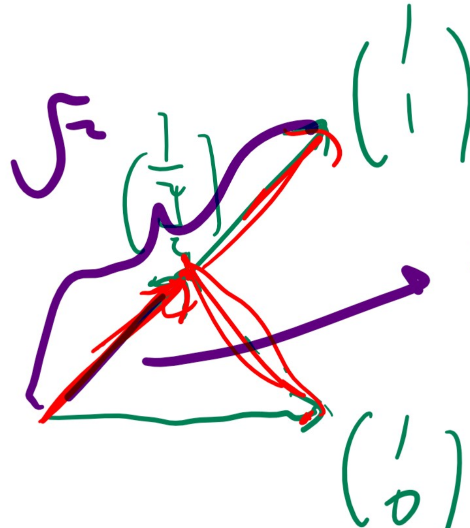
Q2

- 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: $\boxed{\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}}$

put on final

len $u^T x$

dir unit $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$



$\frac{\sqrt{2}}{2} = \frac{1}{\sqrt{2}}$

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

unit $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$

len dir $\sqrt{(1, 1)\begin{pmatrix} 1 \\ 1 \end{pmatrix}} = \sqrt{2}$

$\frac{u^T x}{u^T u} u = \frac{1}{2} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$

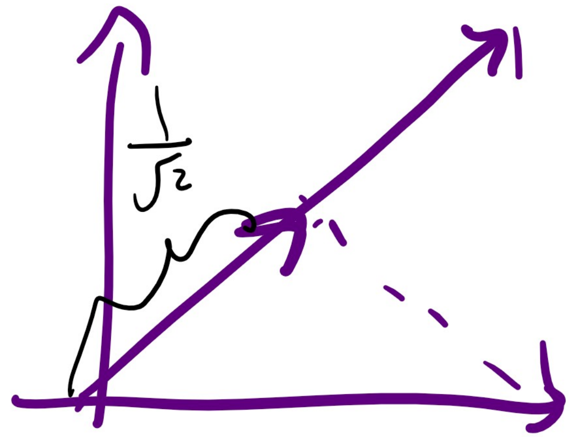
$\boxed{\hat{u} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)}$ unit direction

$\hat{u}^T x = \frac{1}{\sqrt{2}}$, $\left(\frac{1}{2}, \frac{1}{2} \right)$

Variance

Definition

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



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

m x 1 vector

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$\begin{pmatrix} x_{11} \\ x_{12} \\ x_{1n} \end{pmatrix} + \begin{pmatrix} x_{21} \\ x_{22} \\ x_{2n} \end{pmatrix} + \ldots$

m x m matrix

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

sample covariance matrix

$\boxed{\text{NOT } \cancel{(x_i - \hat{\mu})^T (x_i - \hat{\mu})}}$

Normalization

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} (y_1 \; y_2 \; y_n) = \begin{pmatrix} x_1 y_1, & x_1 y_2, & x_1 y_3 \\ x_2 y_1 & x_2 y_2 & \cdots \\ & \cdots & \end{pmatrix}$$

Definition

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

$$x_i = x_i - \hat{\mu}$$

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

why?

Covariance Matrix

Definition

- $\hat{\Sigma}$ is an $m \times m$ matrix and it is usually called the sample covariance matrix. The diagonal elements are variances in each dimension.

$$\hat{\sigma}_j^2 = \hat{\Sigma}_{jj} = \frac{1}{n-1} \sum_{i=1}^{n} x_{ij}^2$$

Projected Variance

Definition

in coord j

- Note that $x_{ij} = e_j^T x_i$, where e_j is the vector of 0 except it is 1 in coordinate j.

$(x_{i1} \ x_{i2} \ \dots \ x_{im}) \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{pmatrix} = x_{ij}$

jth

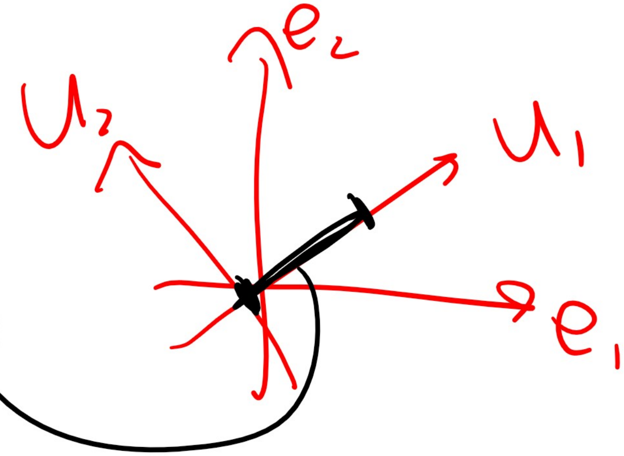
$$\hat{\sigma}_j^2 = e_j^T \hat{\Sigma} e_j = \frac{1}{n-1} e_j^T X^T X e_j$$

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

- The variance of the normalized x_i projected onto direction u_k has a similar expression.

$$u_k^T \hat{\Sigma} u_k = \frac{1}{n-1} u_k^T X^T X u_k$$

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



Maximum Variance Directions

Definition

PC are unit eigen vectors of covariance matrix

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

$u_k^T X u_k = \lambda \leftarrow \lambda$ is maximal variation

PC

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

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

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

variance along direction u_k

eigen value

eigen vector

m eigenvalues pick $\lambda_{max} \longrightarrow$ 1st PC

Eigenvalue

Definition

- The λ represents the projected variance.

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

- 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.

$$\boxed{\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_m} \geqslant 0$$

u_1 u_2 u_m

pc1 pc2

Eigenvalue Algorithm

Definition

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

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

not on final

- 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, Part I

Quiz (Participation)

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

columns are eigen vectors

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

λ_1 λ_2 λ_3

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

- 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: $\begin{bmatrix} -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

Spectral Decomposition Example, Part II

Quiz (Participation)

- Given the following spectral decomposition of $\left(\hat{\Sigma}\right)^{-1}$, what is the first principal component?

$$A^{-1} = \left(P D P^{-1}\right)^{-1} = P \underline{D^{-1}} P^{-1}$$

$$\left(\hat{\Sigma}\right)^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}^{-1}$$

$$\begin{pmatrix} 1 & & \\ & \frac{1}{2} & \\ & & \frac{1}{3} \end{pmatrix}$$

- 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: $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

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.

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

- The new feature space is K dimensional.

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

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

Q25

Reduced Space Example

Quiz (Graded)

$(u_1^T x, \; u_2^T x)$ ← new features.

- 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: $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} \end{bmatrix}$

Number of Dimensions

Discussion

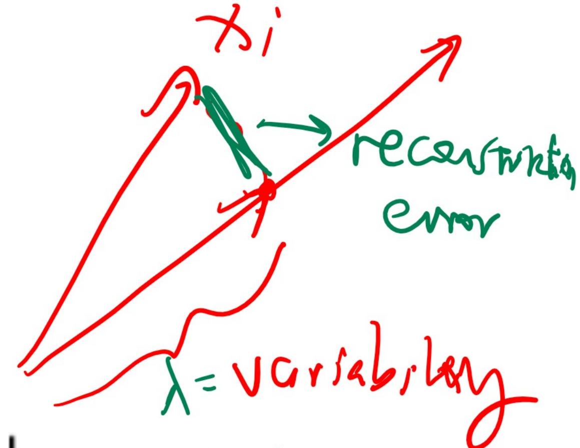
- There are a few ways to choose the number of principal components K.
- K can be selected given prior knowledge or requirement.
- K can be the number of non-zero eigenvalues.
- K can be the number of eigenvalues that are large (larger than some threshold).

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.

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

$$\underbrace{\sqrt{\|x\|^2 - \lambda^2}}_{\min}$$

max

Reconstruction Error Diagram

Discussion

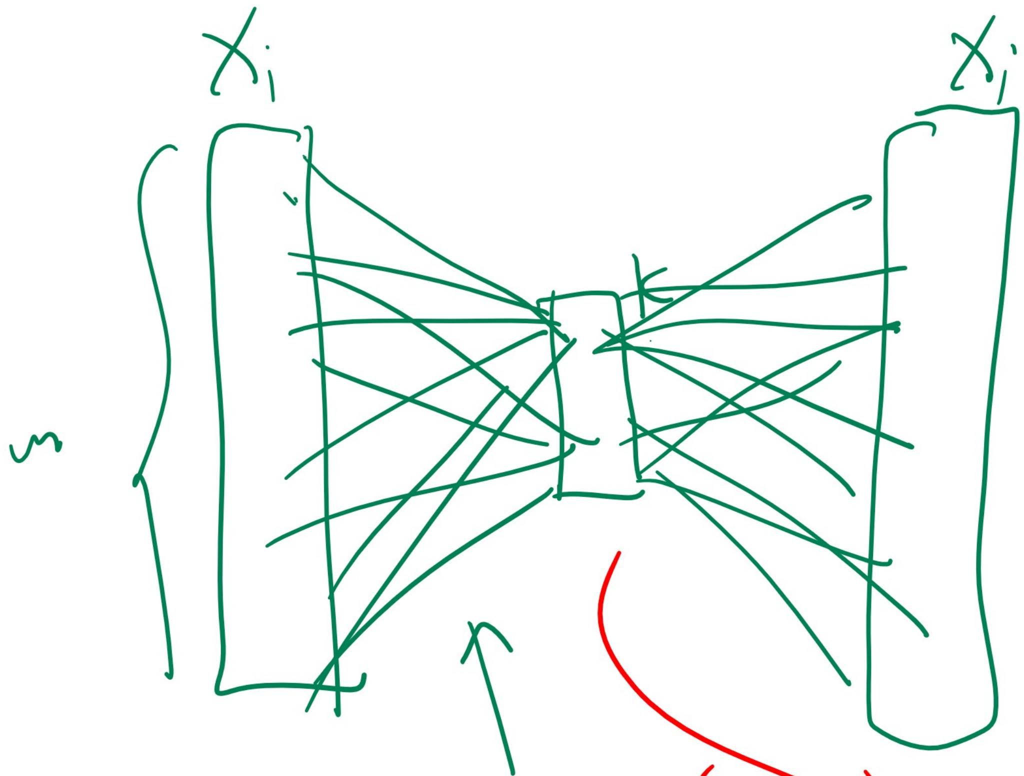
Autoencoder

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



$a_i = g(w^T X)$

nonlinear PCA

$a_i = w^T x_i + b$

$\cancel{\frac{\partial a_i}{\partial z_i}} \frac{\partial z_i}{\partial w_i}$

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.

new feature k,

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

eigenface k

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

m

Kernel PCA

Discussion

- A kernel can be applied before finding the principal components.

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

- The principal components can be found without explicitly computing $\phi(x_i)$, similar to the kernel trick for support vector machines.
- Kernel PCA is a non-linear dimensionality reduction method.

T-Distributed Stochastic Neighbor Embedding

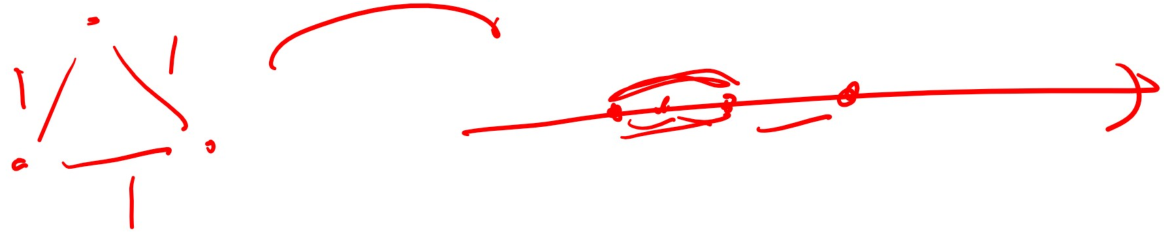
Discussion

- 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

Ranked



preserve distance relation