# UVA CS 4501 - 001 / 6501 – 007 Introduction to Machine Learning and Data Mining

## Lecture 23: Principal Component Analysis (PCA)

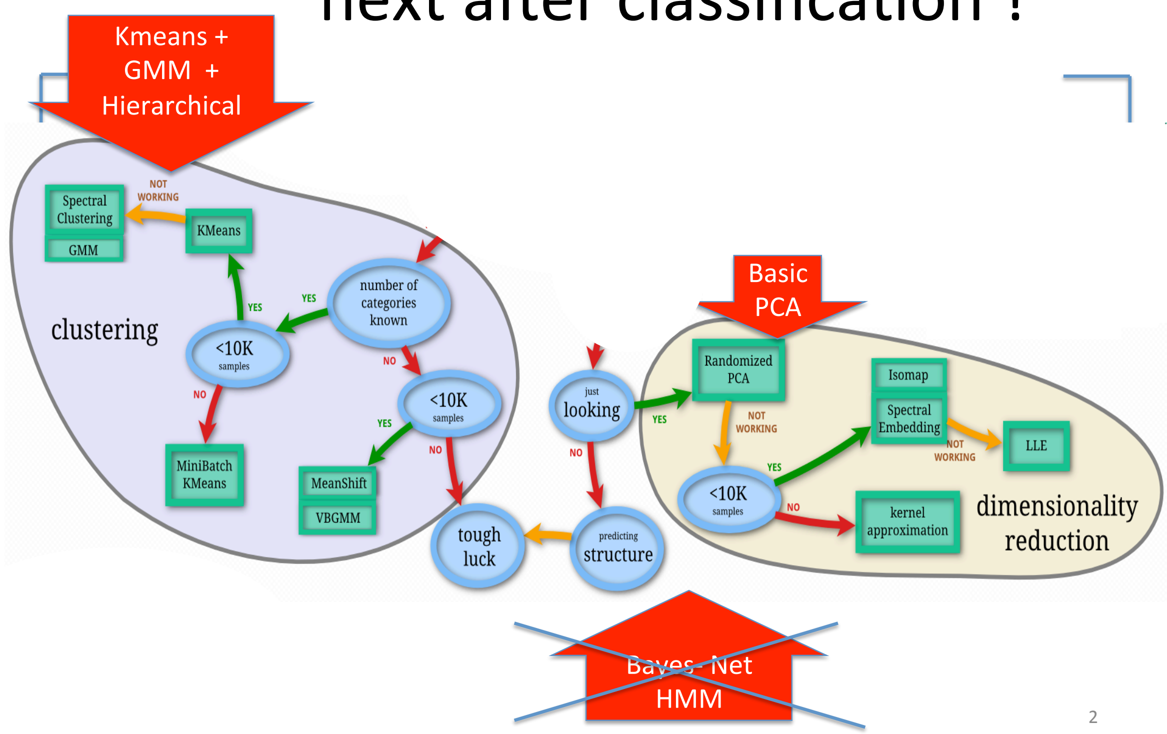
Yanjun Qi / Jane, , PhD

University of Virginia
Department of
Computer Science





## next after classification ?



# Where are we ? ➔ Five major sections of this course

- Regression (supervised)
- Classification (supervised)
  - Feature selection
- Unsupervised models
  - Dimension Reduction (PCA)
  - Clustering (K-means, GMM/EM, Hierarchical )
- Learning theory
- ~~Graphical models~~





# Today

- Dimensionality Reduction (unsupervised) with Principal Components Analysis (PCA)
  - Review of eigenvalue, eigenvector
  - How to project samples into a line capturing the variation of the whole dataset ➔ Eigenvector / Eigenvalue of covariance matrix
  - Another explanation of PCA
  - PCA for dimension reduction
  - Eigenface ➔ PCA for face recognition

## Review: Mean and Variance

- Variance:

$$Var(X) = E((X - \mu)^2)$$
$$Var(X) = E(X^2) - \mu^2$$

  – Discrete RVs:

$$V(\mathrm{X}) = \sum_{v_i} (v_i - \mu)^2 \, \mathrm{P}(\mathrm{X} = v_i)$$

  – Continuous RVs:

$$V(\mathrm{X}) = \int_{-\infty}^{+\infty} (x - \mu)^2 f(x) dx$$

- Covariance:

$$Cov(X,Y) = E((X - \mu_x)(Y - \mu_y)) = E(XY) - \mu_x \mu_y$$

## Review: Covariance matrix

$$C = \begin{pmatrix} v(x_1) \; c(x_1,x_2) \ldots\ldots\ldots c(x_1,x_p) \\ c(x_1,x_2) \; v(x_2) \ldots\ldots\ldots c(x_2,x_p) \\ \\ c(x_1,x_p) \; c(x_2,x_p) \ldots\ldots\ldots v(x_p) \end{pmatrix}$$

## Review: Eigenvector / Eigenvalue

- The eigenvalues $\lambda_i$ are found by solving the equation

$$det(C-\lambda I)=0$$

- Eigenvectors are columns of the matrix U such that

$$C= U\, D\, U^T$$

- Where D= $\begin{pmatrix} \lambda_1 0 \ldots\ldots 0 \\ 0 \quad \lambda_2 \ldots\ldots 0 \\ 0 \\ 0 \ldots\ldots\ldots \lambda_p \end{pmatrix}$

From Dr. S. Narasimhan

## Review: Eigenvalue, e.g.

- Let us take two variables with covariance $c>0$
- **C**= $\begin{pmatrix} 1 & c \\ c & 1 \end{pmatrix}$ **C**-$\lambda$**I**= $\begin{pmatrix} 1-\lambda & c \\ c & 1-\lambda \end{pmatrix}$

$$det(\mathbf{C}-\lambda \mathbf{I})=(1-\lambda)^2-c^2$$

- Solving this we find $\lambda_1 = 1+c$

$$\lambda_2 = 1-c < \lambda_1$$

From Dr. S. Narasimhan

## Review: Eigenvector, e.g.

- Any eigenvector u satisfies the condition

$$\mathbf{C}u=\lambda u$$

$$u = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \quad Cu = \begin{pmatrix} 1 & c \\ c & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} a_1 + ca_2 \\ ca_1 + a_2 \end{pmatrix} = \begin{pmatrix} \lambda a_1 \\ \lambda a_2 \end{pmatrix}$$

Solving we find $u_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$, $u_2 = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$

From Dr. S. Narasimhan



# Today

- Dimensionality Reduction (unsupervised) with Principal Components Analysis (PCA)
  - Review of eigenvalue, eigenvector
  - How to project samples into a line capturing the variation of the whole dataset ➔ Eigenvector / Eigenvalue of covariance matrix
  - Another explanation of PCA
  - PCA for dimension reduction
  - Eigenface ➔ PCA for face recognition

| | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| $s_1$ | | | |
| $s_2$ | | | |
| $s_3$ | | | |
| $s_4$ | | | |
| $s_5$ | | | |
| $s_6$ | | | |

# An unlabeled Dataset X

a data matrix of $n$ observations on $p$ variables $x_1, x_2, \ldots x_p$

- **Data**/*points/instances/examples/samples/records*: [ rows ]
- **Features**/*attributes/dimensions/independent variables/ covariates/predictors/regressors*: [ columns]



# The Goal

We wish to explain/summarize the underlying variance-covariance structure of a large set of variables through a few linear combinations of these variables.

PCA is introduced by Pearson (1901) and Hotelling (1933)

# Applications

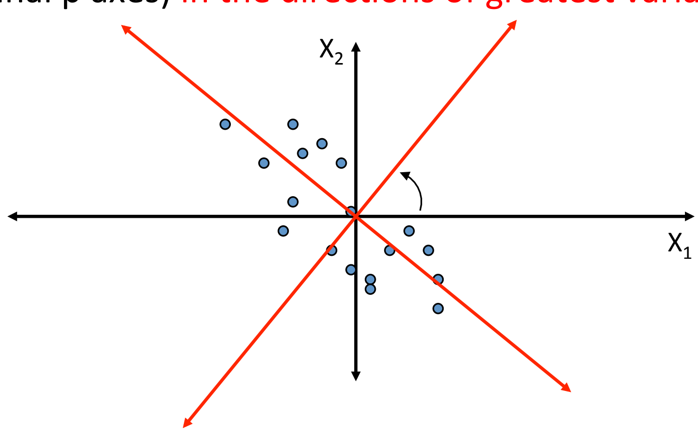
- Uses:
  - Data Visualization
  - Data Reduction
  - Data Classification
  - Trend Analysis
  - Factor Analysis
  - Noise Reduction
- Examples:
  - How many unique "sub-sets" are in the sample?
  - How are they similar / different?
  - What are the underlying factors that influence the samples?
  - How to best present what is "interesting"?
  - Which "sub-set" does this new sample rightfully belong?
  - .......

From Dr. S. Narasimhan

# Trick: Rotate Coordinate Axes

Suppose we have a population measured on **p** random variables $X_1,\ldots,X_p$.

Our goal is to develop a new set of p axes (linear combinations of the original p axes) in the directions of greatest variability:




This could be accomplished by rotating the axes (if data is centered).

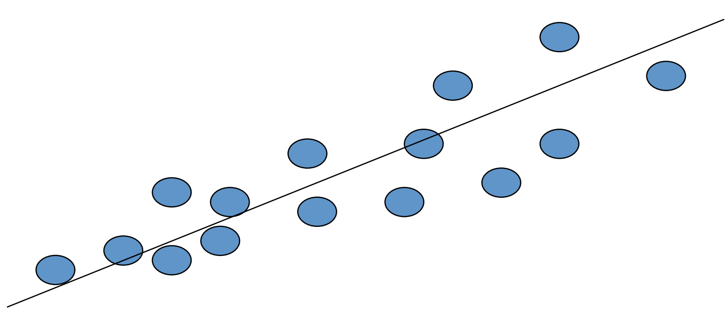
# Algebraic Interpretation

- Given n points in a p dimensional space,
- for large p, how does one project on to a lower-dimensional space while preserving broad trends in the data and allowing it to be visualized?

Data is centered: ➔ (we subtract the mean along each dimension, and center the original axis system at the centroid of all data points, for simplicity)

# Algebraic Interpretation – 1D

- Given n points in a p dimensional space, for large p, how does one project on to a 1 dimensional space?



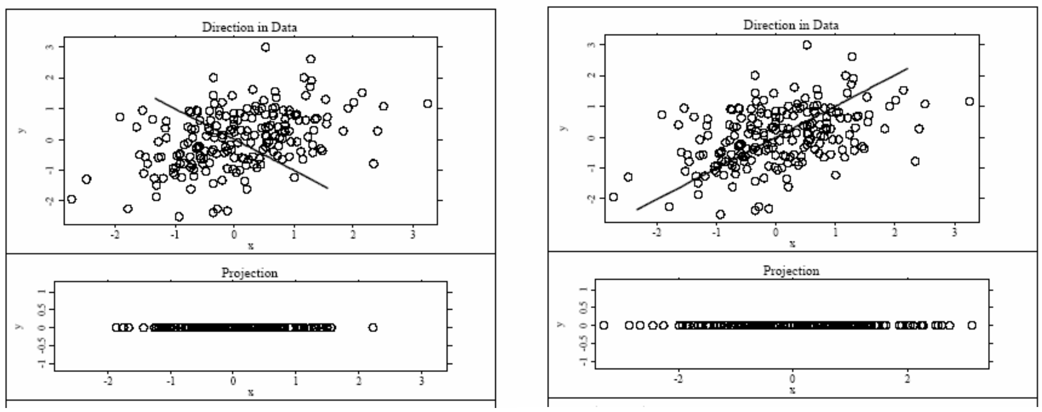
- Choose a line that fits the data so the points are spread out well along the line

From Dr. S. Narasimhan

# Let us see it on a figure

Good

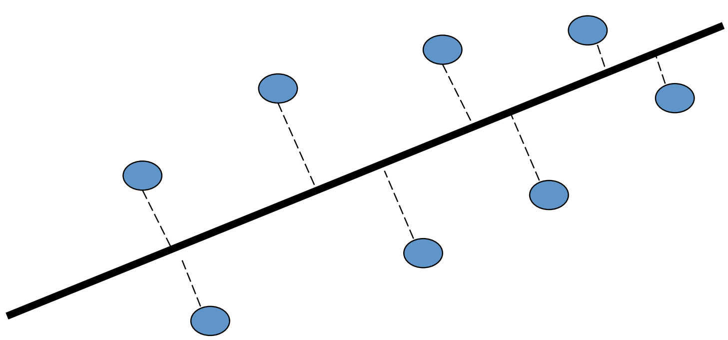
Better




(we subtract the mean along each dimension, and center the original axis system at the centroid of all data points, for simplicity)

# Algebraic Interpretation – 1D

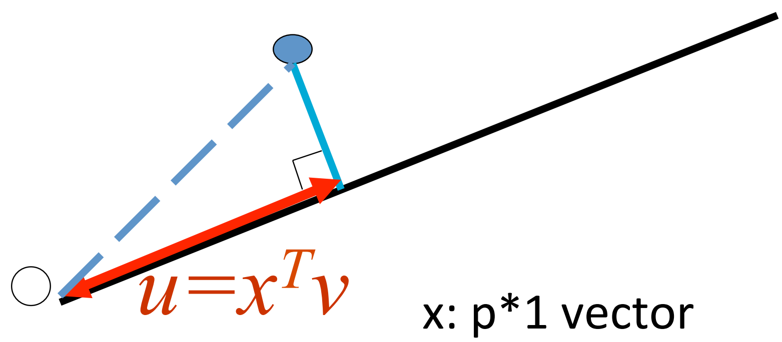
- Formally, minimize sum of squares of distances to the line.



From Dr. S. Narasimhan

# Algebraic Interpretation – 1D

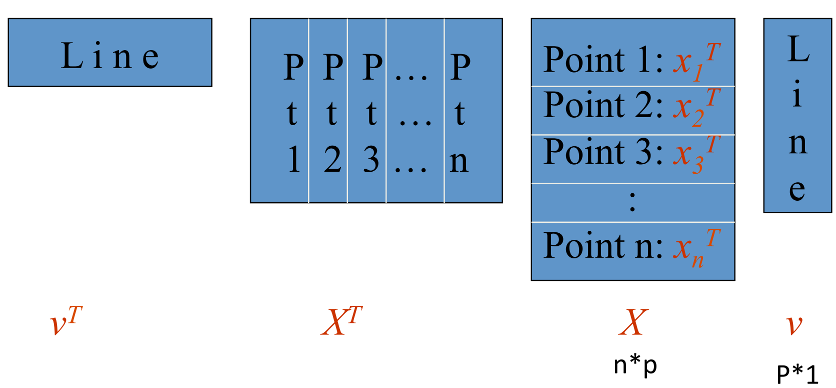
- Minimizing sum of squares of distances to the line is the same as maximizing the sum of squares of the projections on that line, thanks to Pythagoras.




x: p*1 vector
v: p*1 vector

# Algebraic Interpretation – 1D

- How is the sum of squares of projection lengths expressed in algebraic terms?




From Dr. S. Narasimhan

# Algebraic Interpretation – 1D

- How is the sum of squares of projection lengths expressed in algebraic terms?

$$\max(v^T X^T Xv), \text{ subject to } v^T v = 1$$

# Algebraic Interpretation – 1D

- Rewriting this: $\max(v^T X^T Xv), \text{ subject to } v^T v = 1$

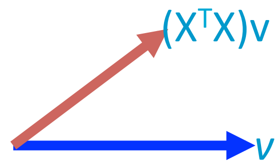
$$v^T X^T X v = \lambda = \lambda\, v^T v = v^T (\lambda v)$$

$$\Leftrightarrow \quad v^T (X^T X v - \lambda v) = 0$$

- Show that the maximum value of $v^T X^T X v$ is obtained for those u satisfying $X^T X v = \lambda v$
- So, find the largest $\lambda$ and associated u such that the matrix $X^T X$ when applied to u, yields a new vector which is in the same direction as u, only scaled by a factor $\lambda$.

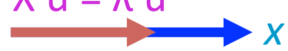
# Algebraic Interpretation – 1D

- $(X^TX)v$ points in some other direction in general




➔ u is an eigenvector and $\lambda$ is corresponding eigenvalue

$$X^TX\,u = \lambda\,u$$




# Algebraic Interpretation – beyond 1D

- How many eigenvectors are there?
- For Real Symmetric Matrices
  - except in degenerate cases when eigenvalues repeat, there are n eigenvectors

    *$u_1, \ldots, u_p$ are the eigenvectors*

    *$\lambda_1 \ldots \lambda_p$ are the eigenvalues, large to small, ordered by its value*

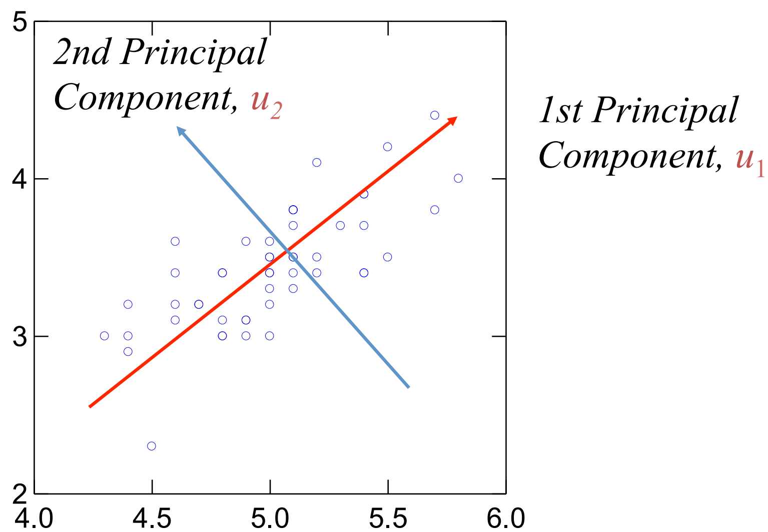
  - all eigenvectors are mutually orthogonal and therefore form a new basis space
    - Eigenvectors for distinct eigenvalues are mutually orthogonal
    - Eigenvectors corresponding to the same eigenvalue have the property that any linear combination is also an eigenvector with the same eigenvalue; one can then find as many orthogonal eigenvectors as the number of repeats of the eigenvalue.

From Dr. S. Narasimhan

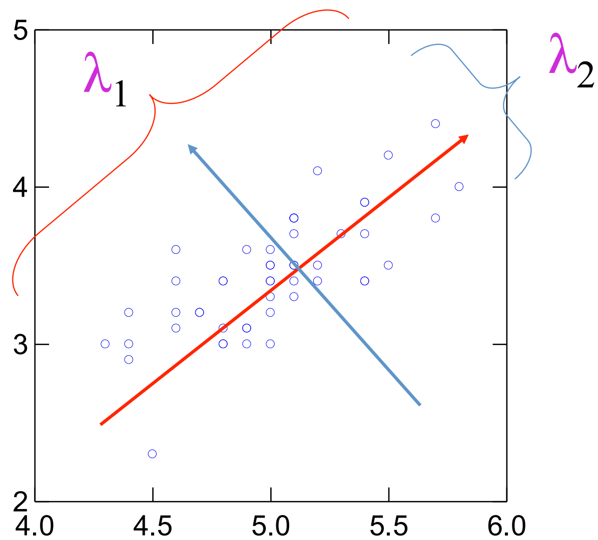
# Algebraic Interpretation – beyond 1D

- For matrices of the form (symmetric) $X^TX$
  - All eigenvalues are non-negative
  - See Handout-1 "linear algebra review" / Page 18,19,20
  - $\lambda_1 \ldots \lambda_p$ *are the eigenvalues, ordering from large to small,*
    - *i.e. Ordered by the PC's importance*

# PCA Eigenvectors ➔ Principal Components



# PCA Eigenvalues



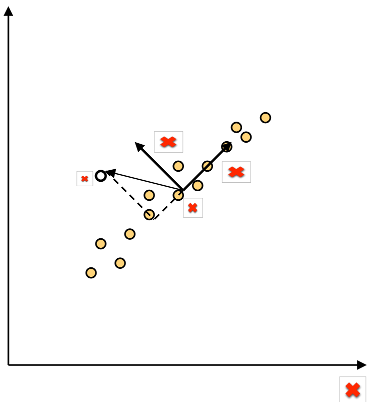



# Today

- Dimensionality Reduction (unsupervised) with Principal Components Analysis (PCA)
  - Review of eigenvalue, eigenvector
  - How to project samples into a line capturing the variation of the whole dataset → Eigenvector / Eigenvalue of covariance matrix
  - Another explanation of PCA
  - PCA for dimension reduction
  - Eigenface → PCA for face recognition

# PCA: *explanation II*

$\overline{x}$ is the mean of the orange points



convert **x** into $\mathbf{v_1}$, $\mathbf{v_2}$ coordinates

Consider the variation along direction **v** among all of the orange points:

$$var(\mathbf{v}) = \sum_{\text{orange point }\mathbf{x}} \|(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}} \cdot \mathbf{v}\|^2$$

$$V(X) = \sum_{v_i} (v_i - \mu)^2 P(X = v_i)$$

$$\begin{aligned}
var(\mathbf{v}) &= \sum_{\mathbf{x}} \|(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}} \cdot \mathbf{v}\| \\
&= \sum_{\mathbf{x}} \mathbf{v}^{\mathbf{T}}(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}}\mathbf{v} \\
&= \mathbf{v}^{\mathbf{T}}\left[\sum_{\mathbf{x}}(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}}\right]\mathbf{v} \\
&= \mathbf{v}^{\mathbf{T}}\mathbf{A}\mathbf{v} \quad \text{where } \mathbf{A} = \sum_{\mathbf{x}}(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}}
\end{aligned}$$

max( var(v) ), subject to $v^T v = 1$

From Dr. S. Narasimhan

# PCA: *explanation II*

From *k* original variables: $x_1,x_2,...,x_k$:

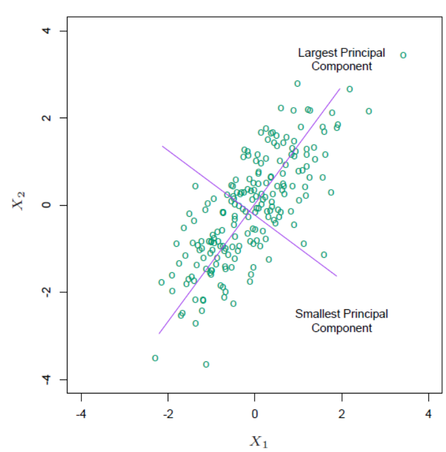
Produce *k* new variables: $u_1,u_2,...,u_k$:

$u_1 = a_{11}x_1 + a_{12}x_2 + ... + a_{1k}x_k$

$u_2 = a_{21}x_1 + a_{22}x_2 + ... + a_{2k}x_k$

...

$u_k = a_{k1}x_1 + a_{k2}x_2 + ... + a_{kk}x_k$



# PCA: *explanation II*

From $k$ original variables: $x_1,x_2,\ldots,x_k$:

Produce $k$ new variables: $u_1,u_2,\ldots,u_k$:

$$\left.\begin{aligned}
u_1 &= a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k \\
u_2 &= a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k \\
&\ldots \\
u_k &= a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kk}x_k
\end{aligned}\right\}$$

$u_k$'s are **Principal Components**

*such that:*

$u_k$'s are uncorrelated (orthogonal)
$u_1$ explains as much as possible of original variance in data set
$u_2$ explains as much as possible of remaining variance
etc.

From Dr. S. Narasimhan

# Interpretation of PCA

- The new variables (PCs) have a variance equal to their corresponding eigenvalue, since

  $$Var(u_i) = u_i^T X^T X u_i = u_i^T \lambda_i u_i = \lambda_i u_i^T u_i = \lambda_i$$

  for all $i=1\ldots p$

- Small $\lambda_i$ ⇔ small variance ⇔ data change little in the direction of component $u_i$

# Today

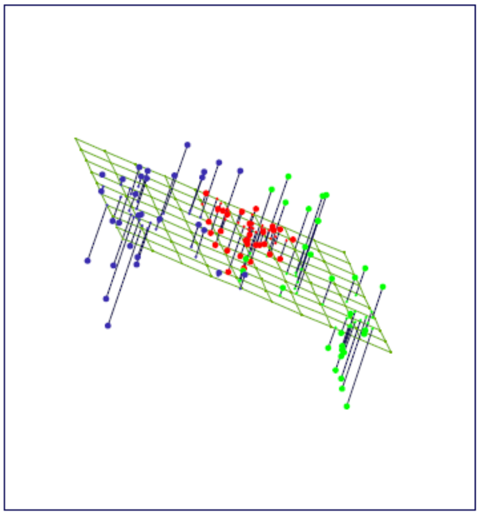
- Dimensionality Reduction (unsupervised) with Principal Components Analysis (PCA)
  - Review of eigenvalue, eigenvector
  - How to project samples into a line capturing the variation of the whole dataset → Eigenvector / Eigenvalue of covariance matrix
  - Another explanation of PCA
  - PCA for dimension reduction
  - Eigenface → PCA for face recognition

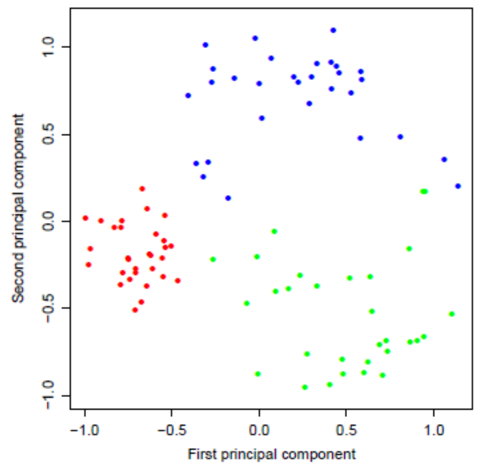


# PCA Summary until now

- Rotates multivariate dataset into a new configuration which is easier to interpret
- PCA is useful for finding new, more informative, uncorrelated features; it reduces dimensionality by rejecting low variance features

# PCA for dimension reduction
# e.g. p=3 (pick top two PCs)






corresponds to choosing a
"2D linear plane"



# PCs, Variance and Least-Squares

- The first PC retains the greatest amount of variation in the sample
- The $k^{th}$ PC retains the $k^{th}$ greatest fraction of the variation in the sample

- The least-squares view: PCs are a series of linear least squares fits to a sample set, each orthogonal to all previous ones

From Dr. S. Narasimhan

## Use PCA to reduce higher dimension

- Suppose each data point is p-dimensional

$$\begin{aligned} var(\mathbf{v}) &= \sum_{\mathbf{x}} \|(\mathbf{x}-\overline{\mathbf{x}})^{\mathrm{T}} \cdot \mathbf{v}\| \\ &= \mathbf{v}^{\mathrm{T}}\mathbf{A}\mathbf{v} \quad \text{where } \mathbf{A} = \sum_{\mathbf{x}} (\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathrm{T}} \end{aligned}$$

  - The eigenvectors of **A** define a new coordinate system
  - We can compress (i.e. perform projection ) the data points by only using the top few eigenvectors
    - corresponds to choosing a "linear subspace"
      - represent points on a line, plane, or "hyper-plane"
    - these eigenvectors are known as the ***principal components***

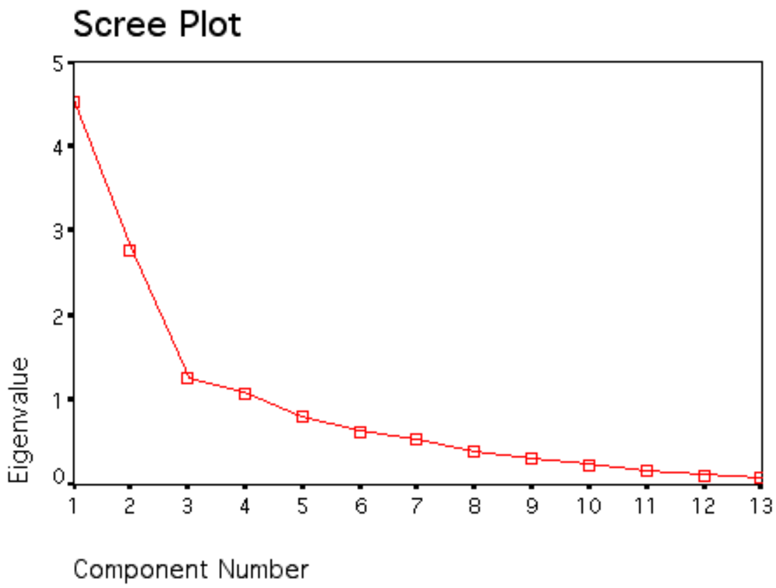
From Dr. S. Narasimhan

## How many components to keep?

- I. Variance: Enough PCs to have a cumulative variance explained by the PCs that is >50-70%
- II. Scree plot: represents the ability of PCs to explain the variation in data, e.g. keep PCs with eigenvalues >1

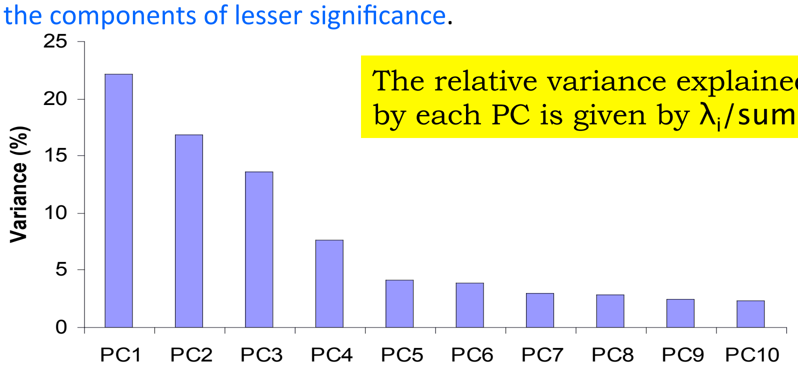
From Dr. S. Narasimhan

# Dimensionality Reduction
# e.g. check eigenvalue




# Dimensionality Reduction
# e.g. check percentage of kept variance

Can *ignore* the components of lesser significance.




The relative variance explained by each PC is given by $\lambda_i/\mathrm{sum}(\lambda_j)$

You do lose some information, but if the eigenvalues are small, you don't lose much

- p dimensions in original data
- Calculate p eigenvectors and eigenvalues
- choose only the first k eigenvectors, based on their eigenvalues
- final projected data set has only k dimensions

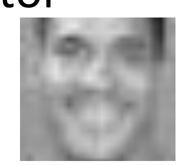
# Today

- Dimensionality Reduction (unsupervised) with Principal Components Analysis (PCA)
  - Review of eigenvalue, eigenvector
  - How to project samples into a line capturing the variation of the whole dataset ➔ Eigenvector / Eigenvalue of covariance matrix
  - Another explanation of PCA
  - PCA for dimension reduction
  - Eigenface ➔ PCA for face recognition



# Example 1: Application to image, e.g. task of face recognition

1. Treat pixels as a vector



$\rightarrow \mathbf{x}$

2. Recognize face by nearest neighbor

$\rightarrow \mathbf{y}_1 \ldots \mathbf{y}_n$

$$k = \underset{k}{\operatorname{argmin}} \left\| \mathbf{y}_k{}^T - \mathbf{x} \right\|$$

From Prof. Derek Hoiem

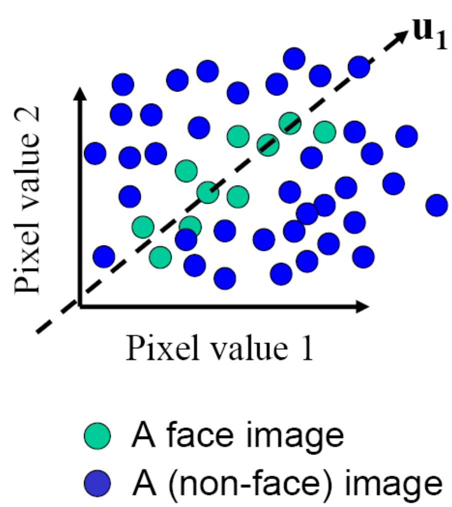
# Example 1: the space of all face images

- When viewed as vectors of pixel values, face images are extremely high-dimensional
  - 100x100 image = 10,000 dimensions
  - Slow and lots of storage
- But very few 10,000-dimensional vectors are valid face images
- We want to effectively model the subspace of face images

From Prof. Derek Hoiem

# Example 1:The space of all face images

- Eigenface idea: construct a low-dimensional linear subspace that best explains the variation in the set of face images




From Prof. Derek Hoiem

## Example 1: Application to Faces, e.g. Eigenfaces (PCA on face images)

1. Compute covariance matrix of face images

2. Compute the principal components ("eigenfaces")
   – K eigenvectors with largest eigenvalues

3. Represent all face images in the dataset as linear combinations of eigenfaces
   – Perform nearest neighbors on these coefficients

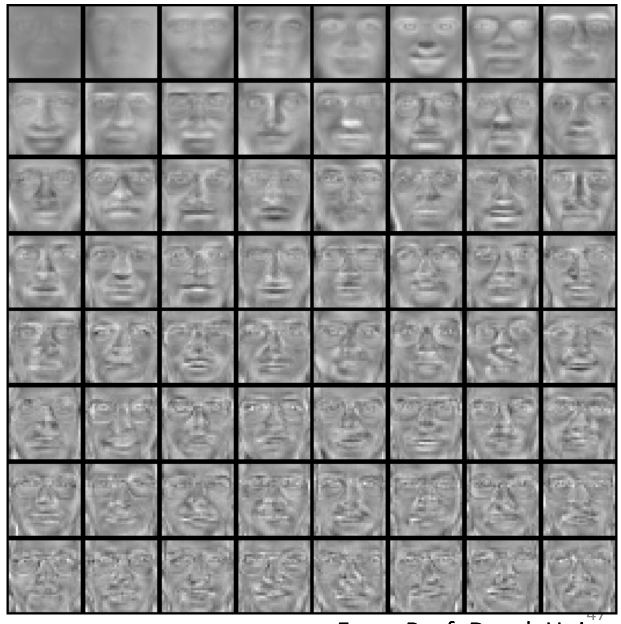
M. Turk and A. Pentland, Face Recognition using Eigenfaces, CVPR 1991

## Example 1: Application to Faces

Training images

# Example 1: Eigenfaces example

Top eigenvectors: $u_1, \ldots u_k$

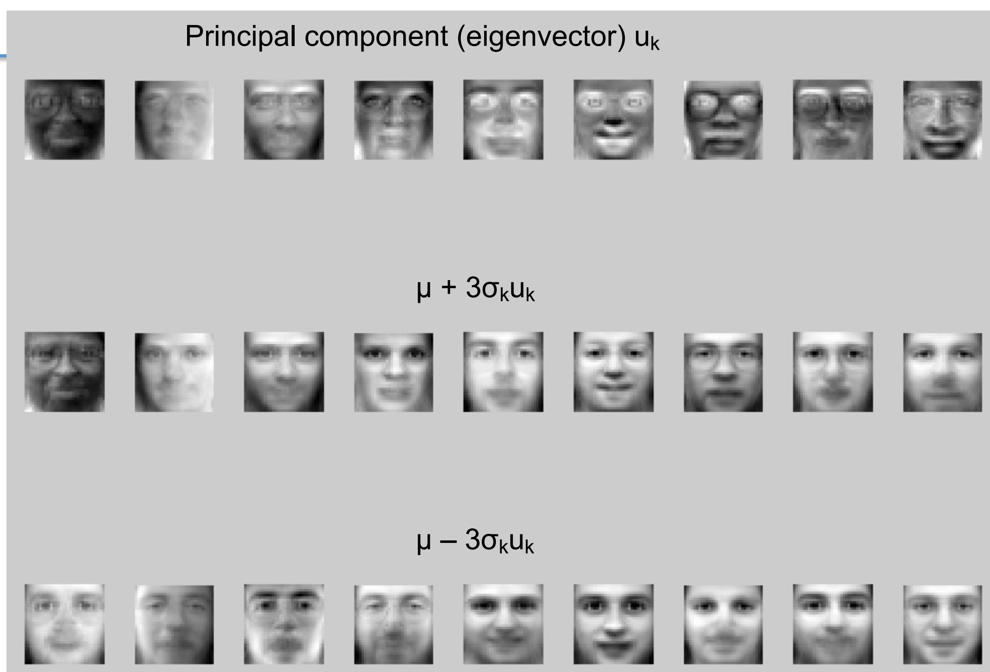


Mean: μ

$$\mu = \frac{1}{N}\sum_{k=1}^{N} x_k$$

From Prof. Derek Hoiem

# Example 1: Visualization of eigenfaces




From Prof. Derek Hoiem

# Example 1: Representation and reconstruction of original x

- Face **x** in "face space" coordinates:

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x}-\mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x}-\mu)]$$

$$= \quad w_1, \ldots, w_k$$

New representation

Remarkably few eigenvector terms are needed to give a fair likeness of most people's faces.

➔ subtract the mean along each dimension, in order to center the original axis system at the centroid of all data points

# Representation and reconstruction

- Face **x** in "face space" coordinates:

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x}-\mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x}-\mu)]$$

$$= \quad w_1, \ldots, w_k$$

New representation

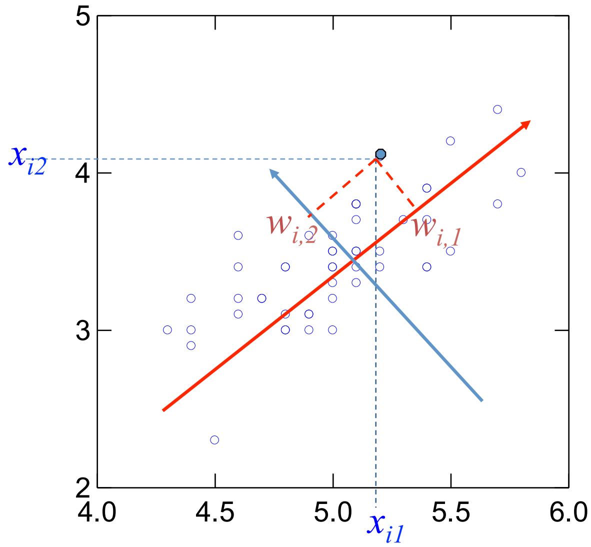
- Reconstruction:


$\hat{x} = \mu + w_1u_1+w_2u_2+w_3u_3+w_4u_4+ \ldots$

A human face may be considered to be a linear combination of these standardized eigen faces

From Prof. Derek Hoiem

## New representation in the lower-dim PC space




From Prof. Derek Hoiem

## Key Property of Eigenspace Representation

Given

- 2 images $\hat{x}_1, \hat{x}_2$ that are used to construct the Eigenspace
- $\hat{g}_1$ is the eigenspace projection of image $\hat{x}_1$
- $\hat{g}_2$ is the eigenspace projection of image $\hat{x}_2$

Then,

$$\boxed{\| \hat{g}_2 - \hat{g}_1 \| \approx \| \hat{x}_2 - \hat{x}_1 \|}$$

That is, distance in Eigenspace is approximately equal to the distance between two original images.

# Classify / Recognition with eigenfaces

***Step I: Process labeled training images***

- Find mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma = \text{sum\_i} (x_i - \mu)(x_i - \mu)^T}$
- Find k principal components (i.e. eigenvectors of $\boldsymbol{\Sigma}$) ➔ $\mathbf{u}_1, \ldots \mathbf{u}_k$
- Project each training image $\mathbf{x}_i$ onto subspace spanned by the **top** principal components: $(w_{i1},\ldots,w_{ik}) = (\mathbf{u}_1^T(\mathbf{x}_i - \boldsymbol{\mu}), \ldots, \mathbf{u}_k^T(\mathbf{x}_i - \boldsymbol{\mu}))$

M. Turk and A. Pentland, Face Recognition using Eigenfaces, CVPR 1991

# Classify / Recognition with eigenfaces

***Step 2: Nearest neighbor based face classification***

Given a novel image $\mathbf{x}$

- Project onto k PC's subspace: $(w_1,\ldots,w_k) = (\mathbf{u}_1^T(\mathbf{x} - \boldsymbol{\mu}), \ldots, \mathbf{u}_k^T(\mathbf{x} - \boldsymbol{\mu}))$
- Optional: check reconstruction error $\mathbf{x} - \hat{\mathbf{x}}$ to determine whether the image is really a face
- Classify as closest training face(s) in the lower k-dimensional subspace

M. Turk and A. Pentland, Face Recognition using Eigenfaces, CVPR 1991

## Is this a face or not?

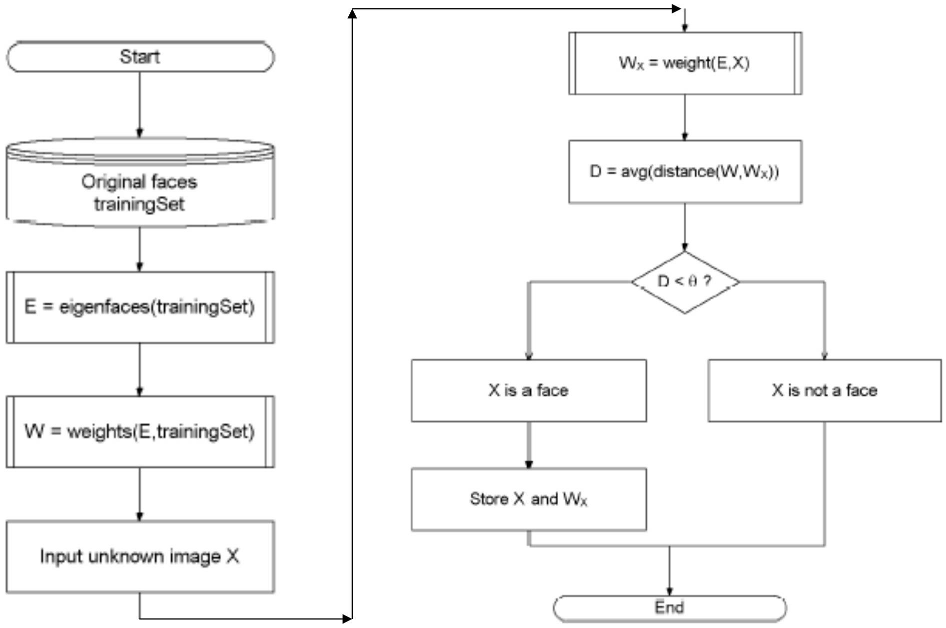



Figure 1: High-level functioning principle of the eigenface-based facial recognition algorithm

## Example 2: e.g. Handwritten Digits

- 16 x 16 gray scale
- Total 658 such 3's
- 130 is shown
- Image $x_i$ : $R^{256}$
- Compute principal components

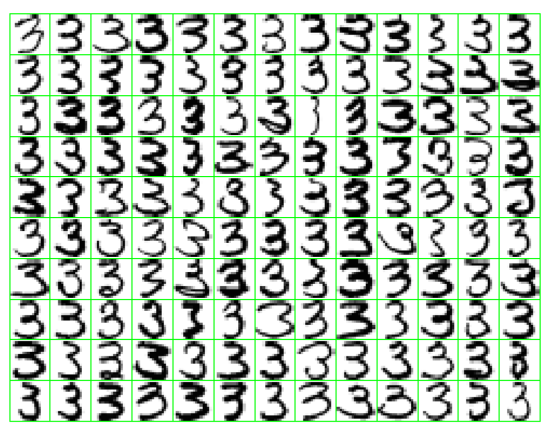
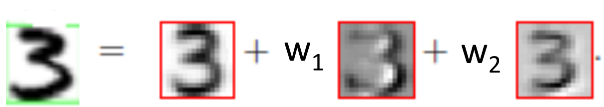


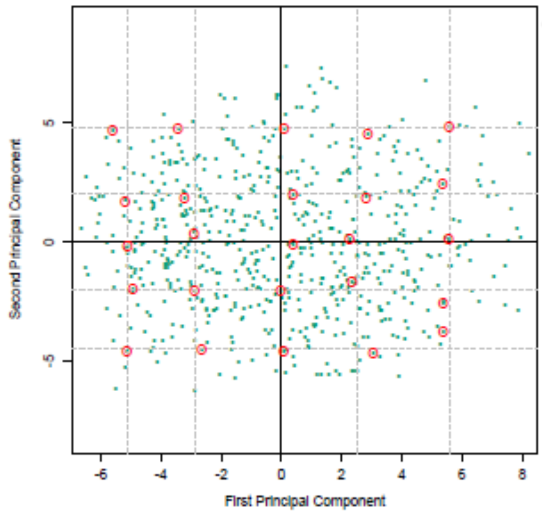
FIGURE 14.22. *A sample of 130 handwritten 3's shows a variety of writing styles.*




e.g. From ESL book

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x}-\mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x}-\mu)]$$

$w_1, \ldots, w_k$



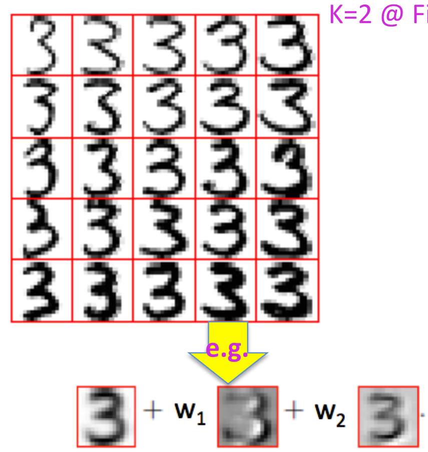
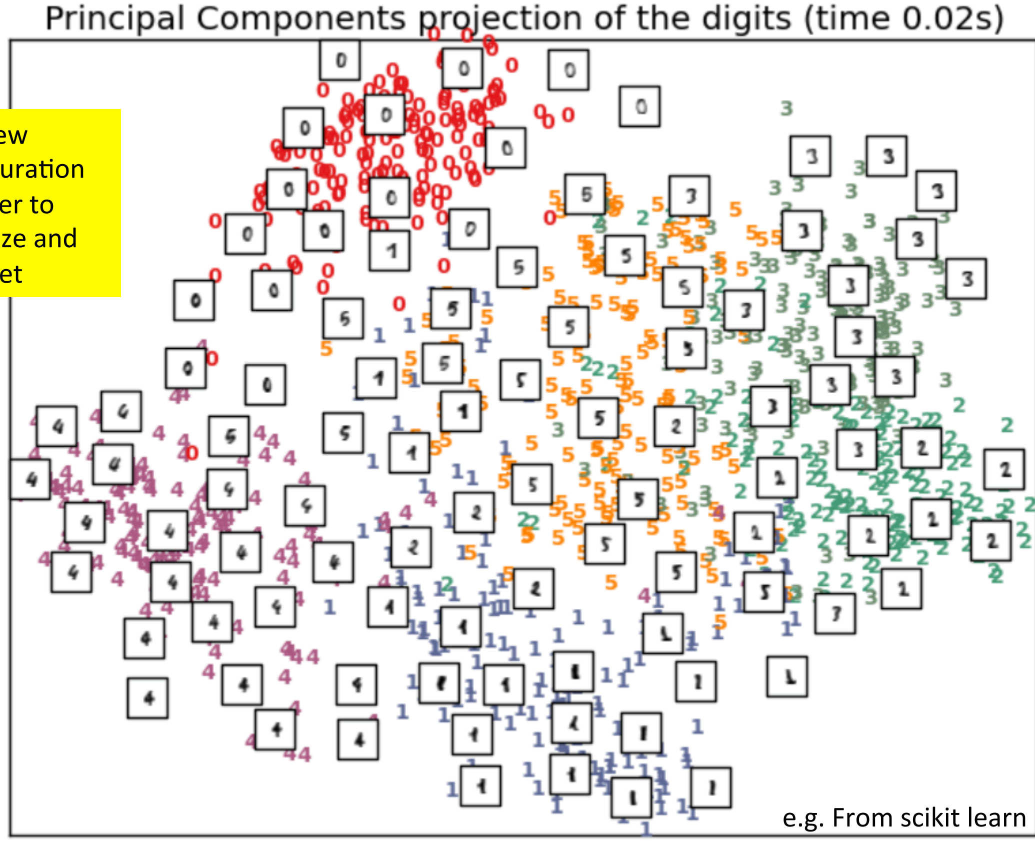




**FIGURE 14.23.** *(Left panel:) the first two principal components of the handwritten threes. The circled points are the closest projected images to the vertices of a grid, defined by the marginal quantiles of the principal components. (Right panel:) The images corresponding to the circled points. These show the nature of the first two principal components.*

e.g. From ESL book



Principal Components projection of the digits (time 0.02s)



The new configuration is easier to visualize and intepret

e.g. From scikit learn

# PCA summary

- General dimensionality reduction technique
- Preserves most of variance with a much more compact representation
  - Lower storage requirements (eigenvectors + a few numbers per face/sample)
  - Faster matching (since matching within lower-dim)

# PCA & Gaussian Distributions.

- PCA is similar to learning a Gaussian distribution for the data.
- $\mu$ is the mean of the distribution.
- Then the estimate of the covariance.
- Dimension reduction occurs by ignoring the directions in which the covariance is small.

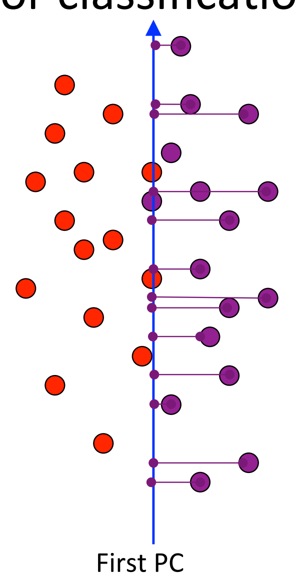
# (1) Limitations of PCA

- PCA is not effective for some datasets.
- For example, if the data is a set of strings
- (1,0,0,0,…), (0,1,0,0…),…,(0,0,0,…,1) then the eigenvalues do not fall off as PCA requires.

# (2) PCA Limitations

- The direction of maximum variance is not always good for classification

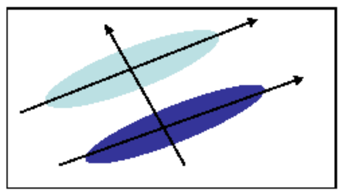
not ideal for discrimination




From Prof. Derek Hoiem

# PCA and Discrimination

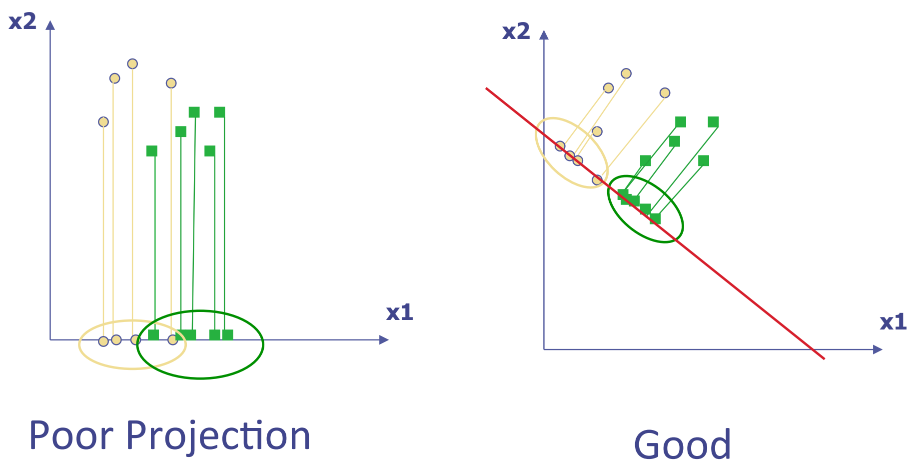
- PCA may not find the best directions for discriminating between two classes.
- Example: suppose the two classes have 2D Gaussian densities as ellipsoids.
- 1st eigenvector is best for representing the probabilities / overall data trend
- 2nd eigenvector is best for discrimination.



From Prof. Derek Hoiem

# PCA Limitations: Illustration of good projection for classification purpose

- Using two classes as example:




From Prof. Derek Hoiem

# References


- Hastie, Trevor, et al. *The elements of statistical learning*. Vol. 2. No. 1. New York: Springer, 2009.
- Dr. S. Narasimhan's PCA lectures
- Prof. Derek Hoiem's eigenface lecture






# Extra: A 2D Numerical Example

From Dr. S. Narasimhan

# PCA Example –STEP 1

- Subtract the mean from each of the data dimensions.

- Subtracting the mean makes variance and covariance calculation easier by simplifying their equations. The variance and co-variance values are not affected by the mean value.

From Dr. S. Narasimhan

# PCA Example –STEP 1

DATA:

| x1 | x2 |
|---|---|
| 2.5 | 2.4 |
| 0.5 | 0.7 |
| 2.2 | 2.9 |
| 1.9 | 2.2 |
| 3.1 | 3.0 |
| 2.3 | 2.7 |
| 2 | 1.6 |
| 1 | 1.1 |
| 1.5 | 1.6 |
| 1.1 | 0.9 |

ZERO MEAN DATA:

| x1 | x2 |
|---|---|
| .69 | .49 |
| -1.31 | -1.21 |
| .39 | .99 |
| .09 | .29 |
| 1.29 | 1.09 |
| .49 | .79 |
| .19 | -.31 |
| -.81 | -.81 |
| -.31 | -.31 |
| -.71 | -1.01 |

From Dr. S. Narasimhan

## PCA Example –STEP 2

- Calculate the covariance matrix

$$\text{cov} = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

- since the non-diagonal elements in this covariance matrix are positive, we should expect that the x1 and x2 variable increase together.

From Dr. S. Narasimhan

## PCA Example –STEP 3

- Calculate the eigenvectors and eigenvalues of the covariance matrix

$$\text{eigenvalues} = \begin{pmatrix} 1.28402771 \\ .0490833989 \end{pmatrix}$$

$$\text{eigenvectors} = \begin{pmatrix} -.677873399 & -.735178656 \\ -.735178656 & .677873399 \end{pmatrix}$$

From Dr. S. Narasimhan

# PCA Example –STEP 3

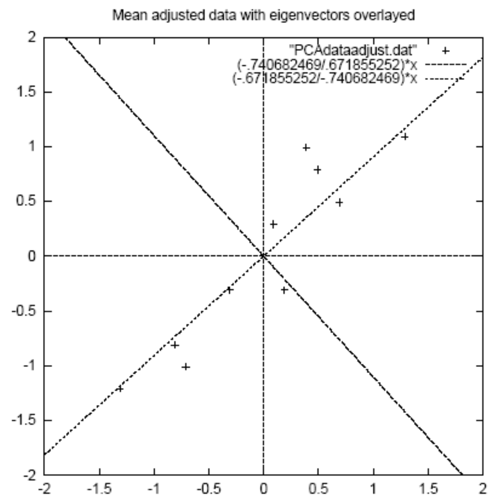



Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

•eigenvectors are plotted as diagonal dotted lines on the plot.
•Note they are perpendicular to each other.
•Note one of the eigenvectors goes through the middle of the points, like drawing a line of best fit.
•The second eigenvector gives us the other, less important, pattern in the data, that all the points follow the main line, but are off to the side of the main line by some amount.

From Dr. S. Narasimhan

# PCA Example –STEP 4

- Reduce dimensionality and form *feature vector*
  the eigenvector with the *highest* eigenvalue is the *principle component* of the data set.

  In our example, the eigenvector with the larges eigenvalue was the one that pointed down the middle of the data.

  Once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives you the components in order of significance.

From Dr. S. Narasimhan

# PCA Example –STEP 4

- Feature Vector

FeatureVector = ($eig_1$ $eig_2$ $eig_3$ … $eig_n$)

We can either form a feature vector with both of the eigenvectors:

$$\begin{pmatrix} -.677873399 & -.735178656 \\ -.735178656 & .677873399 \end{pmatrix}$$

or, we can choose to leave out the smaller, less significant component and only have a single column:

$$\begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

Now, if you like, you can decide to *ignore* the components of lesser significance.

You do lose some information, but if the eigenvalues are small, you don't lose much

# PCA Example –STEP 5

- Deriving the new data

**FinalData = RowFeatureVector x RowZeroMeanData**

RowFeatureVector is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top

RowZeroMeanData is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

From Dr. S. Narasimhan

# PCA Example –STEP 5

FinalData transpose: dimensions along columns

| w1 | w2 |
|---|---|
| -.827970186 | -.175115307 |
| 1.77758033 | .142857227 |
| -.992197494 | .384374989 |
| -.274210416 | .130417207 |
| -1.67580142 | -.209498461 |
| -.912949103 | .175282444 |
| .0991094375 | -.349824698 |
| 1.14457216 | .0464172582 |
| .438046137 | .0177646297 |
| 1.22382056 | -.162675287 |

From Dr. S. Narasimhan

# PCA Example –STEP 5

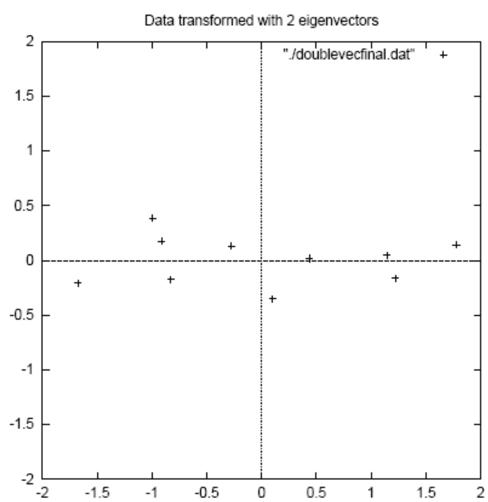



Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

From Dr. S. Narasimhan

# Reconstruction of original Data

- If we reduced the dimensionality, obviously, when reconstructing the data we would lose those dimensions we chose to discard.

- In our example let us assume that we considered only the w1 dimension…

From Dr. S. Narasimhan

# Reconstruction of original Data

| w1 |
|---|
| -.827970186 |
| 1.77758033 |
| -.992197494 |
| -.274210416 |
| -1.67580142 |
| -.912949103 |
| .0991094375 |
| 1.14457216 |
| .438046137 |
| 1.22382056 |

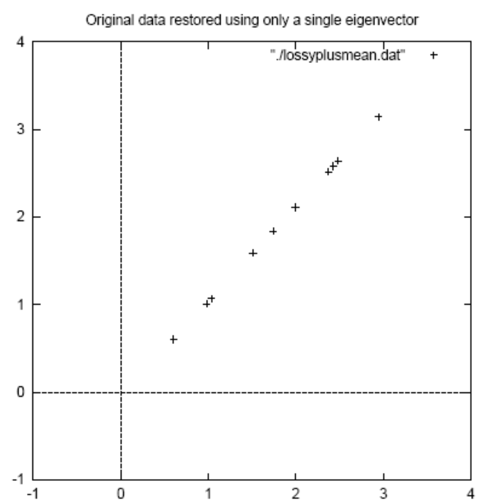



Figure 3.5: The reconstruction from the data that was derived using only a single eigenvector

From Dr. S. Narasimhan