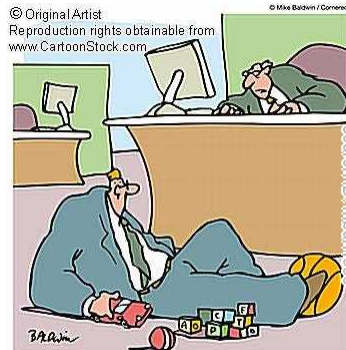


CSEP 573

Machine Learning II: Unsupervised Learning



"Take a course or something. Learning at your own speed isn't working out."

© CSE AI Faculty

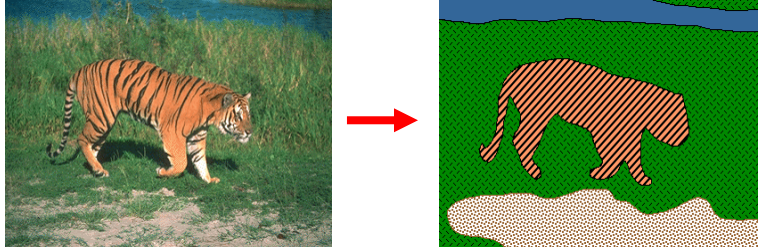
Our agenda today

Unsupervised Learning and Applications

- Clustering
 - Application: Image Segmentation
- Density Estimation and EM algorithm
- Dimensionality Reduction
 - Principal Component Analysis (PCA)
 - Applications: Image Compression, Face Recognition

Guest Lecture by Rawichote Chalodhorn:
Applications of Learning in Robotics

Motivation: Image Segmentation in Computer Vision



Goal: Partition an image into its constituent "objects"

Some slides adapted from Steve Seitz, Linda Shapiro 3

Idea: Image histograms

How many "orange" pixels are in this image?

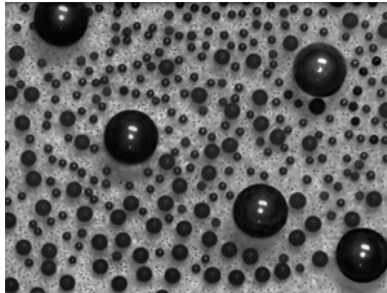
- Look at the *histogram*
- A histogram counts the number of occurrences of each color
 - Given an image $F[x, y] \rightarrow RGB$
 - The histogram is $H_F[c] = |\{(x, y) \mid F[x, y] = c\}|$
i.e., for each color value c on the x-axis, plot # of pixels with that color on y-axis



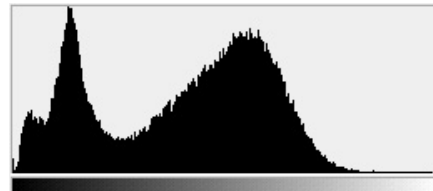
4

Example Histogram of a Grayscale Image

Image



Histogram



Intensity bins

How Many Modes Are There?

- Easy to see, hard to compute

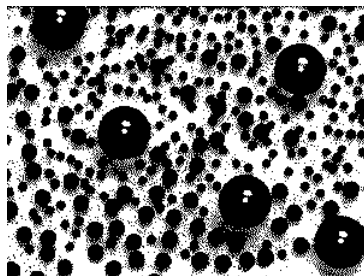
5

Histogram-based segmentation

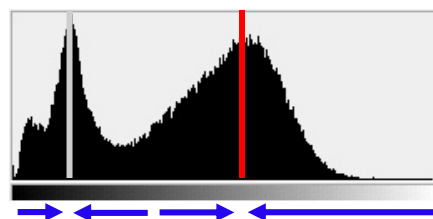
Idea: Break the image into K regions (segments) by

- reducing the number of colors to K
- assigning each pixel to the closest color

Here's what our image looks like if we use two colors (intensities)



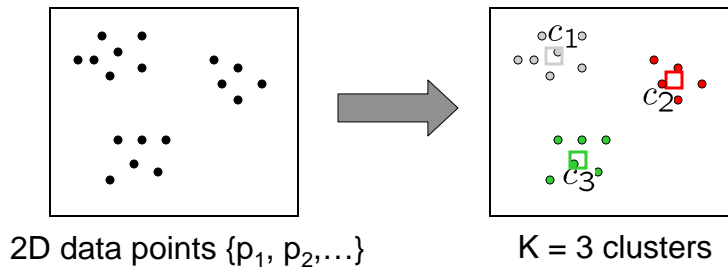
$K = 2$



6

Clustering

- Idea in previous slide can be formalized as *clustering*
- **Problem:** Given unlabeled data points $\{p_1, p_2, \dots, p_N\}$, assign each data point p_j to one of K clusters
 - points within a cluster are "similar" (according to some metric)
- Example of *unsupervised learning* (no label given)



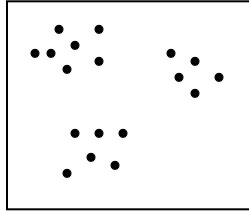
7

Why Clustering?

- **Lots of Applications!**
 - **Biology:** Discovering gene clusters with similar expression patterns, grouping homologous DNA sequences, etc.
 - **Marketing:** Grouping customers with similar traits for segmenting the market, product positioning etc.
 - **Vision:** Image segmentation, feature learning for recognition,...
 - **Search result grouping** (e.g, clusty.com)
 - **Social network analysis** (discovering user communities with similar interests)
 - **Crime analysis** (identification of "hot spots")
 - **Many more!**

8

Clustering: The Problem



Given: Unlabeled data

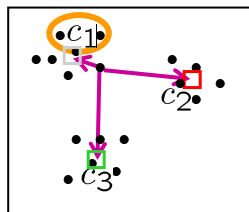
Goal: Assign each point to the cluster it is most similar to

Suppose we are given the number of clusters K ($= 3$ here)

How do we assign each point to a cluster?

9

Suppose you are given the cluster centers c_i



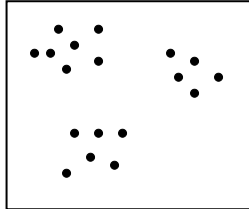
Q: how do you assign points to a cluster?

A: for each point p ,

- Compute distance to cluster centers
- Choose the closest c_i

10

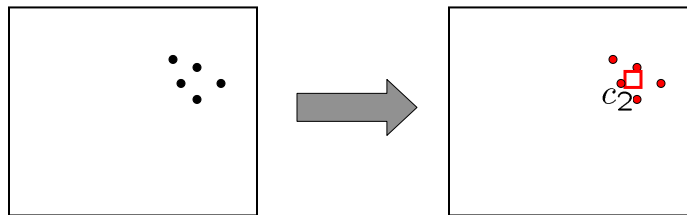
Suppose you are given the cluster centers c_i



But wait...you are not given the cluster centers!
How do you find them?

11

Finding the cluster center



Given a cluster of points, we can easily compute its center
(How?)

12

A chicken-or-egg problem?

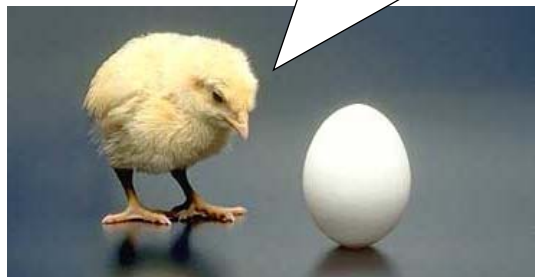
Given cluster centers, we can assign points
To find centers, we need points assigned to a cluster



13

A way out of the impasse

Why not alternate?
(between finding centers and
assigning points)

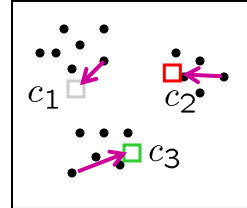


14

Alternate between 2 steps

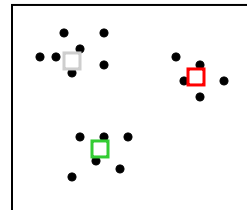
I. Given current estimate of cluster centers c_i :

Assign each point p to closest c_i



II. Given current assignment of points to clusters:

Choose c_i to be the mean of all the points in the cluster



15

K-means clustering

Algorithm

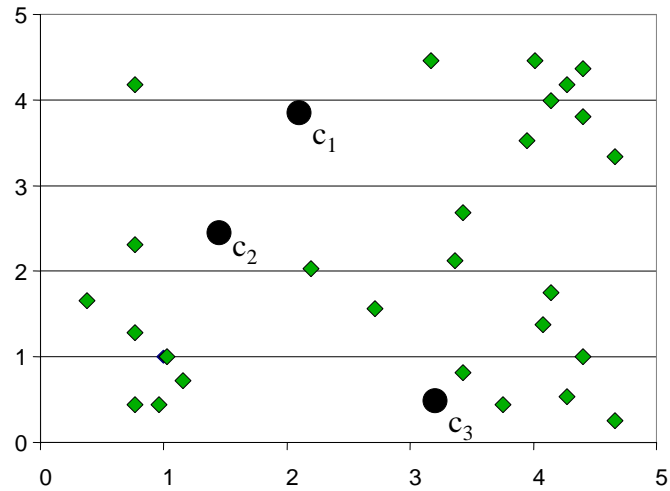
1. Randomly initialize the cluster centers, c_1, \dots, c_k
2. Determine cluster membership
 - For each point p , find the *closest* c_i
 - Put p into cluster i
3. Re-estimate cluster centers
 - Set c_i to be the mean of points in cluster i
4. If c_i have changed, go to 2 else done.

Java demo: http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

16

K-means clustering example

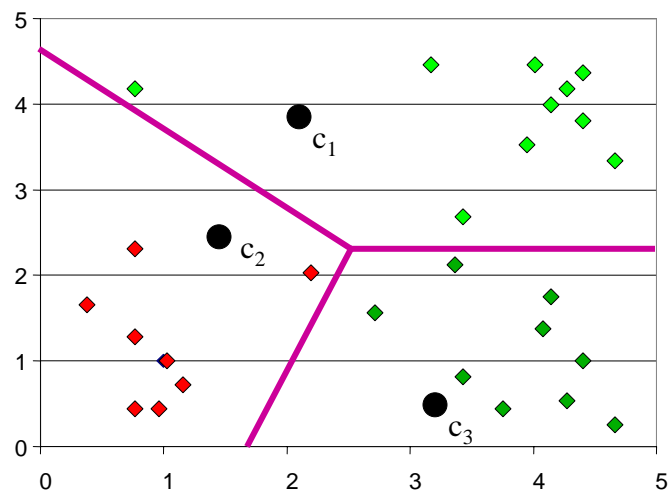
Randomly initialize the cluster centers



17

K-means clustering example

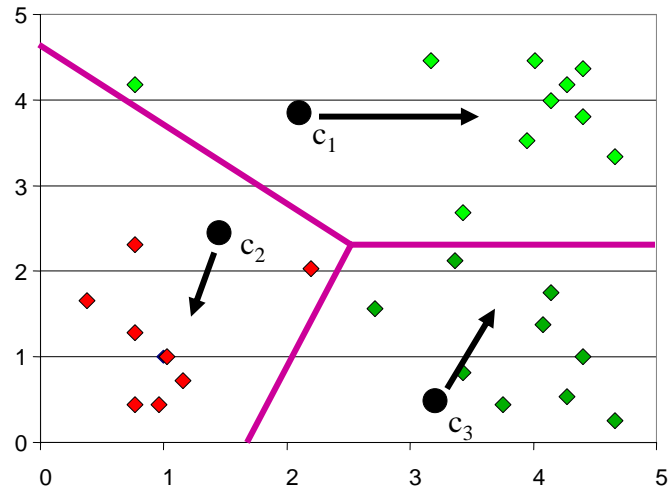
Determine cluster membership



18

K-means clustering example

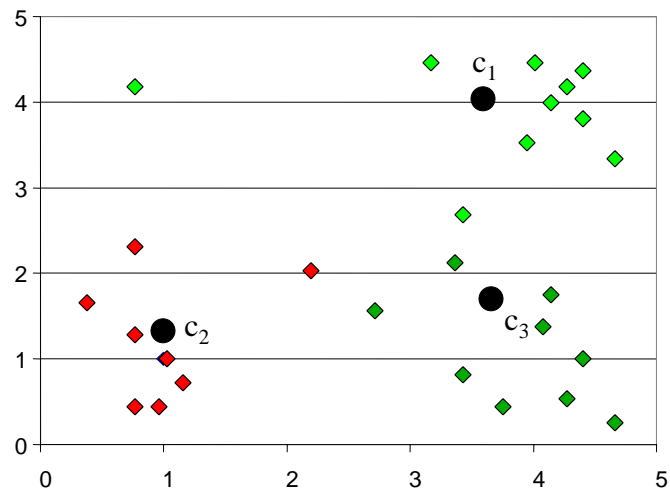
Re-estimate cluster centers



19

K-means clustering example

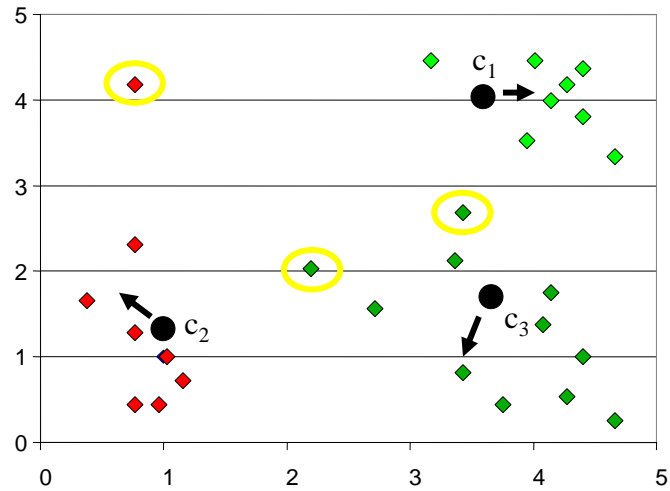
Result of first iteration



20

K-means clustering example

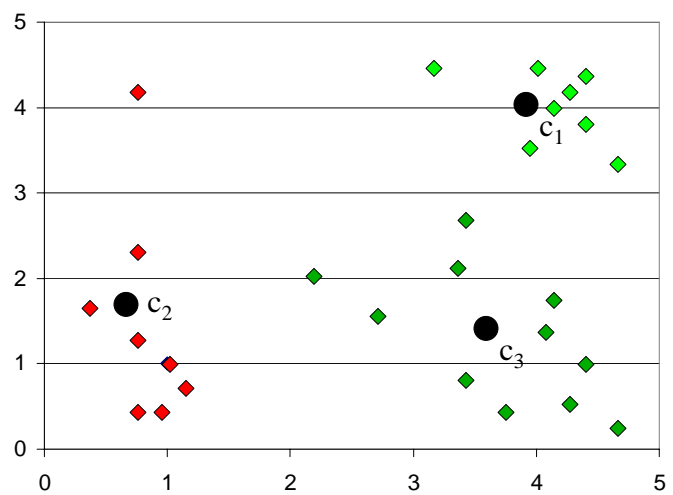
Second iteration



21

K-means clustering example

Result of second iteration



22

K-means clustering

Properties

- Will always converge to *some* solution
- Can be a “local minimum”
 - does not always find the global minimum of objective function:

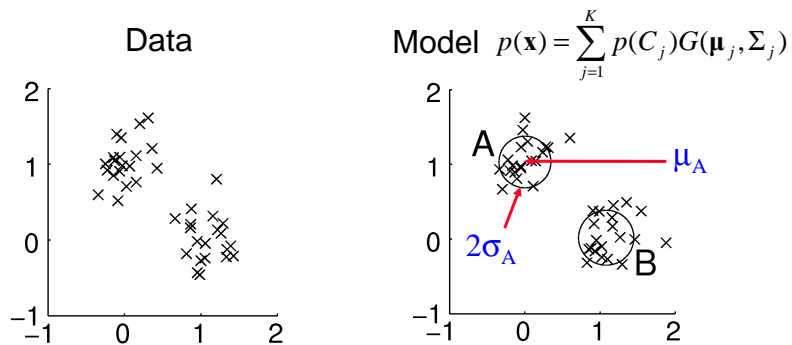
$$\sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$

23

K-means as probability density estimation

K-means can be formalized as estimating the unknown probability density of a data set

- Model data as a *mixture of K Gaussians*
- Estimate not only means but also (co)variances



24

K-Means and the EM Algorithm

- The Expectation Maximization (EM) Algorithm is a general algorithm for unsupervised learning when there are hidden variables (e.g., clusters, non-evidence nodes in Bayesian networks, etc.)
- Like K-means, it involves iterating between 2 steps:
 - E ("expectation") step that estimates posterior probabilities of hidden variables
 - M ("maximization") step that uses the result of E step to update model parameters
- Each iteration improves likelihood of data under the model (or keeps it the same)
- Guaranteed to converge (perhaps to local maximum)

25

Not to be confused with...

The Expectation Minimization Algorithm



26

Density estimation using EM

- EM for Gaussian mixtures (similar to K-means):
 - Initialize K clusters: C_1, \dots, C_K
 (μ_j, Σ_j) and $p(C_j)$ for each cluster j
 - Repeat until convergence:
 - Estimate which cluster each data point belongs to
 $p(C_j | x_i)$ \rightarrow Expectation step
 - Re-estimate cluster parameters
 $(\mu_j, \Sigma_j), p(C_j)$ \rightarrow Maximization step

27

EM algorithm: The details

E step: Compute probability of membership in cluster based on output of previous M step ($p(x_i | C_j) = \text{Gaussian}(\mu_j, \Sigma_j)$)

$$p(C_j | x_i) = \frac{p(x_i | C_j) \cdot p(C_j)}{p(x_i)} = \frac{p(x_i | C_j) \cdot p(C_j)}{\sum_j p(x_i | C_j) \cdot p(C_j)}$$

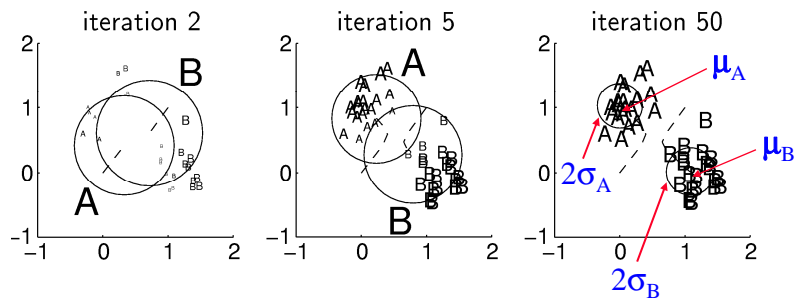
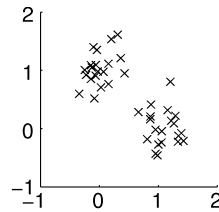
M step: Re-estimate cluster parameters based on output of E step

$$\mu_j = \frac{\sum_i p(C_j | x_i) \cdot x_i}{\sum_i p(C_j | x_i)} \quad \Sigma_j = \frac{\sum_i p(C_j | x_i) \cdot (x_i - \mu_j) \cdot (x_i - \mu_j)^T}{\sum_i p(C_j | x_i)} \quad p(C_j) = \frac{\sum_i p(C_j | x_i)}{N}$$

28

Results from the EM algorithm

Input data:



29

Suppose we are not interested in density estimation but want to reduce the dimensionality of our data

Example application: Image compression

Redundancy in Images

Most natural images (e.g., images of faces) are highly redundant

- Nearby pixels tend to have similar intensities and are therefore highly correlated

- Why?

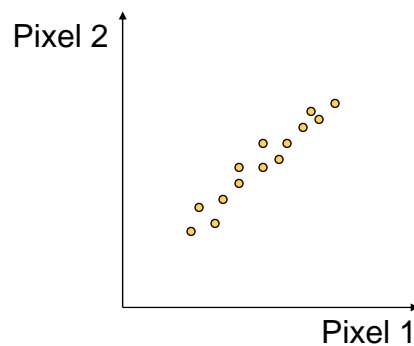
- Due to physical regularities of natural structures generating the image



Can we use unsupervised learning to capture and reduce this redundancy?

31

A Simple 2D Example

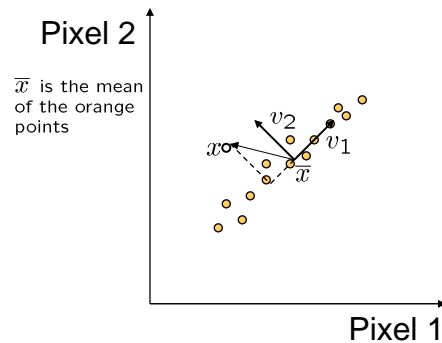


Suppose pixel 1 and pixel 2 are two neighboring pixels

- What does the plot above suggest?
- The two pixels are highly correlated for this data set of images

This and next few slides adapted from Steve Seitz, Linda Shapiro 32

Linear subspaces



Suppose we fit a line v_1
Let v_2 be orthogonal to v_1

Convert an input x into v_1, v_2 coordinates

$$x \rightarrow ((x - \bar{x}) \cdot v_1, (x - \bar{x}) \cdot v_2)$$

What does the v_1 coordinate measure?

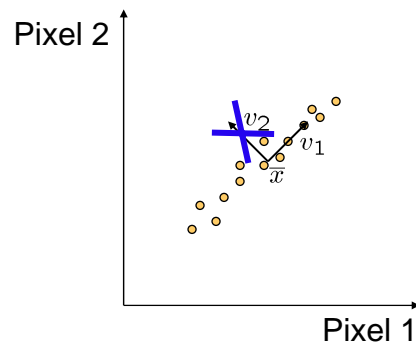
- position along v_1 axis
- use it to specify which point it is

What does the v_2 coordinate measure?

- distance to line (position along v_2 axis)
- near 0 for these pts

33

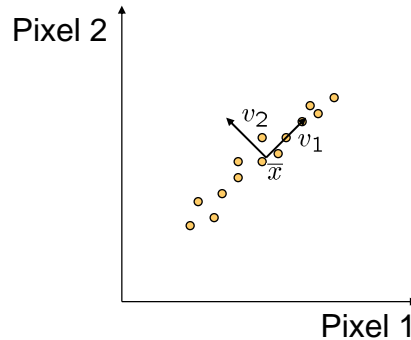
Dimensionality reduction



- We can represent the points with *only* their v_1 coordinates
 - since v_2 coordinates are all essentially 0
- Reduce dimensionality of data from 2D to 1D
- This makes it cheaper to store and compare points
- Bigger deal for higher dimensional inputs (like images!)

34

How do we find $\mathbf{v}_1, \mathbf{v}_2, \dots$?



Consider the variation along some direction \mathbf{v} for all of the N points:

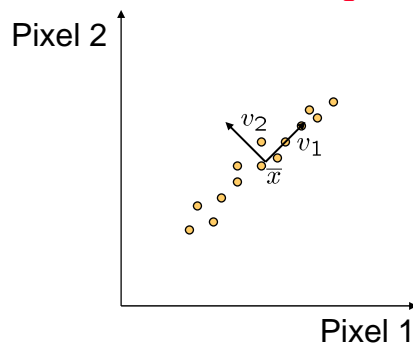
$$var(\mathbf{v}) = 1/N \sum_{\text{orange point } \mathbf{x}} \|(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{v}\|^2$$

What unit vector \mathbf{v} maximizes var ? $\mathbf{v}_1 = \underset{\mathbf{v}}{arg\ max} \{var(\mathbf{v})\}$

\mathbf{v}_2 is then the unit vector orthogonal to \mathbf{v}_1

35

How do we find $\mathbf{v}_1, \mathbf{v}_2, \dots$?



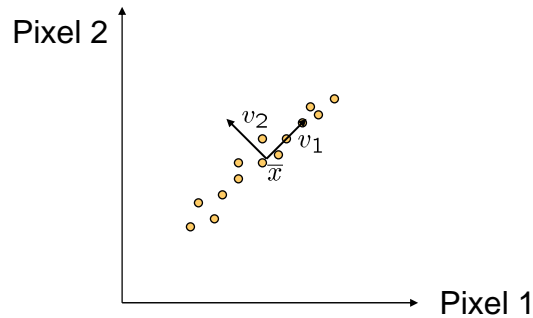
$$\begin{aligned} var(\mathbf{v}) &= \sum_{\mathbf{x}} \|(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{v}\|^2 / N \\ &= \sum_{\mathbf{x}} \mathbf{v}^T (\mathbf{x} - \bar{\mathbf{x}}) (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{v} / N \\ &= \mathbf{v}^T \left[\sum_{\mathbf{x}} (\mathbf{x} - \bar{\mathbf{x}}) (\mathbf{x} - \bar{\mathbf{x}})^T \right] \mathbf{v} / N \\ &= \mathbf{v}^T \mathbf{A} \mathbf{v} \quad \text{where } \mathbf{A} = \sum_{\mathbf{x}} (\mathbf{x} - \bar{\mathbf{x}}) (\mathbf{x} - \bar{\mathbf{x}})^T / N \end{aligned}$$

\mathbf{A} = Covariance matrix of data points

We want to find a unit vector \mathbf{v} that maximizes $\mathbf{v}^T \mathbf{A} \mathbf{v}$

36

Finding \mathbf{v}_1 and \mathbf{v}_2 : The Math



$$\mathbf{v}_1 = \operatorname{argmax}_{\mathbf{v}} (\mathbf{v}^T \mathbf{A} \mathbf{v}) \text{ subject to } \mathbf{v}^T \mathbf{v} = 1$$

Using Lagrange multiplier method,

$$\mathbf{v}_1 = \operatorname{argmax}_{\mathbf{v}} [\mathbf{v}^T \mathbf{A} \mathbf{v} - \lambda (\mathbf{v}^T \mathbf{v} - 1)]$$

Setting derivative wrt \mathbf{v} to 0, we get:

$$\mathbf{A} \mathbf{v} = \lambda \mathbf{v} \quad \text{Thus, } \mathbf{v}_1 \text{ is eigenvector of } \mathbf{A} \text{ with largest eigenvalue } \lambda_1$$

\mathbf{v}_2 is eigenvector of \mathbf{A} with smaller eigenvalue λ_2

37

Principal Component Analysis (PCA)

Suppose each of the N data points is L -dimensional

- Form $L \times L$ data covariance matrix A

$$A = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

- Compute eigenvectors of A
 - Eigenvectors of A define a new coordinate system that is a rotation of the original coordinate system
 - Eigenvector with largest eigenvalue captures the most variation among training vectors \mathbf{x}
 - Eigenvector with smallest eigenvalue has least variation

38

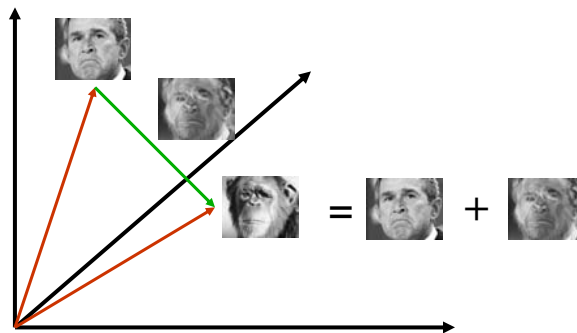
Principal Component Analysis (PCA)

We can compress the data by only using the top few eigenvectors with largest eigenvalues

- corresponds to choosing a "linear subspace" of the original data space
- represent points on a line, plane, "hyper-plane"
- these eigenvectors are known as *principal component vectors*
- procedure is known as *Principal Component Analysis (PCA)*

39

Back in the vector space of faces...

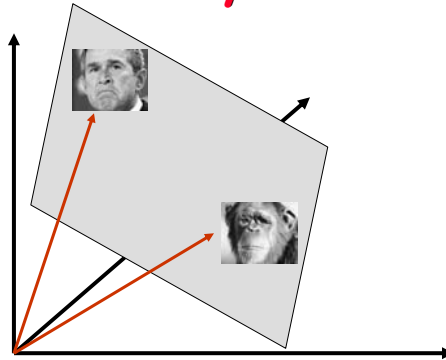


An image is a point in a high dimensional space

- A $P \times Q$ pixels image is a point in R^{PQ}
- Vectors in this space behave similarly to the data points in our 2D case

40

Dimensionality reduction



The space of all faces is a "subspace" of the space of all images

- Suppose this subspace is K dimensional, $K \ll PQ$
- We can find such a subspace using PCA
- This is like fitting a "hyper-plane" to the set of faces
 - spanned by eigenvectors v_1, v_2, \dots, v_K
 - any face $x \approx \bar{x} + a_1v_1 + a_2v_2 + \dots + a_kv_k$

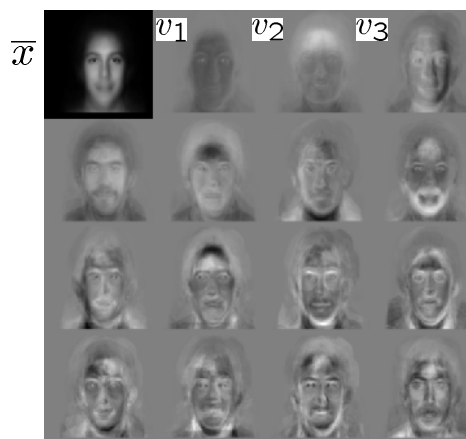
41

Eigenfaces

PCA extracts the eigenvectors $v_1, v_2, v_3, \dots, v_K$ of covariance matrix A

Each one of these vectors is a direction in face space

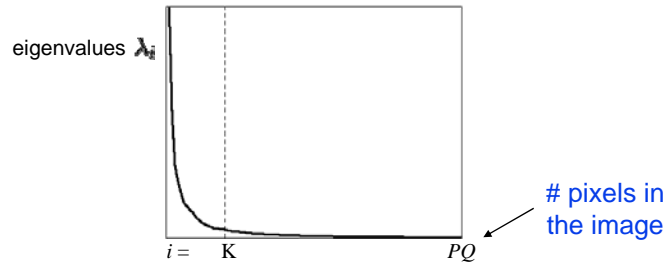
- what do these look like?



The eigenvectors for face images are called "eigenfaces"

42

Choosing the reduced dimension K



How many eigenfaces to use?

Look at the decay of the eigenvalues

- the eigenvalue tells you the amount of variance "in the direction" of that eigenface
- ignore eigenfaces with low variance

43

Application 1: Image Compression

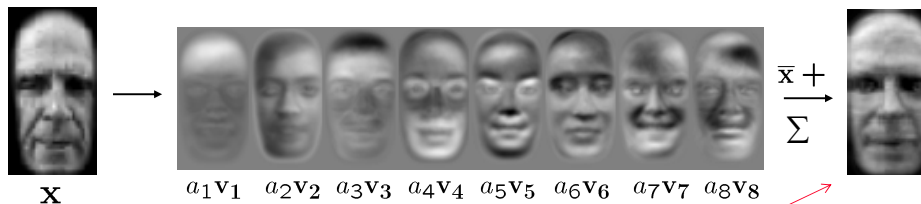
The eigenfaces v_1, \dots, v_K span the space of faces

- An image is converted to eigenface coordinates using dot products ("projection"):

$$x \rightarrow ((x - \bar{x}) \cdot v_1, (x - \bar{x}) \cdot v_2, \dots, (x - \bar{x}) \cdot v_K)$$

Input image (P x Q pixels) $\underbrace{\hspace{1.5cm}}_{a_1} \quad \underbrace{\hspace{1.5cm}}_{a_2} \quad \dots \quad \underbrace{\hspace{1.5cm}}_{a_K}$

Compressed representation of face (K usually much smaller than PQ)



Reconstructed face $x \approx \bar{x} + a_1v_1 + a_2v_2 + \dots + a_Kv_K$

44

Reconstruction using Eigenfaces

- Given image on left, project to Eigenspace, then reconstruct an image (right).



[Turk & Pentland 01]

45

Application 2: Face Recognition

Algorithm:

1. Given a set of images with labels (e.g., names)
 - Run PCA and compute K eigenfaces
 - Calculate and store the K coefficients for each image with its label

2. Given a new image x , calculate K coefficients

$$x \rightarrow (a_1, a_2, \dots, a_K)$$

3. Verify that x is a face

$$\|x - (\bar{x} + a_1v_1 + a_2v_2 + \dots + a_Kv_K)\| < \text{threshold}$$

4. If it is a face, who is it?

- Find label of closest face in database
- Nearest-neighbor in K-dimensional space

46

Break

Next: Applications in Robotic Learning
(Guest Lecture by Dr. Chalodhorn)