Computer Vision: Fall 2022 — Lecture 3

Dr. Karthik Mohan

Univ. of Washington, Seattle

October 6, 2022

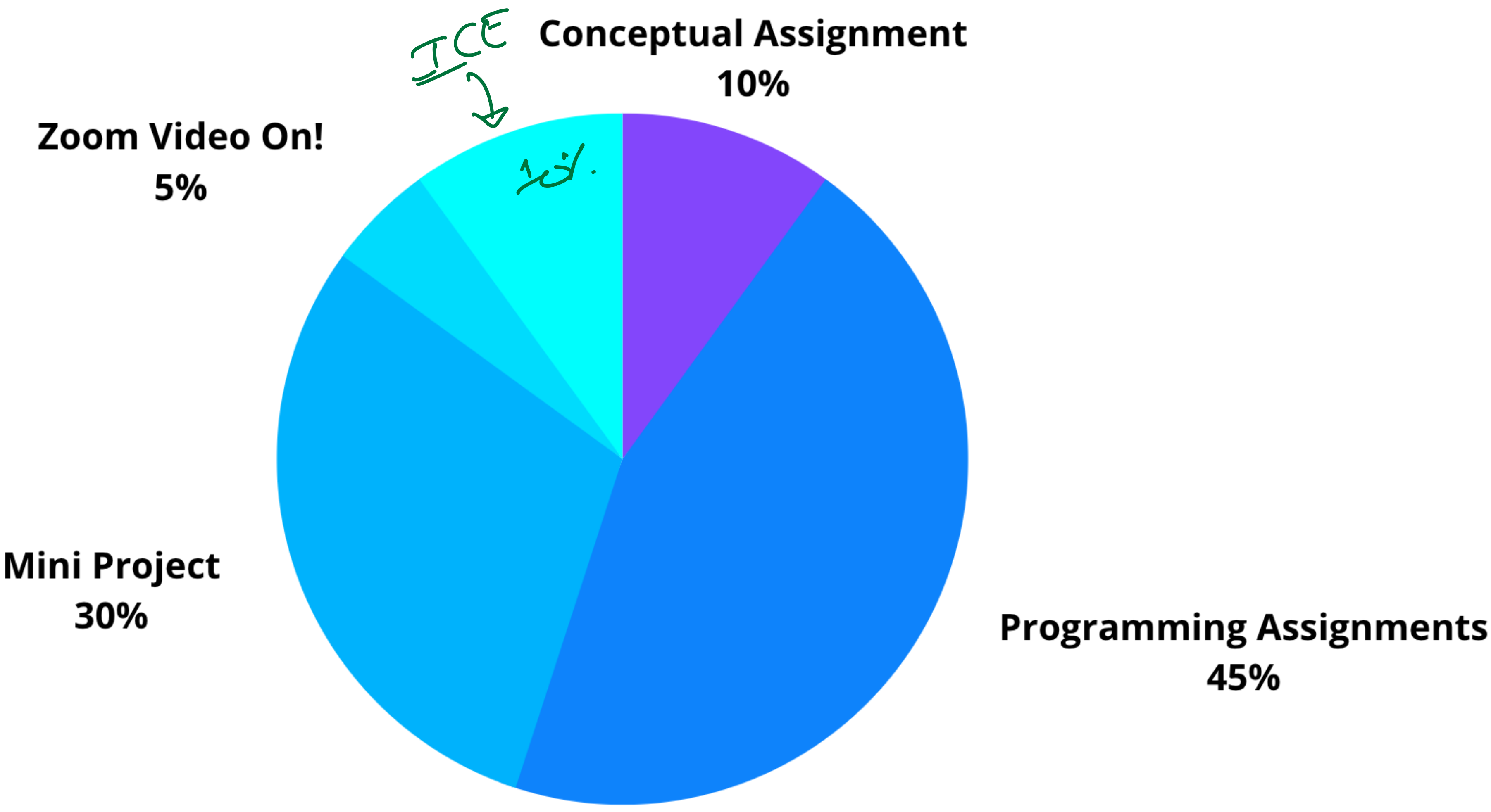
Weekly Logistics

	Day	Timings	Class type
Lecture 1 (In-person)	T	4 pm - 6 pm	(In-person)
Lecture 2	Th	4 pm - 6 pm	Zoom
Office Hours Karthik	T	6 - 6:30 pm	In-person/Zoom
Calendly 15 min Karthik	October		Zoom
Office Hours Ayush	Fri	5-6 pm	Zoom
Quiz Section Ayush	Mon	5-6 pm	Zoom

References for Lecture

1. Image Compression with SVD
2. Wikipedia on Image Convolutions
3. Convolution Playground
4. Deep Learning TextBook by Yoshua Bengio et al } Future

Assessments Breakdown



Today

1. SVD and Image applications
2. Matrix Arithmetic Refresher
3. Convolutions and Image Processing
4. Introduction to clustering and kMeans

ML
Modeling use cases

Notebook on SVD

ICE #1

Matrix Arithmetic

Let $X = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{bmatrix}$ Then the reduced SVD of X with $k = 2$ and *rounded to 1 decimal place* is given by:

```
print(np.round(Utilde,1))
print(np.round(Vtilde,1))
print(np.round(np.diag(Stilde),1))
```

```
[81]  ✓ 0.5s
```

```
[[-0.2  1. ]
 [-0.5  0. ]
 [-0.8 -0.3]]
[[-0.5 -0.6 -0.7]
 [-0.8  0.   0.6]]
[[17.4  0. ]
 [ 0.   0.9]]
```

ICE #1

Matrix Arithmetic

```
print(np.round(Utilde,1))
print(np.round(Vtilde,1))
print(np.round(np.diag(Stilde),1))
[81] ✓ 0.5s
... [[-0.2  1. ]
     [-0.5  0. ]
     [-0.8 -0.3]]
    [[-0.5 -0.6 -0.7]
     [-0.8  0.   0.6]]
    [[17.4  0. ]
     [ 0.   0.9]]
```

Which of the following is true for a) The dot product of the first and second column of the given *Utilde* and b) The dot product of the first and second row of *Vtilde*

1. equals 0 and equals 0
2. close to 0 and close to 0
3. equals 0 and close to 0
4. close to 0 and equals 0

ICE #2

Matrix Rank and Singular Factors

Let $X = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$. Notice we replaced the $(2, 2)$ element with 9 instead of 10 from the previous ICE. Which of the following is true of X (you can say this even without having to compute the SVD of X):

1. The matrix rank of X is 2 and the number of non-zero singular values of X is 2
2. The matrix rank of X is 3 and the number of non-zero singular values of X is 3
3. The matrix rank of X is 3 and the number of non-zero singular values of X is 2
4. The matrix rank of X is 2 and the number of non-zero singular values of X is 3

Two ways to multiply Matrices

1) U_{tilde} S_{tilde} V_{tilde}

3×2 $\quad$ 2×2 $\quad$ 2×3

2)

$$u_1 \quad u_2 \quad v_1 \quad v_2$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \end{bmatrix}$$

$1\times 7+2\times 9$

$u_1 \cdot v_1$

$u_2 \cdot v_1$

$3\times 8+4\times 10$

$u_2 \cdot v_2$

2×2

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \end{bmatrix}$$

$$= \left[7 \times \begin{bmatrix} 1 \\ 3 \end{bmatrix} + 9 \times \begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad 8 \times \begin{bmatrix} 1 \\ 3 \end{bmatrix} + 10 \times \begin{bmatrix} 2 \\ 4 \end{bmatrix} \right]$$

$$X = \overbrace{\begin{bmatrix} U_1 & U_2 & U_3 & \cdots & U_n \end{bmatrix}}^{U} \times \begin{bmatrix} V_1 & V_2 & \cdots & V_m \end{bmatrix}$$

$$X = \begin{bmatrix} UV_1 & UV_2 & UV_3 & \cdots & UV_m \end{bmatrix}$$

$$U_1 V_{11} + U_2 V_{22} + U_3 V_{33} + \cdots$$

vector scalar

ICE #3

Matrix multiplication

Let $X = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$ and $Y = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$. Let $Z = XY$. Note that $Z = [XY_1 XY_2]$. What is Z_2 here?

1. $[11, 32]$
2. $[32, 10]$
3. $[10, 32]$
4. $[11, 10, 32]$

$3\begin{bmatrix} 1 \\ 4 \end{bmatrix} + 4\begin{bmatrix} 2 \\ 5 \end{bmatrix}$

Matrix Arithmetic

How does SVD multiply translate to additive decomposition?

$$X = U \Sigma V$$

Singular values

$$= \underbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & 0 \\ \vdots & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_n \end{bmatrix}} V$$

$$= \begin{bmatrix} \sigma_1 u_1 & \sigma_2 u_2 & \sigma_3 u_3 & \cdots & \sigma_n u_n \end{bmatrix} V$$

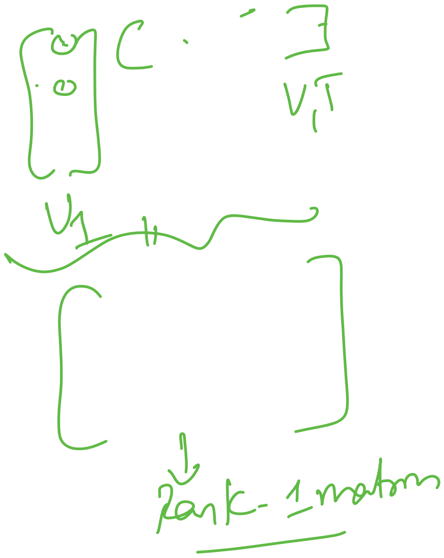
$$X = [\sigma_1 U_1 \ \sigma_2 U_2 \ \cdots \ \sigma_n U_n] \begin{bmatrix} V_1^T \\ V_2^T \\ \vdots \\ V_n^T \end{bmatrix}$$

KEEP

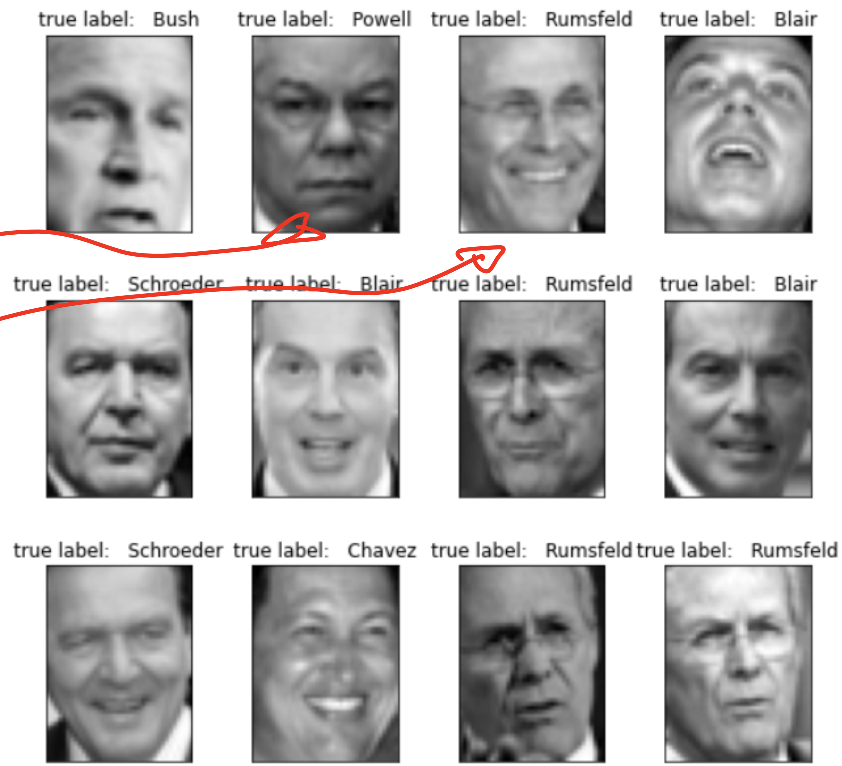
THROW AWAY

$$\boxed{X = \underbrace{\sigma_1 U_1 V_1^T + \sigma_2 U_2 V_2^T + \cdots + \sigma_k U_k V_k^T}_{} + \cdots + \sigma_n U_n V_n^T}$$

⊳ ADDITIVE DECOMP. of the data matrix X into n Rank-1 matrices



Eigen Faces



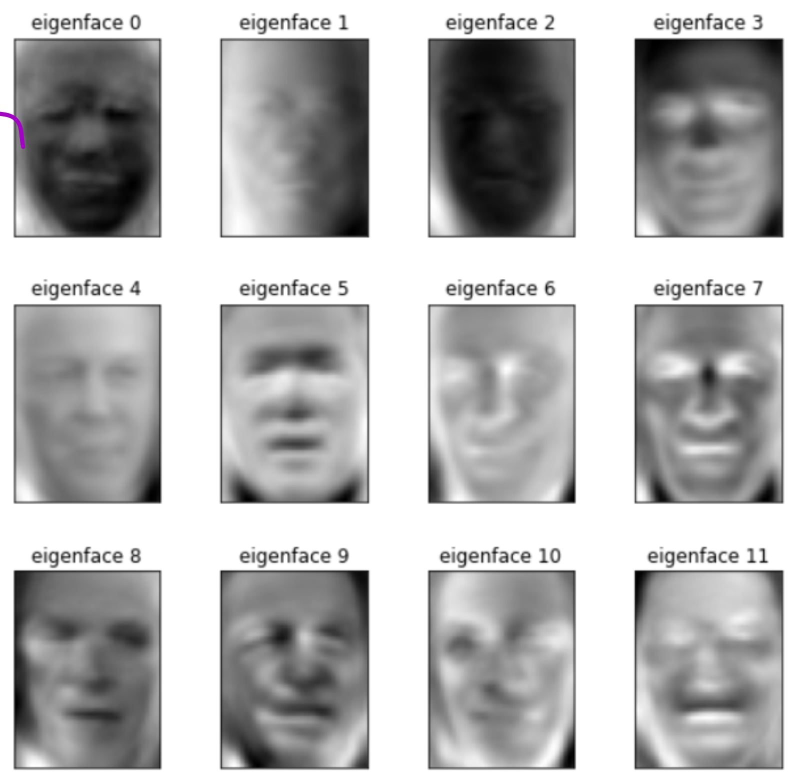
Training Image with True Label (LFW people's dataset)

$X =$

Image that is rep. as a vector

Eigen Faces

These u_j are called **EigenFaces**.



EigenFaces

100×200

V_1

$20k \times 1$

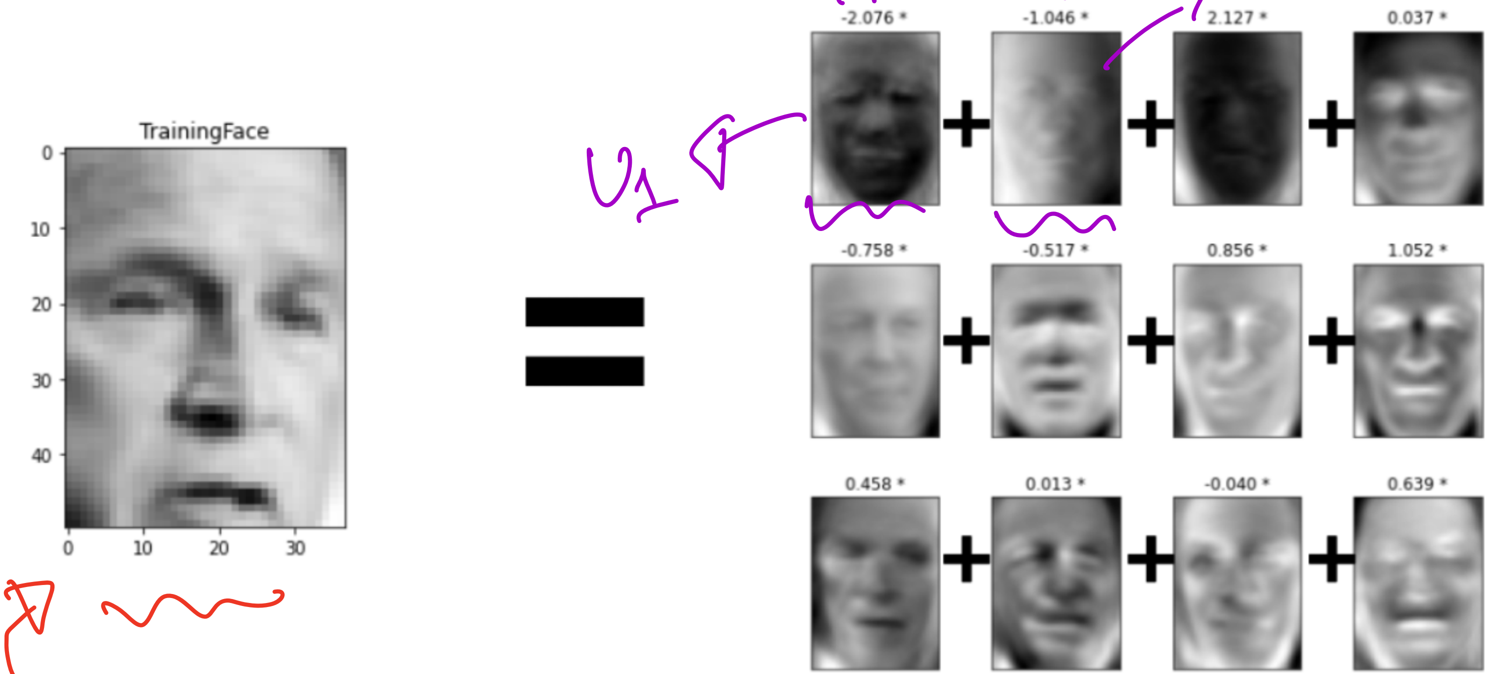
$X = USV^T$

$\left[V_1 \; V_2 \; - - - - \; V_k \right]$

Eigen Faces

$$X_5 = \underbrace{U_1}\,\underbrace{W_{51}} + U_2 W_{52} + U_3 W_{53} + \cdots + U_{12} W_{5(12)}$$

W_{51} W_{52} U_2 U_1



Linear Combination of EigenFaces

orthogonality
— captures different pieces of info.

(k = 12)

X_5

$$\underbrace{\Sigma V^T}_{W}$$

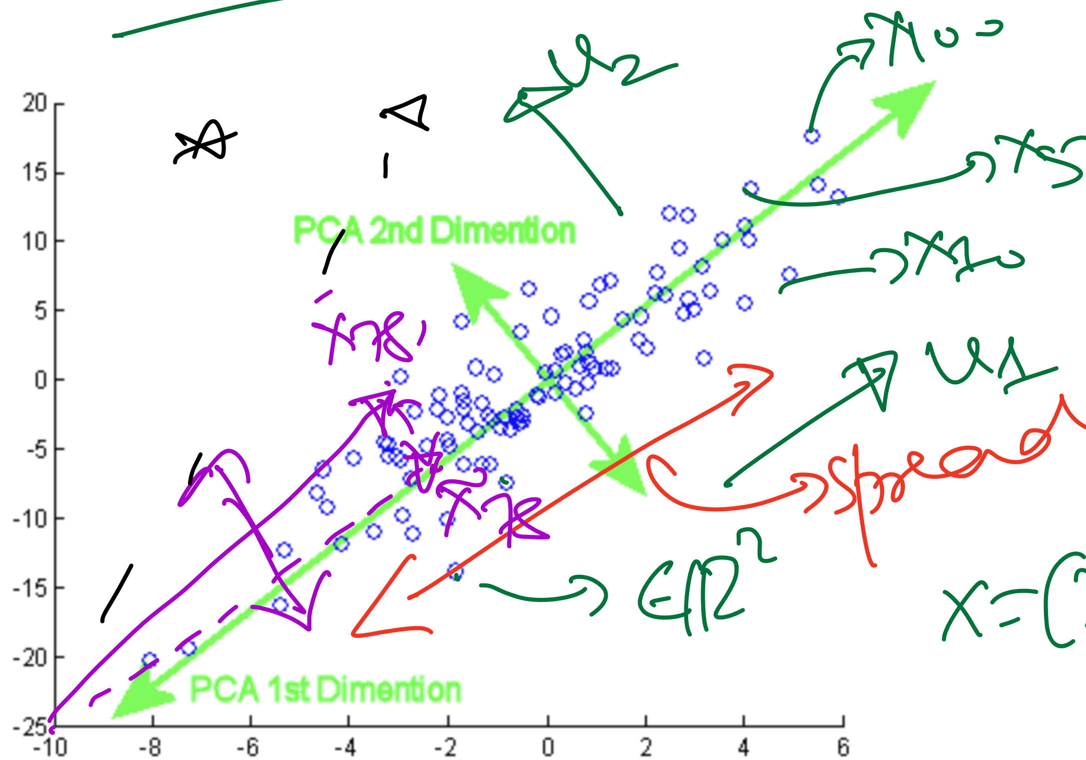
$$[X_1\ X_2 \cdots X_5 \cdots X_{100}] = [\underbrace{U_1\ U_2 \cdots U_5 \cdots U_{12}}][W_1\ W_2 \cdots W_5 \cdots W_{100}]$$

Understanding Matrix Math behind Eigen Faces

Linear Combination of EigenFaces

SVD and PCA

↳ Principal Component Analysis (Statistics)



$X = (x_1, x_2 \; \text{----} \; x_n)$

↳ $\in \mathbb{R}^2$

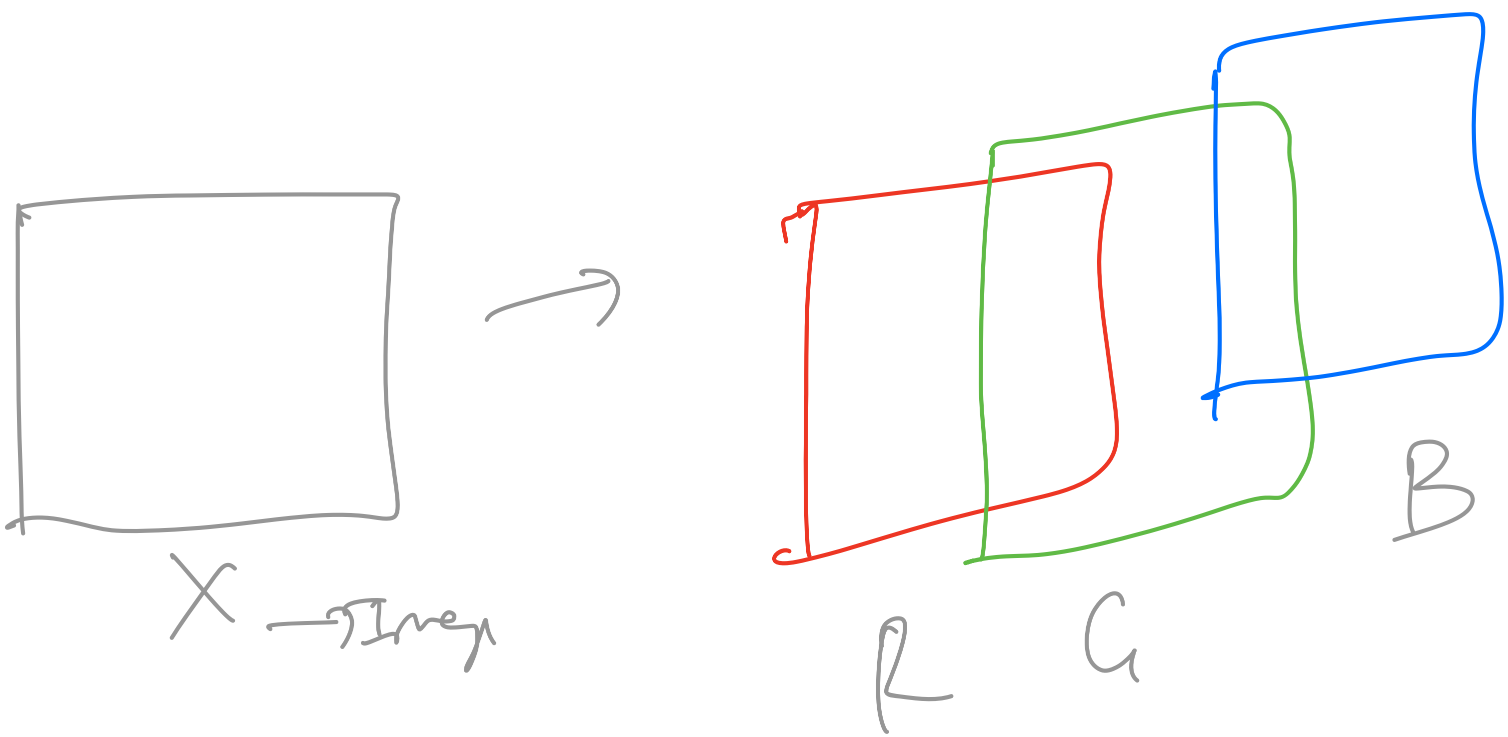
$= U \Sigma V^T$

(SVD) ↳ Principal Components

Assignment 1: Data compression using SVD

Color Color

1. Given an image I - Let R, G, B the matrices corresponding to the Red, Green and Blue Channels of the image matrix



Assignment 1: Data compression using SVD

1. Given an image I - Let R, G, B the matrices corresponding to the Red, Green and Blue Channels of the image matrix
2. Compute SVD of R, G, B - So 3 separate SVDs

Assignment 1: Data compression using SVD

1. Given an image I - Let R, G, B the matrices corresponding to the Red, Green and Blue Channels of the image matrix
2. Compute SVD of R, G, B - So 3 separate SVDs
3. Pick a k for reduced SVD and compute the reduced SVD factors for R,G,B

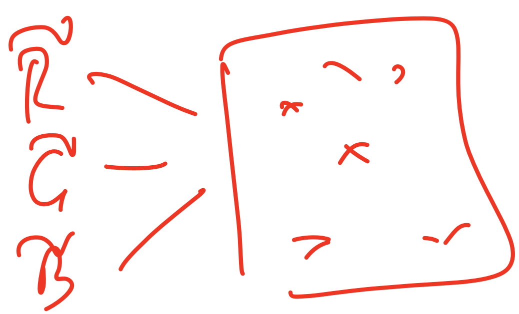
Assignment 1: Data compression using SVD

1. Given an image I - Let R, G, B the matrices corresponding to the Red, Green and Blue Channels of the image matrix
2. Compute SVD of R, G, B - So 3 separate SVDs
3. Pick a k for reduced SVD and compute the reduced SVD factors for R,G,B
4. Compute the reconstructed $\tilde{R}, \tilde{G}, \tilde{B}$ from the reduced SVD factors of each of the matrices

$$\tilde{R} = \tilde{U}_R \tilde{\Sigma}_R \tilde{V}_R^T$$

Assignment 1: Data compression using SVD

1. Given an image I - Let R, G, B the matrices corresponding to the Red, Green and Blue Channels of the image matrix
2. Compute SVD of R, G, B - So 3 separate SVDs
3. Pick a k for reduced SVD and compute the reduced SVD factors for R,G,B
4. Compute the reconstructed $\tilde{R}, \tilde{G}, \tilde{B}$ from the reduced SVD factors of each of the matrices
5. Use $\tilde{R}, \tilde{G}, \tilde{B}$ to reconstruct the image from compression.



Assignment 1: Data compression using SVD

1. Given an image I - Let R, G, B the matrices corresponding to the Red, Green and Blue Channels of the image matrix
2. Compute SVD of R, G, B - So 3 separate SVDs
3. Pick a k for reduced SVD and compute the reduced SVD factors for R,G,B
4. Compute the reconstructed $\tilde{R}, \tilde{G}, \tilde{B}$ from the reduced SVD factors of each of the matrices
5. Use $\tilde{R}, \tilde{G}, \tilde{B}$ to reconstruct the image from compression.
6. Plot the reconstructed images for at least 3 different compression factors (e.g. 2, 5, 10).

Next Topic: Image Processing with Convolutions

What is a convolution?

Convolution

Mathematical operation of sliding a convolution matrix (or kernel) across an input matrix. As the sliding happens, the window of the input matrix gets averaged by the convolution matrix to get a scalar. The scalar is added to the output matrix.

$C = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ 3x3

Blurring an Image

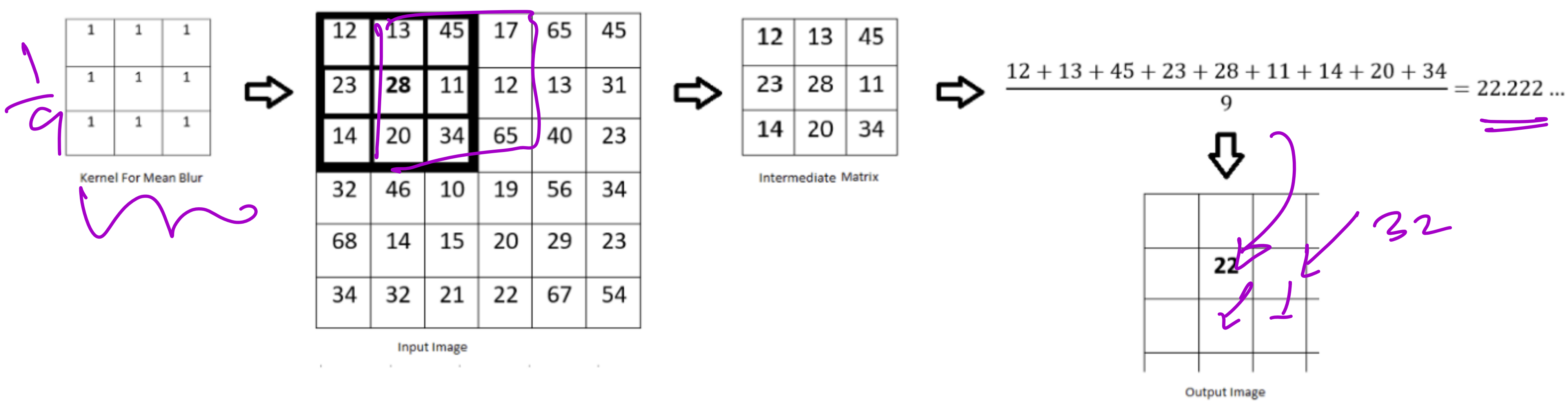
Submatrix · C

Dot-product

O(n^2) size of window

Box Blur Convolution

Box Blur



Applying Box Blur On An Image.

ICE #4

$\frac{1}{4}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}$

Box Blur

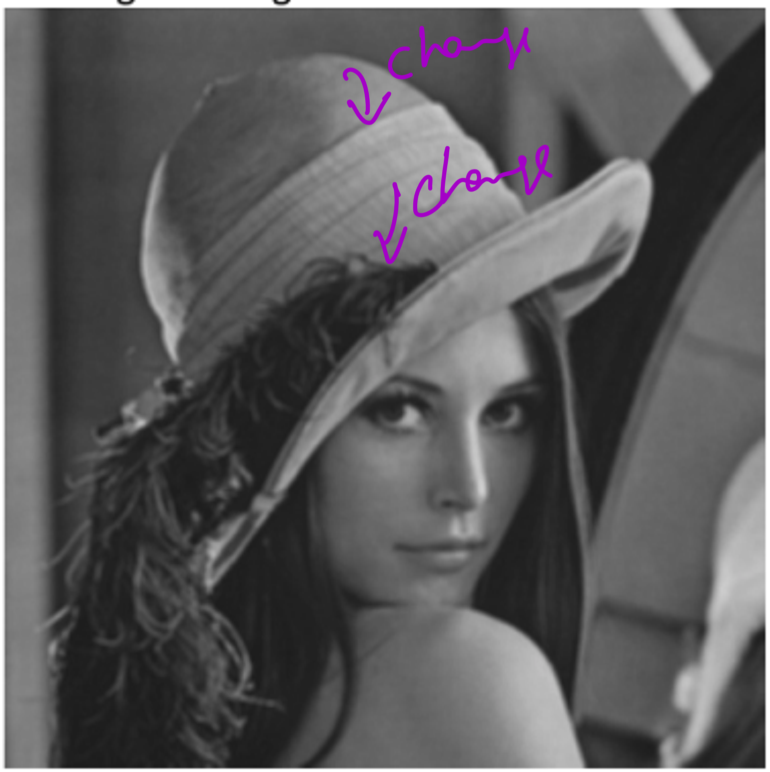
Let $X = \begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$ be the input image. Use the 2x2 box blur to convolve against X. What is the output matrix look like?

1. $\begin{bmatrix}1 & 2\\ 4 & 5\end{bmatrix}$
2. $\begin{bmatrix}3 & 5\\ 8 & 9\end{bmatrix}$
3. $\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}$
4. $\begin{bmatrix}3 & 4\\ 6 & 7\end{bmatrix}$

Edge Detection Kernel

Laplacian Kernel

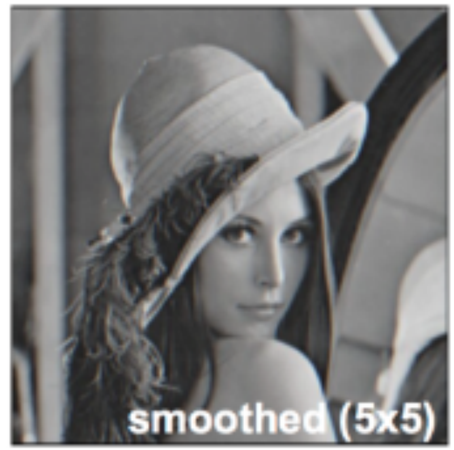
Original Image



Edge Detected

Edge Detection Kernel

−



=

Edge Image

•0	•0	•0
•0	•1	•0
•0	•0	•0

−

$\frac{1}{9}$

•1	•1	•1
•1	•1	•1
•1	•1	•1

Box Blur

$$= \begin{bmatrix} -1/9 & -1/9 & -1/9 \\ -1/9 & 8/9 & -1/9 \\ -1/9 & -1/9 & -1/9 \end{bmatrix} \to C$$

$$X * C = O$$

Edge detection Kernel

Laplacian Edge detection

$C = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$. Then C is called the Laplacian edge detection kernel and identifies edges in images. This is one of the ways to produce edges.

Sobel Edge detection

Original Image:

Edge Detected:

ICE #5

What does this Kernel do?

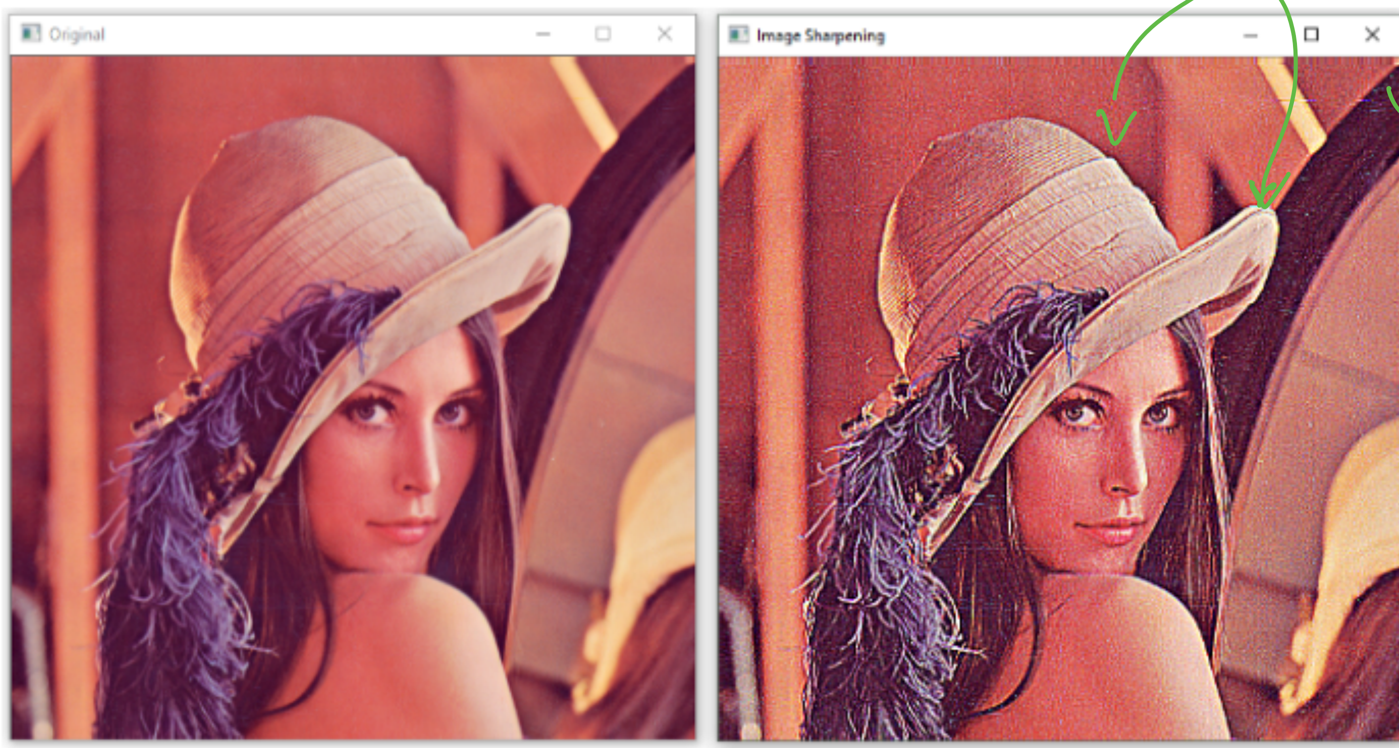
Let $C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. What happens when C is convolved with an image?

1. It blurs the image
2. It sharpens the image
3. It finds edges in the image
4. It leaves the image unchanged

Sharpening an Image

Sharper

Details



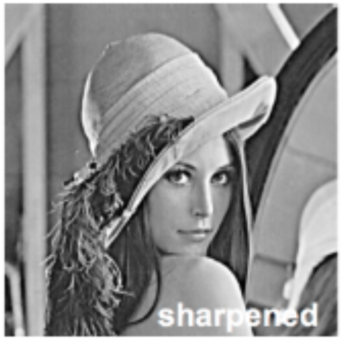
Original Image(Left) and Image after applying Sharpen Filter of size 3×3 (Right)

Sharpening an Image

+ Edges

+ a

=



$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Identity Kernel

Box Blur Kernel

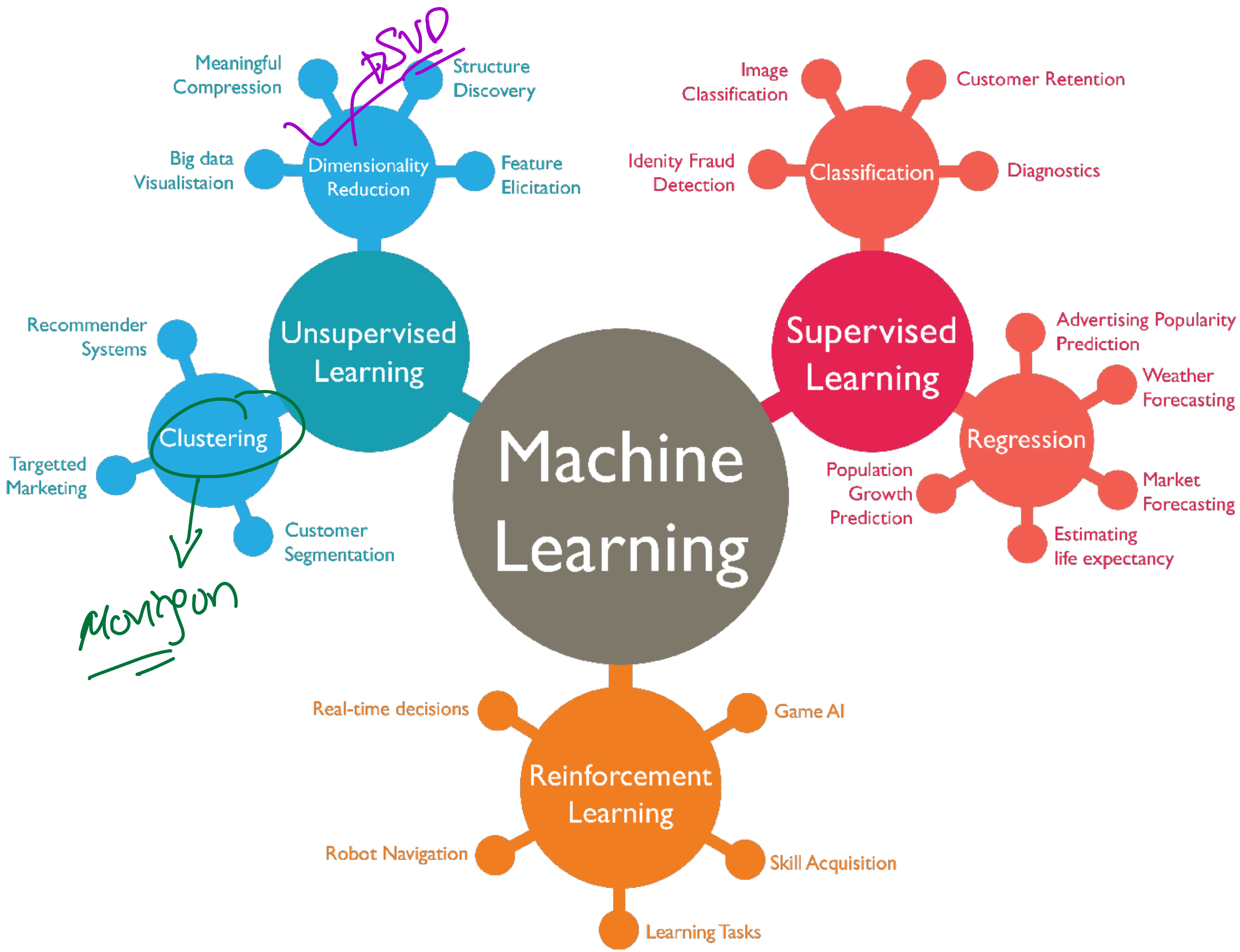
Sharpening kernel (Based on Box Blur)

Convolution Playground

Convolution Playground

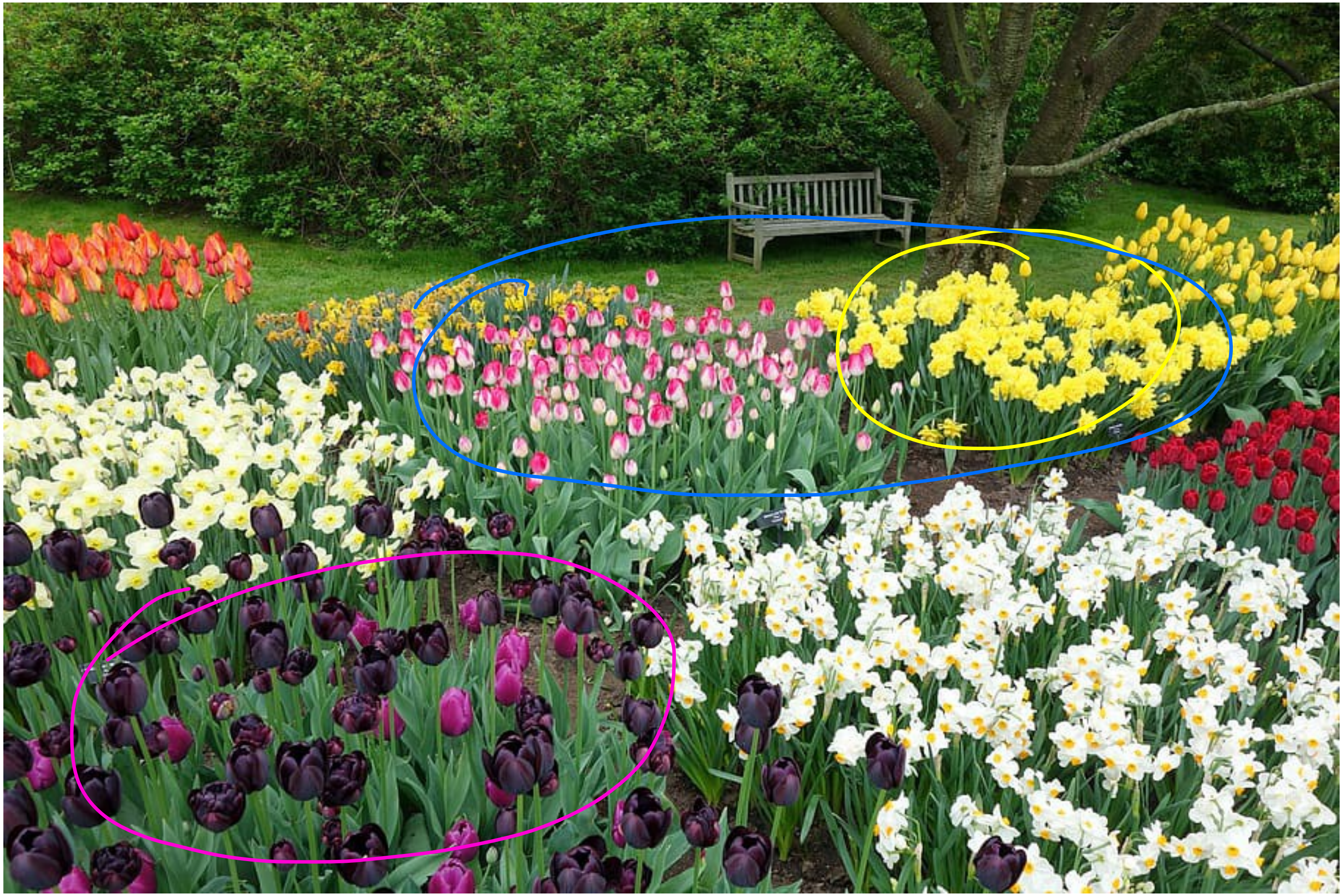
Next Topic: Clustering of Data/Images

Big Picture


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
SVD
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

The clustering problem

K → #clusters



Clustering vs Classification

Difference

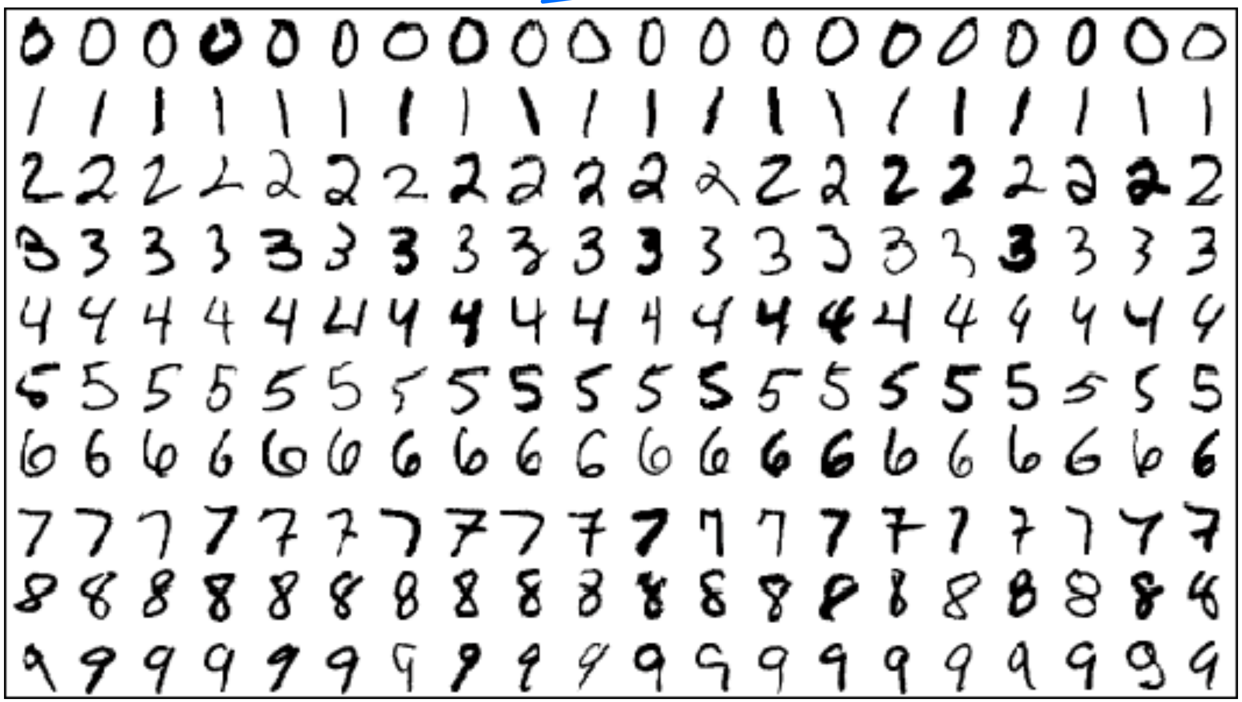
Input Target/output

In the classification problem, you are given (x^i, y_i) - I.e. both the data point i and its true label y_i for training purposes. Example - a flower i and its label (flower type). Whereas in clustering problem, you are just given the data points, i.e. x^i. However, you still want to break up the data points into clusters - where each cluster has relatively similar data points.

Digits Clustering

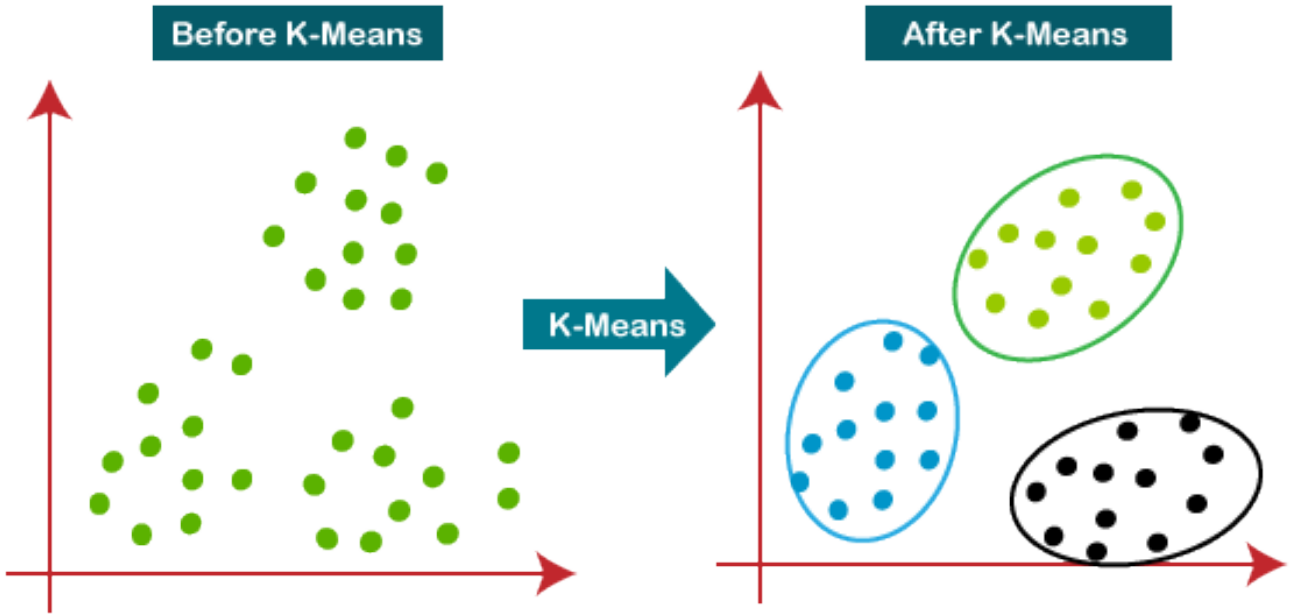
MNIST Digits

Each row is a cluster



(k=10) makes sense.

Clustering of data points



Clustering for News

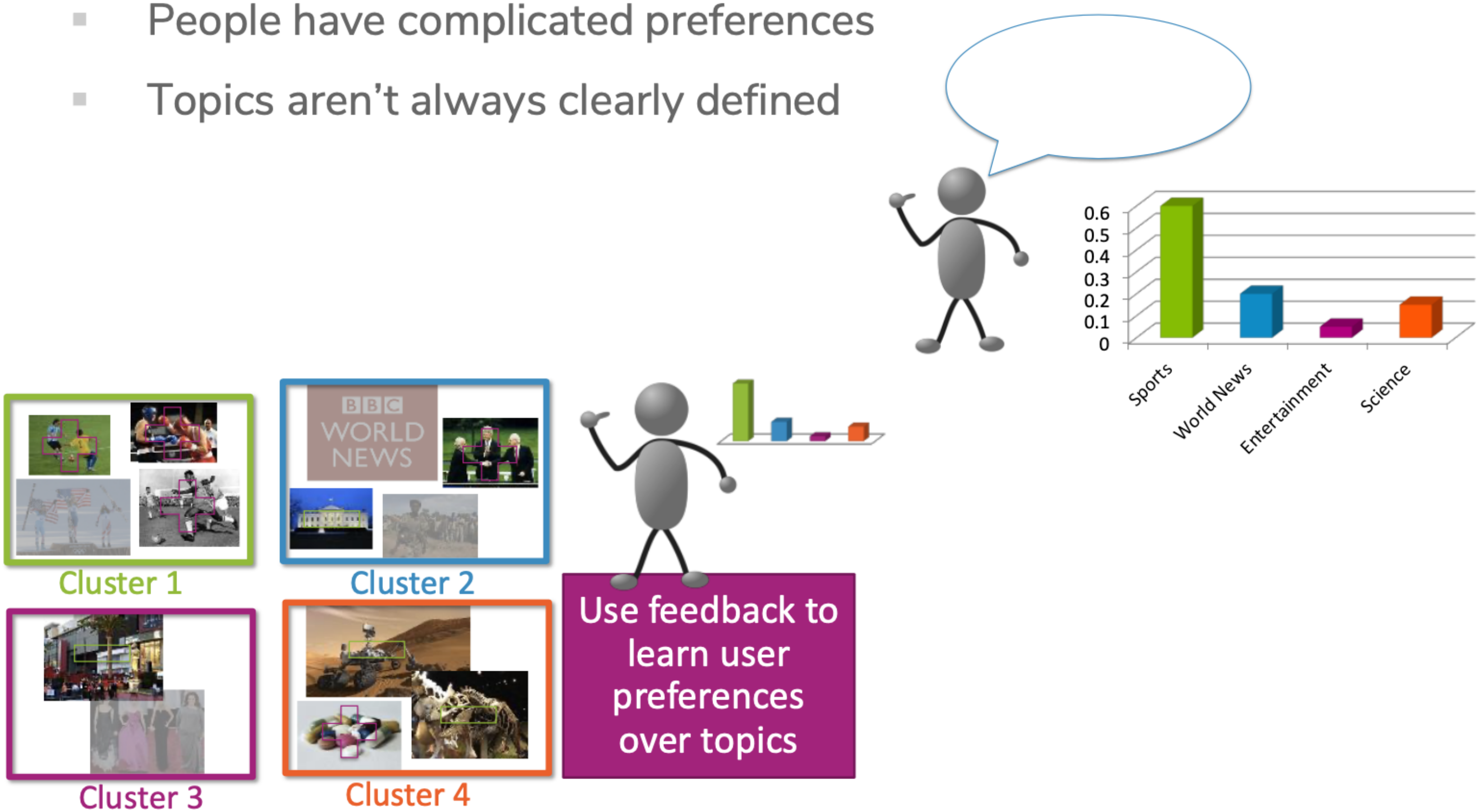
SPORTS

WORLD NEWS

Clustering for News

User preferences are important to learn, but can be challenging to do in practice.

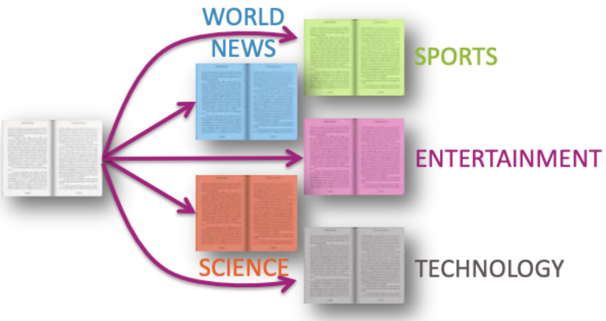
- People have complicated preferences
- Topics aren't always clearly defined



Clustering for News

What if the labels are known? Given labeled training data

Can do multi-class classification methods to predict label



Clustering Basics

- In many real world contexts, there aren't clearly defined labels so we won't be able to do classification
- We will need to come up with methods that uncover structure from the (unlabeled) input data X.
- **Clustering** is an automatic process of trying to find related groups within the given dataset.

$Input: x_1, x_2, \dots, x_n$

$Output: z_1, z_2, \dots, z_n$

Clustering Basics

In their simplest form, a **cluster** is defined by

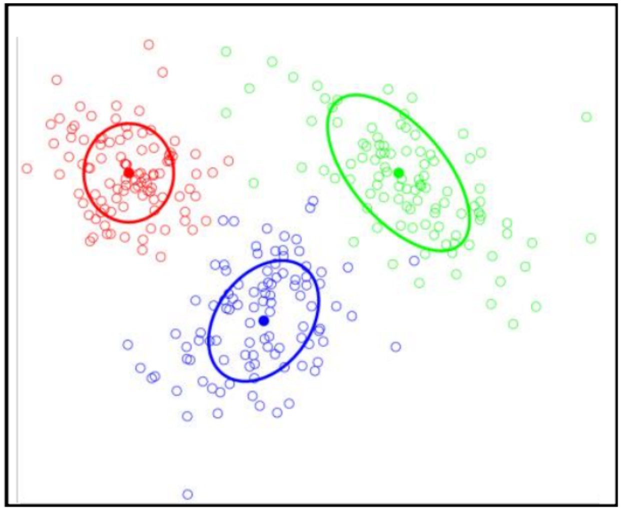
- The location of its center (**centroid**)
- Shape and size of its **spread**

Clustering is the process of finding these clusters and **assigning** each example to a particular cluster.

- x_i gets assigned $z_i \in [1, 2, \dots, k]$
- Usually based on closest centroid

Will define some kind of score for a clustering that determines how good the assignments are

- Based on distance of assigned examples to each cluster



Distance typically used

Euclidean Distance

Distance between two points, x_1, x_1 is given by:

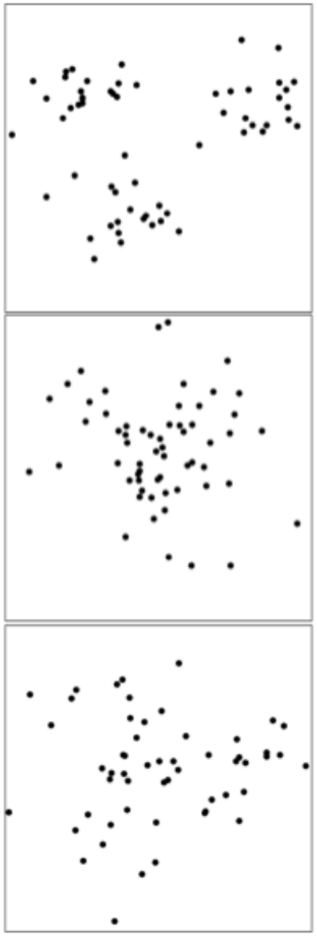
$$\|x_1 - x_2\|_2$$

Clustering on different Data sets

Clustering is easy when distance captures the clusters

Ground Truth (not visible)

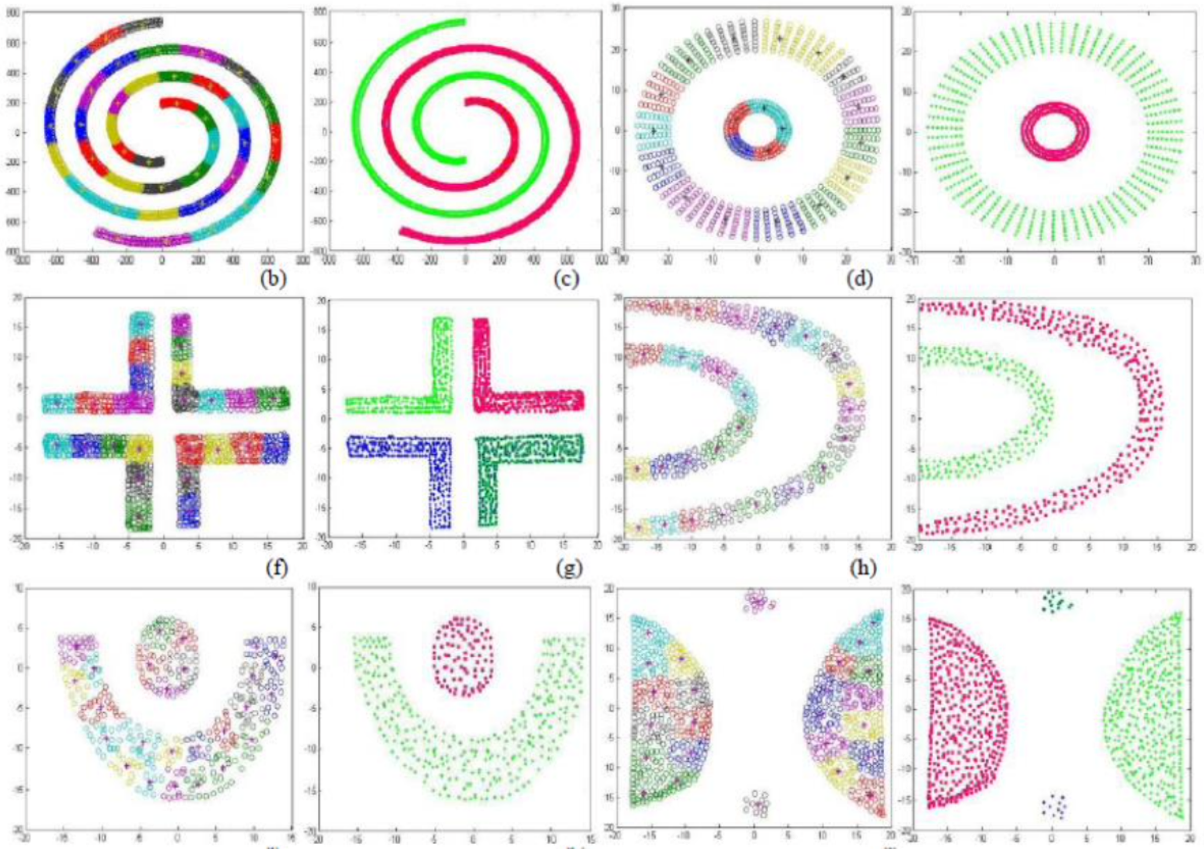
Given Data



Clustering - Hard cases

There are many clusters that are harder to learn with this setup

- Distance does not determine clusters



k-means

k-means++

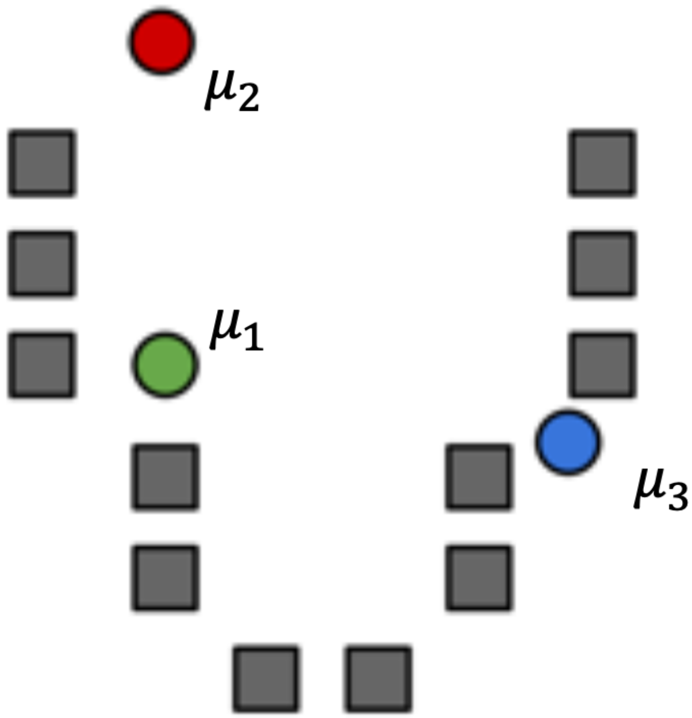
Algorithm 1 k-means algorithm

Specify the number k of clusters to assign.
Randomly initialize k centroids.
repeat
    expectation: Assign each point to its closest centroid.
    maximization: Compute the new centroid (mean) of each cluster.
until The centroid positions do not change.

k-means Clustering

Start by choosing the initial cluster centroids

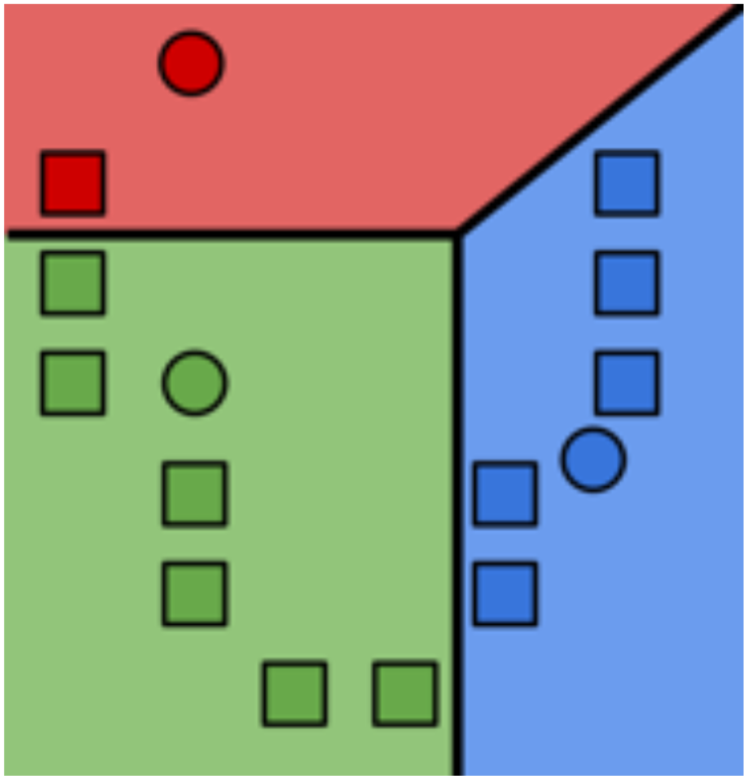
- A common default choice is to choose centroids at random
- Will see later that there are smarter ways of initializing



k-means Clustering

Assign each example to its closest cluster centroid

$$z_i \leftarrow \underset{j \in [k]}{\operatorname{argmin}} \left|\left|\mu_j - x_i\right|\right|^2$$

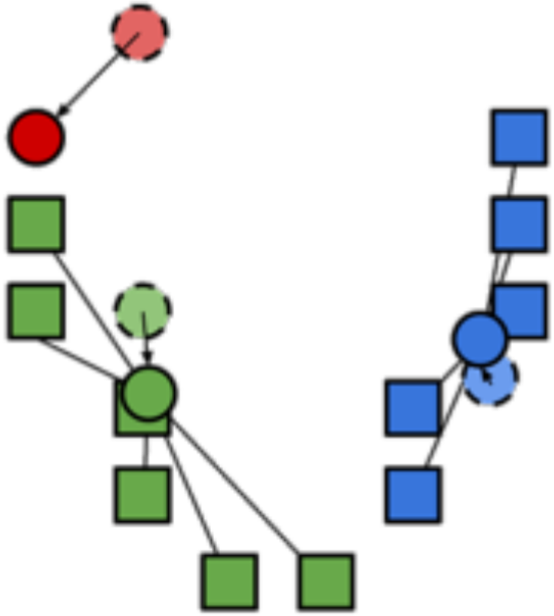


k-means Clustering

Update the centroids to be the mean of all the points assigned to that cluster.

$$\mu_j \leftarrow \frac{1}{n_j} \sum_{i:z_i=j} x_i$$

Computes center of mass for cluster!



k-means Clustering

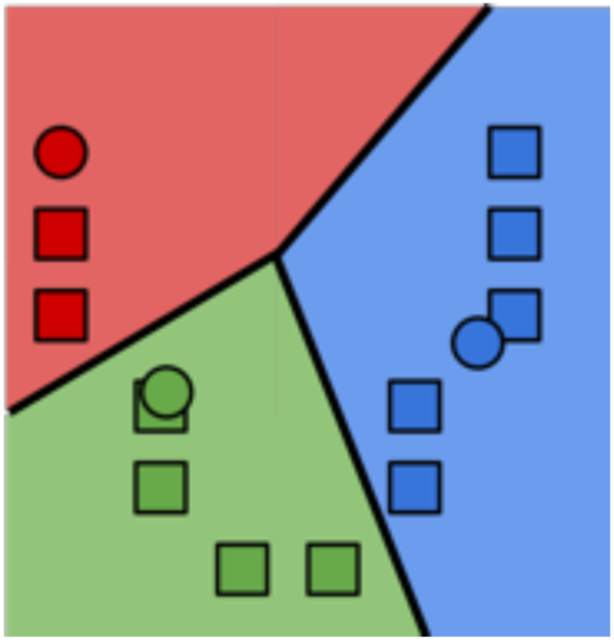
Repeat Steps 1 and 2 until convergence

Will it converge? Yes! Stop when

- Cluster assignments haven't changed
- Some number of max iterations have been passed

What will it converge to?

- Global optimum
- Local optimum
- Neither



Improving kMeans?

kMeans++

Next lecture?