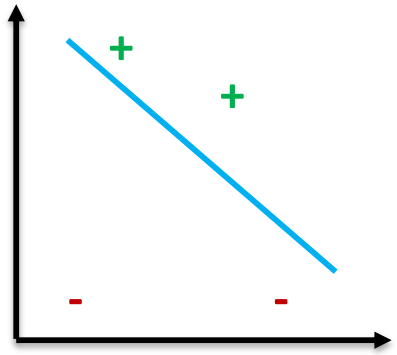
EECS 445

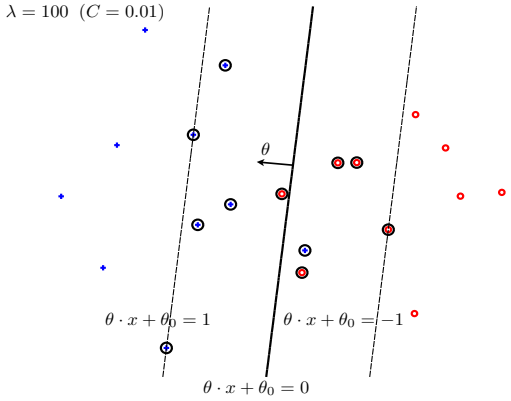
Introduction to Machine Learning

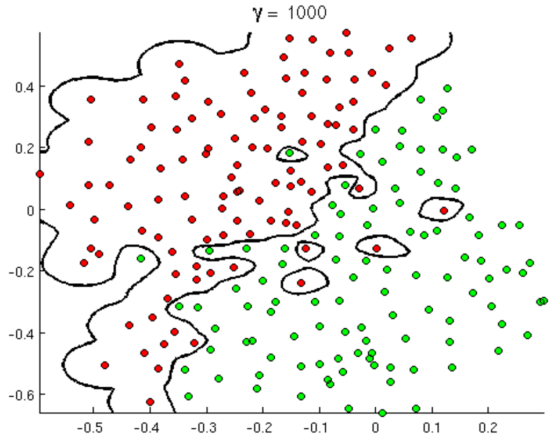
k-Means Clustering

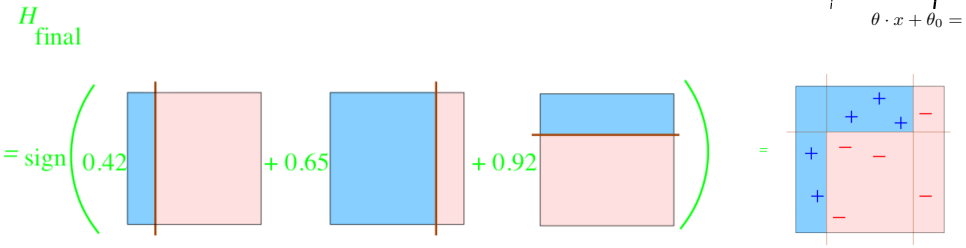
Prof. Kutty

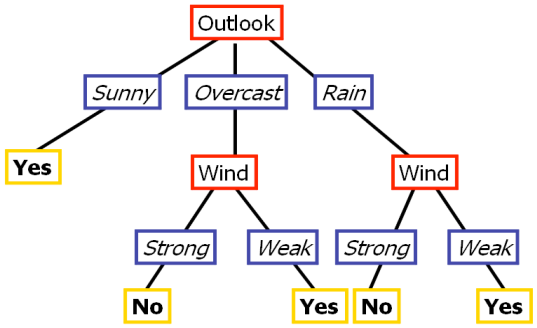
Story so far…

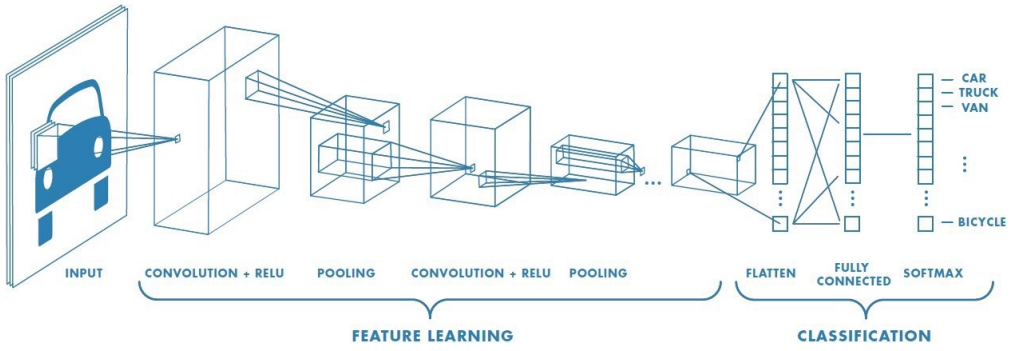

+
+
-
-

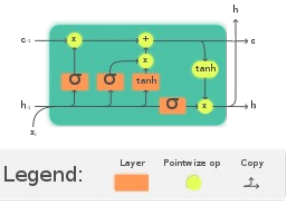

$\lambda = 100 \quad (C = 0.01)$
θ
$\theta \cdot x + \theta_0 = 1$
$\theta \cdot x + \theta_0 = -1$
$\theta \cdot x + \theta_0 = 0$


γ = 1000
0.4
0.2
0
-0.2
-0.4
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2


H_{final}
= sign
0.42
+ 0.65
+ 0.92
=

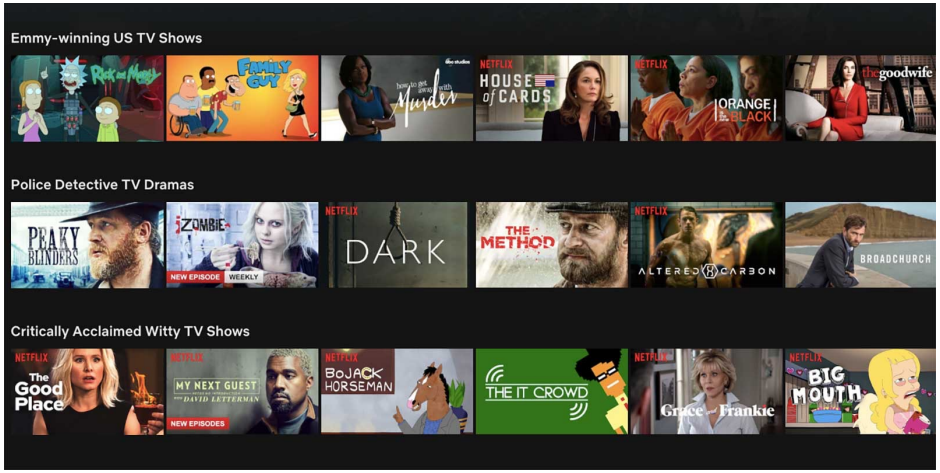

Outlook
Sunny
Overcast
Rain
Yes
Wind
Wind
Strong
Weak
Strong
Weak
No
Yes
No
Yes


CAR
TRUCK
VAN
BICYCLE
INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
FULLY CONNECTED
SOFTMAX
FEATURE LEARNING
CLASSIFICATION

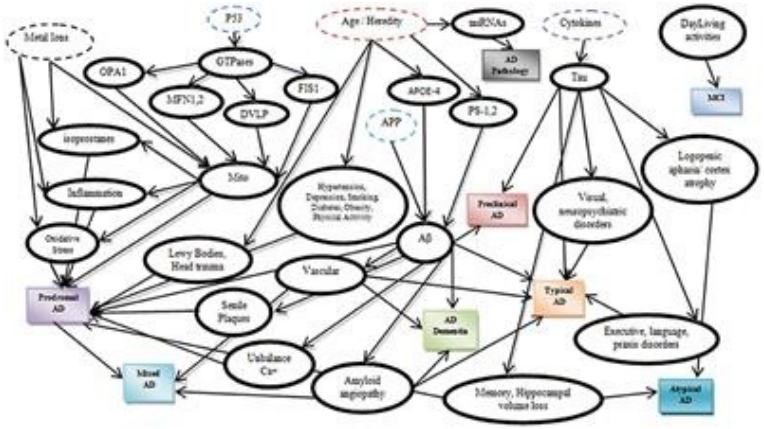

tanh
tanh
Legend:
Layer
Pointwize op
Copy

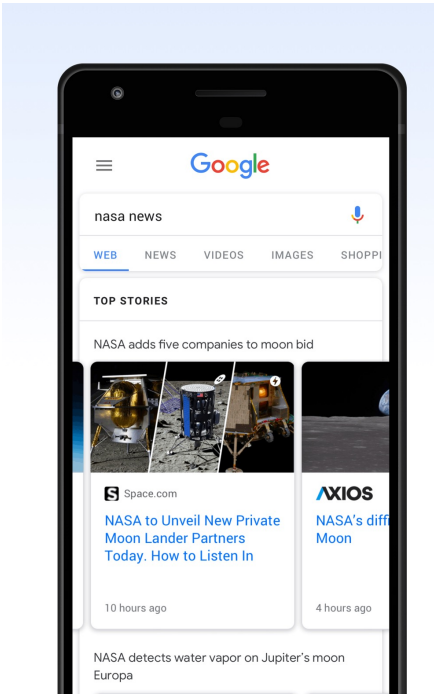
Welcome to Part 3!

Recommender systems



Generative Models





Clustering

Announcements

Proffice hours updated on calendar

IA applications are due *early* April (check email for exact date)

Final exam will be in person:

- *everyone* is expected to take the exam at time set by registrar: 7pm-9pm ET on Th Apr 25, 2024
- students with a *hard* conflict have already reached out to me and Prof. Makar (deadline for notification was in January)
  only one alternate exam time (details TBD but *tentatively* 3pm-5pm)
- students with accommodation will take the exam starting at 6pm

Fall 2024: Computer Graphics and Generative Models

Course Number: EECS 498/598

Credit Hours: 4 credits

Time: Lectures M/W 10:30-12PM; Lab Hours T/TH 3:30-5:30PM

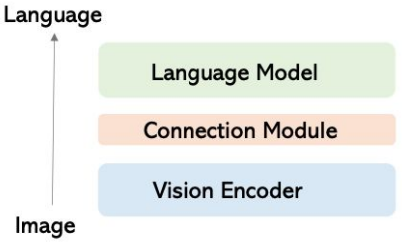
Instructors: Jeong Joon Park

Description: With the impressive recent performance of machine-generated visual content, studying how to create realistic imagery using traditional and AI-based tools is becoming increasingly important. This course will introduce students to the theoretical and practical foundations of computer graphics, as well as the recent advances in generative models to automate the content creation process. This course is designed to prepare both undergraduate and graduate students to learn how visual content can be created and to prepare conducting research in a relevant area.

See more details here

Special Topics Course:

Machine Learning Research Experience



The above images depict how Machine Learning can be used for *style transfer* and for *image to text generation*. But how were the underlying models developed? How did these ideas grow out of state-of-the-art research?

Are you curious about these and other **cutting-edge Machine Learning technologies** and would like to test them out yourself? Are you interested in **research** and are looking for an opportunity to try it? Have you worked in a research lab but are looking for further autonomy and the ability to **propose new ideas**? If so, this course is for you!

The goal of this course is to take students through the broad strokes of doing research, with both *autonomy* and *sufficient scaffolding*, to make this exploration fun while expanding your understanding of the research process.

*can count as MDE/Capstone for CS/DS/CE majors

Here is the course flyer

*un*supervised learning

https://forms.gle/ffiBvNbPjHF8ghi77

Supervised Learning

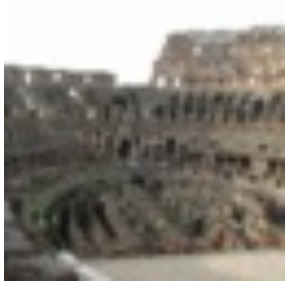
pantheon

colosseum

hofburg_imperial_palace

$$S_n = \{\bar{x}^{(i)}, y^{(i)}\}_{i=1}^{n}$$

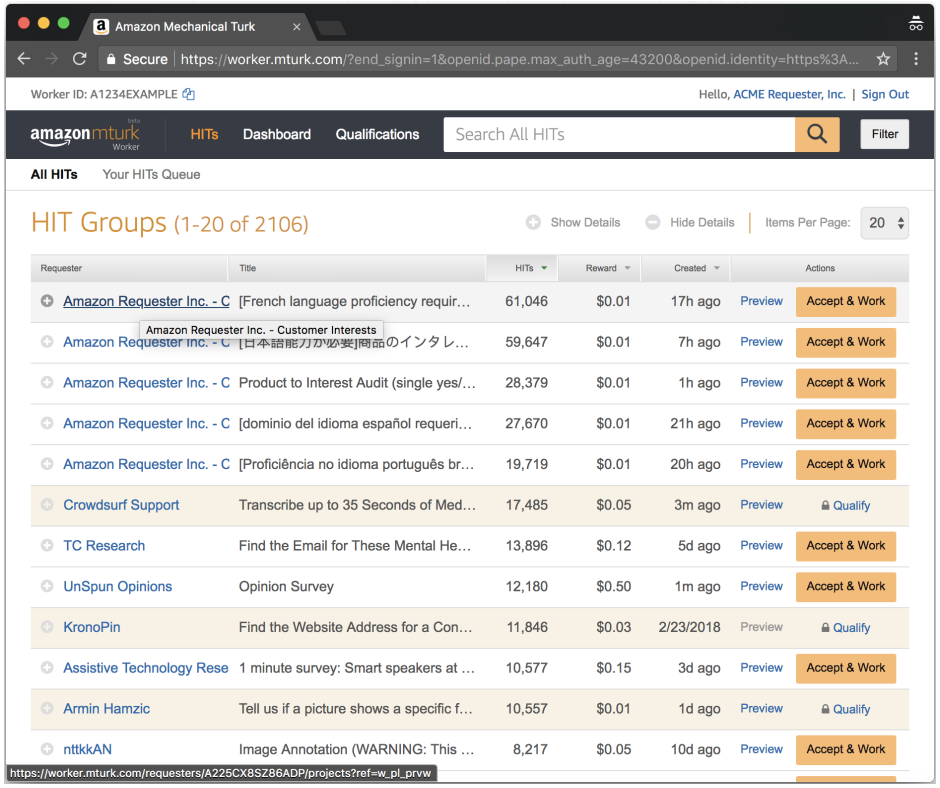
$$\bar{x} \in \mathcal{X} \quad y \in \mathcal{Y}$$



?

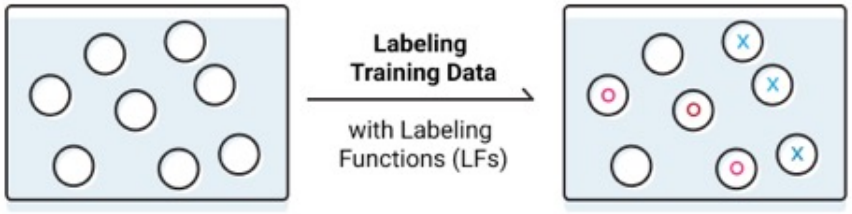
Goal: learn a mapping from $\mathcal{X} \rightarrow \mathcal{Y}$ that generalizes “well” to as yet unseen data.

Where do labels come from



MTurk (crowdsourcing)

Domain experts



other sources…

generate automatically
e.g., snorkel.ai

Caveat: labels can be noisy…

ImageNet given label:
feather boa

Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks
Curtis G. Northcutt, Anish Athalye, Jonas Mueller
NeurIPS 2021

Table 1: Test set errors are prominent across common benchmark datasets. Errors are estimated using confident learning (CL) and validated by human workers on Mechanical Turk.

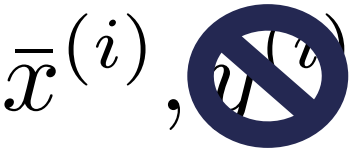
Dataset	Modality	Size	Model	Test Set Errors				
				CL guessed	MTurk checked	validated	estimated	% error
MNIST	image	10,000	2-conv CNN	100	100 (100%)	15	-	0.15
CIFAR-10	image	10,000	VGG	275	275 (100%)	54	-	0.54
CIFAR-100	image	10,000	VGG	2235	2235 (100%)	585	-	5.85
Caltech-256	image	30,607	ResNet-152	4,643	400 (8.6%)	65	754	2.46
ImageNet*	image	50,000	ResNet-50	5,440	5,440 (100%)	2,916	-	5.83
QuickDraw	image	50,426,266	VGG	6,825,383	2,500 (0.04%)	1870	5,105,386	10.12
20news	text	7,532	TFIDF + SGD	93	93 (100%)	82	-	1.11
IMDB	text	25,000	FastText	1,310	1,310 (100%)	725	-	2.9
Amazon	text	9,996,437	FastText	533,249	1,000 (0.2%)	732	390,338	3.9
AudioSet	audio	20,371	VGG	307	307 (100%)	275	-	1.35

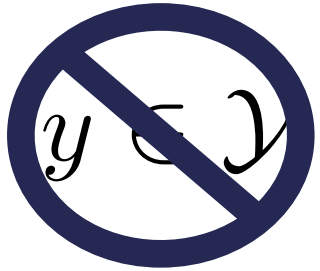
*Because the ImageNet test set labels are not publicly available, the ILSVRC 2012 validation set is used.

Unsupervised Learning

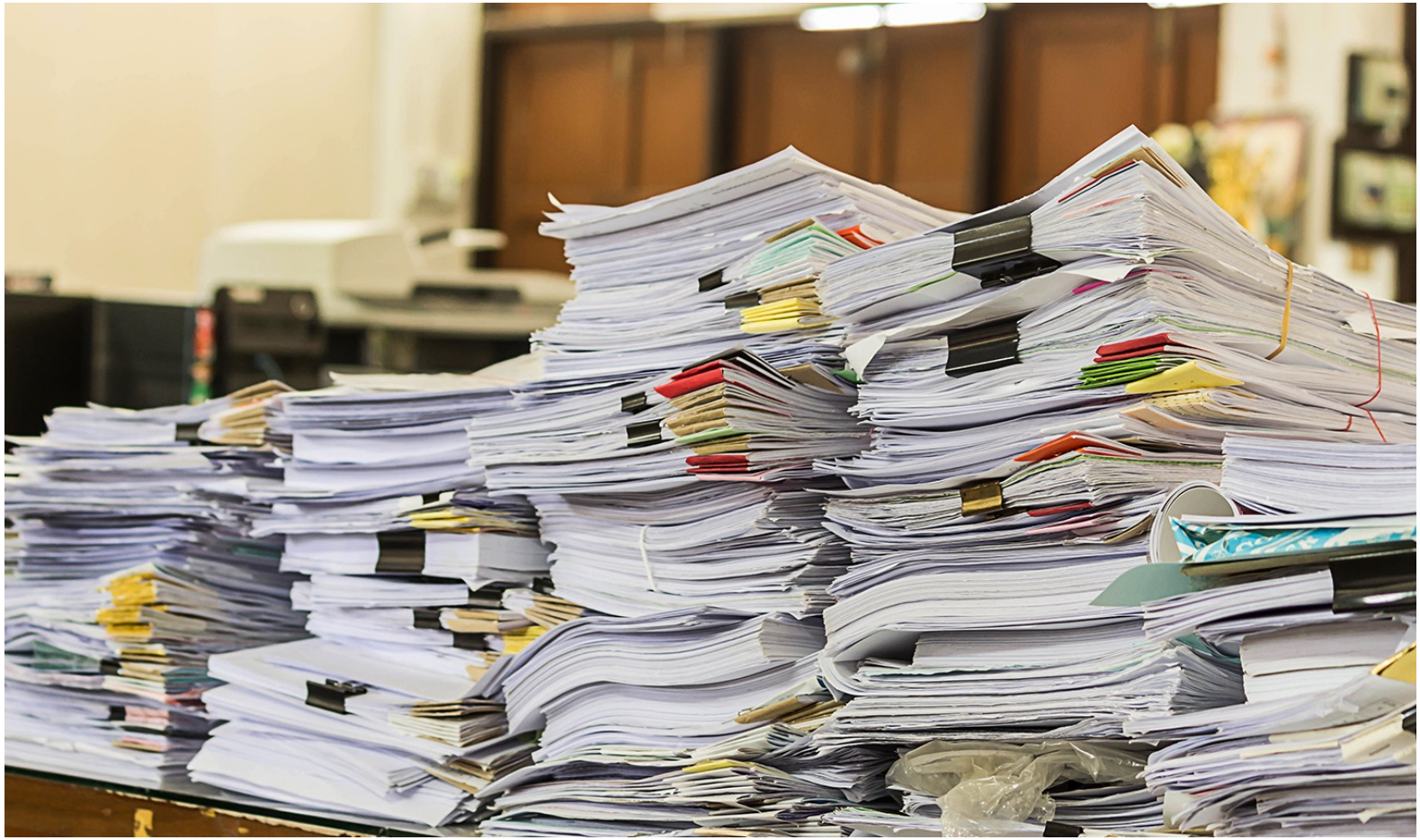
Sometimes labels are not available

$$S_n = \{\bar{x}^{(i)}, y^{(i)}\}_{i=1}^{n} \qquad \bar{x} \in \mathcal{X} \qquad y \in \mathcal{Y}$$



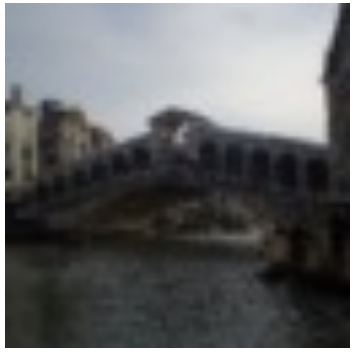


Machine Learning @ Froogle

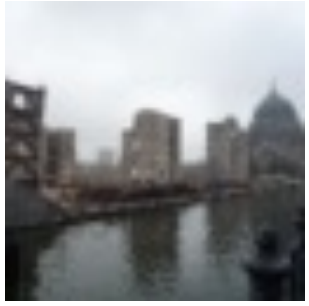


Unlabeled DataSets



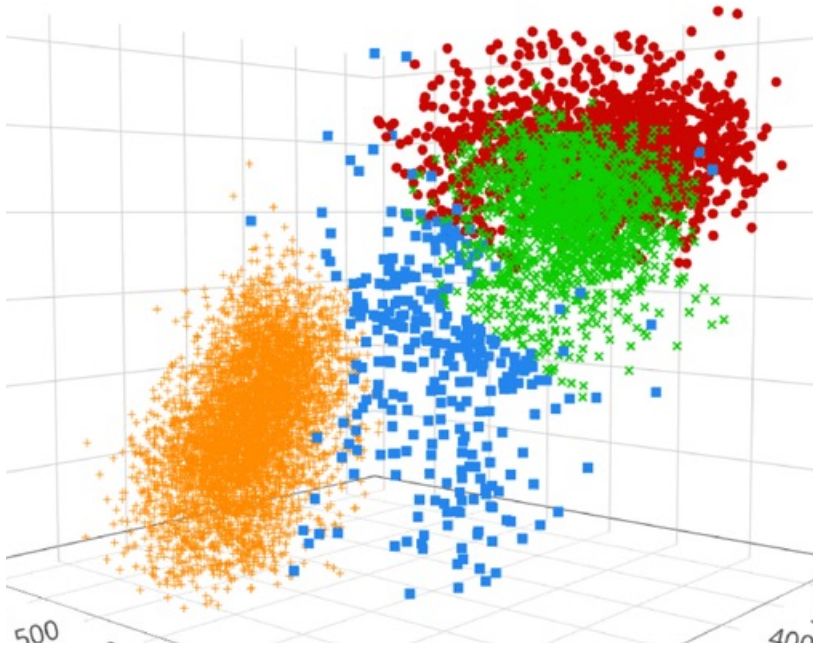






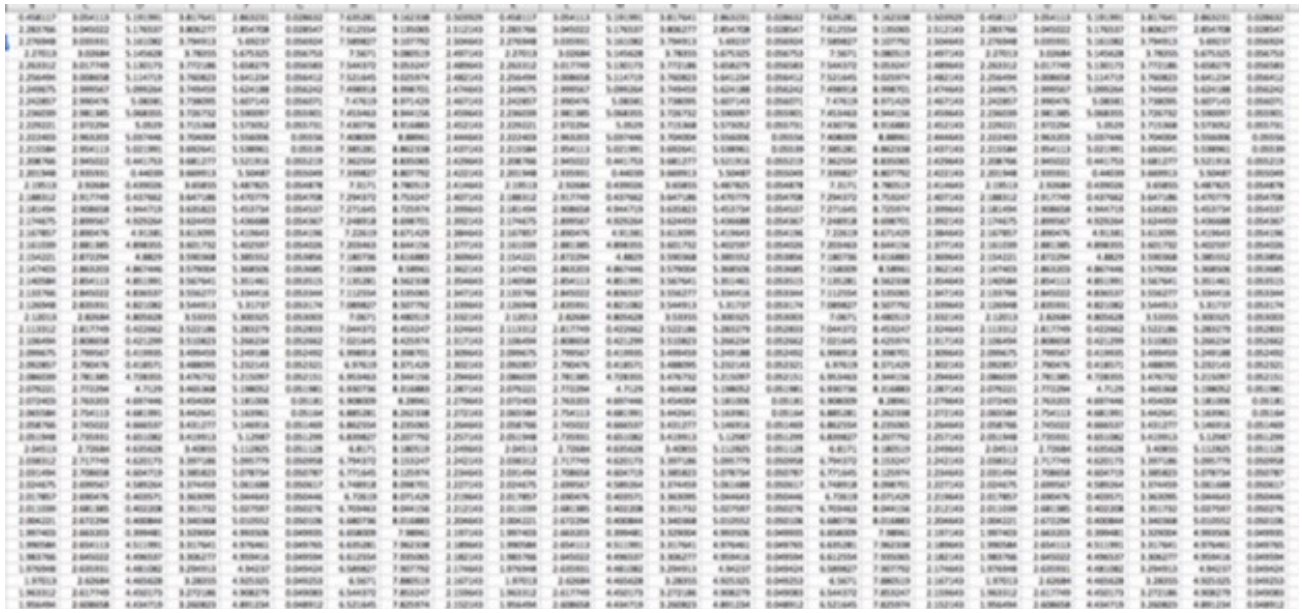
What can we do with unlabeled data?

	IL-8	4-1BB	TIE2
OLINK_STLOUIS_PATIENT_088	5.9349	8.68042	8.3547
OLINK_STLOUIS_PATIENT_089	6.58607	9.91896	8.07328
OLINK_STLOUIS_PATIENT_090	7.22252	5.95889	8.11313
OLINK_STLOUIS_PATIENT_091	5.90759	6.68818	8.57093
OLINK_STLOUIS_PATIENT_092	8.17976	12.31368	8.76246
OLINK_STLOUIS_PATIENT_093	7.01706	8.93585	8.96573
OLINK_STLOUIS_PATIENT_094	7.13768	8.1513	8.66963
OLINK_STLOUIS_PATIENT_095	5.87923	7.5273	8.32331
OLINK_STLOUIS_PATIENT_096	7.65582	7.92265	8.50901
OLINK_STLOUIS_PATIENT_097	8.29086	6.8465	8.49276
OLINK_STLOUIS_PATIENT_098	6.01398	6.47274	7.94174
OLINK_STLOUIS_PATIENT_099	7.86462	12.18905	9.2117



Visualize it

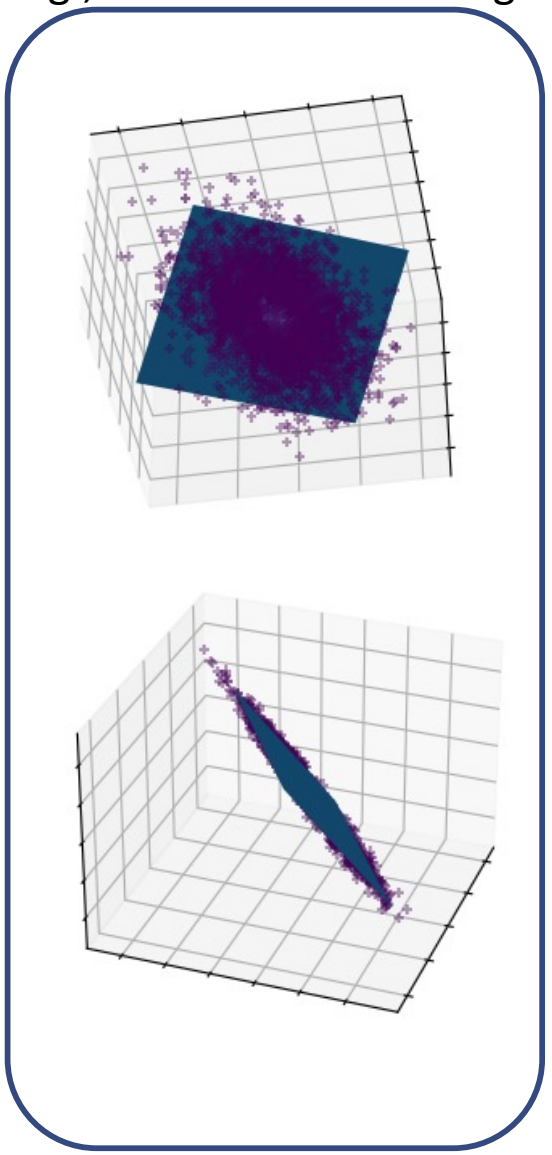
What can we do with high-dim unlabeled data?



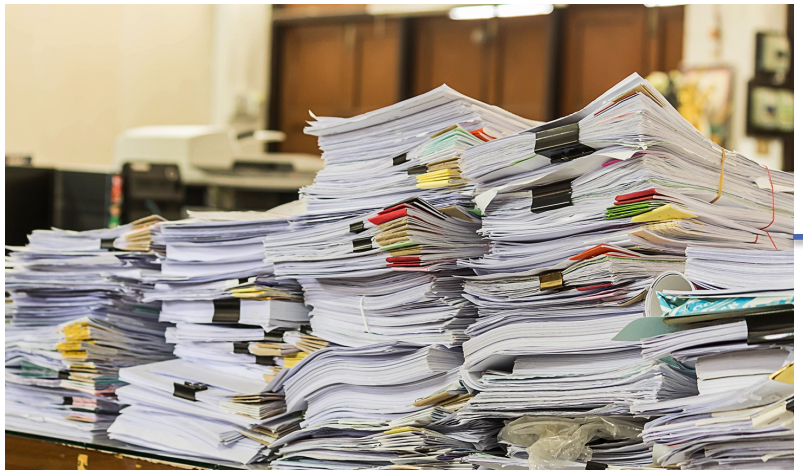
~~Visualize it~~

find a low dimensional embedding
(then maybe visualize)

e.g., 3D to 2D embedding



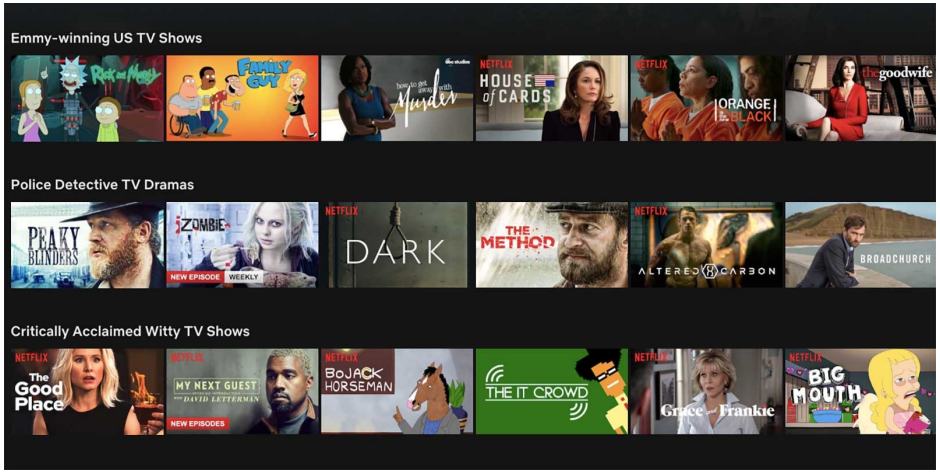
Machine Learning @ Froogle



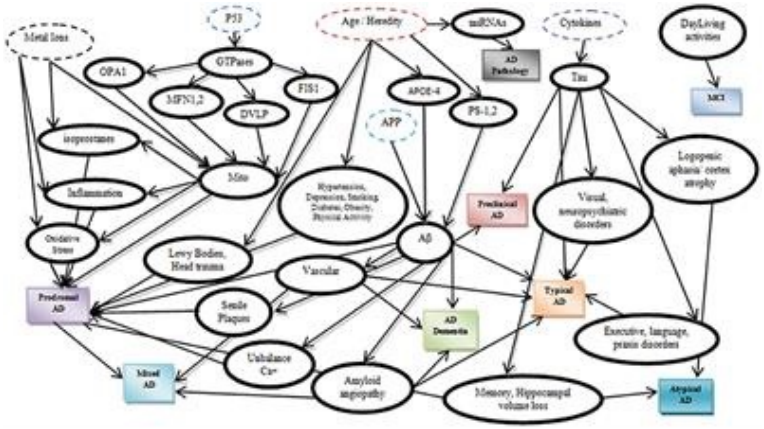
unsupervised learning

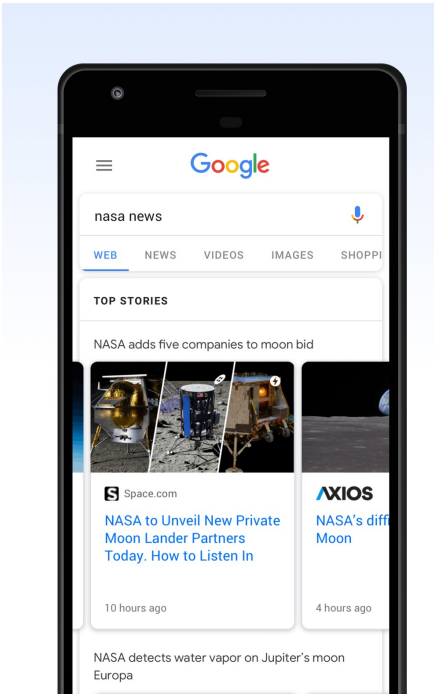
Welcome to unsupervised learning!

Recommender systems



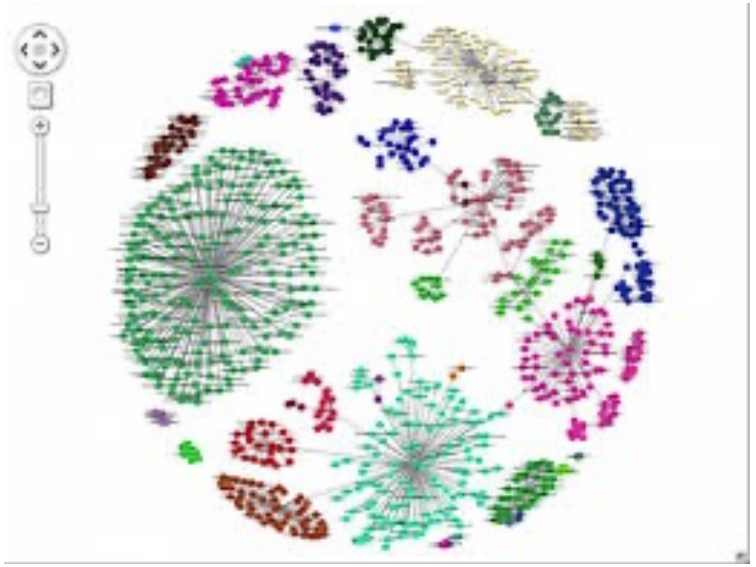
Generative Models





Clustering

Applications

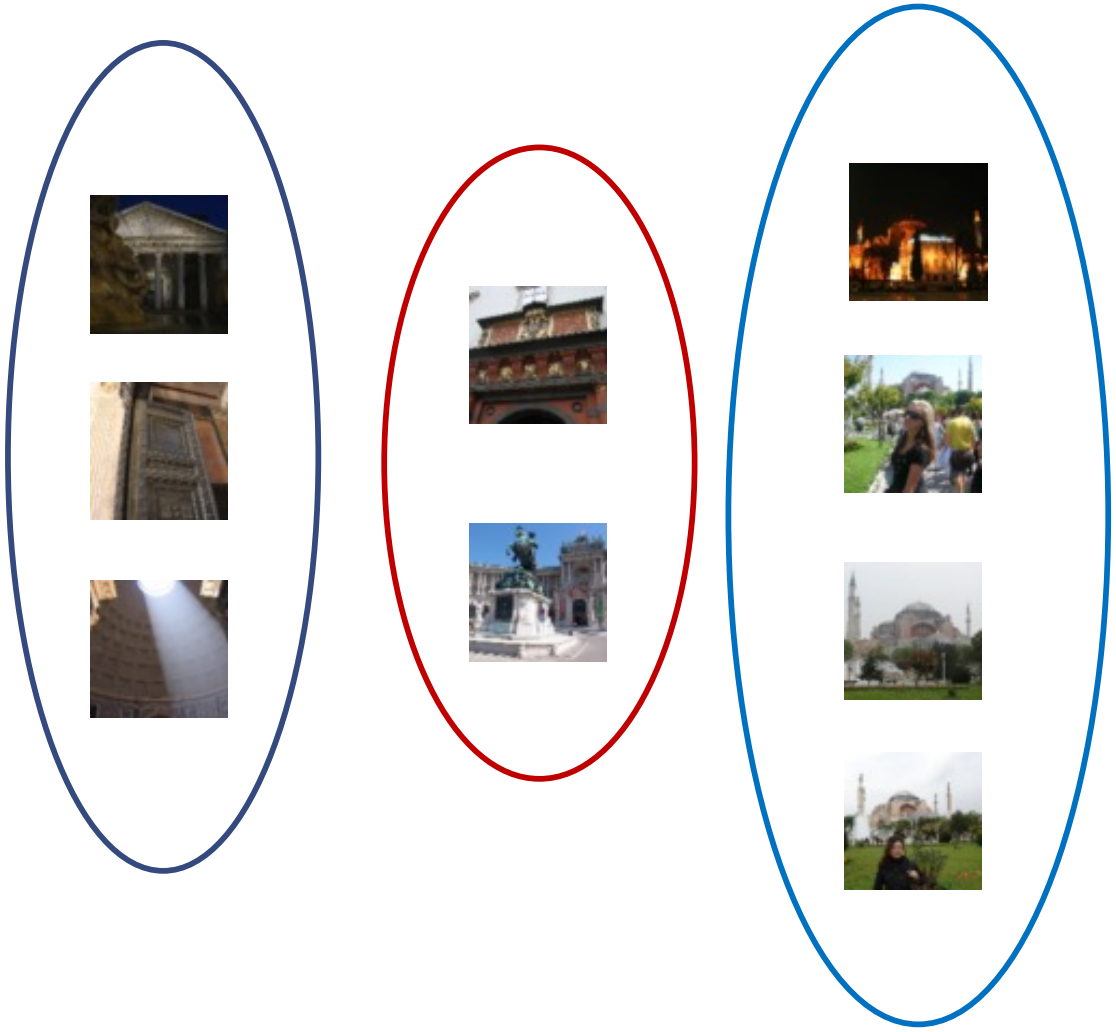


”Now, when there are multiple stories related to your search, we’ll also organize the results by story so it’s easier to unders what’s most relevant a you can make a more informed decision on specific articles to exp

Other Examples:

- finding similar homes for sale
- grouping patients by symptoms
- mining customer purchase patterns
- grouping search results according to topic
- group emails
- image processing → regions of image segmentation…

Unsupervised DataSets



Goal: Learn a representation of the data, uncover useful structure, identify groups/clusters of similar examples

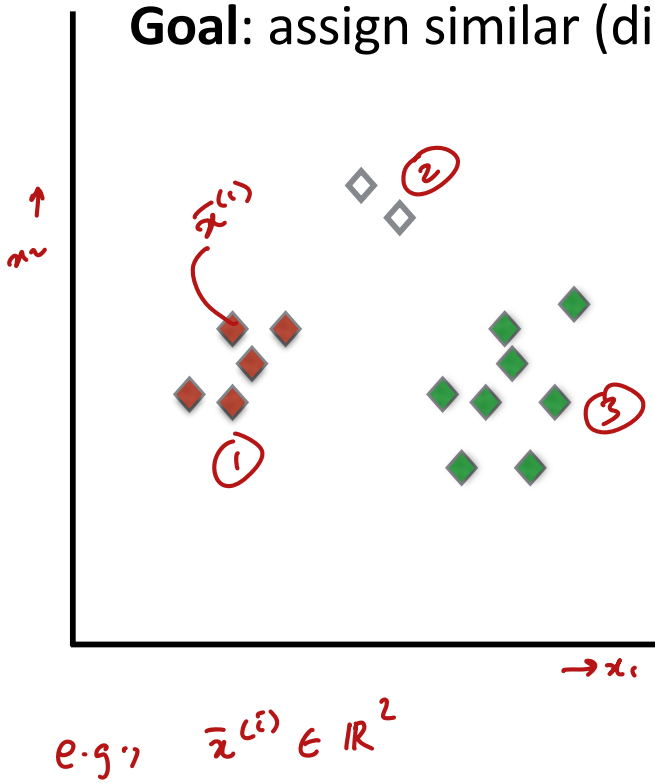
Clustering

Input: $S_n = \{\bar{x}^{(i)}\}_{i=1}^{n}$ $\bar{x}^{(i)} \in \mathbb{R}^d$ k → # clusters

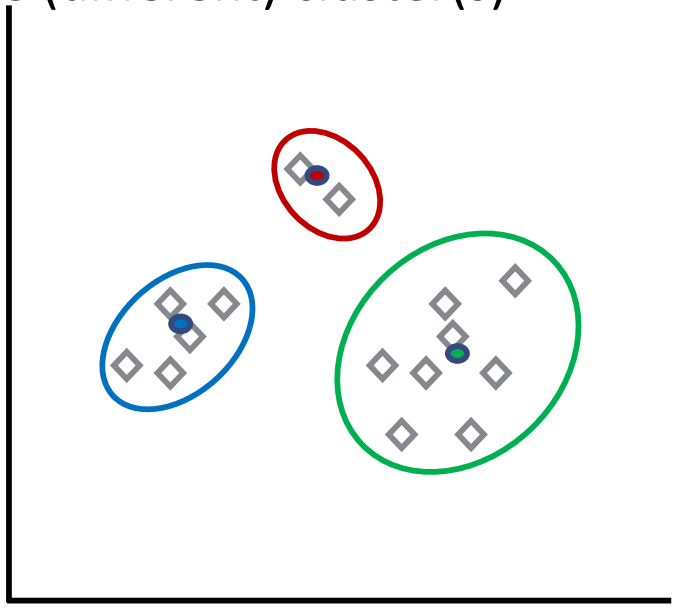
Output: a set of cluster assignments $c_1, \ldots, c_n$ → c_i → cluster assignment for datapoint i i.e., $\bar{x}^{(i)}$

with each $c_i \in \{1, \ldots, k\}$

Goal: assign similar (dissimilar) points to the same (different) cluster(s)



e.g., $c_i = 1$



e.g., $\bar{x}^{(i)} \in \mathbb{R}^2$

Clusters can be described by
their corresponding set of examples or
a representative point

*un*supervised learning

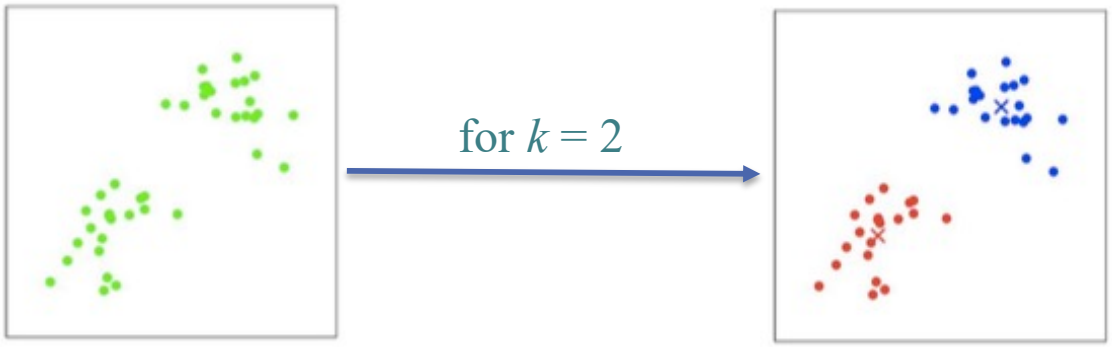
clustering

k-means

k-means Clustering

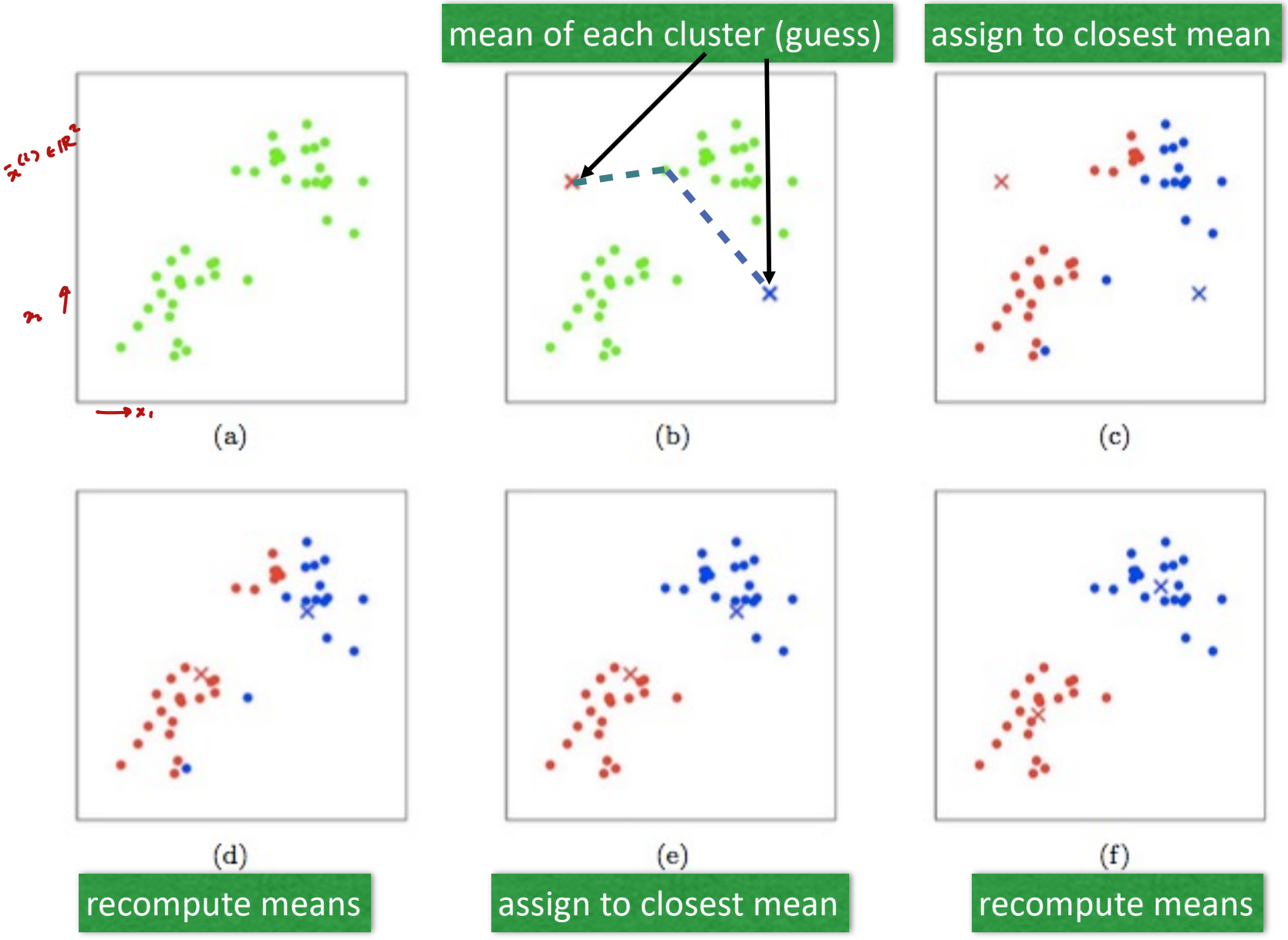
goal

Partition data into k clusters (defined by k "means") such that the sum of squares of Euclidean distances of each point to its cluster's mean is minimized



k-means Clustering

k is a hyperparameter; k = 2



k-means objective function

Datapoints $\bar{x}^{(1)}, \ldots, \bar{x}^{(n)}$ $\in \mathbb{R}^d$

clusters corresponding to each datapoint

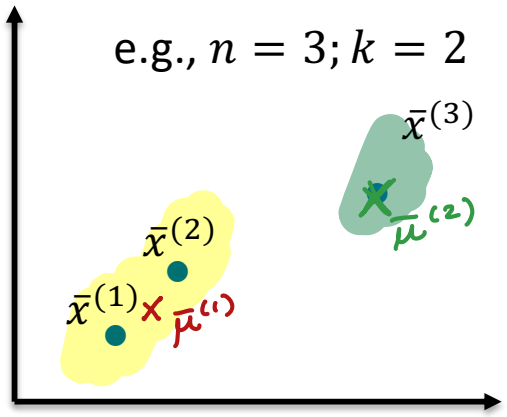
Find → $c_1, \ldots, c_n$ with each $c_i \in \{1, \ldots, k\}$

means corresponding to each cluster

Find → $\bar{\mu}^{(1)}, \ldots, \bar{\mu}^{(k)}$ $\in \mathbb{R}^d$

Goal is to minimize the objective function

$$\sum_{i=1}^{n} \left\| \bar{x}^{(i)} - \bar{\mu}^{(c_i)} \right\|^2$$



$\bar{\mu}^{(1)} = \dfrac{\bar{x}^{(1)} + \bar{x}^{(2)}}{2}$

$\bar{\mu}^{(2)} = \bar{x}^{(3)}$

objective function

$$\left\| \bar{x}^{(1)} - \bar{\mu}^{(c_1)} \right\|^2 + \left\| \bar{x}^{(2)} - \bar{\mu}^{(c_2)} \right\|^2 + \left\| \bar{x}^{(3)} - \bar{\mu}^{(c_3)} \right\|^2$$

$$= \left\| \bar{x}^{(1)} - \bar{\mu}^{(1)} \right\|^2 + \left\| \bar{x}^{(2)} - \bar{\mu}^{(1)} \right\|^2 + \left\| \bar{x}^{(3)} - \bar{\mu}^{(2)} \right\|^2$$

$c_1 = 1$

$c_2 = 1$

$c_3 = 2$

k-means Clustering

algorithm: more formally

Datapoints $\bar{x}^{(1)}, \ldots, \bar{x}^{(n)}$ and fixed k

initialize means $\bar{\mu}^{(1)}, \ldots, \bar{\mu}^{(k)}$

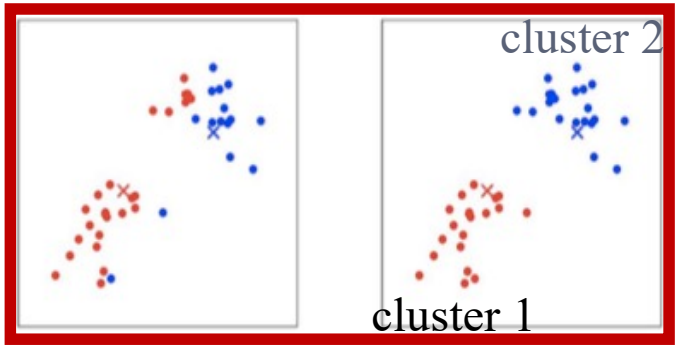
Iteratively

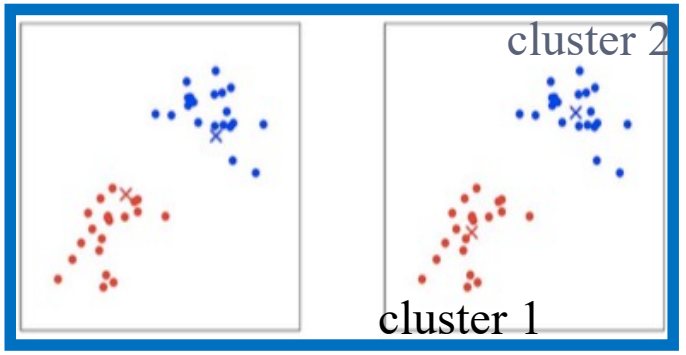
- for each point $\bar{x}^{(i)}$, reassign $\bar{x}^{(i)}$ to

$$c_i = \arg\min_j \left\| \bar{x}^{(i)} - \bar{\mu}^{(j)} \right\|^2$$

- recompute $\bar{\mu}^{(j)} = \frac{\sum_i [\![c_i = j]\!] \bar{x}^{(i)}}{\sum_i [\![c_i = j]\!]}$

indicator $= \begin{cases} 1 & c_i = j \\ 0 & \text{o.w.} \end{cases}$





k-means algorithm

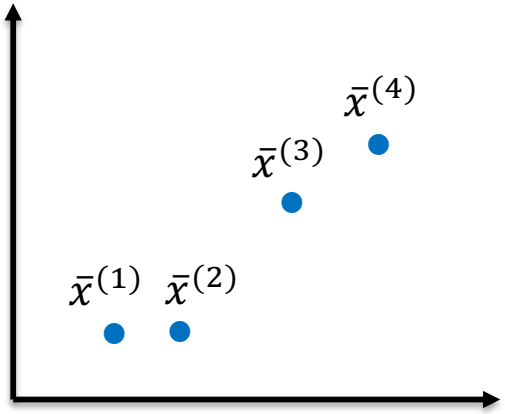
in class exercise: $k = 2; n = 4$

initialize means $\bar{\mu}^{(1)} = \bar{x}^{(1)}, \bar{\mu}^{(2)} = \bar{x}^{(2)}$

Iteratively

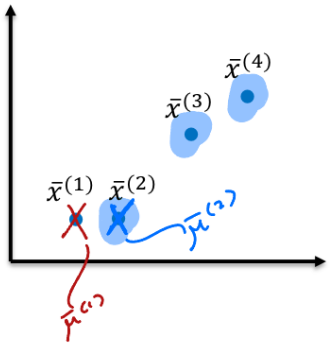
- $\forall i$ reassign $\bar{x}^{(i)}$ to $c_i = \arg\min_j \left\| \bar{x}^{(i)} - \bar{\mu}^{(j)} \right\|^2$
- $\forall j$ recompute $\bar{\mu}^{(j)} = \frac{\sum_i [\![c_i = j]\!] \bar{x}^{(i)}}{\sum_i [\![c_i = j]\!]}$

$$\bar{x}^{(1)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$
$$\bar{x}^{(2)} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$
$$\bar{x}^{(3)} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$
$$\bar{x}^{(4)} = \begin{bmatrix} 5 \\ 4 \end{bmatrix}$$



https://forms.gle/ffiBvNbPjHF8ghi77

Iteration 1:

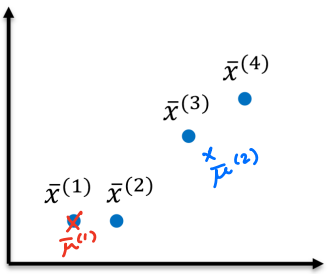


(i) $c_1 = 1$

$c_2 = 2$

$c_3 = 2$

$c_4 = 2$



(ii) $\bar{\mu}^{(1)} = \bar{x}^{(1)}$

$\bar{\mu}^{(2)} = \dfrac{\bar{x}^{(2)} + \bar{x}^{(3)} + \bar{x}^{(4)}}{3}$

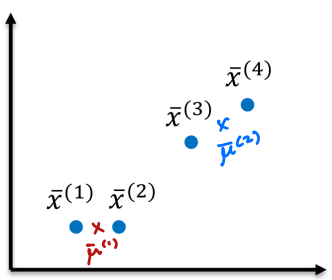
Iteration 2:

(i) $c_1 = 1$

$c_2 = 1$

$c_3 = 2$

$c_4 = 2$



(ii) $\bar{\mu}^{(1)} = \dfrac{\bar{x}^{(1)} + \bar{x}^{(2)}}{2}$

$\bar{\mu}^{(2)} = \dfrac{\bar{x}^{(3)} + \bar{x}^{(4)}}{2}$

~~Iteration 3~~

cluster assignment unchanged

converged!

k-means Clustering

algorithm: details

Datapoints $\bar{x}^{(1)}, \ldots, \bar{x}^{(n)}$ and fixed k

initialize means $\bar{\mu}^{(1)}, \ldots, \bar{\mu}^{(k)}$

until when?

Iteratively

- reassign $\bar{x}^{(i)}$ to $c_i = \arg\min_j \left\| \bar{x}^{(i)} - \bar{\mu}^{(j)} \right\|^2$
- recompute $\bar{\mu}^{(j)} = \frac{\sum_i [\![c_i = j]\!] \bar{x}^{(i)}}{\sum_i [\![c_i = j]\!]}$

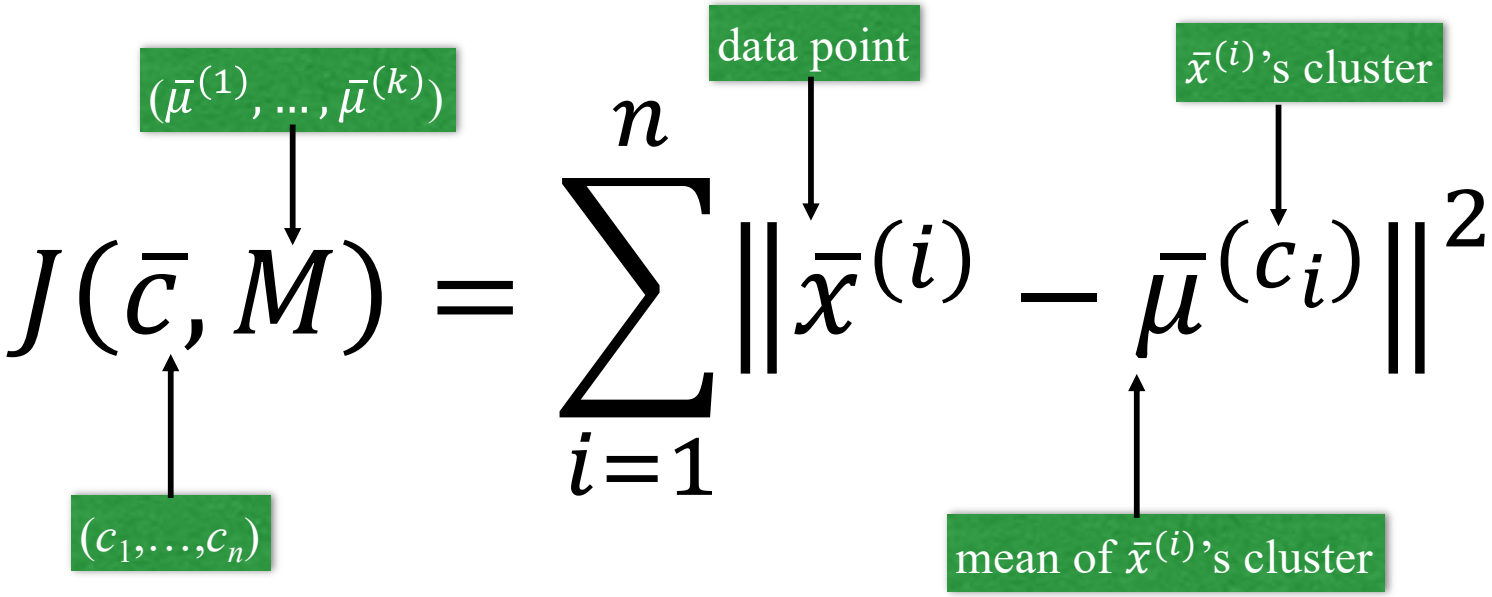
Claim

k-means is guaranteed to converge*

k-means Clustering

convergence

objective function



Claim 1

k-means performs **coordinate descent** on the objective function

k-means Clustering

performs coordinate descent on the objective function

Datapoints $\bar{x}^{(1)}, \ldots, \bar{x}^{(n)}$ and fixed k

initialize means $\bar{\mu}^{(1)}, \ldots, \bar{\mu}^{(k)}$

Iteratively

- reassign $\bar{x}^{(i)}$ to $c_i = \arg\min_j \left\| \bar{x}^{(i)} - \bar{\mu}^{(j)} \right\|^2$
- recompute $\bar{\mu}^{(j)} = \frac{\sum_i [\![c_i = j]\!] \bar{x}^{(i)}}{\sum_i [\![c_i = j]\!]}$

fix M, choose $\bar{c}$ to minimize

$$J(\bar{c}, M) = \sum_{i=1}^{n} \left\| \bar{x}^{(i)} - \bar{\mu}^{(c_i)} \right\|^2$$

fix $\bar{c}$, choose M to minimize

$$J(\bar{c}, M) = \sum_{i=1}^{n} \left\| \bar{x}^{(i)} - \bar{\mu}^{(c_i)} \right\|^2$$

k-means Clustering

convergence

Claim 1

k-means performs **coordinate descent** on the objective function

Claim 2

k-means is guaranteed to converge*

k-means Clustering

Good news…

Iteratively

- reassign $\bar{x}^{(i)}$ to $c_i = \arg\min_j \left\|\bar{x}^{(i)} - \bar{\mu}^{(j)}\right\|^2$
- recompute $\bar{\mu}^{(j)} = \frac{\sum_i [\![c_i = j]\!] \bar{x}^{(i)}}{\sum_i [\![c_i = j]\!]}$

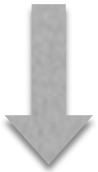
fix $\bar{\mu}$, choose $\bar{c}$ to minimize

$$J(\bar{c}, M) = \sum_{i=1}^{n} \left\|\bar{x}^{(i)} - \bar{\mu}^{(c_i)}\right\|^2$$

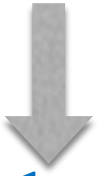
fix $\bar{c}$, choose $\bar{\mu}$ to minimize

$$J(\bar{c}, M) = \sum_{i=1}^{n} \left\|\bar{x}^{(i)} - \bar{\mu}^{(c_i)}\right\|^2$$

J must monotonically decrease



J must (eventually) converge



k-means is guaranteed to converge (but…)

k-means Clustering

Bad news…

J is not convex

not guaranteed to converge to global minimum

k-means Clustering

algorithm: details

Datapoints $\bar{x}^{(1)}, \ldots, \bar{x}^{(n)}$ and fixed k

initialize means $\bar{\mu}^{(1)}, \ldots, \bar{\mu}^{(k)}$

Iteratively

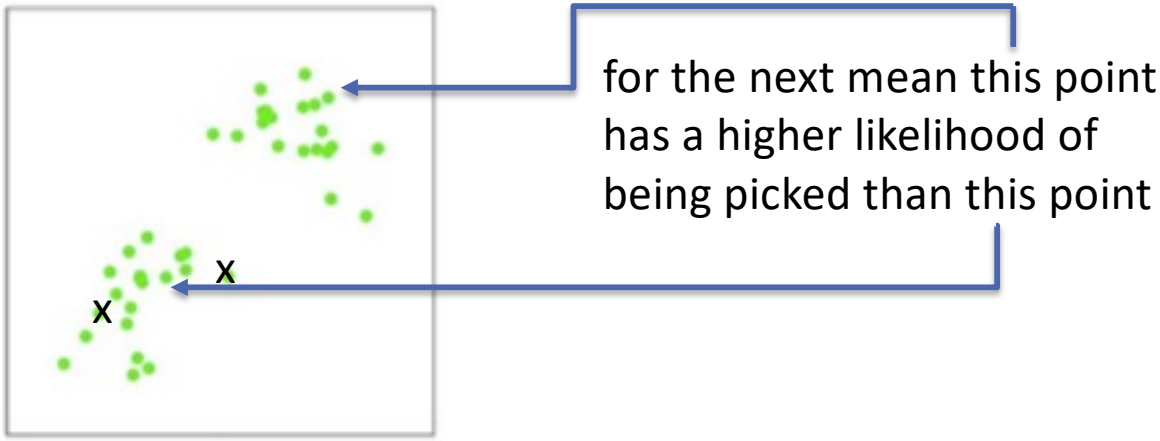
- reassign $\bar{x}^{(i)}$ to $c_i = \arg\min_j \left\| \bar{x}^{(i)} - \bar{\mu}^{(j)} \right\|^2$
- recompute $\bar{\mu}^{(j)} = \frac{\sum_i [\![c_i = j]\!] \bar{x}^{(i)}}{\sum_i [\![c_i = j]\!]}$

How to pick means and find global optimum efficiently

- Option 1: Solve an NP-hard problem!
- Option 2: randomly pick amongst given data (or random points in space) → works sometimes
- Option 3: k-means++ → works often

How to pick means and find global optimum efficiently

- **How to initialize means (k-means++)**
  - Pick points w.p. proportional to distance from already selected means



- *k*-means++ Approximation Guarantee. *The expected value of the objective returned by K-means++ is never more than* O(log *K*) *from optimal and can be as close as* O(1) *from optimal. Even in the former case, with 2K random restarts, one restart will be* O(1) *from optimal (with high probability).*

k-means Clustering

algorithm: details

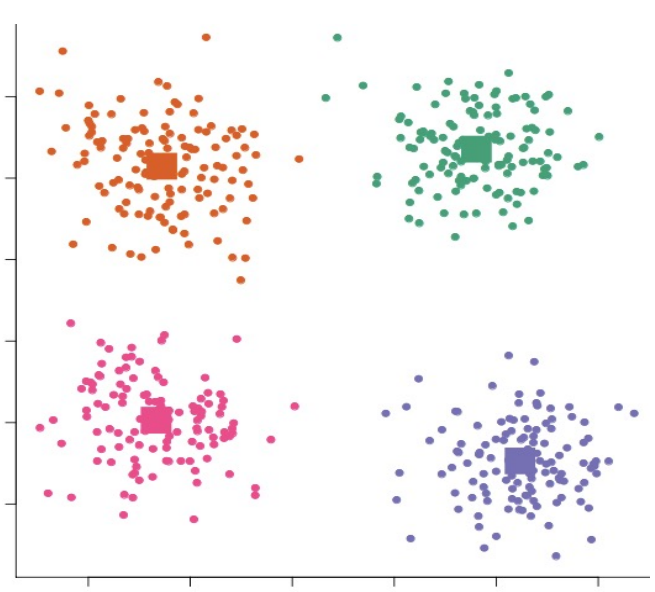
Datapoints $\bar{x}^{(1)}, \ldots, \bar{x}^{(n)}$ and fixed k

initialize means $\bar{\mu}^{(1)}, \ldots, \bar{\mu}^{(k)}$

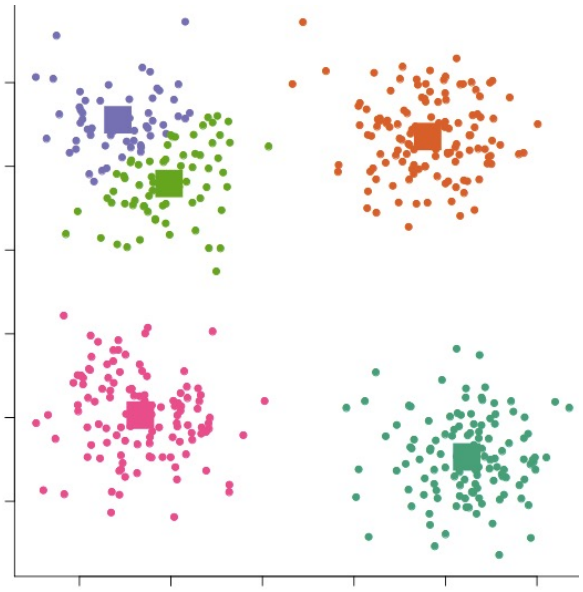
Iteratively

- reassign $\bar{x}^{(i)}$ to $c_i = \arg\min_j \left\| \bar{x}^{(i)} - \bar{\mu}^{(j)} \right\|^2$
- recompute $\bar{\mu}^{(j)} = \frac{\sum_i [\![c_i = j]\!] \bar{x}^{(i)}}{\sum_i [\![c_i = j]\!]}$

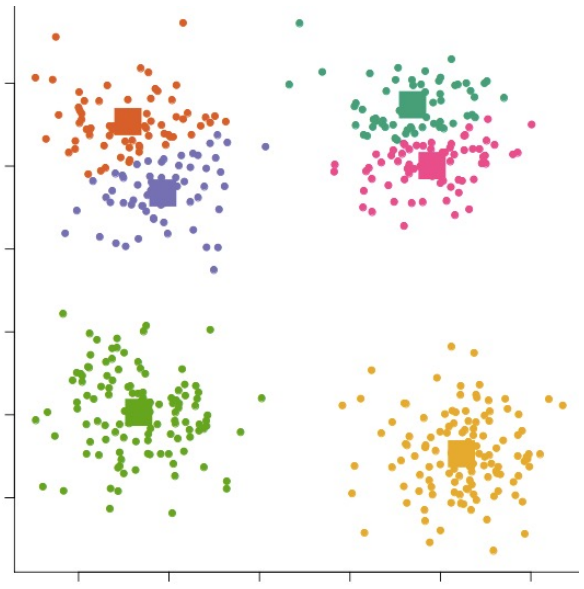
Q: How do we pick k?



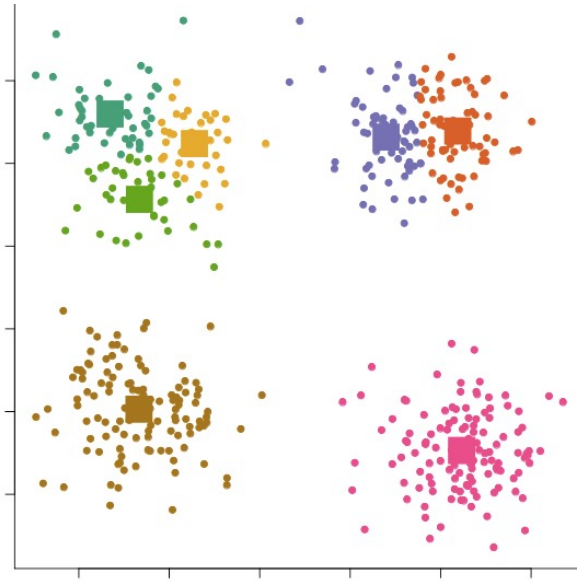
k=4



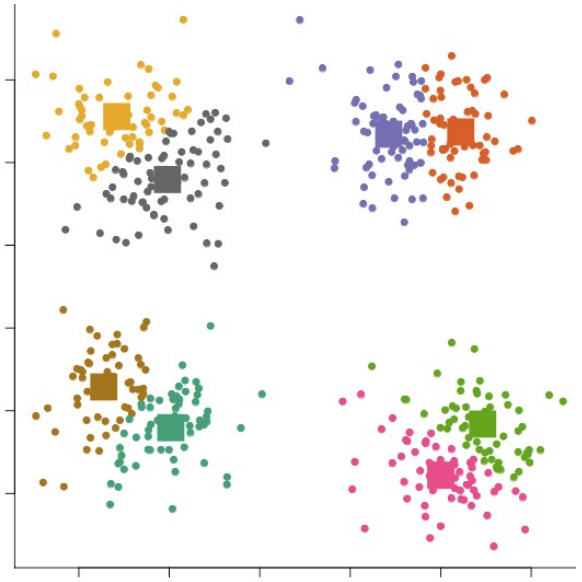
k=5



k=6



k=7



k=8

Elbow method