Introduction to Machine Learning Part 1 and Part 2

Yingyu Liang

yliang@cs.wisc.edu

Computer Sciences Department

University of Wisconsin, Madison

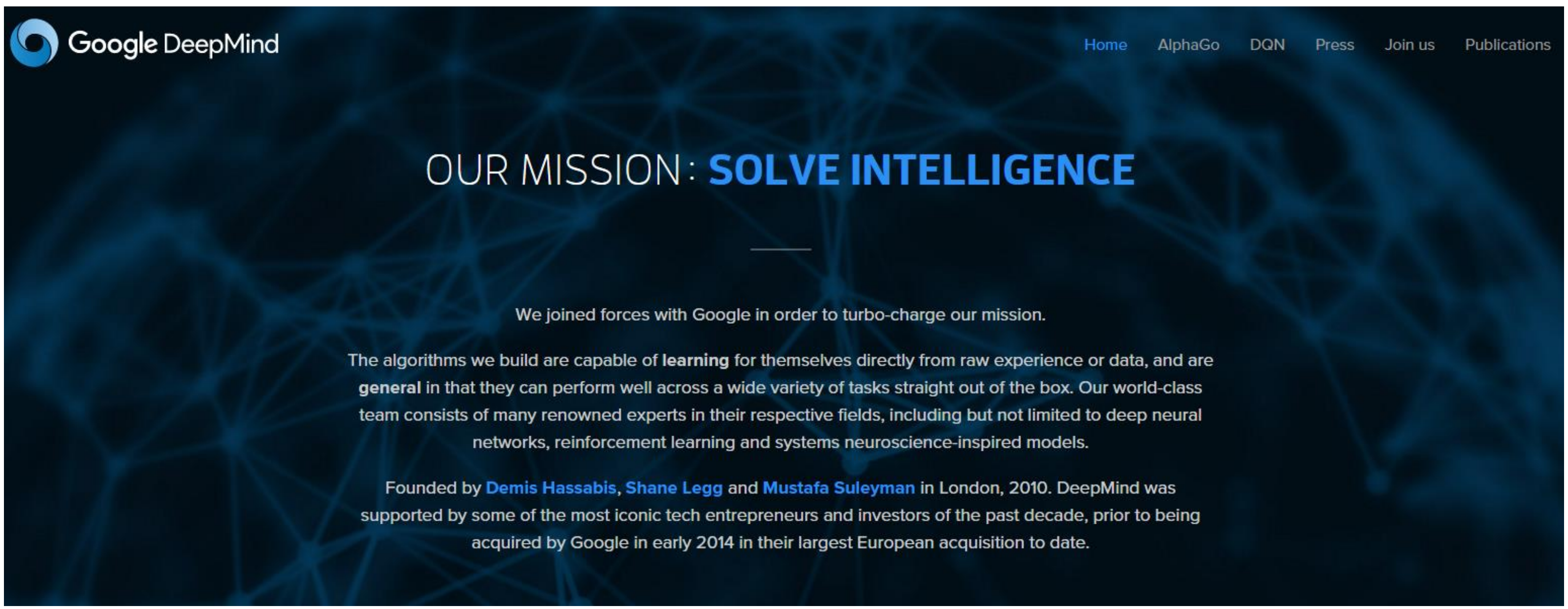
[Partially Based on slides from Jerry Zhu and Mark Craven]

What is machine learning?

- Short answer: recent buzz word

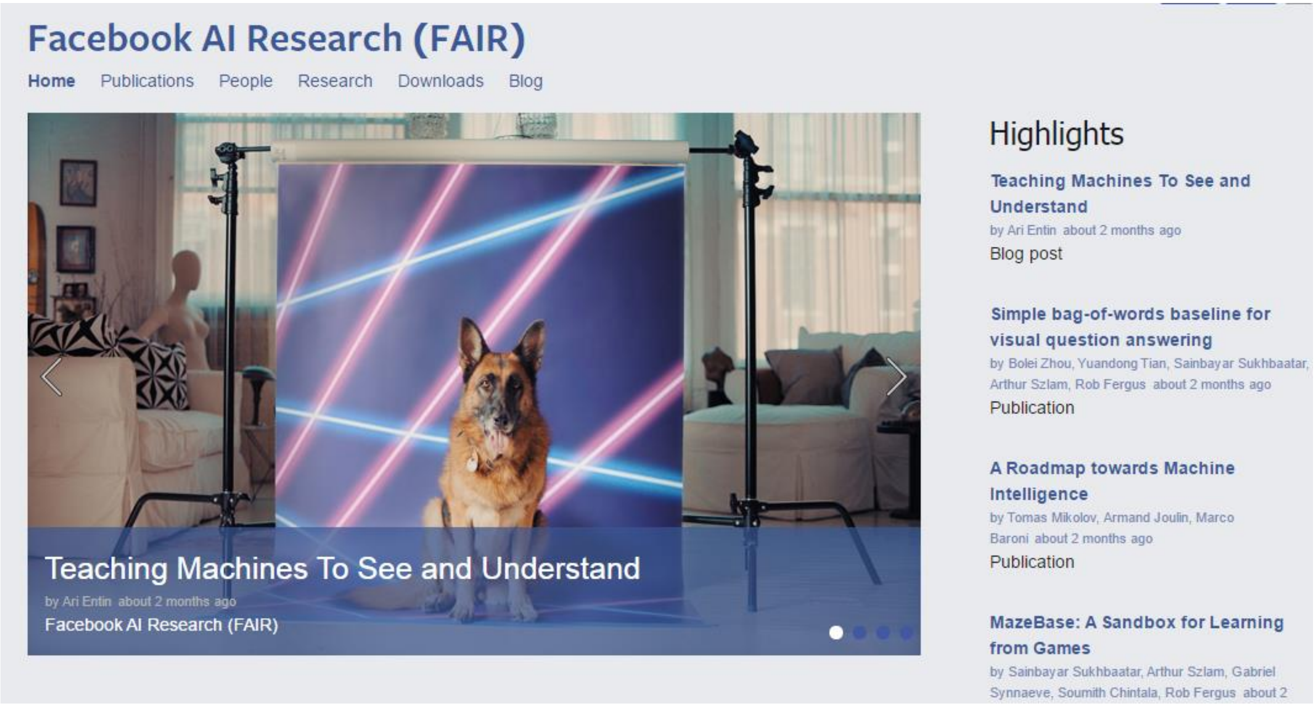
Industry

- Google



Industry

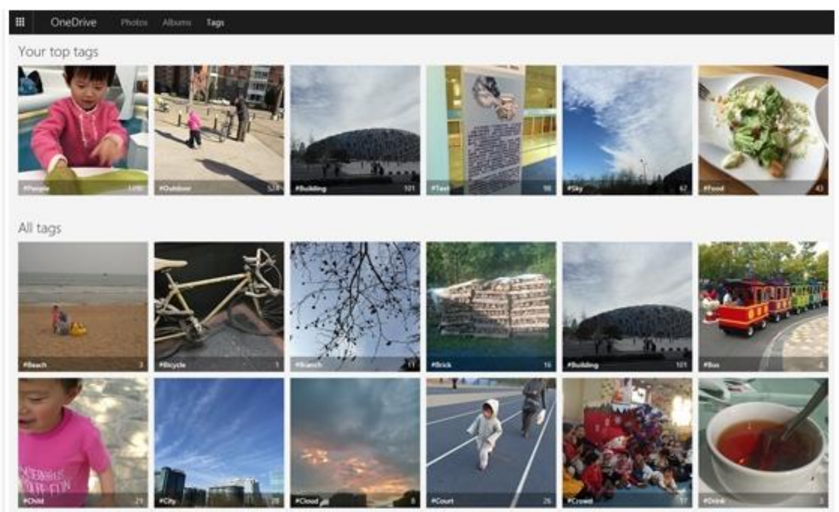
- Facebook



Industry

- Microsoft

Microsoft Researchers' Algorithm Sets ImageNet Challenge Milestone



Industry

- Toyota

SEARCH

The New York Times

TECHNOLOGY

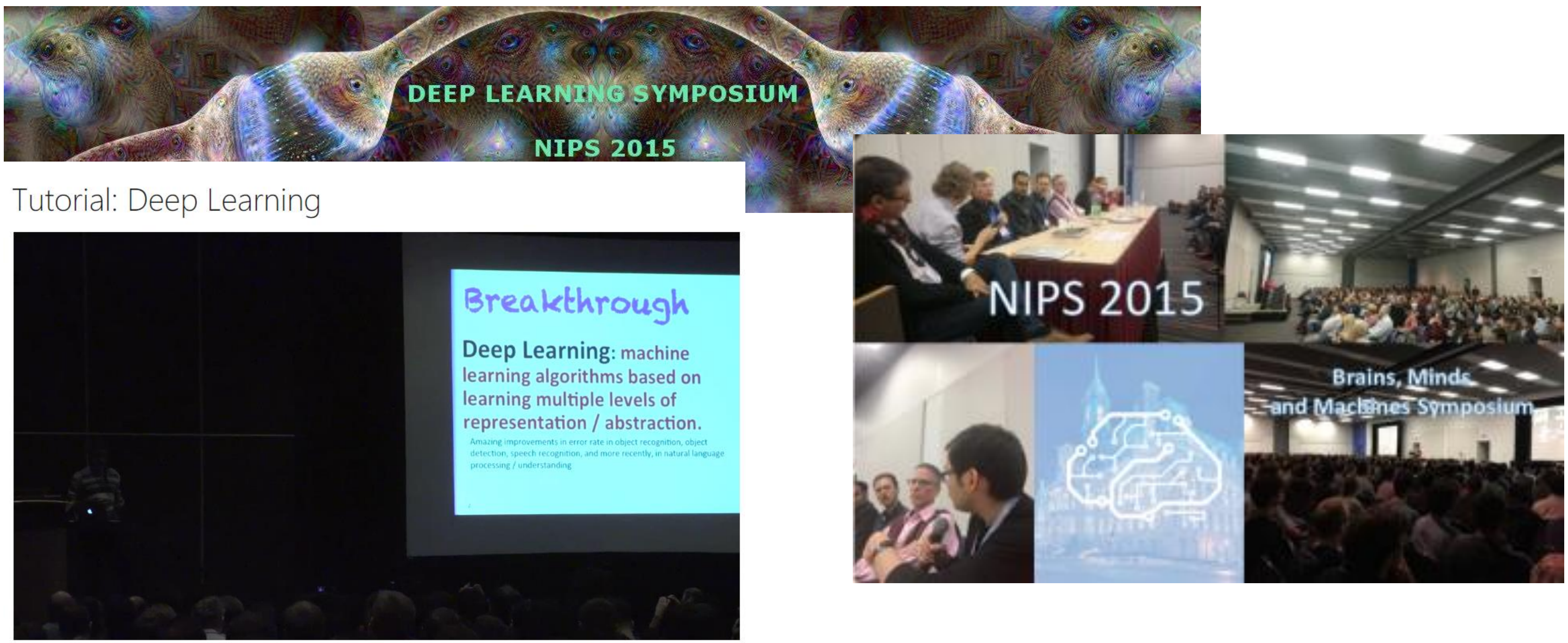
Toyota Invests $1 Billion in Artificial Intelligence

By JOHN MARKOFF NOV. 6, 2015

Gill Pratt, a roboticist who will oversee Toyota's new research laboratory in the United States, at a news conference Friday in Tokyo. Yuya Shino/Reuters

Academy

- NIPS 2015: ~4000 attendees, double the number of NIPS 2014



Academy

- Science special issue
- Nature invited review

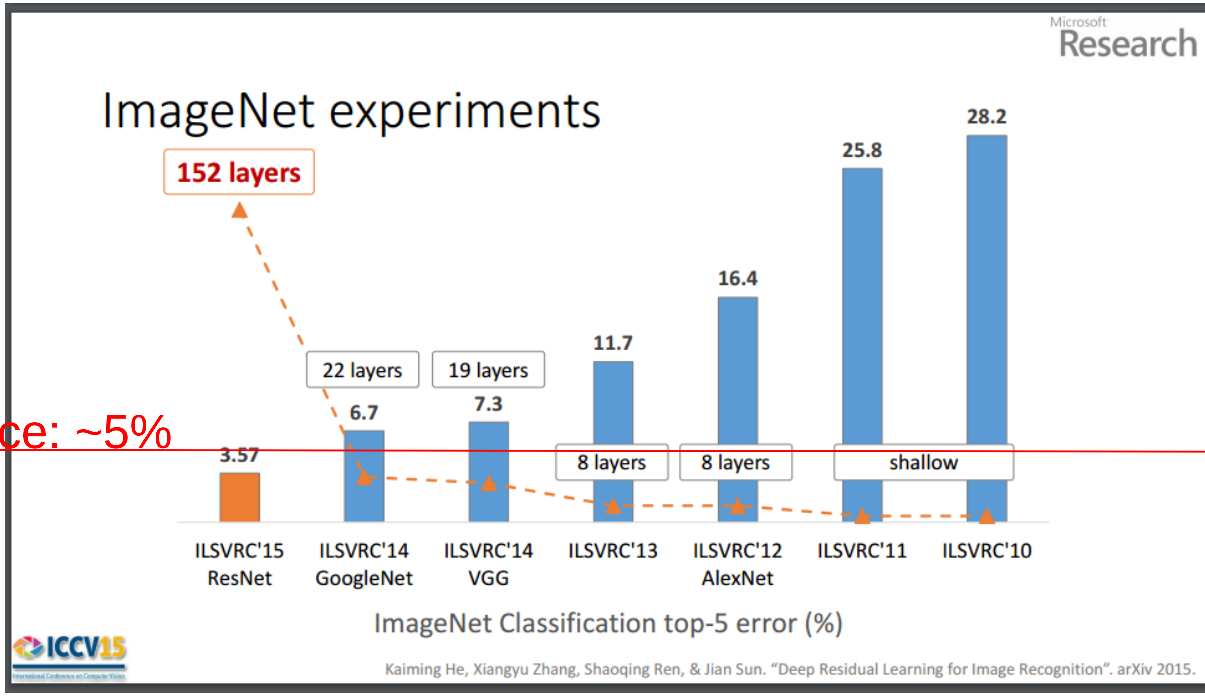
REVIEW

Deep learning

Yann LeCun[1,2], Yoshua Bengio[3] & Geoffrey Hinton[4,5]

Image

- Image classification
  - 1000 classes

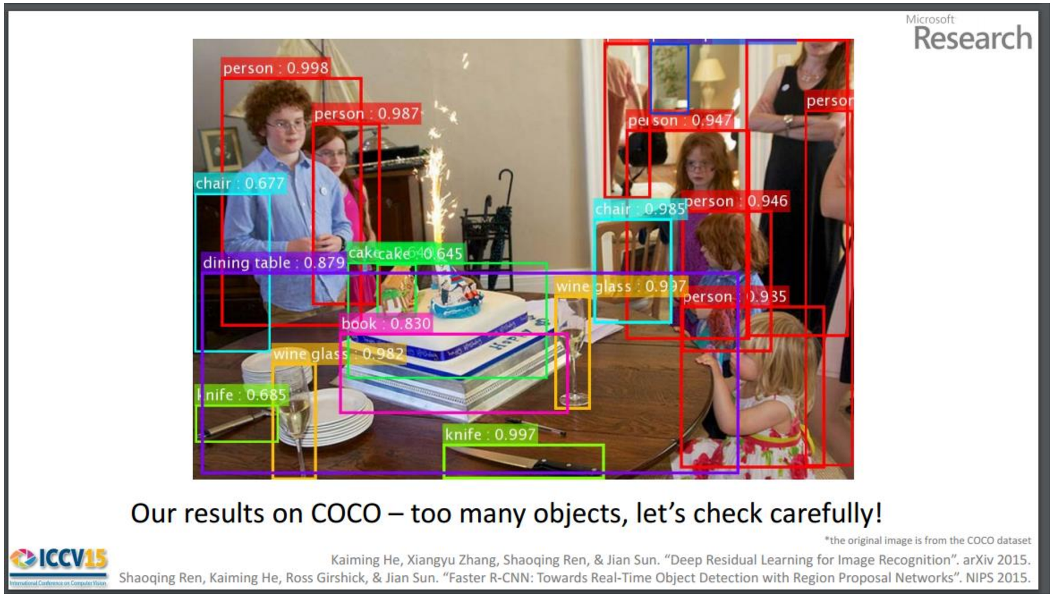


Human performance: ~5%

Slides from Kaimin He, MSRA

Image

- Object location



Slides from Kaimin He, MSRA

Image

- Image captioning

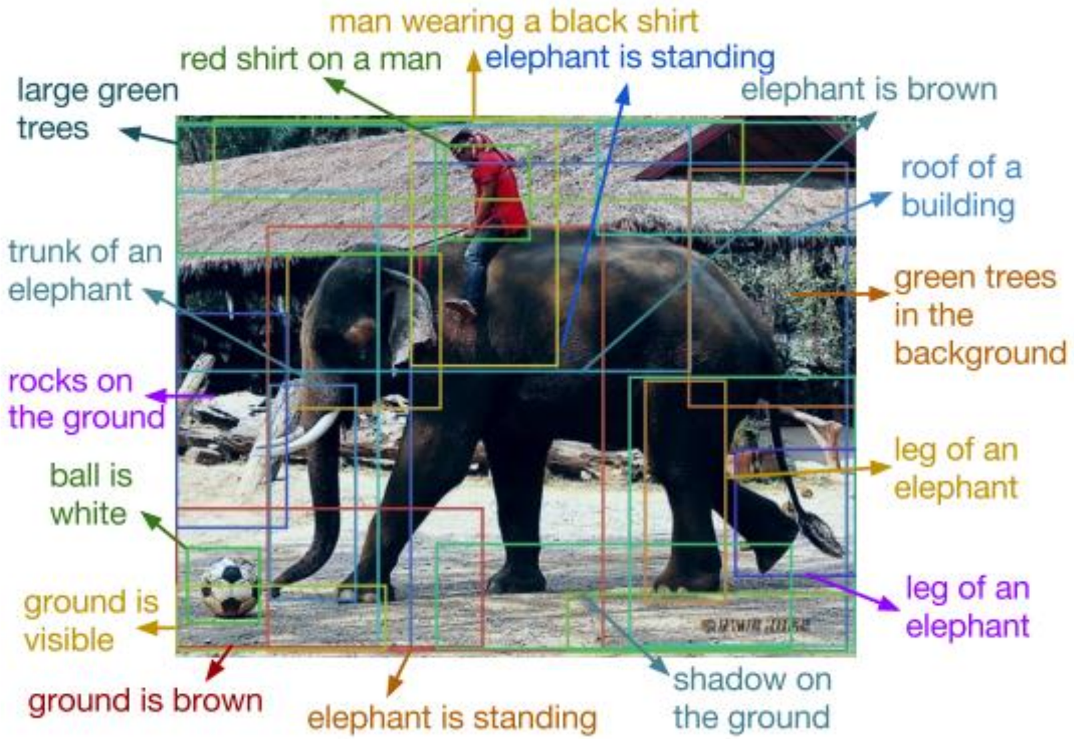


Figure from the paper "DenseCap: Fully Convolutional Localization Networks for Dense Captioning", by Justin Johnson, Andrej Karpathy, Li Fei-Fei

Text

- Question & Answer

I:	Jane went to the hallway.
I:	Mary walked to the bathroom.
I:	Sandra went to the garden.
I:	Daniel went back to the garden.
I:	Sandra took the milk there.
Q:	Where is the milk?
A:	garden

I:	The answer is far from obvious.
Q:	In French?
A:	La réponse est loin d'être évidente.

Figures from the paper "Ask Me Anything: Dynamic Memory Networks for Natural Language Processing ",
by Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Richard Socher

Game

Google DeepMind's Deep Q-learning playing Atari Breakout
From the paper “Playing Atari with Deep Reinforcement Learning”,
by Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller

Game


nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
CONSERVATION
SONGBIRDS À LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452
RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on individuals
PAGE 459
POPULAR SCIENCE
WHEN GENES GOT 'SELFISH'
Dawkins's calling card forty years on
PAGE 462
NATURE.COM/NATURE
28 January 2016 £10
Vol. 529, No. 7587

The impact

- Revival of Artificial Intelligence
- Next technology revolution?

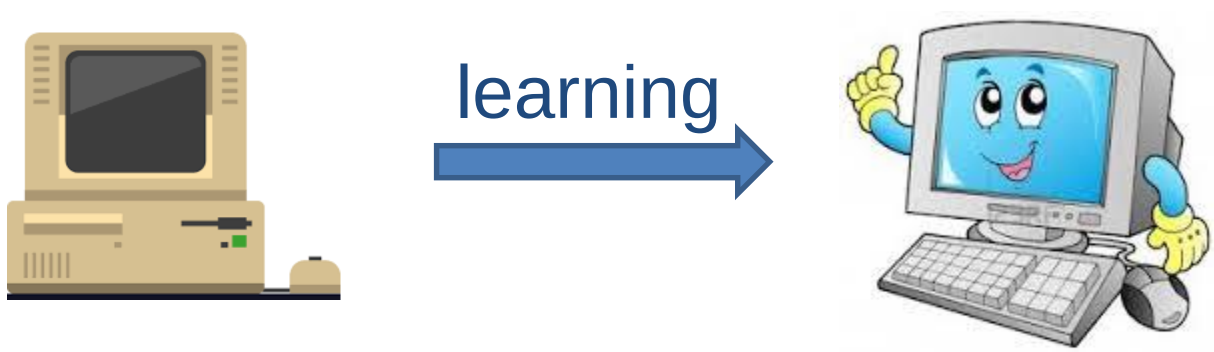
- A big thing ongoing, should not miss

MACHINE LEARNING BASICS

What is machine learning?

- "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T as measured by P, improves with experience E."

  ------- *Machine Learning*, Tom Mitchell, 1997

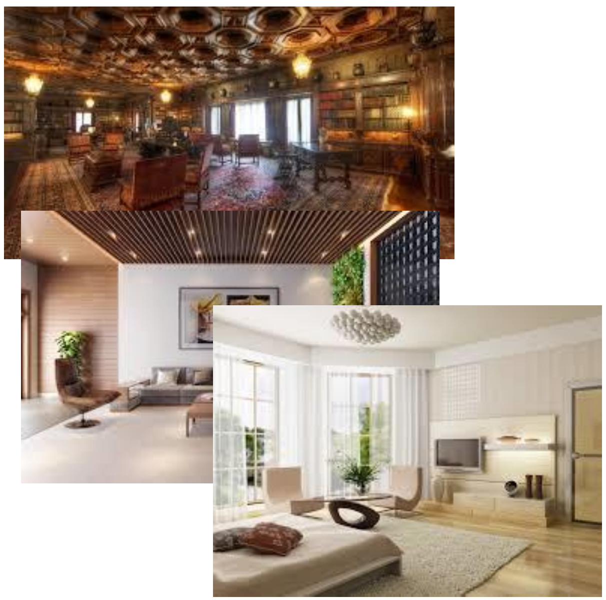


Example 1: image classification

Task: determine if the image is indoor or outdoor

Performance measure: probability of misclassification

Example 1: image classification



Experience/Data:
images with labels

Indoor

outdoor

Example 1: image classification

- A few terminologies
  - Instance
  - Training data: the images given for learning
  - Test data: the images to be classified

Example 1: image classification (multi-class)

ImageNet figure borrowed from vision.standford.edu

Example 2: clustering images

Task: partition the images into 2 groups
Performance: similarities within groups
Data: a set of images

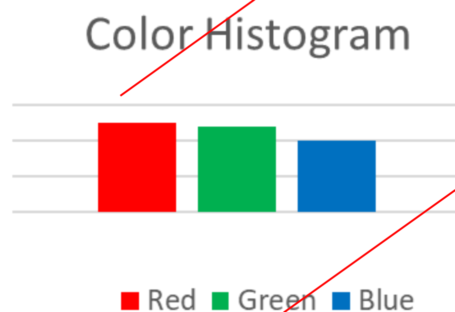
Example 2: clustering images

- A few terminologies
  - Unlabeled data vs labeled data
  - Supervised learning vs unsupervised learning

Feature vectors

Feature vector: x_i

Extract
features



Label: y_i

Indoor

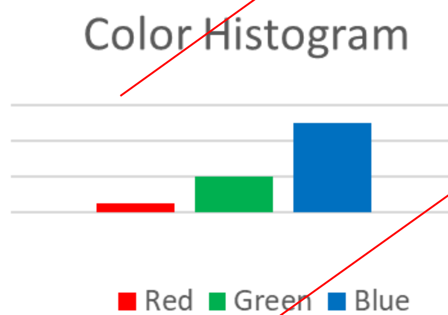
0

Feature space

Feature vectors

Feature vector: x_i

Extract
features



Label: y_i

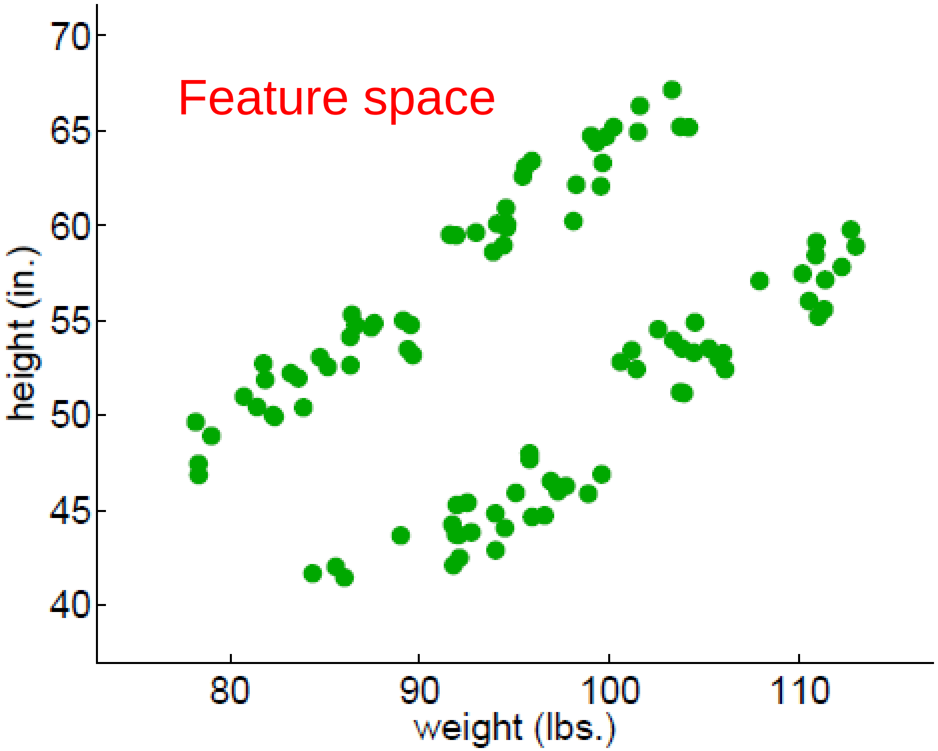
outdoor

1

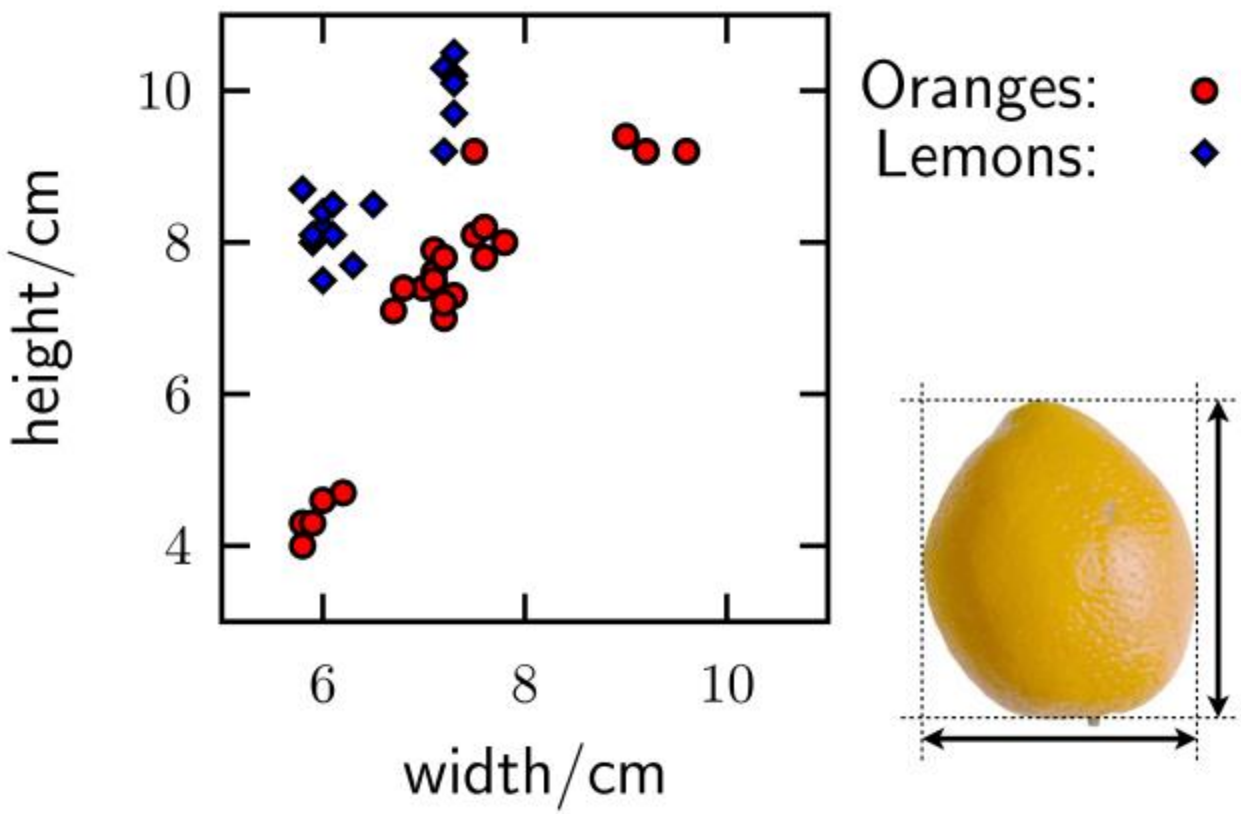
Feature space

Feature Example 2: little green men

- The weight and height of 100 little green men



Feature Example 3: Fruits



- From Iain Murray http://homepages.inf.ed.ac.uk/imurray2/

Feature example 4: text

- Text document
  - Vocabulary of size D (~100,000)
- “bag of word”: counts of each vocabulary entry
  - To marry my true love → (3531:1 13788:1 19676:1)
  - I wish that I find my soulmate this year → (3819:1 13448:1 19450:1 20514:1)
- Often remove stopwords: the, of, at, in, …
- Special “out-of-vocabulary” (OOV) entry catches all unknown words

UNSUPERVISED LEARNING BASICS

Unsupervised learning

in unsupervised learning, we're given a set of instances, without labels

$$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(m)}$$

goal: discover interesting regularities/structures/patterns that characterize the instances

Common tasks:

- clustering, separate the *n* instances into groups
- novelty detection, find instances that are very different from the rest
- dimensionality reduction, represent each instance with a lower dimensional feature vector while maintaining key characteristics of the training samples

Anomaly detection

learning task

given

- training set of instances $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(m)}$

output

- model h that represents "normal" $\boldsymbol{x}$

performance task

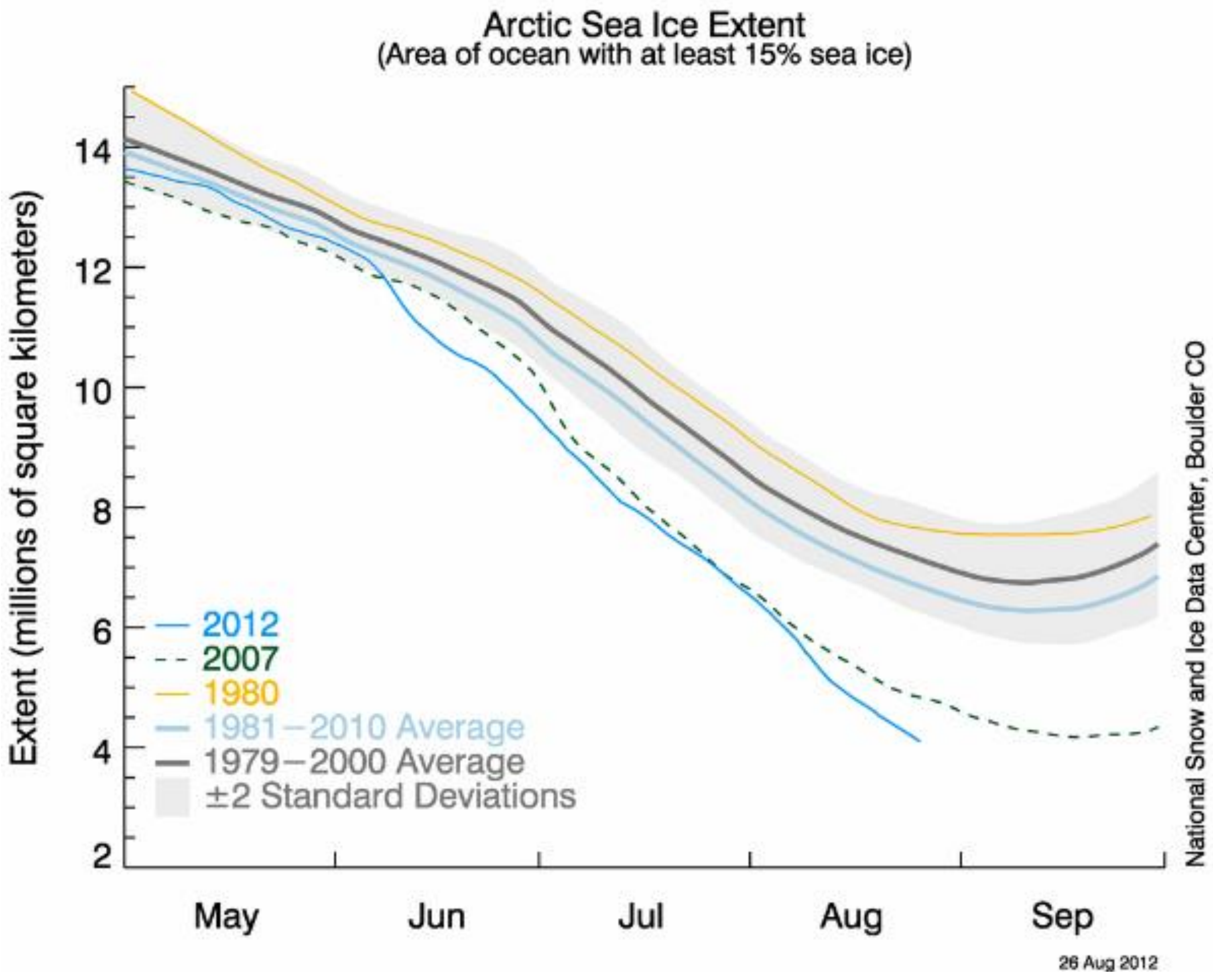
given

- a previously unseen $\boldsymbol{x}$

determine

- if $\boldsymbol{x}$ looks normal or anomalous

Anomaly detection example



Let's say our model is represented by: 1979-2000 average, ±2 stddev

Does the data for 2012 look anomalous?

Dimensionality reduction

given

- training set of instances $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(m)}$

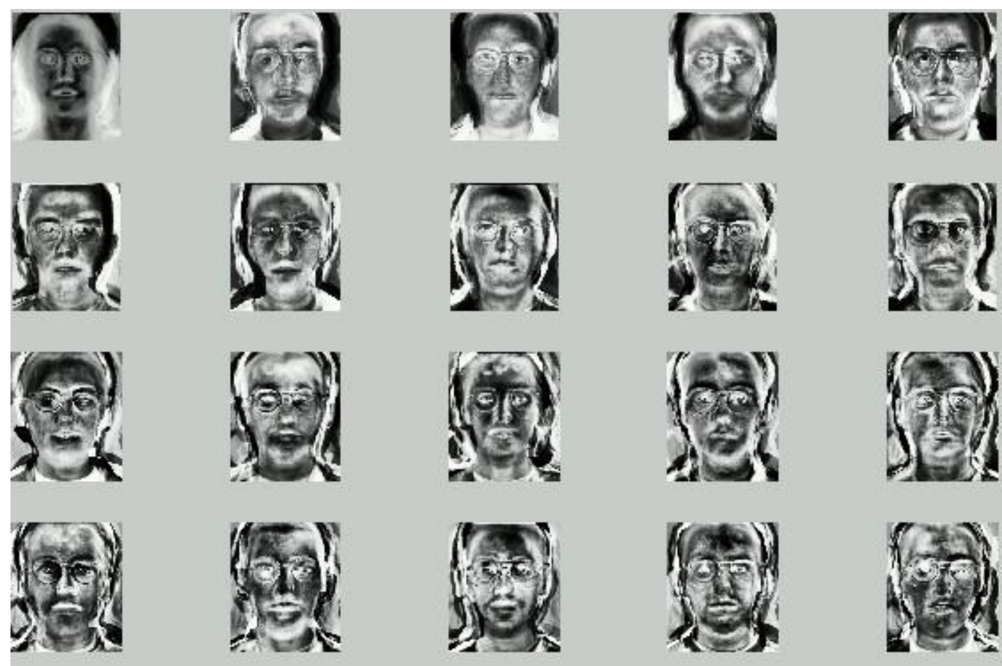
output

- model h that represents each $\boldsymbol{x}$ with a lower-dimension feature vector while still preserving key properties of the data

Dimensionality reduction example

We can represent a face using all of the pixels in a given image

More effective method (for many tasks): represent each face as a linear combination of *eigenfaces*



Clustering

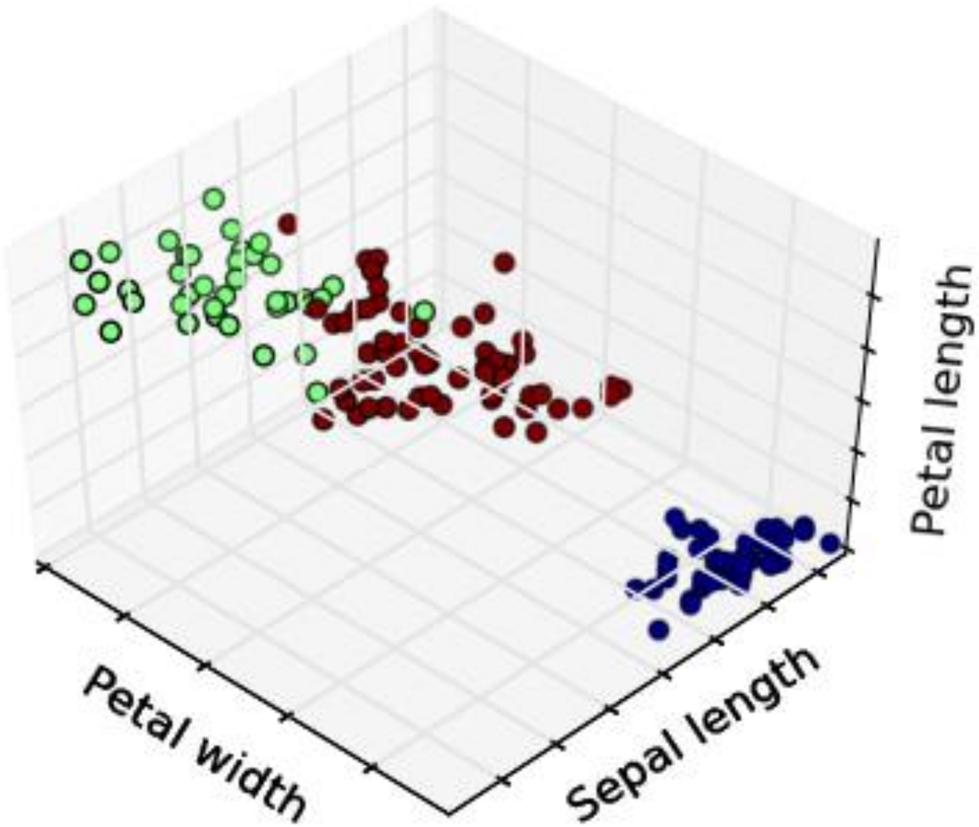
given

- training set of instances $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(m)}$

output

- model h that divides the training set into clusters such that there is intra-cluster similarity and inter-cluster dissimilarity

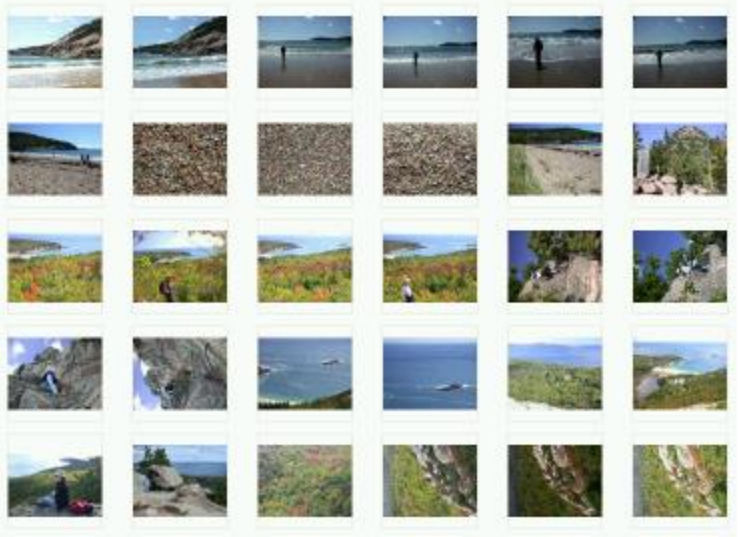
Example 1: Irises



Clustering irises using three different features (the colors represent clusters identified by the algorithm, not y's provided as input)

Example 2: your digital photo collection

- You probably have >1000 digital photos, 'neatly' stored in various folders…
- After this class you'll be about to organize them better
  - Simplest idea: cluster them using image creation time (EXIF tag)
  - More complicated: extract image features



Two most frequently used methods

- Many clustering algorithms. We'll look at the two most frequently used ones:
  - Hierarchical clustering

    Where we build a binary tree over the dataset
  - K-means clustering

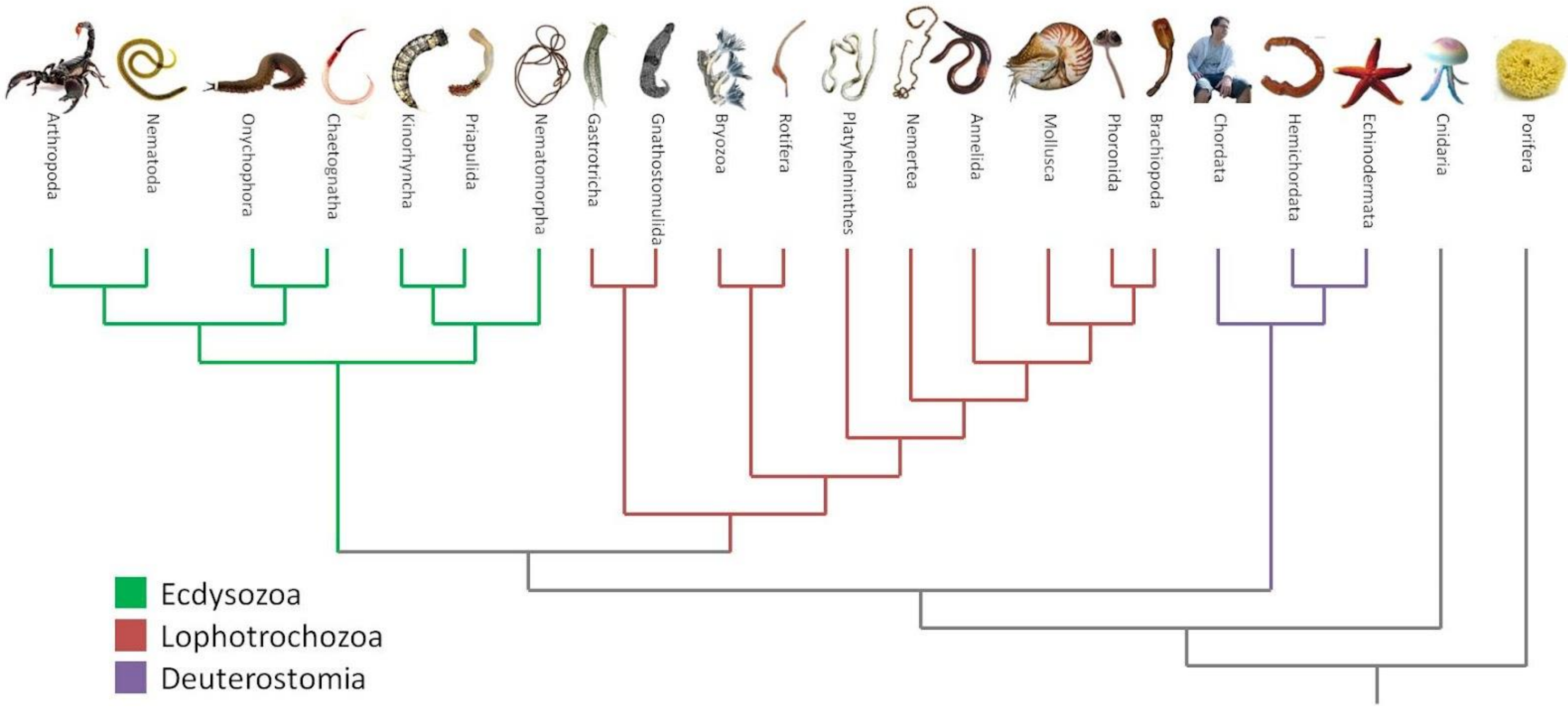
    Where we specify the desired number of clusters, and use an iterative algorithm to find them

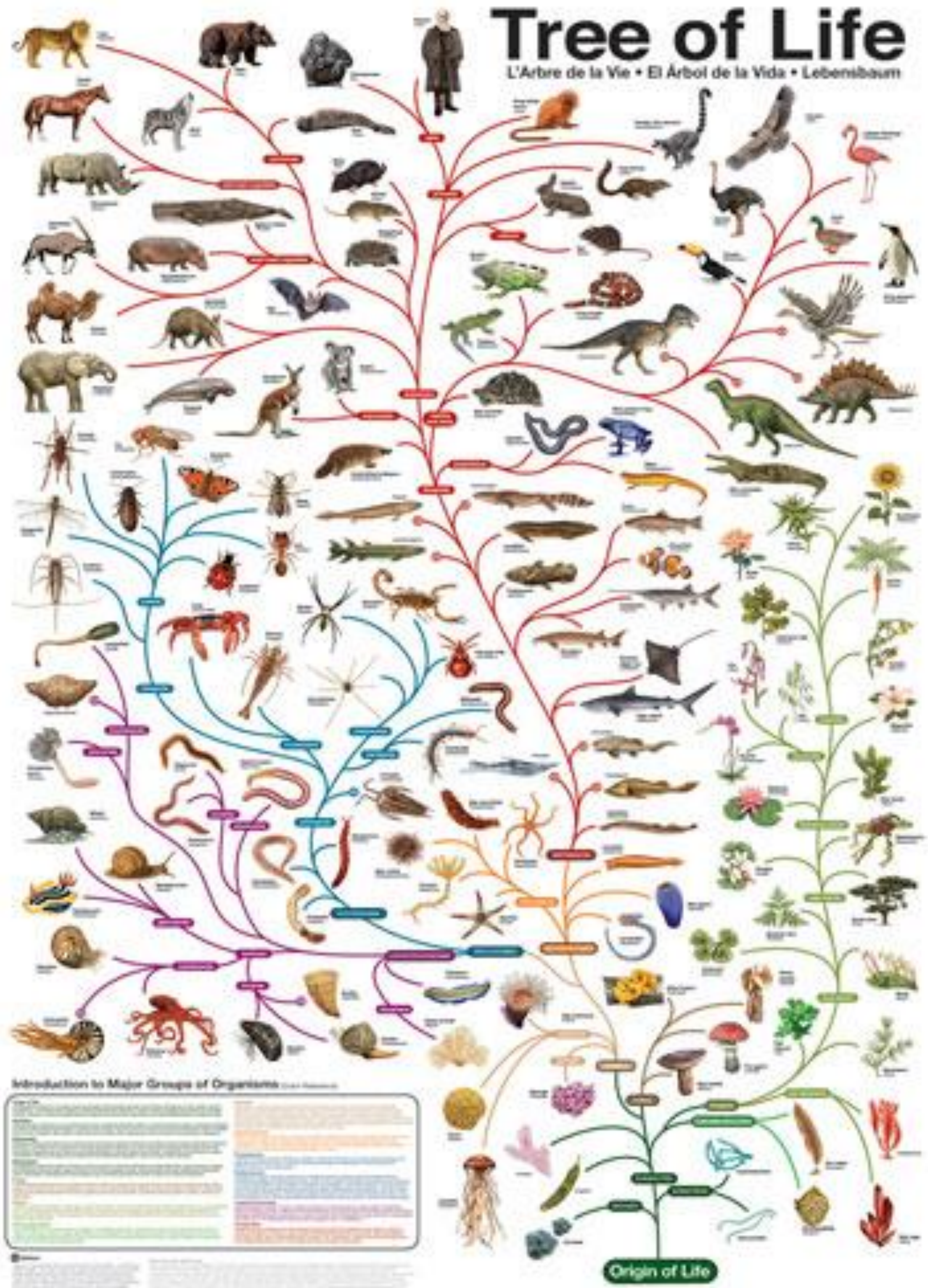
HIERARCHICAL CLUSTERING

Hierarchical clustering

- Very popular clustering algorithm
- Input:
  - A dataset $x_1, \ldots, x_n$, each point is a numerical feature vector
  - Does NOT need the number of clusters

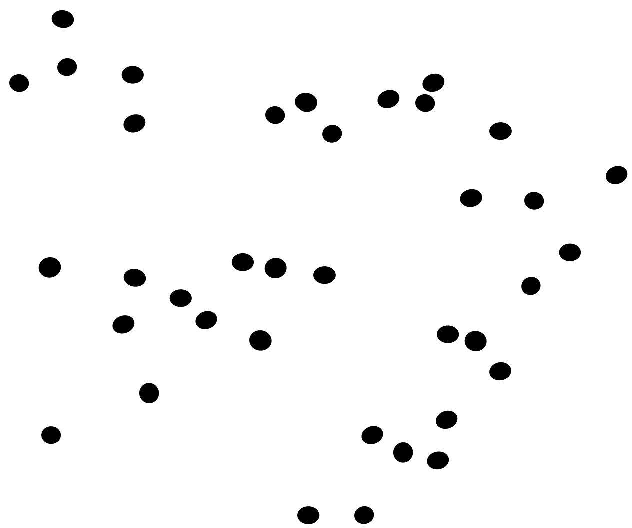
Building a hierarchy




Tree of Life
L'Arbre de la Vie • El Árbol de la Vida • Lebensbaum
Introduction to Major Groups of Organisms
Origin of Life

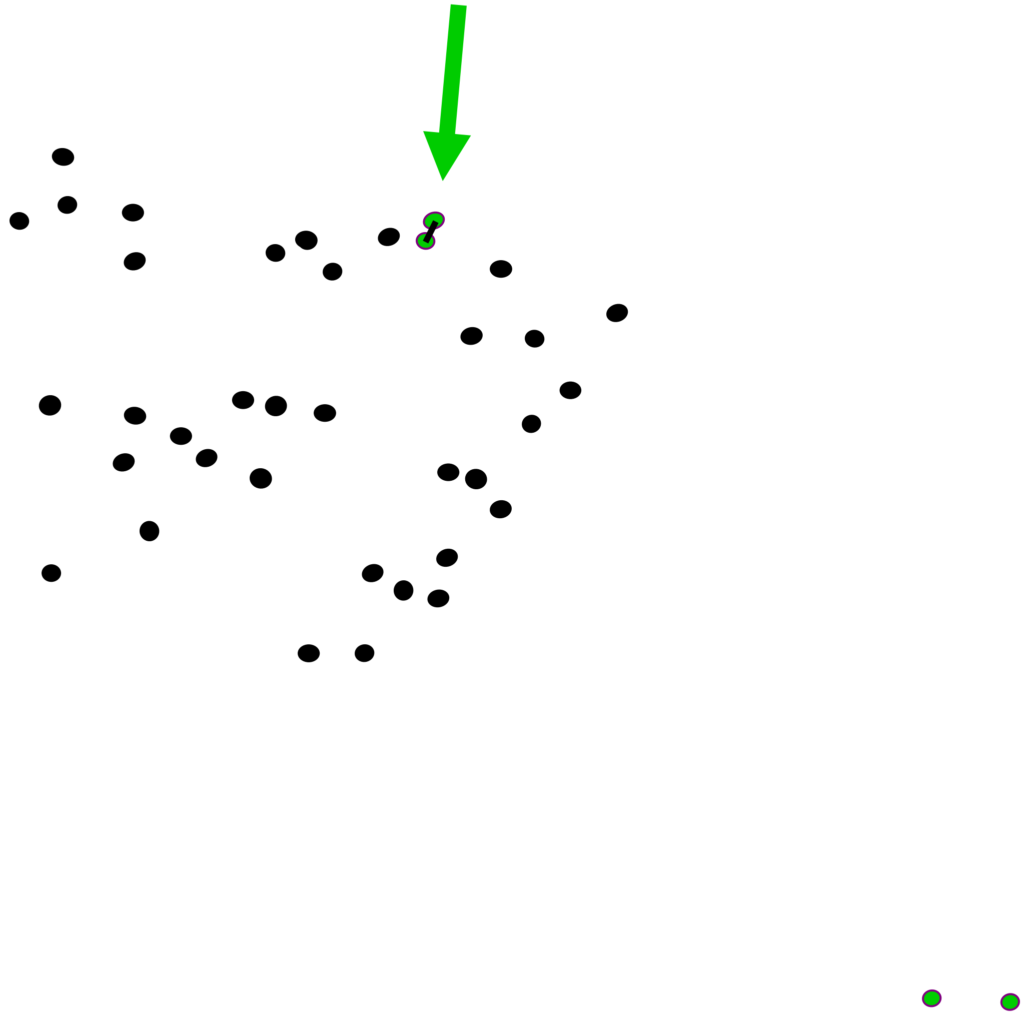
Hierarchical clustering

- Initially every point is in its own cluster



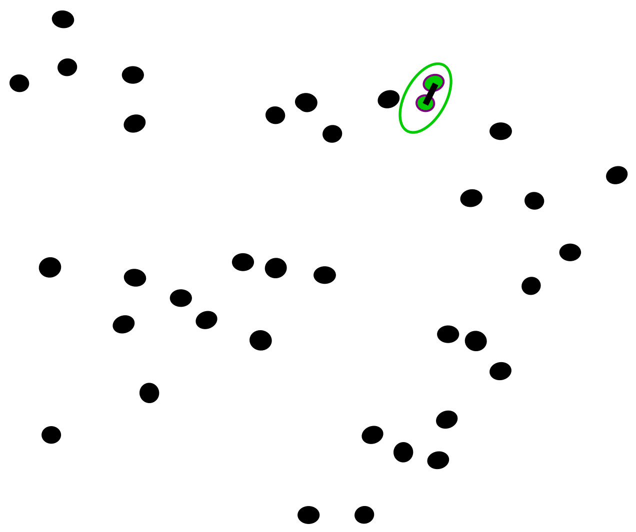
Hierarchical clustering

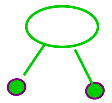
- Find the pair of clusters that are the closest



Hierarchical clustering

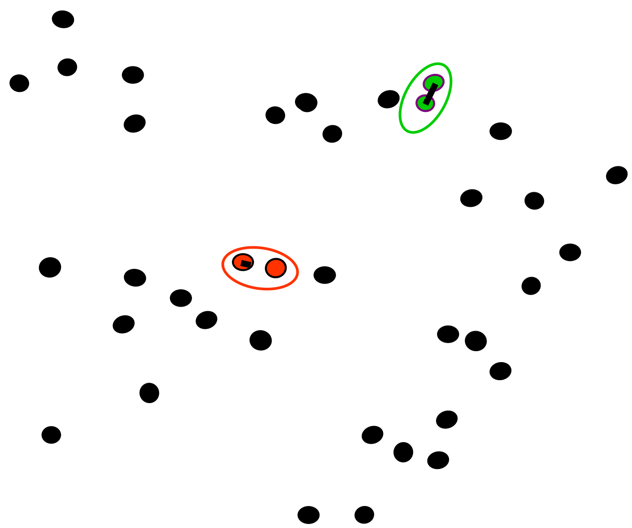
- Merge the two into a single cluster





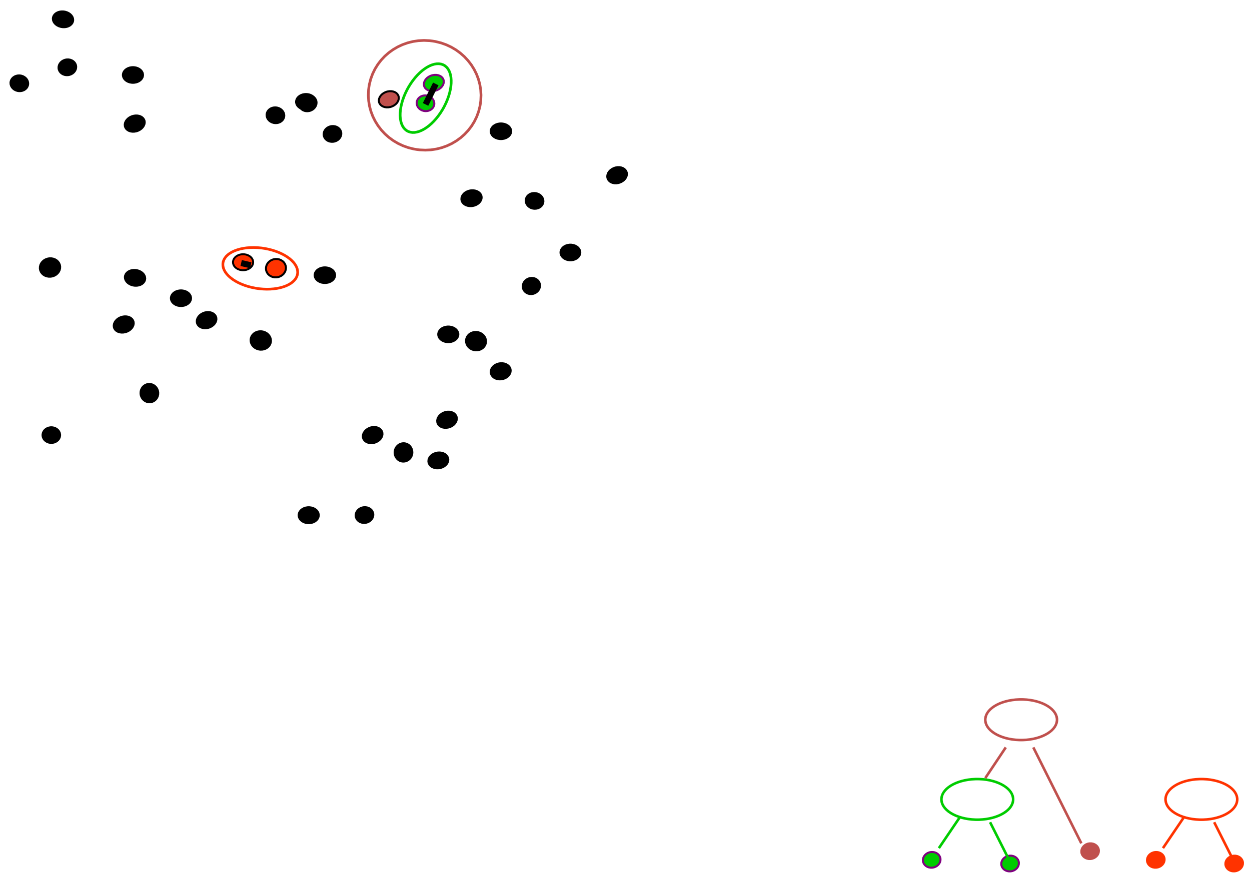
Hierarchical clustering

- Repeat…



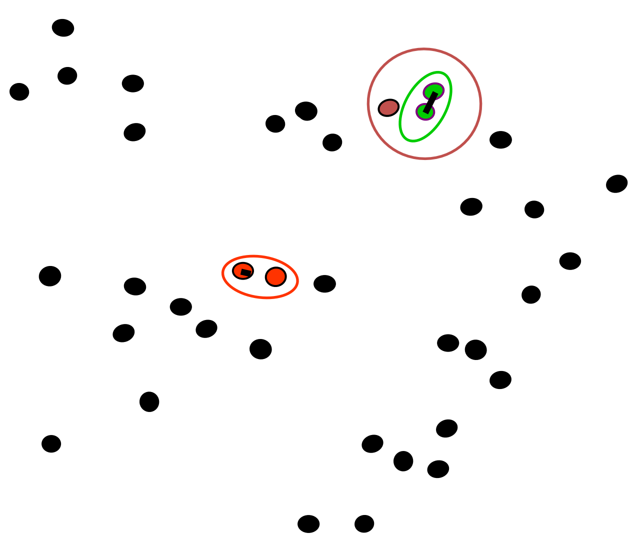
Hierarchical clustering

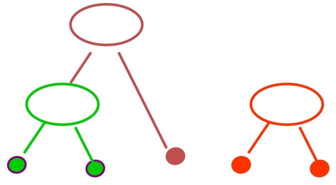
- Repeat…



Hierarchical clustering

- Repeat…until the whole dataset is one giant cluster
- You get a binary tree (not shown here)





Hierarchical Agglomerative Clustering

Input: *a training sample* $\{\mathbf{x}_i\}_{i=1}^{n}$; *a distance function* $d()$.
1. Initially, place each instance in its own cluster (called a singleton cluster).
2. ***while*** *(number of clusters* > 1*)* ***do:***
3. Find the closest cluster pair A, B, *i.e., they minimize* $d(A, B)$.
4. Merge A, B *to form a new cluster.*
Output: *a binary tree showing how clusters are gradually merged from singletons to a root cluster, which contains the whole training sample.*

- Euclidean (L2) distance

$$d(x_i, x_j) = \left\| x_i - x_j \right\| = \sqrt{\sum_{s=1}^{d} (x_{is} - x_{js})^2}$$

Hierarchical clustering

- How do you measure the closeness between two clusters?

Hierarchical clustering

- How do you measure the closeness between two clusters? At least three ways:
  - Single-linkage: the shortest distance from any member of one cluster to any member of the other cluster. Formula?
  - Complete-linkage: the greatest distance from any member of one cluster to any member of the other cluster
  - Average-linkage: you guess it!

Hierarchical clustering

- The binary tree you get is often called a dendrogram, or taxonomy, or a hierarchy of data points
- The tree can be cut at various levels to produce different numbers of clusters: if you want k clusters, just cut the $(k-1)$ longest links
- Sometimes the hierarchy itself is more interesting than the clusters
- However there is not much theoretical justification to it…

K-MEANS CLUSTERING

K-means clustering

- Clustering: What if we want k prototypical examples?





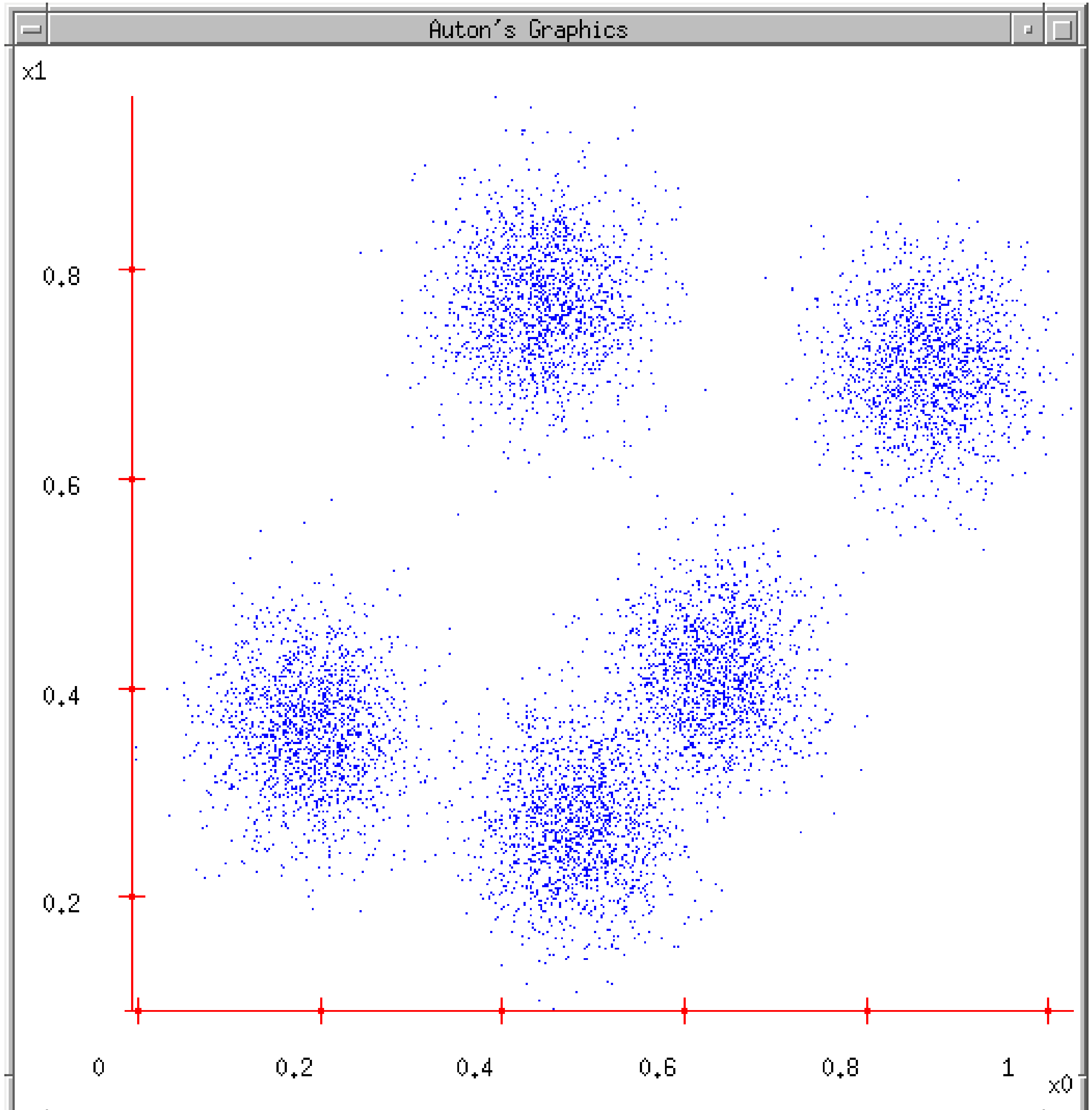

K-means clustering

- Very popular clustering method

- Input:
  - A dataset $x_1, \ldots, x_n$, each point is a numerical feature vector in R^d
  - Assume the number of clusters k is given

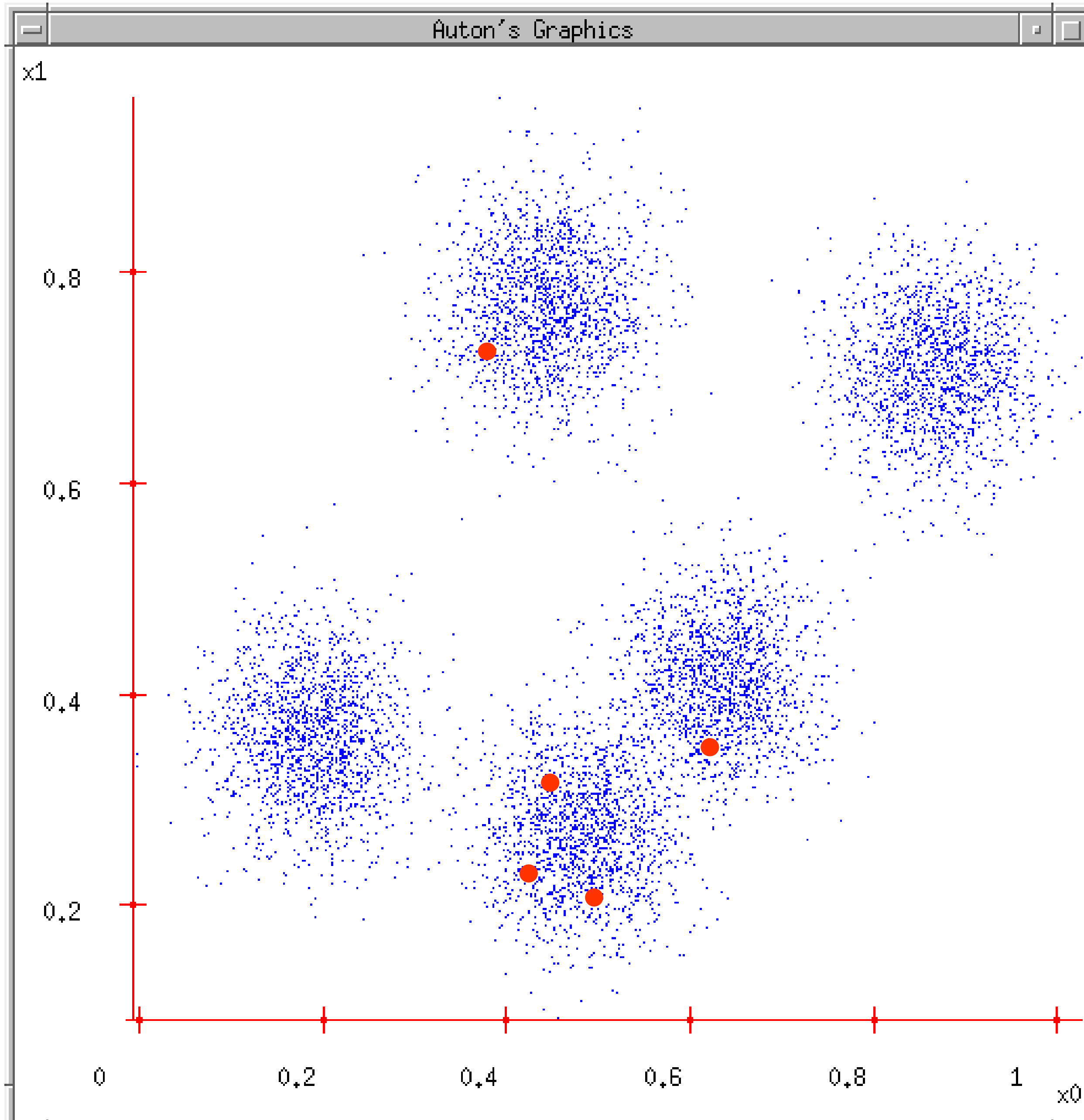
K-means clustering

- Input: dataset, $k = 5$



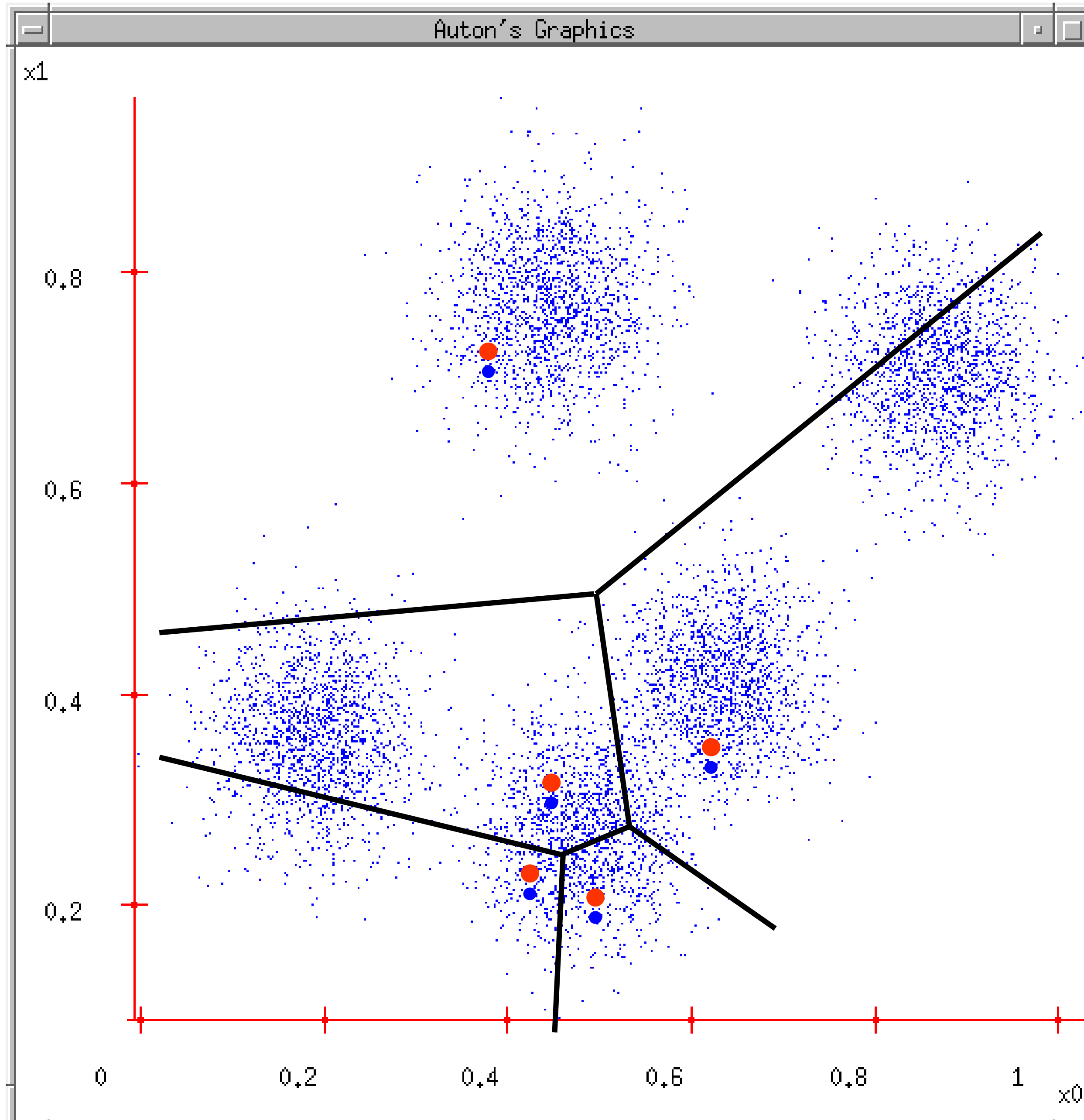
K-means clustering

- Randomly picking 5 positions as initial cluster centers (not necessarily a data point)



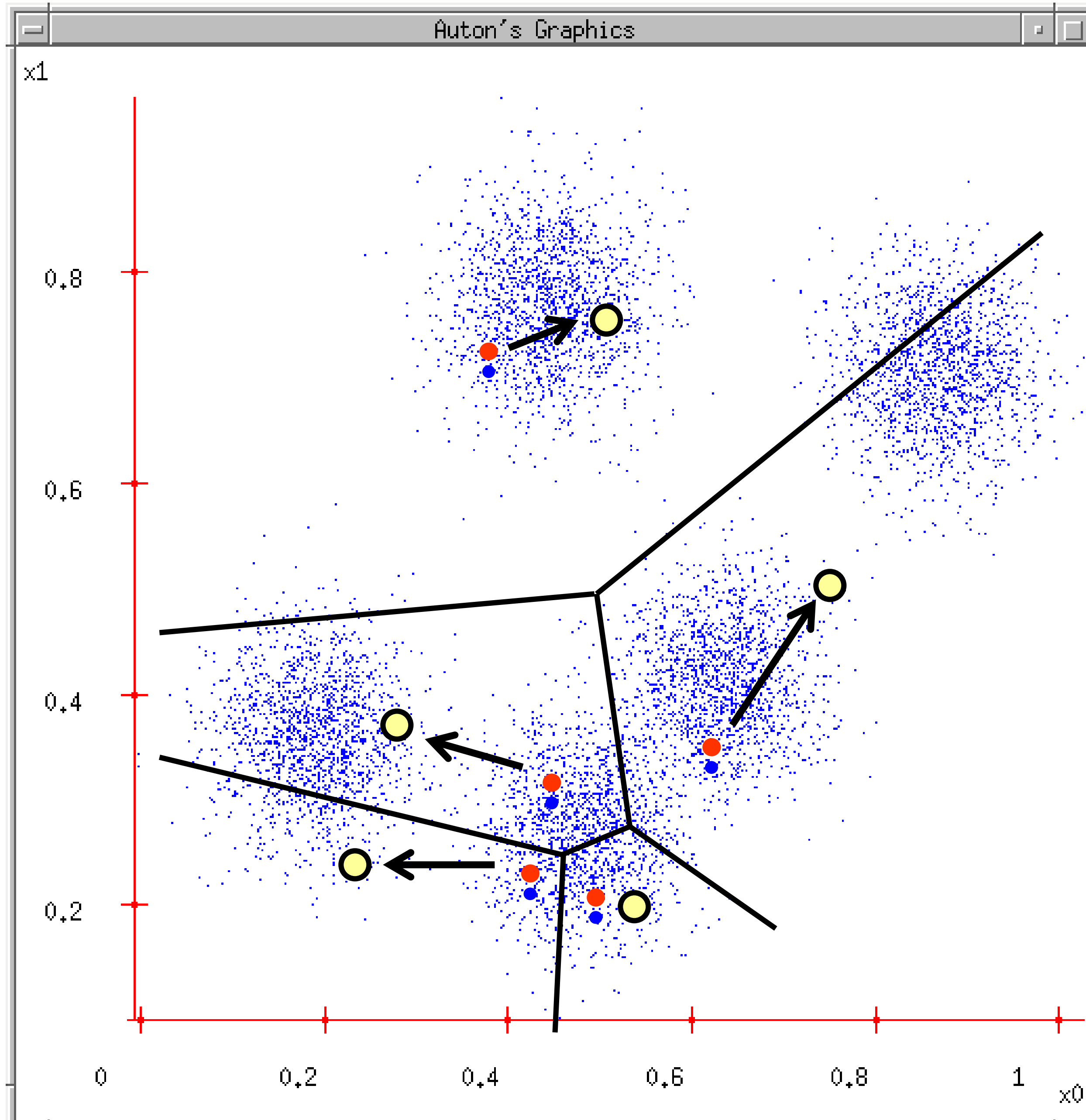
K-means clustering

- Each point finds which cluster center it is closest to. The point is assigned to that cluster.



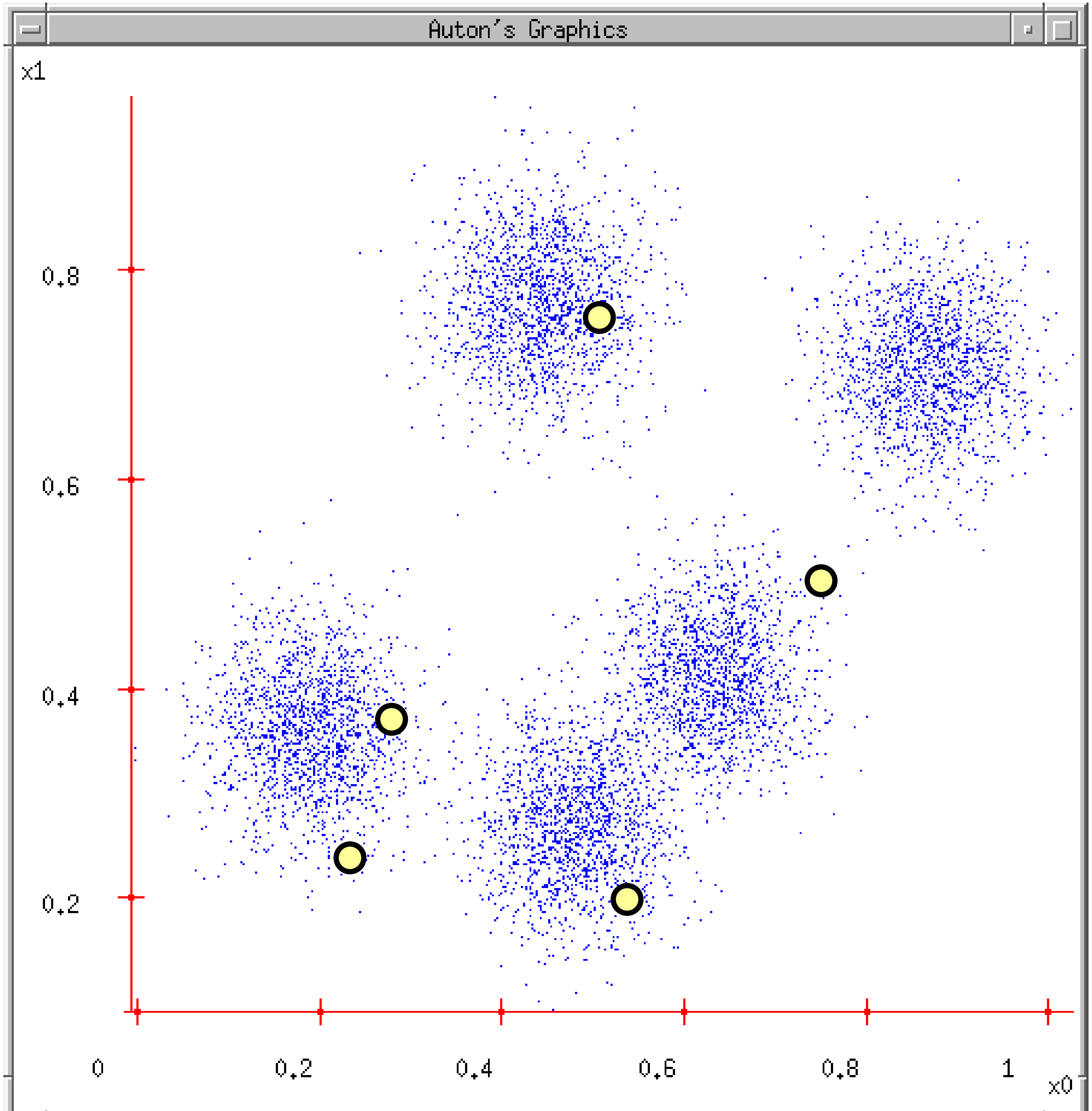
K-means clustering

- Each cluster computes its new centroid, based on which points belong to it



K-means clustering

- Each cluster computes its new centroid, based on which points belong to it
- And repeat until convergence (cluster centers no longer move)…



K-means algorithm

- Input: points $x_1, \ldots, x_n$, number of clusters k
- Select k centers $c_1, \ldots, c_k$
- **Step 1**: for each point x, determine its cluster: find the closest center in Euclidean distance
- **Step 2**: update all cluster centers as the centroids

$$c_i = \sum_{x \text{ in cluster } i} x \,/\text{SizeOf(cluster } i)$$

- Repeat step 1, 2 until the centers don't/slightly change

Questions on k-means

- What is k-means trying to optimize?
- Will k-means stop (converge)?
- Will it find a global or local optimum?
- How to pick starting cluster centers?
- How many clusters should we use?

Distortion

- Clustering as summarization: replace a point x with its center $c_{y(x)}$. How far are you off?
- The distortion of x is measured by squared Euclidean distance:

$$\left\|x - c_{y(x)}\right\|^2 = \sum_{i=1}^{d} \left[x_i - \left(c_{y(x)}\right)_i\right]^2$$

- The distortion of the whole dataset is

$$\sum_{x} \left\|x - c_{y(x)}\right\|^2$$

The optimization objective

- Minimize the distortion of the dataset

$$\min_{\substack{y(x_1),\ldots,y(x_n) \\ c_1,\ldots,c_k}} \sum_x \left\| x - c_{y(x)} \right\|^2$$

Step 1

- Suppose we fix the cluster centers
- Assigning x to its closest cluster center $y(x)$ minimizes the distortion

$$\left\|x - c_{y(x)}\right\|^2$$

Step 2

- Suppose we fix the assignment of points. All you can do is to change the cluster centers
- This is a continuous optimization problem!

$$\min_{c_1,\ldots,c_k} \sum_x \left\| x - c_{y(x)} \right\|^2$$

Step 2

- Suppose we fix the assignment of points. All you can do is to change the cluster centers
- This is a continuous optimization problem!

$$\min_{c_1,\dots,c_k} \sum_x \left\| x - c_{y(x)} \right\|^2$$

- Set the gradient to 0 leads to

$$c_i = \frac{\sum_{y(x)=i} x}{n_i}$$

Repeat (step1, step2)

- Both step1 and step2 minimizes the distortion
- Step1 changes the assignments $y(x)$
- Step2 changes the cluster centers c_z

- However there is no guarantee the distortion is minimized over all… need to repeat
- This is hill climbing (coordinate descent)
- Will it stop?

Repe

- Both step1 an
- Step1 change
- Step2 change

- However the
  is minimizec
- This is hill clin
- Will it stop?

There are finite number of points

Finite ways of assigning points to clusters

In step1, an assignment that reduces distortion has to be a new assignment not used before

Step1 will terminate

So will step 2

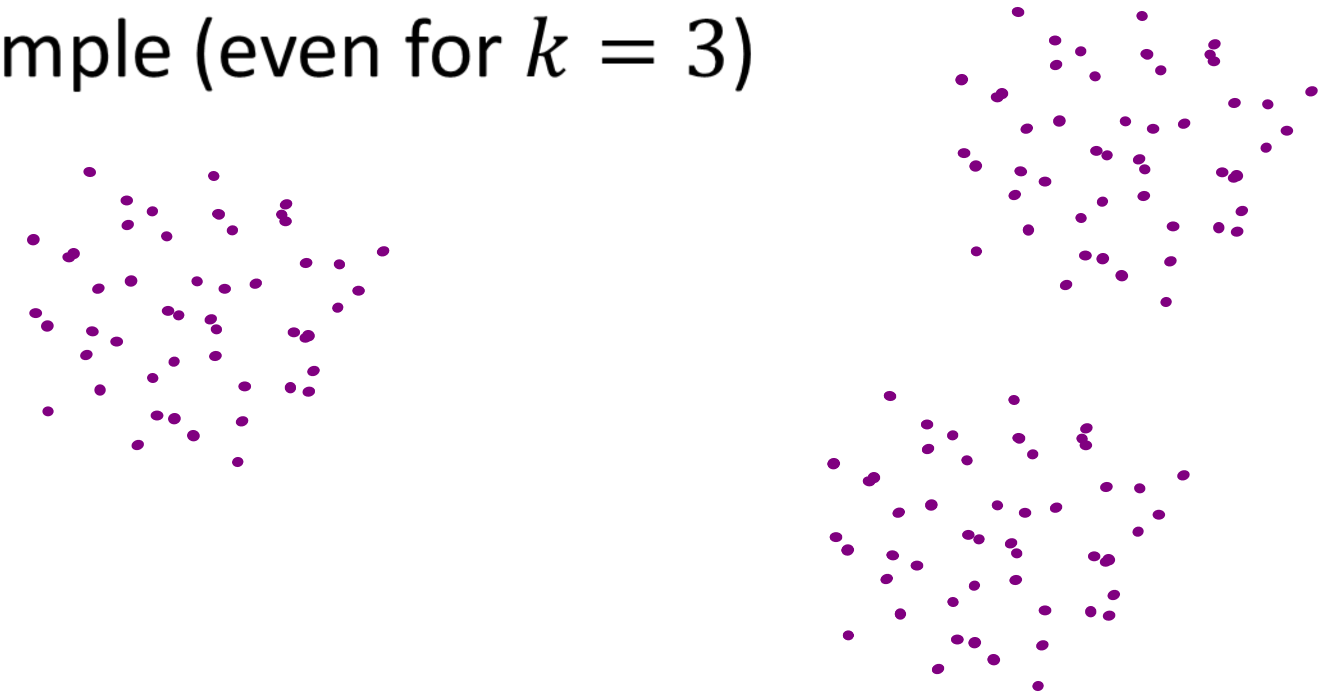
So k-means terminates

Will find global optimum?

- Sadly no guarantee

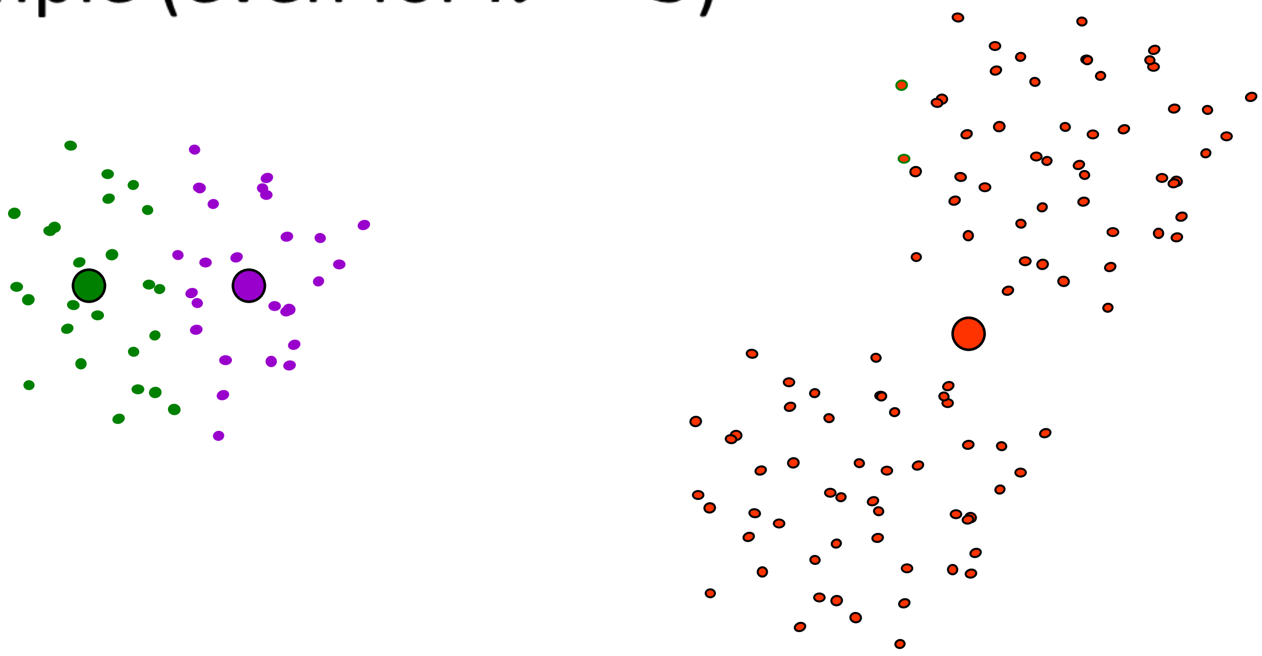
Will find global optimum?

- Sadly no guarantee
- Example (even for $k = 3$)



Will find global optimum?

- Sadly no guarantee
- Example (even for $k = 3$)



Picking starting cluster centers

- Which local optimum k-means goes to is determined solely by the starting cluster centers
  - Be careful how to pick the starting cluster centers. Many ideas. Here's one neat trick:
    1. Pick a random point x_1 from dataset
    2. Find the point x_2 farthest from x_1 in the dataset
    3. Find x_3 farthest from the closer of x_1, x_2
    4. … pick k points like this, use them as starting centers
  - Run k-means multiple times with different starting cluster centers (hill climbing with random restarts)

Picking the number of clusters

- Difficult problem
- Domain knowledge?
- Otherwise, shall we find k which minimizes distortion?

Picking the number of clusters

- Difficult problem
- Domain knowledge?
- Otherwise, shall we find k which minimizes distortion? $k = n$, distortion = 0
- Need to regularize. E.g., the Schwarz criterion

$$\text{distortion} + \lambda(\#param)\log n = \text{distortion} + \lambda d k \log n$$

#dimensions

#clusters

#points