Introduction to Machine Learning and Hierarchical Clustering

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

Academia

- Drastically increasing interest

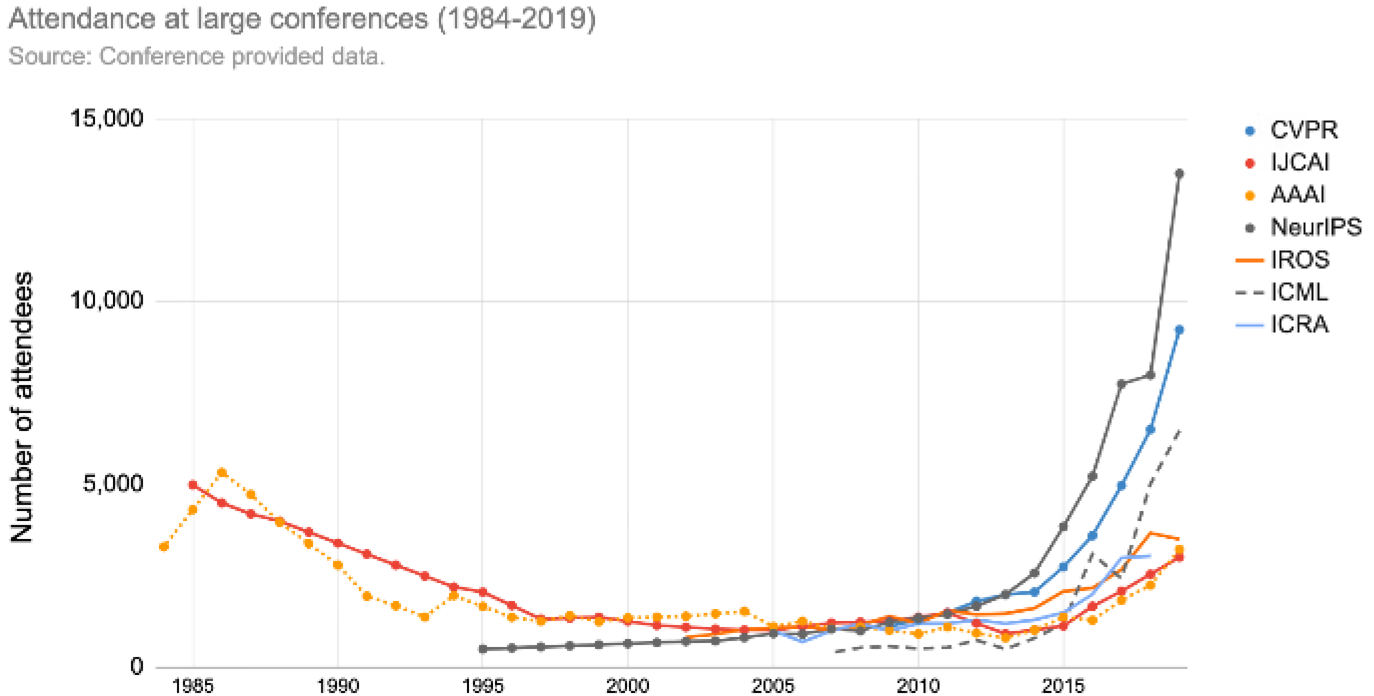


Fig. 2.1a

Academia

- Science special issue
- Nature invited review

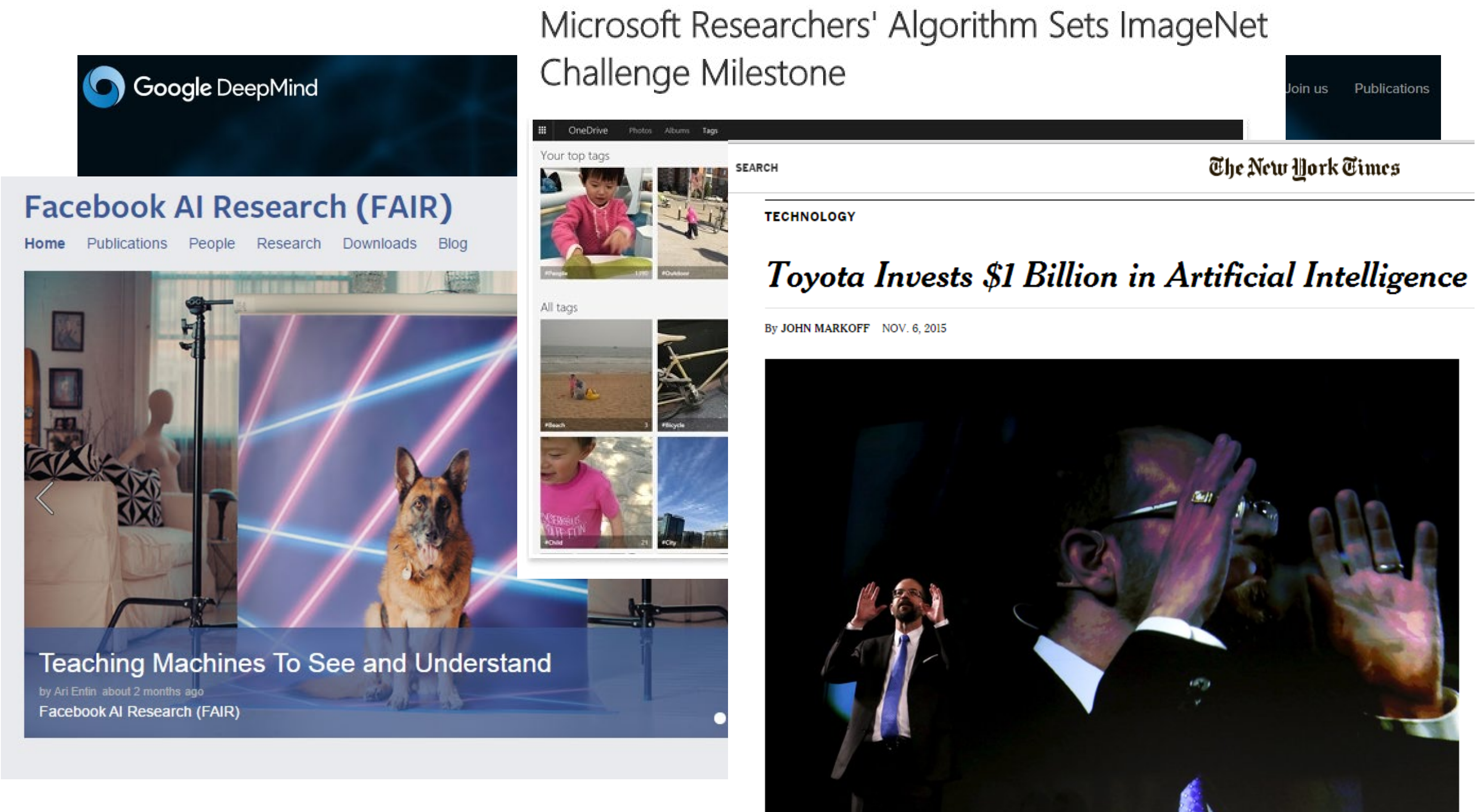
REVIEW

Deep learning

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

Industry

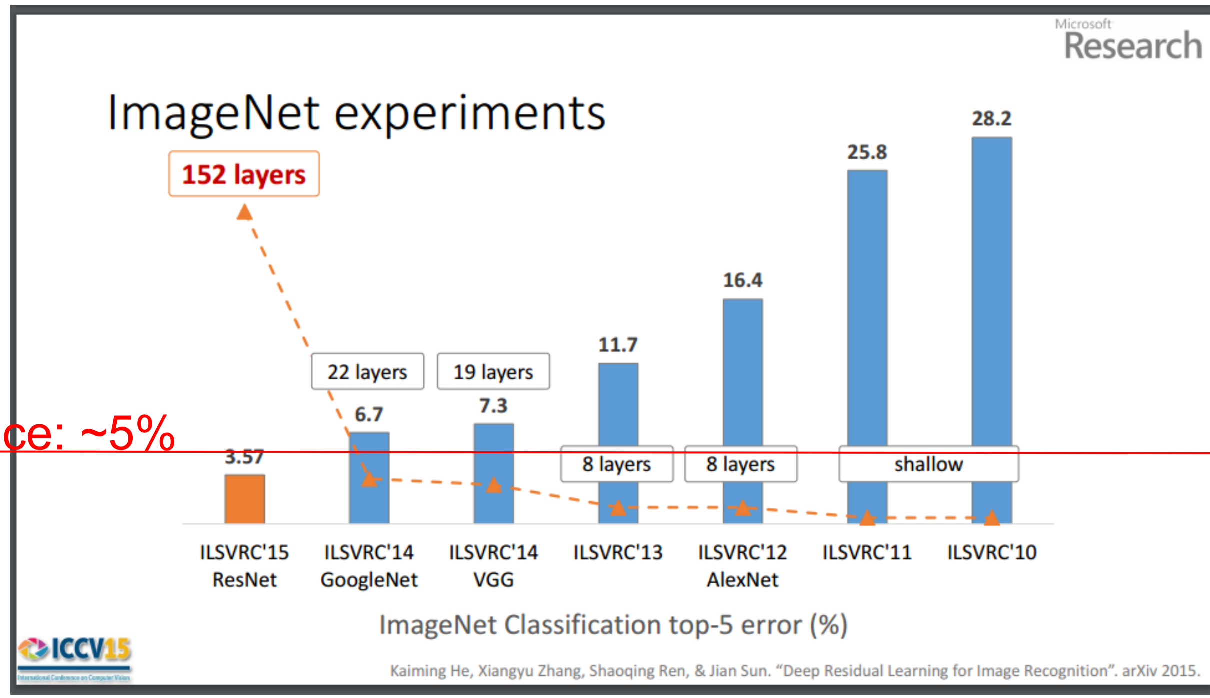
- Google, Facebook, Microsoft, Apple, Toyota, …



Image

- Image classification
  - 1000 classes

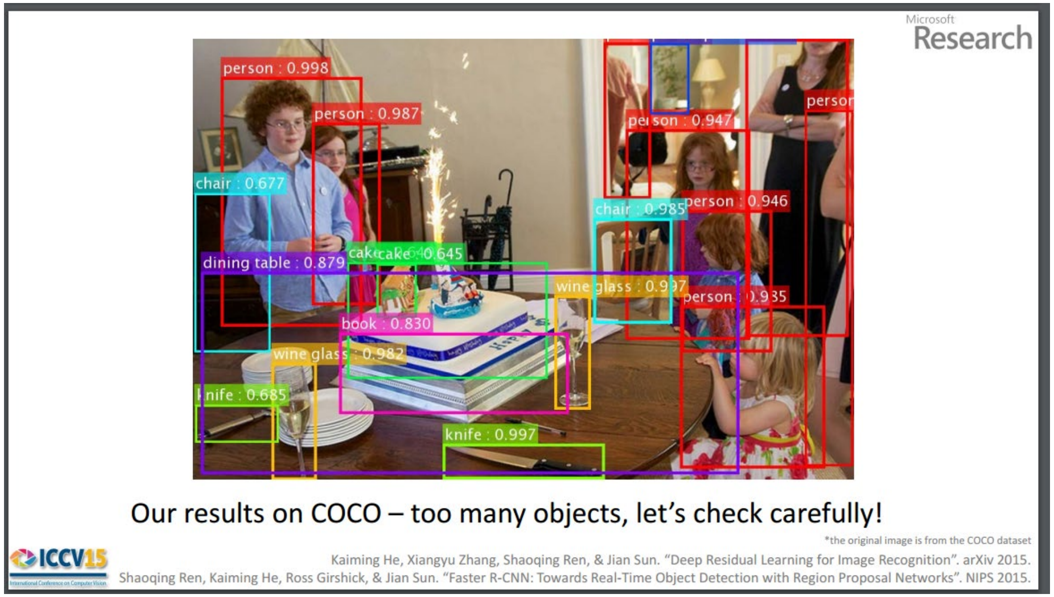
Human performance: ~5%



Slides from Kaimin He, MSRA

Image

- Object location



Slides from Kaimin He, MSRA

Image

- Image captioning

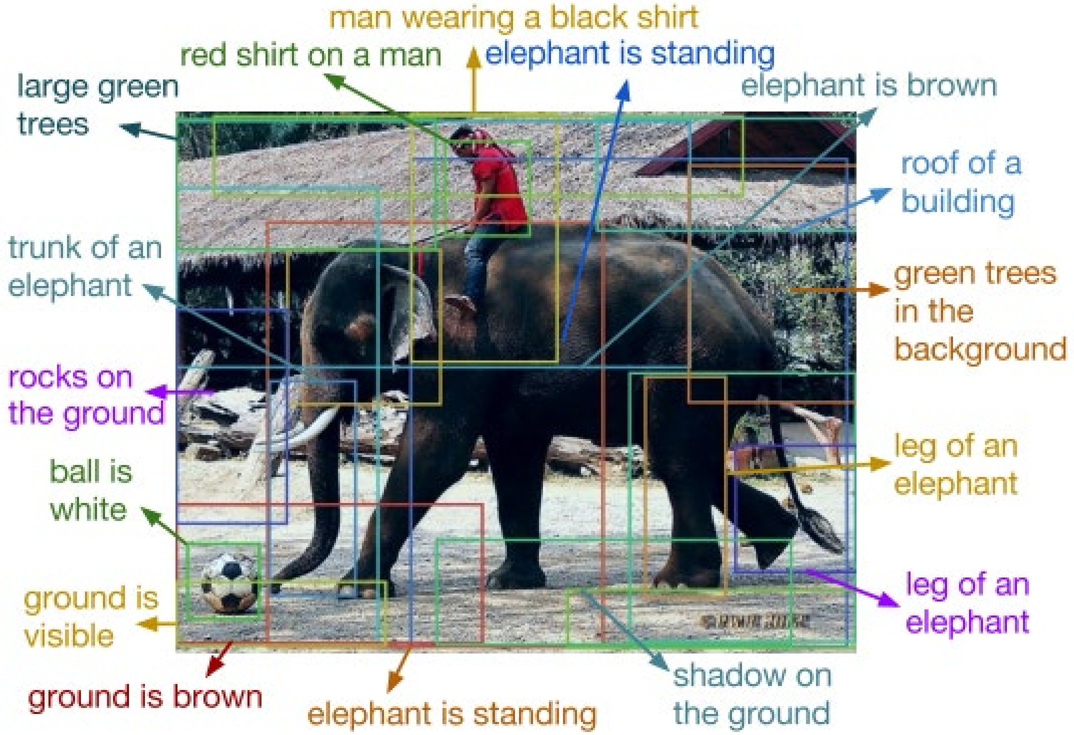


Figure from the paper "DenseCap: Fully Convolutional Localization Networks for Dense Captioning", by Justin Johnson, Andrej Karpathy, Li Fei-Fei

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

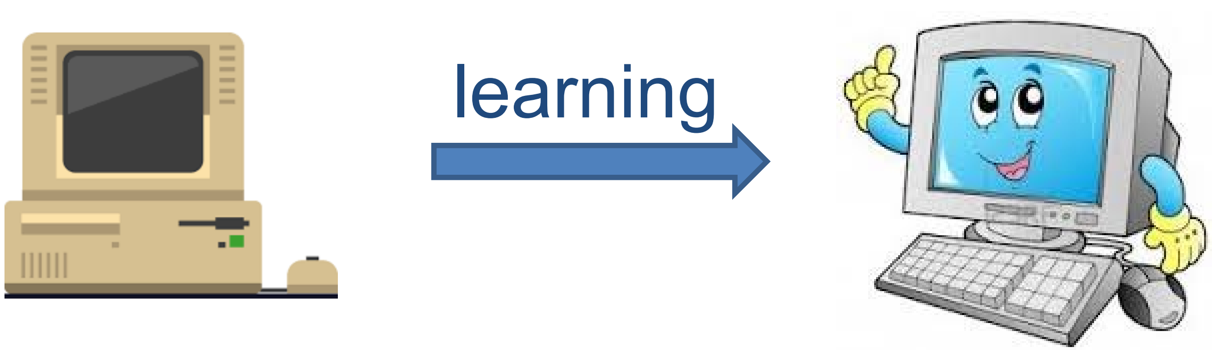

nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
CONSERVATION
SONGBIRDS À LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452
RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on 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

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

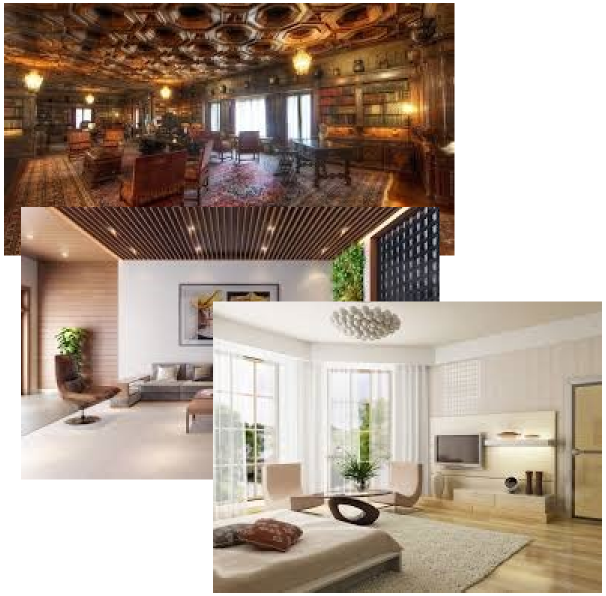


Example 1: image classification

Task: determine if the image is indoor or outdoor

Performance measure: probability of misclassification

Example 1: image classification

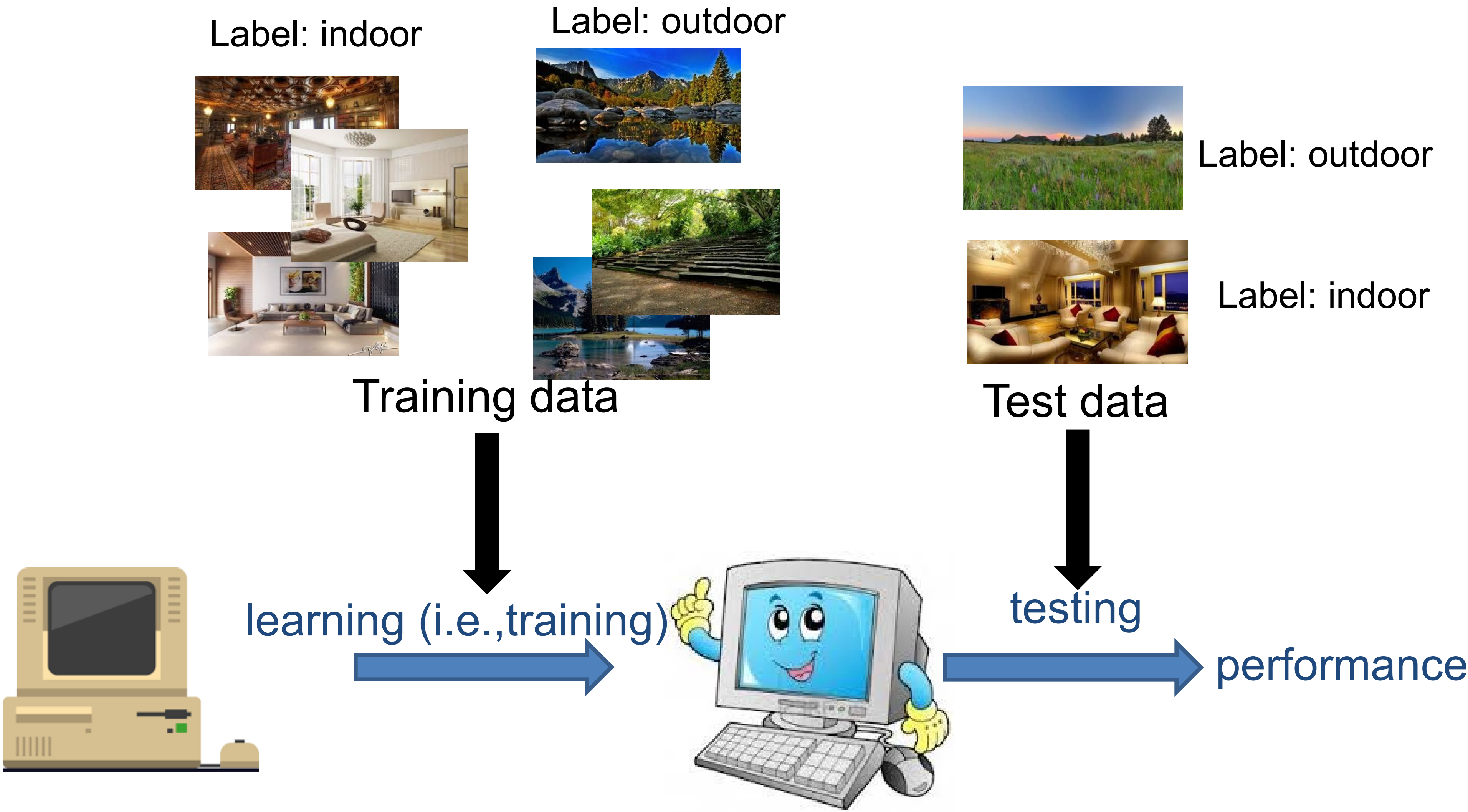


Experience/Data:
images with labels

indoor

outdoor

Example 1: image classification



Example 1: image classification

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

Example 2: clustering images

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

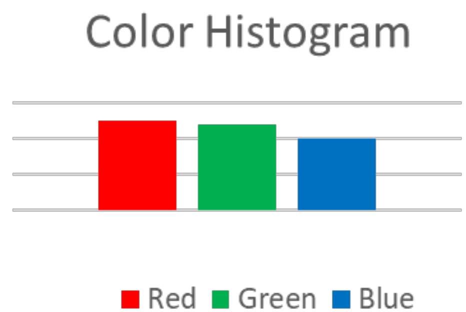
Example 2: clustering images

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

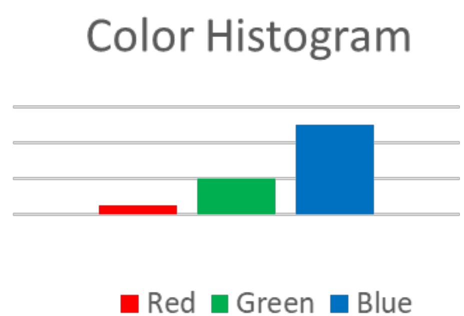
Feature vectors

Feature space

Extract features

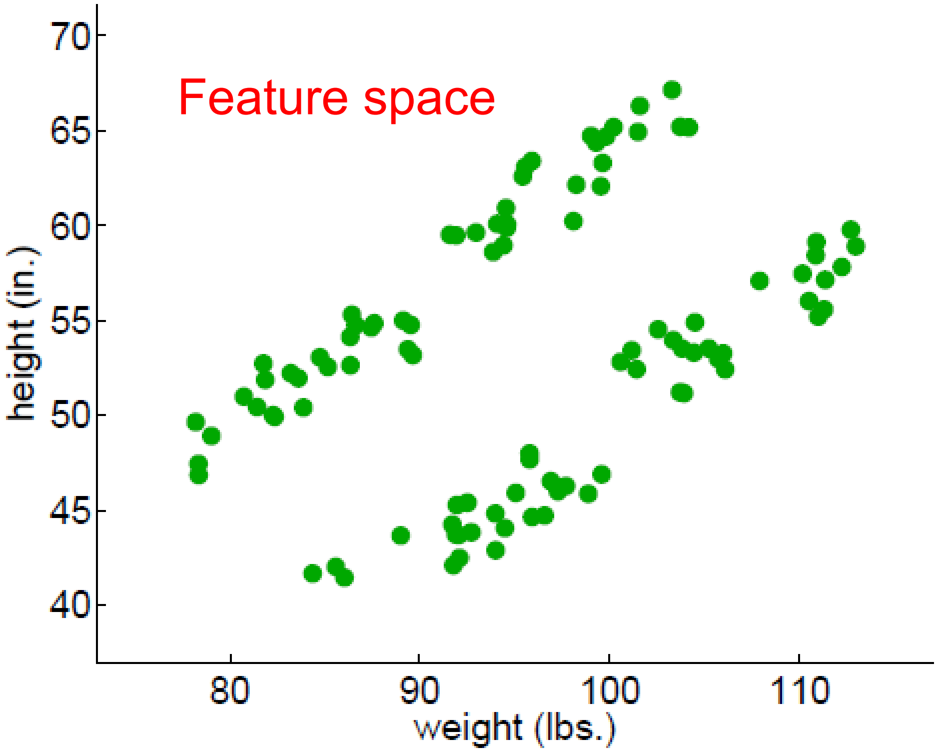


Extract features

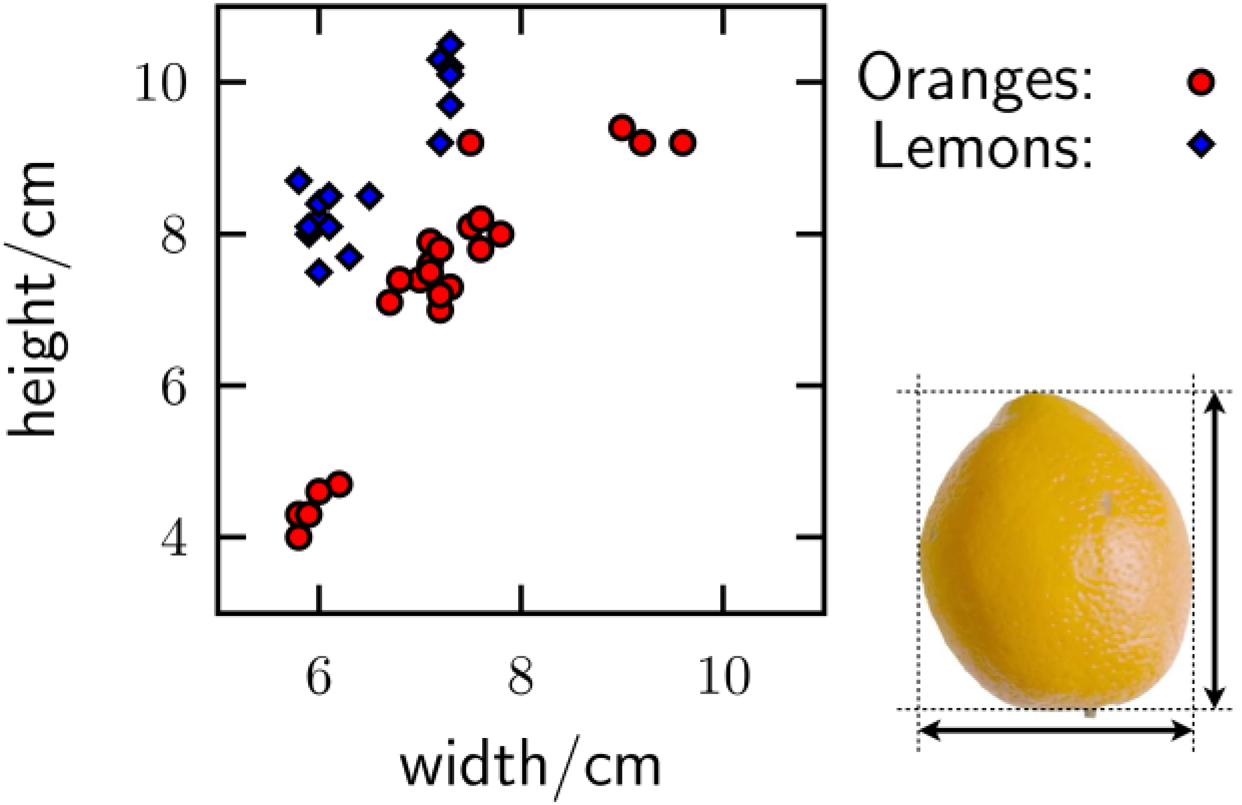


Feature Example 2: little green men

- 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 words": 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:

- novelty/anomaly 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
- clustering, separate the *m* instances into groups

Anomaly detection

training

given

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

output

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

E.g, h is function so that $h(\boldsymbol{x}) = 0$ for normal $\boldsymbol{x}$, and $h(\boldsymbol{x}) = 1$ otherwise.

test

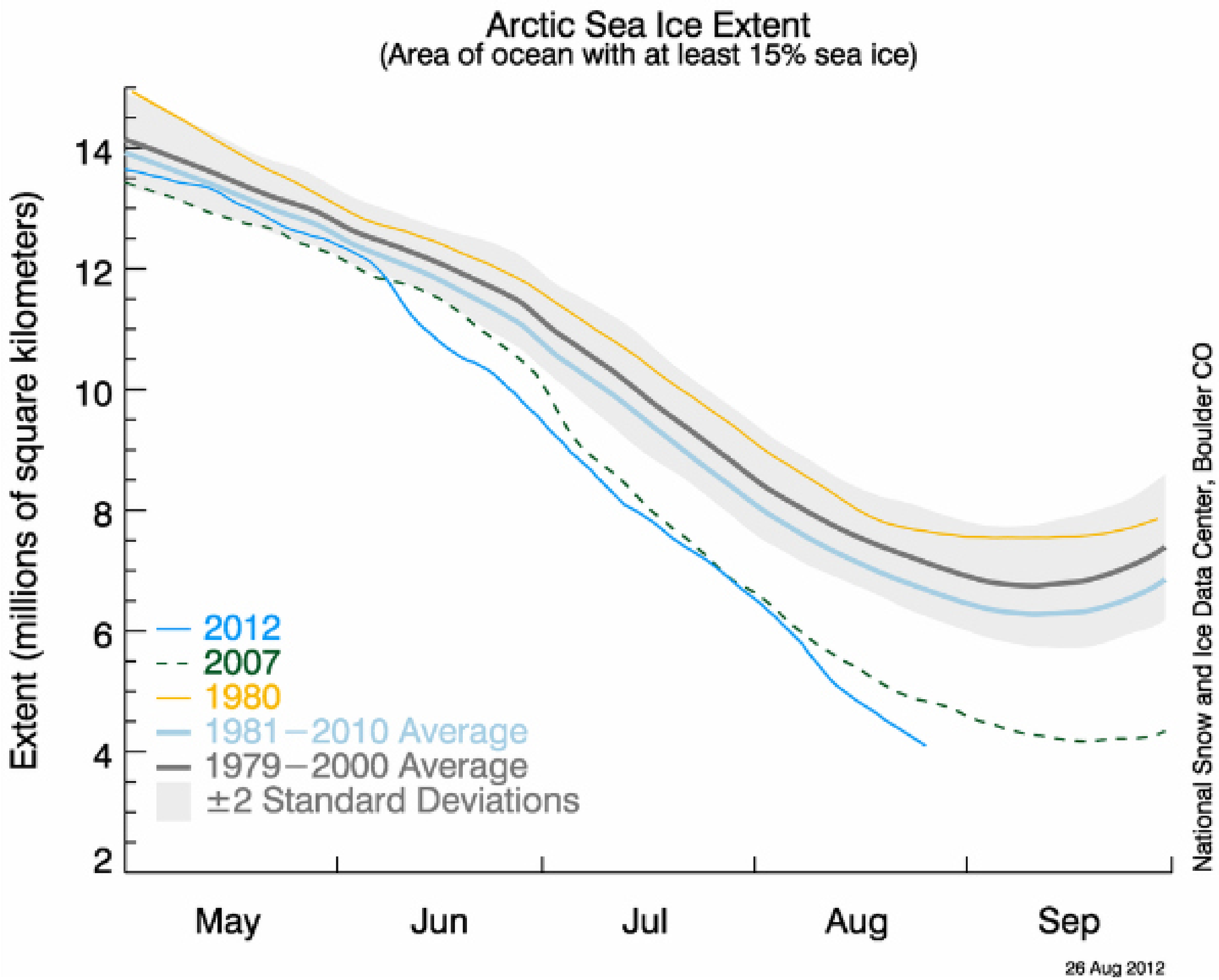
given

- 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

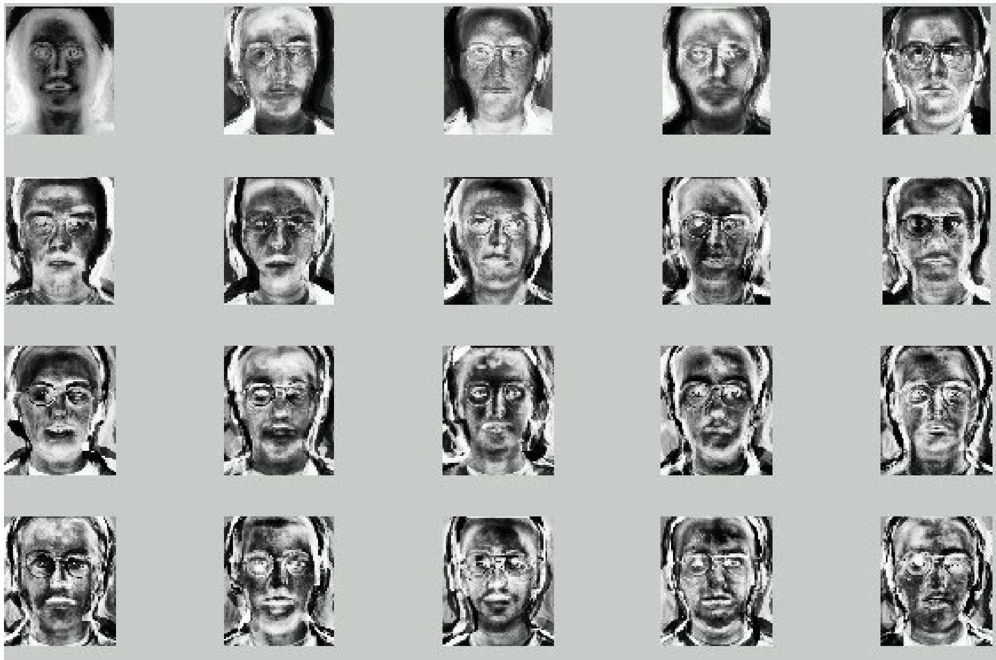
- model h that represents each $\boldsymbol{x}$ with a lower-dimension feature vector while still preserving key properties of the data

E.g, h is a function so that $h(\boldsymbol{x})$ is the new representation in lower dimension

Dimensionality reduction example: PCA

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

Here, h is a function so that
$h(\boldsymbol{x}) = [v_1^T\boldsymbol{x}, v_2^T\boldsymbol{x}, \dots, v_k^T\boldsymbol{x}]$
where v_i are the principle components



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

Clustering

given

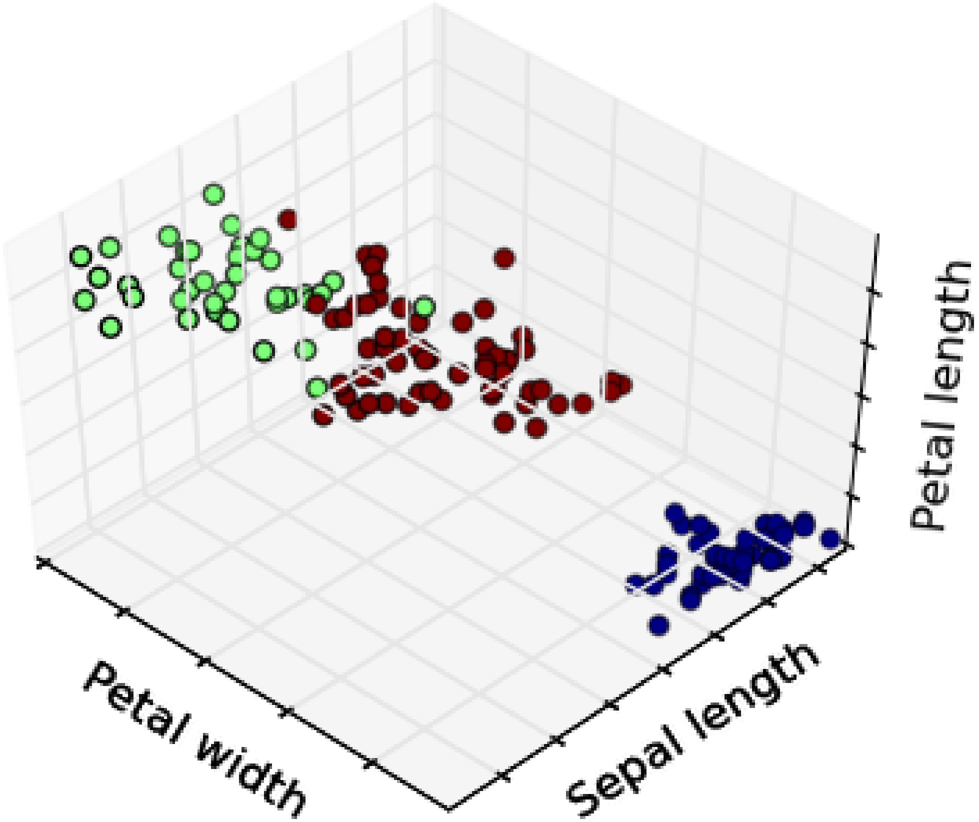
- training set of instances $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(m)}$

output

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

E.g.,, h is a function so that $i = h(\boldsymbol{x})$
means $\boldsymbol{x}$ belongs to the i -th cluster.

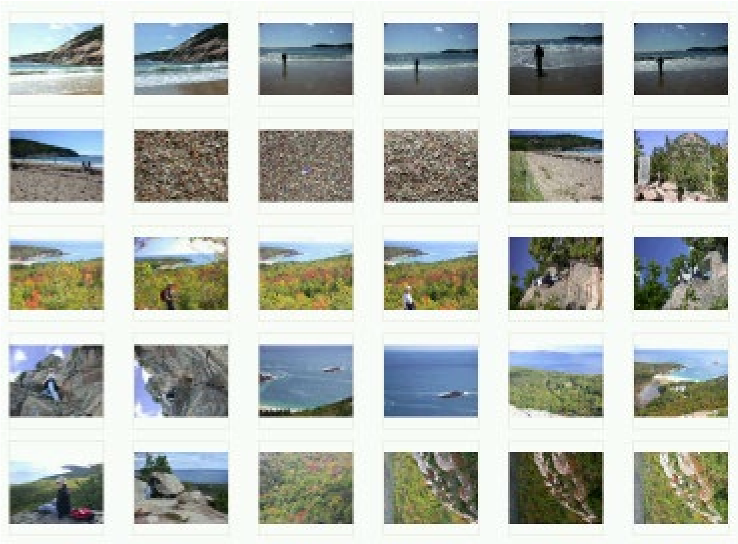
Example 1: Irises



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

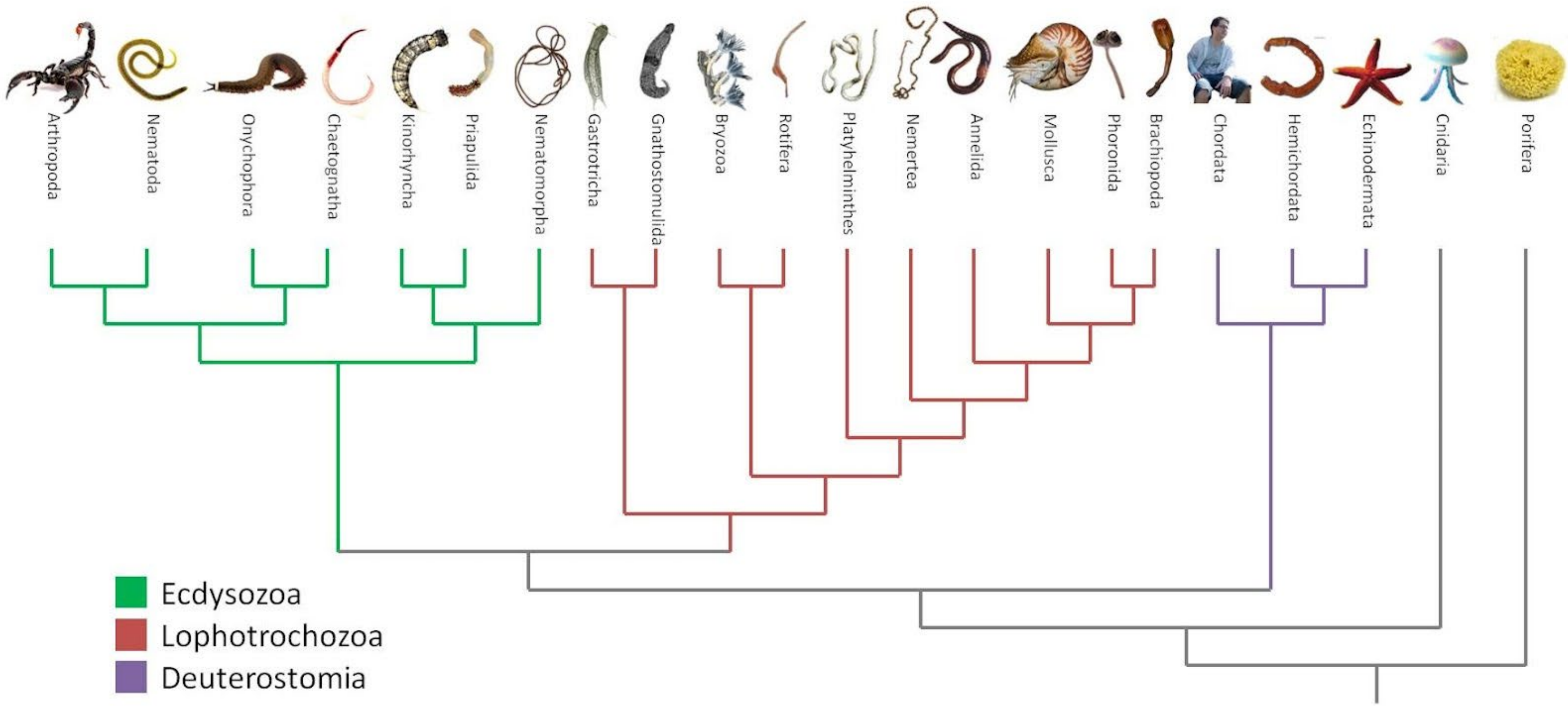
    Where we specify the desired number of clusters, and use an iterative algorithm to find them

HIERARCHICAL CLUSTERING

Hierarchical clustering

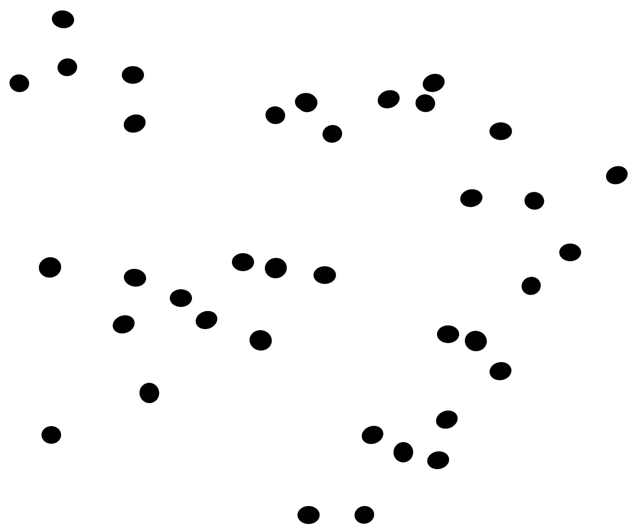
- Very popular clustering algorithm
- Input:
  - A dataset $x_1, \ldots, x_n$. each point is a feature vector
  - Does NOT need the number of clusters

Building a hierarchy



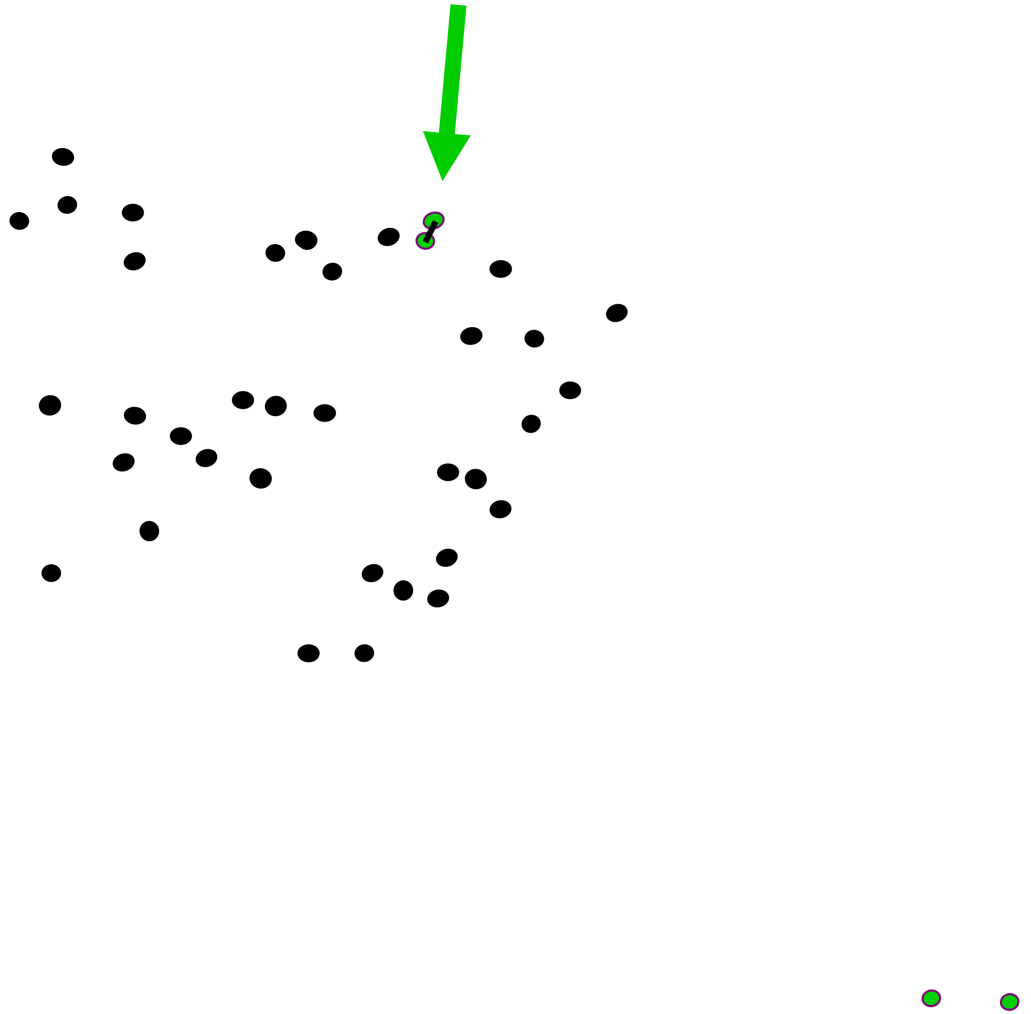
Hierarchical clustering

- Initially every point is in its own cluster



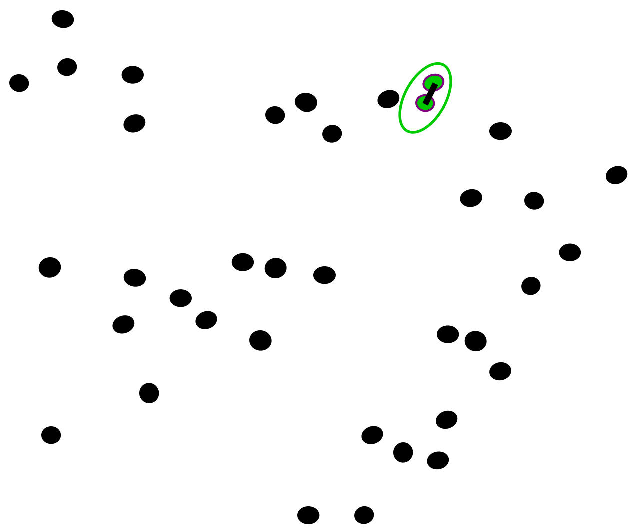
Hierarchical clustering

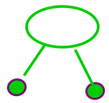
- Find the pair of clusters that are the closest



Hierarchical clustering

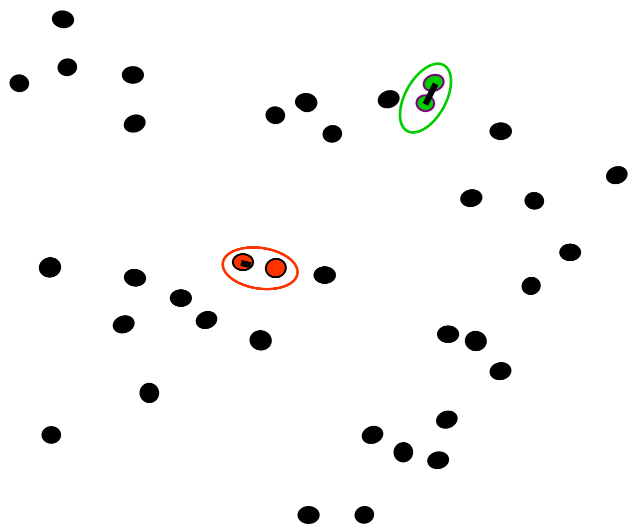
- Merge the two into a single cluster





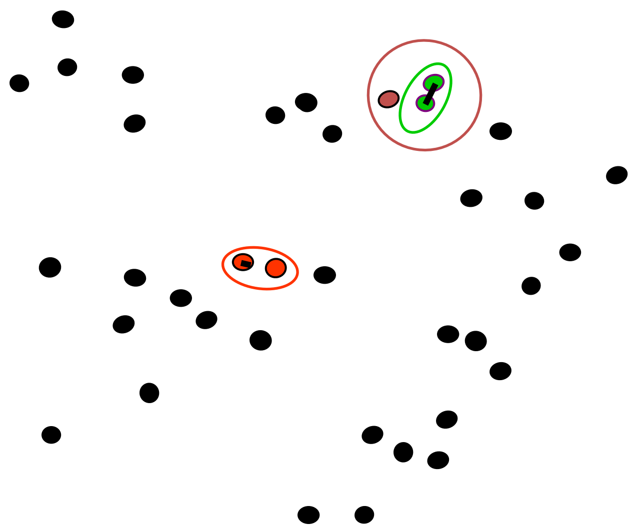
Hierarchical clustering

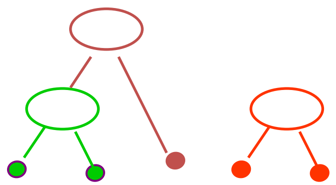
- Repeat…



Hierarchical clustering

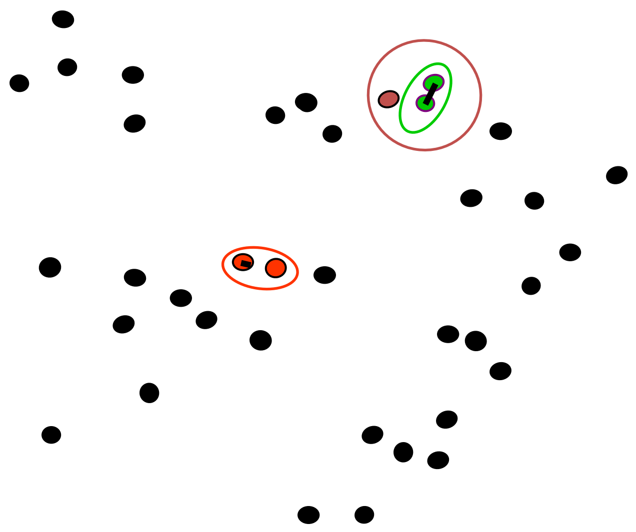
- Repeat…

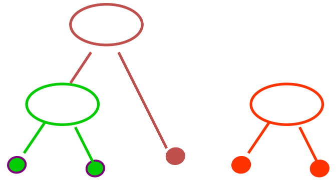




Hierarchical clustering

- 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

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

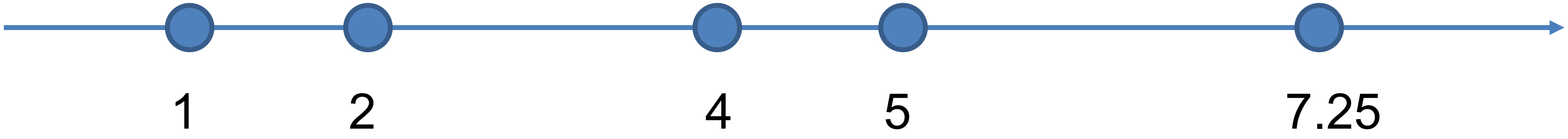
Hierarchical clustering

- How do you measure the closeness between two clusters?

Hierarchical clustering

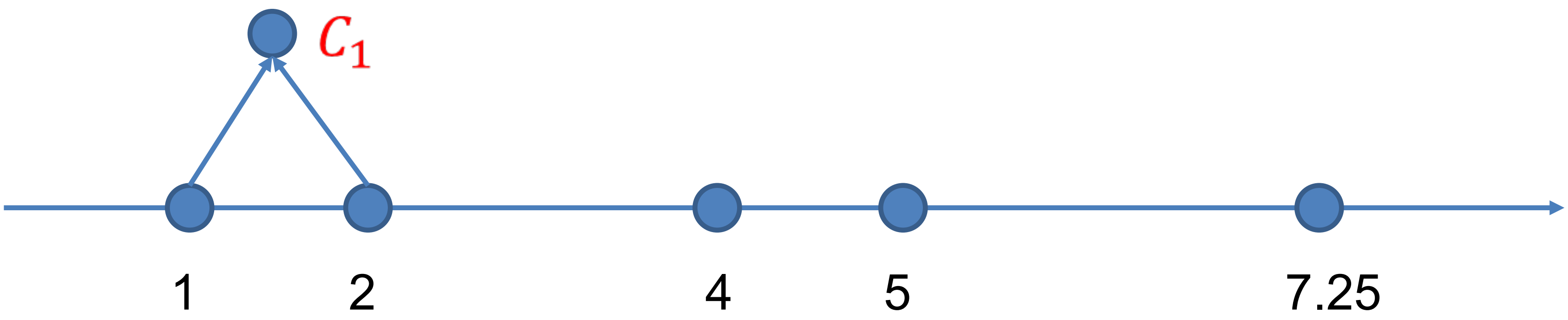
- How do you measure the closeness between two clusters? At least three ways:
  - Single-linkage: the shortest distance from any member of one cluster to any member of the other cluster. Formula? $d(A,B) = \min_{x \in A, y \in B} d(x,y)$
  - Complete-linkage: the greatest distance from any member of one cluster to any member of the other cluster
  - Average-linkage: you guess it!

Single-linkage



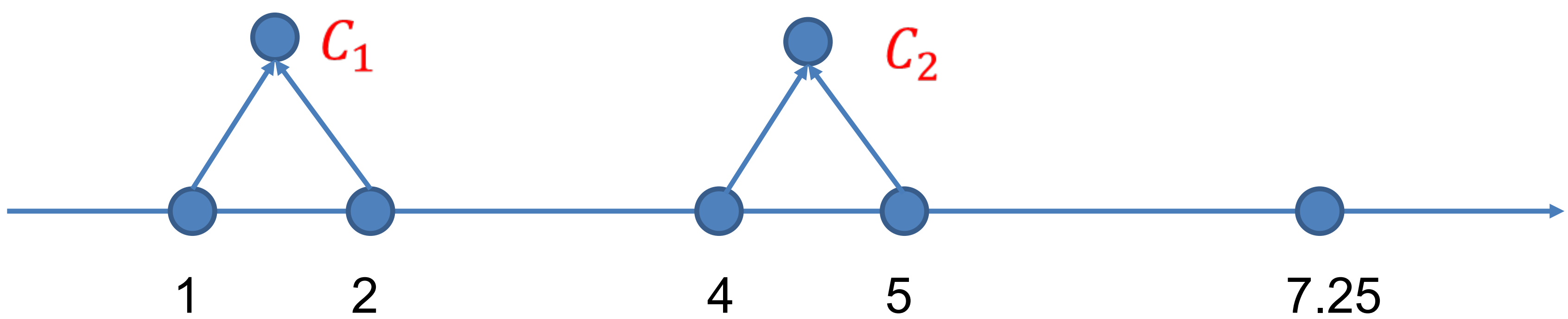
Single-linkage

$$d(C_1, \{4\}) = d(2,4) = 2$$
$$d(\{4\}, \{5\}) = d(4,5) = 1$$

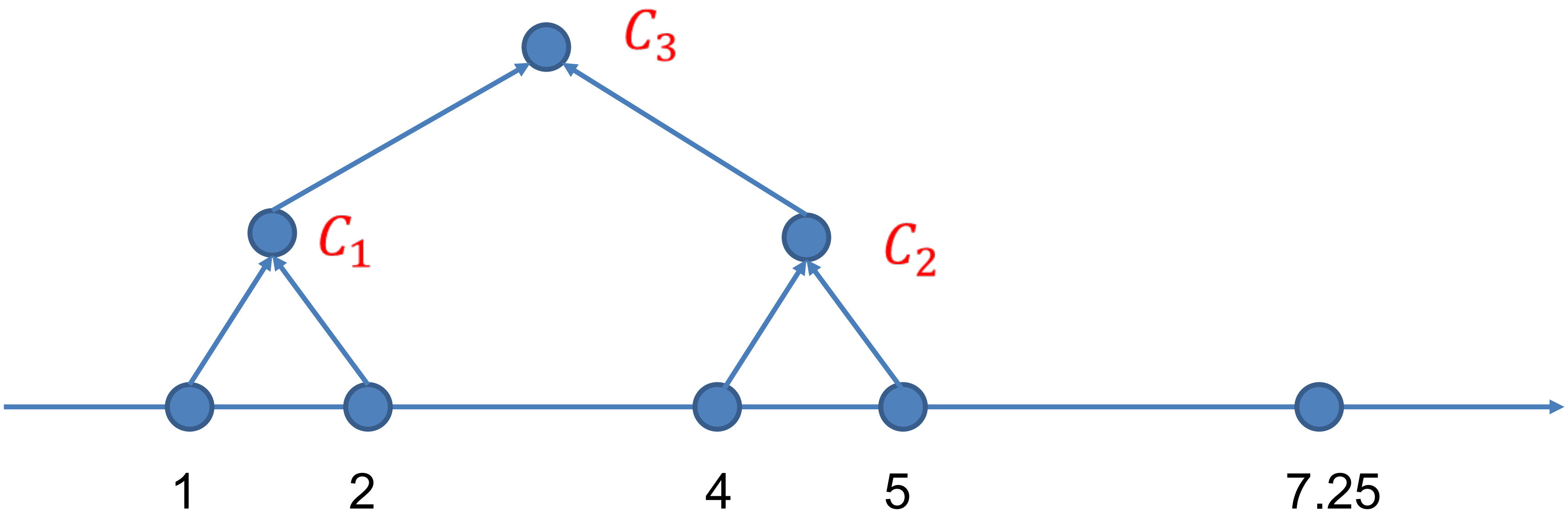


Single-linkage

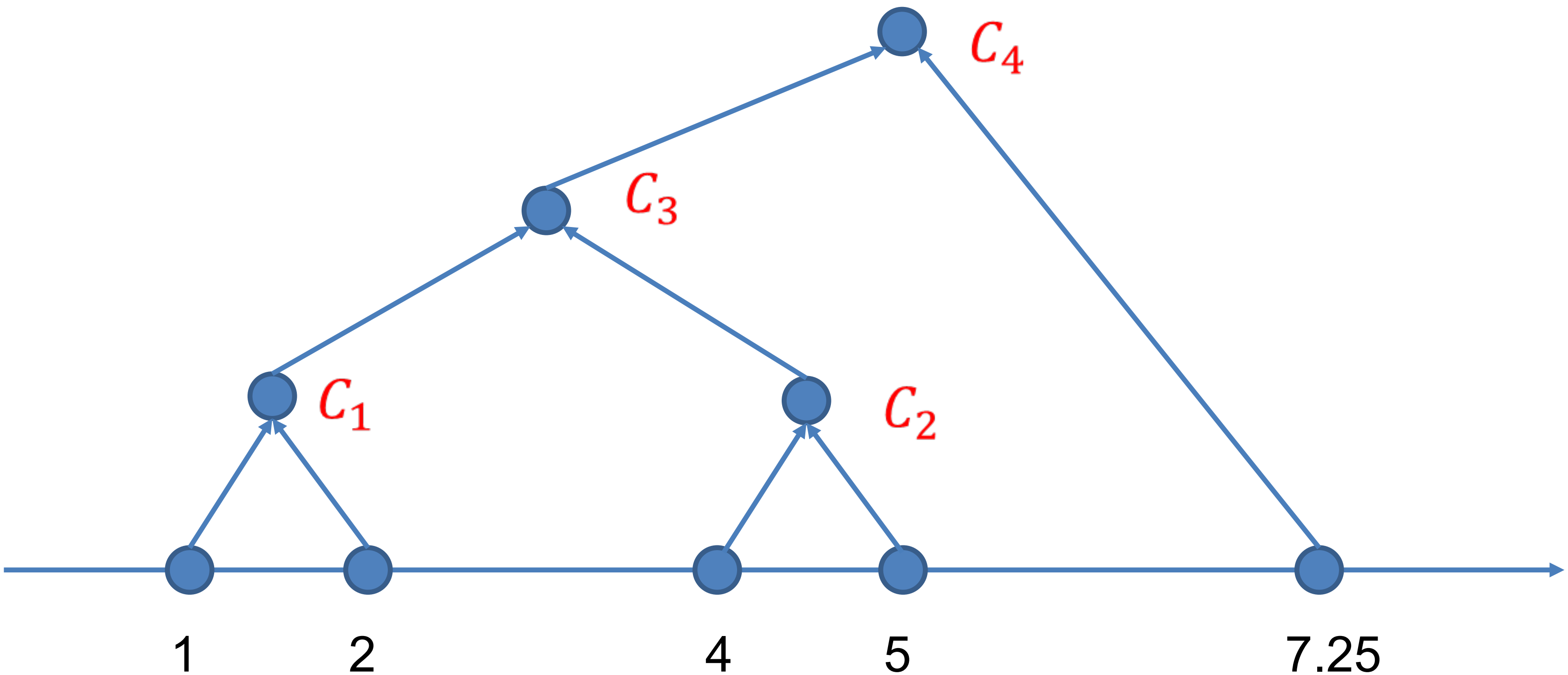
$$d(C_1, C_2) = d(2,4) = 2$$
$$d(C_2, \{7.25\}) = d(5,7.25) = 2.25$$



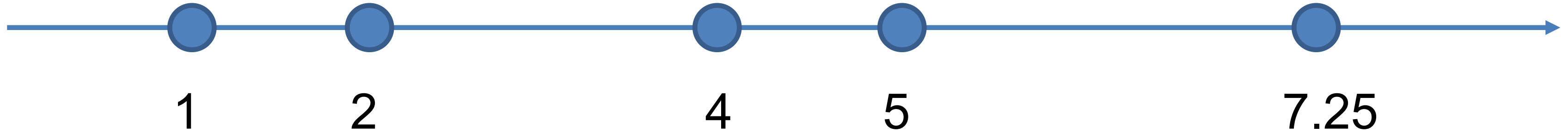
Single-linkage



Single-linkage

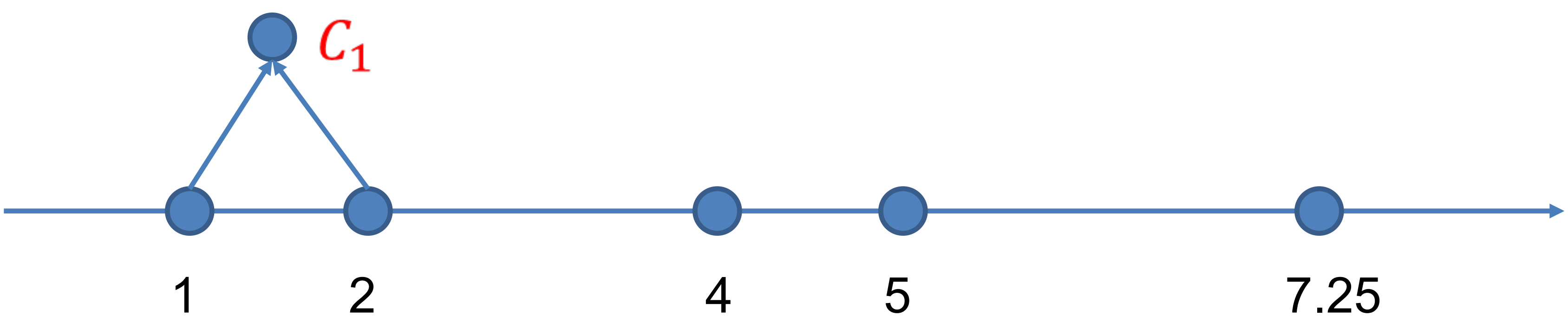


Complete-linkage



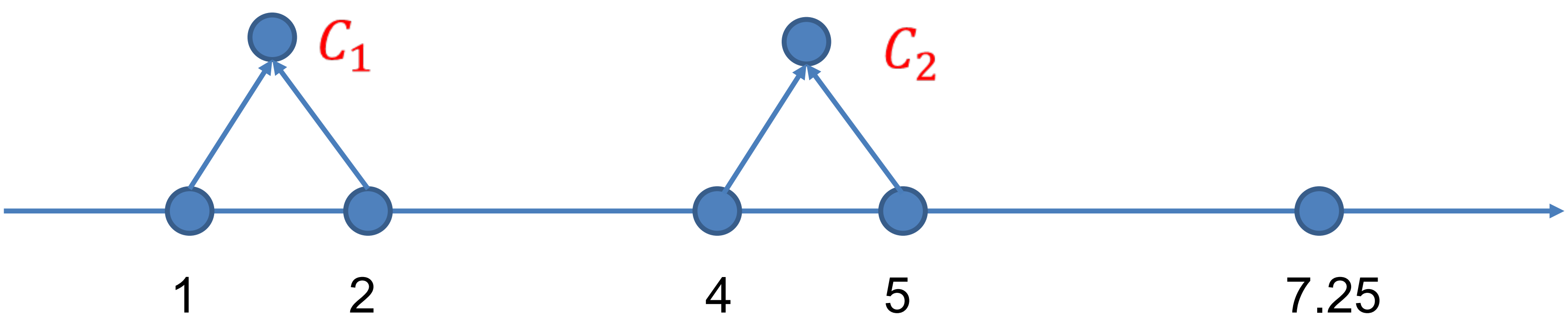
Complete-linkage

$$d(C_1, \{4\}) = d(1,4) = 3$$
$$d(\{4\}, \{5\}) = d(4,5) = 1$$

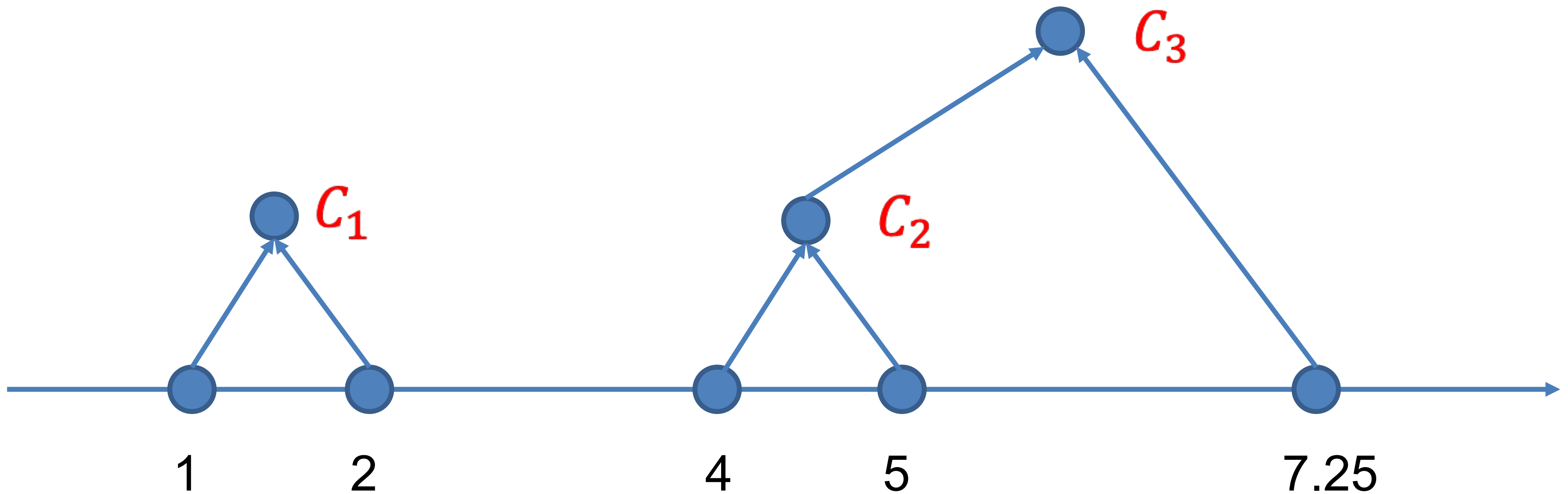


Complete-linkage

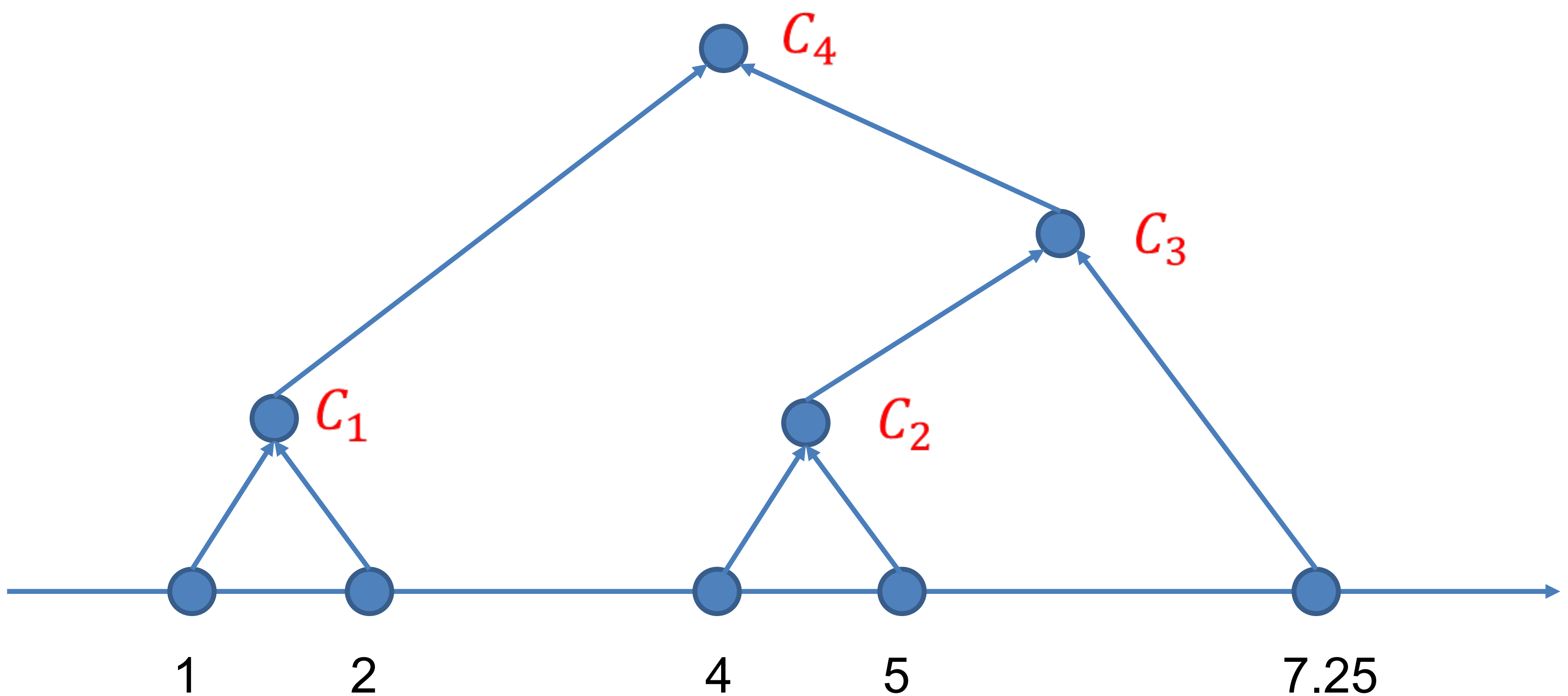
$$d(C_1, C_2) = d(1{,}5) = 4$$
$$d(C_2, \{7.25\}) = d(4{,}7.25) = 3.25$$



Complete-linkage



Complete-linkage



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…