SOICT

HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

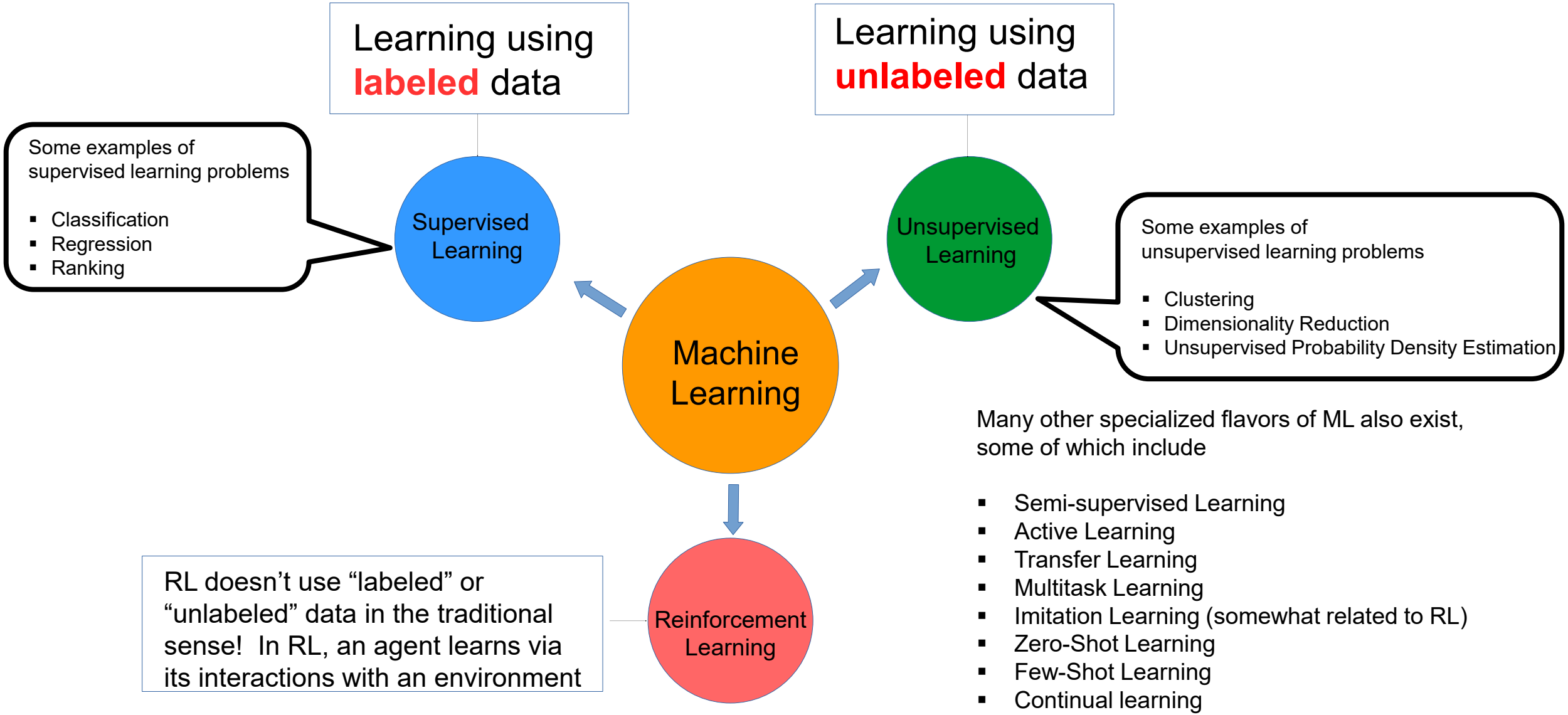
Working with Data and Features

Dam Quang Tuan

Plan for today

- Types of ML problems
- Typical workflow of ML problems
- Various perspectives of ML problems
- Data and Features
- Some basic operations of data and features
- People who need a math refresher can use this handy tutorial

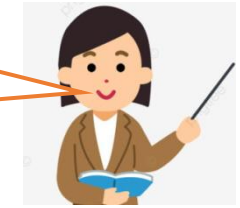
A Loose Taxonomy of ML



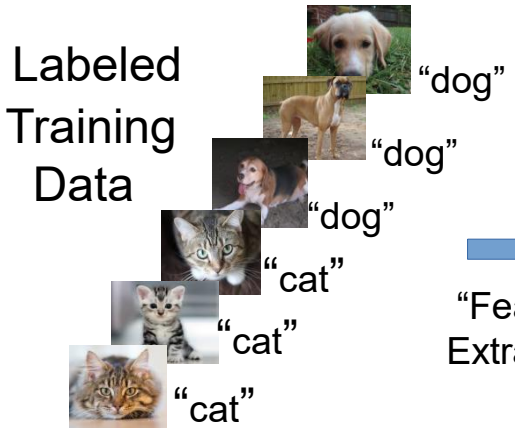
Many other specialized flavors of ML also exist, some of which include

- Semi-supervised Learning
- Active Learning
- Transfer Learning
- Multitask Learning
- Imitation Learning (somewhat related to RL)
- Zero-Shot Learning
- Few-Shot Learning
- Continual learning

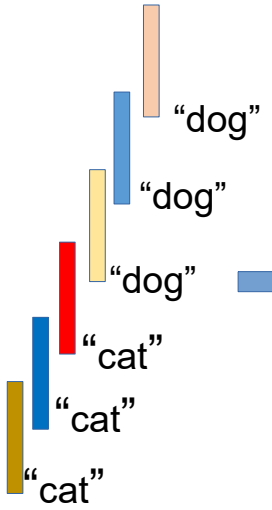
A Typical Supervised Learning Workflow

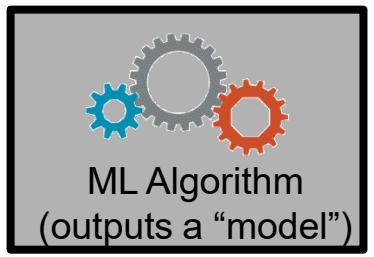


Note: This example is for the problem of binary classification, a supervised learning problem



"Feature" Extraction



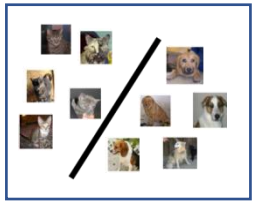


Can you think of a problem you would try to solve using supervised learning?

Feature extraction converts raw inputs to a numeric representation that the ML algo can understand and work with. More on feature extraction later.

Test Image

"Feature" Extraction



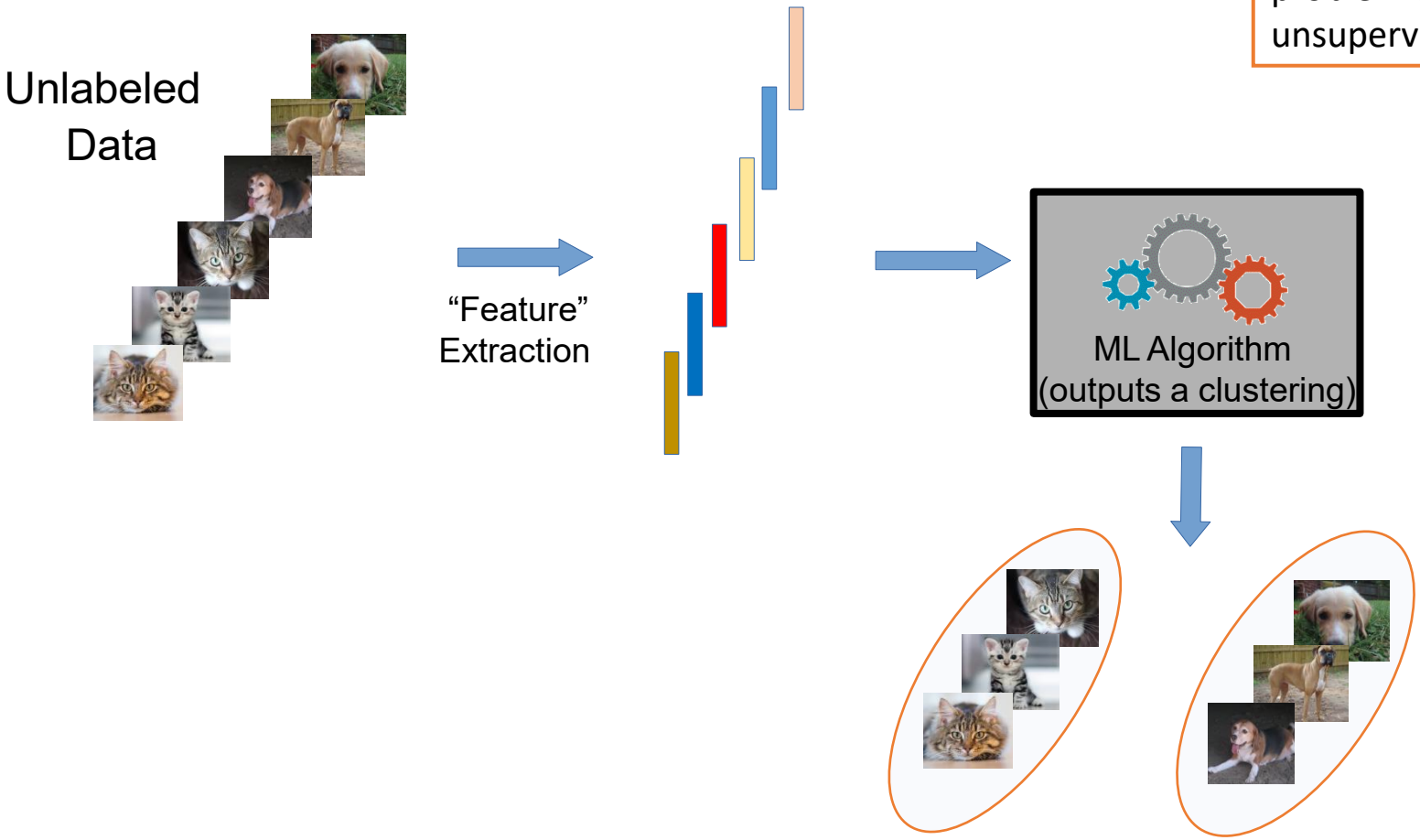
Cat vs Dog Prediction model

Predicted Label (cat/dog)

https://www.pinclipart.com/, http://www.pngtree.com

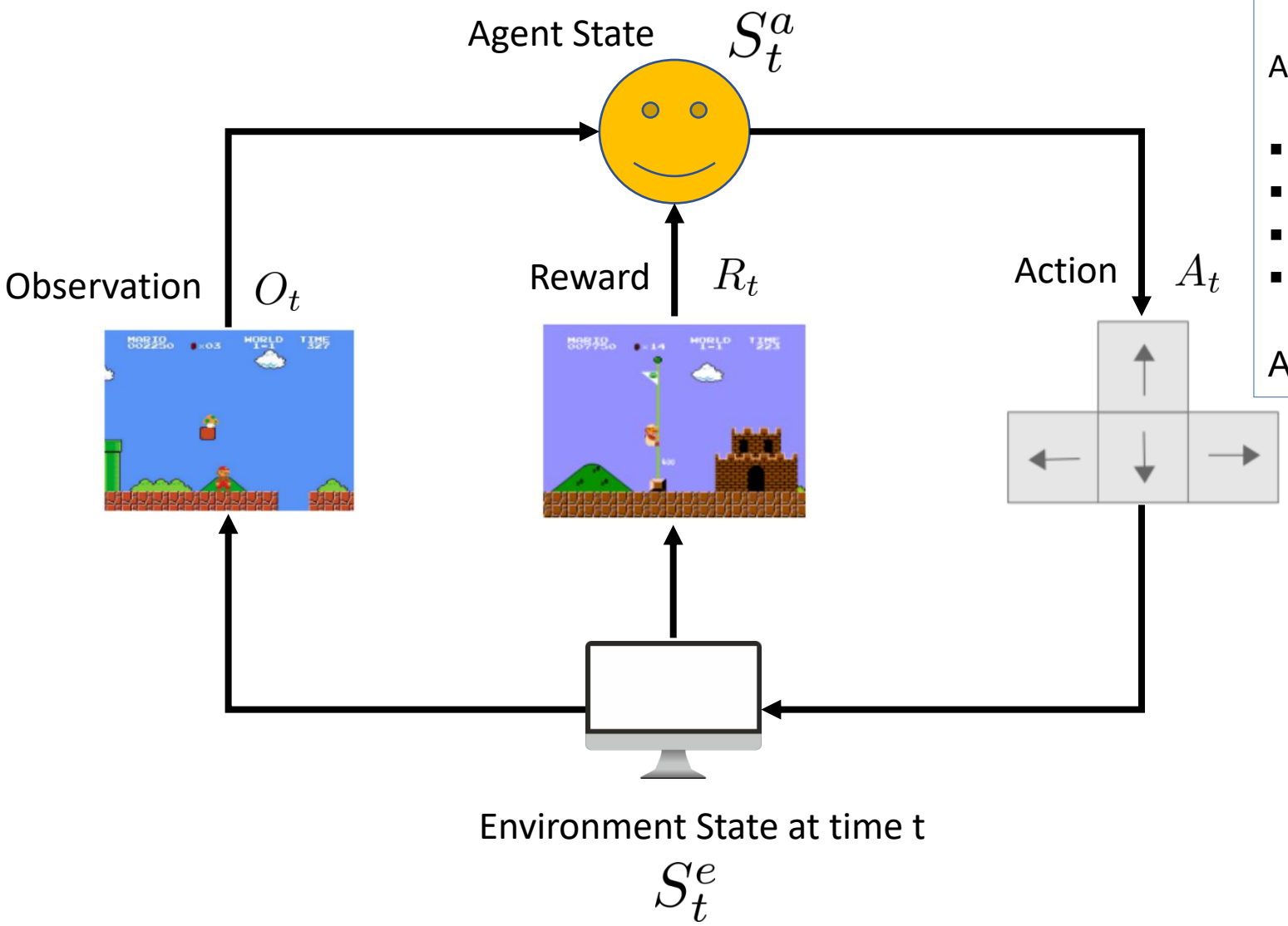
A Typical Unsupervised Learning Workflow

Note: This example is for the problem of data clustering, an unsupervised learning problem



https://www.pinclipart.com/, http://www.pngtree.com

A Typical Reinforcement Learning Workflow



Wish to teach an agent optimal policy for some task

Agent does the following repeatedly

- Senses/observes the environment
- Takes an action based on its current policy
- Receives a reward for that action
- Updates its policy

Agent's goal is to maximize its overall reward

There IS supervision, not explicit (as in Supervised Learning) but rather implicit (feedback based)

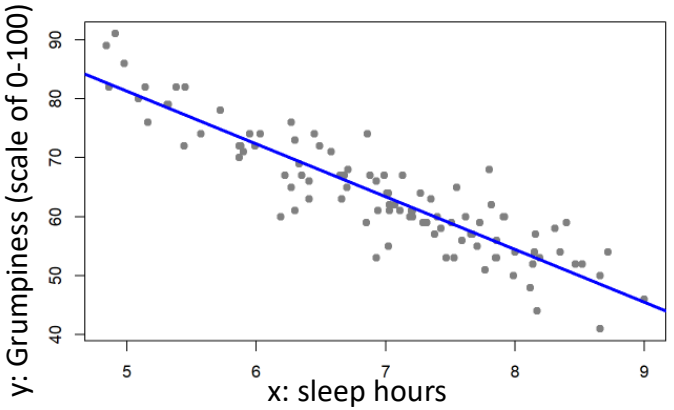
ML: Some Perspectives

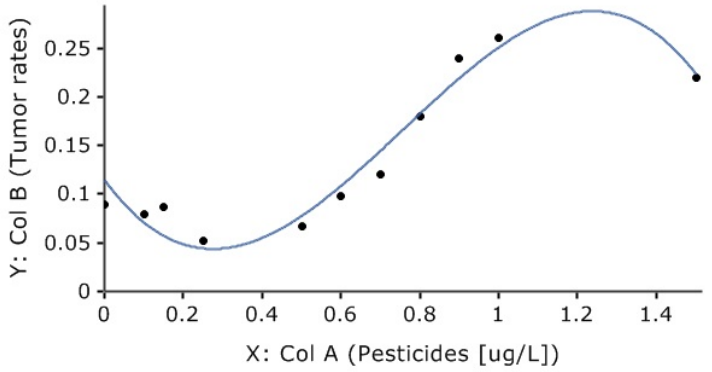
Geometric Perspective

Recall that feature extraction converts inputs into a numeric representation

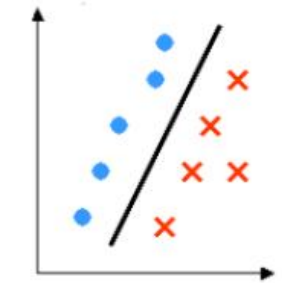
- Basic fact: Inputs in ML problems can often be represented as points or vectors in some vector space
- Doing ML on such data can thus be seen from a geometric view

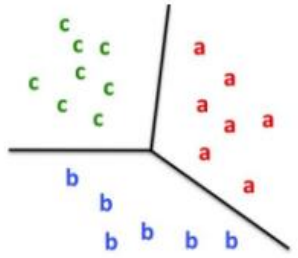
Regression: A supervised learning problem. Goal is to model the relationship between input (x) and real-valued output (y). This is akin to a line or curve fitting problem

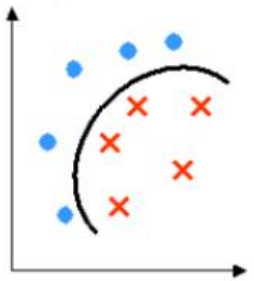


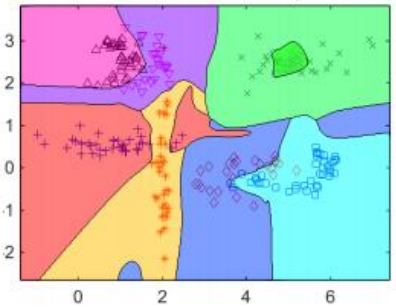


Classification: A supervised learning problem. Goal is to learn a to predict which of the two or more classes an input belongs to. Akin to learning linear/nonlinear separator for the inputs


Two-Class (binary) Linear Classification


Multi-Class Linear Classification

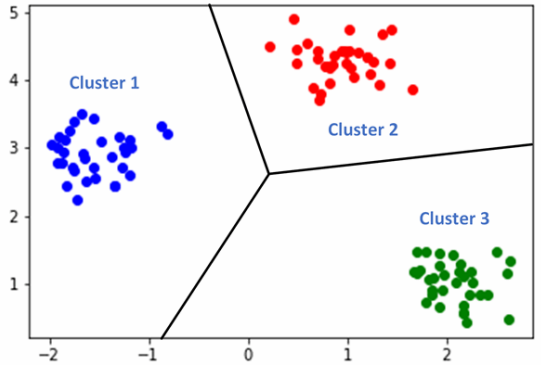

Two-Class (binary) Nonlinear Classification


Multi-Class Nonlinear Classification

Pic from: https://learningstatisticswithr.com/book/regression.html, https://maxstat.de/

Geometric Perspective

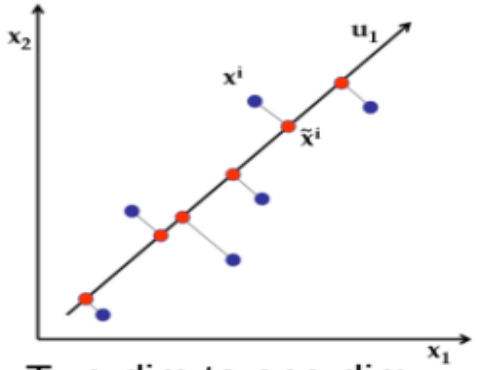
Clustering: An unsupervised learning problem. Goal is to group inputs in a few clusters based on their similarities with each other



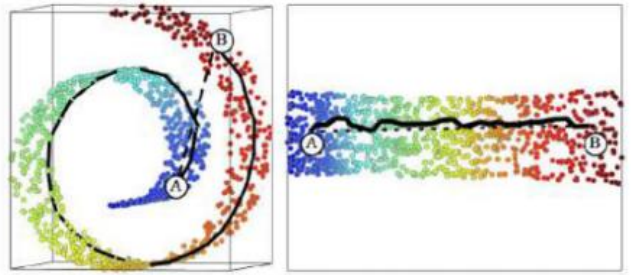
Clustering looks like classification to me. Is there any difference?

Yes. In clustering, we don't know the labels. Goal is to separate them without any labeled "supervision"

Dimensionality Reduction: An unsupervised learning problem. Goal is to compress the size of each input without losing much information present in the data



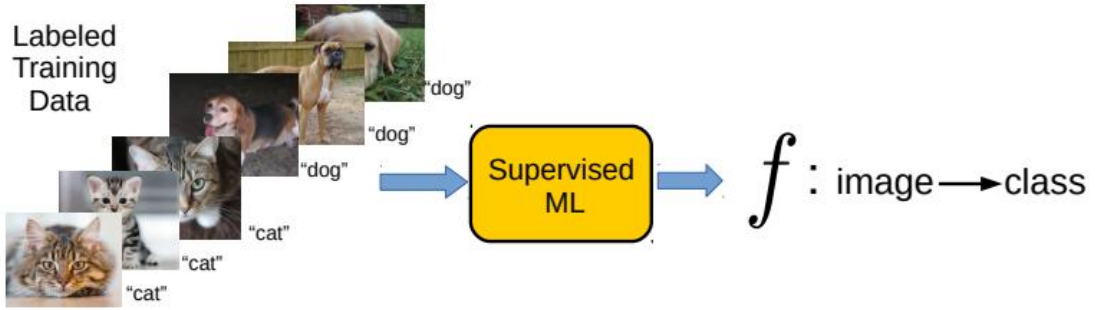
Two-dim to one-dim linear projection



Three-dim to two-dim nonlinear projection (a.k.a. manifold learning)

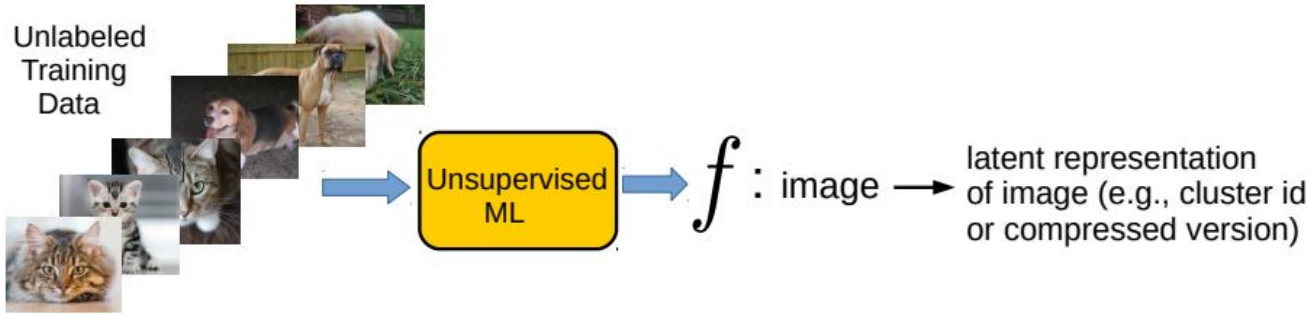
Perspective as function approximation

- Supervised Learning ("predict output given input") can be usually thought of as learning a function *f* that maps each input to the corresponding output



- Unsupervised Learning ("model/compress inputs") can also be usually thought of as learning a function *f* that maps each input to a compact representation

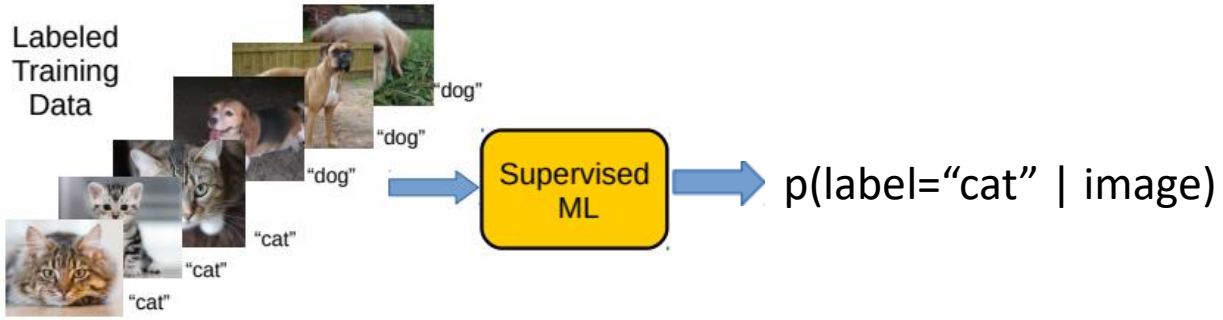
Harder since we don't know the labels in this case



- Reinforcement Learning can also be seen as doing function approximation

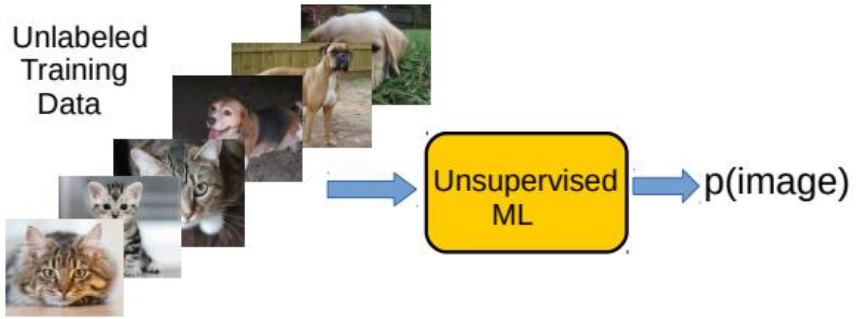
Perspective as probability estimation

- Supervised Learning ("predict output given input") can be thought of as estimating the conditional probability of each possible output given an input



- Unsupervised Learning ("model/compress inputs") can be thought of as estimating the probability density of the inputs

Harder since we don't know the labels in this case



- Reinforcement Learning can also be seen as estimating probability densities

Data and Features

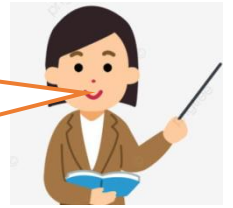
Data and Features

- ML algos require a numeric feature representation of the inputs
- Features can be obtained using one of the two approaches
  - Approach 1: Extracting/constructing features manually from raw inputs
  - Approach 2: Learning the features from raw inputs
- Approach 1 is what we will focus on primarily for now
- Approach 2 is what is followed in Deep Learning algorithms (will see later)
- Approach 1 is not as powerful as Approach 2 but still used widely

Example: Feature Extraction for Text Data

- Consider some text data consisting of the following sentences:
  - John likes to watch movies
  - Mary likes movies too
  - John also likes football
- Want to construct a feature representation for these sentences
- Here is a "bag-of-words" (BoW) feature representation of these sentences

BoW is just one of the many ways of doing feature extraction for text data. Not the most optimal one, and has various flaws (can you think of some?), but often works reasonably well

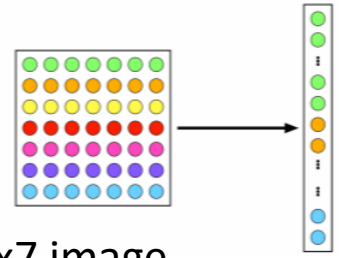


	John	likes	to	watch	movies	Mary	too	also	football
Sentence 1	1	1	1	1	1	0	0	0	0
Sentence 2	0	1	0	0	1	1	1	0	0
Sentence 3	1	1	0	0	0	0	0	1	1

- Each sentence is now represented as a binary vector (each feature is a binary value, denoting presence or absence of a word). BoW is also called "unigram" rep.

Example: Feature Extraction for Image Data

- A very simple feature extraction approach for image data is flattening

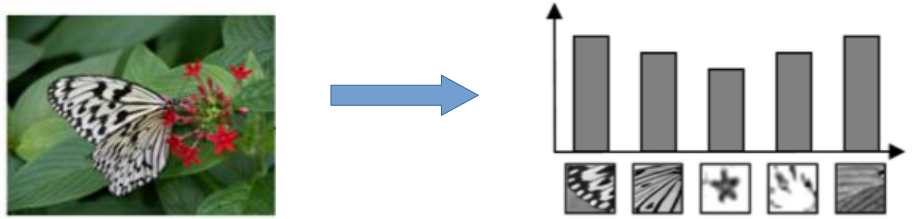


7x7 image (49 pixels)

Vector of pixel intensities

Flattening and histogram based methods destroy the spatial information in the image but often still work reasonably well

- Histogram of visual patterns is another popular feature extr. method for images



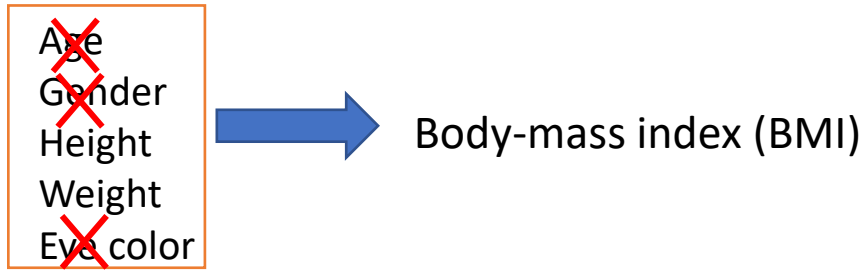
- Many other manual feature extraction techniques developed in computer vision and image processing communities (SIFT, HoG, and others)

Pic credit: cat.uab.cat/Research/object-recognition

Feature Selection

- Not all the extracted features may be relevant for learning the model (some may even confuse the learner)

- Feature selection (a step after feature extraction) can be used to identify the features that matter, and discard the others, for more effective learning

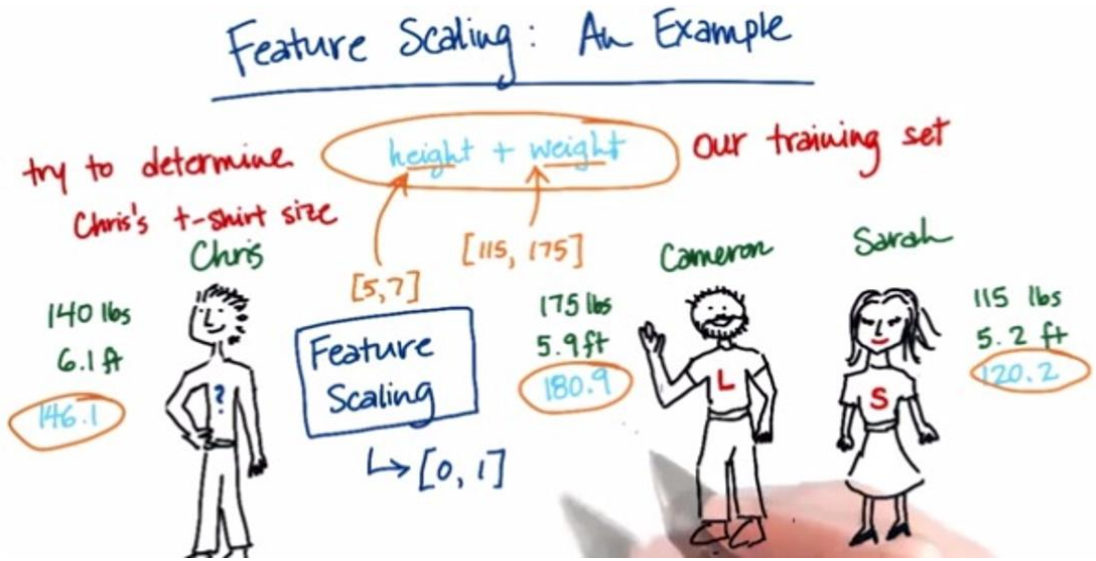


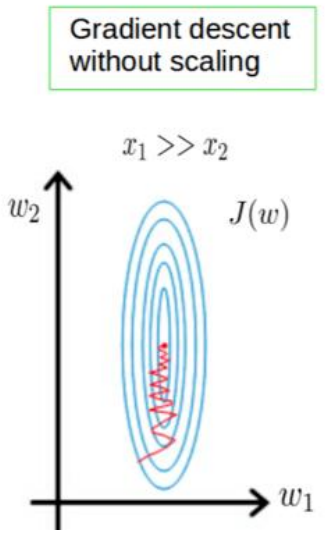
- Many techniques exist – some based on intuition, some based on algorithmic principles (will visit feature selection later)

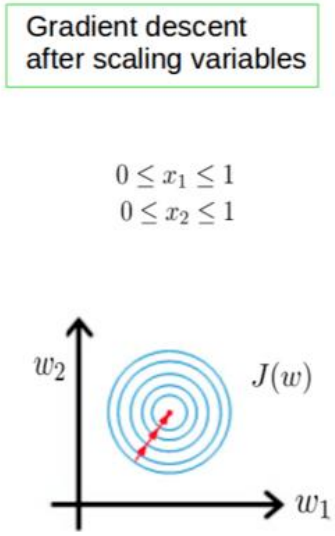
- More common in supervised learning but can also be done for unsup. learning

Some More Postprocessing: Feature Scaling

- Even after feature selection, the features may not be on the same scale
- This can be problematic when comparing two inputs – features that have larger scales may dominate the result of such comparisons
- Therefore helpful to standardize the features (e.g., by bringing all of them on the same scale such as between 0 to 1)







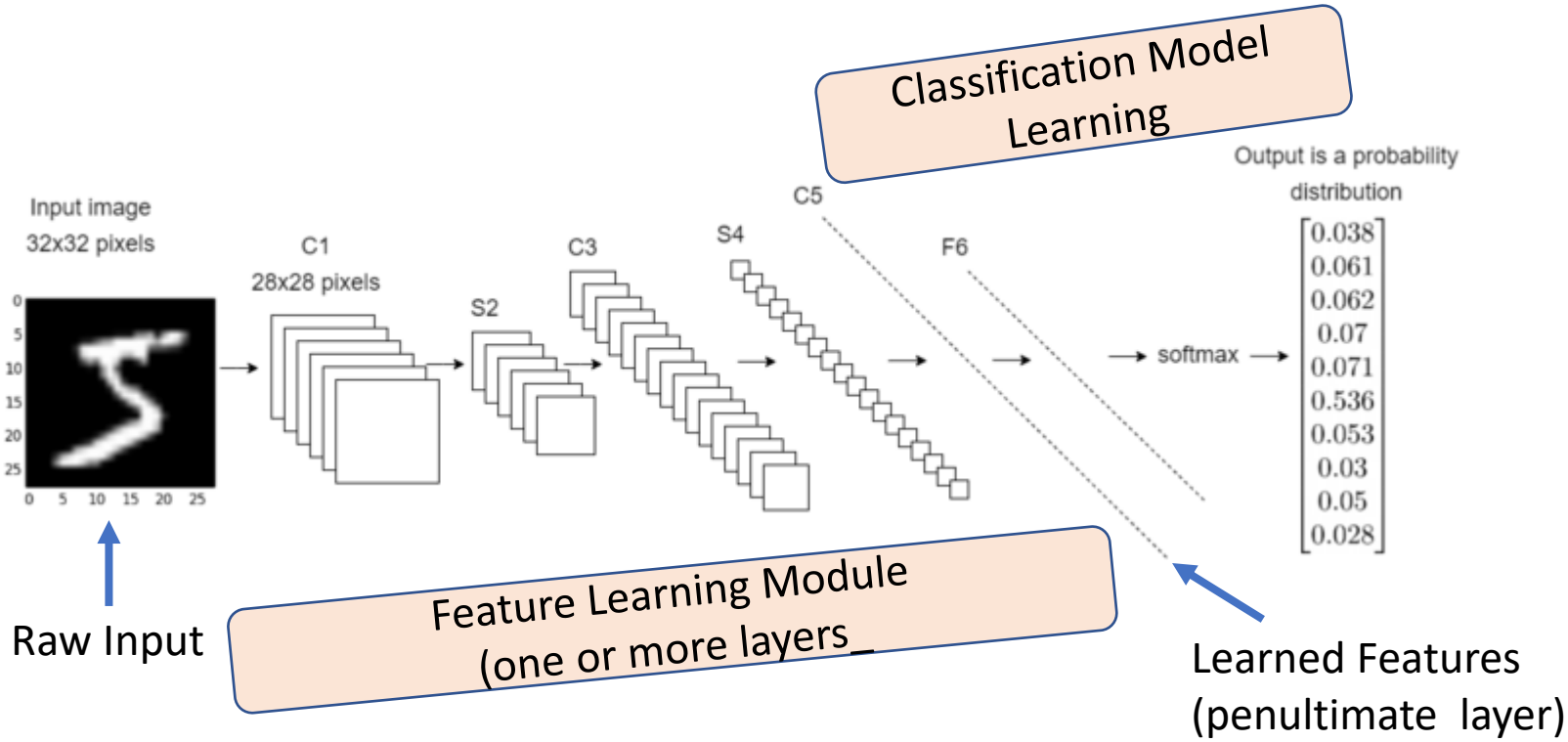
- Also helpful for stabilizing the optimization techniques used in ML algos

Pic credit: https://becominghuman.ai/demystifying-feature-scaling-baff53e9b3fd, https://stackoverflow.com/

Deep Learning: An End-to-End Approach to ML

Deep Learning = ML with automated feature learning from the raw inputs

Feature extraction part is automated via the feature learning module



Pic an adaptation of the original from: https://deepai.org/

Some Notation/Nomenclature/Convention

- Sup. learning requires training data as N input-output pairs $\{(\mathbf{x_n}, y_n)\}_{n=1}^{N}$
- Unsupervised learning requires training data as N inputs $\{\mathbf{x_n}\}_{n=1}^{N}$

RL and other flavors of ML problems also use similar notation

- Each input $\mathbf{x_n}$ is (usually) a vector containing the values of the features or attributes or covariates that encode properties of the it represents, e.g.,
  - For a 7 × 7 image: $\mathbf{x_n}$ can be a 49 × 1 vector of pixel intensities

Size or length of the input $\mathbf{x_n}$ is commonly known as data/input dimensionality or feature dimensionality

- (In sup. Learning) Each y_n is the output or response or label associated with input $\mathbf{x_n}$ (and its value is known for the training inputs)
  - Output can be a scalar, a vector of numbers, or even an structured object (more on this later)

Types of Features and Types of Outputs

- Features as well as outputs can be real-valued, binary, categorical, ordinal, etc.
- Real-valued: Pixel intensity, house area, house price, rainfall amount, temperature, etc
- Binary: Male/female, adult/non-adult, or any yes/no or present/absent type value
- Categorical/Discrete: Zipcode, blood-group, or any “one from a finite many choices“ value
- Ordinal: Grade (A/B/C etc.) in a course, or any other type where relative values matter
- Often, the features can be of mixed types (some real, some categorical, some ordinal, etc.)

Some Basic Operations of Inputs

- Assume each input feature vector $\mathbf{x}_n \in R^D$ to of size D

- Given N inputs $\{\mathbf{x}_n\}_{n=1,}^{N}$ their average or mean can be computed as

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n$$

What does such a "mean" represent?

If inputs are all cat images, mean vector would represents what an "average" cat looks like

- Can compute the Euclidean distance between any pair of inputs $\mathbf{x}_n$ and $\mathbf{x}_m$

$$d(\boldsymbol{x}_n, \boldsymbol{x}_m) = ||\boldsymbol{x}_n - \boldsymbol{x}_m|| = \sqrt{(\boldsymbol{x}_n - \boldsymbol{x}_m)^\top (\boldsymbol{x}_n - \boldsymbol{x}_m)} = \sqrt{\sum_{d=1}^{D} (x_{nd} - x_{md})^2}$$

- .. or Euclidean distance between an input $\mathbf{x}_n$ and the mean $\boldsymbol{\mu}$ of all inputs

- .. and various other operations that we will look at later..

Next Class

- Introduction to Supervised Learning

- A simple Supervised Learning algorithm based on computing distances

References

CS771: Intro to Machine Learning (Fall 2021), Nisheeth Srivastava, IIT Kanpur