Learning in Intelligent Systems

Artificial Intelligence @ Allegheny College

Janyl Jumadinova

September 13-17, 2021

Overview of Learning

Learning in Humans

- The act / process of acquiring, modify or reinforcing knowledge or skills through synthesizing different types of new or existed information.

Learning in Humans

- The act / process of acquiring, modify or reinforcing knowledge or skills through synthesizing different types of new or existed information.
- Key to human survival.

Learning in Humans

- The act / process of acquiring, modify or reinforcing knowledge or skills through synthesizing different types of new or existed information.
- Key to human survival.
- Progress over time tends to follow learning curves (relatively permanent).

Learning in Computing Systems

- Computational methods using "experience" to improve performance.

Learning in Computing Systems

- Computational methods using "experience" to improve performance.
- Experience – data driven task.

Learning in Computing Systems

- Computational methods using “experience” to improve performance.
- Experience – data driven task.
- Computer science – involves learning algorithms, analysis of complexity, and theoretical guarantees.

Learning in Computing Systems

Artificial intelligence | Machine learning

Learning in Computing Systems

Artificial intelligence | Machine learning

- Computer program(s) with adaptive mechanisms that enable computer / machine to learn from experience /example / analogy / rewards.

Learning in Computing Systems

Artificial intelligence | Machine learning

- Computer program(s) with adaptive mechanisms that enable computer / machine to learn from experience /example / analogy / rewards.
- It improves the performance of an intelligent system over time (e.g, reducing error rate, improving rewards).

Why Learning in Computing Systems?

- Understand and improve efficiency of human learning / understanding.

Why Learning in Computing Systems?

- Understand and improve efficiency of human learning / understanding.
- Discover new things or structure that is unknown to humans.

Why Learning in Computing Systems?

- Understand and improve efficiency of human learning / understanding.
- Discover new things or structure that is unknown to humans.
- Fill in skeletal or incomplete knowledge / expert specifications about a domain.

Applications of Learning

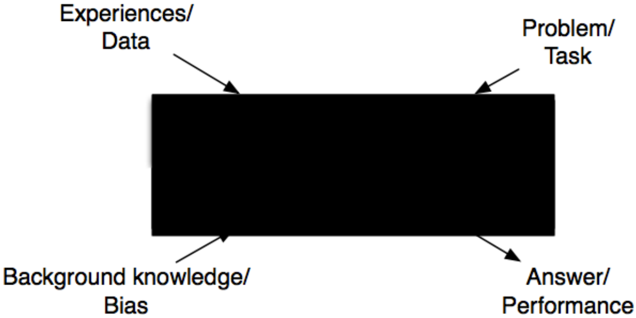
Mainly in decision making / pattern recognition / intelligent systems.

Applications of Learning

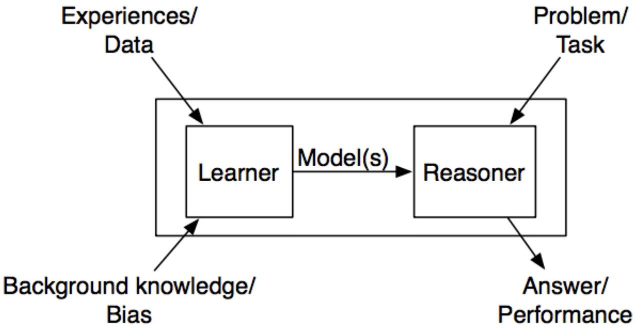
Mainly in decision making / pattern recognition / intelligent systems.

- Robot navigation.
- Automatic speech recognition (Siri in iPhone, Google speech-to-text search).
- Search and recommendation (Google, Amazon, eBay).
- Financial prediction, fraud detection, medical diagnosis.
- Video games, data visualization.

Black-box Learning



Learning Architecture



Learning Paradigms

- **Supervised learning**
  - input-output relationships

Learning Paradigms

- **Supervised learning**
  - input-output relationships
- **Unsupervised learning**
  - relationship among inputs

Learning Paradigms

- **Supervised learning**
  - input-output relationships
- **Unsupervised learning**
  - relationship among inputs
- **Reinforcement learning**
  - input-action relates to rewards / punishment

Supervised Learning

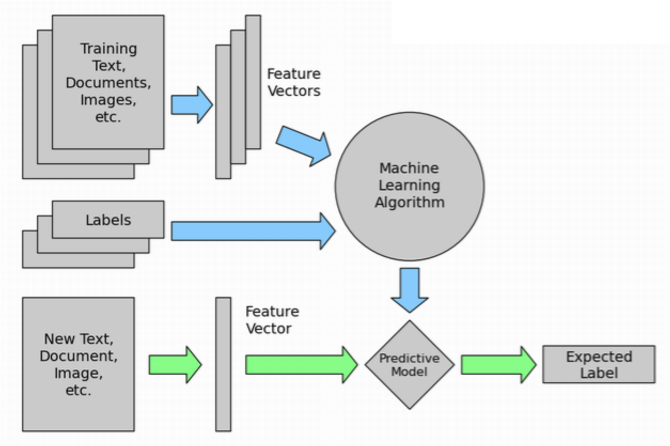
Given examples of inputs and corresponding desired outputs.

Supervised Learning

Given examples of inputs and corresponding desired outputs.
Tasks:

- Classification (categorizing output: correct class)
- Regression (continuous output to predict output based for new inputs)
- Prediction (classify / regression on new input sequences)

Supervised Learning



Unsupervised Learning

Given only inputs and automatically discover representations, features, structure etc.

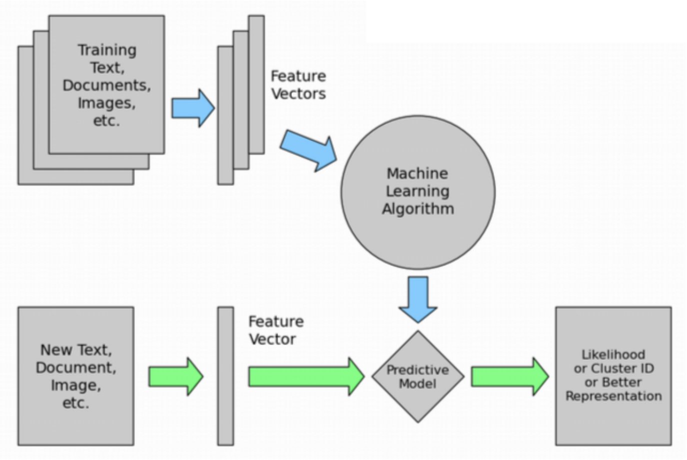
Unsupervised Learning

Given only inputs and automatically discover representations, features, structure etc.

Tasks:

- Clustering (to group similar data into a finite number of clusters / groups)
- Vector Quantization (compress / decode dataset into a new representation but maintaining internal information)
- Outlier Detection (select highly unusual cases/sequences)

Unsupervised Learning



Reinforcement Learning

- Learning approach of getting a computer system to act in the world so as to maximize its rewards.

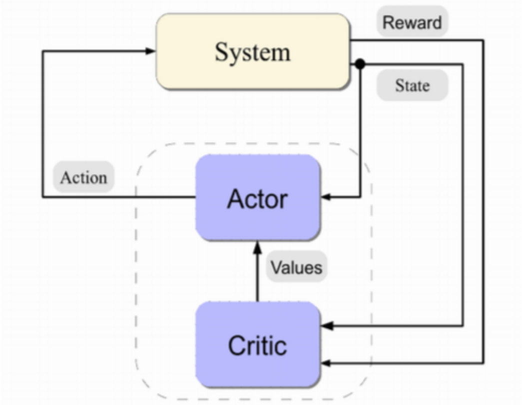
Reinforcement Learning

- Learning approach of getting a computer system to act in the world so as to maximize its rewards.
- Consider teaching a domestic animal. We cannot tell it what to do, but we can reward / punish if it does the right/ wrong thing.

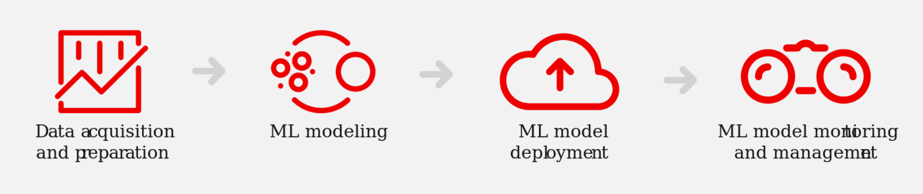
Reinforcement Learning

- Learning approach of getting a computer system to act in the world so as to maximize its rewards.
- Consider teaching a domestic animal. We cannot tell it what to do, but we can reward / punish if it does the right/ wrong thing.
- Process to determine what it did that made it get the reward / punishment – “credit assignment problem.”

Reinforcement Learning



Learning Lifecycle



https://www.openshift.com/

Activity 5: Algorithmic Bias

Google Search

Supervised Learning

Supervised Learning: Performance Measures

- A **feature** is a measurable property or a characteristic of the object we are trying to analyze (columns in a data set).

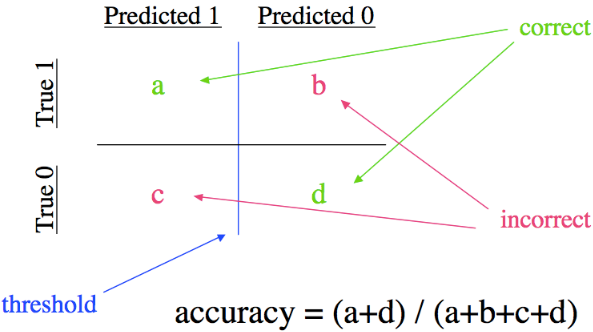
Supervised Learning: Performance Measures

- A **feature** is a measurable property or a characteristic of the object we are trying to analyze (columns in a data set).
- **Discrimination** attempts to separate distinct sets of objects.
- **Classification** attempts to allocate new objects to predefined groups.

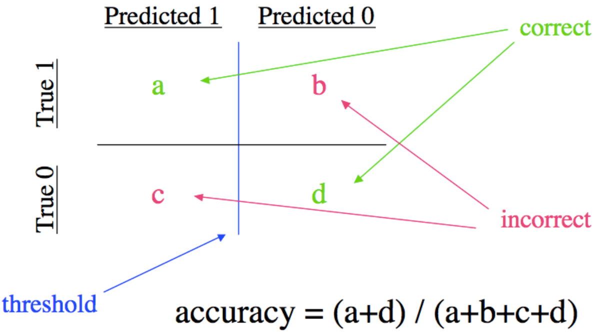
Performance Measures

- **Cost ratio** is a ratio of *false positives* (given condition is present when it is not) to *false negatives* (given condition is not present when it actually is).
- **Confusion matrix** (error matrix): a table to visualize the performance of an algorithm with rows/columns representing instances of predictions and columns/rows representing instances of actual class.

Confusion Matrix



Confusion Matrix



- a is a *true positive (TP)*.
- d is a *true negative (FN)*.
- c is a *false positive (FP)*.
- b is a *false negative (FN)*.

Classification Accuracy

Number of correctly classified examples divided by the total number of examples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Classification Accuracy

Number of correctly classified examples divided by the total number of examples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Error = 1 - Accuracy \tag{2}$$

Performance Measures

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Higher the recall the better class is correctly recognized (small number of FN).

Performance Measures

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Higher the recall the better class is correctly recognized (small number of FN).

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Higher the precision the better indication of an example labeled as positive being indeed positive (small number of FP).

Performance Measures

- High recall, low precision: Most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.
- Low recall, high precision: Miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP).

Performance Measures

- High recall, low precision: Most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.
- Low recall, high precision: Miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP).

$$F1 = 2\frac{Precision * Recall}{Precision + Recall} \tag{5}$$

F1 Score is used to find a balance between Precision and Recall.

Performance Measures

- Receiver Operator Characteristic **ROC curve**: plot of TP vs. FP.

Performance Measures

- Receiver Operator Characteristic **ROC curve**: plot of TP vs. FP.
- ROC Curves summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds.

Performance Measures

- Receiver Operator Characteristic **ROC curve**: plot of TP vs. FP.
- ROC Curves summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds.
- Precision-Recall curves summarize the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds.

Performance Measures

- Receiver Operator Characteristic **ROC curve**: plot of TP vs. FP.
- ROC Curves summarize the trade-off between the true positive rate and false positive rate for a predictive model using different probability thresholds.
- Precision-Recall curves summarize the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds.
- ROC curves are appropriate when the observations are balanced between each class, whereas precision-recall curves are appropriate for imbalanced datasets.

What-If Tool

Smile Detection Demo

https://pair-code.github.io/what-if-tool/

Computer Vision

Make computers understand images and video.

Computer Vision

Make computers understand images and video.

Computer Vision

Make computers understand images and video

Computer Vision

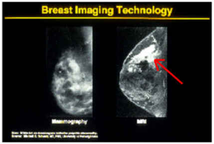
Make computers understand images and video

Computer Vision

- What kind of scene?
- Where are the cars?
- How far is the building?

Why computer vision matters?

Safety

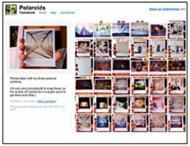


Health

Security

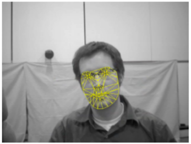
Comfort

Fun

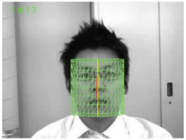


Access

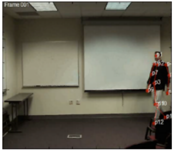
Applications of Computer Vision



"Face Recognition"



"Pose Estimation"



"Body Tracking"

"Speech Reading"

"Palm Recognition"

"Car Tracking"

Segmentation

- Compact representation for image data in terms of a set of components.

Segmentation

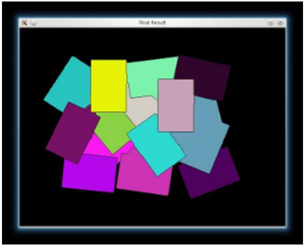
- Compact representation for image data in terms of a set of components.
- Components share "common" visual properties.

Segmentation

- Compact representation for image data in terms of a set of components.
- Components share "common" visual properties.
- Properties can be defined at different level of abstractions.

Segmentation

- Compact representation for image data in terms of a set of components.
- Components share "common" visual properties.
- Properties can be defined at different level of abstractions.



From: `https://docs.opencv.org`

Segmentation

- Tokens
  - whatever we need to group (pixels, points, surface elements, etc.).

Segmentation

- Tokens

  - whatever we need to group (pixels, points, surface elements, etc.).
- Bottom up segmentation

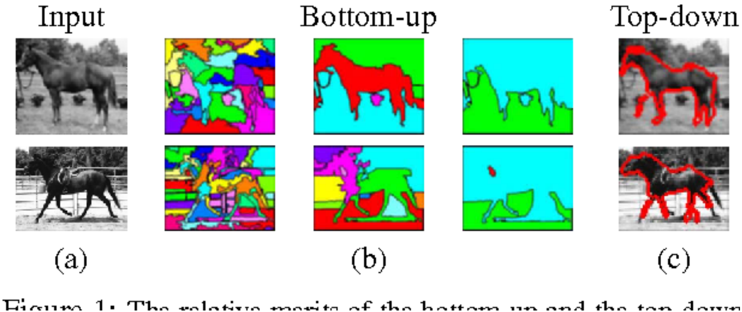
  - tokens belong together because they are locally coherent.

Segmentation

- Tokens
  - whatever we need to group (pixels, points, surface elements, etc.).
- Bottom up segmentation
  - tokens belong together because they are locally coherent.
- Top down segmentation
  - tokens belong together because they lie on the same object.

Segmentation

- Tokens
  - whatever we need to group (pixels, points, surface elements, etc.).
- Bottom up segmentation
  - tokens belong together because they are locally coherent.
- Top down segmentation
  - tokens belong together because they lie on the same object.



From: "Combining Top-Down and Bottom-Up Segmentation" by Eran Borenstein, Eitan Sharon, Shimon Ullman"

What is Segmentation?

Clustering image elements that "belong together"

- Partitioning
  - Divide into regions/sequences with coherent internal properties.
- Grouping
  - Identify sets of coherent tokens in image.

OpenCV

- An open source BSD licensed computer vision library
  - Patent-encumbered code isolated into "non-free" module (SIFT, SURF, some of the Face Detectors, etc.)
- Available on all major platforms
  - Android, iOS, Linux, Mac OS X, Windows
- Written primarily in C++
  - Bindings available for Python, Java, even MATLAB (in 3.0).
- Well documented at `http://docs.opencv.org`
- Source available at `https://github.com/Itseez/opencv`

OpenCV

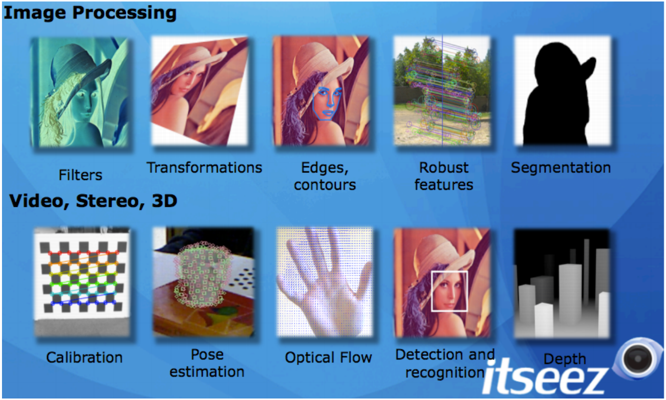


Image Processing
Filters
Transformations
Edges, contours
Robust features
Segmentation
Video, Stereo, 3D
Calibration
Pose estimation
Optical Flow
Detection and recognition
Depth
itseez

OpenCV: Pixel

- Grayscale: each pixel has a value between 0 (black) and 255 (white) - values between 0 and 255 are varying shades of gray.

OpenCV: Pixel

- Grayscale: each pixel has a value between 0 (black) and 255 (white) - values between 0 and 255 are varying shades of gray.
- Color: pixels are normally represented in the RGB color space
  - one value for the Red component, one for Green, and one for Blue,
  - each of the three colors is represented by an integer in the range 0 to 255,
  - how "much" of the color there is.

OpenCV: Coordinate System

- The point (0, 0) corresponds to the upper left corner of the image
- x value increases as we move to the right
- y value increases as we move down

OpenCV: Image Representation

- OpenCV represents images as `NumPy` arrays (matrices).
- `NumPy` is a library for the Python programming language that provides support for large, multi- dimensional arrays.
- To access a pixel value, we need to supply the x and y coordinates of the pixel.
- OpenCV actually stores RGB values in the order of Blue, Green, and Red.

Images

How to input or output an image?

How to input or output an image?

Load Image
Read image from disk.

`cv::imread(filename,0/1);`

0: read as grayscale image
1: read as color image

Save Image
Write image to disk.

`cv::imwrite(filename,im);`

Visualize Image
Show image in a window.

`cv::imshow(title,im);`

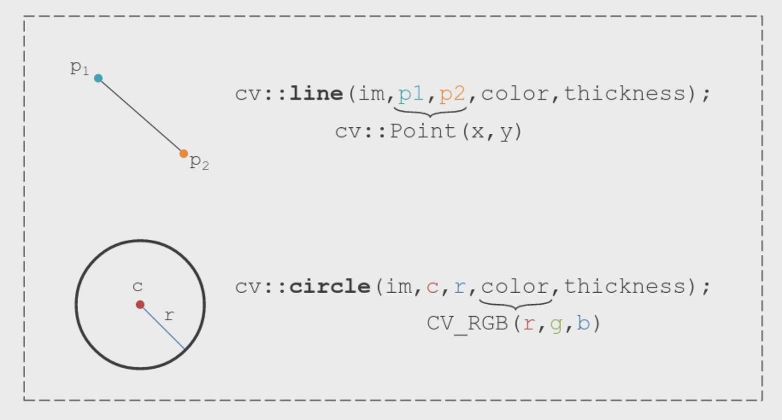
Note: if CV_32FC1, the gray value range is 0 to 1. Everything above 1 is white and everything below 0 is black.

Waitkey
Waits n milliseconds for user input.

`cv::waitkey(n);`

If n == -1, it waits forever.
Note: There must be a waitkey to show the image.

Drawing Primitives



Drawing Primitives

```
rectangle = np.zeros((300, 300), dtype = "uint8")
cv2.rectangle(rectangle, (25, 25), (275, 275), 255, -1)
```

OpenCV

1. Load an image from the disk, display it on our screen, and write it to file in a different format.
2. Access and manipulate pixels.