Lecture-1
Introduction

CS 277:
Machine Learning
and
Data Science

By:
Dr. Joydeep Chandra
Associate Professor
Dept. of CSE, IIT Patna

Tentative course plan

- Week 1: Introduction, Probability Distributions, Density Estimation
- Week 2: Density Estimation, Linear Regression
- Week 3: Logistic Regression, Linear Discriminant Analysis
- Week 4: Naïve Bayes, Decision trees
- Week 5: Support Vector Machines, Quiz 1
- Week 6: Feature selection techniques, Wrapper and Filter approaches
- Week 7: Mid semester exam
- Week 8: Mid semester break
- Week 9: Feature selection techniques, Forward and Backward selection, PCA
- Week 10: Unsupervised learning, K means, K medoid, hierarchical techniques, Expectation Maximization
- Week 11: Density based methods, Validity indices and similarity measures
- Week 12: Advanced clustering techniques
- Week 13: Graphical models
- Week 14: Graphical models, Quiz 2
- Week 15: Semi supervised learning, Active learning
- Week 16: Topic Modelling, LDA
- Week 17: End Sem Exam

Course Books

- Pattern Recognition and Machine Learning by Christopher M. Bishop
- The elements of statistical learning by Hastie, Tibshirani and Friedman

Course grading

- Programming assignments: 10%
- Quizzes: 15%
- Mid sem: 25%
- End sem 50%

What is Machine Learning?

- How can we solve a specific problem?

What is Machine Learning?

- How can we solve a specific problem?
  - As computer scientists we **write a program** that encodes a set of rules that are useful to solve the problem

Figure: How can we make a robot cook?

What is Machine Learning?

- How can we solve a specific problem?
  - As computer scientists we **write a program** that encodes a set of rules that are useful to solve the problem
  - In many cases is **very difficult to specify those rules**, e.g., given a picture determine whether there is a cat in the image

What is Machine Learning?

- How can we solve a specific problem?
  - As computer scientists we **write a program** that encodes a set of rules that are useful to solve the problem
  - In many cases is **very difficult to specify those rules**, e.g., given a picture determine whether there is a cat in the image
- Learning systems are not directly programmed to solve a problem, instead **develop own program** based on:

What is Machine Learning?

- How can we solve a specific problem?
  - As computer scientists we **write a program** that encodes a set of rules that are useful to solve the problem
  - In many cases is **very difficult to specify those rules**, e.g., given a picture determine whether there is a cat in the image
- Learning systems are not directly programmed to solve a problem, instead **develop own program** based on:
  - **Examples** of how they should behave

What is Machine Learning?

- How can we solve a specific problem?
  - As computer scientists we **write a program** that encodes a set of rules that are useful to solve the problem
  - In many cases is **very difficult to specify those rules**, e.g., given a picture determine whether there is a cat in the image
- Learning systems are not directly programmed to solve a problem, instead **develop own program** based on:
  - **Examples** of how they should behave
  - From **trial-and-error** experience trying to solve the problem

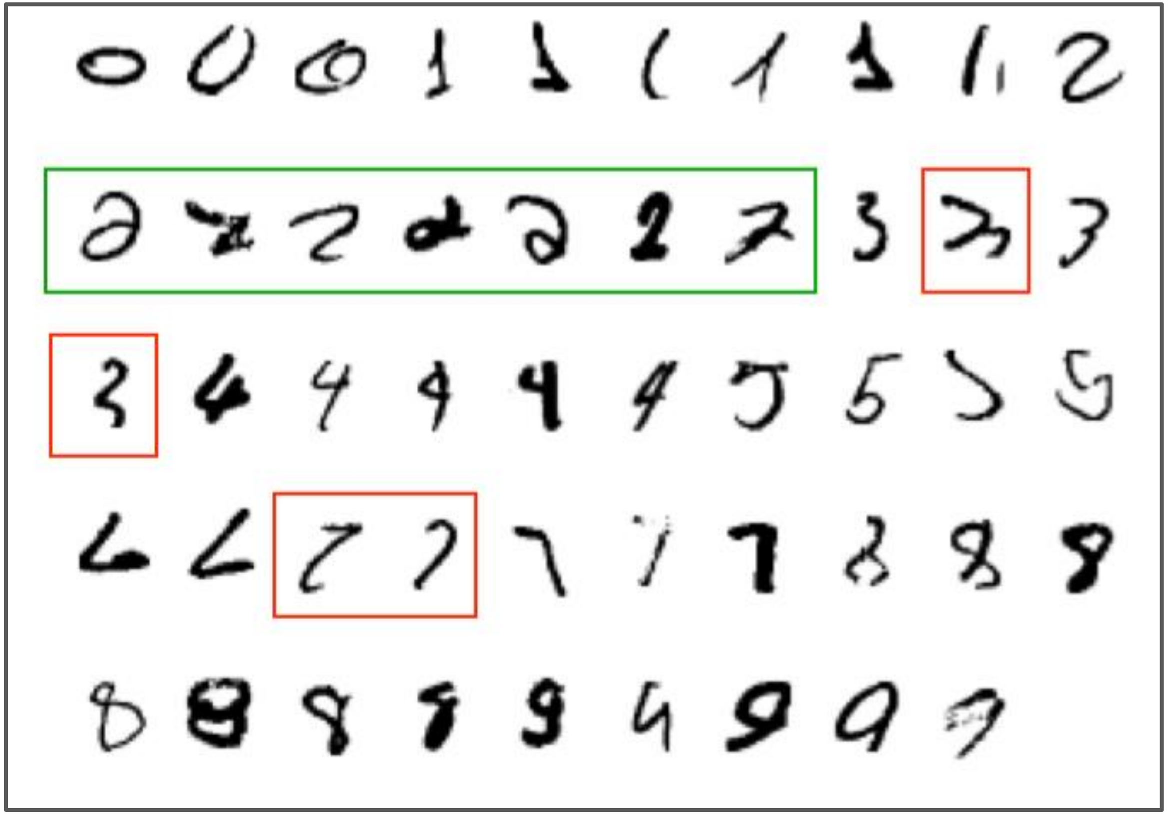
What is Machine Learning?

- How can we solve a specific problem?
  - As computer scientists we **write a program** that encodes a set of rules that are useful to solve the problem
  - In many cases is **very difficult to specify those rules**, e.g., given a picture determine whether there is a cat in the image
- Learning systems are not directly programmed to solve a problem, instead **develop own program** based on:
  - **Examples** of how they should behave
  - From **trial-and-error** experience trying to solve the problem
- Different than standard CS:
  - Want to implement unknown function, only have access e.g., to sample input-output pairs (training examples)

What is Machine Learning?

- How can we solve a specific problem?
  - As computer scientists we **write a program** that encodes a set of rules that are useful to solve the problem
  - In many cases is **very difficult to specify those rules**, e.g., given a picture determine whether there is a cat in the image
- Learning systems are not directly programmed to solve a problem, instead **develop own program** based on:
  - **Examples** of how they should behave
  - From **trial-and-error** experience trying to solve the problem
- Different than standard CS:
  - Want to implement unknown function, only have access e.g., to sample
  - input-output pairs (training examples)
- Learning simply means incorporating information from the training examples into the system

Tasks that requires machine learning: What makes a 2?



Tasks that benefits from machine learning: cooking!

Robots learn to cook by watching YouTube

When it comes to learning how to cook, it turns out that robots may not be so different from humans after all... or are they?

Sci-Tech

January 20, 2015
4:26 PM PST

by Michelle Starr
@riding_red

When it comes to teaching robots how to do things, there are some very key differences. A human knows what you mean when you say "I need a cup". A robot needs to be taught that that means it has to turn around, go to the cupboard, open it, take out the cup, close the cupboard, turn back around, return to you, manoeuvre the cup over the bench, and release the cup.

John T. Consoli, UMD

This is one of the key parts of figuring out machine learning: How can you program a robot so that it can intuit that a plastic cup, a glass and a mug may all be classified under the general term "cup"? How can you design a robot that is able to teach itself?

One way, as researchers at the University of Maryland Institute for Advanced Computer Studies are finding out, is YouTube. More specifically, cooking tutorials on YouTube. By watching these videos, [illegible] arn the complicated series of grasping and manipulation motions required for

Why use learning?

- It is very hard to write programs that solve problems like recognizing a handwritten digit

Why use learning?

- It is very hard to write programs that solve problems like recognizing a handwritten digit
  - What distinguishes a 2 from a 7?

Why use learning?

- It is very hard to write programs that solve problems like recognizing a handwritten digit
  - What distinguishes a 2 from a 7?
  - How does our brain do it?

Why use learning?

- It is very hard to write programs that solve problems like recognizing a handwritten digit
  - What distinguishes a 2 from a 7?
  - How does our brain do it?
- Instead of writing a program by hand, we collect examples that specify the correct output for a given input

Why use learning?

- It is very hard to write programs that solve problems like recognizing a handwritten digit
  - What distinguishes a 2 from a 7?
  - How does our brain do it?
- Instead of writing a program by hand, we collect examples that specify the correct output for a given input
- A machine learning algorithm then takes these examples and produces a program that does the job

Why use learning?

- It is very hard to write programs that solve problems like recognizing a handwritten digit
  - What distinguishes a 2 from a 7?
  - How does our brain do it?
- Instead of writing a program by hand, we collect examples that specify the correct output for a given input
- A machine learning algorithm then takes these examples and produces a program that does the job
  - The program produced by the learning algorithm may look very different from a typical hand-written program. It may contain millions of numbers.

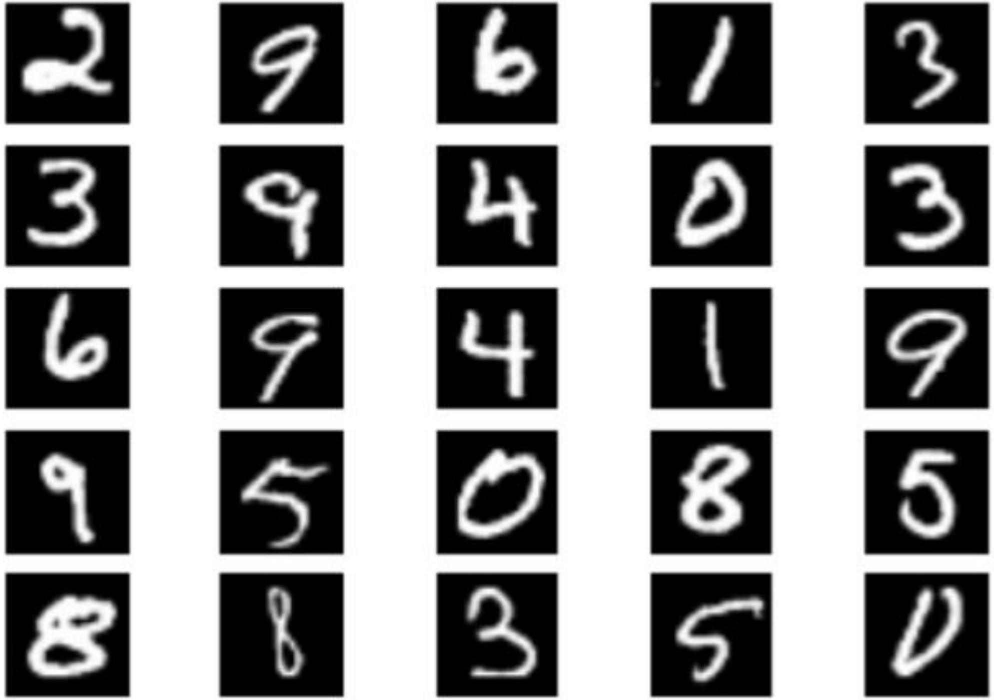
Why use learning?

- It is very hard to write programs that solve problems like recognizing a handwritten digit
  - What distinguishes a 2 from a 7?
  - How does our brain do it?
- Instead of writing a program by hand, we collect examples that specify the correct output for a given input
- A machine learning algorithm then takes these examples and produces a program that does the job
  - The program produced by the learning algorithm may look very different from a typical hand-written program. It may contain millions of numbers.
  - If we do it right, the program works for new cases as well as the ones we trained it on.

Learning algorithms are useful in many tasks

1. **Classification:** Determine which discrete category the example is

Examples of Classification



What digit is this?

Examples of Classification

Is this a dog?

Examples of Classification

what about this one?

Examples of Classification

Am I going to pass the exam?

Do I have diabetes?

Learning algorithms are useful in many tasks

1. **Classification:** Determine which discrete category the example is
2. **Recognizing patterns:** Speech Recognition, facial identity, etc

Examples of Recognizing patterns

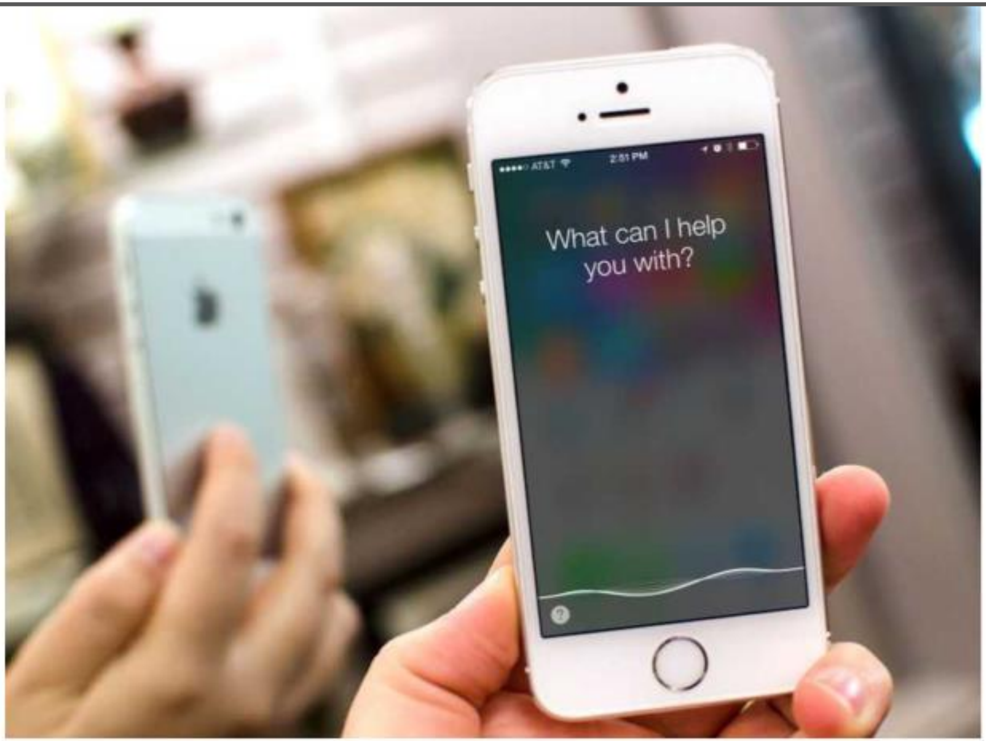


Figure: Siri: https://www.youtube.com/watch?v=8ciagGASro0

Examples of Recognizing patterns

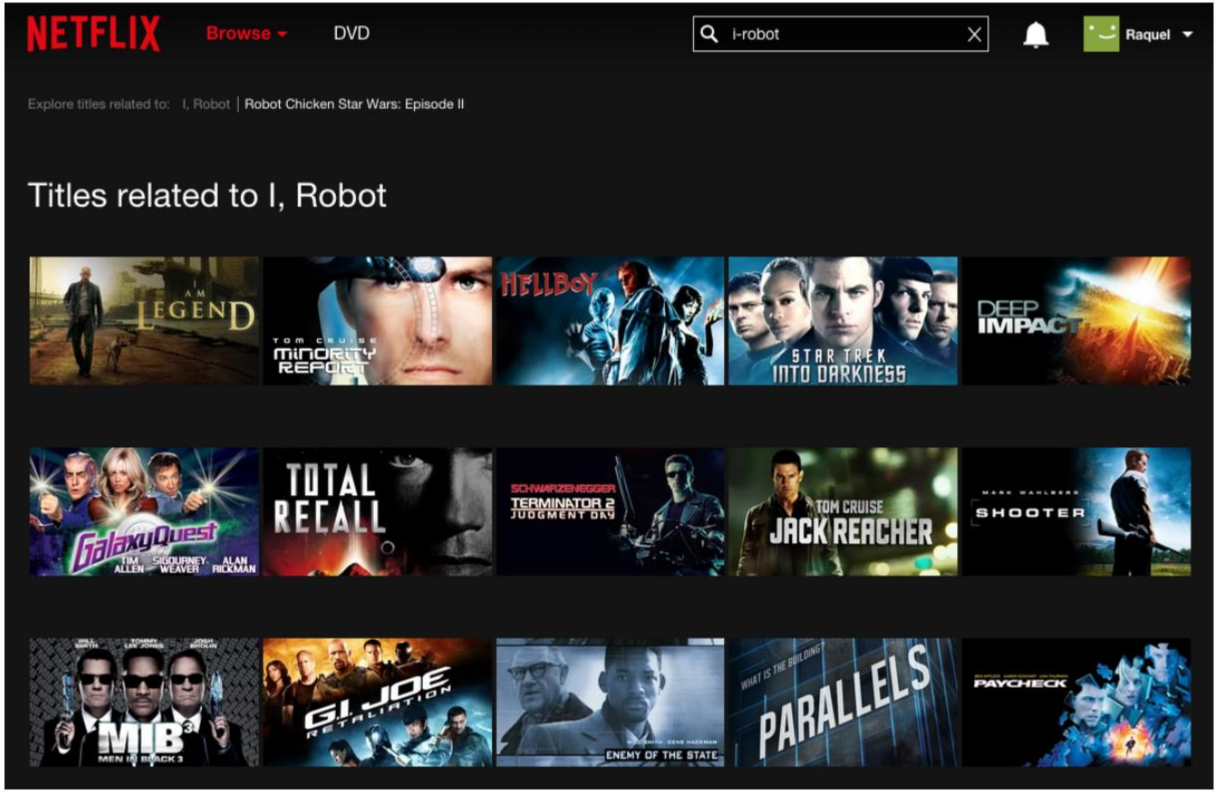
Figure: Photomath: https://photomath.net/

Learning algorithms are useful in many tasks

1. **Classification:** Determine which discrete category the example is
2. **Recognizing patterns:** Speech Recognition, facial identity, etc
3. **Recommender Systems:** Noisy data, commercial pay-off (e.g., Amazon, Netflix).

Examples of Recommendation systems

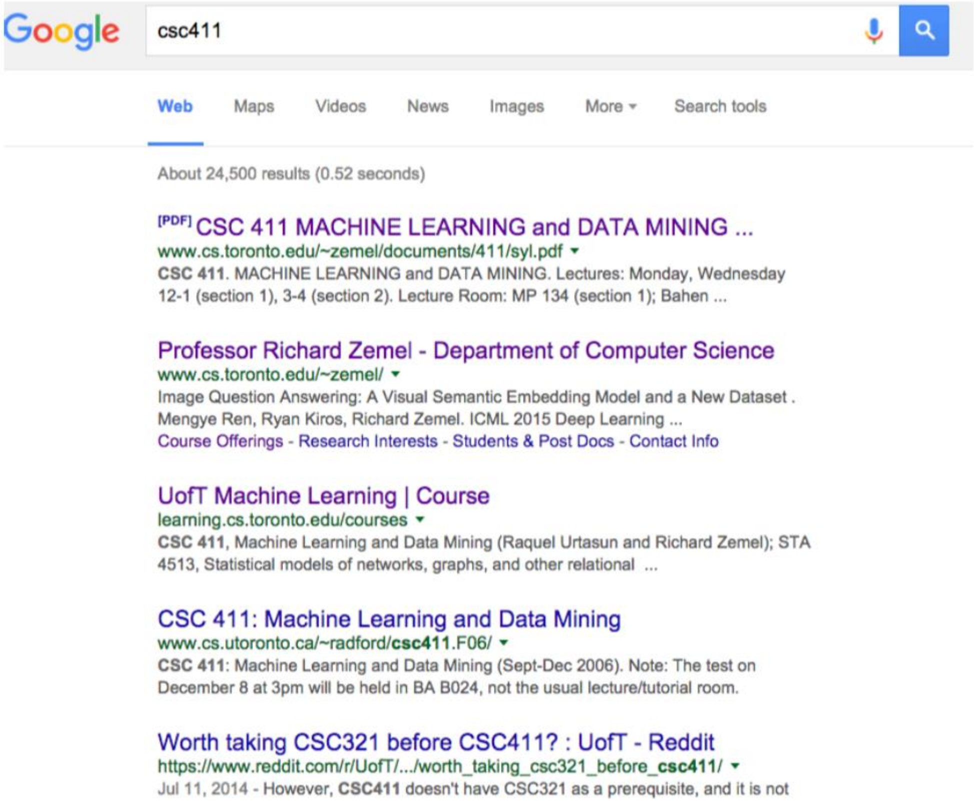
Examples of Recommendation systems



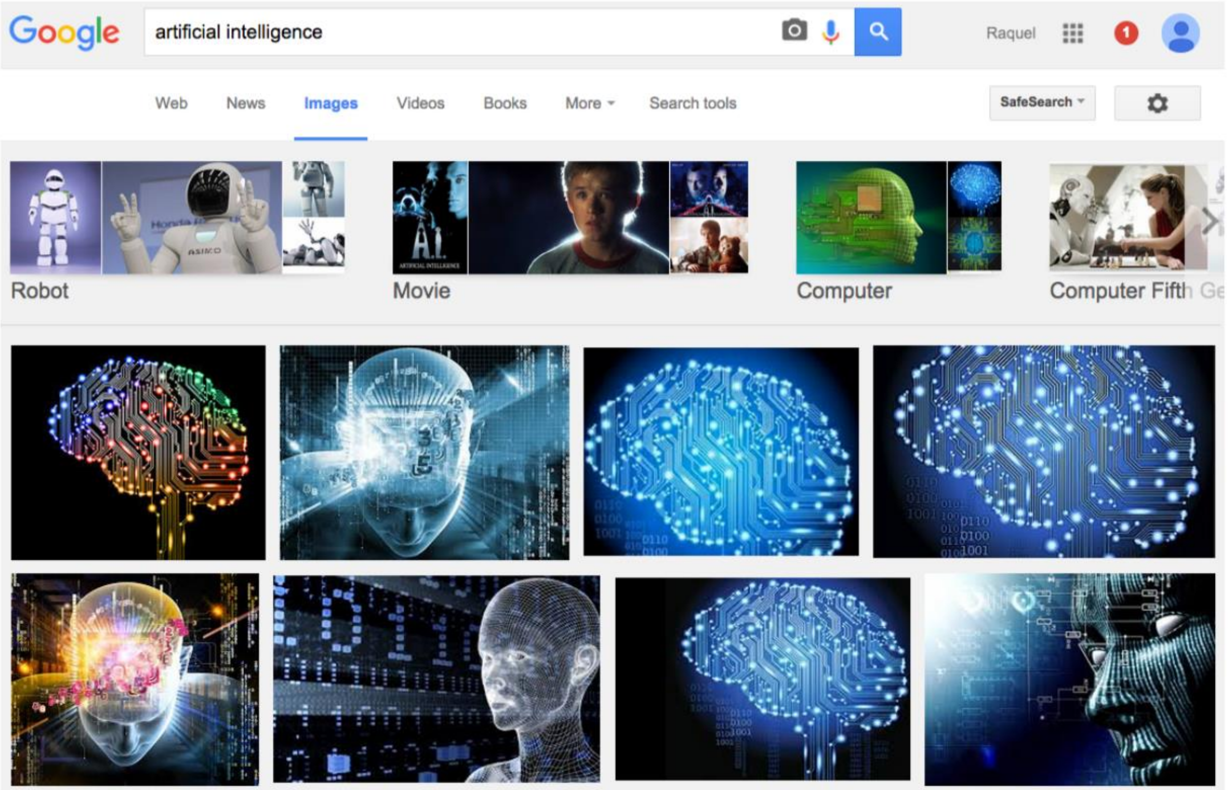
Learning algorithms are useful in many tasks

1. **Classification:** Determine which discrete category the example is
2. **Recognizing patterns:** Speech Recognition, facial identity, etc
3. **Recommender Systems:** Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval:** Find documents or images with similar content

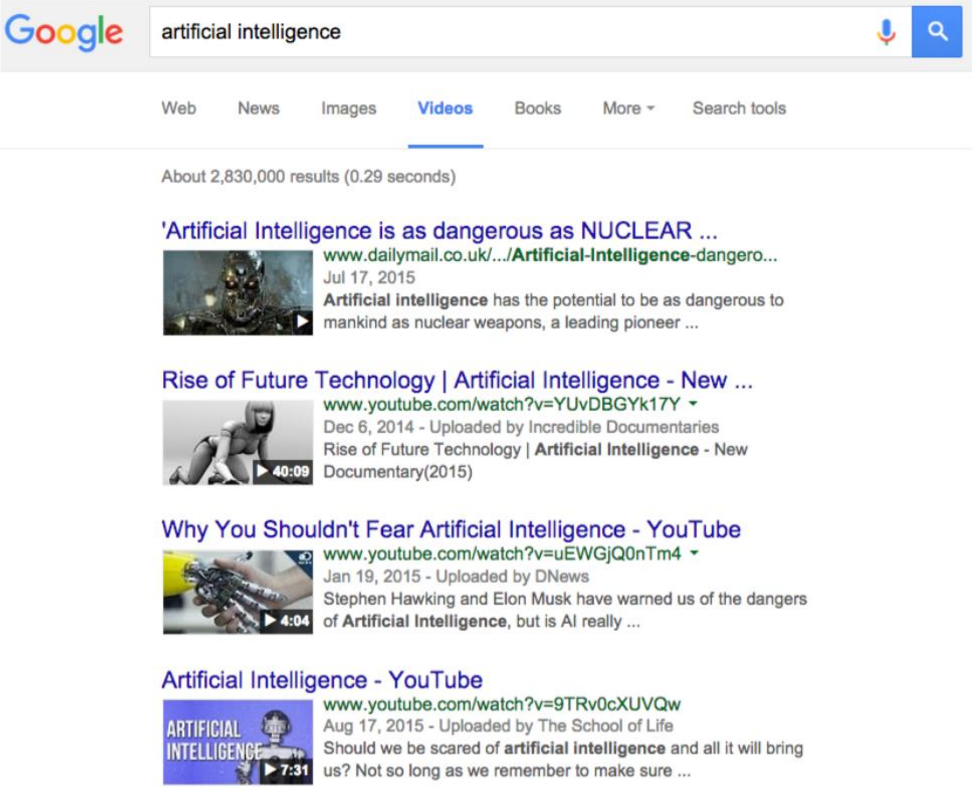
Examples of Information Retrieval



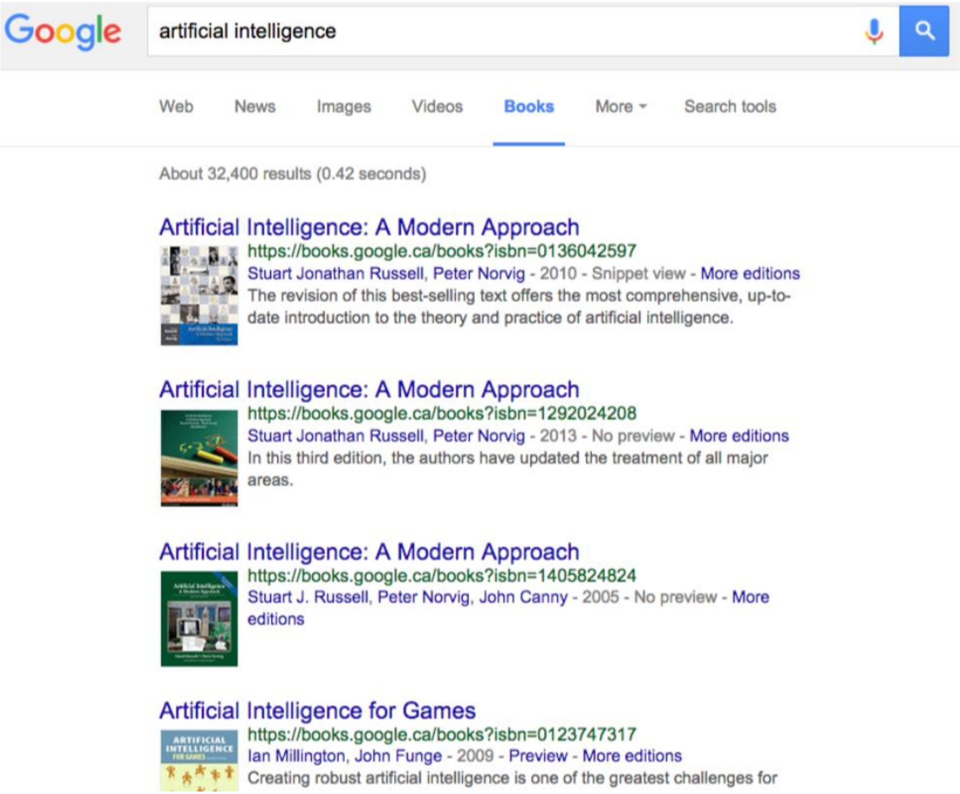
Examples of Information Retrieval



Examples of Information Retrieval



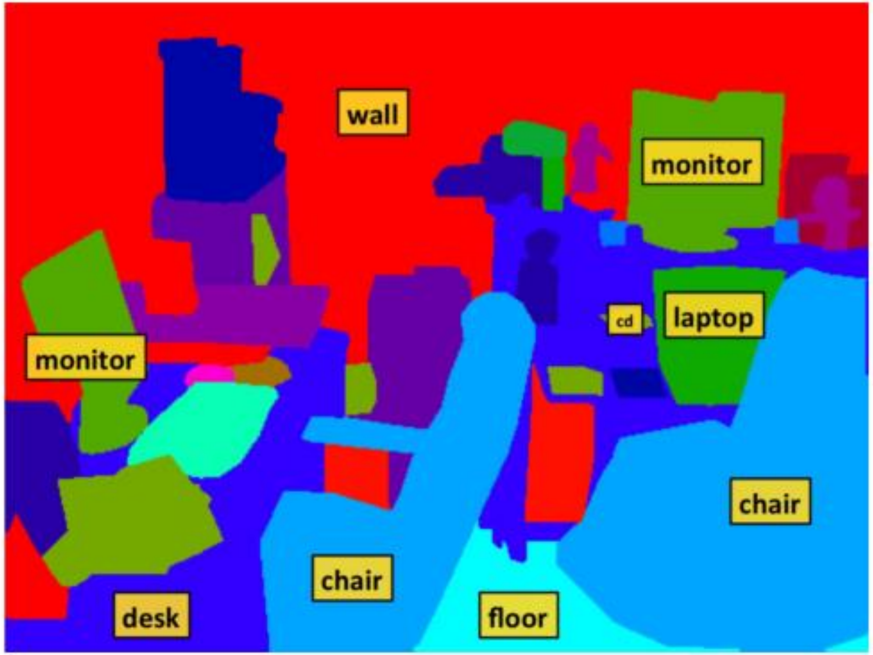
Examples of Information Retrieval



Learning algorithms are useful in many tasks

1. **Classification:** Determine which discrete category the example is
2. **Recognizing patterns:** Speech Recognition, facial identity, etc
3. **Recommender Systems:** Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval:** Find documents or images with similar content
5. **Computer vision:** detection, segmentation, depth estimation, optical flow, etc

Computer Vision



Computer Vision

Figure: Kinect: https://www.youtube.com/watch?v=op82fDRRqSY

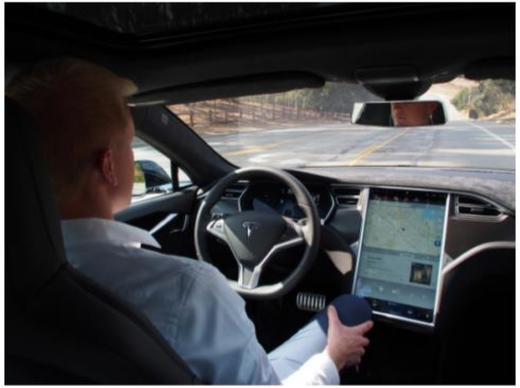
Computer Vision

[Gatys, Ecker, Bethge. A Neural Algorithm of Artistic Style. Arxiv'15.]

Learning algorithms are useful in many tasks

1. **Classification:** Determine which discrete category the example is
2. **Recognizing patterns:** Speech Recognition, facial identity, etc
3. **Recommender Systems:** Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval:** Find documents or images with similar content
5. **Computer vision:** detection, segmentation, depth estimation, optical flow, etc
6. **Robotics:** perception, planning, etc

Autonomous Driving



Flying Robots

Figure: Video: https://www.youtube.com/watch?v=YQIMGV5vtd4

Learning algorithms are useful in many tasks

1. **Classification:** Determine which discrete category the example is
2. **Recognizing patterns:** Speech Recognition, facial identity, etc
3. **Recommender Systems:** Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval:** Find documents or images with similar content
5. **Computer vision:** detection, segmentation, depth estimation, optical flow, etc
6. **Robotics:** perception, planning, etc
7. **Learning to play games**

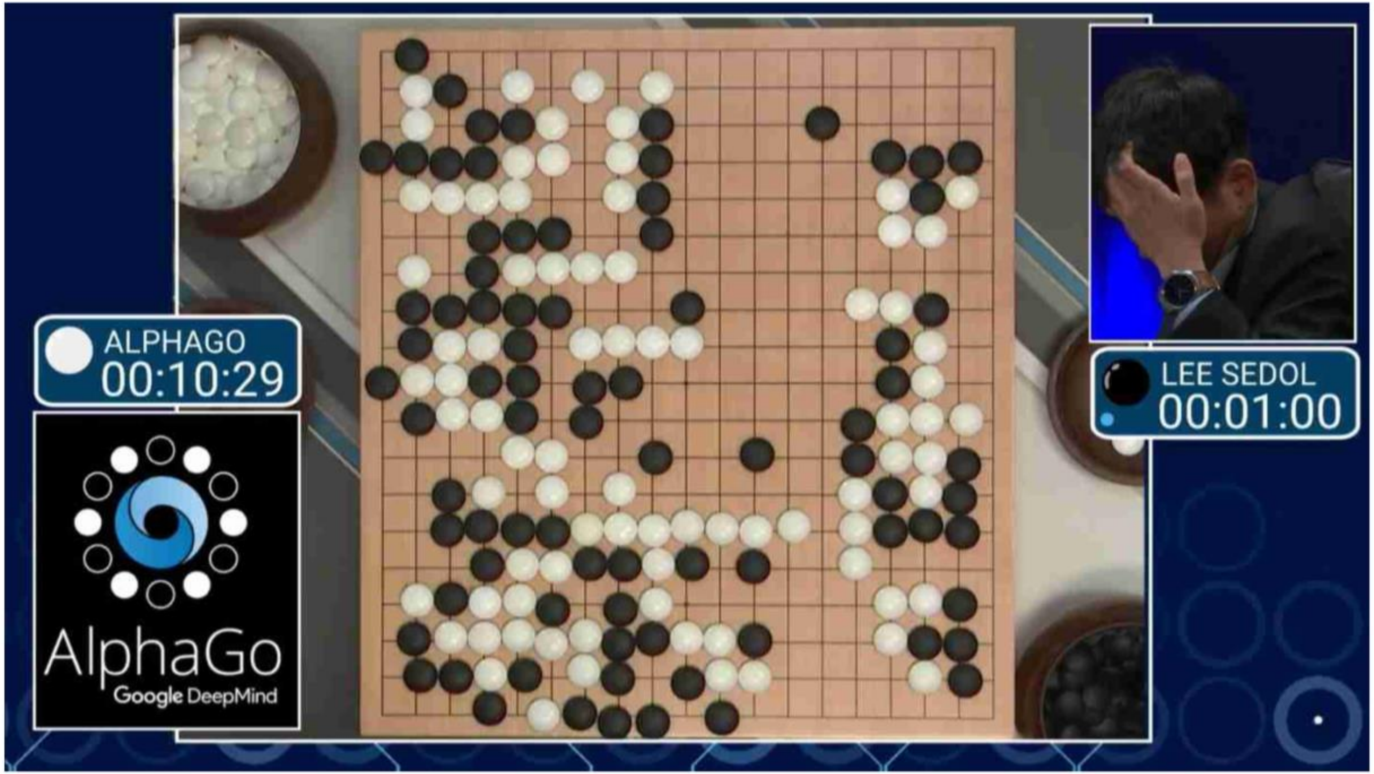
Playing Games: Atari

Figure: Video: https://www.youtube.com/watch?v=V1eYniJ0Rnk

Playing Games: Super Mario

Figure: Video: https://www.youtube.com/watch?v=wfL4L_14U9A

Playing Games: Alpha Go



Learning algorithms are useful in many tasks

1. **Classification:** Determine which discrete category the example is
2. **Recognizing patterns:** Speech Recognition, facial identity, etc
3. **Recommender Systems:** Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval:** Find documents or images with similar content
5. **Computer vision:** detection, segmentation, depth estimation, optical flow, etc
6. **Robotics:** perception, planning, etc
7. **Learning to play games**
8. **Recognizing anomalies:** Unusual sequences of credit card transactions, panic situation at an airport

Learning algorithms are useful in many tasks

1. **Classification:** Determine which discrete category the example is
2. **Recognizing patterns:** Speech Recognition, facial identity, etc
3. **Recommender Systems:** Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval:** Find documents or images with similar content
5. **Computer vision:** detection, segmentation, depth estimation, optical flow, etc
6. **Robotics:** perception, planning, etc
7. **Learning to play games**
8. **Recognizing anomalies:** Unusual sequences of credit card transactions, panic situation at an airport
9. **Spam filtering, fraud detection:** The enemy adapts so we must adapt too

Learning algorithms are useful in many tasks

1. **Classification:** Determine which discrete category the example is
2. **Recognizing patterns:** Speech Recognition, facial identity, etc
3. **Recommender Systems:** Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval:** Find documents or images with similar content
5. **Computer vision:** detection, segmentation, depth estimation, optical flow, etc
6. **Robotics:** perception, planning, etc
7. **Learning to play games**
8. **Recognizing anomalies:** Unusual sequences of credit card transactions, panic situation at an airport
9. **Spam filtering, fraud detection:** The enemy adapts so we must adapt too
10. **Many more!**

Types of learning tasks

- Supervised: correct output known for each training example

Types of learning tasks

- Supervised: correct output known for each training example
    - Learn to predict output when given an input vector

Types of learning tasks

- Supervised: correct output known for each training example
    - Learn to predict output when given an input vector
        - Classification: 1-of-N output (speech recognition, object recognition, medical diagnosis)

Types of learning tasks

- Supervised: correct output known for each training example
  - Learn to predict output when given an input vector
    - Classification: 1-of-N output (speech recognition, object recognition, medical diagnosis)
    - Regression: real-valued output (predicting market prices, customer rating)

Types of learning tasks

- Supervised: correct output known for each training example
  - Learn to predict output when given an input vector
    - Classification: 1-of-N output (speech recognition, object recognition, medical diagnosis)
    - Regression: real-valued output (predicting market prices, customer rating)
- Unsupervised learning
  - Create an internal representation of the input, capturing regularities/structure in data

Types of learning tasks

- Supervised: correct output known for each training example
    - Learn to predict output when given an input vector
        - Classification: 1-of-N output (speech recognition, object recognition, medical diagnosis)
        - Regression: real-valued output (predicting market prices, customer rating)
- Unsupervised learning
    - Create an internal representation of the input, capturing regularities/structure in data
    - Examples: form clusters; extract features

Types of learning tasks

- Supervised: correct output known for each training example
  - Learn to predict output when given an input vector
    - Classification: 1-of-N output (speech recognition, object recognition, medical diagnosis)
    - Regression: real-valued output (predicting market prices, customer rating)
- Unsupervised learning
  - Create an internal representation of the input, capturing regularities/structure in data
  - Examples: form clusters; extract features
    - How do we know if a representation is good?

Types of learning tasks

- Supervised: correct output known for each training example
  - Learn to predict output when given an input vector
    - Classification: 1-of-N output (speech recognition, object recognition, medical diagnosis)
    - Regression: real-valued output (predicting market prices, customer rating)
- Unsupervised learning
  - Create an internal representation of the input, capturing regularities/structure in data
  - Examples: form clusters; extract features
    - How do we know if a representation is good?
- Reinforcement learning
  - Learn action to maximize payoff
    - Not much information in a payoff signal
    - Payoff is often delayed

Machine Learning vs Data Mining

- **Data-mining:** Typically using very simple machine learning techniques on very large databases because computers are too slow to do anything more interesting with ten billion examples

Machine Learning vs Data Mining

- **Data-mining:** Typically using very simple machine learning techniques on very large databases because computers are too slow to do anything more interesting with ten billion examples
- Previously used in a negative sense
  - misguided statistical procedure of looking for all kinds of relationships in the data until finally find one

Machine Learning vs Data Mining

- **Data-mining:** Typically using very simple machine learning techniques on very large databases because computers are too slow to do anything more interesting with ten billion examples
- Previously used in a negative sense
    - misguided statistical procedure of looking for all kinds of relationships in the data until finally find one
- Now lines are blurred: many ML problems involve tons of data

Machine Learning vs Statistics

- ML uses statistical theory to build models

Machine Learning vs Statistics

- ML uses statistical theory to build models
- A lot of ML is rediscovery of things statisticians already knew; often disguised by differences in terminology

Machine Learning vs Statistics

- ML uses statistical theory to build models
- A lot of ML is rediscovery of things statisticians already knew; often disguised by differences in terminology
- But the emphasis is very different:
    - Good piece of statistics: Clever proof that relatively simple estimation procedure is asymptotically unbiased.

Machine Learning vs Statistics

- ML uses statistical theory to build models
- A lot of ML is rediscovery of things statisticians already knew; often disguised by differences in terminology
- But the emphasis is very different:
  - Good piece of statistics: Clever proof that relatively simple estimation procedure is asymptotically unbiased.
  - Good piece of ML: Demo that a complicated algorithm produces impressive results on a specific task.
- Can view ML as applying computational techniques to statistical problems. But go beyond typical statistics problems, with different aims (speed vs. accuracy).

Any Questions??