CSC 411: Lecture 01: Introduction

Rich Zemel, Raquel Urtasun and Sanja Fidler

University of Toronto

Today

- Administration details
- Why is machine learning so cool?

The Team I

- **Instructors**:
  - Raquel Urtasun
  - Richard Zemel
- **Email**:
  - csc411prof@cs.toronto.edu
- **Offices**:
  - Raquel: 290E in Pratt
  - Richard: 290D in Pratt
- **Office hours**: TBA

The Team II

- **TA's**:
  - Siddharth Ancha
  - Azin Asgarian
  - Min Bai
  - Lluis Castrejon Subira
  - Kaustav Kundu
  - Hao-Wei Lee
  - Renjie Liao
  - Shun Liao
  - Wenjie Luo
  - David Madras
  - Seyed Parsa Mirdehghan
  - Mengye Ren
  - Geoffrey Roeder
  - Yulia Rubanova
  - Elias Tragas
  - Eleni Triantafillou
  - Shenlong Wang
  - Ayazhan Zhakhan
- **Email**:
  - csc411ta@cs.toronto.edu

Admin Details

- Liberal wrt waiving pre-requisites
  - But it is up to you to determine if you have the appropriate background
- Do I have the appropriate background?
  - Linear algebra: vector/matrix manipulations, properties
  - Calculus: partial derivatives
  - Probability: common distributions; Bayes Rule
  - Statistics: mean/median/mode; maximum likelihood
  - Sheldon Ross: A First Course in Probability

Course Information (Section 1)

- **Class**: Mondays at 11-1pm in AH 400
- **Instructor**: Raquel Urtasun
- **Tutorials**: Monday, 3-4pm, same classroom
- **Class Website**:

  http://www.cs.toronto.edu/~urtasun/courses/CSC411_Fall16/CSC411_Fall16.html
- The class will use Piazza for **announcements** and **discussions**:

  https://piazza.com/utoronto.ca/fall2016/csc411/home
- First time, sign up here:

  https://piazza.com/utoronto.ca/fall2016/csc411
- Your grade will **not depend on your participation on Piazza**. It's just a good way for asking questions, discussing with your instructor, TAs and your peers

Course Information (Section 2)

- **Class**: Wednesdays at 11-1pm in MS 2170
- **Instructor**: Raquel Urtasun
- **Tutorials**: Wednesday, 3-4pm, BA 1170
- **Class Website**:

  http://www.cs.toronto.edu/~urtasun/courses/CSC411_Fall16/CSC411_Fall16.html
- The class will use Piazza for **announcements** and **discussions**:

  https://piazza.com/utoronto.ca/fall2016/csc411/home
- First time, sign up here:

  https://piazza.com/utoronto.ca/fall2016/csc411/home
- Your grade will **not depend on your participation on Piazza**. It's just a good way for asking questions, discussing with your instructor, TAs and your peers

Course Information (Section 3)

- **Class**: Thursdays at 4-6pm in KP 108
- **Instructor**: Richard Zemel
- **Tutorials**: Thursday, 6-7pm, same class
- **Class Website**:

  http://www.cs.toronto.edu/~urtasun/courses/CSC411_Fall16/CSC411_Fall16.html
- The class will use Piazza for **announcements** and **discussions**:

  https://piazza.com/utoronto.ca/fall2016/csc411/home
- First time, sign up here:

  https://piazza.com/utoronto.ca/fall2016/csc411/home
- Your grade will **not depend on your participation on Piazza**. It's just a good way for asking questions, discussing with your instructor, TAs and your peers

Course Information (Section 4)

- **Class**: Fridays at 11-1pm in MS 2172
- **Instructor**: Richard Zemel
- **Tutorials**: Thursday, 3-4pm, same class
- **Class Website**:

  http://www.cs.toronto.edu/~urtasun/courses/CSC411_Fall16/CSC411_Fall16.html
- The class will use Piazza for **announcements** and **discussions**:

  https://piazza.com/utoronto.ca/fall2016/csc411/home
- First time, sign up here:

  https://piazza.com/utoronto.ca/fall2016/csc411/home
- Your grade will **not depend on your participation on Piazza**. It's just a good way for asking questions, discussing with your instructor, TAs and your peers

Textbook(s)

- Christopher Bishop: *"Pattern Recognition and Machine Learning"*, 2006

Textbook(s)

- Christopher Bishop: *"Pattern Recognition and Machine Learning"*, 2006
- Other Textbooks:
  - Kevin Murphy: *"Machine Learning: a Probabilistic Perspective"*
  - David Mackay: *"Information Theory, Inference, and Learning Algorithms"*
  - Ethem Alpaydin: *"Introduction to Machine Learning"*, 2nd edition, 2010.

Requirements (Undergrads)

- Do the readings!

Requirements (Undergrads)

- Do the readings!
- Assignments:
  - Three assignments, first two worth 15% each, last one worth 25%, for a total of 55%
  - Programming: take code and extend it
  - Derivations: pen(cil)-and-paper

Requirements (Undergrads)

- Do the readings!
- Assignments:
  - Three assignments, first two worth 15% each, last one worth 25%, for a total of 55%
  - Programming: take code and extend it
  - Derivations: pen(cil)-and-paper
- Mid-term:
  - One hour exam
  - Worth 20% of course mark

Requirements (Undergrads)

- Do the readings!
- Assignments:
  - Three assignments, first two worth 15% each, last one worth 25%, for a total of 55%
  - Programming: take code and extend it
  - Derivations: pen(cil)-and-paper
- Mid-term:
  - One hour exam
  - Worth 20% of course mark
- Final:
  - Focused on second half of course
  - Worth 25% of course mark

Requirements (Grads)

- Do the readings!

Requirements (Grads)

- Do the readings!
- Assignments:
  - Three assignments, first two worth 15% each, last one worth 25%, for a total of 55%
  - Programming: take code and extend it
  - Derivations: pen(cil)-and-paper

Requirements (Grads)

- Do the readings!
- Assignments:
  - Three assignments, first two worth 15% each, last one worth 25%, for a total of 55%
  - Programming: take code and extend it
  - Derivations: pen(cil)-and-paper
- Mid-term:
  - One hour exam
  - Worth 20% of course mark

Requirements (Grads)

- Do the readings!
- Assignments:
  - Three assignments, first two worth 15% each, last one worth 25%, for a total of 55%
  - Programming: take code and extend it
  - Derivations: pen(cil)-and-paper
- Mid-term:
  - One hour exam
  - Worth 20% of course mark
- Final:
  - Focused on second half of course
  - Worth 25% of course mark

More on Assigments

- Collaboration on the assignments is not allowed. Each student is responsible for his/her own work. Discussion of assignments should be limited to clarification of the handout itself, and should not involve any sharing of pseudocode or code or simulation results. Violation of this policy is grounds for a semester grade of F, in accordance with university regulations.

More on Assigments

- Collaboration on the assignments is not allowed. Each student is responsible for his/her own work. Discussion of assignments should be limited to clarification of the handout itself, and should not involve any sharing of pseudocode or code or simulation results. Violation of this policy is grounds for a semester grade of F, in accordance with university regulations.
- The schedule of assignments is included in the syllabus. Assignments are due at the beginning of class/tutorial on the due date.

More on Assigments

- Collaboration on the assignments is not allowed. Each student is responsible for his/her own work. Discussion of assignments should be limited to clarification of the handout itself, and should not involve any sharing of pseudocode or code or simulation results. Violation of this policy is grounds for a semester grade of F, in accordance with university regulations.
- The schedule of assignments is included in the syllabus. Assignments are due at the beginning of class/tutorial on the due date.
- Assignments handed in late but before 5 pm of that day will be penalized by 5% (i.e., total points multiplied by 0.95); a late penalty of 10% per day will be assessed thereafter.

More on Assigments

- Collaboration on the assignments is not allowed. Each student is responsible for his/her own work. Discussion of assignments should be limited to clarification of the handout itself, and should not involve any sharing of pseudocode or code or simulation results. Violation of this policy is grounds for a semester grade of F, in accordance with university regulations.
- The schedule of assignments is included in the syllabus. Assignments are due at the beginning of class/tutorial on the due date.
- Assignments handed in late but before 5 pm of that day will be penalized by 5% (i.e., total points multiplied by 0.95); a late penalty of 10% per day will be assessed thereafter.
- Extensions will be granted only in special situations, and you will need a Student Medical Certificate or a written request approved by the instructor at least one week before the due date.

More on Assigments

- Collaboration on the assignments is not allowed. Each student is responsible for his/her own work. Discussion of assignments should be limited to clarification of the handout itself, and should not involve any sharing of pseudocode or code or simulation results. Violation of this policy is grounds for a semester grade of F, in accordance with university regulations.
- The schedule of assignments is included in the syllabus. Assignments are due at the beginning of class/tutorial on the due date.
- Assignments handed in late but before 5 pm of that day will be penalized by 5% (i.e., total points multiplied by 0.95); a late penalty of 10% per day will be assessed thereafter.
- Extensions will be granted only in special situations, and you will need a Student Medical Certificate or a written request approved by the instructor at least one week before the due date.
- Final assignment is a bake-off: competition between ML algorithms. We will give you some data for training a ML system, and you will try to develop the best method. We will then determine which system performs best on unseen test data. Grads can do own project.

Provisional Calendar (Section 1)

- Intro + Linear Regression
- Linear Classif. + Logistic Regression
- Non-parametric + Decision trees
- Multi-class + Prob. Classif I
- Thanksgiving
- Prob. Classif II + NNets I
- Nnet II + Clustering
- Midterm + Mixt. of Gaussians
- Reading Week
- PCA/Autoencoders + SVM
- Kernels + Ensemble I
- Ensemble II + RL

Provisional Calendar (Sections 2,3,4)

- Intro + Linear Regression
- Linear Classif. + Logistic Regression
- Non-parametric + Decision trees
- Multi-class + Prob. Classif I
- Prob. Classif II + NNets I
- Nnet II + Clustering
- Midterm + Mixt. of Gaussians
- PCA/Autoencoders + SVM
- Kernels + Ensemble I
- Ensemble II + RL

What is Machine Learning?

- How can we solve a specific problem?

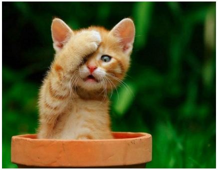
What is Machine Learning?

- 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

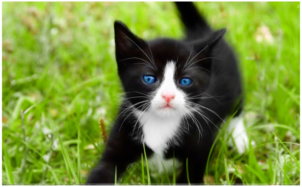
Figure : How can we make a robot cook?

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

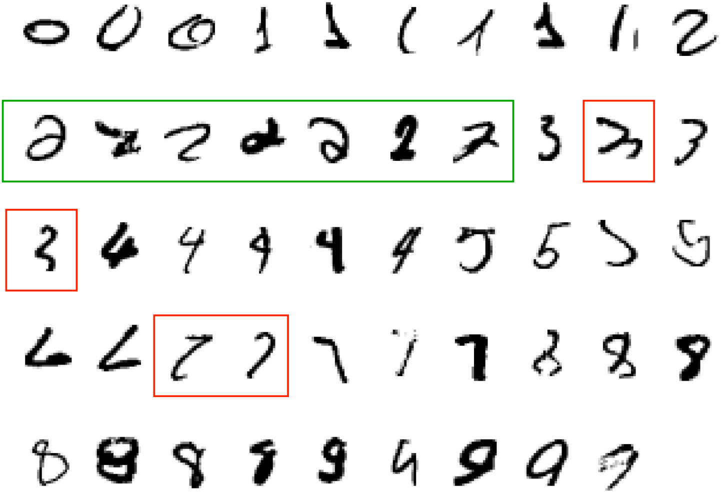
What is Machine Learning?

- 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!

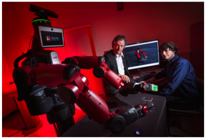
Robots learn to cook by watching YouTube

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, ...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.

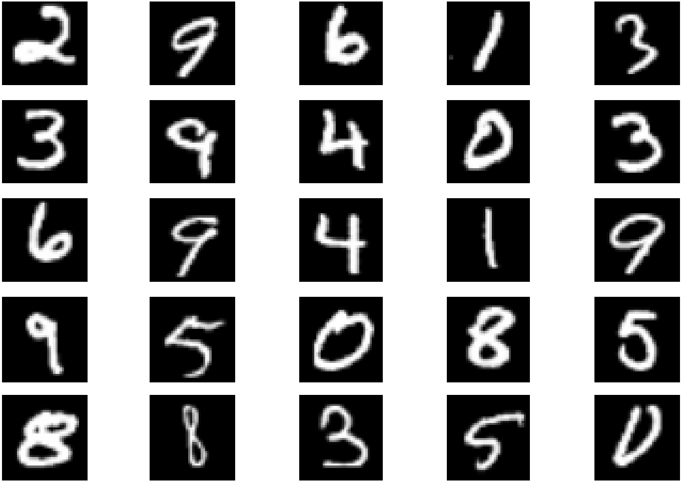
Why use learning?

- 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?

Examples of Classification

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

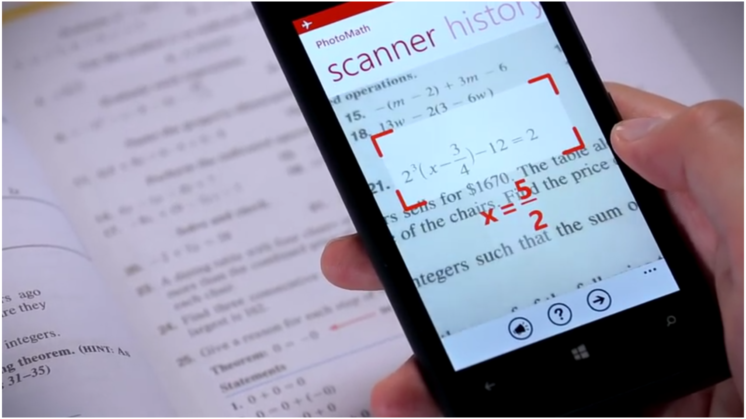
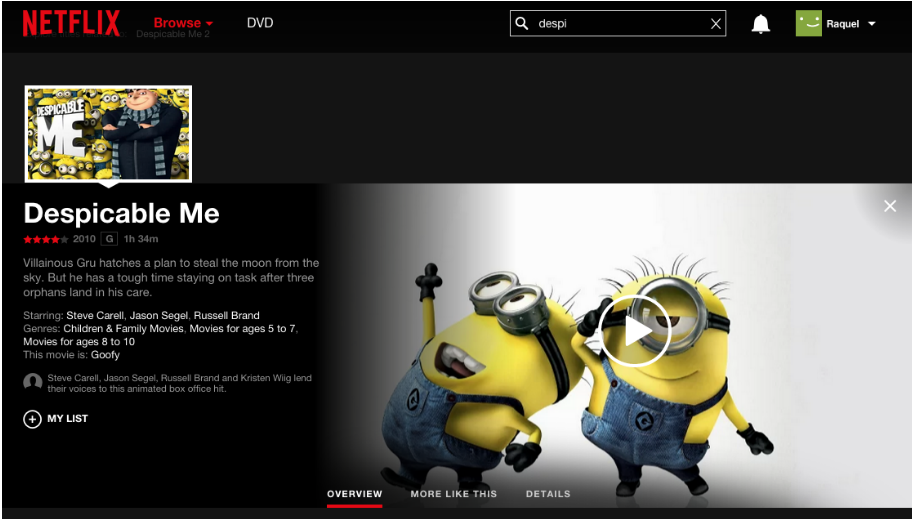


Figure : Photomath: https://photomath.net/

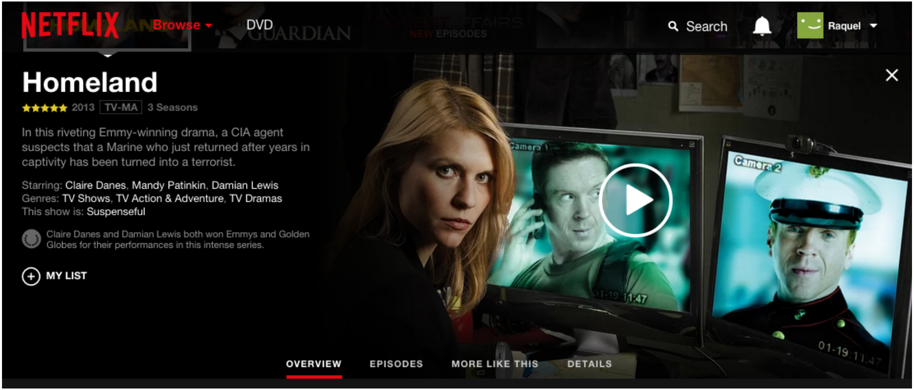
Learning algorithms are useful in other tasks

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).

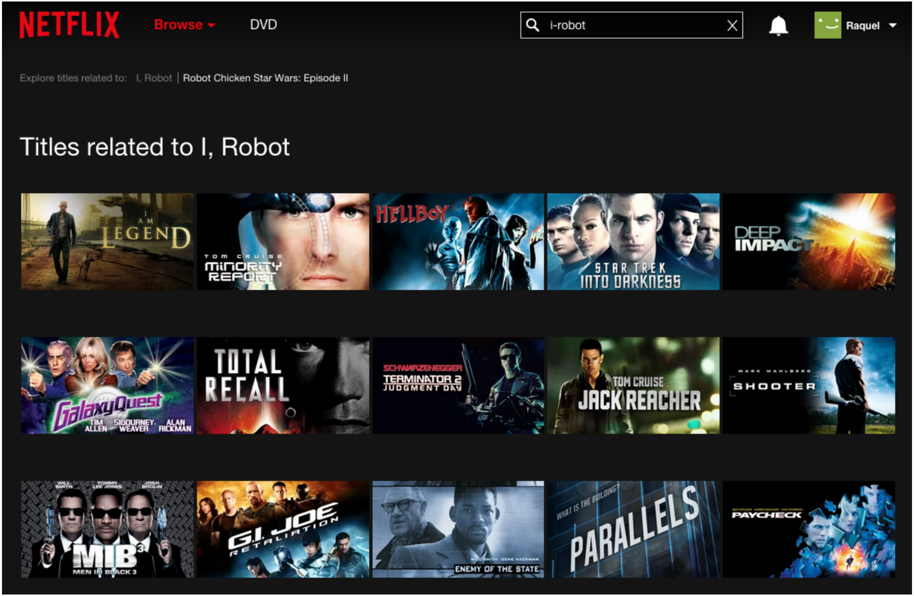
Examples of Recommendation systems



Examples of Recommendation systems



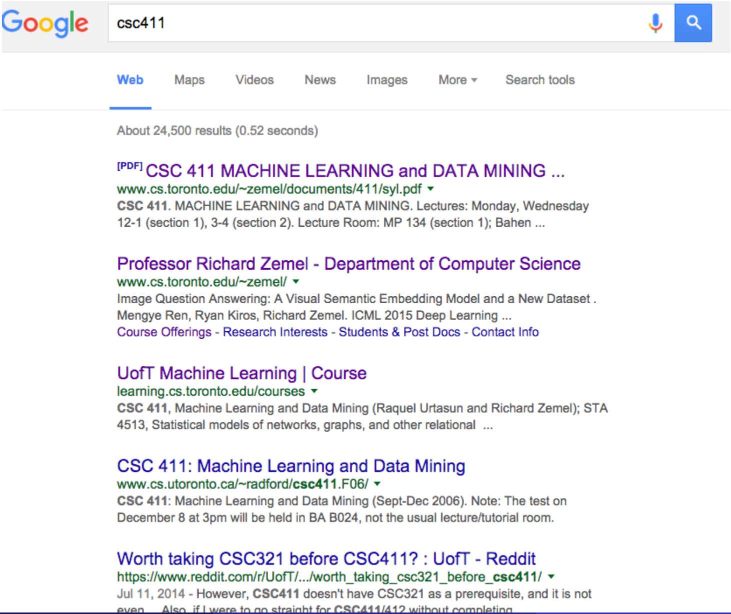
Examples of Recommendation systems



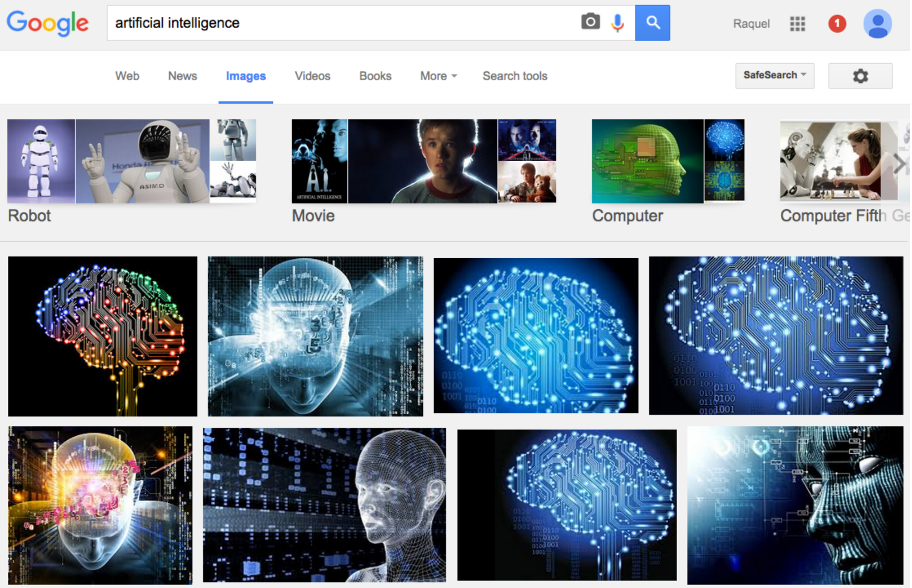
Learning algorithms are useful in other tasks

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

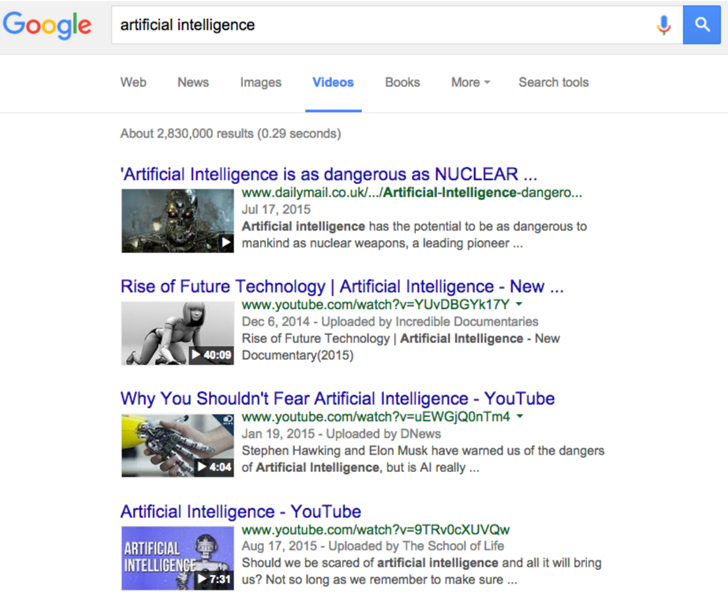
Examples of Information Retrieval



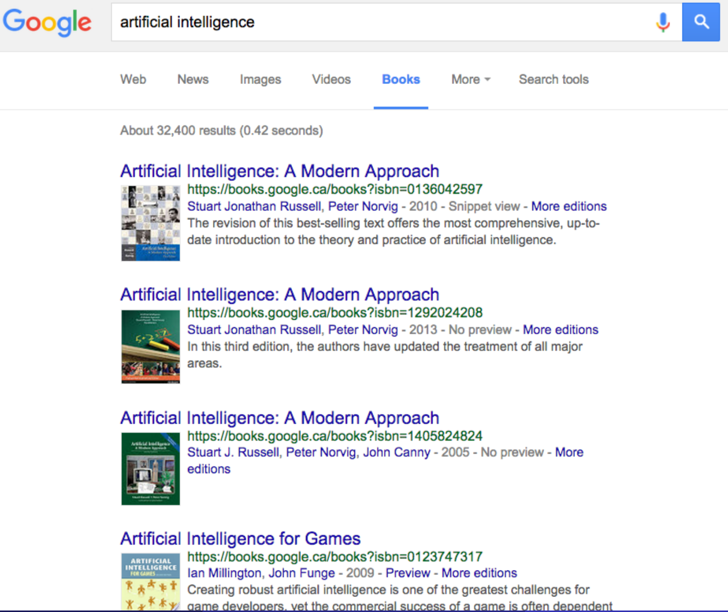
Examples of Information Retrieval



Examples of Information Retrieval



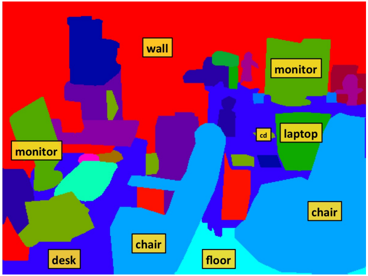
Examples of Information Retrieval



Learning algorithms are useful in other 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

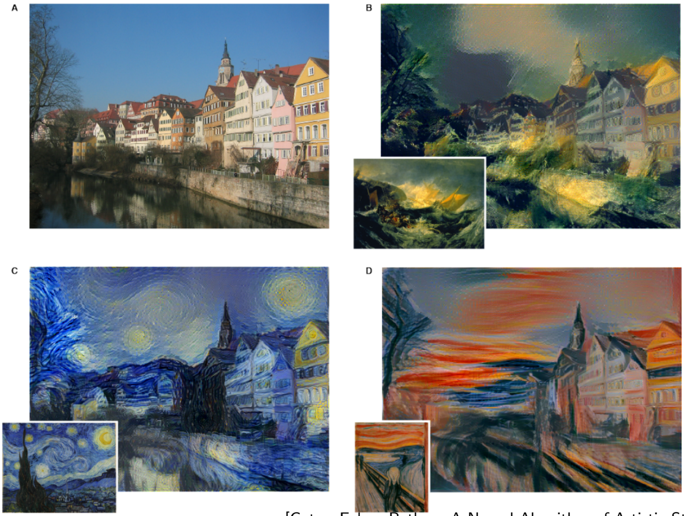
Computer Vision



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 other 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 other 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

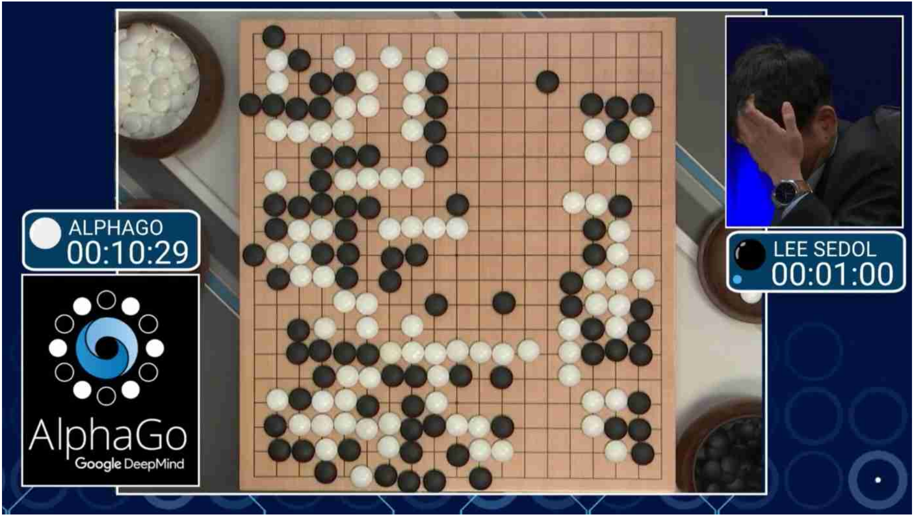
Playing Games: Atari

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

Playing Games: Super Mario

Figure : Video: https://www.youtube.com/watch?v=wfL4L_l4U9A

Playing Games: Alpha Go



Learning algorithms are useful in other 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 other 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 other 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!

Human Learning

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 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
- But problems with AI flavor (e.g., recognition, robot navigation) still domain of ML

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.

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).

Cultural gap (Tibshirani)

MACHINE LEARNING
- weights
- learning
- generalization
- supervised learning
- unsupervised learning
- large grant: $1,000,000
- conference location: Snowbird, French Alps

STATISTICS
- parameters
- fitting
- test set performance
- regression/classification
- density estimation, clustering
- large grant: $50,000
- conference location: Las Vegas in August

Course Survey

Please complete the following survey this week:
https://docs.google.com/forms/d/e/1FAIpQLScd5JwTrh55gW-O-5UKXLidFPvvH-XhVxr36AqfQzsrdDNxGQ/viewform?usp=send_form

Initial Case Study

- What grade will I get in this course?

Initial Case Study

- What grade will I get in this course?
- Data: entry survey and marks from this and previous years

Initial Case Study

- What grade will I get in this course?
- Data: entry survey and marks from this and previous years
- Process the data
  - Split into training set; and test set
  - Determine representation of input;
  - Determine the representation of the output;

Initial Case Study

- What grade will I get in this course?
- Data: entry survey and marks from this and previous years
- Process the data
  - Split into training set; and test set
  - Determine representation of input;
  - Determine the representation of the output;
- Choose form of model: linear regression

Initial Case Study

- What grade will I get in this course?
- Data: entry survey and marks from this and previous years
- Process the data
  - Split into training set; and test set
  - Determine representation of input;
  - Determine the representation of the output;
- Choose form of model: linear regression
- Decide how to evaluate the system's performance: objective function

Initial Case Study

- What grade will I get in this course?
- Data: entry survey and marks from this and previous years
- Process the data
  - Split into training set; and test set
  - Determine representation of input;
  - Determine the representation of the output;
- Choose form of model: linear regression
- Decide how to evaluate the system's performance: objective function
- Set model parameters to optimize performance

Initial Case Study

- What grade will I get in this course?
- Data: entry survey and marks from this and previous years
- Process the data
  - Split into training set; and test set
  - Determine representation of input;
  - Determine the representation of the output;
- Choose form of model: linear regression
- Decide how to evaluate the system's performance: objective function
- Set model parameters to optimize performance
- Evaluate on test set: generalization