# CSC 411: Lecture 01: Introduction

Raquel Urtasun & Rich Zemel

University of Toronto

Sep 14, 2015

# Today

- Administration details
- Why is machine learning so cool?

# Admin Details

- Liberal wrt waiving pre-requisites
  - But it is up to you to determine if you have the appropriate background
- Tutorials:
  - Fridays, same hour as lecture, same place
- 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
- Webpage of the course:
  http://www.cs.toronto.edu/~urtasun/courses/CSC411/CSC411_Fall15.html

# Textbooks

- 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

- Do the readings!
- Assignments:
  - Three assignments, first two worth 12.5% each, last one worth 15%, for a total of 40%
  - Programming: take Matlab/Python code and extend it
  - Derivations: pen(cil)-and-paper
- Mid-term:
  - One hour exam on Oct. 26th
  - Worth 25% of course mark
- Final:
  - Focus on second half of course
  - Worth 35% 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.
- 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.

# Resources

- Course on Piazza at piazza.com/utoronto.ca/fall2015/csc411/home
  - Register to have access at piazza.com/utoronto.ca/fall2015/csc411
  - Communicate announcements
  - Forum for discussion between students
  - Q/A for instructors/TAs and students: We will monitor as much as possible
- Office hours:
  - 1h/week per section
  - TBD exactly when
- Lecture notes, assignments, readings and some announcements will be available on the course webpage

# Calendar

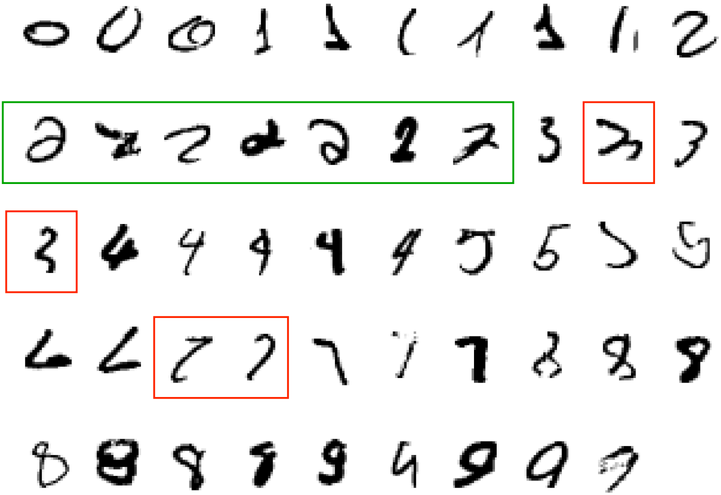
| Date | Topic | Assignments |
|---|---|---|
| Sep 14 | Introduction | |
| Sep 16 | Linear Regression | |
| *Sep 18* | *Probability for ML & Linear regression* | |
| Sep 21 | Linear Classification | |
| Sep 23 | Logistic Regression | |
| *Sep 25* | *Optimization for ML* | |
| Sep 28 | Nonparametric Methods | |
| Sep 30 | Decision Trees | |
| *Oct 2* | *kNN & Decision Trees* | Asst 1 Out |
| Oct 5 | Multi-class Classification | |
| Oct 7 | Probabilistic Classifiers | |
| *Oct 9* | | |
| [Oct 12] | Thanksgiving: No class | |
| Oct 14 | Probabilistic Classifiers II | |
| *Oct 16* | *Naive Bayes and Gaussian Bayes Classifier* | |
| Oct 19 | Neural Networks I | Asst 1 In |
| Oct 21 | Neural Networks II | |
| *Oct 23* | *Mid-term review* | |
| Oct 26 | MIDTERM | |

| Date | Topic | Assignments |
|---|---|---|
| Oct 28 | Clustering | Assit 2 Out |
| *Oct 30* | *Clustering* | |
| Nov 2 | Mixture of Gaussians & EM | |
| Nov 4 | PCA & Autoencoders | |
| *Nov 6* | *PCA Tutorial* | |
| [Nov 9] | Mid-term break: No class | |
| Nov 11 | Kernels and Margins | Asst 2 In |
| *Nov 13* | *SVM Tutorial* | Asst3 Out |
| Nov 16 | Support Vector Machines | |
| Nov 18 | Ensemble Methods I | |
| *Nov 20* | *Bagging & Boosting* | |
| Nov 23 | Ensemble Methods II | |
| Nov 25 | Bayesian Methods | |
| *Nov 27* | | |
| Nov 30 | Reinforcement Learning I | |
| Dec 2 | Reinforcement Learning II | |
| *Dec 4* | | |
| Dec 7 | Final & Wrap-up | Ass 3 In |

# 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 to sample input-output pairs (training examples)
- Learning simply means incorporating information from the training examples into the system

# Task that requires machine learning: What makes a 2?



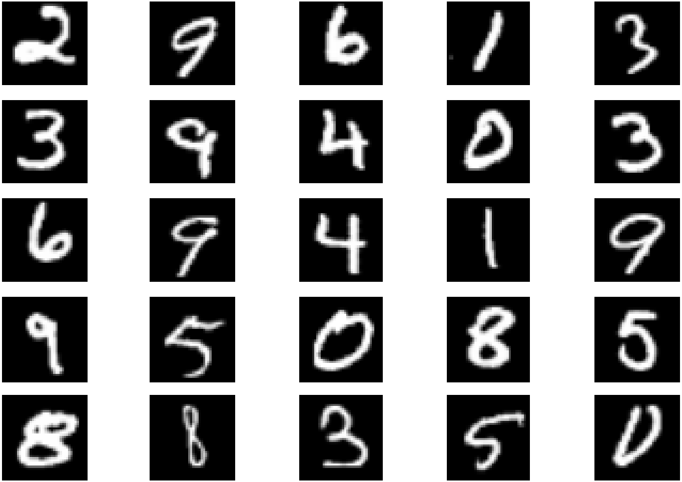
# Why use learning?

- 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 other tasks

1. Classification: Determine which discrete category the example is

# Examples of Classification



# Learning algorithms are useful in other tasks

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

# Examples of Recognizing patterns

# Learning algorithms are useful in other tasks

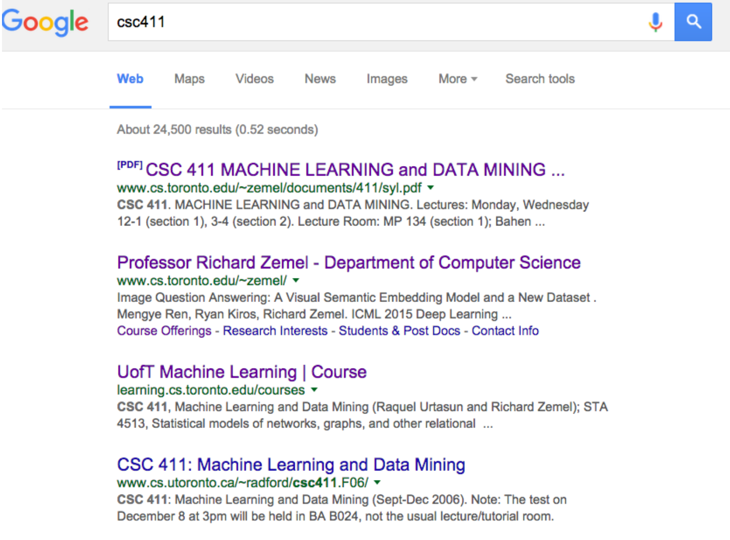
1. Classification: Determine which discrete category the example is
2. Recognizing patterns: Speech Recognition, facial identity, etc
3. Recommender Systems: Noisy data, commercial pay-off (e.g., Amazon, Netflix).

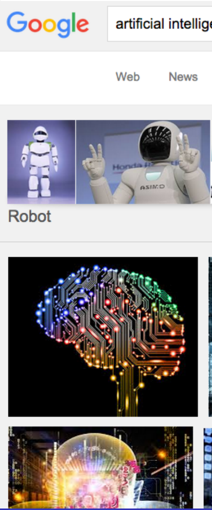
# Examples of Recommendation systems

# 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

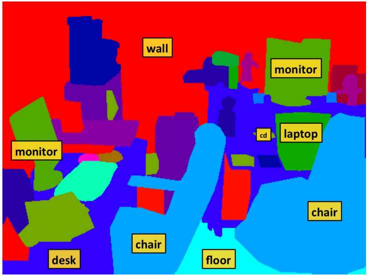


# Learning algorithms are useful in other tasks

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




# 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

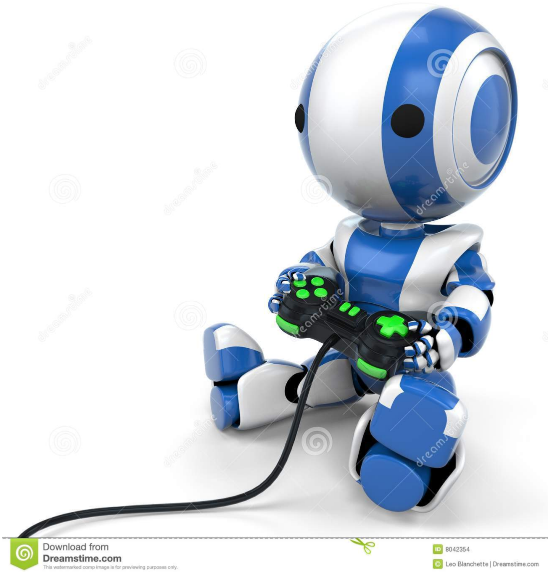

# Flying Robots



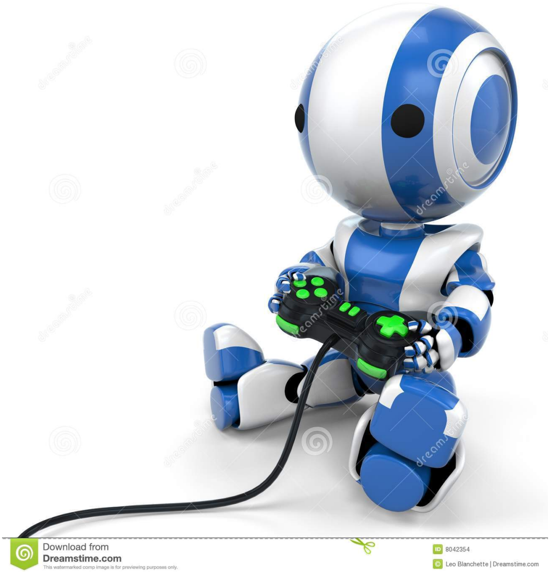

# Learning algorithms are useful in other tasks

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



# Playing Games: Super Mario



# 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 task

- 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
- 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 – core task is inference from a sample
- 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/1O6xRNnKp87GrDM74tkvOMhMIJmwz271TgWdYb6ZitKO/viewform?usp=send_form

# Initial Case Study

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