CSC 411: Introduction to Machine Learning

Lecture 1: Introduction

Mengye Ren and Matthew MacKay

University of Toronto

Course outline

- Broad introduction to machine learning
  - First half: algorithms and principles for supervised learning
    - Nearest neighbors
    - Decision trees
    - Ensembles
    - Linear regression
    - Logistic regression
    - Neural nets
    - SVMs
  - Unsupervised learning:
    - PCA
    - K-means
    - Mixture models
  - Basics of reinforcement learning

Prerequisites

Do I have the appropriate background?

- **Linear algebra**: vector/matrix manipulations, properties.
- **Calculus**: partial derivatives/gradient.
- **Probability**: common distributions; Bayes Rule.
- **Statistics**: expectation, variance, covariance, median; maximum likelihood.

Course Websites

Course Website:
https://www.cs.toronto.edu/~mren/teaching/csc411_19s/

We will use Quercus for **announcements**. You should all have been automatically signed up.

We will use Piazza for **discussions**.

- URL: https://piazza.com/utoronto.ca/winter2019/csc411 (Also sent out through Quercus announcement)
- Your grade **does not depend on your participation on Piazza**. It's just a good way for asking questions, discussing with your instructor, TAs and your peers

Where and When

Instructors:

- Mengye Ren
  - Lecture: SS2135 1-3 PM Tuesday
  - Office Hour: BA2283 3-4 PM Tuesday (after lecture)
- Matthew MacKay
  - Lecture: SF1105 6-8 PM Thursday
  - Office Hour: BA2283 2-3 PM Monday

Emails for administrative purposes only (e.g. medical documentations). For material-related questions, use Piazza or ask your instructor/TA in person during class or office hours.

Course Information

- While cell phones and other electronics are not prohibited in lecture, talking, recording or taking pictures in class is strictly prohibited without the consent of your instructor. Please ask before doing!
- `http://www.illnessverification.utoronto.ca` is the only acceptable form of direct medical documentation.
- For accessibility services: If you require additional academic accommodations, please contact UofT Accessibility Services as soon as possible, `studentlife.utoronto.ca/as`.

Course Information

Recommended readings will be given for each lecture. But the following will be useful throughout the course:

- Hastie, Tibshirani, and Friedman: "The Elements of Statistical Learning"
- Christopher Bishop: "Pattern Recognition and Machine Learning", 2006.
- Kevin Murphy: "Machine Learning: a Probabilistic Perspective", 2012.
- David Mackay: "Information Theory, Inference, and Learning Algorithms", 2003.
- Shai Shalev-Shwartz & Shai Ben-David: "Understanding Machine Learning: From Theory to Algorithms", 2014.

There are lots of freely available, high-quality ML resources.

Requirements and Marking

- 7–8 “weekly” assignments.
  - Combination of pencil & paper derivations and short programming exercises
  - Equally weighted, for a total of 45%
  - Lowest homework mark is dropped
- Read some classic papers.
  - Worth 5%, honor system.
- Midterm
  - Tentative: Feb 15 (outside regular lecture hours)
  - Worth 15% of course mark
- Final Exam
  - Three hours
  - Date and time TBA
  - Worth 35% of course mark

More on Assignments

- Collaboration on the assignments is not allowed.
- 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 will be posted on the course web page.
- Assignments should be handed in by 11:59pm; a late penalty of 10% per day for a maximum of 3 days.
- Extensions will be granted only in special situations
- You will need a Student Medical Certificate or a written request approved at least one week before the due date.

Related Courses

- **CSC 421** (neural nets) and **CSC 412** (probabilistic graphical models) both build upon the material in this course.
- If you've already taken **CSC 321**, there will be 3–4 weeks of redundant material.

What is learning?

"The activity or process of gaining knowledge or skill by studying, practicing, being taught, or experiencing something."

Merriam Webster dictionary

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Tom Mitchell

Applications of machine learning

- For many problems, it's difficult to program the correct behavior by hand
  - recognizing people and objects
  - understanding human speech
- Machine learning approach: program an algorithm to automatically learn from data, or from experience
- Why might you want to use a learning algorithm?
  - hard to code up a solution by hand (e.g. vision, speech)
  - system needs to adapt to a changing environment (e.g. spam detection)
  - want the system to perform *better* than the human programmers
  - privacy/fairness (e.g. ranking search results)

Relations to statistics

- It's similar to statistics...
  - Both fields try to uncover patterns in data
  - Both fields draw heavily on calculus, probability, and linear algebra, and share many of the same core algorithms
- But it's not statistics...
  - Stats is more concerned with **helping scientists and policymakers** draw good conclusions; ML is more concerned with **building autonomous agents**
  - Stats puts more emphasis on **interpretability and mathematical rigor**; ML puts more emphasis on predictive **performance, scalability, and autonomy**

Relations to AI

- Nowadays, "machine learning" is often brought up with "artificial intelligence" (AI)
- AI does not often imply a learning based system
  - Symbolic reasoning
  - Rule based system
  - Tree search
  - etc.
- Learning based system $\rightarrow$ more free learnable parameters $\rightarrow$ learned based on the data $\rightarrow$ more flexibility, good at solving pattern recognition problems.

Relations to human learning

- It is tempting to imagine machine learning as a component in AI just like human learning in ourselves.
- Human learning is:
  - Very data efficient
  - An entire multitasking system (vision, language, motor control, etc.)
  - Takes at least a few years :)
- For serving specific purposes, machine learning doesn't have to look like human learning in the end.
- It may borrow ideas from biological systems (e.g. neural networks).
- There may also be biological constraints.

What is machine learning?

- Types of machine learning
  - **Supervised learning:** have labeled examples of the correct behavior
  - **Reinforcement learning:** learning system receives a reward signal, tries to learn to maximize the reward signal
  - **Unsupervised learning:** no labeled examples – instead, looking for interesting patterns in the data

History of machine learning

- 1957 — Perceptron algorithm (implemented as a circuit!)
- 1959 — Arthur Samuel wrote a learning-based checkers program that could defeat him
- 1969 — Minsky and Papert's book *Perceptrons* (limitations of linear models)
- 1980s — Some foundational ideas
  - Connectionist psychologists explored neural models of cognition
  - 1984 — Leslie Valiant formalized the problem of learning as PAC learning
  - 1988 — Backpropagation (re-)discovered by Geoffrey Hinton and colleagues
  - 1988 — Judea Pearl's book *Probabilistic Reasoning in Intelligent Systems* introduced Bayesian networks

History of machine learning

- 1990s — the “AI Winter”, a time of pessimism and low funding
- But looking back, the ’90s were also sort of a golden age for ML research
    - Markov chain Monte Carlo
    - Variational inference
    - Kernels and support vector machines
    - Boosting
    - Convolutional networks
- 2000s — applied AI fields (vision, NLP, etc.) adopted ML
- 2010s — deep learning
    - 2010–2012 — neural nets smashed previous records in speech-to-text and object recognition
    - increasing adoption by the tech industry
    - 2016 — AlphaGo defeated the human Go champion

History of machine learning

A top ML conference, NeurIPS (used to be called NIPS), sold out the ticket for the meeting happened in December 2018.

Computer vision: Object detection, semantic segmentation, pose estimation, and almost every other task is done with ML.

Figure 4. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

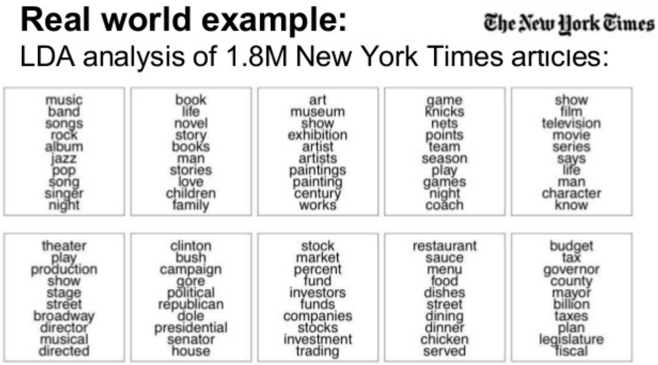
Instance segmentation - Link

DAQUAR 1553
What is there in front of the sofa?
Ground truth: table
IMG+BOW: table (0.74)
2-VIS+BLSTM: table (0.88)
LSTM: chair (0.47)

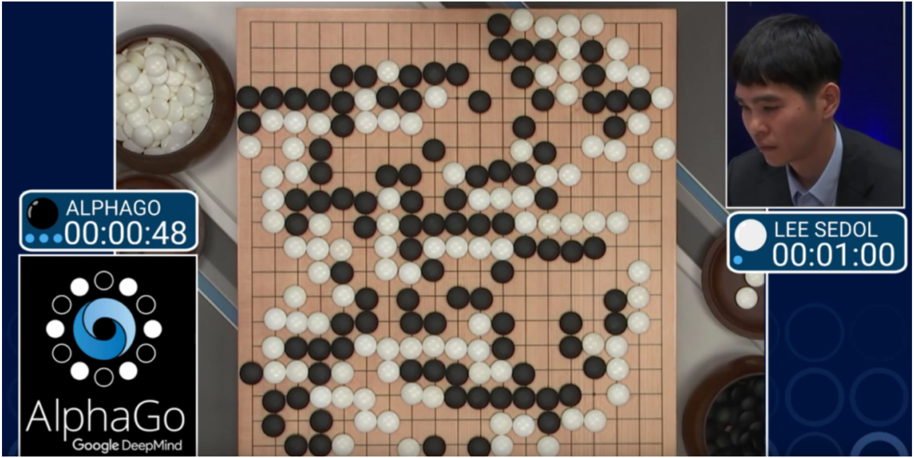
COCOQA 5078
How many leftover donuts is the red bicycle holding?
Ground truth: three
IMG+BOW: two (0.51)
2-VIS+BLSTM: three (0.27)
BOW: one (0.29)

Speech: Speech to text, personal assistants, speaker identification...

NLP: Machine translation, sentiment analysis, topic modeling, spam filtering.

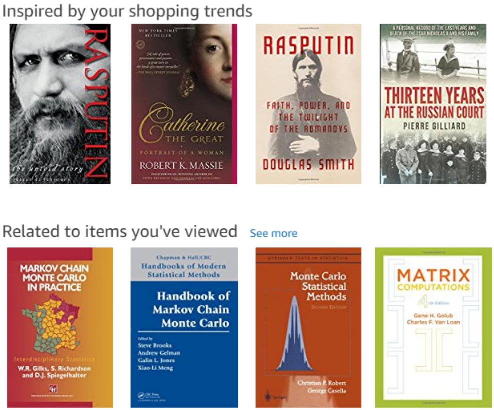


Playing Games



DOTA2 - Link

E-commerce & Recommender Systems : Amazon, netflix, ...



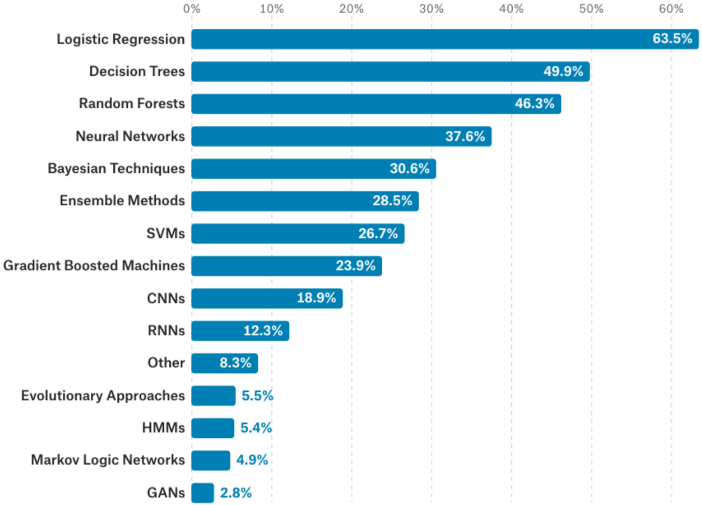
Why taking this class?

Why not just learn neural nets first (like CSC 421)?

- The techniques in this course are still the first things to try for a new ML problem.
  - E.g., try logistic regression before building a deep neural net!
- The principles you learn in this course will be essential to really understand neural nets.
  - 3–4 weeks of csc321 were devoted to background material covered in this course!
- A better foundation for CSC 412 (probablistic graphical models)

Why this class?

2017 Kaggle survey of data science and ML practitioners: what data science methods do you use at work?



ML Workflow

ML workflow sketch:

1. Should I use ML on this problem?
   - Is there a pattern to detect?
   - Can I solve it analytically?
   - Do I have data?
2. Gather and organize data.
3. Preprocessing, cleaning, visualizing.
4. Establishing a baseline.
5. Choosing a model, loss, regularization, ...
6. Optimization
7. Hyperparameter search.
8. Analyze performance and mistakes, and iterate back to step 5 (or 3).

Implementing machine learning systems

- You will often need to derive an algorithm (with pencil and paper), and then translate the math into code.
- Array processing (NumPy)
  - **vectorize** computations (express them in terms of matrix/vector operations) to exploit hardware efficiency
  - This also makes your code cleaner and more readable!

$z = Wx + b$

```
z = np.zeros(m)
for i in range(m):
    for j in range(n):
        z[i] += W[i, j] * x[j]
    z[i] += b[i]
```

```
z = np.dot(W, x) + b
```

Implementing machine learning systems

- Neural net frameworks: PyTorch, TensorFlow, Theano, etc.
  - Automatic differentiation
  - Compiling computation graphs
  - Libraries of algorithms and network primitives
  - Support for graphics processing units (GPUs)
- Why take this class if these frameworks do so much for you?
  - Understanding what goes on beneath the hood.
  - You will know what to do if something goes wrong.