CSC 411: Introduction to Machine Learning

Lecture 1 - Introduction

Roger Grosse, Amir-massoud Farahmand, and Juan Carrasquilla

University of Toronto

This course

- Broad introduction to machine learning
  - First half: algorithms and principles for supervised learning
    - nearest neighbors, decision trees, ensembles, linear regression, logistic regression, SVMs
  - Unsupervised learning: PCA, K-means, mixture models
  - Basics of reinforcement learning
- Coursework is aimed at advanced undergrads, but we'll try to keep things interesting for the grad students.

Course Information

Course Website:
`https://www.cs.toronto.edu/~rgrosse/courses/csc411_f18/`

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

- Did you receive the announcement on Thursday?

We will use Piazza for **discussions**.

- URL to be sent out
- 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

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

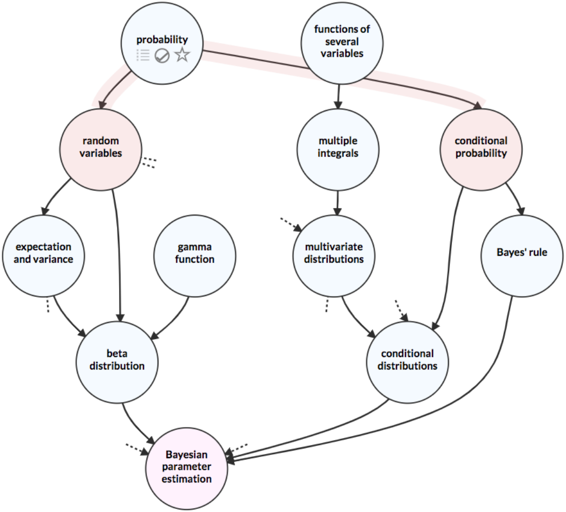
Recommended readings will be given for each lecture. But the following will be useful throughout the course:

- 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.

Course Information

See Metacademy (`https://metacademy.org`) for additional background, and to help review prerequisites.



Requirements and Marking (Undergraduates)

- 8–10 “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
    - Oct. 19, 6–7pm
    - Worth 15% of course mark
- Final Exam
    - Three hours
    - Date and time TBA
    - Worth 35% of course mark

Final Projects (Grad Students Only)

- Grad students may choose between the following:
  - Follow the undergrad requirements (the path of least resistance)
  - Replace the second half of the weekly homeworks with a final project (for those who are excited about getting research experience)
- The project is meant to be a small research project, comparable to a workshop submission.
  - You must work in groups of 2–3.
- **Everybody must take the final exam!**
- Marking scheme if you choose the final project:
  - 25% project
  - 20% weekly homeworks (Homeworks 1 through 4)
  - 15% midterm
  - 35% final exam
  - 5% readings (honor system)

More on Assignments

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 will be posted on the course web page.

Assignments should be handed in by 11:59pm; a late penalty of 10% per day will be assessed thereafter (up to 3 days, then submission is blocked).

Extensions will be granted only in special situations, and you will need a Student Medical Certificate or a written request approved by the course coordinator at least one week before the due date.

Related Courses

- **csc421** (neural nets) and **csc412** (probabilistic graphical models) both build upon the material in this course.
- If you've already taken **csc321**, there will be 3–4 weeks of redundant material. Sorry.
- We will probably stop cross-listing this as an undergrad and grad course. Next year, we expect to split **csc2515** off into a stand-alone grad course.

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

What is 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)

What is machine learning?

- 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

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

- Early developments
  - 1957 — perceptron algorithm (implemented as a circuit!)
  - 1959 — Aurthur 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

We passed a dubious milestone on Tuesday:

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.

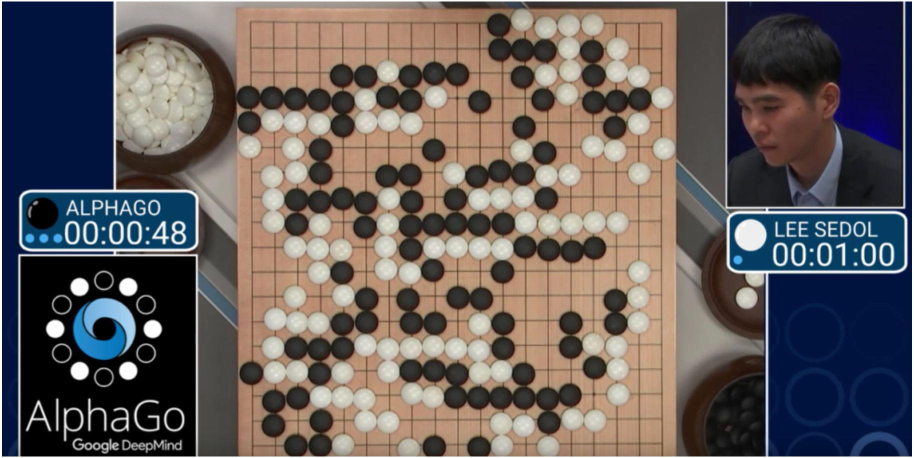
Real world example: The New York Times

LDA analysis of 1.8M New York Times articles:

music	book	art	game	show
band	life	museum	knicks	film
songs	novel	show	nets	television
rock	story	exhibition	points	movie
album	books	artist	team	series
jazz	man	artists	season	says
pop	stories	paintings	play	life
song	love	painting	games	man
singer	children	century	night	character
night	family	works	coach	know

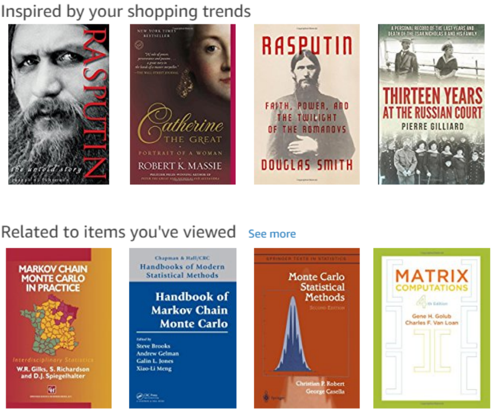
theater	clinton	stock	restaurant	budget
play	bush	market	sauce	tax
production	campaign	percent	menu	governor
show	gore	fund	food	county
stage	political	investors	dishes	mayor
street	republican	funds	street	billion
broadway	dole	companies	dining	taxes
director	presidential	stocks	dinner	plan
musical	senator	investment	chicken	legislature
directed	house	trading	served	fiscal

Playing Games



DOTA2 - Link

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



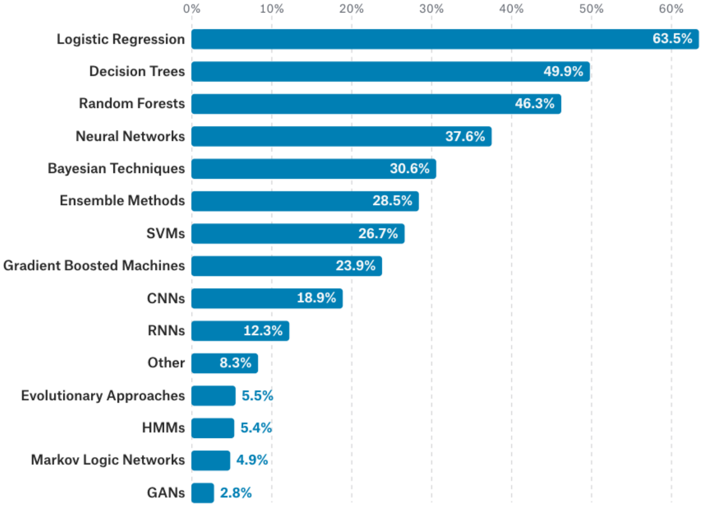
Why this class?

Why not jump straight to csc421, and learn neural nets first?

- 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!
- There's a whole world of probabilistic graphical models.

Why this class?

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



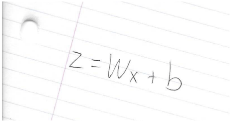
ML Workflow

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 (could be simple, could be a Phd...).
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!




```python
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]
```

```python
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?
  - So you know what to do if something goes wrong!
  - Debugging learning algorithms requires sophisticated detective work, which requires understanding what goes on beneath the hood.
  - That's why we derive things by hand in this class!

Questions?

?