CSC 411: Introduction to Machine Learning

Lecture 1 - Introduction

Ethan Fetaya, James Lucas and Emad Andrews

University of Toronto

Today

- Administration details
- Why is machine learning so cool?

The Team I

Instructors:

- Ethan Fetaya
  - Section 1 (AH 400) Mon. 11am-1pm (tutorials Mon. 3-4pm)
  - Section 2 (OI 2212) Wed. 11am-1pm (tutorials Wed. 3-4pm)
  - Office Hours : Prat 290A (for now). 11-12pm Tuesday, 8-10am Wednesday.
- Emad Andrews
  - Section 3 (MP 103) Thursday. 4-6pm (tutorials Thu. 6-7pm)
  - Office Hours : BA3219 6-8pm Thursday
- James Lucas
  - Section 4 (RW 117) Friday. 11-1pm (tutorials Fri. 3-4pm)
  - Office Hours : TBD

email : csc411-20179-instrs@cs.toronto.edu

- Please send 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.
- You must use an academic email account when sending us emails. Otherwise, they might be filtered as spam and deleted automatically.

The Team II

TA's

- Eleni Triantafillou
- Aryan Arbabi
- Ladislav Rampasek
- Jixuan Wang
- Yingzhou Wu
- Shengyang Sun
- Tian Qi Chen
- Chris Cremer
- Yulia Rubanova
- Bowen Xu
- Seyed Kamyar Seyed Ghasemipour
- Tingwu Wang
- Harris Chan
- Bettencourt Jesse

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/gradient.
- Probability: common distributions; Bayes Rule.
- Statistics: expectation, variance, covariance, median; maximum likelihood.

Course Information

Course Website:

`http://www.cs.toronto.edu/~jlucas/teaching/csc411`

- you are expected to check course website regularly. All announcements posted are considered to have been announced to the class and not having read or seen an announcement is not an accepted reason for not following guidelines or missing deadlines

The class will use Piazza for **announcements** and **discussions**:

- `https://piazza.com/class/fall2017/csc411`
- First time, sign up here: `https://piazza.com/utoronto.ca/csc411`
- Your grade **does not depend on your participation on Piazza**. Its just a good way for asking questions, discussing with your instructor, TAs and your peers

More on 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

Textbooks:

- Christopher Bishop: Pattern Recognition and Machine Learning, 2006 (main textbook).
- 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.

Requirements (Undergrads)

- Do the readings!
- Read 5 classic papers.
  - 5 points.
  - Honor system.
- Assignments.
  - Three assignments, worth 15% each, for a total of 45%.
  - Programming: take Python code and extend it.
  - Derivations: pen(cil)-and-paper
- Mid-term:
  - One hour exam on week of Oct. 12 - Oct. 18
  - Worth 20% of course mark.
- Final:
  - Focused on second half of course.
  - Worth 30% of course mark.

Requirements (Grads)

- Do the readings!
- Read 5 classic papers.
  - 5 points.
  - Honor system.
- Assignments.
  - Three assignments, worth 15% each, for a total of 45%.
  - Programming: take Python code and extend it.
  - Derivations: pen(cil)-and-paper
- Mid-term:
  - One hour exam on week of Oct. 12 - Oct. 18
  - Worth 20% of course mark.
- Project:
  - Worth 30% of course mark.

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 is included in the syllabus.

Assignments should be handed in by 10 pm; 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.

Provisional Calendar

- Sept. 7-13 :
  - Introduction; Linear regression
- Sept. 14-20 :
  - Linear classification & Logistic regression
- Sept. 21-27:
  - Nearest neighbor & Decision trees,
  - Assignment 1 release on Sept. 21
- Sept. 28-Oct. 4 :
  - Multi-class classification & Probabilistic Classifiers I,
  - Reading assignment 1 release
- Oct. 5-11 :
  - Probabilistic Classifiers II & Neural Networks I,
  - Assignment 1 due on Oct. 5 & Reading assignment 2 release

Provisional Calendar II

- Oct. 12-18 :
  - Neural Networks II & PCA,
  - Midterm
- Oct. 19-25:
  - t-SNE & Clustering,
  - Assignment 2 release on Oct. 19
- Oct. 26-Nov. 1:
  - Mixture of Gaussian & EM,
  - Reading assignment 3 release
- Nov. 2-Nov8 :
  - Nov 6-10 Reading week
  - Assignment 2 due on Nov. 2

Provisional Calendar III

- Nov. 9-15 :
  - SVM & Kernels
  - Assignment 3 release on Nov.13 & Reading assignment 4 release
- Nov. 16-22:
  - Ensembles Learning
- Nov. 23-29:
  - Reinforcement learning
  - Assignment 3 due on Nov. 27
- Nov. 30-Dec. 7:
  - Learning theory;
  - Reading assignment 5 release
- Dec. 9 - 20:Final Exam Period

What is learning?

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

Merriam Webster dictionary

$ML \neq AI$.

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
- However, In many cases is very difficult to specify those rules
  - Some tasks (vision, speech, NLP) are too complicated to code.
  - Some systems need to adapt.
  - Handle noise.
  - Etc.

Instead of explicitly writing a program to solve a specific problem, we use examples (*training data*) to train the computer to perform this task *(to generalize)*.

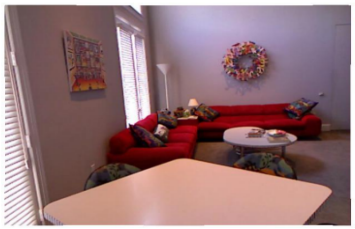
What is machine learning?

- 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

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 assistance, speaker identification...

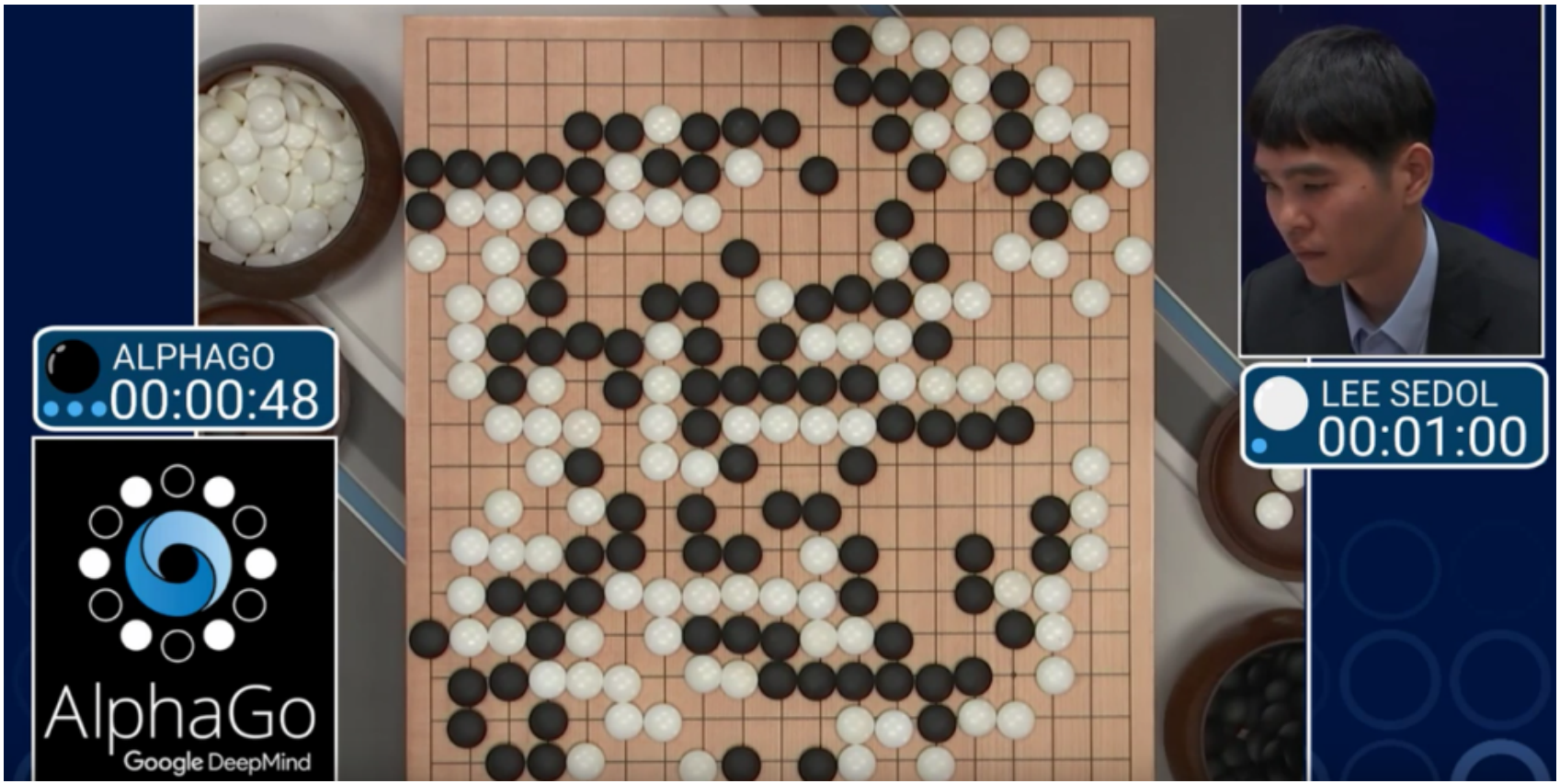
NLP: Machine translation, sentiment analysis, topic modeling, spam filtering.

Real world example: The New York Times

LDA analysis of 1.8M New York Times articles:

music band songs rock album jazz pop song singer night	book life novel story books man stories love children family	art museum show exhibition artist artists paintings painting century works	game knicks nets points team season play games night coach	show film television movie series says life man character know
theater play production show stage street broadway director musical directed	clinton bush campaign gore political republican dole presidential senator house	stock market percent fund investors funds companies stocks investment trading	restaurant sauce menu food dishes street dining dinner chicken served	budget tax governor county mayor billion taxes plan legislature fiscal

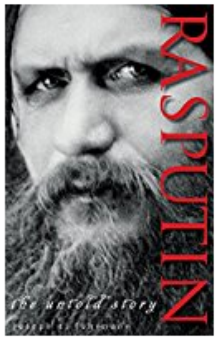
Playing Games

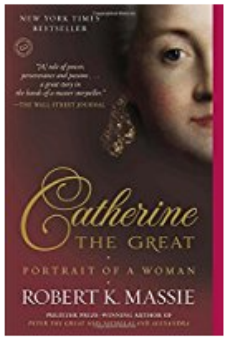


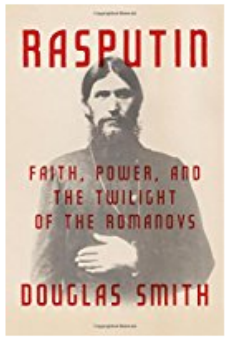
DOTA2 - ▸ Link

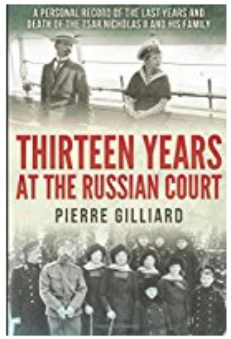
E-commerce & Recommender Systems : Amazon, netflix, ...

Inspired by your shopping trends

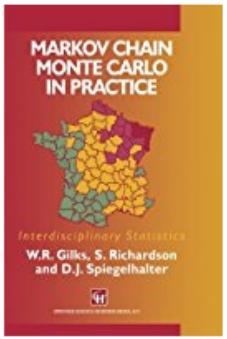


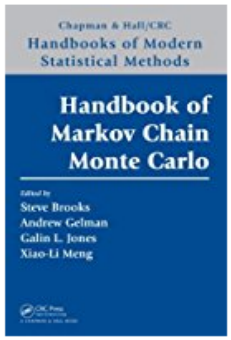


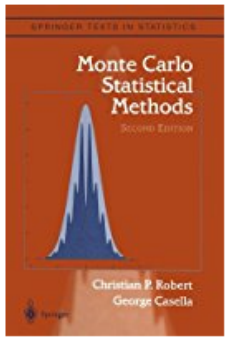


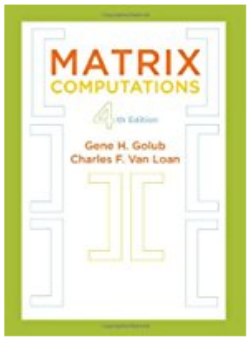


Related to items you've viewed See more









ML broad categories:

- Supervised learning (correct outputs known). Given (x, y) pairs learn a mapping from x to y. Example: Sentiment analysis.
  - Classification: categorical output (object recognition, medical diagnosis)
  - Regression: real-valued output (predicting market prices, customer rating)
- Unsupervised learning. Given data points find some structure in the data. Example: Dimensionality reduction.
- Online learning. Supervised learning when the data is given sequentially, by an adversary, No separate train/test phases. Example: Spam filtering.
- Reinforcement learning. Learn actions to maximize future rewards. Delayed playoffs, agent controls what he sees. Example: Flying drones.
- Various smaller categories, e.g. active learning, semi-supervised learning.

Supervised learning mathematical set-up:

- An input space $\mathcal{X}$. Examples: $\mathbb{R}^n$, images, texts, sound recordings, etc.
- An output space $\mathcal{Y}$. Examples: $\{\pm 1\}$, $\{1, ..., k\}$, $\mathbb{R}$.
- An **unknown** distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$.
- A loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Examples: $0-1$ loss, square loss.
- A set of m **i.i.d** samples $(x_1, y_1), ..., (x_m, y_m)$ sampled from the distribution $\mathcal{D}$.

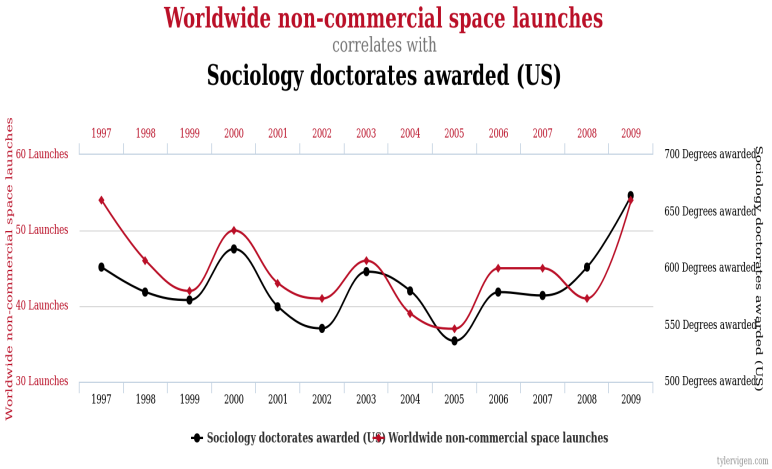
The goal: return a function (hypothesis) $h : \mathcal{X} \to \mathcal{Y}$ that minimizes the expect loss (risk) with respect to $\mathcal{D}$ i.e. find h that minimizes $L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)]$

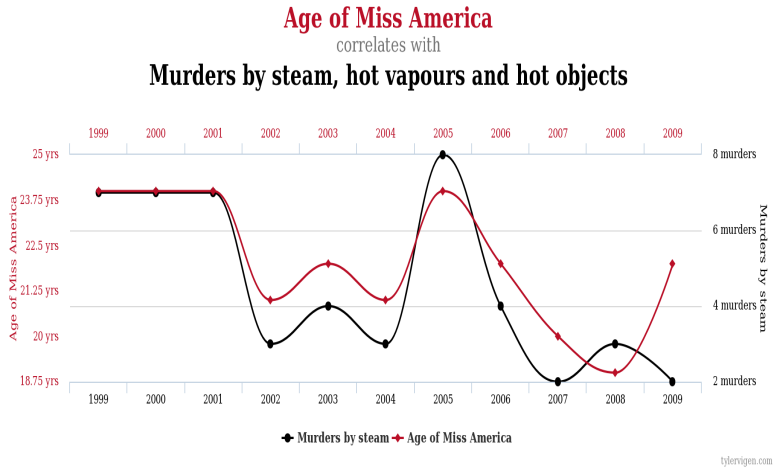
We want to minimize $L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)]$, but we don't know $L_{\mathcal{D}}$. We can approximate it by the *empirical loss* $L_S(h) = \frac{1}{n}\sum_{i=1}^{m} \ell(h(x_i), y_i)$

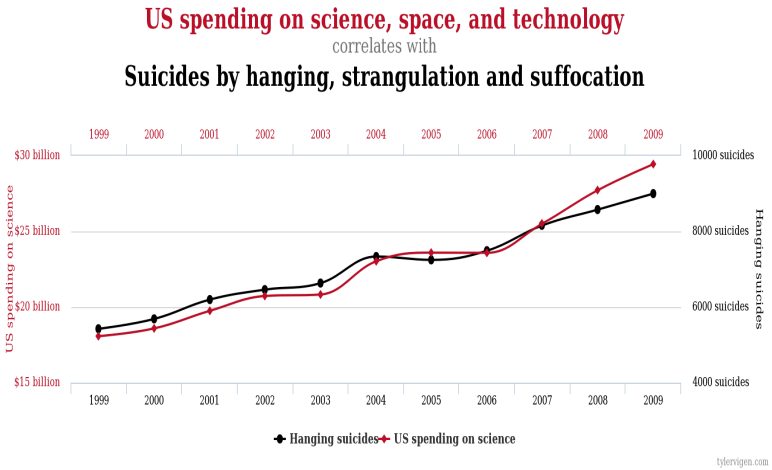
For a specific function h, $L_S(h) \approx L_{\mathcal{D}}(h)$, but if we try to fit a very complex model we might find a solution that works on our training examples and doesn't *generalize* to other examples. That means we *overfit*.

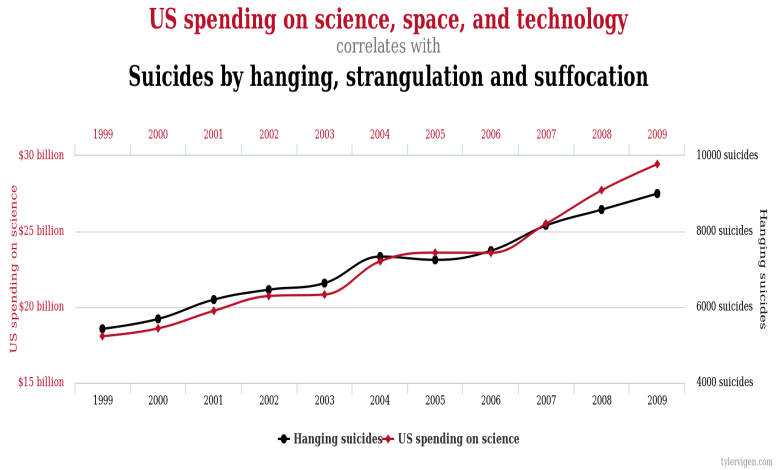
The main challenge: Find a model that is rich enough to find the patterns in your data, but does not fit random noise in our data.

"If you torture the data long enough, it will confess." -Ronald Coase









Images taken from spurious correlations

ML viewpoints:

- Agnostic approach. Trying to minimize loss on unseen data.
- Discriminative approach. Fit $P(y|x;\theta)$ by some parametric model.
- Generative approach. Fit $P(x,y;\theta)$ by some parametric model, and use it to determine $P(y|x;\theta)$.
- Bayesian approach. Instead of a single model θ we have a distribution over θ, $p(\theta)$ so $p(y|x) = \int p(y|x,\theta)p(\theta)$

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
- 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).

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).

Questions?

?