COMP 551 – Applied Machine Learning
Lecture 1: Introduction

Instructor: Herke van Hoof (*herke.vanhoof@mail.mcgill.ca)*

Slides mostly by: Joelle Pineau

Class web page: *www.cs.mcgill.ca/~hvanho2/comp551*

Outline for today

- Overview of the syllabus
- Summary of course content
- Broad introduction to Machine Learning (ML)
- Examples of ML applications

Course objectives

- To develop an understanding of the fundamental concepts of ML.
  - Algorithms, models, practices.
- **To emphasize good methods and practices for effective deployment of real systems.**
- To acquire hands-on experience with basic tools, algorithms and datasets.

About you

117 enrolled, 56 waitlist, primarily from:

- Computer Science, Computer Engineering (**approx. 50%**)
- Electrical, Software, Mechanical, Mining Engineering

… and a few from:

- Physics, Linguistics, Economics, Psychology, Philosophy, Biology, Math, Neuroscience, Human Genetics, Materials Engineering, Information Studies

Approx. 10% PhD, 30% Masters, 60% Bachelors candidates.

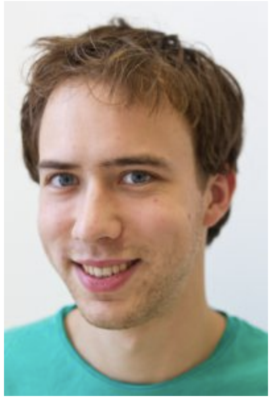
About the course instructors

- Ryan Lowe

- Currently pursuing a PhD
  in the reasoning and learning lab

- Ryan's research interests
  - Deep learning
  - Reinforcement learning
  - Multi-agent communication
  - Dialogue systems, generative models for natural language
  - Causal models

About the course instructors



- Herke van Hoof
- Post-doctoral researcher in the reasoning and learning lab
- Herke's research interests
  - Reinforcement learning
  - Active learning
  - Robotics
  - Sensing

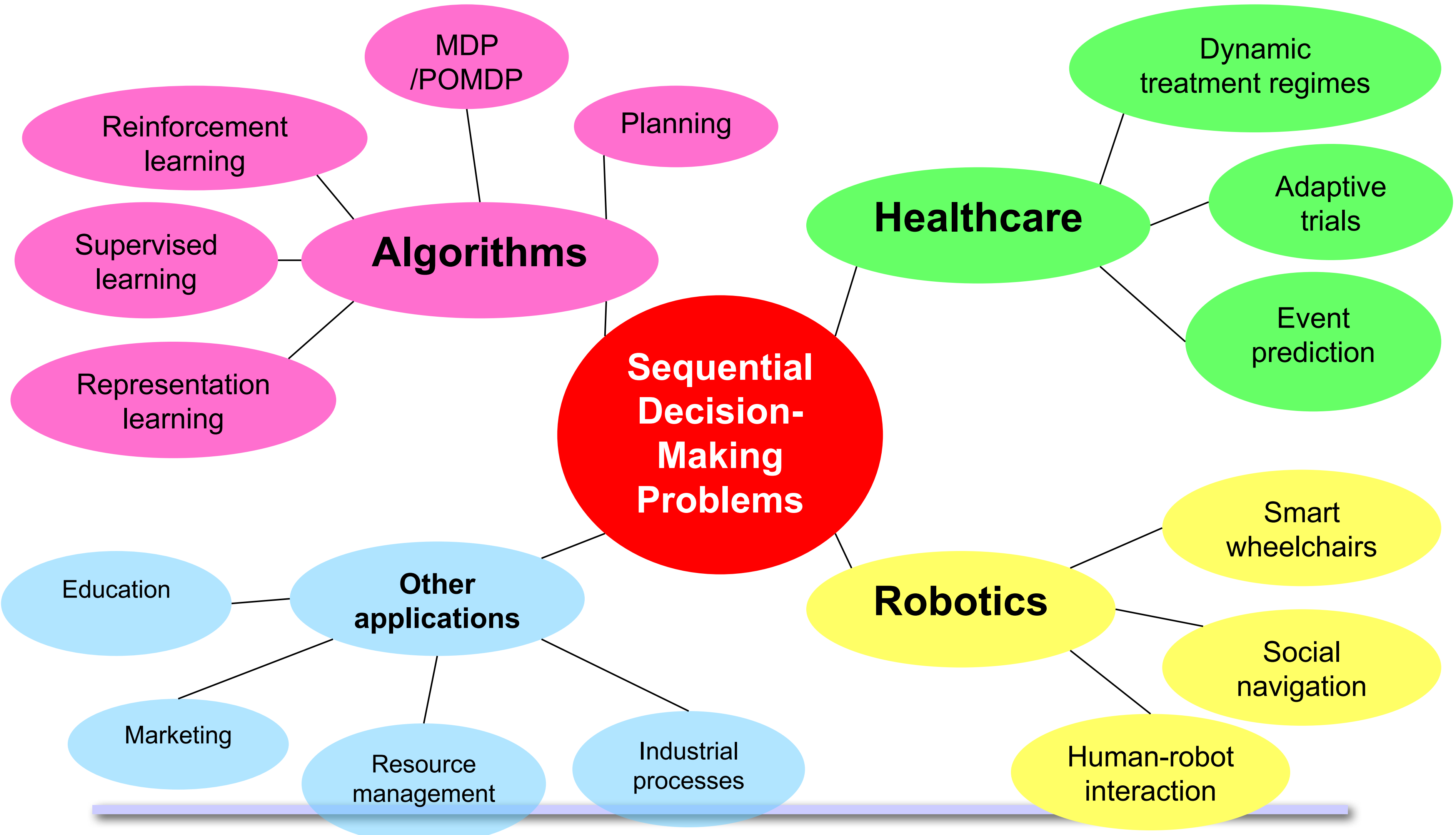
The rest of the teaching team

- TA's:
  - Harsh Satija
  - Sanjay Thakur
  - Lucas Page-Caccia
  - Ali Emami

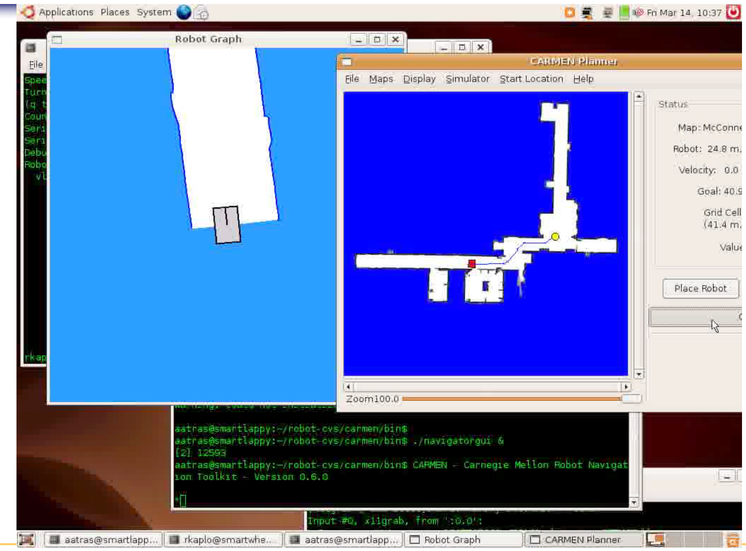
- See the course website for contact details and office hours!

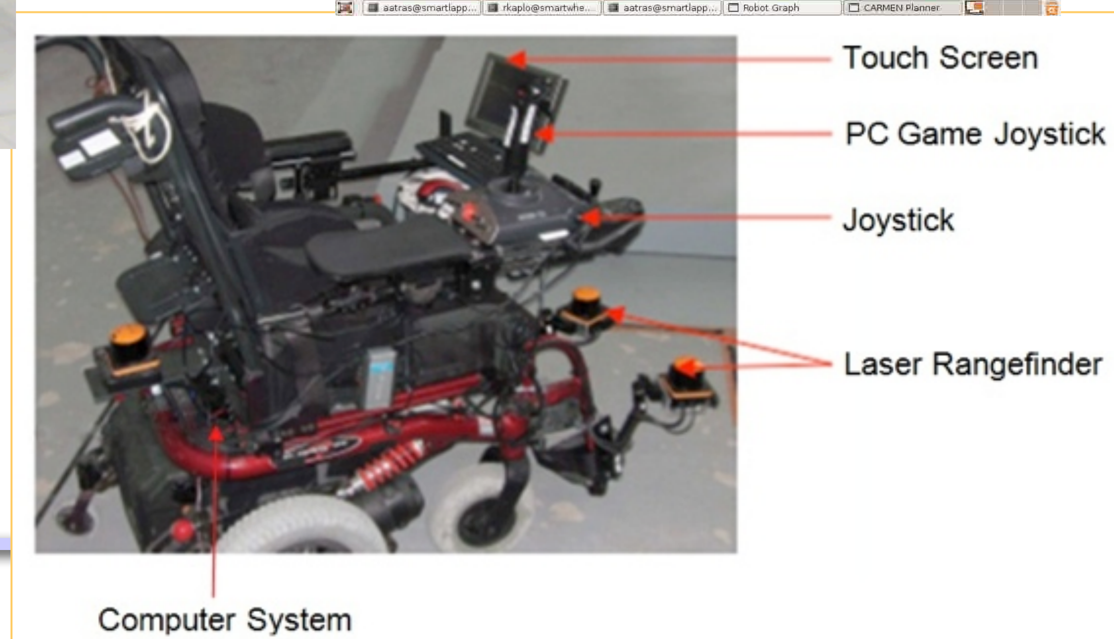
  cs.mcgill.ca/~hvanho2/comp551

Research areas in the lab



From the lab to the real world





About the course: Two sections

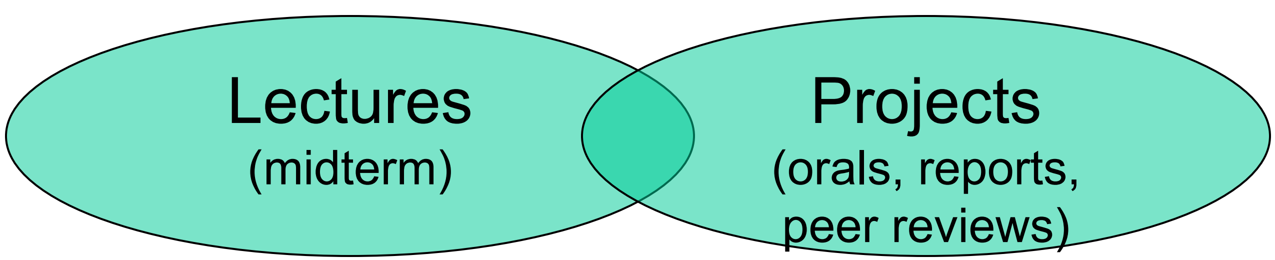
- This is Section 2
- The sections will have the same midterm and the same assignments
- Order of lectures will be slightly different
- Direct questions and submit work to the section that you're enrolled in!

About the course: Tentative list of topics

- Linear regression.
- Linear classification.
- Performance evaluation, overfitting, cross-validation, bias-variance analysis, error estimation.
- Dataset analysis.
- Naive Bayes.
- Decision and regression trees.
- Support vector machines.
- Neural networks.
- Deep learning.
- Unsupervised learning and clustering.
- Feature selection.
- Dimensionality reduction.
- Regularization.
- Data structures and Map-Reduce.
- Ensemble methods.
- Cost-sensitive learning.
- Online / streaming data.
- Time-series analysis.
- Semi-supervised learning.
- Recommendation systems.
- Applications.

About the course

- During class:
  - Primarily lectures
- Outside of class:
  - 4 optional tutorial sessions.
  - Complete 5 projects, peer review work of colleagues, review your notes, read papers, watch videos.



IMPORTANT!
These target <u>different</u>, but complementary, knowledge & skills!

About the course

- **Prerequisites**:
  - Knowledge of a programming language (Matlab, R are ok; Python is best.)
  - Knowledge of probabilities/statistics (e.g. MATH-323, ECSE-305).
  - Knowledge of calculus and linear algebra.
  - Some AI background is recommended (e.g. COMP-424, ECSE-526) but not required.

- **Anterequisites**:
  - If you took COMP-652 before 2014, you CANNOT take COMP-551.
  - However taking COMP-652 during/after Winter 2014 is ok (course was re-designed to avoid overlap).

About the course

Evaluation:

- One midterm (35%)
- Five data analysis projects + peer reviews (65%)

Coursework policy:

All course work should be submitted online (details to be given in class), by 11:59pm, on the assigned due date. Late work will be subject to a 30% penalty, and can be submitted up to 1 week after the deadline.

No make-up midterm will be given.

About the course

- Five projects:
  1. Mini project linear regression (TBC) 10%
  2. Mini project linear classification (TBC) 10%
  3. Mini project SVM (TBC) 10%
  4. Case study neural networks. 15%
  5. Final project. (reproducibility study) 20%
- Format:
  - Projects will be submitted as written report + working code/data.
  - Final project will involve a short oral presentation.
- Mini projects: individual
- Case study + final project: teams of 3. Work with different people for each project.

About the course

- I will not be using the classroom recording system.
- **My advice**: Do not to register for two courses in same time block.

  Plan on attending every class.

- Slides and projects will be available on the class website.
- MyCourses is available for discussions and finding project teams.

Course material

No mandatory textbook, but a few good textbooks are recommended on the syllabus (some freely available online).

- Shalev-Schwartz & Ben-David. *Understanding Machine Learning*. Cambridge University Press. 2014.
    - *More theoretical.*
- Hastie, Tibshirani & Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd Edition. Springer. 2009.
    - *More mathematical.*
- Bishop. *Pattern Recognition and Machine Learning*. Springer. 2007.
    - *More practical, more accessible.*
- Goodfellow, Bengio &Courville. *Deep Learning*. MIT Press. 2016.
    - *For neural networks and deep learning modules.*

Software tools

- **Many** software packages are available, including broad ML libraries in Java, C++, Python and others.
- Most projects can be completed in a language of your choice, but support, tips, and tutorials will be based on Python
- Many advanced packages for specialized algorithms.
- Strong push in the community to make software freely available.

Expectations

The courses is intended for **hard-working, technically skilled, highly motivated students**.

- Take notes during class. Do the readings. Review the slides.
- Participate in discussions and sessions. Ask questions.
- Respect the coursework policy.

Participants are expected to show **initiative, creativity, scientific rigour, critical thinking, and good communication skills.**

- Be prepared to work hard on the projects. Work well in a team.
- Provide constructive feedback in peer-reviews.

Read this carefully

- **Some of the course work will be individual, other components can be completed in groups.** It is the responsibility of each student to understand the policy for each work, and ask questions of the instructor if this is not clear.
- It is the responsibility of each student to **carefully acknowledge all sources** (papers, code, books, websites, individual communications) using appropriate referencing style when submitting work.
- **We will use automated systems to detect possible cases of text or software plagiarism.** Cases that warrant further investigation will be referred to the university disciplinary officers. Students who have concerns about how to properly use and acknowledge third-party software should consult a McGill librarian or the TAs.

Questions?

What is machine learning?

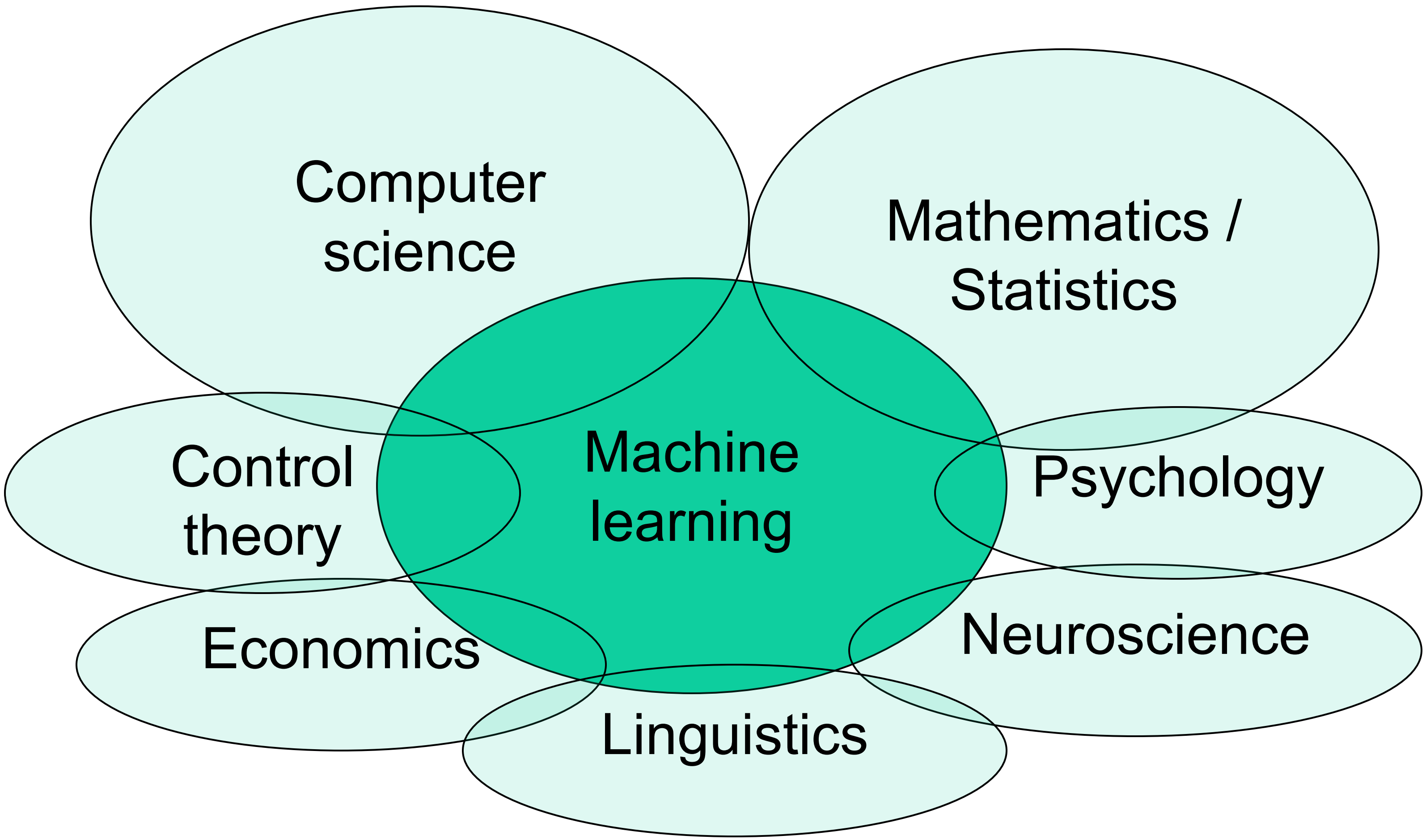
- A definition (by Tom Mitchell):

  “How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?”

- More technically:

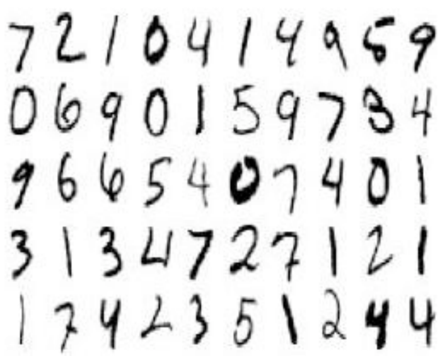
  "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"

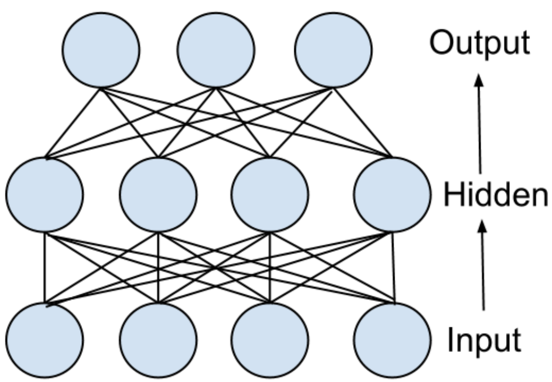
About machine learning

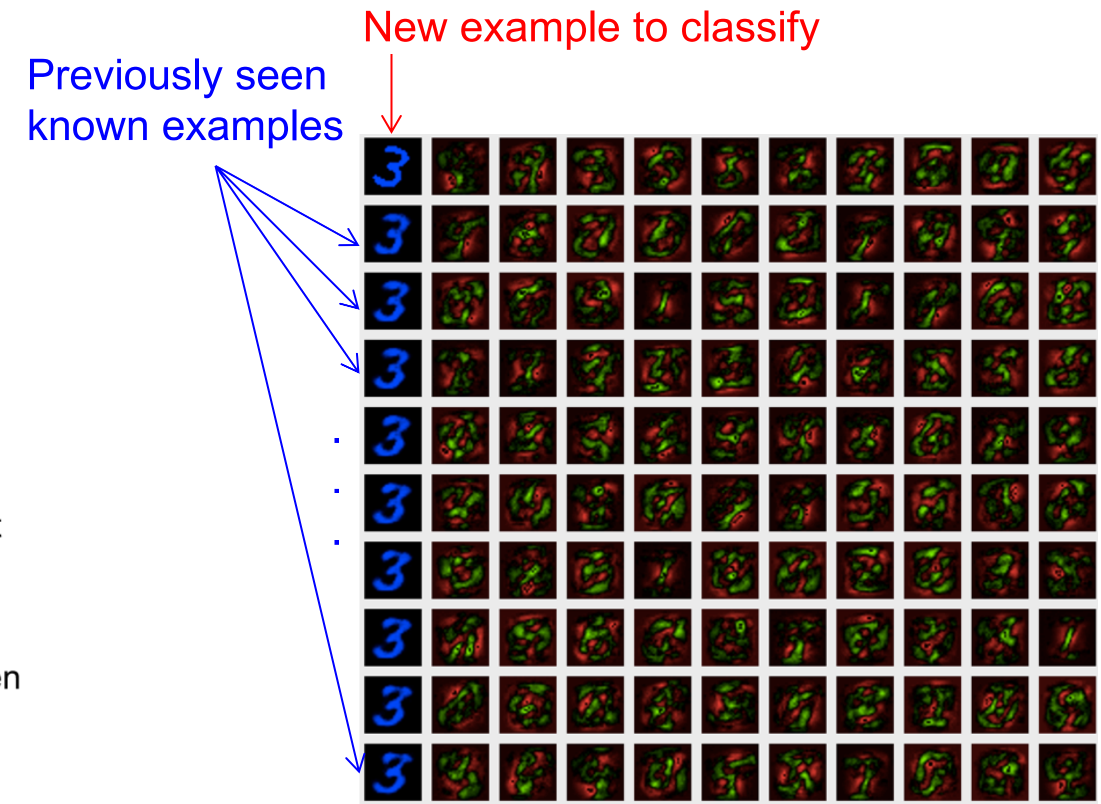


Case study #1: Optimal character recognition

- Handwritten digit recognition: >99% accuracy (on a large dataset).





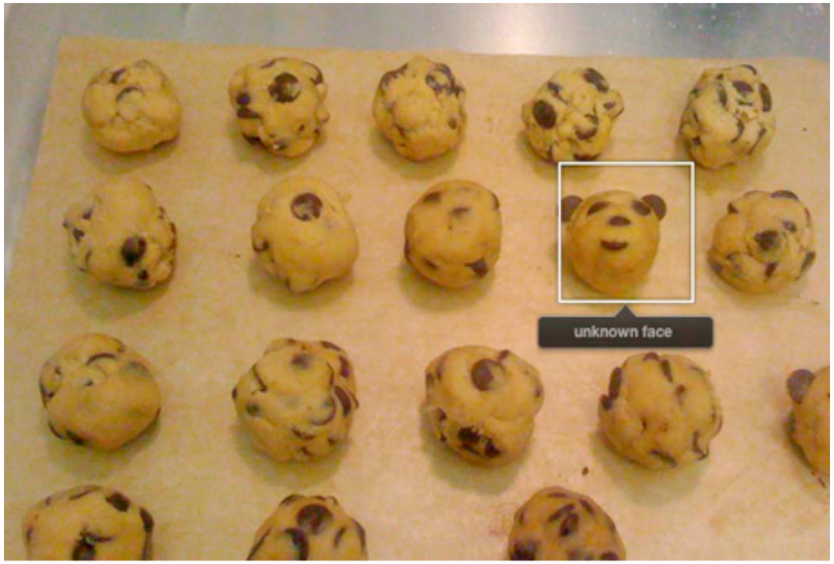


Boxes represent the weights into a hidden node in a neural network learner.

Case study #2: Computer Vision

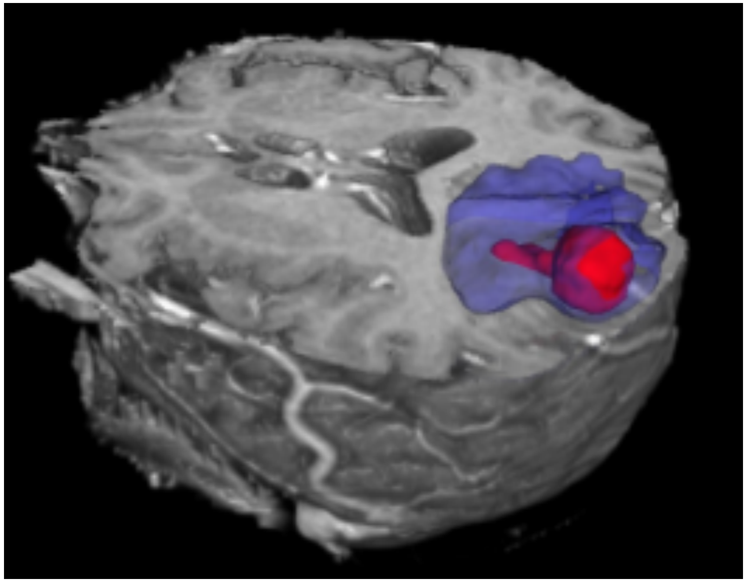
- Face recognition.

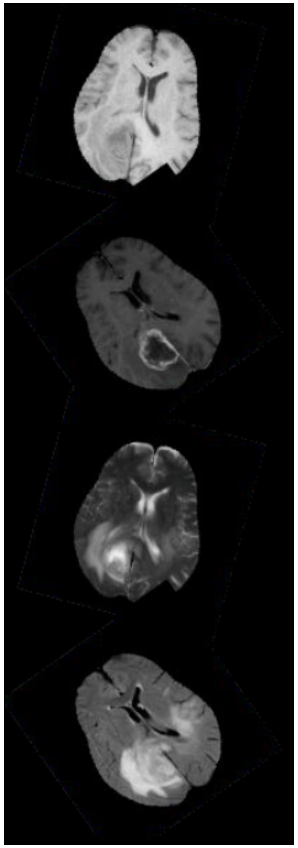
Not always perfect!



Case study #2: Computer Vision

- Voxel-level tumour segmentation





Case study #3: Text analysis

- Learning a language model:

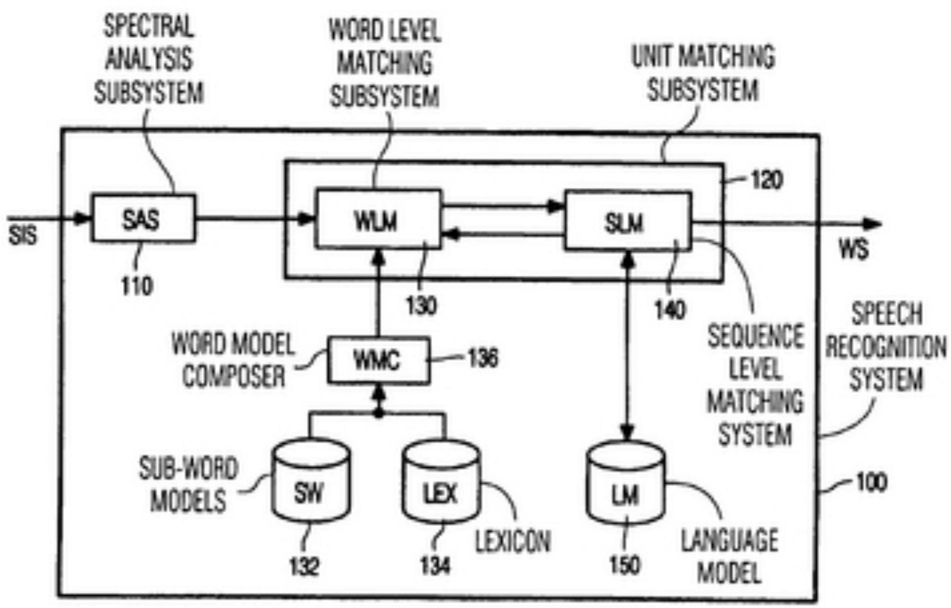
Text corpus → Statistical language model

$$P(W_n|W_{n-1}) = \frac{P(W_{n-1}, W_n)}{P(W_{n-1})}$$

Case study #3: Text analysis

- Learning a language model:

Text corpus → Statistical language model

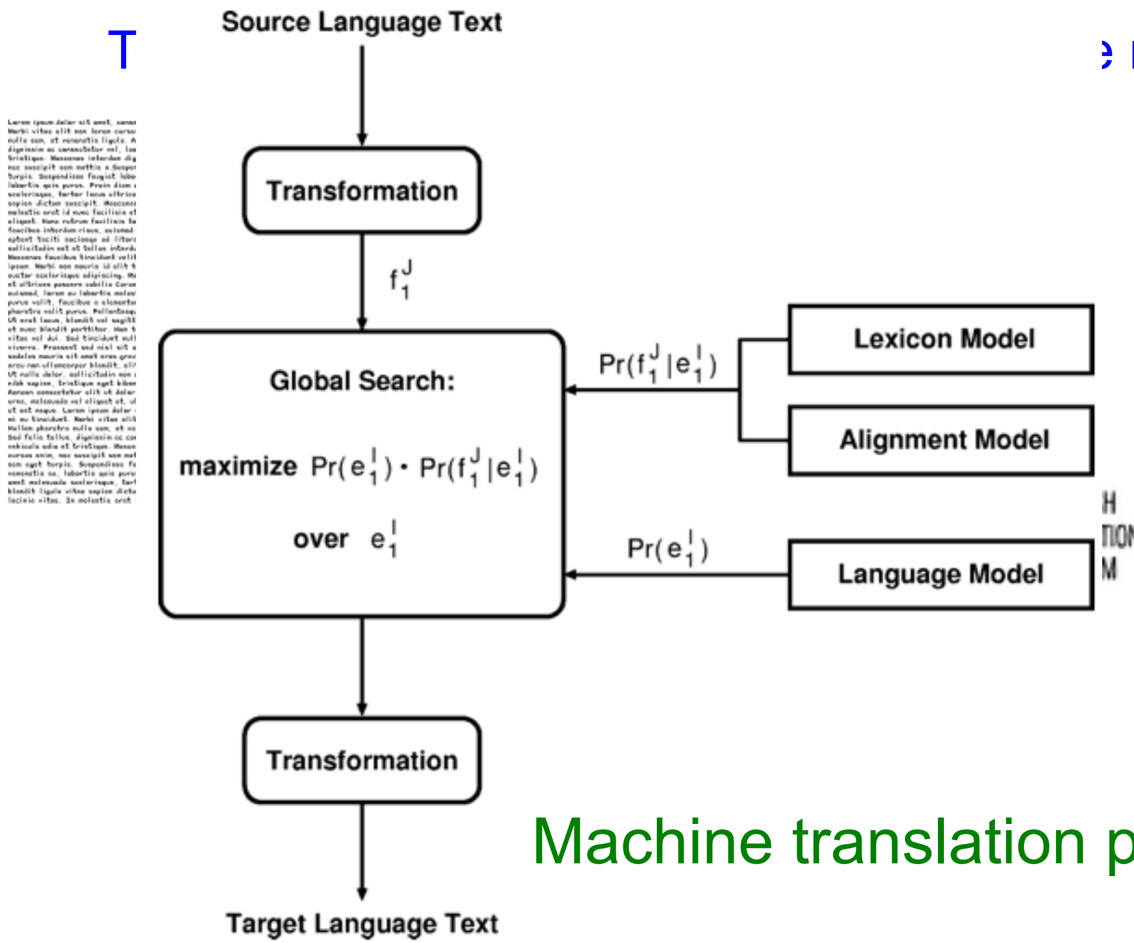


$\frac{{}_{-1}, W_n)}{V_{n-1})}$

Speech recognition pipeline

Case study #3: Text analysis

- Learning a language model:



Machine translation pipeline

Case study #3: Text analysis

- From vision input to text output:

"Two pizzas sitting on top of a stove top oven"

"A group of young people playing a game of frisbee"

Case study #3: Text analysis

- Still some work to do!

"A refrigerator filled with lots of food and drinks"

"A yellow school bus parked in a parking lot"

Case study #4: The Netflix Prize

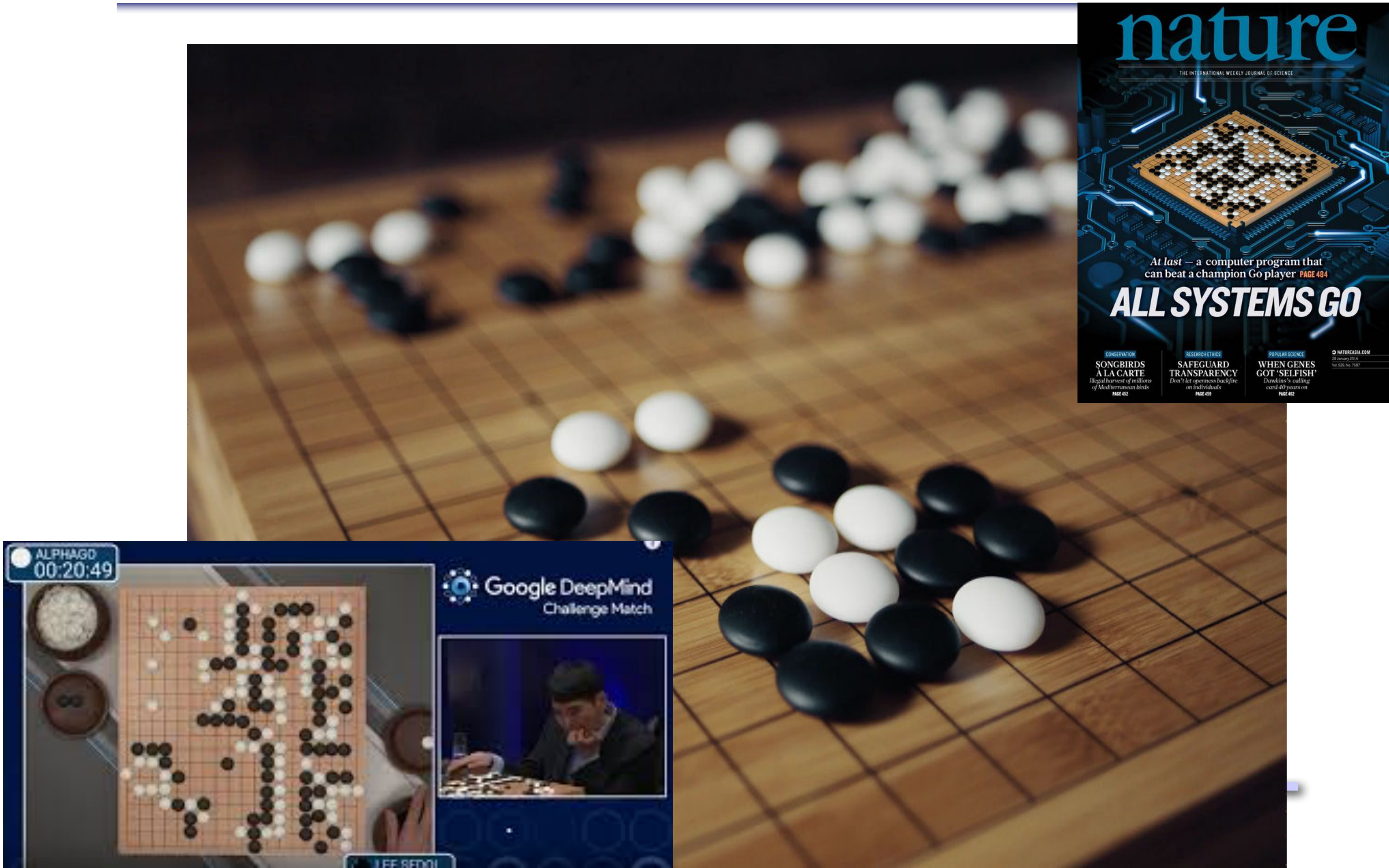
Task: Improve Netflix's recommendation system by 10%.

Training data: 10^8 movie ratings, to build the ML algorithm.

Test set: 1.5×10^6 ratings to evaluate final performance.

Quiz set: 1.5×10^6 ratings to calculate leaderboard scores.

Case study #5: Playing games



Back to the definition

*“computer systems that automatically improve with **experience**”*

*"A computer program that … improves it performance at tasks in T, as measured by performance measure P, improves with **experience** E"*

- What is “experience”?

Back to the definition

*“computer systems that automatically improve with **experience**”*

*"A computer program that … improves it performance at tasks in T, as measured by performance measure P, improves with **experience** E"*

- What is “experience”?

lines of text

labeled images

games played

Representing data

- Machine learning algorithms (typically) only see numbers
- Typically, we create **one vector** representing **each experience**
- Either with **raw** data values (pixel, characters) or **preprocessed** data (words, colors, shapes)
- Vectors organized in a **table**

Machine learning problems - terminology

- Data is often presented in tables
- Columns are called **input variables** or **features** or **attributes**.
- The columns we are trying to predict (outcome and time) are called **output variables** or **targets**.
- A row in the table is called a **training example** or **instance**.
- The whole table is called a **data set**.

tumor size	texture	perimeter	...	outcome	time
18.02	27.6	117.5		N	31
17.99	10.38	122.8		N	61
20.29	14.34	135.1		R	27
...					

Back to the definition

"computer systems that automatically improve with experience"

*"A computer program that … improves it performance at **tasks** in T, as measured by performance measure P, improves with experience E"*

- What are the **tasks**?
- We've seen some examples
- Can we categorize them?

Main types of machine learning problems

- Supervised learning
  - Classification
  - Regression
- Unsupervised learning
- Reinforcement learning

Supervised learning

Training data: Examples of input coupled with desired output

Goal: Predict the output for new inputs

- This is the most typical set-up for prediction tasks
- Sometimes, it can be hard to obtain training data matched with the correct output
- Two main types of supervised learning:
  - Classification
  - Regression

Supervised learning - Classification

Goal: Learning a function for a categorical output.

E.g.: Spam filtering. The output ("Spam?") is binary.

	Sender in address book?	Header keyword	Word 1	Word 2	…	Spam?
x1	Yes	Schedule	Hi	Profesor	…	No
x2	Yes	meeting	Joelle	I	…	No
x3	No	urgent	Unsecured	Business	…	Yes
x4	No	offer	Hello	I	…	Yes
x5	No	cash	We'll	Help	…	Yes
x6	No	comp-551	Dear	Professor	…	No
…						

Supervised learning - Regression

Goal: Learning a function for a continuous output.

E.g.: Predict sale prices of houses

	Building class	Zoning classification	Frontage	Lot size	…	Sale price
x1	60	RL	65	8450	⋯	208500
x2	20	RL	80	9600	⋯	181500
x3	60	RL	68	11250	⋯	223500
x4	70	RL	60	9550	⋯	140000
x5	60	RL	84	14260	⋯	250000
x6	60	RL	65	8450	⋯	143000
…						

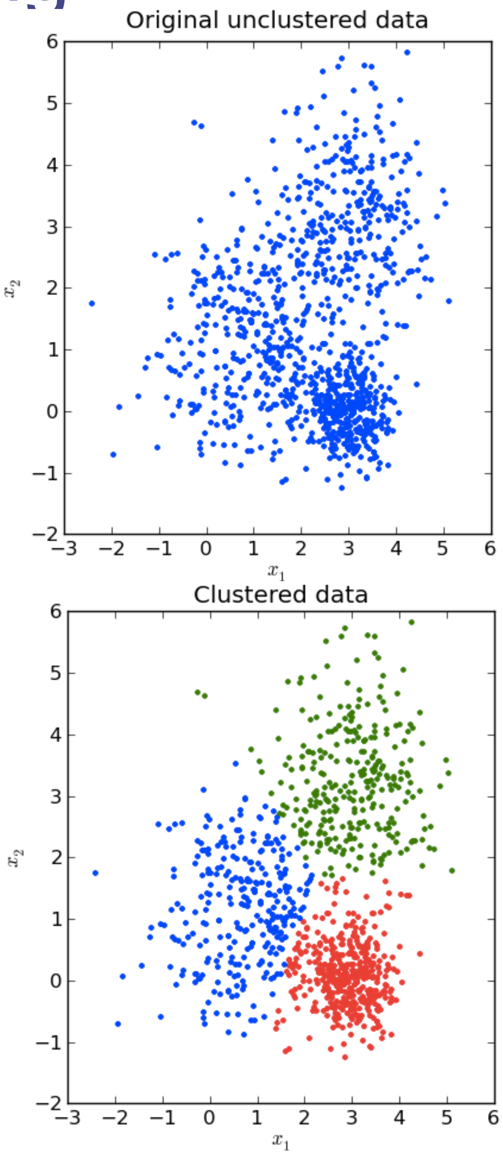
Unsupervised learning

Training data: Input data only

Goal: Find some useful structure in the data set

Exampels:

- Organizing data into groups (clustering)
- Infer how typical any input is (density estimation)
- Find what differences between instances are most relevant (dimension reduction)



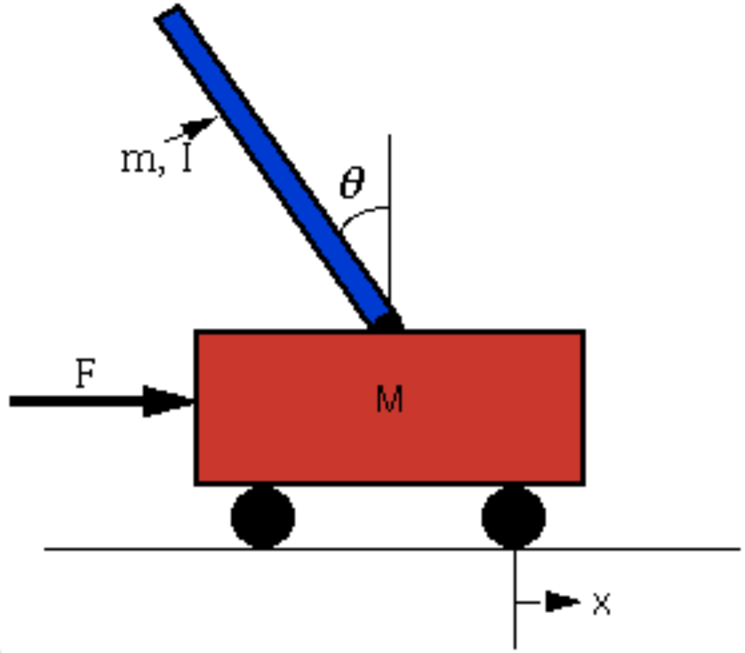
Reinforcement learning

Training data: Input (system state), tried output (action), cost or reward associated with tried output

Goal: Learning a sequence of actions that optimizes costs/rewards.

E.g.: Balancing an inverted pendulum.

Note: most actions in the dataset usually sub-optimal



Types of machine learning problems

- Supervised learning
  - Classification
  - Regression
- Unsupervised learning
- Reinforcement learning

This course (Comp551 - Applied ML)	Machine Learning (Comp652)
Reinforcement Learning (Comp 782)	

ML today

- Currently the method of choice for many applications:
  - Speech recognition
  - Computer vision
  - Robot control
  - Fraud detection

  and growing…
- Why so many applications?

ML today

- Currently the method of choice for many applications:
  - Speech recognition
  - Computer vision
  - Robot control
  - Fraud detection

  and growing…

- Why so many applications?
  - Increase in number of sensors/devices ➔ We have loads of data!
  - Increase in computer speed and memory ➔ We can process the data!
  - Better ML algorithms and software for easy deployment.
  - Increasing demand for customized solutions (e.g personalized news).

Harvard Business Review

SEARCH

THE MAGAZINE | BLOGS | AUDIO & VIDEO | BOOKS | CASES | WEBINARS

Guest | limited access

Register today and save 20%* off your first order! Details

THE MAGAZINE

October 2012

ARTICLE PREVIEW To read the full article: **Sign in** or **Register** for free. HBR Subscribers **activate your free archive access »**

Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and D.J. Patil

Comments (87)

RELATED

Executive Summary

ALSO AVAILABLE

- Buy PDF

Research challenge: Big data

- Old-style $O(n^2)$ algorithms simply won't work.
- Fitting the data on a single machine may not be feasible. Work from a stream of examples (process every example only once.)
- Must distribute computations across multiple machines.

E.g. Predicting which ad is interesting (from John Langford)

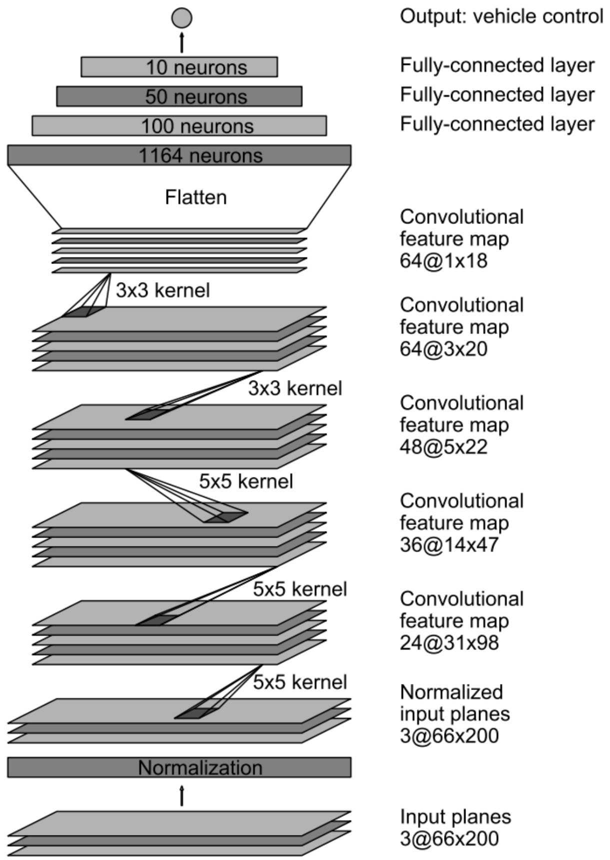
- 2.1TB sparse features
- 17B examples
- 16M parameters
- 1K computation nodes

Research challenge: End-to-end learning

- From raw features => high-order decision.

E.g.

- Single characters => Text classification
  - https://arxiv.org/abs/1509.01626
- Pixels => Steering angle for autonomous driving
  - https://arxiv.org/pdf/1604.07316v1.pdf



Lots of other (inter-disciplinary) challenges

- Many open questions about algorithmic methods and theoretical characterization.
  - Inferring the right representation / model.
  - Exploration vs Exploitation

- Weakness in evaluation methods.

- Privacy concerns in obtaining and releasing data.

- Many exciting under-explored applications!

DOI:10.1145/2347736.2347755

Tapping into the "folk knowledge" needed to advance machine learning applications.

BY PEDRO DOMINGOS

A Few Useful Things to Know About Machine Learning

MACHINE LEARNING SYSTEMS automatically learn programs from data. This is often a very attractive alternative to manually constructing them, and in the last decade the use of machine learning has spread rapidly throughout computer science and beyond.

is needed to successfully develop machine learning applications is not readily available in them. As a result, many machine learning projects take much longer than necessary or wind up producing less-than-ideal results. Yet much of this folk knowledge is fairly easy to communicate. This is the purpose of this article.

» key insights

- Machine learning algorithms can figure out how to perform important tasks by generalizing from examples. This is

Final comments

- Come to class! Come prepared!

- For next class:
  - (Must) Read this paper:
    - *http://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf*
  - (If necessary) Review basic algebra, probability, statistics.
    - Ch.1-2 of Bishop.
    - *http://www.cs.mcgill.ca/~dprecup/courses/ML/Materials/prob-review.pdf*
    - *http://www.cs.mcgill.ca/~dprecup/courses/ML/Materials/linalg-review.pdf*
    - Many online resources.
  - (Optional) Read Chap.1-2 of Bishop, Ch. 1 of Hastie et al. or Ch.2 of Shalev-Schwartz et al.