

Machine Learning (COMP-652 and ECSE-608)

Lecturers: Audrey Durand and Riashat Islam

Email: audrey.durand@mcgill.ca - riashat.islam@mail.mcgill.ca

Teaching assistants: Koustuv Sinha, Pierre Thodoroff, Nishanth Anand

Emails: {[koustuv.sinha](mailto:koustuv.sinha@mail.mcgill.ca),[pierre.thodoroff](mailto:pierre.thodoroff@mail.mcgill.ca),[nishanth.anand](mailto:nishanth.anand@mail.mcgill.ca)}@mail.mcgill.ca

Class web page: <https://rllabmcgill.github.io/COMP-652>

Course Content

- Introduction, linear models, regularization and Bayesian interpretation (3 lectures)
- Regression, kernels and Gaussian processes (3 lectures)
- Neural networks and deep learning (3 lectures)
- Latent variables and unsupervised learning (1 lecture)
- Clustering and dimensionality reduction (1 lecture)
- Reinforcement learning and Markov processes (5 lectures)
- Active learning and Bayesian optimization (1 lecture)
- Probabilistic graphical models (1 lecture)
- Optimization (1 lecture)
- Inference (2 lectures)
- Generative models (1 lecture)

Today's Lecture Outline

- Administrative issues
- What is machine learning?
- Types of machine learning
- Linear hypotheses
- Error functions

Administrative issues

- Class materials:
 - No required textbook, but several textbooks available
 - Required or recommended readings (from books or research papers) posted on the class web page
 - Class notes: posted on the web page
- Prerequisites:
 - Knowledge of a programming language
 - Knowledge of probability/statistics, calculus and linear algebra; general facility with math
 - Some AI background is recommended *but not required*¹

¹show of hands: who took COMP-424? COMP-551?

Office hours

- Right after class: 4pm-5pm Mondays and Wednesdays (one of the lecturers).
- 10am-11am Tuesdays (Pierre), 11am-12pm Wednesdays (Nishanth), Thursdays 3pm-4pm (Koustuv).
- Meeting with any of the lecturers or TAs at other times by appointments only.

Evaluation

- Three homework assignments (30%)
- Project (30%)
- Final examination (40%)
- Participation to class discussions (up to 1% extra credit)

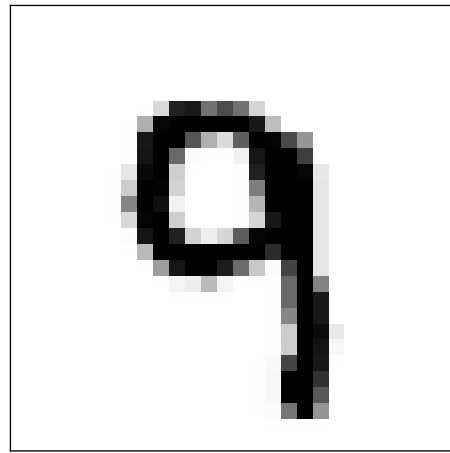
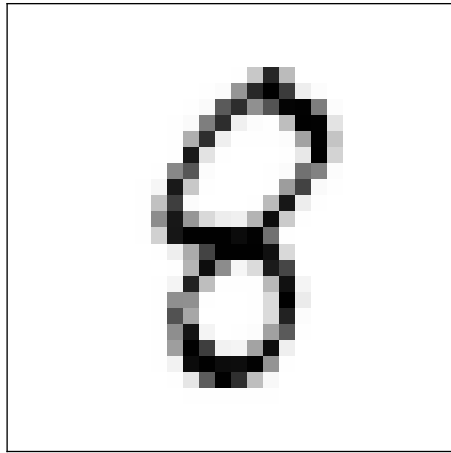
What is machine learning?

- A. Samuel, who coined the term *machine learning*:

“Field of study that gives computers the ability to learn without being explicitly programmed.” [*Some Studies in Machine Learning Using the Game of Checkers*, 1959]

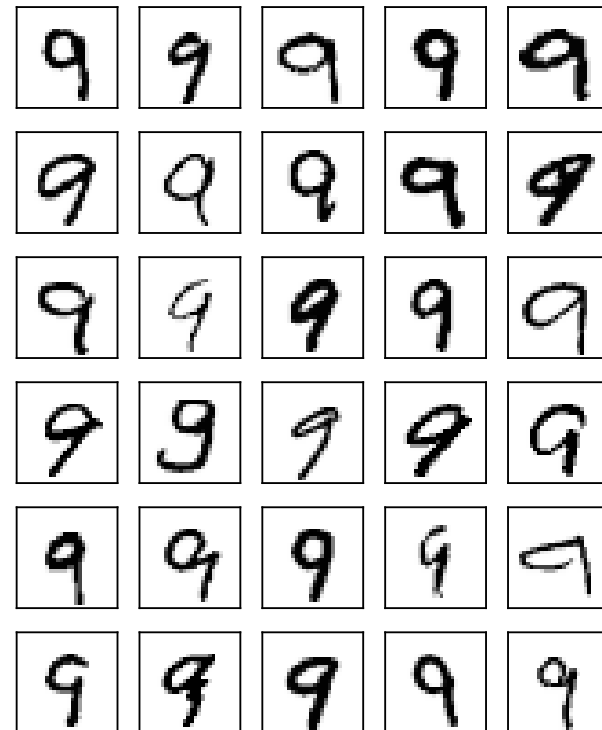
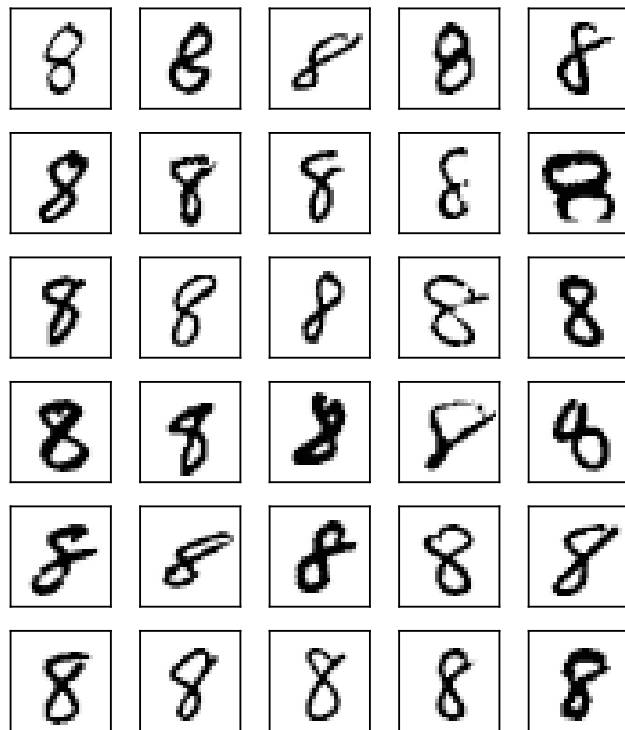
What is learning?

- Let's say you want to write a program distinguishing images of the digits 8 and 9...



What is learning?

- The Machine Learning way:



What is learning?

- A. Samuel, who coined the term *machine learning*:

“Field of study that gives computers the ability to learn without being explicitly programmed.” [*Some Studies in Machine Learning Using the Game of Checkers*, 1959]
- H. Simon: Any process by which a system improves its performance

“Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the task or tasks drawn from the same population more efficiently and more effectively the next time.” [*Machine Learning I*, 1983]
- M. Minsky:

“Learning is making *useful* changes in our minds.” [*The Society of Mind*, 1985]
- R. Michalski:

“Learning is constructing or modifying representations of what is being experienced.” [Understanding the Nature of Learning, 1986]

Why study machine learning?

Engineering reasons:

- Easier to build a learning system than to hand-code a working program!
E.g.:
 - Robot that learns a map of the environment by exploring
 - Programs that learn to play games by playing against themselves
- Improving on existing programs, e.g.
 - Instruction scheduling and register allocation in compilers
 - Combinatorial optimization problems
- Solving tasks that require a system to be adaptive, e.g.
 - Speech and handwriting recognition
 - “Intelligent” user interfaces

Why study machine learning?

Scientific reasons:

- Discover knowledge and patterns in highly dimensional, complex data
 - Sky surveys
 - High-energy physics data
 - Sequence analysis in bioinformatics
 - Social network analysis
 - Ecosystem analysis
- Understanding animal and human learning
 - How do we learn language?
 - How do we recognize faces?
- Creating real AI!

“If an expert system—brilliantly designed, engineered and implemented—cannot learn not to repeat its mistakes, it is not as intelligent as a worm or a sea anemone or a kitten.” (Oliver Selfridge).

Very brief history

- Studied ever since computers were invented (e.g. Samuel's checkers player, 1959)
- Very active in 1960s (neural networks)
- Died down in the 1970s
- Revival in early 1980s (decision trees, backpropagation, temporal-difference learning) - coined as "machine learning"
- Exploded starting in the 1990s
- Now: very active research field, several yearly conferences (e.g., ICML, NIPS), major journals (e.g., Machine Learning, Journal of Machine Learning Research), rapidly growing number of researchers
- The time is right to study in the field!
 - Lots of recent progress in algorithms and theory
 - Flood of data to be analyzed
 - Computational power is available
 - Growing demand for industrial applications

What are good machine learning tasks?

- There is no human expert
E.g., DNA analysis
- Humans can perform the task but cannot explain how
E.g., character recognition
- Desired function changes frequently
E.g., predicting stock prices based on recent trading data
- Each user needs a customized function
E.g., news filtering

Important application areas

- Bioinformatics: sequence alignment, analyzing microarray data, information integration, ...
- Computer vision: object recognition, tracking, segmentation, active vision, ...
- Robotics: state estimation, map building, decision making
- Graphics: building realistic simulations
- Speech: recognition, speaker identification
- Financial analysis: option pricing, portfolio allocation
- E-commerce: automated trading agents, data mining, spam, ...
- Medicine: diagnosis, treatment, drug design,...
- Computer games: building adaptive opponents
- Multimedia: retrieval across diverse databases

Kinds of learning

Based on the information available:

- Supervised learning
- Reinforcement learning
- Unsupervised learning

Based on the role of the learner

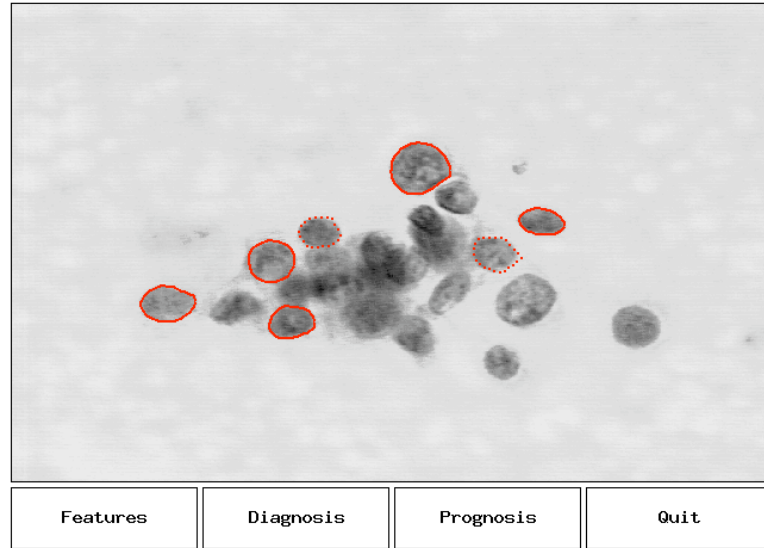
- Passive learning
- Active learning

Passive and active learning

- Traditionally, learning algorithms have been *passive learners*, which take a given batch of data and process it to produce a hypothesis or model
Data \rightarrow Learner \rightarrow Model
- *Active learners* are instead allowed to query the environment
 - Ask questions
 - Perform experiments
- Open issues: how to query the environment optimally? how to account for the cost of queries?

Example: A data set

Cell Nuclei of Fine Needle Aspirate



- Cell samples were taken from tumors in breast cancer patients before surgery, and imaged
- Tumors were excised
- Patients were followed to determine whether or not the cancer recurred, and how long until recurrence or disease free

Data (continued)

- Thirty real-valued variables per tumor.
- Two variables that can be predicted:
 - Outcome (R=recurrence, N=non-recurrence)
 - Time (until recurrence, for R, time healthy, for N).

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
. . .					

Terminology

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
...					

- Columns are called *input variables* or *features* or *attributes*
- The outcome and time (which we are trying to predict) are called *output variables* or *targets*
- A row in the table is called *training example* or *instance*
- The whole table is called *(training) data set*.
- The problem of predicting the recurrence is called *(binary) classification*
- The problem of predicting the time is called *regression*

More formally

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
. . .					

- A training example i has the form: $\langle x_{i,1}, \dots, x_{i,n}, y_i \rangle$ where n is the number of attributes (30 in our case).
- We will use the notation \mathbf{x}_i to denote the column vector with elements $x_{i,1}, \dots, x_{i,n}$.
- The training set D consists of m training examples
- We denote the $m \times n$ matrix of attributes by \mathbf{X} and the size- m column vector of outputs from the data set by \mathbf{y} .

Supervised learning problem

- Let \mathcal{X} denote the space of input values
- Let \mathcal{Y} denote the space of output values
- Given a data set $D \subset \mathcal{X} \times \mathcal{Y}$, find a function:

$$h : \mathcal{X} \rightarrow \mathcal{Y}$$

such that $h(\mathbf{x})$ is a “*good predictor*” for the value of y .

- h is called a *hypothesis*
- Problems are categorized by the type of output domain
 - If $\mathcal{Y} = \mathbb{R}$, this problem is called *regression*
 - If \mathcal{Y} is a categorical variable (i.e., part of a finite discrete set), the problem is called *classification*

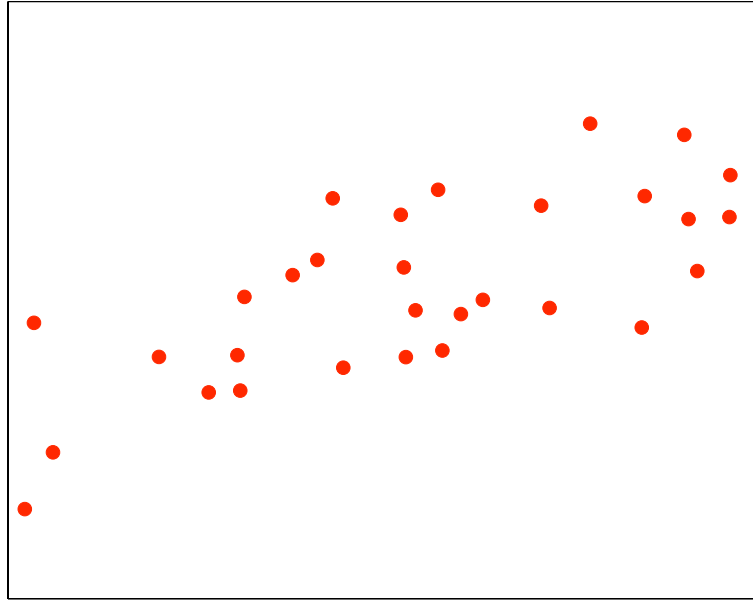
Steps to solving a supervised learning problem

1. Decide what the input-output pairs are.
2. Decide how to encode inputs and outputs.
This defines the input space \mathcal{X} , and the output space \mathcal{Y} .
(We will discuss this in detail later)
3. Choose a class of hypotheses/representations \mathcal{H} .
4. ...

Hypothesis class

- All possible hypotheses that we are considering in the search
- Assumptions made on the problem
- You can only be as good as the best hypothesis in the class
- So... Is a larger class always better?
- How can we *control* the size of the class?

Example: What hypothesis class should we pick?



x	y
0.86	2.49
0.09	0.83
-0.85	-0.25
0.87	3.10
-0.44	0.87
-0.43	0.02
-1.10	-0.12
0.40	1.81
-0.96	-0.83
0.17	0.43

Linear hypothesis

- Suppose y was a linear function of \mathbf{x} :

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1x_1(+ \cdots)$$

- w_i are called *parameters* or *weights*
- To simplify notation, we can add an attribute $x_0 = 1$ to the other n attributes (also called *bias term* or *intercept term*):

$$h_{\mathbf{w}}(\mathbf{x}) = \sum_{i=0}^n w_i x_i = \mathbf{w}^T \mathbf{x}$$

where \mathbf{w} and \mathbf{x} are vectors of size $n + 1$.

How should we pick \mathbf{w} ?

Error minimization!

- Intuitively, \mathbf{w} should make the predictions of $h_{\mathbf{w}}$ close to the true values y on the data we have
- Hence, we will define an *error function* or *cost function* to measure how much our prediction differs from the "true" answer
- We will pick \mathbf{w} such that the error function is minimized

How should we choose the error function?

Least mean squares (LMS)

- Main idea: try to make $h_{\mathbf{w}}(\mathbf{x})$ close to y on the examples in the training set
- We define a *sum-of-squares* error function

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^m (h_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2$$

(the 1/2 is just for convenience)

- We will choose \mathbf{w} such as to minimize $J(\mathbf{w})$

Steps to solving a supervised learning problem

1. Decide what the input-output pairs are.
2. Decide how to encode inputs and outputs.
This defines the input space \mathcal{X} , and the output space \mathcal{Y} .
3. Choose a class of hypotheses/representations \mathcal{H} .
4. Choose an error function (cost function) to define the best hypothesis
5. Choose an algorithm for searching efficiently through the space of hypotheses.

Next time: Find the hypothesis that minimizes the error function!