Introduction

CMPUT 267: Basics of Machine Learning

Chapter 1

What is machine learning?

- Mitchell: "The field of machine learning is concerned with the question of how to construct computer programs that **automatically** improve with **experience**."
- Russell & Norvig: "... the subfield of AI concerned with programs that learn from **experience**."
- Murphy: "The goal of machine learning is to develop methods that can **automatically** detect patterns in **data**, and then to use the uncovered patterns to predict future data of other outcomes of interest."

What is this course about?

We want to be able to have good rules (or functions) for predicting outcomes
e.g., predict the temperature tomorrow, based on weather over the last five days

You could construct these rules by hand, or learn them from **data:**

- But the data are often **incomplete**:
  - Partial observability: Incomplete knowledge of environment
  - Incomplete knowledge of other agents' actions
- **Machine learning algorithms** are one way to learn from incomplete data

Course goal:
Understand machine learning algorithms by **deriving them** from the beginning.

- with a focus on prediction on new data

Example: Predicting house prices

- **Goal**: we want to predict house prices, given only the age of the house

  $f(age) =$ price of the house

- **Dataset**: house sales this year, with attributes **age** and target value **price**

  $\{(age_1, price_1), (age_2, price_2), \ldots, (age_9, price_9)\}$

- **Idea**: A function that accurately outputs price from age for these specific pairs might also provide good predictions for new houses

Formalizing the problem

Definitions:

Let x be **age** and y be **price**

Let $D = \{(x_1, y_1), \ldots, (x_9, y_9)\}$ be our dataset

Objective:

We want to make the **difference** between $f(x_i)$ and y_i **small**

$$\min_{f \text{ in function space}} \sum_{i=1}^{9} (f(x_i) - y_i)^2$$

Questions:

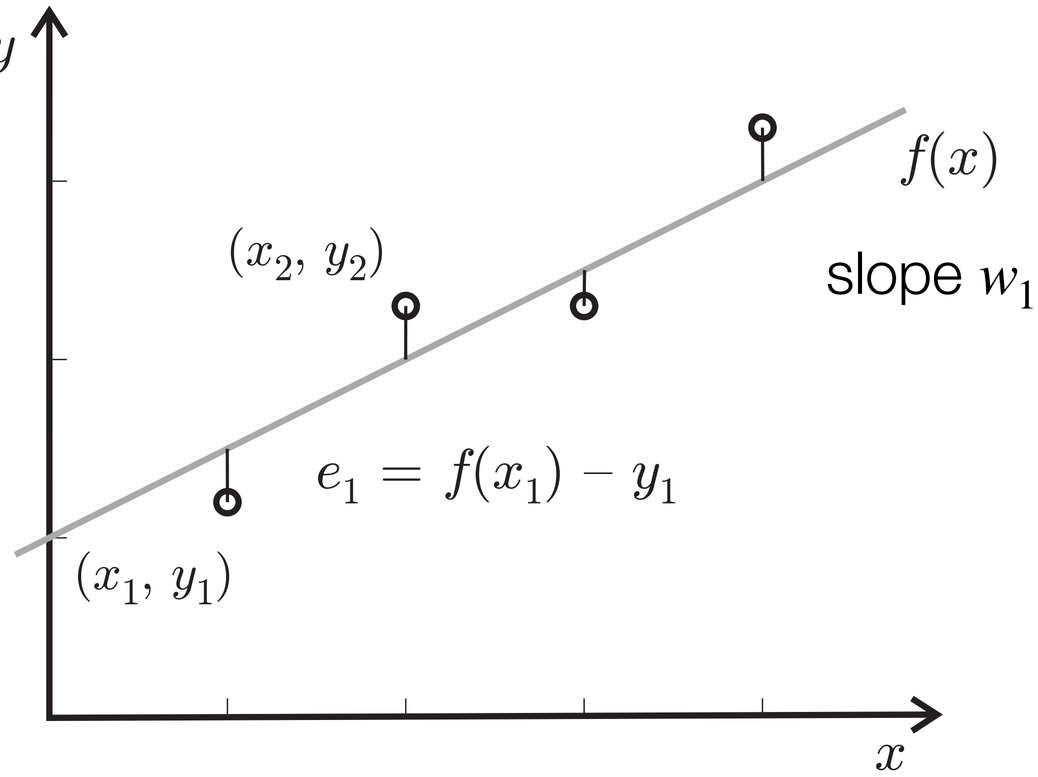
1. Why are we **squaring** the difference?
2. Why are we **summing** up the errors?
3. What could we consider for the **function space**?

Linear function space

Definition:
A function f is a **linear function** of x if it can be written as $f(x) = w_0 + w_1 x$

$$\min_{f \text{ in linear functions}} \sum_{i=1}^{9} (f(x_i) - y_i)^2$$

where $e_i = f(x_i) - y_i = w_0 + w_1 x_i - y_i$

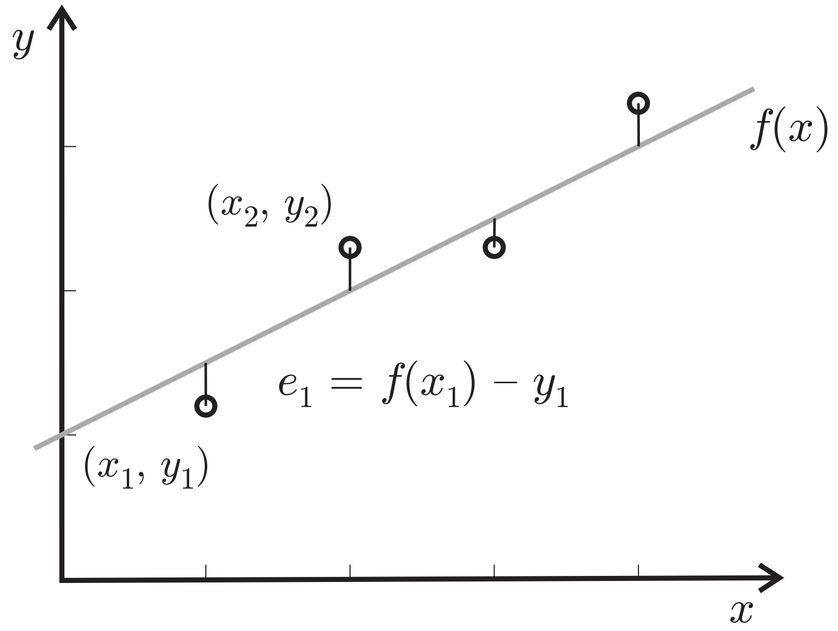


Solving for the optimal function

Objective then becomes:

$$\min_{f \text{ in function space}} \sum_{i=1}^{9} (f(x_i) - y_i)^2$$

$$= \min_{w_0, w_1} \sum_{i=1}^{9} (\underbrace{w_0 + w_1 x_i}_{f(x_i)} - y_i)^2$$



Questions:

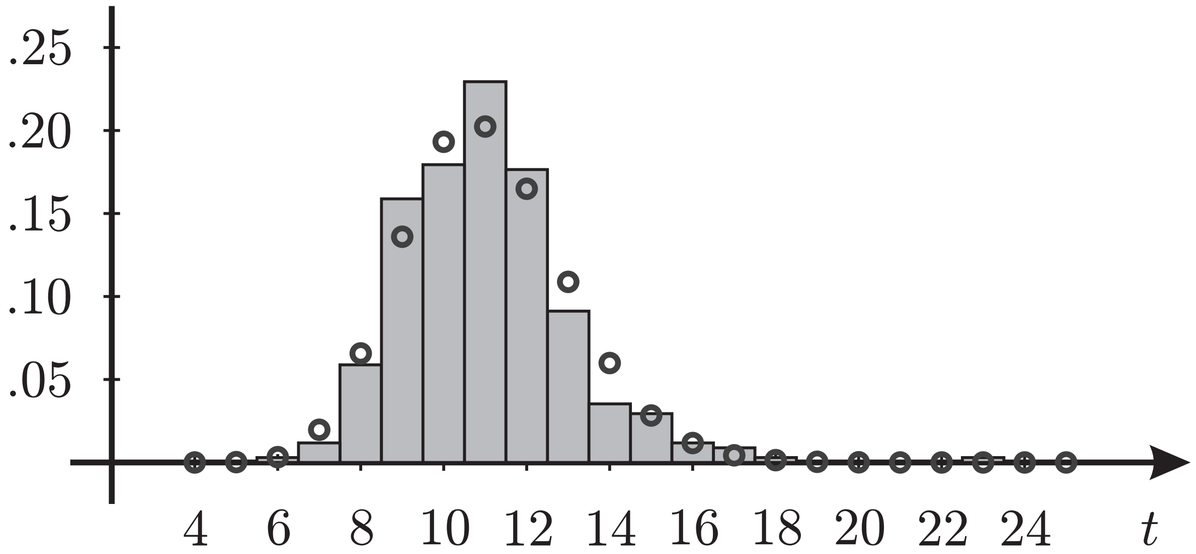
1. Would you use this to predict the value of a house? Why/why not?
2. Will this predict well? How do we know?
3. What is missing to make these assessments?

Probabilities!

- **Question:** Is it likely that there is a **deterministic** function from **age** to **price**?
  - Many houses will have the same **age** but different **price**...
- We can instead use a probabilistic approach:
  - Learn a function that gives a **distribution** over **targets** (**price**) given **attributes** of the item (**age**)

Probabilities let us specify our uncertainty in predictions

- Imagine we have a distribution over the prices for a given age of 40 years
  - The x-axis is the prices (in 10s of thousands)
  - The y-axis is the probability of that price, for the age of 40 years
- We might reasonably **predict** our house price is something like 100000 or 120000, based on this distribution
- But we also know that the spread is quite large, and that there is **uncertainty** in our prediction



Probabilities let us specify our uncertainty in predictions

- We might reasonably **predict** our house price is something like 100000 or 120000, based on this distribution
- But we also know that the spread is quite large, and that there is **uncertainty** in our prediction
- A further nuance: this distribution itself is an **estimate** (from data) and so we actually have **uncertainty** about it too!
  - We will also use **probabilities** to quantify this **uncertainty**

Probabilities!

- **Question:** Does this mean that we think the world is **stochastic** rather than deterministic?
  - Stochasticity can come from **partial observability**
  - Maybe the outcome *really is* deterministic if we knew **age**, and **size**, and **number of rooms**, and **distance to airport**, and **whether the queen lives there**, and …
  - But we will always have limitations on what variables we can measure (only partially observe the true state of the system/item)

Course topics

1. Probability Background (ch.2)
2. Estimation with Sample Averages (ch.3)
   - Concentration inequalities: how confident should we be in our estimates?
   - Sample complexity and convergence rate
3. Optimization (ch.4)
4. Parameter Estimation (ch.5)
   - Maximum likelihood and MAP
   - Beyond point estimates: Bayesian estimation
5. Stochastic Gradient Descent and Handling Big Datasets (ch. 6)

Course topics #2

5. Prediction (ch.7)
   - Formalizing the prediction objective
6. Linear & Polynomial Regression (ch.8)
7. Generalization Error and Evaluating Models (ch.9)
8. Regularization and Constraining the Function Space (ch.10)
9. Logistic Regression and Linear Classifiers (ch.11)
10. Bayesian Linear Regression (ch.12)

Course Structure: Part 1 (Chapters 1-6) focused on providing need background
Part 2 (Chapters 1-7) more directly about ML concepts

Course essentials

- **Course information:** https://marthawhite.github.io/mlbasics/
  - Schedule and readings
- **Access-controlled course information:** eClass
  - Getting Started and FAQ (please visit this today!)
  - Video recordings, links to lecture meetings and assignment submission
- **My Office hours**: Tuesday, noon - 1:00 pm (my office ATH 3-05 and on Zoom)

Lectures

- **Lectures will be in-person in a classroom AND on Zoom**
  - I highly encourage you to come to class, you will get more out of the course
  - But I also know sometimes you cannot (sick, etc.)
- In class, I will project my screen and share my screen on Zoom
- I will monitor the Zoom chat and ask any questions posted there
- **The lectures will be recorded**
  - Again, I highly encourage you to come to class (preferably in-person, but next best on zoom), rather than only watching asynchronously

Teaching Assistants

Brett Daley
Esraa Elelimy
Olya Mastikhina
Golnaz Mesbahi
Kevin Roice
Csongor Szepesvari

- **Office hours:** see eClass for excel sheet to sign up for office hours slot
- **Office hours start next week**

Unofficial Lab Session

- There is no official lab
- One of the office hours will be a 2 hour chunk in a large classroom, to facilitate asking about assignments
  - **Wednesday 4-6 pm**
  - CSC B-43
- The first Unofficial Lab Session next week will be a mathematics refresher session, where you can go over exercise questions and ask TAs questions if you get stuck (see https://eclass.srv.ualberta.ca/pluginfile.php/9124125/mod_resource/content/1/267_review.pdf)
- If you cannot make this time from 4-6 pm, there will be two other office hours where you can get help

Readings

- Readings are from the Basics of ML textbook
  - Available on course site and written by myself
  - It's a fast read
  - I have modified these notes each time I have taught this course. They are stabilizing, but I have a few more clarifications to add
- See the schedule for sections and for reading deadlines
- Readings have an associated marked component called Readings Exercises

Lectures

- Lectures will mostly involve me writing on my iPad (like a whiteboard)
- I highly encourage you to ask any question
  - You can raise your hand and then ask outloud
  - You can type questions in Zoom chat
  - We will use Discord for any questions you think of outside of class, that I and the TAs (and your peers) will answer on Discord
- We will have short 3 minute breaks in class
  - 80 minutes is otherwise an awfully long time to listen to someone talk
  - Sometimes I will put up an exercise question for the break
- I will post my written notes afterwards and videos will be published

You should still take your own notes

- The notes I upload will be whatever is left at the end of class. This means
  - It won't include things I said outloud
  - Sometimes I might modify a picture as we go, and it gets a bit messy by the end
  - Sometimes it might be useful to erase a couple of things, and so that will also not be in the uploaded notes
- Further, there is a known phenomena where **writing things down helps you remember and learn them**
  - If you write it, then you might realize that something is confusing

Course Discussion

- We have create a **Discord group; please sign up!**
  - **Details in FAQ and Getting Started linked on eClass**
- I want to generate as much class discussion as possible
- Please go there first to ask questions
- Please answer your classmates questions!
  - We'll step in if there is misinformation, but in many cases you can all help each other faster than we can get to the question
  - Peer discussions can very beneficial

Some Discord Rules

- **Please keep discussions respectful and about course content**
  - We have the channel random if you want to post less pertinent things, like being worried about the exam, or expressing excitement about something
- **Feel free to discuss Assignment questions on Discord**. You should avoid giving away answers, but you can help each other out.
  - Don’t answer the question: What is the answer to Q1?
  - Do answer: I’m getting these weird predictions from my learned function. Has anyone else seen this?
  - Do answer: My code is running really slow. Anyone have any ideas what might be wrong?
  - Do answer: I’m seeing NaNs and for the life of me cannot figure out why. Any thoughts?
  - Do answer: My derivation has gotten really complicated. Did other people also have complicated answers, or did I do something wrong?
- **TAs will not answer direct messages**. You must email the TA email address.

Course Pre-reqs

- Official pre-reqs
  - Calculus 1, Computing 1 (CMPUT 174/274)
- Co-requisites include
  - Stats 1, Linear Algebra 1 (Math 125/127), CMPUT 175, CMPUT 272
- Co-requisites are ideally taken as Pre-reqs, but are not strictly necessary as pre-reqs
  - This gives you the flexibility in scheduling
  - But you still have those concepts reinforced either before or during this course, so that by the middle of the course it is helping

Course Expectations

- This course does not have many official pre-reqs, and I do not expect you to be skilled yet with math
  - Some students will have more math background; don’t worry too much about it
- But, I expect that you want to learn these foundations, even if they are hard
- I expect you to ask questions and come to me during the course if you are struggling
- I expect some amount of struggle and a somewhat steep learning curve; as the course progresses, it will become easier
- We will all make mistakes (me, you, the TAs); communication is important to mitigate this
- I welcome feedback, both during and outside of lecture

A brief interlude with some unsolicited advice
(after all, I’m here to impart my knowledge to you)

"Why is there so much math?"

- **This course is very mathematical**, with detailed derivations
  - This is **absolutely necessary**
- "But I just want to use machine learning to solve Problem X!"
  1. **Applying algorithms correctly** is much easier when you understand their development and assumptions
     - You will be more effective at solving Problem X if you **understand the algorithms** that you apply
     - This means understanding their derivation
  2. **Formalizing the problem** is often half the battle to solving it effectively!
     - Comfort with math is an important part of being a computer scientist

Problem solving

- CS is about problem solving through the medium of computing
  - Not about becoming an expert programmer
- Primary goal is carefully designing solutions to problems, by:
  - **Formalizing** the problem
  - **Understanding** different potential approaches
  - **Evaluating** the solution
- Comfort with mathematical concepts enables **clarity** through logical thinking

“Why can’t we just use packages?”

- Possible quote: “I did a bunch of cool stuff with pytorch this summer. Why don’t I get to do that in this course?”
- Using packages is a high-level rather than a low-level enterprise
- Consider the following analogy. You are taking an operating systems course
- They don’t teach you how to be an expert Mac OSx user
  - Though knowing all the shortcuts would be pretty useful
- They teach you the fundamental ideas underlying operating systems
  - deadlocks, interrupt processing, process scheduling, …

“Why can’t we just use packages?”

- Possible quote: “I did a bunch of cool stuff with pytorch this summer. Why don’t I get to do that in this course?”
- They don’t teach you how to be an expert Mac OSx user
  - Though knowing all the shortcuts would be pretty useful
- They teach you the fundamental ideas underlying operating systems
  - deadlocks, interrupt processing, process scheduling, …
- In the “real-world”, you might feel like being an expert Mac user ends up being more useful than knowing about deadlocks, but that doesn’t mean that that is what we should be teaching you about operating systems
  - If you wanted that, then you could take a course on using Macs

“Why can’t we just use packages?”

- Possible quote: “I did a bunch of cool stuff with pytorch this summer. Why don’t I get to do that in this course?”
- In the “real-world”, you might end up using pytorch a lot. But that doesn’t mean that is what we should focus on
  - Lots of people can learn how to use packages (or take a course)
  - Your competitive advantage is knowing the foundations, that will let you use those packages more effectively and solve problems more effectively

While I am giving you unsolicited advice…

- **Writing** and **clear thinking** are extremely important skills for CS
- You learn a lot of really useful things in CS, that will serve you well afterwards, but only if you take the **learning** seriously
  - CS is not a professional degree where you need the piece of paper
  - You can get jobs in CS without a CS degree, and a CS degree does not guarantee you get a job in CS
  - If you get a CS degree with the bare minimum, then …
- Our goal is to teach you how to approach problems and continue learning in the future (not necessarily to teach you the tools or languages of today)

Some career advice

- **Internships** are a good idea
- We are not a professional organization, but you are skilled. Be **confident** in yourself (aka, you're pretty great)
- Practice some **professional skills:** writing resumes, giving talks (toastmasters), general writing clarity (take a writing course)
- If you want to go into **graduate studies**, you need good marks
  - Or you need to start working with a prof in undergraduate research early
  - The UofA is a research-focused university, meaning profs spend much of their time on research (good for you too since they are at the cutting edge)

Back to the course organization

Grading

- 30%: Assignments
  - Mixture of mathematical problems and programming exercises
- 5%: Quiz on **February 16**
- 20%: Midterm exam on **March 16**
- 35%: Final exam **April 17**
- 10%: Readings Exercises

More on Grading

- **There will be no arguing for marks, for assignments or exams**
  - Arguing introduces a lot of bias: it benefits those willing to argue and punishes those that are not willing to argue
  - We will of course fix incorrectly tabulated marks
- We have a double marking system for exams, to try to ensure consistency
- Assignments are meant to be challenging. Exams are much simpler and meant to test if you understood the fundamental ideas.
- Final letter grades are given based on relative performance, but they are not on a Gaussian curve. The cut-offs are adjusted each year

More on Grading

- Final letter grades are given based on relative performance, but they are not on a Gaussian curve. The cut-offs are adjusted each year
- Last year's announcement on letter grades: "A+: mark > 97, A: mark > 92, A-: mark > 87, B+: mark > 81, B: mark > 76, B-: mark > 72, C+: mark > 68, C: mark > 63, C-: mark > 57, D: mark > 53
  You might wonder where this spacing comes from. It gets adjusted each year due to performance. Grade allocation is tough, since there are always some near the boundary. But I hope you remember that I have already erred on the side of being somewhat generous (rather than strict) for the grade decisions."

Assignments

- Four assignments
- Coarse binned grading:
  - 80 - 100 → 100
  - 60 - 80 → 80
  - 40 - 60 → 60
  - **0 - 40 → 0**

Three exams

- Giving **clear** answers to written answer questions is a **skill**
  - It takes practice!
  - First quiz is your chance to practice this skill with low stakes
  - It's only 5% of the grade (less than one assignment)
- Practice questions will be available
- Exams are in-person
  - Please contact me asap if you will miss one
- For all exams you are allowed a **two page cheat-sheet**
  - Must be created on your own, not collaboration allowed on cheat sheets

Collaboration policy

Detailed version on the syllabus section of the website

You are **encouraged to discuss assignments** with other students:

1. You must **list** everyone you talked with about the assignment.
2. You **must not** share each other's **written work or code**.
3. You must **write up** your solutions individually

No collaboration allowed on **exams**

Academic conduct

- Submitting someone else's work as your own is **plagiarism**.
- So is helping someone else to submit your work as their own.
- We report **all cases** of academic misconduct to the university.
- The university takes academic misconduct **very seriously**.
  Possible consequences:
  - Zero on the assignment or exam (virtually guaranteed)
  - Zero for the course
  - Permanent notation on transcript
  - Suspension or expulsion from the university
- **If you are thinking of cheating, since you are stuck or doing poorly, please just talk to me instead. We'll figure it out.**

Additional Questions

- Any questions you have are likely answered in the FAQ and Getting Started document that we have linked on eClass
  - Policies like “No late assignments accepted”, “How to contact TAs”, “What to do if you are going to miss a deadline or exam”
  - “How can I get extra resources?” and “How can I brush up on my math background?”

Readings

- **It is critical that you do the readings**
- I wrote the notes, and in-class lectures follow them quite closely
- If you read and understand the notes, you have learned a lot about ML
- Ideally, you read the notes before I lecture on it, to help you understand them better (seeing it a second time)
- Marked **Readings Exercises** encourage you to actually do the readings

Readings Exercises

- Four readings exercises
  - Ch 1-3, Ch 4-5, Ch 6-9 and Ch 10-12
- Complete Exercises on eClass (auto-graded questions)
- Each Readings Exercise has 5 questions for the relevant readings
- To help you follow the readings, rather than to really test you, so
  - The questions are relatively simple
  - You can two attempts for each Readings Exercise (best of the two)
  - We drop the lowest mark of the four

Finally: Using Julia instead of Python

- Julia is a newish programming language, focused on numerical computing
  - syntax similar to Matlab
- **I know that this is likely a big deal for some of you**
- But, do not fret!
  - We have provided a tutorial notebook and other helper resources
  - The programming is through notebooks; the number of lines of code you implement is not that big
  - You will find Julia quite easy to use and the notebook interface nice
  - In the notebooks themselves, we tell you about the syntax you need