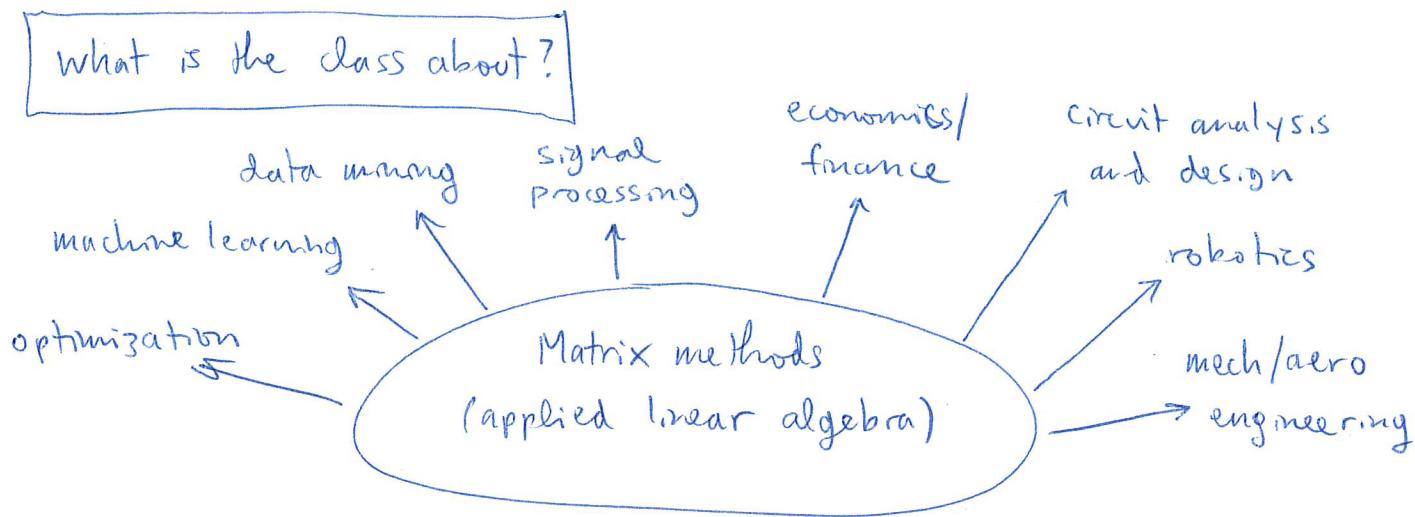


Welcome to ECE/CS/ME 532 "theory and applications of pattern recognition".

Instructor: Laurent Lessard (laurent.lessard@mcgill.ca)



- theory we cover is fundamental in many fields.
- this class focuses on (mostly) machine learning / pattern recognition.

Who should take this class?

- interested in machine learning
- interested in one of the other topics above but lack math background.
- interested in career related to data science / machine learning.
- interested in research → gateway to more advanced courses (e.g. CS 761).

Prerequisites: {

- vector calculus / mathematical maturity
- exposure to linear algebra
- exposure to scientific computing (Matlab/Python/Julia).

Today's lecture

→ overview of some types of problems we will learn about this semester.

## Misc. topics

\* Coding!

Matlab :

- + built for matrices! slick interface, good help files, debugging / profiling, industry / engineering standard.
- requires license (free for UW students), as a programming language it's chunky

Python w/ numpy :

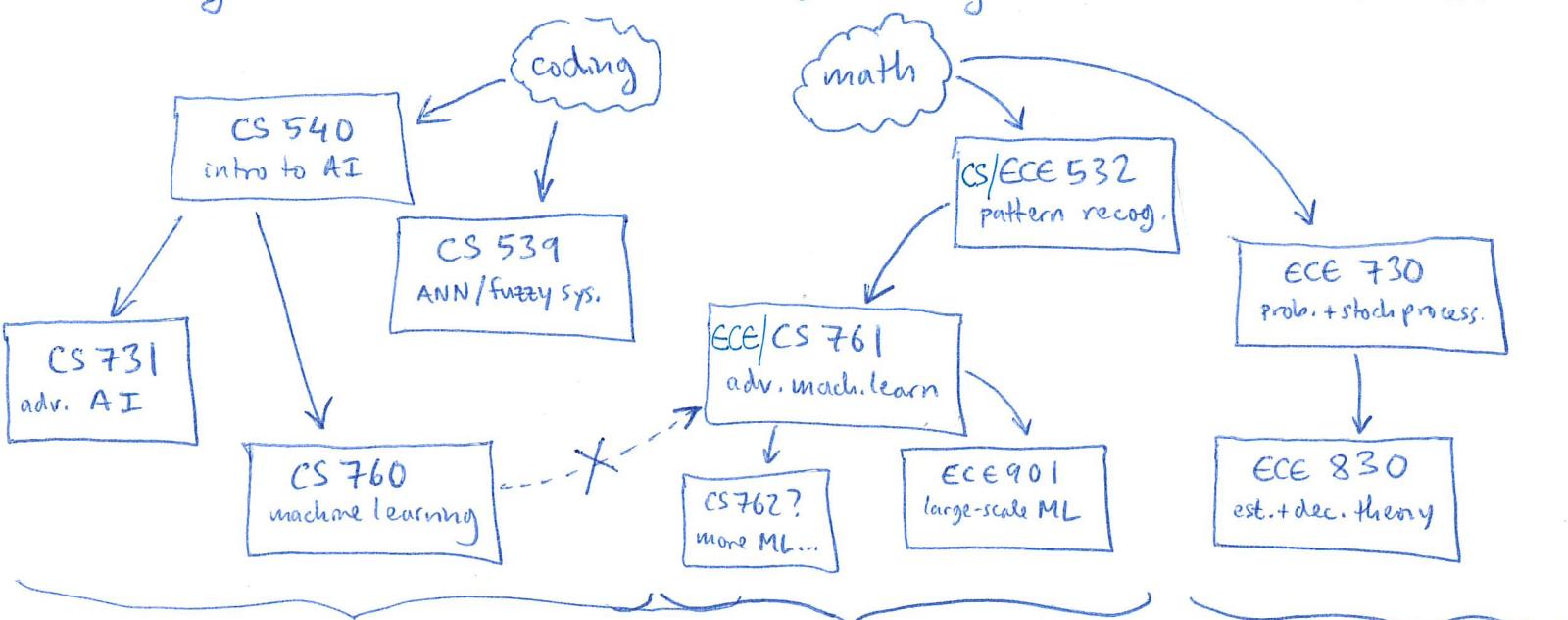
- + fast, used by data science industry, a legit language on its own, functional, modular.
- syntax not ideal. Not originally built for matrix computations. (e.g. 0-based indexing)

Julia:

- + built for matrices, very modern, JIT technology, very fast, syntax similar to both Matlab & Python
- relatively new, sometimes difficult to find help / support. Plotting tools lacking.

If in doubt, use Matlab. It's what I'll use in class.

Either way, focus of ECE532 is on the math, not on writing code. This is not a programming class! Other classes:



more applied survey-style courses.  
(typically lots of coding)

ML track.  
(math + research focus)

Signal processing  
track.

## Administrative stuff

- [informal class survey : waitlist, gr/lug , Matlab/Python/Julia? , ECE/CS/ME?]
- office hours: TBA ~ will announce + start next week.
- class is being recorded, I will repeat questions (annoying, I know!)
- textbook : Free online, will use for ~  $\frac{1}{2}$  of course.  
see class website for details.
- grading: {
  - Homework (20%): roughly weekly, roughly graded 0-4 about 10 in total. (due Friday).
  - Exams (2x 20%): two midterms after hours. tentative dates: Mon Oct. 10 , Mon Nov. 21.
  - Project (40%): groups of up to 3. Details TBA.  
There is no final exam!
- Late policy survey: {
  - (of course, exceptions for exceptional circumstances... email instructor.)
  - 1) no late HW accepted, automatically drop lowest.
  - 2) all HW count, option to turn it in on Monday (-2 pts)
- We will use Gradescope for assignments (more about this on Thu.)

## Important To Do

try it out!

- 1) We will use Piazza for Q&A. Sign up now.
- 2) Piazza has link to class website (notes, admin, HW#0 )
- 3) Enroll ASAP if interested! (by Sept 9 or \$50 fee)

Today's lecture: Vectors and Matrices in machine learning / pattern recognition (3)  
with examples / preview of what's to come.

A matrix is an array of numbers (data).

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}$$

### Example 1

"term-document matrix" used in "information retrieval".

Doc 1: "The Google matrix  $P$  is a model of the internet".

Doc 2: " $P_{ij}$  is nonzero if there is a link from webpage  $j$  to  $i$ ".

Doc 3: "The Google matrix is used to rank all web pages!"

Doc 4: "the ranking is done by solving a matrix eigenvalue problem".

Doc 5: "England dropped out of the top 10 in the FIFA ranking".

Keywords/terms are underlined. Count the frequency:

term	Doc 1	Doc 2	...	
eigenvalue	0	0		
England	0	0		$A \in \mathbb{R}^{10 \times 5}$
FIFA	0	0		
Google	1	0		
internet	1	0		
link	0	1		$A_{ij} = \text{number of times term } i \text{ occurs in document } j.$ (or it can be binary)
matrix	1	0		
page	0	1		
rank	0	0		
web	0	1		

Suppose we want to find documents relevant to Google and internet  
 this is represented by the query vector:

$$q = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{10}$$

(the query is itself a document).

vectors are denoted by lower-case letters and one columns by default.

Mathematical problem: which columns of A are "close" to q?

→ we need some notion of "distance". Some examples:

$$\ell_1 \text{ norm} : \|a_{\cdot j} - q\|_1 = \sum_{i=1}^{10} |a_{ij} - q_{ij}|$$

$$\ell_2 \text{ norm} : \|a_{\cdot j} - q\|_2 = \sqrt{\sum_{i=1}^{10} |a_{ij} - q_{ij}|^2}$$

$$\ell_\infty \text{ norm} : \|a_{\cdot j} - q\|_\infty = \max_i |a_{ij} - q_{ij}|$$

$$\ell_0 \text{ distance} : \|a_{\cdot j} - q\|_0 = \sum_{i=1}^{10} \mathbf{1}_{\{a_{ij} \neq q_{ij}\}}$$

(count number of non-matches)

different notions of "distance" appropriate for different circumstances!

More on this later...

in real-world information-retrieval,  $m \approx 10^6$  terms,  $n \approx 10^9$  documents.

also, most entries of A are zero (A is sparse).

## Matrices as linear operators

matrices can be viewed as a table of data, or as linear operators  
 if  $A \in \mathbb{R}^{m \times n}$ , then  $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ . so  $y = Ax$  ( $x \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^m$ ).

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}}_{y \in \mathbb{R}^{2 \times 1}} = \underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}}_{A \in \mathbb{R}^{2 \times 3}} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}}_{x \in \mathbb{R}^{3 \times 1}} = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \end{pmatrix}$$

$$y_i = \sum_{j=1}^n a_{ij} x_j \quad (i=1, 2, \dots, m)$$

\* it's a linear combination (mixture) of columns:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix} x_1 + \begin{pmatrix} a_{12} \\ a_{22} \end{pmatrix} x_2 + \begin{pmatrix} a_{13} \\ a_{23} \end{pmatrix} x_3$$

$$y = \sum_{j=1}^n a_{ij} x_j$$

\* it's an inner product (dot product) of rows:

$$y_1 = (a_{11} \ a_{12} \ a_{13}) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = a_{1 \cdot} \cdot x$$

$$y_2 = (a_{21} \ a_{22} \ a_{23}) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = a_{2 \cdot} \cdot x$$

(how aligned is each row of A with x?)

Both interpretations are useful!

More on this later...

### EXAMPLE

	steak	potatoes	coleslaw	b. beans
protein	5	1	0	2
carbs	0	4	3	6
fats	2	0	1	1

Servings:

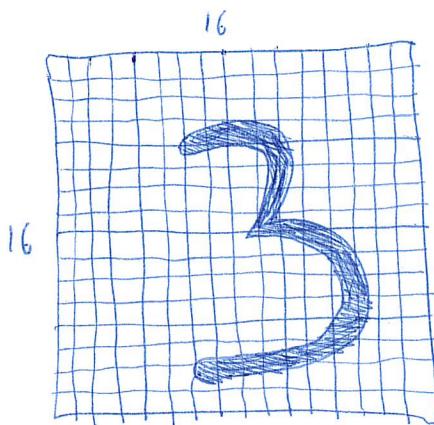
1	steak
1	potatoes
1	coleslaw
0	b. beans

energy (cal)

4	protein
4	carbs
9	fat

(6)

## Example 2 classification of hand written digits



think of this as a vector  $a \in \mathbb{R}^{256}$   
of 0's and 1's (or grayscale values).

take several samples:

$$\boxed{3} \quad \boxed{3} \quad \boxed{3} \quad \boxed{3} \quad \dots$$

\* stack the columns into a big matrix:

$$A^{(3)} = \begin{bmatrix} & & \\ & & \\ & \dots & \\ & & \\ & & \dots & \\ & & & \dots & \end{bmatrix} \in \mathbb{R}^{256 \times N} \quad (\text{say } N=10^4)$$

↑↑↑  
different samples of "3".

\* repeat for all other digits:

$$A = \begin{bmatrix} A^{(0)} & A^{(1)} & \dots & A^{(9)} \end{bmatrix} \in \mathbb{R}^{256 \times 10^5}$$

o's    1's    ...    9's.

given some handwritten digit  $b \in \mathbb{R}^{256}$ , is it a 0, 1, 2, ... or a 9?

\* similar to term-document problem. i.e.

$$\min_j \|b - a_{ij}\| \quad (\text{find column of } A \text{ that best matches } b)$$

- this is very inefficient (requires  $10^5$  comparisons).
- not practical if we need to recognize a large number of digits  
(i.e. there are many unknown  $b$ 's.)

(7)

Approach: For each digit, find a small number of "canonical" or "representative" digits such that every column of  $A^{(3)}$  is well approximated by a linear combination of representatives.

Let representatives be:  $\boxed{\dots} \in \mathbb{R}^{256 \times 12} = U^{(3)}$ .

For each  $a_{:,j}$ , there is some  $w^{(j)}$  such that  $a_{:,j} \approx U^{(3)}w^{(j)}$

(chosen such that  $\min_w \|a_{:,j} - Uw\|$  is small for each  $j$ .)

\* Instead of using  $A^{(3)}$ , use  $U^{(3)}$ !

before:  $\min_j \|b - a_{:,j}\| \rightarrow$  after:  $\min_w \|b - Uw\|$

$(256 \times 10^5 \text{ ops}) \quad (256 \times 12 \text{ ops}) \quad \sim 10^4 \text{ times faster!}$

\* Compute  $\min_w \|b - U^{(1)}w\|, \min_w \|b - U^{(2)}w\|, \dots, \min_w \|b - U^{(9)}w\|$   
smaller = better fit.

{ Process of approximating columns of  $A$  using  
 representatives  $U$  is called the  
 singular value decomposition (SVD) }  $\rightarrow$  second part of class  
 (btwn midterm #1, #2)

{ Process of solving  $\min_w \|b - Uw\|$  is  
 called least squares (regression) }  $\rightarrow$  first part of class  
 (before midterm #1)

Example 3. Recommendation systems (information filtering)

examples: Netflix, YouTube, Amazon, Pandora, ...

Netflix example:

movie	user 1	user 2	...
Jurassic Park	-	-	
Love Actually	-	4	
Mission Impossible	5	-	(matrix typically has
Braveheart	2	5	many missing entries)
Finding Dory	-	2	
Batman v. Superman	4	-	
:	:	:	

New user: rates Jurassic Park 5 stars.

what should we recommend?

general idea: millions of users... can we think about "canonical" or  
"representative" users? Can we do this automatically?