Data Mining 2026 Spring!

This class:
- linear algebra + probability
- machine learning
- foundations of gen AI

challenging course!

Prereqs: linear algebra, probability, algorithms, calc 3

Time Expectation: 12.5 hr/week

2.5	1	1	2	1	1	1
class	Problem Set			OH	discussion	"rock"

- Earlier = better
- More drawing = better
- More talking = better

www.rtealwitter.com/datamining2026spring

- discord for communication
- reading for each lecture
- these slides are online

Plan:
- Warm up
  - ↳ math review
  - ↳ Page Rank
- Supervised Learning
- Unsupervised Learning

$\boxed{\text{Math Review}}$

machine learning ?

calculus and linear algebra!

Derivatives

$f : \mathbb{R} \to \mathbb{R}$

"maps real number to real number"

$$f'(x) = \frac{\partial}{\partial x} f(x)$$

$$= \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Function: $f(x)$	Derivative: $\frac{\partial}{\partial x} f(x)$
x^2	$2x$
x^a	
$ax+b$	
$\ln(x)$	
e^x	

Q: If my investment earns 10% per year, how much more will I have in 60 years?

In ML, we'll combine many functions (e.g., neural net)

$f: \mathbb{R} \to \mathbb{R}$ $\quad$ $g: \mathbb{R} \to \mathbb{R}$

$f(x) = \ln(x)$ $\quad$ $g(x) = x^2$

Chain Rule

$$\frac{\partial \left[g(f(x))\right]}{\partial x} = g'(f(x))\, f'(x)$$

$$=$$

Product Rule

$$\frac{\partial}{\partial x}\left[g(x) \cdot f(x)\right]$$

$$= g(x)\, f'(x) + f(x)\, g'(x)$$

$$=$$

Gradients

In ML, process high-dim input

$f: \mathbb{R}^d \to \mathbb{R}$

"maps d real numbers to one real"

$f(x_1, x_2, \ldots, x_d)$ ← visualize as mountain with x_1 north/south, x_2 east/west

$$= f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}\right) = f(\underline{x})$$

$\frac{\partial}{\partial x_i} f(\underline{x})$ = how changing x_i changes f

$$= \lim_{h \to 0} \frac{f(x_1, \ldots, x_i + h, \ldots, x_d) - f(x_1, \ldots, x_i, \ldots, x_d)}{h}$$

$$\nabla_{\underline{x}} f(\underline{x}) = \begin{bmatrix} \frac{\partial f(\underline{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\underline{x})}{\partial x_2} \end{bmatrix}$$

Vector and Matrix Multiplication

↑ crucial to ML, neural nets only succeeded with GPUs

$\underline{u} \in \mathbb{R}^d \qquad \underline{v} \in \mathbb{R}^d$

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_d \end{bmatrix} \qquad \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix}$$

$$\underline{u} \cdot \underline{v} = \sum_{i=1}^{d} u_i v_i = u_1 v_1 + u_2 v_2 + \dots + u_d v_d$$

$$= \underline{u}^T \underline{v} = \begin{bmatrix} u_1 & u_2 & \dots & u_d \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix}$$

$$= \underline{v}^T \underline{u} = \begin{bmatrix} v_1 & v_2 & \dots & v_d \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_d \end{bmatrix}$$

$$= \langle \underline{u}, \underline{v} \rangle$$

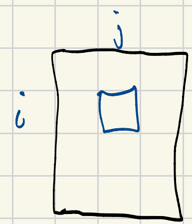
ℓ_2-norm

$$\|\underline{u}\|_2^2 = \underline{u} \cdot \underline{u} = \sum_{i=1}^{d} u_i^2$$

$A \in \mathbb{R}^{d \times m} \qquad B \in \mathbb{R}^{m \times n}$

$$\begin{array}{cc} & \\ i & \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ a_{21} & & & & \\ \vdots & & & & \\ & & & & a_{dm} \end{bmatrix} \end{array} \qquad \begin{array}{c} j \\ \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \vdots & & \\ b_{m1} & & b_{mn} \end{bmatrix} \end{array}$$

$$AB \qquad \underbrace{d \times m \;\; m \times n}_{d \times n}$$



$$[AB]_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}$$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$AB = ?$ $\qquad BA = ?$