Introduction to Statistical Machine Learning
CSC/DSCC 265/465

Lecture 3: Math Review – Part II

Cantay Caliskan

Plan for today

- Proofs
- Matrix algebra
- Bayesian vs. Frequentist schools
- Probability

Notes and updates

Notes and updates

- Let's have a two-day extension on the 1st problem set
  - New deadline: Friday, January 28, 11:59 PM

- Reminder: Please let Ms. Lisa Altman (lisa.altman@rochester.edu) know about registration-related issues

- Please use **Slack** for questions and discussion

UNIVERSITY of ROCHESTER

Proofs

Proofs in Math

- Definition: A proof is a series of **mathematical statements** that have a **logical flow**
- You start with what you assume to be **true**, and end with what you are trying to **prove:**
  - **Beginning:** Things we are assuming to be true (definitions etc.)
  - **Middle:** Logical statements following from the Beginning
  - **End:** Show what you are trying to prove or provide a counterexample
- Your options:
  - You have to plan your proof. Three options:
    1) Show that a statement holds by deriving your explanations from other **related Math rules**
    2) Show that a statement holds for **n-many numbers** and the **(n+1)th number** (***proof by induction***)
    3) Show that a statement doesn't hold by giving one **counterexample**

Hint: Think about the edge cases!

Proof: Example

> ***Example 1.*** *Using the field axioms, prove that $a(b-c) = ab - ac$ for any real numbers a, b, c. You may use the fact that $x.0 = 0$ for any real number x.*

BEGINNING
- field axioms
- definition $x - y = x + (-y)$
- given $x.0 = 0$

MIDDLE

$$\begin{array}{rcll} a(b-c) & = & a(b+(-c)) & \text{definition} \\ & = & ab + a(-c) & \text{distributive law} \end{array}$$

$$\begin{array}{rcll} ac + a(-c) & = & a(c+(-c)) & \text{distributive law} \\ & = & a.0 & \text{additive inverse} \\ & = & 0 & \text{given} \\ \therefore\ a(-c) & = & -(ac) & \text{definition of additive inverse} \end{array}$$

$$\therefore\ ab + a(-c) = ab - ac$$

END

$$\therefore \text{ by line 2, } \ a(b-c) = ab - ac \ \text{ as required}$$

Source: Eugenia Cheng

Very helpful resource: https://deopurkar.github.io/teaching/algebra1/cheng.pdf

Proof: Another example

Example 3. *Prove by induction that* $\forall n \in \mathbb{N}, 1 + \cdots + n = \dfrac{n(n+1)}{2}$

BEGINNING Principle of Induction

MIDDLE

$$\begin{aligned} \text{for } n = 1, \text{ LHS} &= 1 \\ \text{RHS} &= \frac{1(1+1)}{2} \\ &= 1 \end{aligned}$$

$\therefore$ result is true for $n = 1$

If result is true for $n = k$ then

$$\begin{aligned} 1 + \cdots + k + (k+1) &= \frac{k(k+1)}{2} + (k+1) \\ &= \frac{k(k+1) + 2(k+1)}{2} \\ &= \frac{(k+1)(k+2)}{2} \qquad \text{i.e. result true for } n = k+1 \end{aligned}$$

$\therefore$ result true for $k \implies$ result true for $k+1$

END $\therefore$ by the Principle of Induction, the result is true for all $n \in \mathbb{N}$

QED (Quod erat demonstrandum): 'This was what we needed to show.'

And: let's continue with matrices

Matrix algebra

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special matrices
- Matrix inverse

Let’s continue here.

- Matrix rank
- Eigenvalues and Eigenvectors
- Matrix Calculus

Matrix Inverse

- Given a matrix **A**, its inverse $\mathbf{A}^{-1}$ is a matrix such that:
  - $\mathbf{A}*\mathbf{A}^{-1} = \mathbf{A}^{-1}*\mathbf{A}$
- Example:

$$\mathbf{A} = \begin{pmatrix} -1 & \frac{3}{2} \\ 1 & -1 \end{pmatrix} \quad \mathbf{A}^{-1} = \begin{pmatrix} 2 & 3 \\ 2 & 2 \end{pmatrix}$$

Thus: **A** is an **invertible** matrix, or **non-singular**.

- Some further properties:

$$(\mathbf{A}^{-1})^{-1} = \mathbf{A}$$

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

- Question: When is a matrix not invertible?
- Answer: When the determinant is zero.

Inverting a matrix

- Let's take a matrix ***A***

$$\mathbf{A} = \begin{pmatrix} -1 & \frac{3}{2} \\ 1 & -1 \end{pmatrix}$$

- <u>First</u>: Create an augmented version of the matrix

$$\left(\begin{array}{cc|cc} -1 & \frac{3}{2} & 1 & 0 \\ 1 & -1 & 0 & 1 \end{array}\right)$$

- <u>Goal</u>: Create an identity matrix on the left-hand side through linear combination of the identity matrix on the right-hand side

$$\left(\begin{array}{cc|cc} -1 & \frac{3}{2} & 1 & 0 \\ 0 & \frac{1}{2} & 1 & 1 \end{array}\right) \left(\begin{array}{cc|cc} -1 & 0 & -2 & -3 \\ 0 & \frac{1}{2} & 1 & 1 \end{array}\right) \left(\begin{array}{cc|cc} 1 & 0 & 2 & 3 \\ 0 & 1 & 2 & 2 \end{array}\right)$$

This is the inverse!

<u>Question</u>:
What is $\mathbf{A}^{-1}$?

Inverse of a matrix

- How is the inverse of a matrix useful?
  - Reminder: We cannot divide matrices
  - But: We can multiply a matrix by its ***inverse***

- ***Let's say we have a differential equation with 'some' unknowns:***
  - What is the best linear estimator that summarizes the relationship?

$$\hat{\boldsymbol{\beta}} = \left(\mathrm{X}^{\mathsf{T}}\mathrm{X}\right)^{-1}\mathrm{X}^{\mathsf{T}}\mathbf{y}$$

The coefficient vector for the least-squares hyperplane

  - But, really helpful in many other fields:
    - **3D graphics** -> Calculating the position of an object in space
    - **Encryption** -> Decoding a message

Pseudoinverse

- It is 'sometimes' hard to find the inverse of a matrix
- Also called ***generalized inverse*** or ***Moore-Penrose inverse***
- Common uses:
  - Compute a 'best fit' solution to a system of linear equations
  - Find the minimum norm solution to a system of linear equations

- Say you have three matrices **A**, **B** and **X**, where:
  - $\mathbf{AX = B}$
  - You know **A** and **B**, and you want to solve for **X**

- You could calculate the inverse and pre-multiply by it:
  - $\mathbf{A^{-1}AX = A^{-1}B}$
  - $\mathbf{X = A^{-1}B}$

This is not always solvable!

Pseudoinverse

- **Python** command: np.linalg.inv(A)*B

```
LinAlgError: Singular matrix
```

```
Read 75 images.
enter the number class 15
enter the eigen numbers you want 37
[U,D,V]=svd(inv(sw)*sb);
                        !--error 19
Problem is singular.
```

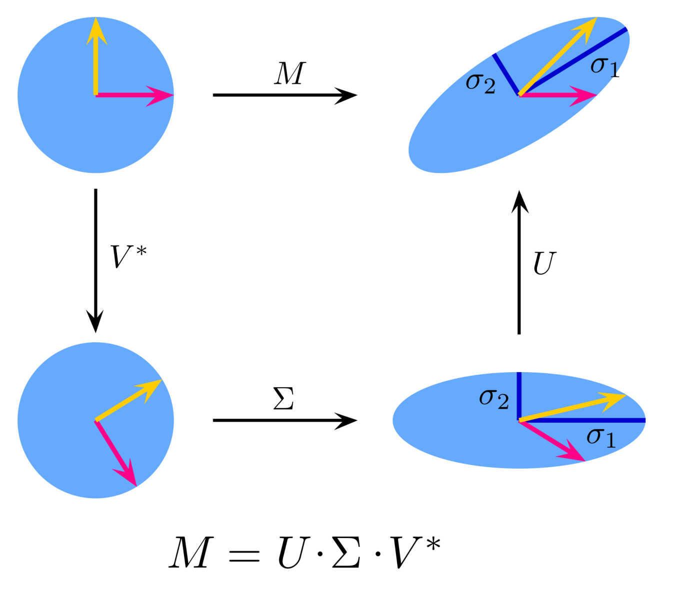
Oh, matrix cannot be inverted!

- Calculating inverses for large matrices is problematic, because:
  - Some matrices are ***sparse*** (= have many zeros)
  - Some matrices have ***very large*** and ***very small*** numbers together
  - But, maybe your matrix is ***non-invertible***

- To solve this problem: Python will try numerical solutions:
  - If there is no exact solution, <u>closest solution</u> will be returned
  - If there are many solutions, <u>smallest solution</u> will be returned

Pseudoinverse

- How to compute a ***pseudoinverse***:
  - Rank decomposition
  - The QR method
  - Singular value decomposition (SVD)
  - Block matrices
  - The iterative method of Ben-Israel and Cohen



Singular value decomposition (SVD)

Linear Independence

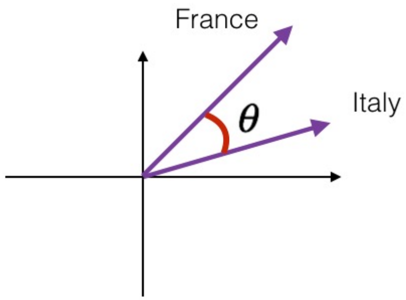
- Closely conncted to the ***invertibility*** of a matrix:
  - Suppose we have a set of vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_n$

  - **Linear dependence**: If $\mathbf{v}_1$ can be expressed as a linear combination of the other vectors $\mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_n$, then $\mathbf{v}_1$ is ***linearly dependent*** on the other vectors
    - Reminder: What is a linear combination?
      - Let's say we have three scalars **a**, **b**, **c**. Example: $\mathbf{a = 5b + 4c}$
      - Or, let's say $\mathbf{v}_1 = 0.5 * \mathbf{v}_2 - 0.3 * \mathbf{v}_3$

  - **Linear independence:** No vector is a linear combination of the other vectors
    - Common cause: One of the vectors is a **null vector**
    - OR: The vectors are **perpendicular** to each other

Example: Linear (In)dependence

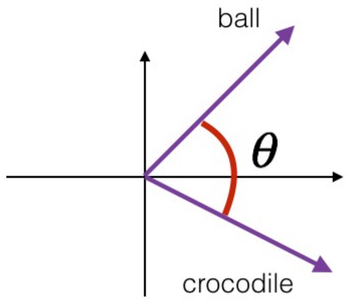
- Cosine similarity, word-embeddings:



France and Italy are quite similar

θ is close to 0°

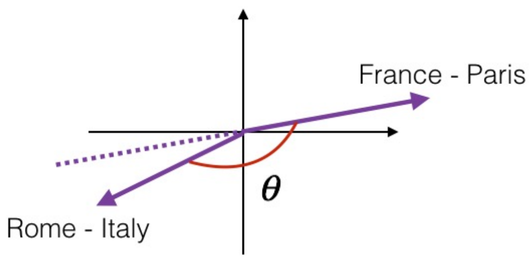
$\cos(\theta) \approx 1$



ball and crocodile are not similar

θ is close to 90°

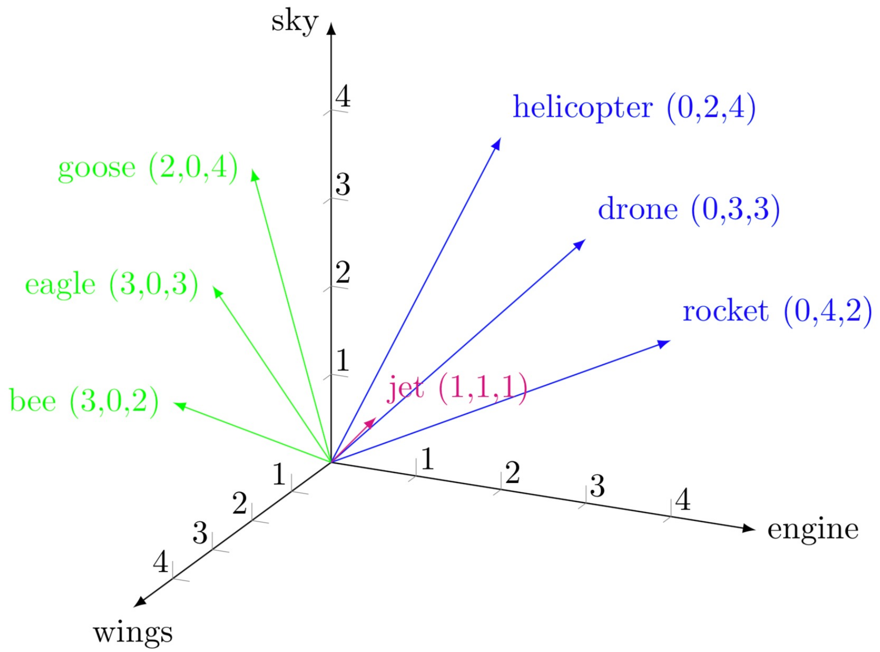
$\cos(\theta) \approx 0$



the two vectors are similar but opposite
the first one encodes (city - country)
while the second one encodes (country - city)

θ is close to 180°

$\cos(\theta) \approx -1$



Matrix algebra

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special matrices
- Matrix inverse
- Matrix rank ← **Let's continue here.**
- Eigenvalues and Eigenvectors
- Matrix Calculus

Matrix rank

- ***Column / row rank***:

- $Col-rank(A)$ = The maximum number of linearly independent column vectors in matrix **A**
- $Row-rank(A)$ = The maximum number of linearly independent row vectors in matrix **A**
- $Col-rank(A) = Row-rank(A)$(always!)

- Question: What are the ranks of the following matrices?

$$\begin{bmatrix} 1 & 0 & 1 \\ -2 & -3 & 1 \\ 3 & 3 & 0 \end{bmatrix}$$

Rank = 2

$$A = \begin{bmatrix} 1 & 1 & 0 & 2 \\ -1 & -1 & 0 & -2 \end{bmatrix}$$

Rank = 1

$$A^{\mathrm{T}} = \begin{bmatrix} 1 & -1 \\ 1 & -1 \\ 0 & 0 \\ 2 & -2 \end{bmatrix}$$

Rank = 1

Question: What is the minimum rank?

Matrix rank

- ***Rank*** tells the dimension of an output (when you want to transform a matrix)

- If rank of matrix A is 1, then the following transformation:

p' = Ap

maps points onto a line.

- Example: Here is a matrix with rank = 1:

$$\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} x \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + y \\ 2x + 2y \end{bmatrix}$$

All points are mapped to the line y = 2x

Matrix rank

- If a square matrix (***m x m***) is rank *m*, we say it's 'full rank'
  - This matrix would map an ***m x 1*** vector uniquely to another ***m x 1*** vector
  - An inverse matrix can be found

- If ***rank < m***, we say it's 'singular'
  - At least one dimension is a linear combination of another dimension
  - There is no inverse

- Reminder: There are no inverses for non-square matrices

Matrix algebra

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special matrices
- Matrix inverse
- Matrix rank
- Eigenvalues and Eigenvectors

Let’s continue here.

- Matrix Calculus

Eigenvector and eigenvalue

- An eigenvector ***v*** is a non-zero vector of a linear transformation ***A*** that <u>does not change its direction</u> when this linear transformation ***A*** is applied to it.

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}, where\ \boldsymbol{v} \neq 0$$

OR

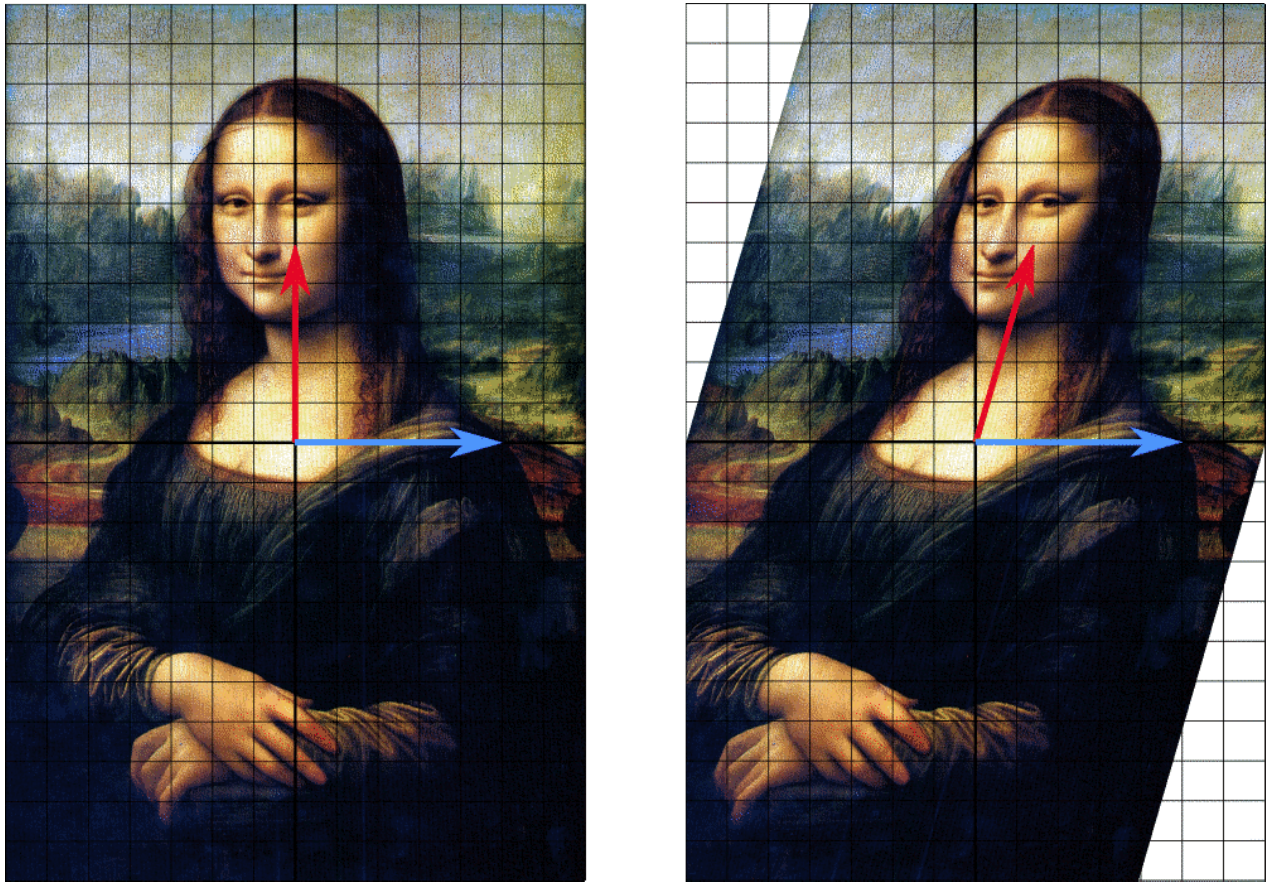
$$(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{v} = \boldsymbol{0}, where\ \boldsymbol{v} \neq 0$$

- Here, λ is a scalar value, and also called an ***eigenvalue***

- <u>Easier interpretation:</u>
    - ***Eigenvector*** points in a direction in which it is stretched by the linear transformation
    - ***Eigenvalue*** is the factor by which it is stretched

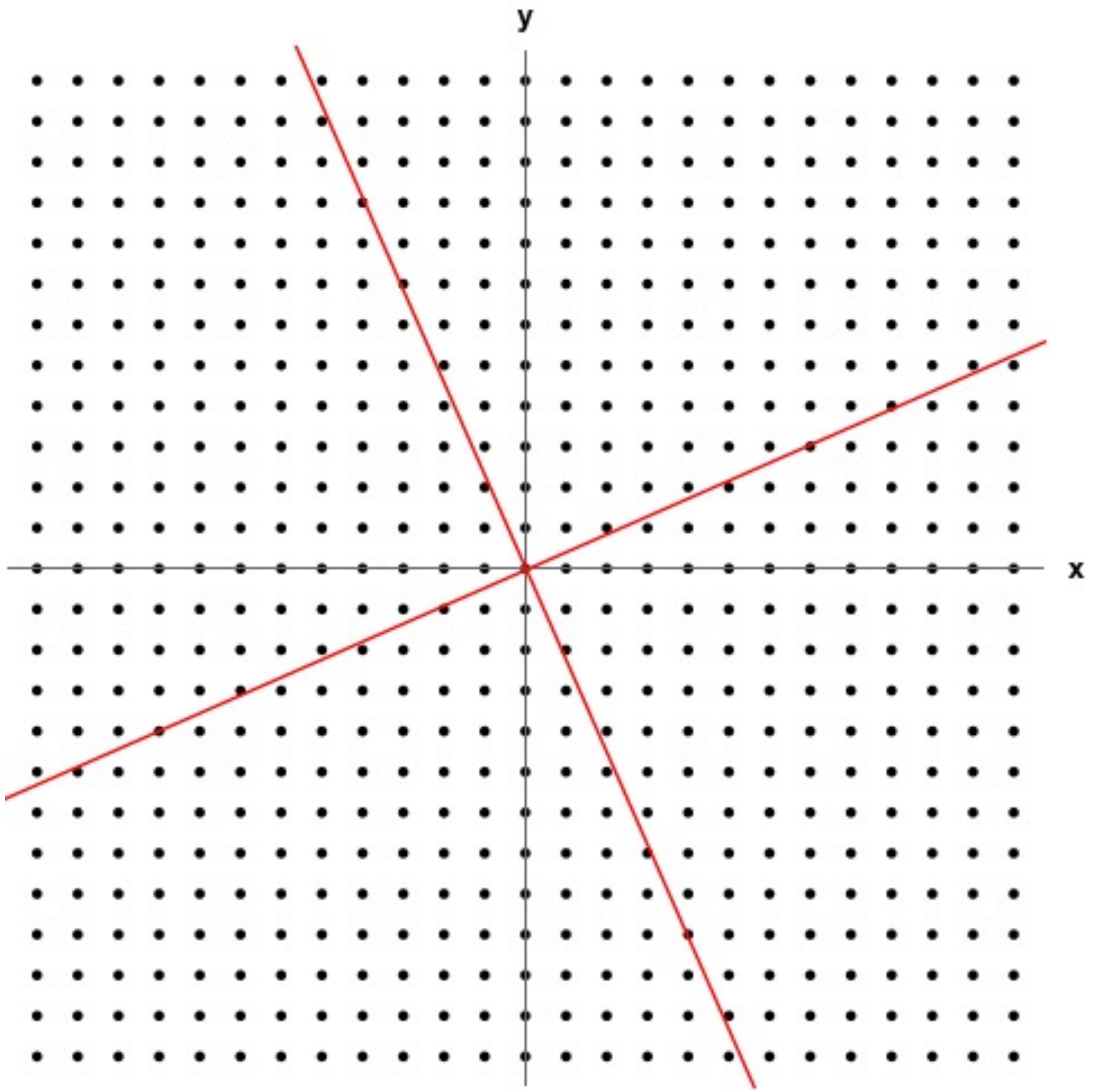
Mona Lisa

This is called shear mapping



- When we rotate Mona Lisa, the **red arrow** changes direction, but the **blue arrow** doesn't
- **Blue arrow** is an eigenvector of this rotation (=linear transformation)

Where are the eigenvectors?



Properties of eigenvalues

- The trace of a matrix ***A*** is equal to the sum of its eigenvalues

$$tr(A) = \sum_{i=1}^{n} \lambda_i$$

- The determinant of a matrix ***A*** is equal to the product of its eigenvalues

$$|A| = \prod_{i=1}^{n} \lambda_i$$

- The rank of matrix ***A*** is equal to the number of non-zero eigenvalues of ***A***

- The eigenvalues of a diagonal matrix ***D*** are the diagonal entries of ***D***

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$

Question: What are the eigenvalues here?

Eigenvector and eigenvalue

- Dimension reduction
  - Example: Principal component analysis in object/image recognition
  - Reminder: A lot of the ML is ‘summarizing data’
- Calculating network centralities
  - Example: PageRank algorithm from Google
- Physics
  - Measurement in quantum mechanics
- Face recognition
- Compressing data
- Regularization
- Noise reduction

Diagonalization (=How to find eigenvalues)

- <u>Question</u>: Find all the eigenvalues and eigenvectors of:

$$A = \begin{bmatrix} 5 & -4 \\ 8 & -7 \end{bmatrix}$$

- <u>Step 1</u>: Compute the characteristic polynomial:

$$\det(A - \lambda I) = \begin{vmatrix} 5-\lambda & -4 \\ 8 & -7-\lambda \end{vmatrix} = \lambda^2 + 2\lambda - 3.$$

- <u>Step 2</u>: Find the unique eigenvalues:
  - The polynomial is solved for $\boldsymbol{\lambda}$ **= -3** and $\boldsymbol{\lambda}$ **= 1**
  - Then: $\boldsymbol{Av = -3 * v}$ $\boldsymbol{and}$ $\boldsymbol{Av = 1 * v}$
  - Solve <u>two</u> equations: $\begin{bmatrix} 5 & -4 \\ 8 & -7 \end{bmatrix} * \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 1 * \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ and $\begin{bmatrix} 5 & -4 \\ 8 & -7 \end{bmatrix} * \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = -3 * \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$
  - Eigenvectors are ***(1,2)*** and ***(1,1)***

Matrix algebra

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special matrices
- Matrix inverse
- Matrix rank
- Eigenvalues and Eigenvectors
- Matrix Calculus ⇐ **Let's continue here.**

Gradient of a matrix

- Called ***gradient*** or ***gradient matrix*** (also called ***score***)
- Let a function $\boldsymbol{f}: \boldsymbol{R}^{m\ x\ n} \rightarrow \boldsymbol{R}$ take as input a matrix ***A*** of size **m x n**.
- Then the gradient of ***A*** is the following:

$$\nabla_A f(A) \in \mathbb{R}^{m \times n} = \begin{bmatrix} \frac{\partial f(A)}{\partial A_{11}} & \frac{\partial f(A)}{\partial A_{12}} & \cdots & \frac{\partial f(A)}{\partial A_{1n}} \\ \frac{\partial f(A)}{\partial A_{21}} & \frac{\partial f(A)}{\partial A_{22}} & \cdots & \frac{\partial f(A)}{\partial A_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(A)}{\partial A_{m1}} & \frac{\partial f(A)}{\partial A_{m2}} & \cdots & \frac{\partial f(A)}{\partial A_{mn}} \end{bmatrix}$$

- Useful when you have many linear combinations the parameters of which you would like to *minimize* or *maximize* (example: neural networks)

Gradient of a matrix

- Every entry in the matrix can be written as:

$$(\nabla_A f(A))_{ij} = \frac{\partial f(A)}{\partial A_{ij}}$$

- The size of the gradient is always equal to the size of the matrix
  - Reminder: Scalars and vectors are also matrices. So:
  - A scalar has a gradient of size ***1 x 1***
  - A vector has a gradient of size ***1 x n***
  - A matrix has a gradient of size ***m x n***

Question: Gradient

- Example: For $x \in R^n$, let $f(x) = b^T x$ for some known vector $b \in R^n$

$$f(x) = \begin{bmatrix} b_1 & b_2 & \dots & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- Question: What is the **gradient** (or the **Jacobian**) here?
- Or: What is $\nabla_x f(x)$

Question: Gradient

- Example: For $x \in R^n$, let $f(x) = b^T x$ for some known vector $b \in R^n$

$$f(x) = \sum_{i=1}^{n} b_i x_i$$

$$\frac{\partial f(x)}{\partial x_k} = \frac{\partial}{\partial x_k} \sum_{i=1}^{n} b_i x_i = b_k$$

- So, we can conclude that the **gradient** is the vector ***b***.

Jacobian Matrix

- You may also have heard about the ***Jacobian matrix***
  - Similar to ***gradient***

- If you have a vector-valued function y = f(x) where $x, y \in R^n$, then the gradient of ***y*** with respect to ***x*** is a **Jacobian matrix**

$$J = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{pmatrix}$$

Example: Jacobian

- Let's say we have <u>multiple functions</u> with <u>multiple parameters</u>:

$$f(x,y) = 3x^2 * y$$

$$g(x,y) = 2x + y^8$$

$$J = \begin{bmatrix} \nabla f(x,y) \\ \nabla g(x,y) \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} & \frac{\partial f(x,y)}{\partial y} \\ \frac{\partial g(x,y)}{\partial x} & \frac{\partial g(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} 6yx & 3x^2 \\ 2 & 8y^7 \end{bmatrix}$$

This is called the denominator layout of Jacobian

Gradient vs. Jacobian

- The ***gradient*** is the vector formed by the partial derivatives of a scalar function

- The ***Jacobian matrix*** is the matrix formed by the partial derivatives of a vector function

- More formally:
  - ***Gradient***: $$\nabla f(x, y) = \begin{pmatrix} f'_x \\ f'_y \end{pmatrix}$$

  - ***Jacobian matrix***: $$J\left(f(x, y), g(x, y) \right) = \begin{pmatrix} f'_x & g'_x \\ f'_y & g'_y \end{pmatrix} = \left(\nabla f; \nabla g \right)$$

  - Question: What is the similarity/difference?

Hessian Matrix

- Or: ***The Hessian***
  - **The gradient of every entry of the gradient of a vector function**
- ***The Hessian matrix*** (or ***H***) with respect to **x** is the *n x n* matrix of partial derivatives

$$\nabla_x^2 f(x) \in \mathbb{R}^{n\times n} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

What is this matrix similar to?

- <u>Question</u>: Why is ***the Hessian*** always symmetric?
- <u>Answer</u>: The order of partial derivatives doesn't matter (=Young's theorem)

The Hessian

- The first column is the gradient of $\dfrac{\partial f(x)}{\partial x_1}$

$$\nabla_x^2 f(x) \in \mathbb{R}^{n\times n} = \begin{bmatrix} \boxed{\begin{matrix} \frac{\partial^2 f(x)}{\partial x_1^2} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} \\ \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} \end{matrix}} & \begin{matrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f(x)}{\partial x_2^2} \\ \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_2} \end{matrix} & \begin{matrix} \cdots \\ \cdots \\ \ddots \\ \cdots \end{matrix} & \begin{matrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots \\ \frac{\partial^2 f(x)}{\partial x_n^2} \end{matrix} \end{bmatrix}$$

Gradient | Jacobian | Hessian

- ***Gradient***: Vector of the 1st partial derivatives of a scalar function
  - You use gradient to find the minimum cost

- ***Jacobian***: Matrix of the 1st partial derivatives (gradients) of a vector function
  - You use Jacobian to see how your cost function behaves in one of the n-dimensions

- ***Hessian***: Matrix of the 2nd partial derivatives (gradients) of a scalar function
  - You use Hessian to adjust your gradient (and minimize your cost usually more rapidly)

- <u>Quick</u> clarification:
  - Scalar function = Function in a scalar field
  - Vector function = Function in a vector field

UNIVERSITY of ROCHESTER

Common vector derivatives

Scalar derivative			Vector derivative		
$f(x)$	$\rightarrow$	$\frac{\mathrm{d}f}{\mathrm{d}x}$	$f(\mathbf{x})$	$\rightarrow$	$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}}$
bx	$\rightarrow$	b	$\mathbf{x}^T\mathbf{B}$	$\rightarrow$	$\mathbf{B}$
bx	$\rightarrow$	b	$\mathbf{x}^T\mathbf{b}$	$\rightarrow$	$\mathbf{b}$
x^2	$\rightarrow$	$2x$	$\mathbf{x}^T\mathbf{x}$	$\rightarrow$	$2\mathbf{x}$
bx^2	$\rightarrow$	$2bx$	$\mathbf{x}^T\mathbf{B}\mathbf{x}$	$\rightarrow$	$2\mathbf{B}\mathbf{x}$

Please do the following until next lecture!

- Review what we have just gone through (matrix algebra and probability)

- Continue with your problem set (due date ***Friday, January 28, 11:59 PM***)

- Read ***Chapter 4*** from our book

- <u>And reminder:</u>
    - If you are planning to drop, please do so at your earliest opportunity