Linear Algebra

IST 718 – Advanced Information Analytics

Adjunct Professor Willard Williamson

linkedin.com/in/ncc-1701-d

Attendance

- Please fill out the class attendance sheet
- Place a “1” (which stands for present) in the row next to your name for today’s date in the attendance sheet.
- Attendance sheet is here: https://docs.google.com/spreadsheets/d/1G5tiqXQn9mwyubWdINtPI_QONXH1_2U8oYikksgWlss/edit?usp=sharing

Objectives

- Will take an “application oriented” approach to linear algebra in this class as opposed to a theoretical approach
- Not a comprehensive survey of linear algebra
- Focused on linear algebra subset most relevant to machine learning

Linear Algebra Review Material

- Course Text Book: Deep Learning by Ian Goodfellow, Yoshua Bengio & Aaron Couville, MIT Press, 2017:
  - http://www.deeplearningbook.org/contents/linear_algebra.html
  - https://www.deeplearningbook.org/slides/02_linear_algebra.pdf
- A good review from Upenn:
  - https://www.ling.upenn.edu/courses/cogs501/LinearAlgebraReview.html

Linear Algebra Deep Dive

- Introduction to Linear Algebra, Gilbert Strang:
  - http://math.mit.edu/~gs/linearalgebra/
  - Dr. Gilbert Strang is a world famous MIT math professor
  - Note that Dr. Strang has free linear algebra courses on YouTube.

Introduction

- Quote by Gilbert Strang: Linear Algebra has become as basic and applicable as calculus, and fortunately it is easier.
- Linear algebra appears in virtually every branch of applied mathematics, physics, mathematical economics, etc.
- Linear algebra is very important to
  - Big data analytics
  - Machine learning – especially deep learning
  - Inferential and exploratory statistics
  - Many other fields in science

Scalars

- Represented by Greek letters α, β, γ
- Represents integers, reals, rationals, etc.
- Scalars are quantities that are fully described by a magnitude alone.
- Examples:
  - $\alpha = 0.1$
  - $\beta = 1^{-10}$
  - $\gamma = 3$

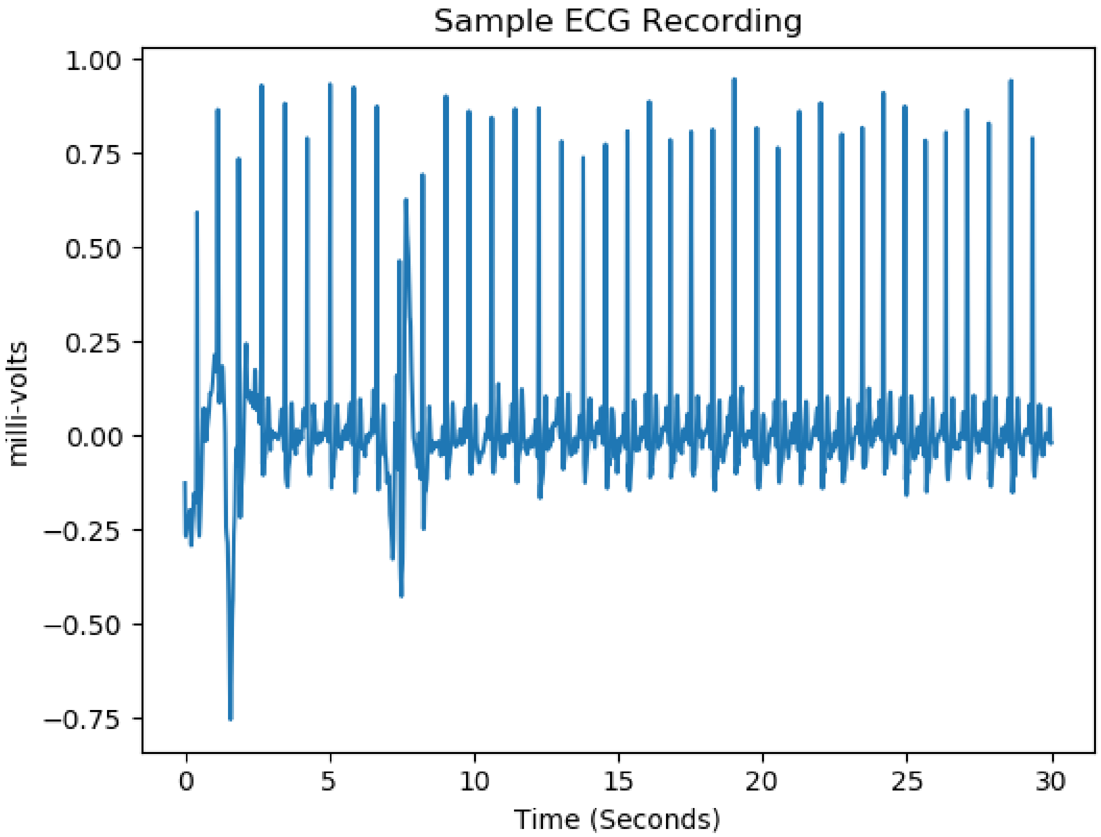
Vectors

- Represented by lower case letters
- A vector is a 1-D array of scalars:
- $a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \cdots \\ a_1 \end{bmatrix}$
- A vector is also a single column matrix

Vectors Continued

- In physics, a vector describes magnitude and direction
- Example notation for the type and size of a vector:
  - A real vector of size (length) n: $\mathbb{R}^n$ *where:*
    - $\mathbb{R}$ represents the real numbers
    - *n represents the length of the vector*
- In data science, a vector is often just a column of numbers that describes an event.
  - Example: A vector could describe the voltage signal of a heartbeat EKG reading (see next slide)

EKG Data Vector

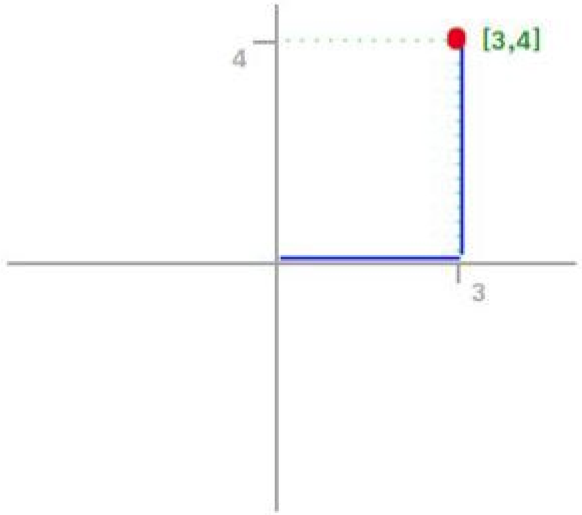


Norms

- A norm is a function that measures the length of a vector.
- Often referred to as L^p Norm where p is the order of the norm function
- $\| x \|_p = (\sum_i |x_i|^p)^{1/p}$

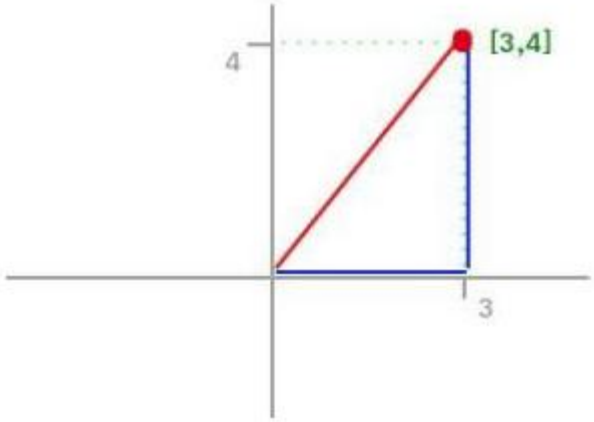
L1 Norm

- L^1 Norm is sometimes called the taxicab or Manhattan norm because it represents the total distance traveled between the start and end of the vector.
- One of the most common Norms
- $\| x \|_1 = |3| + |4| = 7$
- Grows at the same rate in all locations
- Does a good job of discriminating between elements that are exactly zero and elements that are small but non-zero
- Frequently used in machine learning regularization



L2 Norm

- L^2 Norm calculates the shortest distance between the start and end of the vector
- Also known as the "Euclidian" Norm
- $\| x \|_2 = \sqrt{3^2 + 4^2} = 5$
- Frequently used in machine learning regularization



Vector Dot Product

- The dot product of 2 vectors yields a scalar
- Given:
  - $a = [a_1 \;\; a_2 \;\; a_3]$
  - $b = [b_1 \;\; b_2 \;\; b_3]$
- $a * b = \sum_i a_i b_i$
- $[a_1 \;\; a_2 \;\; a_3] * [b_1 \;\; b_2 \;\; b_3] = \; [a_1 b_1 + a_2 b_2 + a_3 b_3]$

Vector Cross Product

- The cross product of 2 vectors yields a matrix
- $u \otimes v = uv^t$
- $\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} [v_1 \ v_2 \ v_3] = \begin{bmatrix} u_1v_1 & u_1v_2 & u_1v_3 \\ u_2v_1 & u_2v_2 & u_2v_3 \\ u_3v_1 & u_3v_2 & u_3v_3 \\ u_4v_1 & u_4v_2 & u_4v_3 \end{bmatrix}$

Matrices

- A 2 dimensional array of scalars
- Represented by capital letters like A, B, C, etc.
- $A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$
- The above matrix has m rows and n columns
- The matrix dimension is denoted $A_{m \times n}$ where m = number of rows and n = number of columns
- Can provide information about how vector changes over time

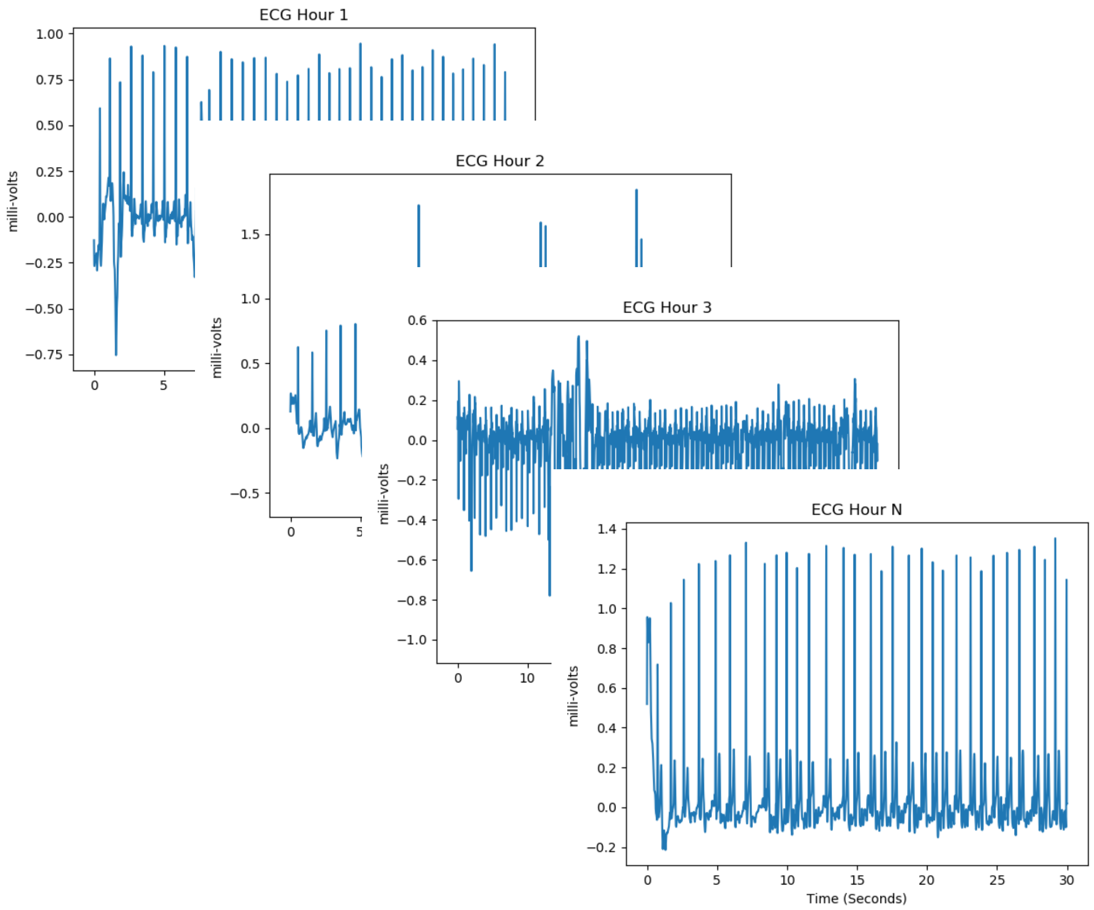
EKG Matrix: Each Col Is A Recording

unprocessed_data - NumPy array

	0	1	2	3	4
0	-0.127	-0.162	-0.197	-0.229	-0.245
1	0.128	0.157	0.189	0.226	0.25
2	0.056	0.073	0.085	0.093	0.1
3	0.519	0.619	0.723	0.827	0.914
4	-0.188	-0.239	-0.274	-0.316	-0.356
5	-0.266	-0.316	-0.367	-0.407	-0.423
6	0.021	0.022	0.024	0.026	0.028
7	-0.187	-0.236	-0.286	-0.34	-0.375
8	0.051	0.056	0.059	0.063	0.065
9	-1.028	-1.225	-1.418	-1.599	-1.747
10	-0.069	-0.089	-0.108	-0.125	-0.135
11	0.072	0.088	0.103	0.112	0.118

Format Resize ☑ Background color

Save and Close Close



Matrix Operations

- Addition / subtraction
- Scalar multiplication
- Matrix dot product
- Transpose

Matrix Addition / Subtraction

- Matrices must be the same size to add or subtract
- Matrix addition is commutative: $\mathrm{A} + \mathrm{B} = \mathrm{B} + \mathrm{A}$
- Add or subtract each matrix element between operands
- Resulting matrix has the same size as the 2 matrix operands
- $\mathrm{A} + \mathrm{B} = \mathrm{C}$
- $\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$

Matrix Multiplication by Scalar

- To multiply a matrix by a scalar, “broadcast” the scalar across all of the matrix elements and multiply:
- A * 2 = C
- $\begin{bmatrix} a & b \\ c & d \end{bmatrix} * 2 = \begin{bmatrix} 2a & 2b \\ 2c & 2d \end{bmatrix}$

Matrix (Dot) Product

- Multiply each row element of matrix A by the corresponding column element in matrix B and sum the result
- Matrix multiplication is not commutative: AB != BA
- $A_{i,k} * B_{k,j} = C_{i,j}$
  - The number of columns in A must match the number of rows in B (see the ‘k’ index in the equation above)
  - The size of the resulting matrix C is equal to the number of rows in A and the number of columns in B (see the green i, j indices in the above equation).
- $\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} * \begin{bmatrix} q & r & s & t \\ u & v & w & x \end{bmatrix} = \begin{bmatrix} (a*q+b*u) & (a*r+b*v) & (a*s+b*w) & (a*t+b*x) \\ (c*q+d*u) & (c*r+d*v) & (c*s+d*w) & (c*t+d*x) \\ (e*q+f*u) & (e*r+f*v) & (e*s+f*w) & (e*t+f*x) \end{bmatrix}$

Matrix Transpose

- Swap rows and columns
- $(A^T)_{i,j} = A_{j,i}$
- $A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ A_{3,1} & A_{3,2} \end{bmatrix}$
- $A^T = \begin{bmatrix} A_{1,1} & A_{2,1} & A_{3,1} \\ A_{1,2} & A_{2,2} & A_{3,2} \end{bmatrix}$
- Property: $(AB)^T = B^T A^T$

Tensors

- A tensor is an array of numbers, that may have
  - zero dimensions, and be a scalar
  - one dimension, and be a vector
  - two dimensions, and be a matrix
  - or more dimensions.

Difference Between Tensor and Matrix

- A matrix is a 2-dimensional grid of numbers
- A tensor is a generalized matrix
- A tensor includes all of the following
  - 0 dimension (AKA scalar)
  - 1 dimension (AKA vector)
  - 2 dimension (AKA matrix)
  - Or higher dimension
- A tensor is more general and flexible than a matrix
- The dimension of the tensor is known as it's "rank". Example, a 3 dimensional numpy array (array[][][]) is a tensor of rank 3.

Identity Matrix

- Multiplying a matrix by the identity matrix produces the original matrix.
- A*I = A where ‘I’ is the identity matrix
- Identity matrix has ones on it’s diagonal and zeros everywhere else
- Example: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Linear Combinations

- A linear combination is an expression constructed from a set of terms by multiplying each term by a constant and adding the results.
- Example: The linear combination of the terms x and y would be constructed by multiplying each of the terms by a constant and adding.
  - $ax + by$ where a and b are constants

Linear Combinations

- A linear combination can be performed on the columns or rows of a matrix.
- Linear combinations of matrix rows are very important to principal component analysis (covered later in the course).
- Example matrix column linear combination for $\begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \\ u_{31} & u_{32} \\ u_{41} & u_{42} \end{bmatrix}$
- $c_1 \begin{bmatrix} u_{11} \\ u_{21} \\ u_{31} \\ u_{41} \end{bmatrix} + c_2 \begin{bmatrix} u_{12} \\ u_{22} \\ u_{32} \\ u_{42} \end{bmatrix}$