Machine Learning Foundations
Intro to Linear Algebra

An Interactive Primer on
the Theory and Practice of
Tensor Manipulation in Python

Jon Krohn, Ph.D.

jonkrohn.com/talks

github.com/jonkrohn/ML-foundations

Machine Learning Foundations
Intro to Linear Algebra

Slides: jonkrohn.com/talks

Code: github.com/jonkrohn/ML-foundations

Stay in Touch:

jonkrohn.com to sign up for email newsletter

linkedin.com/in/jonkrohn

youtube.com/c/JonKrohnLearns

twitter.com/JonKrohnLearns

The Pomodoro Technique

Rounds of:

- 25 minutes of work
- with 5 minute breaks

Questions best handled at breaks, so save questions until then.

When people ask questions that have already been answered, do me a favor and let them know, politely providing response if appropriate.

Except during breaks, I recommend attending to this lecture only as topics are not discrete: Later material builds on earlier material.

POLL

Where are you?

- The Americas
- Europe / Middle East / Africa
- Asia-Pacific
- Extra-Terrestrial Space

POLL

What are you?

- Developer / Engineer
- Scientist / Analyst / Statistician / Mathematician
- Combination of the Above
- Other

POLL

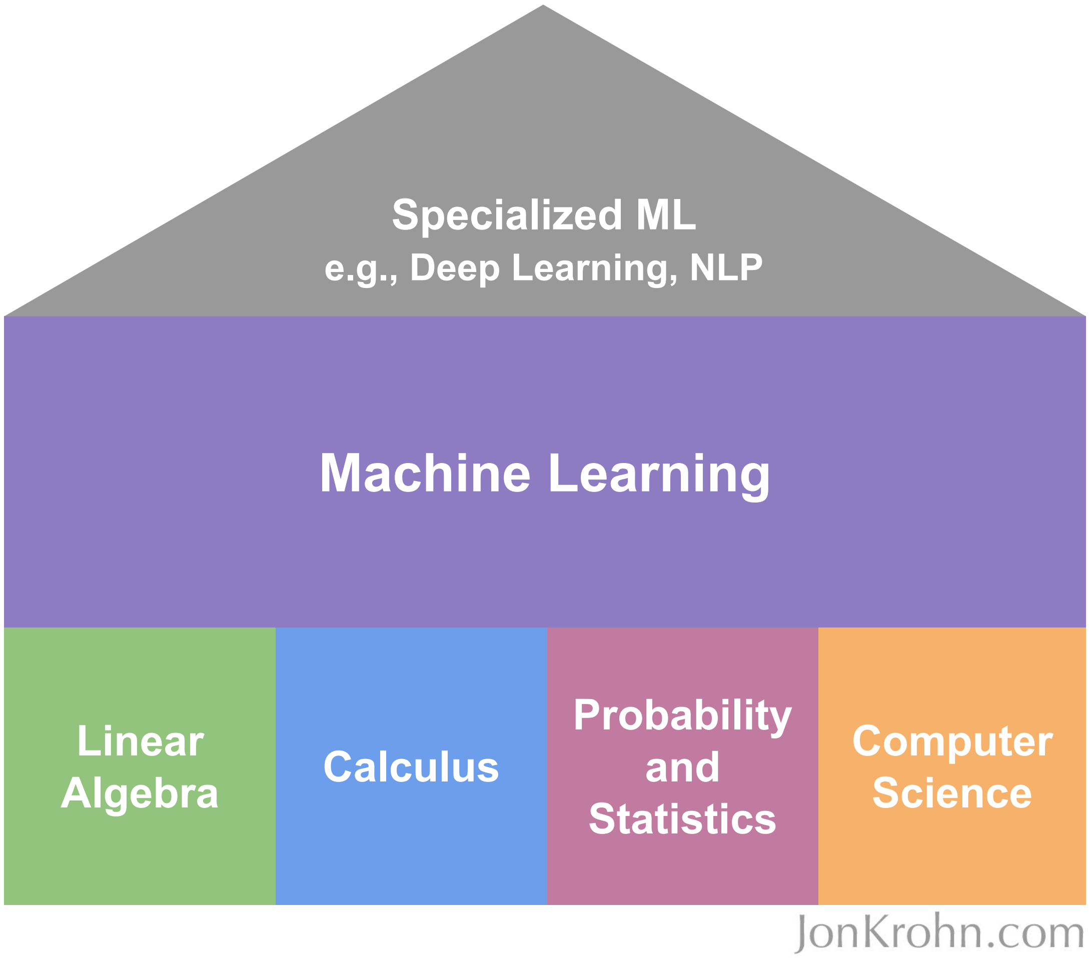
What is your level of familiarity with Linear Algebra?

- Little to no exposure
- Some understanding of the theory
- Deep understanding of the theory
- Deep understanding of the theory and experience applying linear algebra operations with code

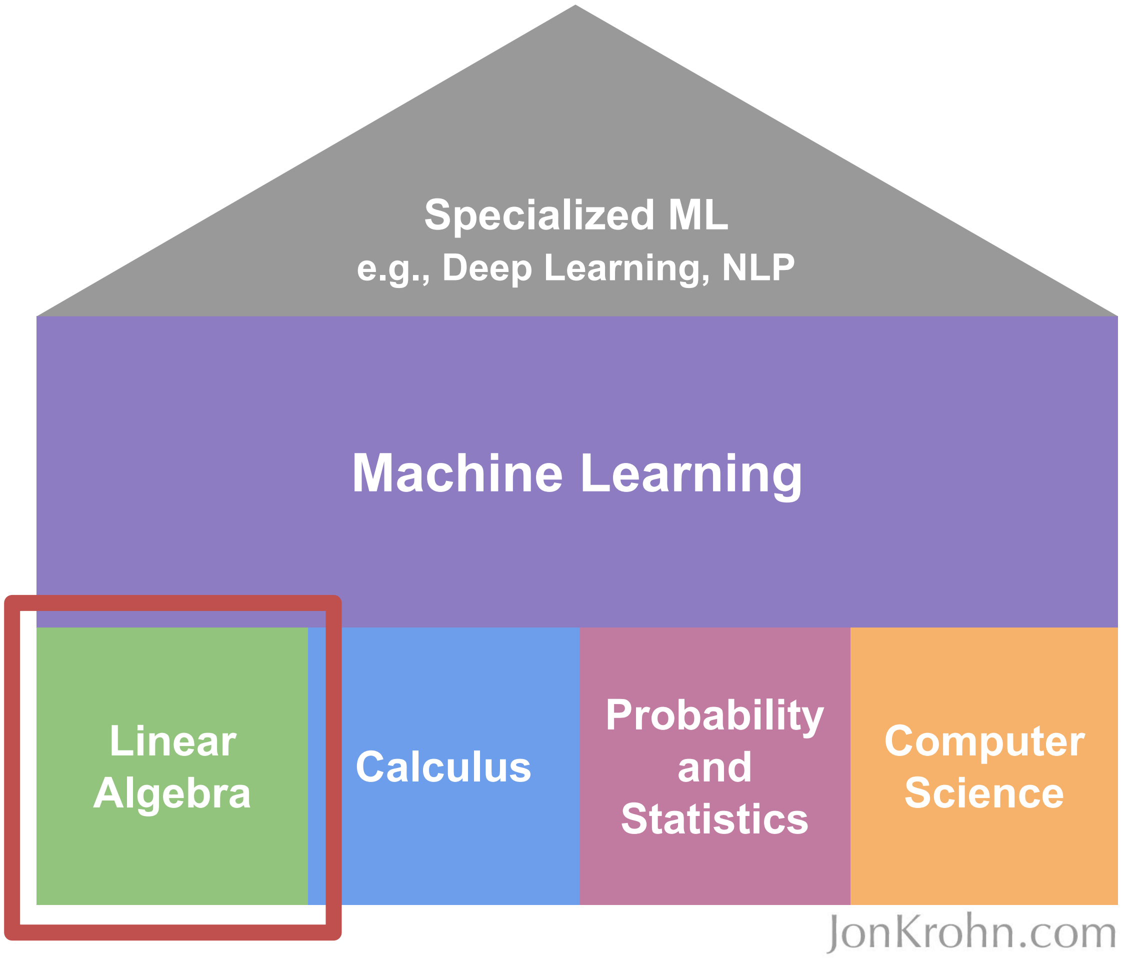
POLL

What is your level of familiarity with Machine Learning?

- Little to no exposure, or exposure to theory only
- Experience applying machine learning with code
- Experience applying machine learning with code and some understanding of the underlying theory
- Experience applying machine learning with code and strong understanding of the underlying theory



Specialized ML
e.g., Deep Learning, NLP
Machine Learning
Linear Algebra
Calculus
Probability and Statistics
Computer Science
JonKrohn.com


Specialized ML
e.g., Deep Learning, NLP
Machine Learning
Linear Algebra
Calculus
Probability and Statistics
Computer Science
JonKrohn.com

ML Foundations Series

Intro to Linear Algebra is **foundational for**:

1. **Intro to Linear Algebra**
2. **Linear Algebra II: Matrix Operations**
3. Calculus I: Limits & Derivatives
4. Calculus II: Partial Derivatives & Integrals
5. Probability & Information Theory
6. Intro to Statistics
7. Algorithms & Data Structures
8. **Optimization**

Intro to Linear Algebra

1. Data Structures for Algebra
2. Common Tensor Operations
3. Matrix Properties

Intro to Linear Algebra

1. **Data Structures for Algebra**
2. Common Tensor Operations
3. Matrix Properties

Segment 1: Data Structures for Algebra

- What Linear Algebra Is
- A Brief History of Algebra
- Tensors
- Scalars
- Vectors and Vector Transposition
- Norms and Unit Vectors
- Basis, Orthogonal, and Orthonormal Vectors
- Arrays in NumPy
- Matrices
- Tensors in TensorFlow and PyTorch

What Algebra Is

Algebra is arithmetic that includes non-numerical entities like x:

$$
\begin{aligned}
2x + 5 &= 25 \\
2x + 5 - 5 &= 25 - 5 \\
2x &= 20 \\
2x/2 &= 20/2 \\
x &= 10
\end{aligned}
$$

We have determined x **must equal 10** because $2(10) + 5 = 25$

What Linear Algebra Is

If it has an exponential term, it *isn't* linear algebra, e.g.:

$$2x^2 + 5 = 25$$

$$2\sqrt{x} + 5 = 25$$

What Linear Algebra Is

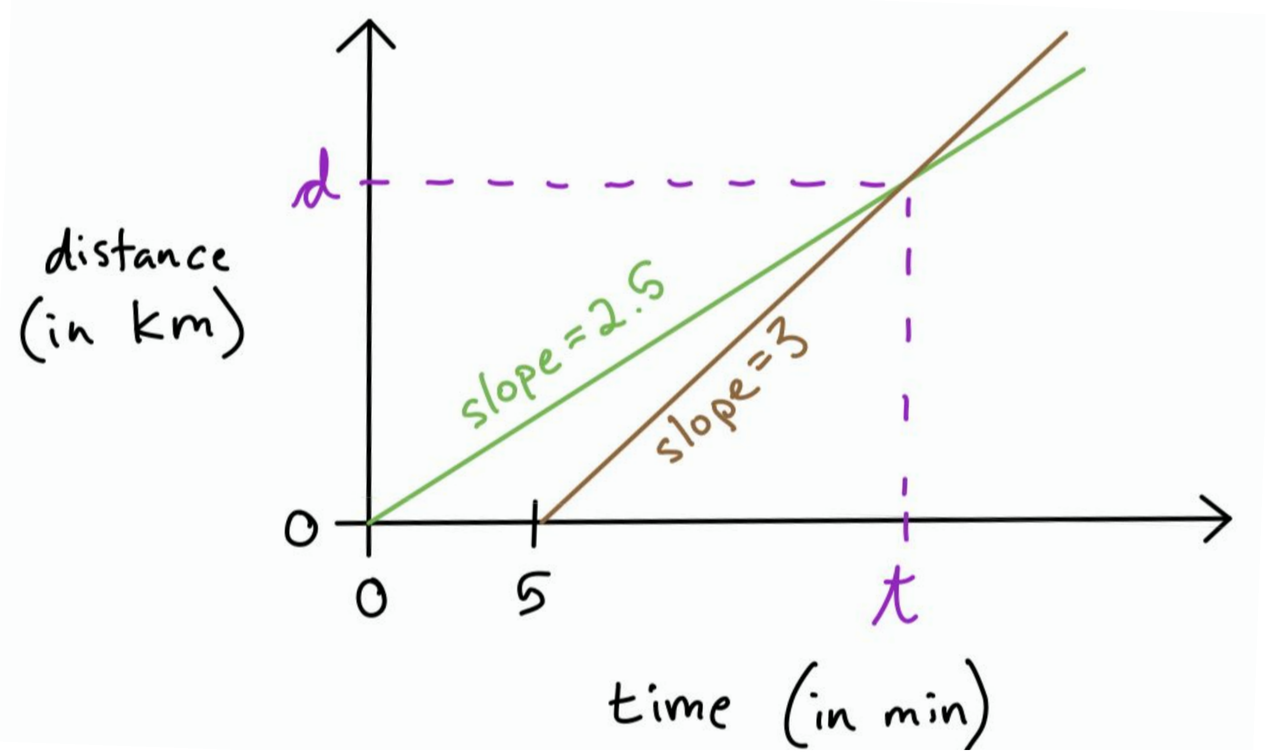
“Solving for unknowns within system of linear equations”

Consider the following example:

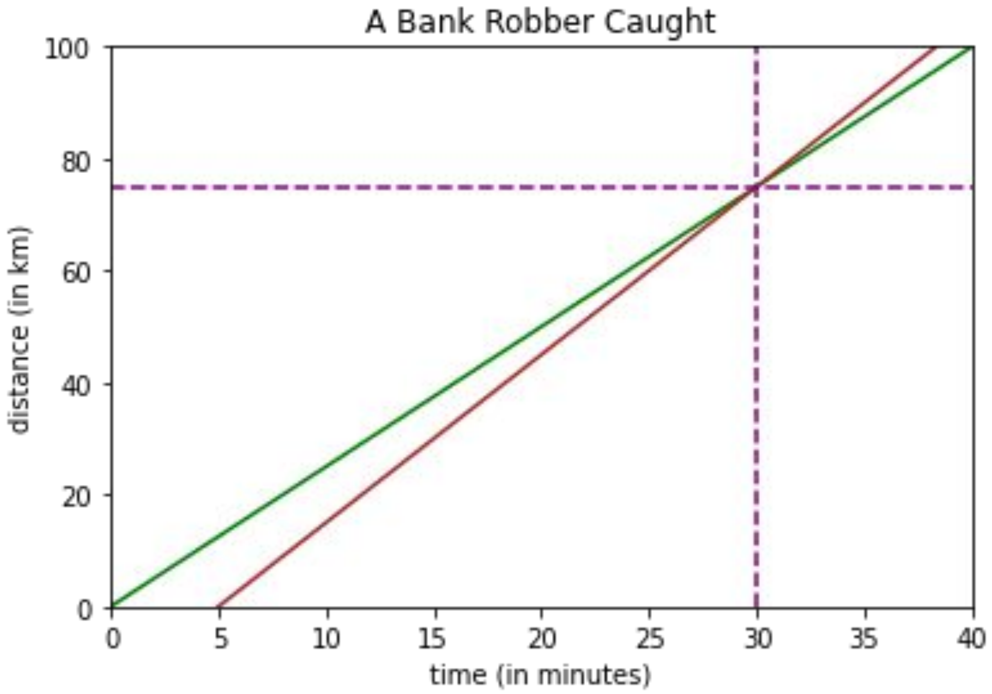
- Sheriff has 180 km/h car
- Bank robber has 150 km/h car and five-minute head start
- How long does it take the sheriff to catch the robber?
- What distance will they have traveled at that point?
- (For simplicity, let’s ignore acceleration, traffic, etc.)

What Linear Algebra Is

Problem could be solved graphically with a plot:



Note that: *150 km/h = 2.5 km/min* *180 km/h = 3 km/min*



Hands-on code demo: `1-intro-to-linear-algebra.ipynb`

What Linear Algebra Is

Alternatively, problem can be solved *algebraically*:

Equation 1: $d = 2.5t$

Equation 2: $d = 3(t - 5)$

$2.5t = 3(t - 5)$

$2.5t = 3t - 15$

$2.5t - 3t = -15$

$-0.5t = -15$

$t = -15/-0.5 = 30$ min

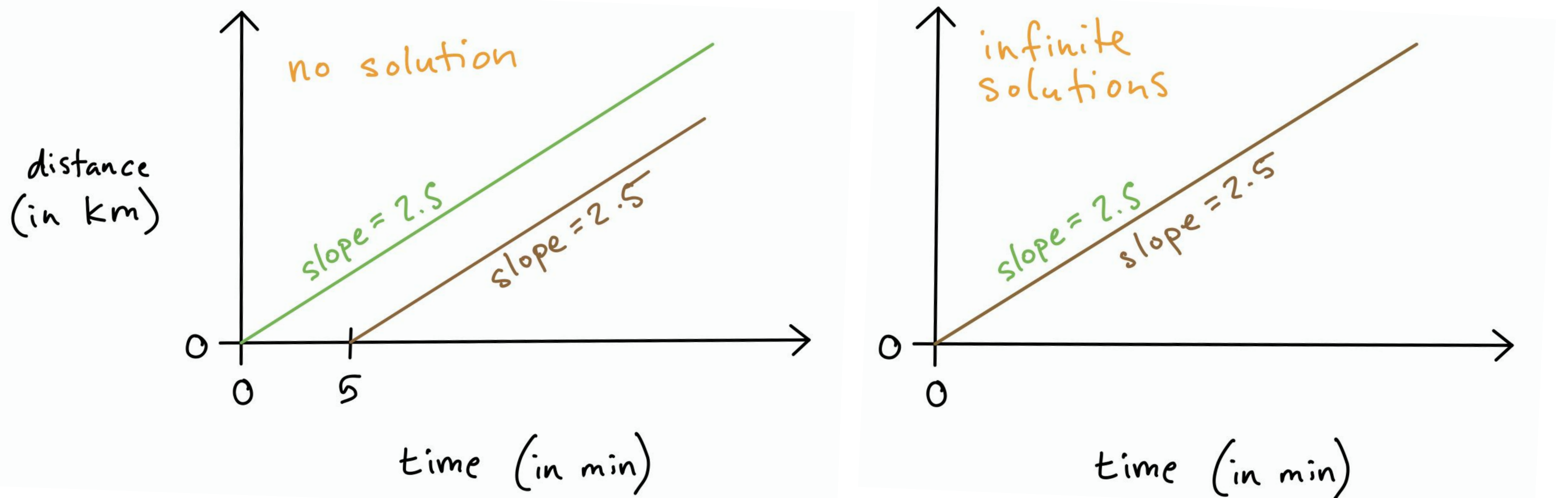
$d = 2.5t = 2.5(30) = 75$ km

$d = 3(t - 5) = 3(30 - 5) = 3(25) = 75$ km

What Linear Algebra Is

No solution if sheriff's car is same speed as bank robber's.

Infinite solutions if same speed *and* same starting time.



These are the only three options in linear algebra: one, no, or infinite solutions.

It is impossible for lines to cross multiple times.

What Linear Algebra Is

In a given system of equations:

- Could be *many* equations
- Could be *many* unknowns in each equation

house price

distance to school

there could be m features (many!)

$$y = a + b x_1 + c x_2 + \dots + m x_m$$

"y-intercept"

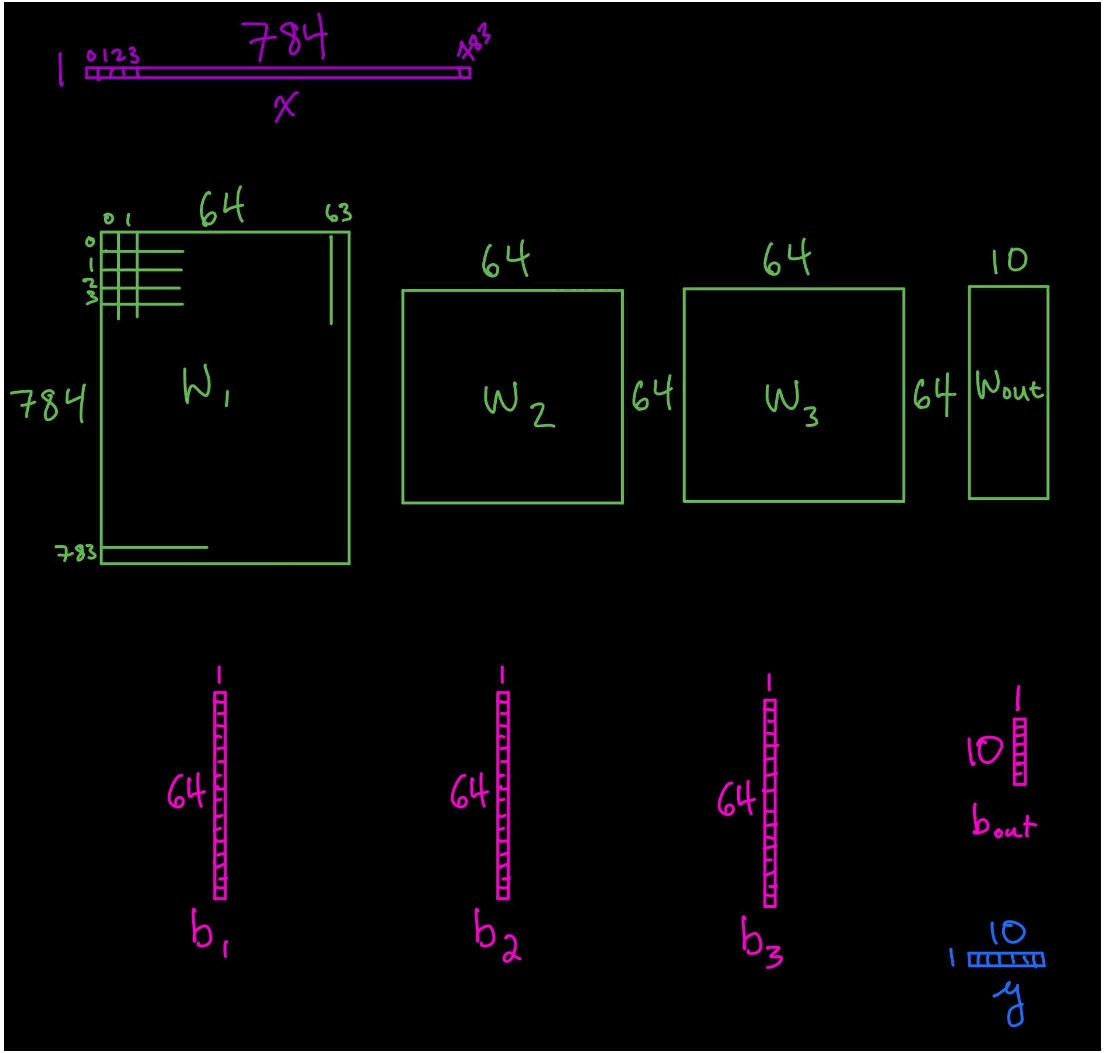
number of bedrooms

$$y = a + b x_1 + c x_2 + \dots + m x_m$$

$$\left[\begin{array}{c|ccccccccc}
y_1 & a & + & b x_{1,1} & + & c x_{1,2} & + & \dots & + & m x_{1,m} \\
y_2 & a & + & b x_{2,1} & + & c x_{2,2} & + & \dots & + & m x_{2,m} \\
\vdots & \vdots & & \vdots & & \vdots & & & & \vdots \\
y_n & a & + & b x_{n,1} & + & c x_{n,2} & + & \dots & + & m x_{n,m}
\end{array}\right]$$

For any house i in the dataset, y_i = price and $x_{i,1}$ to $x_{i,m}$ are its features.

We solve for parameters a, b, c to m



See:

jonkrohn.com/deepTF1

jonkrohn.com/convTF1

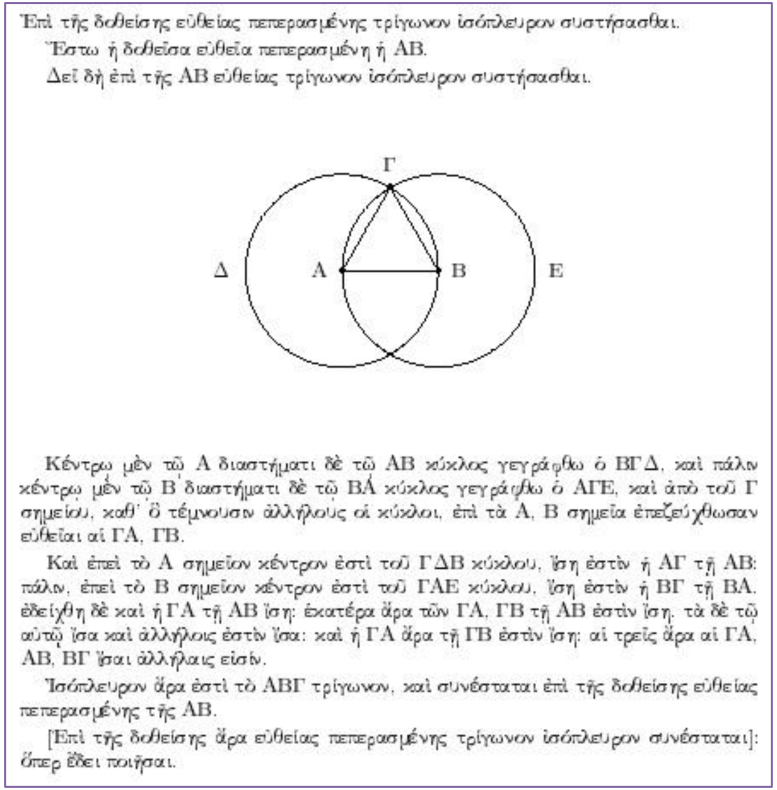
A Brief History of Algebra

Al-Khwārizmī'
(c. 780 - c. 850)

wrote *The Compendious Book on Calculation by Completion* (Arabic: “al-jabr”) *and Balancing*

A Brief History of Algebra

- 1900 BCE: Babylonian "rhetorical"
- 1650 BCE: Egyptians
- 6th century BCE: Indians
- 400-300 BCE: Greeks
- 250 BCE: Chinese
- Europeans much later:
  - 12th century: Arabic to Latin trans.
  - 13th century: rivaled others



Ἐπὶ τῆς δοθείσης εὐθείας πεπερασμένης τρίγωνον ἰσόπλευρον συστήσασθαι.

Ἔστω ἡ δοθεῖσα εὐθεῖα πεπερασμένη ἡ ΑΒ.

Δεῖ δὴ ἐπὶ τῆς ΑΒ εὐθείας τρίγωνον ἰσόπλευρον συστήσασθαι.

Κέντρῳ μὲν τῷ Α διαστήματι δὲ τῷ ΑΒ κύκλος γεγράφθω ὁ ΒΓΔ, καὶ πάλιν κέντρῳ μὲν τῷ Β διαστήματι δὲ τῷ ΒΑ κύκλος γεγράφθω ὁ ΑΓΕ, καὶ ἀπὸ τοῦ Γ σημείου, καθ' ὃ τέμνουσιν ἀλλήλους οἱ κύκλοι, ἐπὶ τὰ Α, Β σημεῖα ἐπεζεύχθωσαν εὐθεῖαι αἱ ΓΑ, ΓΒ.

Καὶ ἐπεὶ τὸ Α σημεῖον κέντρον ἐστὶ τοῦ ΓΔΒ κύκλου, ἴση ἐστὶν ἡ ΑΓ τῇ ΑΒ· πάλιν, ἐπεὶ τὸ Β σημεῖον κέντρον ἐστὶ τοῦ ΓΑΕ κύκλου, ἴση ἐστὶν ἡ ΒΓ τῇ ΒΑ. ἐδείχθη δὲ καὶ ἡ ΓΑ τῇ ΑΒ ἴση· ἑκατέρα ἄρα τῶν ΓΑ, ΓΒ τῇ ΑΒ ἐστιν ἴση. τὰ δὲ τῷ αὐτῷ ἴσα καὶ ἀλλήλοις ἐστὶν ἴσα· καὶ ἡ ΓΑ ἄρα τῇ ΓΒ ἐστιν ἴση· αἱ τρεῖς ἄρα αἱ ΓΑ, ΑΒ, ΒΓ ἴσαι ἀλλήλαις εἰσίν.

Ἰσόπλευρον ἄρα ἐστὶ τὸ ΑΒΓ τρίγωνον, καὶ συνέσταται ἐπὶ τῆς δοθείσης εὐθείας πεπερασμένης τῆς ΑΒ.

[Ἐπὶ τῆς δοθείσης ἄρα εὐθείας πεπερασμένης τρίγωνον ἰσόπλευρον συνέσταται]· ὅπερ ἔδει ποιῆσαι.

A Brief History of Algebra

Contemporary applications:

- Solving for unknowns in ML algos, including deep learning
- Reducing dimensionality (e.g., principal component analysis)
- Ranking results (e.g., with eigenvector)
- Recommenders (e.g., singular value decomposition, SVD)
- Natural language processing (e.g., SVD, matrix factorization)
  - Topic modeling
  - Semantic analysis

Exercise

Jill designs solar panels as a hobby.

On April 1st, Jill's "Mark I" design begins generating power: 1 kJ/day.

On May 1st, her "Mark II" design begins generating 4 kJ of power per day.

1. What day is it when Jill's Mark II design has generated as much total energy as the Mark I design?
2. How much total energy have both generated by that day?
3. What would the solutions to (1.) and (2). be if Mark II design generated 1kJ of power per day?

Solutions

1. 40 days from April 1st, which is May 10th
2. 40 kJ generated by each design for a total of 80 kJ
3. No solutions.

1

$$e = 1t$$

$$e = 4(t - 30)$$

$$4(t - 30) = 1t$$

$$4t - 120 = t$$

$$4t - t = 120$$

$$3t = 120$$

$$t = 40 \quad = \text{May 10th}$$

(2)

$$e = 1t = 1(40) = 40 \text{ kJ}$$

$$e = 4(t-30) = 4t - 120$$

$$= 4(40) - 120$$

$$= 160 - 120$$

$$= 40 \text{ kJ}$$

$\therefore$ 80 kJ total

Tensors

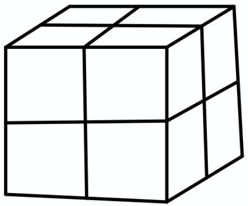
"ML generalization of vectors and matrices to any number of dimensions"

scalar x

vector $\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$

matrix $\begin{bmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \end{bmatrix}$

3-tensor



Dimensions	Mathematical Name	Description
0	scalar	magnitude only
1	vector	array
2	matrix	flat table, e.g., square
3	3-tensor	3D table, e.g., cube
n	*n*-tensor	higher dimensional

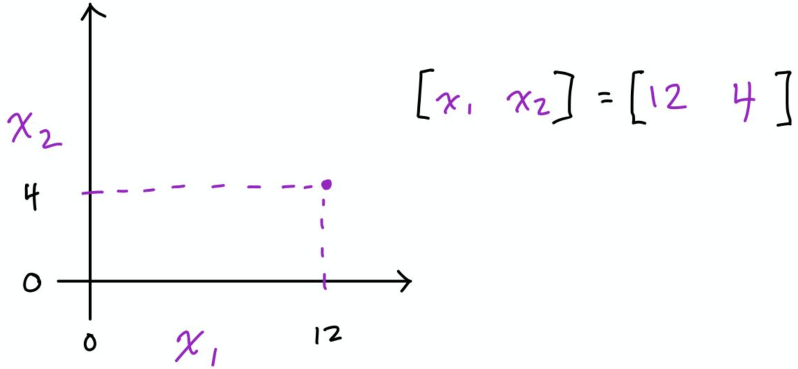
Scalars

- No dimensions
- Single number
- Denoted in lowercase, italics, e.g.: x
- Should be *typed*, like all other tensors: e.g., int, float32

Hands-on code demo: `1-intro-to-linear-algebra.ipynb`

Vectors

- One-dimensional array of numbers
- Denoted in lowercase, italics, bold, e.g.: $\boldsymbol{x}$
- Arranged in an order, so element can be accessed by its index
  - Elements are scalars so *not* bold, e.g., second element of $\boldsymbol{x}$ is x_2
- Representing a point in space:
  - Vector of length two represents location in 2D matrix (shown)
  - Length of three represents location in 3D cube
  - Length of *n* represents location in *n*-dimensional space



Vector Transposition

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

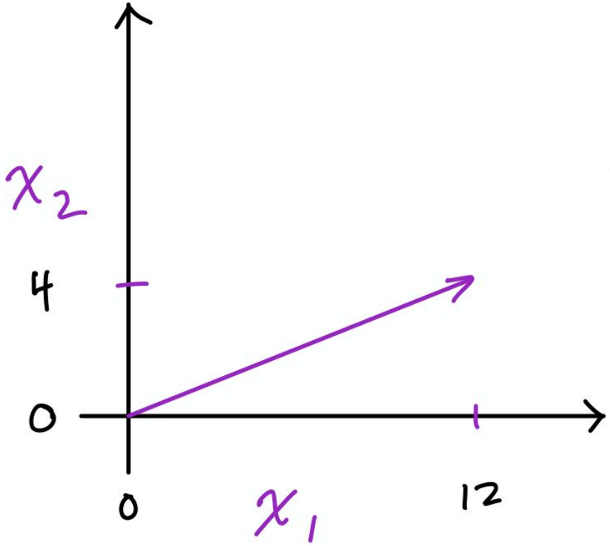
row vector — column vector

shape is (1, 3) — (3, 1)

Hands-on code demo

Norms

Vectors represent a magnitude and direction from origin:



$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} 12 & 4 \end{bmatrix}$$

Norms are functions that quantify vector magnitude.

L^2 Norm

- Described by:

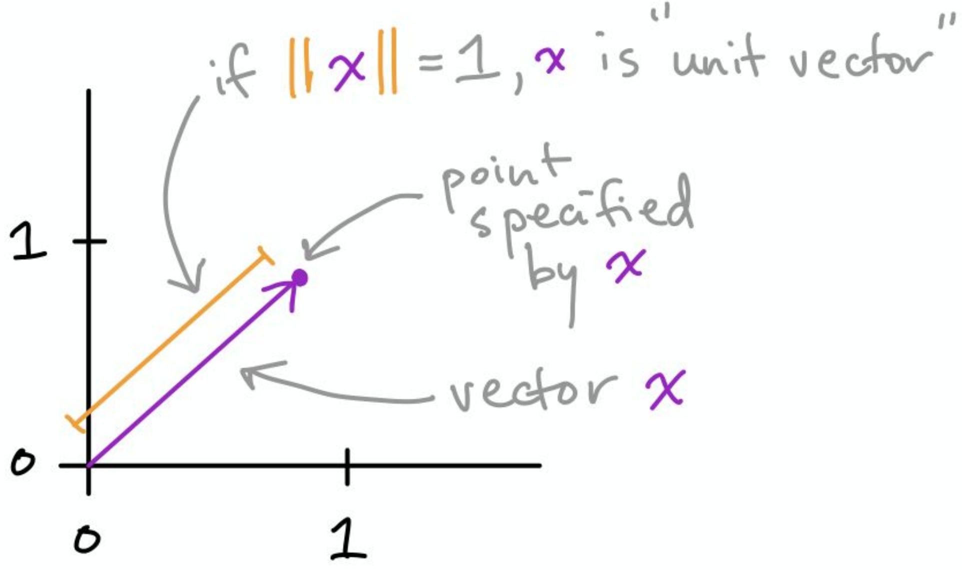
$$\|\boldsymbol{x}\|_2 = \sqrt{\sum_i x_i^2}$$

- Measures simple (Euclidean) distance from origin
- Most common norm in machine learning
  - Instead of $||\boldsymbol{x}||_2$, it can be denoted as $||\boldsymbol{x}||$

Hands-on code demo

Unit Vectors

- Special case of vector where its length is equal to one
- Technically, $\boldsymbol{x}$ is a unit vector with "unit norm", i.e.: $||\boldsymbol{x}|| = 1$



Beyond the L^2 Norm

- Over next few slides, we'll skim over other common norms in ML
- For our immediate purposes, only L^2 is important

L^1 Norm

- Described by:

$$\|x\|_1 = \sum_i |x_i|$$

- Another common norm in ML
- Varies linearly at all locations whether near or far from origin
- Used whenever difference between zero and non-zero is key

Hands-on code demo

Squared L^2 Norm

- Described by: $$||\boldsymbol{x}||_2^2 = \sum_i x_i^2$$
- Computationally cheaper to use than L^2 norm because:
    - Squared L^2 norm equals simply $\boldsymbol{x}^T\boldsymbol{x}$
    - Derivative (used to train many ML algorithms) of element x requires that element alone, whereas L^2 norm requires $\boldsymbol{x}$ vector
- Downside is it grows slowly near origin so can't be used if distinguishing between zero and near-zero is important

Hands-on code demo

Max Norm (or L^∞ Norm)

- Described by:

$$\|x\|_\infty = \max_i |x_i|$$

- Final norm we'll discuss; also occurs frequently in ML
- Returns the absolute value of the largest-magnitude element

Hands-on code demo

Generalized L^p Norm

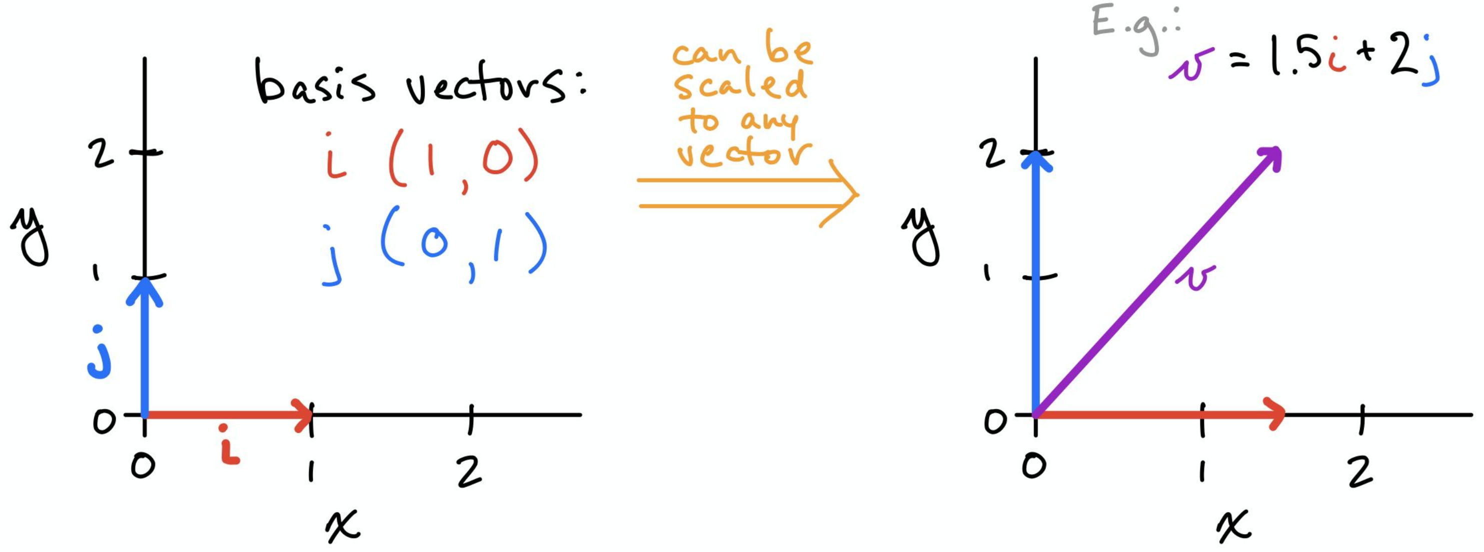
- Described by:

$$\|x\|_p = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}}$$

- p must be:
  - A real number
  - Greater than or equal to one
- Can derive L^1, L^2, and L^∞ norm formulae by substituting for p
- Norms, particularly L^1 and L^2, used to regularize objective functions

Basis Vectors

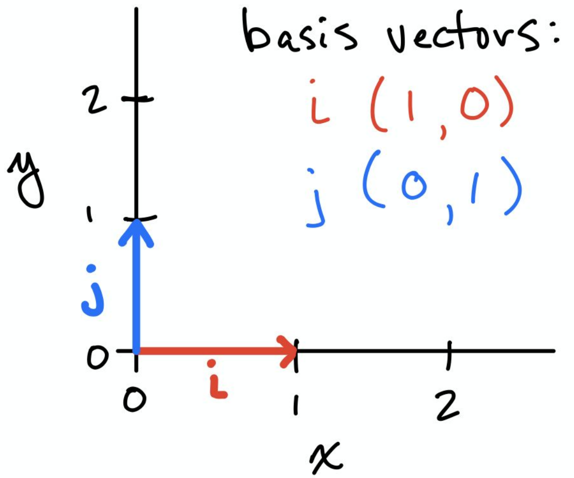
- Can be scaled to represent *any* vector in a given vector space
- Typically use unit vectors along axes of vector space (shown)



Orthogonal Vectors

- $\boldsymbol{x}$ and $\boldsymbol{y}$ are orthogonal vectors if $\boldsymbol{x}^T\boldsymbol{y} = 0$
- Are at 90° angle to each other (assuming non-zero norms)
- n-dimensional space has max n mutually orthogonal vectors (again, assuming non-zero norms)
- **Orthonormal** vectors are orthogonal *and* all have unit norm
  - Basis vectors are an example

Hands-on code demo



Matrices

- Two-dimensional array of numbers
- Denoted in uppercase, italics, bold, e.g.: $\boldsymbol{X}$
- Height given priority ahead of width in notation, i.e.: (n_{row}, n_{col})
  - If $\boldsymbol{X}$ has three rows and two columns, its shape is (3, 2)
- Individual scalar elements denoted in uppercase, italics only
  - Element in top-right corner of matrix $\boldsymbol{X}$ above would be $X_{1,2}$
- Colon represents an entire row or column:
  - Left column of matrix $\boldsymbol{X}$ is $\boldsymbol{X}_{:,1}$
  - Middle row of matrix $\boldsymbol{X}$ is $\boldsymbol{X}_{2,:}$

$$\begin{bmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \\ X_{3,1} & X_{3,2} \end{bmatrix} \qquad \begin{bmatrix} 25 & 2 \\ 5 & 26 \\ 3 & 7 \end{bmatrix}$$

Hands-on code demo

Generic Tensor Notation

- Upper-case, bold, italics, sans serif, e.g., $\boldsymbol{\mathsf{X}}$
- In a 4-tensor $\boldsymbol{\mathsf{X}}$, element at position (i, j, k, l) denoted as $\boldsymbol{\mathsf{X}}_{(i, j, k, l)}$

Hands-on code demo of Higher-Rank Tensors

Exercises

1. What is the transpose of this vector?

$$\begin{bmatrix} 25 \\ 2 \\ -3 \\ -23 \end{bmatrix}$$

2. Using algebraic notation, what are the dimensions of this matrix $\boldsymbol{Y}$?

$$y = \begin{bmatrix} 42 & 4 & 7 & 99 \\ -99 & -3 & 17 & 22 \end{bmatrix}$$

3. Using algebraic notation, what is the position of the element in this matrix $\boldsymbol{Y}$ with the value of 17?

Solutions

1. $$\begin{bmatrix} 25 \\ 2 \\ -3 \\ -23 \end{bmatrix}^T = \begin{bmatrix} 25 & 2 & -3 & -23 \end{bmatrix}$$

2. $(2, 4)$

3. $Y_{2,3}$

Intro to Linear Algebra

1. Data Structures for Algebra
2. **Common Tensor Operations**
3. Matrix Properties

Segment 2: Tensor Operations

- Tensor Transposition
- Basic Tensor Arithmetic
- Reduction
- The Dot Product
- Solving Linear Systems

Tensor Transposition

- Transpose of scalar is itself, e.g.: $x^{\mathrm{T}} = x$
- Transpose of vector, seen earlier, converts column to row (and vice versa)
- Scalar and vector transposition are special cases of **matrix transposition**:
  - Flip of axes over **main diagonal** such that:

$$(X^{\mathrm{T}})_{i,j} = X_{j,i}$$

Hands-on code demo

$$\begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ x_{3,1} & x_{3,2} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} x_{1,1} & x_{2,1} & x_{3,1} \\ x_{1,2} & x_{2,2} & x_{3,2} \end{bmatrix}$$

Exercises

1. What is $\boldsymbol{Y}^{\mathrm{T}}$?

$$Y = \begin{bmatrix} 42 & 4 & 7 & 99 \\ -99 & -3 & 17 & 22 \end{bmatrix}$$

2. What is the Hadamard product of these matrices?

$$\begin{bmatrix} 25 & 10 \\ -2 & 1 \end{bmatrix} \odot \begin{bmatrix} -1 & 7 \\ 10 & 8 \end{bmatrix}$$

3. What is the dot product of the tensors $\boldsymbol{w}$ and $\boldsymbol{x}$?

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

$$x = \begin{bmatrix} 5 & 10 & 0 \end{bmatrix}$$

Solutions

$$y^T = \begin{bmatrix} 42 & -99 \\ 4 & -3 \\ 7 & 17 \\ 99 & 22 \end{bmatrix}$$

$$\begin{bmatrix} 25 & 10 \\ -2 & 1 \end{bmatrix} \odot \begin{bmatrix} -1 & 7 \\ 10 & 8 \end{bmatrix} = \begin{bmatrix} -25 & 70 \\ -20 & 8 \end{bmatrix}$$

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

$$x = \begin{bmatrix} 5 & 10 & 0 \end{bmatrix}$$

$$\begin{aligned} w \cdot x &= w_1 x_1 + w_2 x_2 + w_3 x_3 \\ &= (-1)(5) + (2)(10) + (-2)(0) \\ &= -5 + 20 + 0 \\ &= 15 \end{aligned}$$

Answers

1. $y^T = \begin{bmatrix} 42 & -99 \\ 4 & -3 \\ 7 & 17 \\ 99 & 22 \end{bmatrix}$

2. $\begin{bmatrix} -25 & 70 \\ -20 & 8 \end{bmatrix}$

3. 15

Solving Linear Systems

In the next segment, we'll cover enough theory to use matrices to solve some simple linear systems *computationally*.

To understand the problem of solving linear systems more intimately, let's first solve some *algebraically on paper*.

Solving Linear Systems

Method 1: Substitution

- Use whenever there's a variable in system with coefficient of 1

For example, when solving for x and y in the following system:

$$y = \mathbf{3x}$$

$$-5x + 2\mathbf{y} = 2$$

...we can substitute $\mathbf{y}$ with $\mathbf{3x}$ in the second equation.

Substitution

$$\begin{cases} y = 3x \\ -5x + 2y = 2 \end{cases}$$

$$\begin{aligned} -5x + 2y &= 2 \\ -5x + 2(3x) &= 2 \\ -5x + 6x &= 2 \\ x &= 2 \end{aligned}$$

$$\begin{aligned} y &= 3x \\ &= 3(2) \\ &= 6 \end{aligned}$$

$$\therefore (x, y) = (2, 6)$$

Hands-on code demo

Exercises

Solve for the unknowns in the following systems of equations:

1. $x + y = 6$ and $2x + 3y = 16$

2. $-x + 4y = 0$ and $2x - 5y = -6$

3. $y = 4x + 1$ and $-4x + y = 2$

Solutions

1. (2, 4)
2. (-8, -2)
3. No solution.

Solving Linear Systems

Method 2: Elimination

- Typically best option if no variable in system has coefficient of 1
- Use *addition property* of equations to eliminate variables
    - If necessary, multiply one or both equations to make elimination of a variable possible

For example, solve for the unknowns in the following system:

$$2x - 3y = 15$$

$$4x + 10y = 14$$

...by multiplying the first equation by **-2** and adding the equations.

Elimination

$$\begin{cases} (2x - 3y = 15) \times -2 \\ 4x + 10y = 14 \end{cases}$$

$$\begin{cases} -4x + 6y = -30 \\ 4x + 10y = 14 \end{cases}$$

$$\Downarrow$$

$$16y = -16$$

$$y = -1$$

$$2x - 3y = 15$$

$$2x - 3(-1) = 15$$

$$2x + 3 = 15$$

$$2x = 12$$

$$x = 6$$

$$\therefore (x, y) = (6, -1)$$

Hands-on code demo

Exercises

Solve for the unknowns in the following systems of equations:

1. $4x - 3y = 25$ and $-3x + 8y = 10$

2. $-9x - 15y = -15$ and $3x + 5y = -10$

3. $4x + 2y = 4$ and $-5x - 3y = -7$

Solutions

1. (10, 5)
2. No solution.
3. (-1, 4)

Intro to Linear Algebra

1. Data Structures for Algebra
2. Common Tensor Operations
3. **Matrix Properties**

Segment 3: Matrix Properties

- The Frobenius Norm
- Matrix Multiplication
- Symmetric and Identity Matrices
- Matrix Inversion
- Diagonal Matrices
- Orthogonal Matrices

Frobenius Norm

- Described by: $$\|X\|_F = \sqrt{\sum_{i,j} X_{i,j}^2}$$

- Analogous to L^2 norm of vector
- Measures the size of matrix in terms of Euclidean distance
  - It's the sum of the magnitude of all the vectors in $\boldsymbol{X}$

Hands-on code demo

Matrix Multiplication

$$\begin{array}{cc} m & \begin{bmatrix} \quad C \quad \end{bmatrix} \\ & p \end{array} = \begin{array}{cc} m & \begin{bmatrix} \quad A \quad \end{bmatrix} \\ & n \end{array} \begin{array}{cc} n & \begin{bmatrix} B \end{bmatrix} \\ & p \end{array}$$

$$C_{i,k} = \sum_{j} A_{i,j} B_{j,k}$$

Matrix Multiplication (with a Vector)

$$\begin{bmatrix} 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3\cdot 1 + 4\cdot 2 \\ 5\cdot 1 + 6\cdot 2 \\ 7\cdot 1 + 8\cdot 2 \end{bmatrix} = \begin{bmatrix} 3 + 8 \\ 5 + 12 \\ 7 + 16 \end{bmatrix} = \begin{bmatrix} 11 \\ 17 \\ 23 \end{bmatrix}$$

Hands-on code demo

(Matrix-by-)Matrix Multiplication

$$\begin{bmatrix} 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix} \begin{bmatrix} 1 & 9 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 3\cdot1+4\cdot2 & 3\cdot9+4\cdot0 \\ 5\cdot1+6\cdot2 & 5\cdot9+6\cdot0 \\ 7\cdot1+8\cdot2 & 7\cdot9+8\cdot0 \end{bmatrix} = \begin{bmatrix} 11 & 27 \\ 17 & 45 \\ 23 & 63 \end{bmatrix}$$

Hands-on code demo

$$y = a + bx_1 + cx_2 + \dots + mx_m$$

$$\begin{bmatrix} y_1 & a + bx_{1,1} + cx_{1,2} + \dots + mx_{1,m} \\ y_2 & a + bx_{2,1} + cx_{2,2} + \dots + mx_{2,m} \\ \vdots & \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\ y_n & a + bx_{n,1} + cx_{n,2} + \dots + mx_{n,m} \end{bmatrix}$$

For any house i in the dataset, y_i = price and $x_{i,1}$ to $x_{i,m}$ are its features.

We solve for parameters a, b, c to m

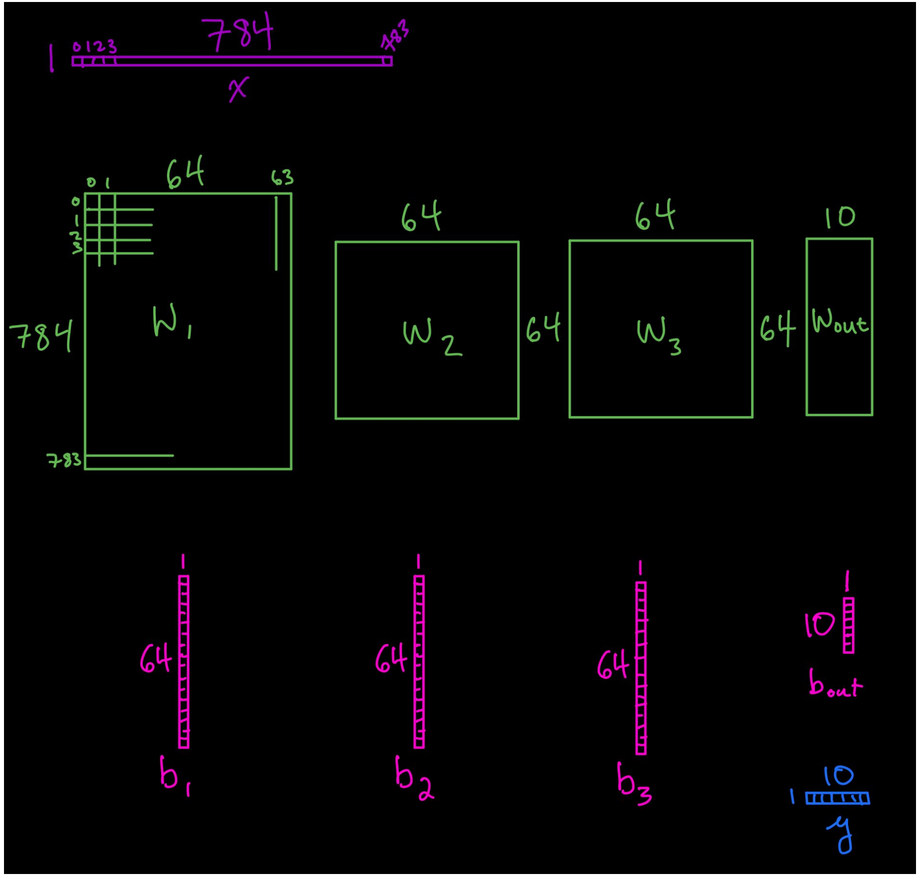
Strictly speaking, x extends rightward to m-1 not m because of the presence of a on the far left.

Matrix Multiplication (in Regression)

$$\begin{array}{c}n\\ \text{cases}\\ \text{tall}\end{array}\left\{\begin{bmatrix} y_1\\ y_2\\ \vdots\\ y_n\end{bmatrix}\right. = \underbrace{\begin{bmatrix}1 & x_{1,1} & x_{1,2} & \cdots & x_{1,m}\\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,m}\\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,m}\end{bmatrix}}_{m\text{ features wide}} \begin{bmatrix}a\\ b\\ c\\ \vdots\\ m\end{bmatrix}$$

In other words, the matrix represents an *m*-dimensional space.

Matrix Multiplication (in Deep Learning)



See:

- artificial-neurons.ipynb
- jonkrohn.com/deepTF1
- jonkrohn.com/convTF1
- jonkrohn.com/convTF2
- jonkrohn.com/deepPT

Symmetric Matrices

Special matrix case with following properties:

- Square
- $\boldsymbol{X}^T = \boldsymbol{X}$

Hands-on code demo

$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 7 & 8 \\ 2 & 8 & 9 \end{bmatrix}$$

Identity Matrices

Symmetric matrix where:

- Every element along main diagonal is 1
- All other elements are 0
- Notation: $\boldsymbol{I}_n$ where n = height (or width)
- n-length vector unchanged if multiplied by $\boldsymbol{I}_n$

Hands-on code demo

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

I_4

Exercises

Using paper and pen(cil), multiply the following tensors:

1. $$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -2 \end{bmatrix}$$

2. Repeat Q1 using the same vector but replace matrix with $\boldsymbol{I_3}$

3. $$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 1 & 1 \\ -2 & 2 \end{bmatrix}$$

Solutions

1. [-3, -9, -15]
2. [-1, 1, -2]
3. $$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 1 & 1 \\ -2 & 2 \end{bmatrix} = \begin{bmatrix} -3 & 5 \\ -9 & 14 \\ -15 & 23 \end{bmatrix}$$

Matrix Inversion

- Clever, convenient approach for solving linear equations
- An alternative to manually solving with substitution or elimination

- **Matrix inverse** of $\boldsymbol{X}$ is denoted as $\boldsymbol{X}^{-1}$
  - Satisfies: $\boldsymbol{X}^{-1}\boldsymbol{X} = \boldsymbol{I}_n$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$I_4$$

Matrix Inversion

$$
\text{n cases tall}\left\{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}\right. = \underbrace{\begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{bmatrix}}_{m \text{ features wide}} \begin{bmatrix} a \\ b \\ c \\ \vdots \\ m \end{bmatrix}
$$

The regression formula can be represented as:

$\boldsymbol{y} = \boldsymbol{Xw}$ ($\boldsymbol{w}$ is the vector of weights a through m)

Matrix Inversion

In the equation $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}$:

- We know the outcomes $\boldsymbol{y}$, which could be house prices
- We know the features $\boldsymbol{X}$, which are predictors like bedroom count
- Vector $\boldsymbol{w}$ contains the unknowns, the model's learnable parameters

Assuming $\boldsymbol{X}^{-1}$ exists, matrix inversion can solve for $\boldsymbol{w}$:

$$\boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}$$

$$\boldsymbol{X}^{-1}\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^{-1}\boldsymbol{y}$$

$$\boldsymbol{I}_n\boldsymbol{w} = \boldsymbol{X}^{-1}\boldsymbol{y}$$

$$\boldsymbol{w} = \boldsymbol{X}^{-1}\boldsymbol{y}$$

Matrix Inversion

$$\begin{cases} 4b + 2c = 4 \\ -5b - 3c = -7 \end{cases}$$

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ -5 & -3 \end{bmatrix} \qquad y = \begin{bmatrix} 4 \\ -7 \end{bmatrix}$$

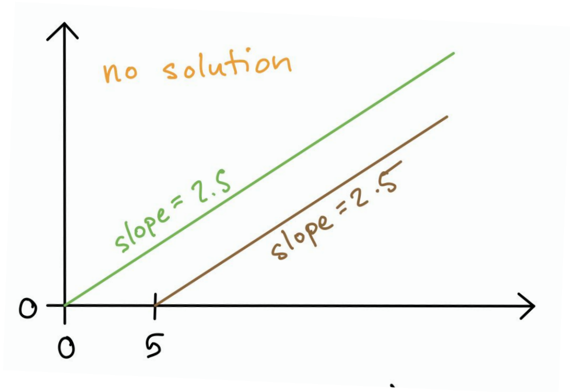
$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix} = X^{-1} y$$

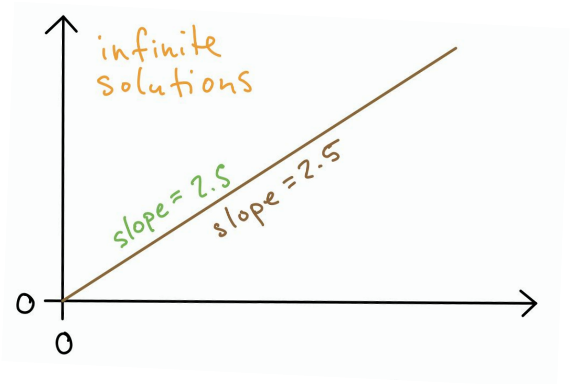
Hands-on code demo

Matrix Inversion

Nifty trick, but can only be calculated if:

- Matrix isn't "singular"
- That is, all columns of matrix must be linearly independent
  - **E.g., if a column is [1, 2], another can't be [2, 4] or also be [1, 2]**

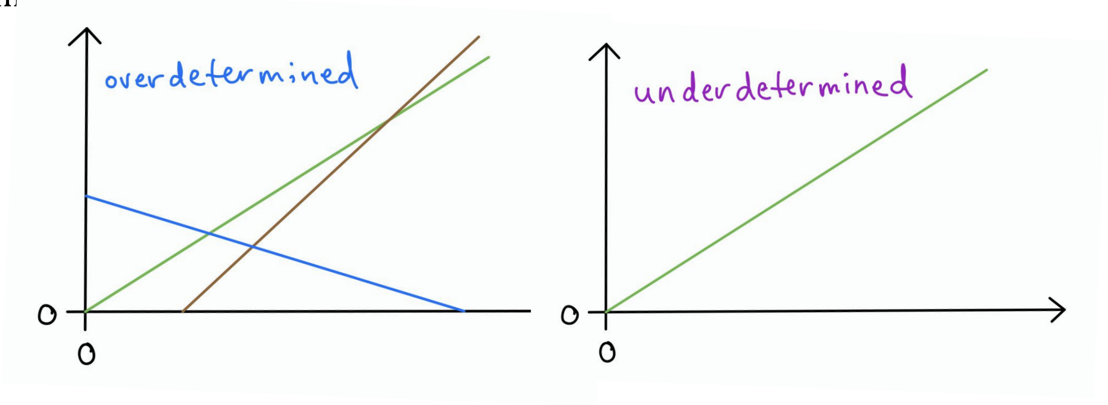




Matrix Inversion

...Can also only be calculated if:

- Matrix is square: $n_{row} = n_{col}$ (i.e., "vector span" = "matrix range")
  - Avoids **overdetermination**: $n_{row} > n_{col}$ -- i.e.: $n_{equations} > n_{dimensions}$
  - Avoids **underdetermination**: $n_{row} < n_{col}$ -- i.e.: $n_{equations} <$ $n_{\text{dimensions}}$



Note that solving for unknowns may still be possible by other means if matrix can't be inverted (Algebra II)

Diagonal Matrices

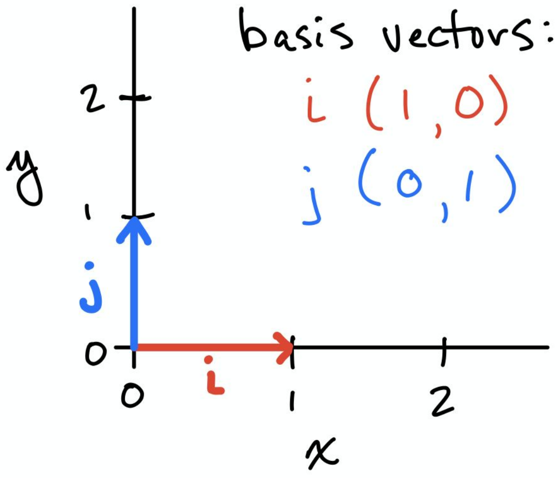
- Nonzero elements along main diagonal; zeros everywhere else
- Identity matrix is an example
- If square, denoted as diag($\boldsymbol{x}$) where $\boldsymbol{x}$ is vector of main-diagonal elements
- Computationally efficient:
  - Multiplication: diag($\boldsymbol{x}$)$\boldsymbol{y}$ = $\boldsymbol{x} \odot \boldsymbol{y}$
  - Inversion: diag($\boldsymbol{x}$)$^{-1}$ = diag$[1/\boldsymbol{x}_1, \ldots, 1/\boldsymbol{x}_n]^T$
    - Can't divide by zero so $\boldsymbol{x}$ can't include zero
- Can be non-square and computation still efficient:
  - If $h > w$, simply add zeros to product
  - If $w > h$, remove elements from product

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

I_4

Orthogonal Matrices

Recall orthonormal vectors from earlier:



In orthogonal matrices, orthonormal vectors:

- Make up all rows
- Make up all columns

This means: $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{I}$

Which also means: $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}^{-1}\boldsymbol{I} = \boldsymbol{A}^{-1}$

Calculating $\boldsymbol{A}^{\mathrm{T}}$ is cheap, therefore so is calculating $\boldsymbol{A}^{-1}$

Note that "orthonormal matrix" isn't a thing.

And there's no name for a matrix made of orthogonal vectors.

Exercises

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

I_3

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

K

Identity matrices are orthogonal.

1. With paper and pencil, use the dot product to demonstrate that any two columns of $\boldsymbol{I}_3$ are orthogonal to each other.
2. Similarly, use paper and pencil to demonstrate that each of the columns of $\boldsymbol{I}_3$ has unit norm.
3. Repeat Exercises (1.) and (2.) using code, e.g., NumPy.
4. Now using matrix $\boldsymbol{K}$ instead of $\boldsymbol{I}_3$, repeat Exercises (1.) through (3.) to assess whether $\boldsymbol{K}$ is orthogonal.

POLL *with Multiple Answers Possible*

What follow-up topics interest you most?

- More Linear Algebra
- Calculus
- Probability / Statistics
- Computer Science (e.g., algorithms, data structures)
- Machine Learning Basics
- Advanced Machine Learning, incl. Deep Learning
- Something Else

Stay in Touch

jonkrohn.com to sign up for email newsletter

linkedin.com/in/jonkrohn

youtube.com/c/JonKrohnLearns

twitter.com/JonKrohnLearns

NEBULA

PLACEHOLDER
FOR:

5-Minute Timer

PLACEHOLDER
FOR:

10-Minute Timer

PLACEHOLDER
FOR:

15-Minute Timer