Data Science for Mathematicians
Lesson 1:
The Data Science Landscape & The Geometry of Data

Abstract

This lecture introduces the foundational premise of the course: that the vast field of data science is, at its core, a modern form of applied mathematics. We will formally establish the language of linear algebra as the primary tool for describing and manipulating data. The central thesis is the reframing of a dataset from a tabular collection of numbers into a geometric object—a cloud of points residing in a high-dimensional vector space. We will revisit fundamental concepts such as the dot product and vector norms, reinterpreting them as powerful measures of similarity and distance between data points. This geometric perspective provides the rigorous mathematical foundation upon which virtually all machine learning algorithms, including those to be studied in subsequent weeks, are built.

1 Introduction to a Language for Data

Welcome to Data Science for Mathematicians. The primary objective of this course is to bridge the perceived gap between the abstract, axiomatic world of theoretical mathematics and the practical, high-impact field of data science. Many of you have spent years building a sophisticated toolkit in areas like linear algebra, calculus, and probability theory. You have been trained to think in terms of structures, transformations, and proofs. This course is designed to demonstrate that this rigorous training is not merely a prerequisite for data science; it is its very essence.

This course is deliberately distinct from vocational data science programs or software-specific bootcamps. Our goal is not to master the application programming interface (API) of a particular library like Scikit-learn or TensorFlow. While such tools are indispensable for the practitioner, our purpose is to understand the mathematical principles that make them possible. We will focus on the *why* behind the algorithms, not just the *how*. The aim is to cultivate the ability to reason about data from first principles, to derive models, to understand their limitations, and to critically evaluate the assumptions upon which they are built. A mathematician's training in abstraction, logical deduction, and the pursuit of proof provides a profound advantage in a field that is too often treated as a collection of black-box techniques.

The central idea of this inaugural lecture, and indeed the entire course, is a fundamental shift in perspective. We will learn to see data not as numbers in a spreadsheet, but as a geometric object. The core conceptual leap is this: an observation, represented as a row in a table, is in fact a point in a high-dimensional space. A feature, represented as a column, corresponds to a dimension of that space. Our entire dataset, therefore, is a cloud of points, and its structure—its shape, orientation, and the distances between its points—contains the patterns we seek to uncover.

Your existing knowledge of vector spaces, inner products, norms, and matrix decompositions is the precise language required to describe and manipulate this geometric structure. The most powerful tool a mathematician brings to this field is the ability to abstract. The act of viewing a row of disparate measurements—a person's age, their body mass index, the number of children they have—as a single, unified entity, a vector x, is an act of profound abstraction. It is this single step that unlocks the entire machinery of linear algebra and allows us to translate questions about data into well-posed mathematical problems concerning the properties of a vector space: its dimensionality, the angles between its vectors, and the existence of lower-dimensional subspaces that might capture the data's essential characteristics. Our first task is not merely to learn a new technique, but to adopt this new way of seeing.

2 The Data Science Landscape: An Applied Mathematician's Perspective

2.1 Defining Data Science

Before we delve into the mathematical formalism, it is essential to situate our work within the broader context of data science.

Definition 2.1. Data Science is an interdisciplinary field that employs scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured and unstructured data. It lies at the intersection of statistics, computer science, and domain expertise, and is fundamentally grounded in the principles of applied mathematics.

As illustrated in Figure 1, the intersection of these fields gives rise to specialized sub-domains such as Machine Learning (the blend of algorithmic computation and statistical inference) and traditional research methodology. Crucially, the visual model posits that this entire interdisciplinary structure is not merely a collection of tools but is fundamentally grounded in the principles of applied mathematics—specifically linear algebra, calculus, and optimization—which provide the rigorous theoretical basis for extracting knowledge from structured and unstructured data.

Based on the definition provided, here are four examples categorized by whether they fit the criteria of data ccience. The key differentiator here is the use of modeling to predict or understand a complex, uncertain (stochastic) system, rather than simply retrieving information or performing exact calculations.

Example 2.2 (Streaming Recommendations)**.** In this scenario, a media streaming platform utilizes user interaction history (viewing time, clicks, pauses) and content metadata (genre, director) to predict future viewing preferences. This fits the definition because it treats user preference as a stochastic (random) system rather than a fixed rule. The system employs matrix factorization or neural networks—modern applied mathematics techniques—to analyze massive datasets of unstructured behavior. It does not merely list available movies; it builds a predictive model to extract the insight of "what the user is likely to enjoy next," effectively quantifying human taste through computation.

Example 2.3 (Predictive Maintenance in Manufacturing)**.** A factory equips its machinery with sensors that measure vibration, temperature, and sound in real-time. Instead of replacing parts on a fixed schedule, they analyze this sensor data to predict when a machine is likely to fail. This application lies at the intersection of domain expertise (mechanical engineering) and statistics. The problem involves extracting knowledge (the probability of imminent failure) from a continuous stream of structured sensor data. Unlike a simple threshold alarm (which would be engineering), this approach uses historical failure data to model the complex degradation patterns of the machinery, optimizing the system based on probabilistic outcomes.

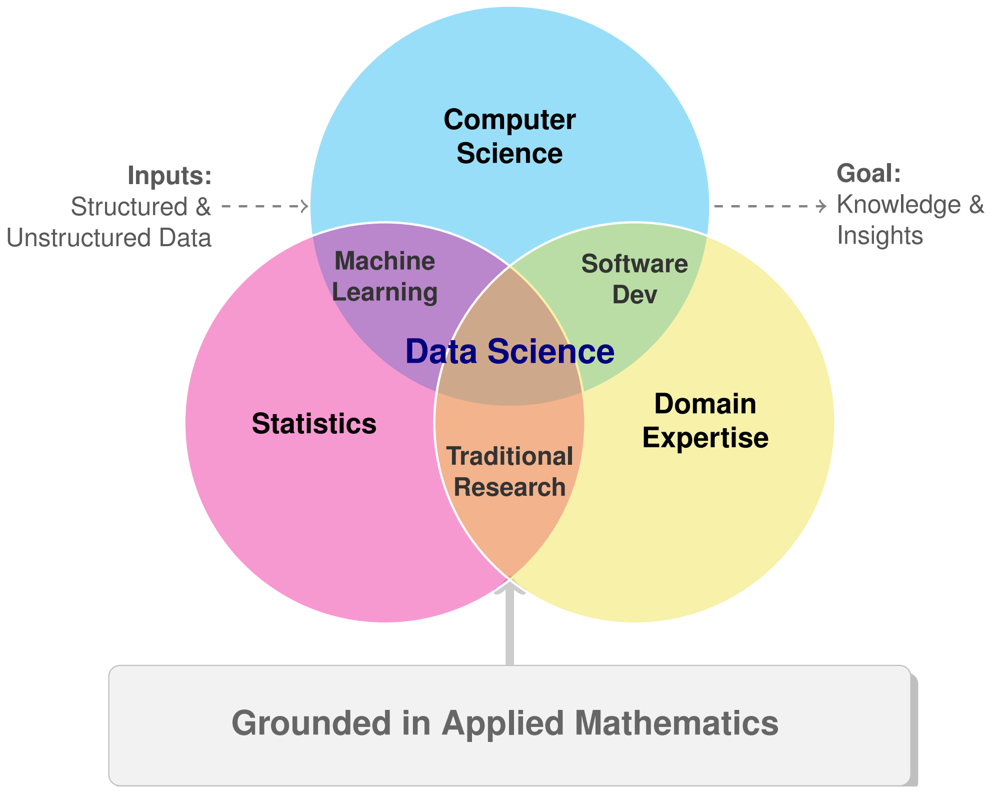


Figure 1: A Venn diagram illustrating the multidisciplinary nature of data science.

Example 2.4 (Digital Archiving and Record Digitization)**.** A hospital scans thousands of paper patient records and manually enters patient names, dates of birth, and ID numbers into a central SQL database for secure storage and easy retrieval. While this involves computer science and data storage, it lacks the extraction of insights. The process is deterministic: the goal is to create a faithful digital copy of physical information. There is no mathematical modeling, no statistical inference regarding patient health, and no attempt to describe a complex system. It is a data management and engineering task, not an investigative scientific inquiry.

Example 2.5 (Payroll and Tax Calculation)**.** A corporate software system calculates the monthly salary for employees. It takes the number of hours worked, multiplies it by the hourly rate, subtracts a fixed percentage for taxes as defined by government law, and outputs the final paycheck amount. This is an example of classical deterministic calculation. The system is not *learning* from the data, nor is it dealing with a stochastic system; the relationship between hours worked and pay is defined by precise, unvarying arithmetic rules. While it uses applied mathematics (arithmetic and algebra), it does not require the scientific method or statistical modeling to derive a solution, as the solution is already analytically known and fixed.

From our perspective, data science can be viewed as a modern, computationally intensive form of applied mathematics. While classical applied mathematics might have focused on deriving analytical solutions to differential equations describing physical systems, data science focuses on building and analyzing models that describe complex, often stochastic, systems from which we have observational data. The explosion in computational power and data storage has not changed the fundamental nature of the inquiry; it has simply expanded the scale and complexity of the problems we can address.

Table 1: Summary of Distinction: Data Science vs. Non-Data Science

Feature	**Data Science Case**	**Non-Data Science Case**
Primary Goal	Extracting unknown insights or predictions.	Executing known rules or storage.
System Type	**Stochastic** (probabilistic/uncertain)	**Deterministic** Deterministic (fixed/certain).
Math Basis	Statistical modeling, optimization and inference.	Discrete mathematics, arithmetic, boolean logic, and set theory.

Table 1 delineates the critical boundaries between data science and general computational tasks. By juxtaposing the operational goals, the nature of the systems involved, and the underlying mathematical frameworks, we isolate the defining characteristic of data science: the probabilistic modeling of stochastic phenomena versus the deterministic execution of defined logic. Readers should observe that the presence of large datasets alone does not constitute data science; rather, it is the method of inquiry—extracting unknown insights through modeling—that defines the field.

2.2 The Data Science Workflow

A typical data science project, regardless of the specific domain, follows a canonical workflow. Understanding this process allows us to identify where a mathematician's skills provide the most significant leverage.

1. **Problem Formulation & Business Understanding**: This is the crucial first step of translating a often vaguely-defined business or scientific objective into a precise, quantifiable problem. For example, a business goal like "reduce customer churn" must be reformulated as a specific machine learning task, such as "build a binary classifier to predict the probability of a customer cancelling their subscription within the next quarter."

2. **Data Acquisition and Cleaning**: This stage involves gathering data from various sources (databases, APIs, logs) and preparing it for analysis. This includes handling missing values, correcting errors, and reconciling inconsistencies—a process often referred to as data wrangling or munging.

3. **Exploratory Data Analysis (EDA)**: Before any formal modeling, the data is investigated to understand its underlying structure. This is often done through visualization (plotting distributions, scatter plots) and summary statistics to form initial hypotheses about the data.

4. **Modeling**: A mathematical model is selected and trained on the data to perform the task defined in the first step. This could be a statistical model, a machine learning algorithm, or a custom-derived formulation.

5. **Evaluation**: The performance of the model is rigorously assessed using quantitative metrics. This step measures how well the model generalizes to new, unseen data and determines its fitness for the intended purpose.

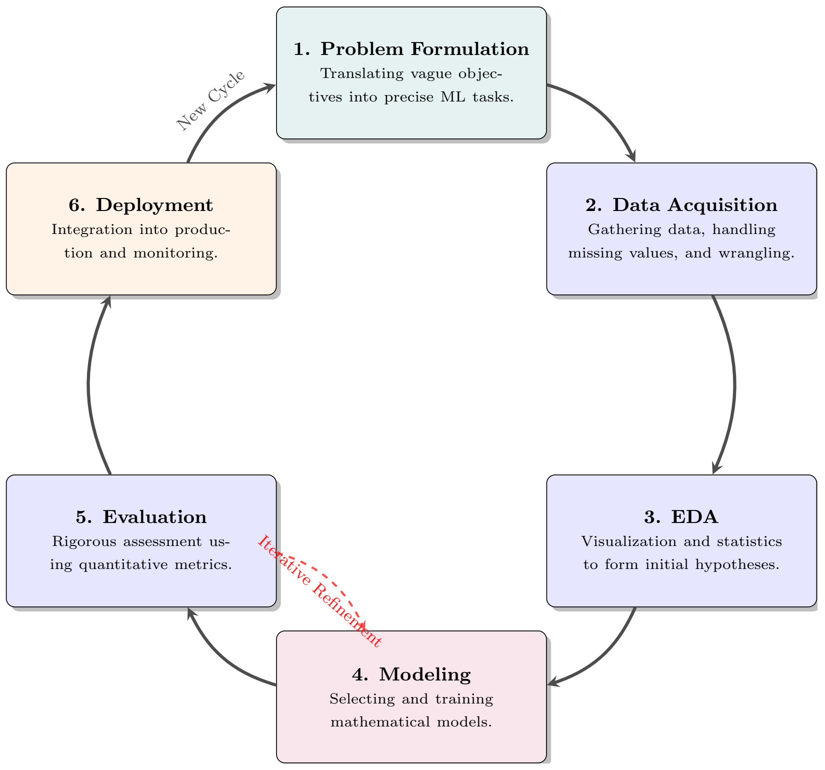

1. Problem Formulation
Translating vague objectives into precise ML tasks.
2. Data Acquisition
Gathering data, handling missing values, and wrangling.
3. EDA
Visualization and statistics to form initial hypotheses.
4. Modeling
Selecting and training mathematical models.
5. Evaluation
Rigorous assessment using quantitative metrics.
6. Deployment
Integration into production and monitoring.
New Cycle
Iterative Refinement

Figure 2: The data science workflow

6. **Deployment & Monitoring**: The finalized model is integrated into a production environment where it can make real-time predictions or provide insights. Its performance is continuously monitored to detect degradation over time.

Figure 2 highlights the cyclical nature of the workflow, specifically connecting Deployment back to Problem Formulation, while preserving the internal iterative loop between Modeling and Evaluation.

2.3 The Mathematician's Role in the Workflow

While programming and software engineering skills are necessary for implementing this workflow, the unique and indispensable value of a mathematician is concentrated in the stages that demand the highest levels of abstraction and logical rigor: problem formulation, modeling, and evaluation.

The data science workflow is not merely a sequence of technical steps; it is the construction of a rigorous, logical argument. The mathematician's role is to serve as the architect of this argument, ensuring its soundness by providing the correct mathematical *grammar*.

- **Problem Formulation as Axiomatization:** A practitioner might approach a problem by testing multiple algorithms to see which performs best on a given metric—an empirical, "guess-and-check" approach. A mathematician, by contrast, begins by formalizing the problem itself. Is this a problem of regression, classification, or clustering? What is the nature of the output space? What loss function appropriately captures the penalty for incorrect predictions? This stage is akin to choosing the axioms of a system. The choice of framework (e.g., supervised vs. unsupervised learning) dictates the entire logical structure of the solution that follows.

- **Model Selection as Theorem Proving:** Once the problem is framed, a model must be chosen. This is not an arbitrary choice. A model is a set of mathematical assumptions about the data-generating process. A linear model, for example, assumes an additive relationship between features and the outcome. A mathematician understands that choosing a model is equivalent to stating a theorem: "Assuming the data follows properties A, B, and C, then algorithm X is an optimal way to find the solution." The skill lies in selecting a model whose assumptions are justified by both domain knowledge and evidence from exploratory data analysis.
- **Critical Analysis as Peer Review:** Every model, no matter how complex, is a simplification of reality. Its validity rests entirely on its assumptions. The mathematician's training is to relentlessly question these assumptions. Are the features truly independent as required by Naive Bayes? Is the variance of the errors constant, as assumed in ordinary least squares? What are the mathematical consequences if these assumptions are violated? This critical posture, analogous to the peer review of a mathematical proof, is what separates robust, reliable data science from the fragile and often misleading application of black-box algorithms. It ensures the logical argument holds from premise to conclusion.

In this view, the mathematician is not just another practitioner in the workflow; they are the arbiter of its logical and mathematical coherence, elevating their role from analyst to problem architect.

3 The Geometry of Data: From Observations to Vectors

We now begin the formal process of building our geometric framework. The first step is to establish a precise mapping from the concrete world of data collection to the abstract world of linear algebra.

3.1 A Concrete Example: Medical Insurance Costs

Throughout this lecture, we will use a simplified version of a common dataset used for predicting personal medical insurance costs.

Example 3.1 (Medical insurance costs)**.** An insurance provider collects data on its customers to understand the factors that influence their healthcare expenses. An "observation" corresponds to a single individual. For each individual, several "features" are recorded. For our purposes, we will consider a subset of four numerical features:

- **Age**: The age of the primary beneficiary in years (integer).
- **BMI**: Body Mass Index, a measure of body fat based on height and weight (continuous, real-valued).
- **Children**: The number of children/dependents covered by the insurance plan (integer).
- **Charges**: The individual medical costs billed by the health insurance in US dollars (continuous, real-valued). This is often the variable we wish to predict.

A small sample of such a dataset might look like the table below.

Table 2: Sample Data for Medical Insurance Costs

Observation	**Age**	**BMI**	**Children**	**Charges ($)**
Person 1 (x_1)	19	27.9	0	16884.92
Person 2 (x_2)	35	35.5	1	44501.40
Person 3 (x_3)	62	26.3	0	27808.72
Person 4 (x_4)	41	21.8	1	6272.48
Person 5 (x_5)	25	42.1	2	48824.45

3.2 The Foundational Leap: Observations as Vectors

The critical step, as previously mentioned, is to cease viewing each row as a collection of disparate numbers and instead see it as a single, unified mathematical object: a vector.

Definition 3.2 (Observation Vector)**.** An observation consisting of p numerical features is represented as a **vector** $x \in \mathbb{R}^p$. Each component of the vector, x_j for $j = 1, \ldots, p$, corresponds to the value of the j-th feature.

Example 3.3. Consider the first individual (Person 1) from the medical insurance dataset in Example 3.1. The observed features are Age = 19, BMI = 27.9, Children = 0, and Charges = 16884.92. We formalize this observation as a vector

$$x_1 = \begin{bmatrix} 19 \\ 27.9 \\ 0 \\ 16884.92 \end{bmatrix} \in \mathbb{R}^4.$$

Here, the vector resides in a 4-dimensional Euclidean space, where each axis corresponds to one of the measured features.

Similarly, for the second individual (Person 2), the features correspond to Age = 35, BMI = 35.5, Children = 1, and Charges = 44501.40. The corresponding observation vector

$$x_2 = \begin{bmatrix} 35 \\ 35.5 \\ 1 \\ 44501.40 \end{bmatrix} \in \mathbb{R}^4.$$

By representing these disparate database rows as vectors in the same vector space $\mathbb{R}^4$, we enable the use of algebraic operations (such as calculating the distance between vectors) to quantify the similarity between different patients.

This act of representation is the gateway to applying the entirety of geometric and algebraic reasoning to our data.

3.3 The Feature Space

Once we conceptualize individual observations as vectors, the entire dataset becomes a collection of vectors. This collection resides within a specific vector space which we call the feature space.

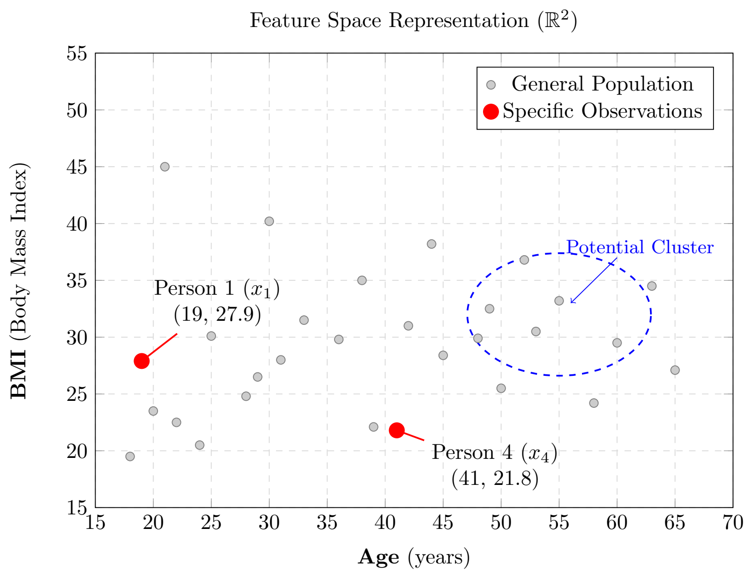


Figure 3: Visualizing the dataset as a point cloud in $\mathbb{R}^2$. Each dot represents a customer vector $x = [\text{Age}, \text{BMI}]^T$.

Definition 3.4. The p-dimensional vector space $\mathbb{R}^p$, in which each dimension corresponds to a feature and each point corresponds to a possible observation, is called the **feature space**. A dataset of n observations is thus a set of n points, $\{x_1, x_2, \dots, x_n\}$, forming a **point cloud** within this feature space.

The implications of this definition are profound. The abstract *structure* in the data, which data scientists seek to uncover, is now given a concrete geometric meaning. It is the literal geometry of this point cloud.

- **Clusters** in the data (e.g., groups of similar customers) correspond to dense regions of points in the feature space.
- **Trends** and relationships (e.g., charges increasing with age) correspond to the point cloud being elongated or oriented along certain directions within the space.
- **Outliers** are points that lie far away from the main body of the cloud.

Example 3.5 (2D Feature Space and Visualization)**.** Let us simplify the medical insurance problem to consider only two features: Age and BMI ($p = 2$). The feature space is the 2-dimensional Euclidean plane, $\mathbb{R}^2$. The horizontal axis represents Age, and the vertical axis represents BMI. In this feature space, see Figure 3, Person 1 corresponds to the point $(19, 27.9)$ and Person 4 corresponds to the point $(41, 21.8)$. If we were to plot all n customers in the database, they would form a scatter plot—a point cloud—scattered across this plane. Clusters of points in this space would indicate groups of customers with similar physical characteristics and ages.

Example 3.6 (High-Dimensional Feature Space)**.** When we include all four features (Age, BMI, Children, Charges), the feature space becomes $\mathbb{R}^4$. While humans cannot easily visualize spatial relationships in four dimensions, the mathematical definition holds: each customer is a single point fixed at a specific coordinate in this 4-dimensional abstract space. In this context, the point cloud is the collection of all customer vectors suspended in this hyper-space. Data science techniques, such as regression or clustering, operate by analyzing the geometric shape and density of this cloud. For instance, finding customers who are *close* to each other in this feature space implies finding individuals with similar risk profiles and insurance costs.

Remark 3.7. While the geometric view is powerful, it also reveals a fundamental challenge in data science. As the number of features p increases, the volume of the feature space grows exponentially. For a fixed number of data points n, the space becomes increasingly empty, and the points become sparse. This phenomenon, known as the **curse of dimensionality**, makes it statistically difficult to find meaningful patterns, as the distance between any two points can become less informative. Much of advanced machine learning is dedicated to developing methods to overcome this challenge.

3.4 The Data Matrix: Our Central Object

Finally, we organize our collection of n observation vectors into a single, elegant mathematical structure called the *data matrix*.

Definition 3.8. A dataset consisting of n observations, $\{x_1, \dots, x_n\}$, each with p features, is represented by a **data matrix** $X \in \mathbb{R}^{n \times p}$. By convention, each row of X is the transpose of an observation vector, x_i^T. Each column of X is a vector $X_{:,j} \in \mathbb{R}^n$ containing all values for a single j-th feature. The entry x_{ij} is the value of the j-th feature for the i-th observation.

Example 3.9 (Design Matrix for Medical Insurance)**.** Using the insurance dataset of $n = 5$ observations and $p = 4$ features (Age, BMI, Children, Charges), we construct the data matrix $X \in \mathbb{R}^{5\times4}$ by stacking the transpose of each observation vector x_i.

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ x_3^T \\ x_4^T \\ x_5^T \end{bmatrix} = \begin{bmatrix} 19 & 27.9 & 0 & 16884.92 \\ 35 & 35.5 & 1 & 44501.40 \\ 62 & 26.3 & 0 & 27808.72 \\ 41 & 21.8 & 1 & 6272.48 \\ 25 & 42.1 & 2 & 48824.45 \end{bmatrix}$$

In this matrix, the scalar entry $x_{2,2} = 35.5$ represents the BMI of the second individual. The third column vector, $X_{:,3} = [0, 1, 0, 1, 2]^T \in \mathbb{R}^5$, represents the Children feature across the entire population sample.

Example 3.10 (High-Dimensional Data: MNIST Digits)**.** Consider a computer vision task involving the MNIST dataset, which consists of handwritten digits.

A single observation (a digit) is originally captured as a 28×28 pixel grayscale grid. Mathematically, this is a matrix $M \in \mathbb{R}^{28\times28}$, where each entry m_{jk} represents a pixel intensity (e.g., 0 for black, 255 for white).

To perform standard linear algebra operations, we must *flatten* this grid. We concatenate the rows of matrix M into a single long sequence, resulting in an observation vector $x \in \mathbb{R}^{28\times28} = \mathbb{R}^{784}$.

$$x = [m_{1,1}, \dots, m_{1,28}, m_{2,1}, \dots, m_{28,28}]^T$$

Consequently, the feature space is $\mathbb{R}^{784}$. While we cannot visualize 784 dimensions, the geometric intuition remains: every possible 28×28 image exists as a single point in this space. All images of the number “1” will form a cluster in one region of this high-dimensional space, distinct from the cluster formed by images of the number “8”.

If we have a dataset of $n = 10,000$ such images, we stack the transpose of their observation vectors to form the data matrix $X \in \mathbb{R}^{10,000\times784}$.

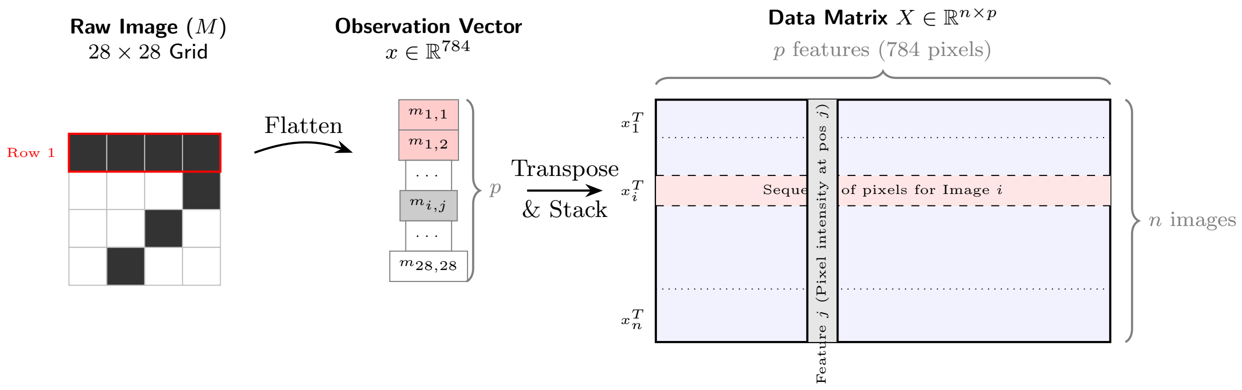


Figure 4: The process of flattening a 2D MNIST digit into a vector and stacking it into the Data Matrix X. Rows represent individual images; columns represent specific pixel positions.

- The i-th row x_i^T represents the complete i-th image.
- The j-th column represents the intensity of the j-th pixel across all 10,000 images.

The data matrix is the central mathematical object for most of this course. Nearly every algorithm we will study, from linear regression to principal component analysis, can be expressed as a series of operations (e.g., multiplication, decomposition) on this matrix. It contains all the information of our dataset in a structured form amenable to the tools of linear algebra.

4 Measuring Relationships in Feature Space: The Dot Product Revisited

Having established our data as a collection of vectors, we now revisit fundamental linear algebraic operations, starting with the dot product, and reinterpret them as powerful tools for data analysis.

4.1 The Dot Product as a Measure of Similarity

You are all familiar with the dot product (or inner product) of two vectors $u, v \in \mathbb{R}^p$. While often introduced simultaneously via geometry and algebra, in rigorous data science we define the operation algebraically and derive the geometric intuition as a consequence.

Definition 4.1 (The Dot Product)**.** Let $u = (u_1, \ldots, u_p)$ and $v = (v_1, \ldots, v_p)$ be vectors in $\mathbb{R}^p$. The **dot product** of u and v, denoted $u \cdot v$, is defined as the sum of the products of their corresponding components:

$$u \cdot v = u^T v = \sum_{i=1}^{p} u_i v_i$$

Example 4.2 (Basic Computation in $\mathbb{R}^3$)**.** Consider two vectors $u, v \in \mathbb{R}^3$:

$$u = \begin{bmatrix} 2 \\ 5 \\ -1 \end{bmatrix}, \quad v = \begin{bmatrix} 4 \\ -2 \\ -3 \end{bmatrix}$$

To compute the dot product $u \cdot v$, we multiply corresponding components and sum the results:

$$\begin{aligned} u \cdot v &= (2)(4) + (5)(-2) + (-1)(-3) \\ &= 8 - 10 + 3 \\ &= 1 \end{aligned}$$

The result is a scalar (a single real number), which is why the dot product is also referred to as the scalar product.

Example 4.3 (The Dot Product as a Weighted Sum)**.** In data science, the dot product is the mechanism behind linear regression. Suppose we wish to predict a risk scor based on a patient's features.

- Let $x \in \mathbb{R}^2$ be the feature vector for a patient: $x = [\text{Age}, \text{BMI}]^T = [40, 30]^T$.
- Let $w \in \mathbb{R}^2$ be a *weight vector* determined by a machine learning model: $w = [0.5, 2.0]^T$.

The model's prediction is simply the dot product of the weights and the features:

$$\text{Score} = w \cdot x = (0.5)(40) + (2.0)(30) = 20 + 60 = 80$$

Here, the dot product acts as an accumulator, aggregating the contribution of each feature weighted by its importance (w) into a single predictive value.

This algebraic definition allows for efficient computation, particularly when using matrix operations. However, its utility in data science stems from its deep connection to geometry, formalized by the following theorem.

Theorem 4.4 (Geometric Interpretation)**.** *Let u and v be non-zero vectors in $\mathbb{R}^p$, and let $\theta \in [0, \pi]$ be the angle between them. Then:*

$$u \cdot v = \|u\|_2 \|v\|_2 \cos\theta$$

where $\|x\|_2 = \sqrt{x \cdot x}$ denotes the Euclidean norm (length) of vector x.

Proof. Consider the triangle formed by the vectors u, v, and their difference $u - v$. The Law of Cosines states that the squared length of the third side is

$$\|u - v\|_2^2 = \|u\|_2^2 + \|v\|_2^2 - 2\|u\|_2\|v\|_2 \cos\theta.$$

Alternatively, we can expand the term $\|u - v\|_2^2$ using the distributivity and symmetry properties of the dot product as follows.

$$\begin{aligned} \|u - v\|_2^2 &= (u - v) \cdot (u - v) \\ &= u \cdot u - u \cdot v - v \cdot u + v \cdot v \\ &= \|u\|_2^2 - 2(u \cdot v) + \|v\|_2^2. \end{aligned}$$

By equating the two expressions for $\|u - v\|_2^2$

$$\|u\|_2^2 + \|v\|_2^2 - 2\|u\|_2\|v\|_2 \cos\theta = \|u\|_2^2 - 2(u \cdot v) + \|v\|_2^2.$$

Subtracting $\|u\|_2^2 + \|v\|_2^2$ from both sides yields:

$$-2\|u\|_2\|v\|_2 \cos\theta = -2(u \cdot v).$$

Dividing by -2, we arrive at the conclusion

$$u \cdot v = \|u\|_2\|v\|_2 \cos\theta.$$

□

In a data science context, we move beyond the purely computational view and interpret the dot product as a measure of **similarity** or **alignment** between two data vectors. The geometric definition makes this interpretation clear:

- If $u \cdot v > 0$, then $\cos\theta > 0$, which implies the angle θ is acute ($0 \leq \theta < \pi/2$). The vectors point in a generally similar direction. A large positive value indicates strong alignment.
- If $u \cdot v < 0$, then $\cos\theta < 0$, implying θ is obtuse ($\pi/2 < \theta \leq \pi$). The vectors point in generally opposite directions.
- If $u \cdot v = 0$, then $\cos\theta = 0$, implying $\theta = \pi/2$. The vectors are **orthogonal**. In a data context, this suggests a form of "unrelatedness" or independence in a linear sense.

This geometric interpretation is the foundation of techniques like *cosine similarity*, which normalizes the dot product to depend only on the angle, providing a pure measure of orientation similarity irrespective of vector magnitudes:

$$\cos\theta = \frac{u \cdot v}{\|u\|_2 \|v\|_2}$$

This metric is widely used in fields like natural language processing to compare the similarity of documents represented as high-dimensional vectors.

Example 4.5 (Collaborative Filtering)**.** Consider a movie recommendation system where user preferences are represented as *mean-centered ratings* (where 0 is neutral, positive is "like", and negative is "dislike").

- **User A (Sci-Fi fan)**: Loves *Star Wars* (+2) and *Dune* (+2).
- **User B (Sci-Fi fan)**: Loves *Star Wars* (+1) and *Dune* (+3).

$$u_A = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad u_B = \begin{bmatrix} 1 \\ 3 \end{bmatrix}.$$

$$u_A \cdot u_B = (2)(1) + (2)(3) = 2 + 6 = 8.$$

Since $u_A \cdot u_B = 8 > 0$, the angle is acute. The vectors point in the same general direction, indicating the users share a similar taste profile. The system should recommend movies User A likes to User B.

Example 4.6 (Financial Correlation)**.** Consider the daily percentage returns of two financial assets over a 3-day period.

- **Asset A (Oil Company)**: Stock generally rises when oil prices go up.
- **Asset B (Airline Company)**: Stock generally falls when oil prices go up (increased fuel costs).

$$v_{\text{Oil}} = \begin{bmatrix} 2\% \\ 1\% \\ 3\% \end{bmatrix}, \quad v_{\text{Air}} = \begin{bmatrix} -1\% \\ -2\% \\ -1\% \end{bmatrix}$$

$$v_{\text{Oil}} \cdot v_{\text{Air}} = (2)(-1) + (1)(-2) + (3)(-1) = -2 - 2 - 3 = -7.$$

Since $v_{\text{Oil}} \cdot v_{\text{Air}} = -7 < 0$, the angle is obtuse. The vectors point in opposite directions, indicating a *negative correlation*. When one asset performs well, the other tends to perform poorly.

Example 4.7 (Document Similarity)**.** In Natural Language Processing, documents can be represented as vectors of word counts. Let our vocabulary be: {“Data”, “Algorithm”, “Oven”, “Flour”}.

- **Document 1 (Tech Article)**: “Data and Algorithm...” $\to [1, 1, 0, 0]^T$
- **Document 2 (Recipe)**: “Oven and Flour...” $\to [0, 0, 1, 1]^T$

$$d_1 \cdot d_2 = (1)(0) + (1)(0) + (0)(1) + (0)(1) = 0.$$

Since the dot product is exactly 0, the vectors are *orthogonal.* Geometrically, they are at 90°. In a data context, this means the documents share *no common vocabulary* and are likely about completely unrelated topics.

4.2 A Geometric Link to Statistics: Covariance

One of the most elegant and powerful connections in this field is the link between the geometry of the dot product and the statistical concept of covariance. This bridge allows us to translate geometric intuition directly into statistical reasoning, and vice-versa.

To establish this link, we must first introduce the concept of *centering* the data. Consider two features from our dataset, represented by two column vectors $u, v \in \mathbb{R}^n$. Each vector contains the n observations for that feature.

Definition 4.8 (Mean Centering)**.** Let $u = (u_1, \dots, u_n)^T \in \mathbb{R}^n$ be a vector of observations for a single feature. The sample mean is $\bar{u} = \frac{1}{n}\sum_{i=1}^{n} u_i$. The *mean-centered vector*, denoted $\tilde{u}$, is obtained by subtracting the mean from each component:

$$\tilde{u} = u - \bar{u}\mathbf{1} = \begin{pmatrix} u_1 - \bar{u} \\ u_2 - \bar{u} \\ \vdots \\ u_n - \bar{u} \end{pmatrix}$$

where $\mathbf{1}$ is the n-dimensional vector of ones.

Remark 4.9. The components of a mean-centered vector sum to zero, i.e., $\sum_{i=1}^{n} \tilde{u}_i = 0$.

Example 4.10. Consider a feature vector $u \in \mathbb{R}^5$ representing a small dataset:

$$u = \begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \\ 10 \end{bmatrix}.$$

First, we calculate the sample mean $\bar{u}$:

$$\bar{u} = \frac{2 + 4 + 6 + 8 + 10}{5} = \frac{30}{5} = 6.$$

Next, we construct the mean-centered vector $\tilde{u}$ by subtracting 6 from every component:

$$\tilde{u} = \begin{bmatrix} 2-6 \\ 4-6 \\ 6-6 \\ 8-6 \\ 10-6 \end{bmatrix} = \begin{bmatrix} -4 \\ -2 \\ 0 \\ 2 \\ 4 \end{bmatrix}.$$

As expected by the definition, the sum of the components of $\tilde{u}$ is $(-4) + (-2) + 0 + 2 + 4 = 0$.

Example 4.11 (Normalizing User Bias in Recommendations)**.** In recommendation systems, different users have different rating scales (biases). Some users are "easy graders" (average rating 4.5/5), while others are "harsh critics" (average rating 2.5/5). Mean centering removes this bias.

Suppose a user rates four movies on a scale of 1 to 5

$$u_{\text{ratings}} = \begin{bmatrix} 5 \\ 5 \\ 3 \\ 3 \end{bmatrix}.$$

The user's average rating is $\bar{u} = 4$. The mean-centered vector is

$$\tilde{u}_{\text{centered}} = \begin{bmatrix} 5-4 \\ 5-4 \\ 3-4 \\ 3-4 \end{bmatrix} = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \end{bmatrix}.$$

In the centered vector, positive values (+1) now clearly indicate movies the user liked *more than usual*, and negative values (-1) indicate movies they liked *less than usual*, regardless of their baseline strictness.

Before defining covariance formally, consider the geometric relationship between two features. If we wish to understand if two variables are related (e.g., "Do people with higher BMIs tend to have higher insurance charges?"), we first look at how they move *relative to their averages.*

Imagine drawing a scatter plot of the data and placing a new set of axes centered exactly at the sample means $(\bar{u}, \bar{v})$, see Figure 5. This divides our data into four quadrants:

- **Top-Right**: Observations where both features are above average. Here, $(u_i - \bar{u}) > 0$ and $(v_i - \bar{v}) > 0$, so their product is positive.
- **Bottom-Left**: Observations where both features are below average. Here, both deviations are negative, but their product (negative $\times$ negative) is still positive.
- **Top-Left & Bottom-Right**: Observations where one feature is above average and the other is below. In these regions, the product of deviations is negative.

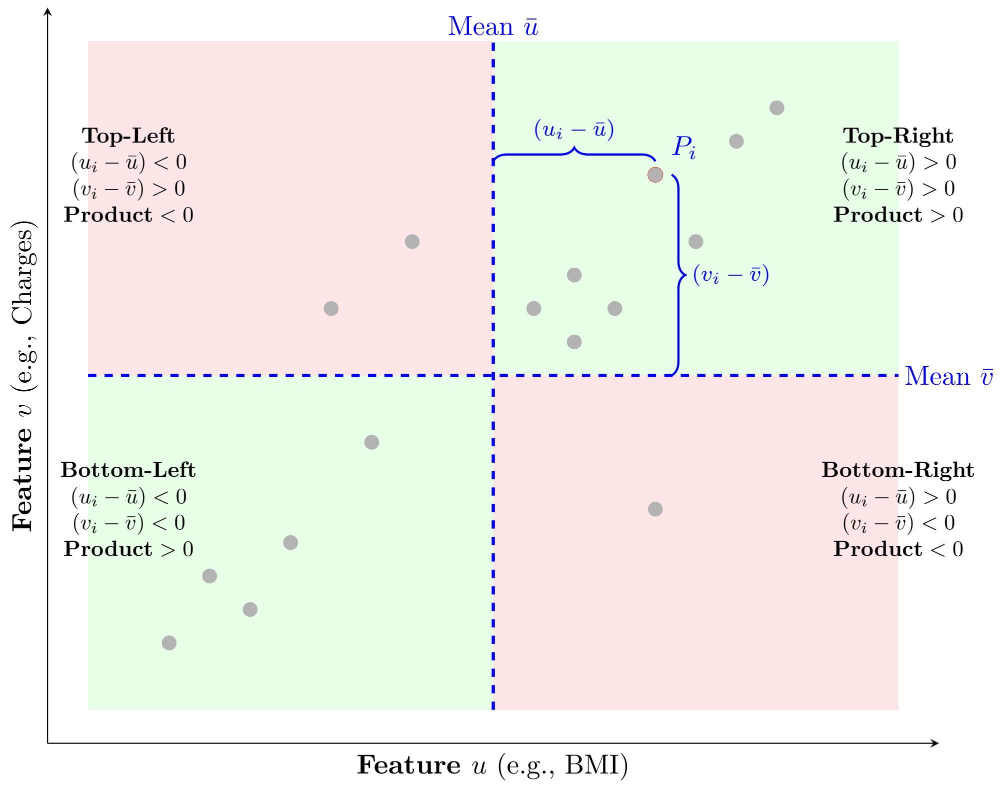
Covariance is essentially the *average* of these products. If the majority of data points lie in the Top-Right and Bottom-Left quadrants, the positive products outweigh the negative ones, resulting in a positive covariance (a positive trend). If the data is scattered randomly across all four quadrants, the positive and negative products cancel each other out, resulting in a covariance near zero.

Definition 4.12 (Sample Covariance)**.** Given two features represented by vectors $u = (u_1, \dots, u_n)^T$ and $v = (v_1, \dots, v_n)^T$ in $\mathbb{R}^n$, the **sample covariance** measures the joint variability of these two variables. It is defined as:

$$\text{Cov}(u, v) = \frac{1}{n-1} \sum_{i=1}^{n} (u_i - \bar{u})(v_i - \bar{v})$$

where $\bar{u}$ and $\bar{v}$ are the sample means of u and v, respectively.

Remark 4.13. The term $\frac{1}{n-1}$ is used (instead of $\frac{1}{n}$) to provide an unbiased estimate of the population covariance.



Mean $\bar{u}$
Mean $\bar{v}$
$(u_i - \bar{u})$
P_i
$(v_i - \bar{v})$
Top-Left $(u_i - \bar{u}) < 0$ $(v_i - \bar{v}) > 0$ **Product** < 0
Top-Right $(u_i - \bar{u}) > 0$ $(v_i - \bar{v}) > 0$ **Product** > 0
Bottom-Left $(u_i - \bar{u}) < 0$ $(v_i - \bar{v}) < 0$ **Product** > 0
Bottom-Right $(u_i - \bar{u}) > 0$ $(v_i - \bar{v}) < 0$ **Product** < 0
Feature u (e.g., BMI)
Feature v (e.g., Charges)

Figure 5: Geometric view of Covariance. Points in the green quadrants contribute positively to the sum, while points in the red quadrants contribute negatively. A strong positive trend exists when most points inhabit the green regions.

Example 4.14. Consider a small dataset with $n = 3$ observations. Let $u = [1, 2, 3]^T$ and $v = [2, 4, 6]^T$.

Compute Means: $\bar{u} = 2$ and $\bar{v} = 4$.

Compute Deviations:

- $u - \bar{u}\mathbf{1} = [-1, 0, 1]^T$
- $v - \bar{v}\mathbf{1} = [-2, 0, 2]^T$

Sum of Products:

$$\sum_{i=1}^{3}(u_i - \bar{u})(v_i - \bar{v}) = (-1)(-2) + (0)(0) + (1)(2) = 2 + 0 + 2 = 4$$

Normalize:

$$\text{Cov}(u, v) = \frac{1}{3-1}(4) = \frac{4}{2} = 2$$

The positive result confirms that as u increases, v also increases.

Example 4.15. Consider a logistics company tracking delivery trucks. Let u be the average speed of the truck and v be the time taken to reach the destination.

- When a driver travels faster than average ($u_i - \bar{u} > 0$), the time taken is less than average ($v_i - \bar{v} < 0$). The product of deviations is negative.
- When a driver is stuck in traffic and moves slower than average ($u_i - \bar{u} < 0$), the time taken is greater than average ($v_i - \bar{v} > 0$). The product of deviations is, again, negative.

Since the terms in the summation are consistently negative, the final $\text{Cov}(u, v)$ will be negative. This mathematically quantifies the inverse relationship: as one feature increases, the other tends to decrease.

With the definitions of the dot product and mean centering established, we arrive at a crucial insight that the statistical formula for covariance is structurally identical to the algebraic dot product.

Recall that the numerator of the sample covariance is $\sum(u_i - \bar{u})(v_i - \bar{v})$. If we view the centered features $\tilde{u}$ and $\tilde{v}$ as vectors, this summation is exactly the definition of their dot product. This realization allows us to compress the complex summation formula into a concise vector operation, implying that covariance is nothing more than a measure of how much two feature vectors align in the centered feature space.

Theorem 4.16 (Covariance as a Dot Product)**.** *Let $u, v \in \mathbb{R}^n$ be two vectors representing the values of two features across n observations. Let $\tilde{u}$ and $\tilde{v}$ be the corresponding mean-centered vectors. The sample covariance between the two features is given by the scaled dot product of their centered vectors:*

$$\text{Cov}(u, v) = \frac{1}{n-1}(\tilde{u} \cdot \tilde{v}).$$

Proof. By the standard statistical definition, the sample covariance between two variables u and v is the sum of the product of their deviations from the mean, normalized by $n-1$:

$$\mathrm{Cov}(u,v) = \frac{1}{n-1}\sum_{i=1}^{n}(u_i - \bar{u})(v_i - \bar{v})$$

Recall the definition of the mean-centered vectors $\tilde{u}$ and $\tilde{v}$. Their i-th components are defined explicitly as the deviations from the mean:

$$\tilde{u}_i = u_i - \bar{u} \quad \text{and} \quad \tilde{v}_i = v_i - \bar{v}$$

Now, consider the algebraic definition of the dot product for the vectors $\tilde{u}$ and $\tilde{v}$:

$$\begin{aligned}\tilde{u}\cdot\tilde{v} &= \sum_{i=1}^{n}\tilde{u}_i\tilde{v}_i \\ &= \sum_{i=1}^{n}(u_i - \bar{u})(v_i - \bar{v})\end{aligned}$$

Substituting this dot product result back into the covariance formula (1), we obtain:

$$\mathrm{Cov}(u,v) = \frac{1}{n-1}\underbrace{\sum_{i=1}^{n}(u_i - \bar{u})(v_i - \bar{v})}_{\tilde{u}\cdot\tilde{v}} = \frac{1}{n-1}(\tilde{u}\cdot\tilde{v})$$

Thus, covariance is simply the inner product of centered feature vectors, scaled by the degrees of freedom. □

This theorem is not merely a computational convenience; it represents a deep conceptual unification. A statistical question—"To what degree do these two variables vary together?"—is mathematically identical to a geometric question—"What is the dot product (and thus, related to the angle) between these two centered vectors in $\mathbb{R}^n$?".

This unification has profound consequences. The statistical concept of two variables being *uncorrelated* ($\mathrm{Cov}(u,v) = 0$) is geometrically equivalent to their centered vectors being *orthogonal* ($\tilde{u}\cdot\tilde{v} = 0$). This allows us to apply our powerful geometric intuition about orthogonality to understand statistical independence. This principle is the cornerstone of dimensionality reduction techniques like principal component analysis (PCA), which seeks to find a new, orthogonal basis for the feature space where the features (the principal components) are uncorrelated. This is a purely geometric operation (a change of basis to the eigenvectors of the covariance matrix) that solves a fundamental statistical problem (finding directions of maximal variance).

5 Quantifying Distance and Magnitude: A Survey of Norms

While the dot product measures the relationship *between* vectors, norms measure a property of a *single* vector: its magnitude, length, or size. The choice of norm defines the geometry of our feature space, particularly how we measure distance between data points.

Recall from linear algebra the formal definition of a norm.

Definition 5.1 (Norm)**.** A **norm** on a vector space V over a field $\mathbb{F}$ (typically $\mathbb{R}$ or $\mathbb{C}$) is a function $\|\cdot\| : V \to \mathbb{R}$ that satisfies the following properties for all vectors $x, y \in V$ and any scalar $\alpha \in \mathbb{F}$:

1. $\|x\| \geq 0$ (Non-negativity)
2. $\|x\| = 0 \iff x = \mathbf{0}$ (Positive definiteness)
3. $\|\alpha x\| = |\alpha|\|x\|$ (Absolute homogeneity)
4. $\|x + y\| \leq \|x\| + \|y\|$ (Triangle inequality / Subadditivity)

A vector space equipped with a norm is called a normed vector space.

The most familiar and widely used norm is the Euclidean norm, or L_2 norm.

Example 5.2 (L_2 Norm)**.** For a vector $x \in \mathbb{R}^n$, the L_2 norm, or Euclidean norm, is defined as the square root of the sum of the squared components:

$$\|x\|_2 = \sqrt{\sum_{j=1}^{n} x_j^2} = \sqrt{x^T x}.$$

To prove that $\|\cdot\|_2$ is a norm, we must verify the four defining properties for all vectors $x, y \in \mathbb{R}^n$ and any scalar $\alpha \in \mathbb{R}$.

Non-negativity: For any $x \in \mathbb{R}^n$, each component squared x_j^2 is a real number ≥ 0. The sum of non-negative numbers is non-negative, and the square root of a non-negative number is non-negative, i.e.,

$$\|x\|_2 = \sqrt{\sum_{j=1}^{n} x_j^2} \geq 0.$$

Positive Definiteness: We need to show $\|x\|_2 = 0 \iff x = \mathbf{0}$.

- ($\Leftarrow$) If $x = \mathbf{0}$, then $x_j = 0$ for all j. Thus, $\|x\|_2 = \sqrt{\sum 0^2} = 0$.
- ($\Rightarrow$) If $\|x\|_2 = 0$, then $\sqrt{\sum x_j^2} = 0$, which implies $\sum_{j=1}^{p} x_j^2 = 0$. Since squares of real numbers are non-negative, a sum of non-negative terms is zero if and only if every individual term is zero. Therefore, $x_j^2 = 0 \implies x_j = 0$ for all j, so $x = \mathbf{0}$.

Absolute Homogeneity: For any scalar $\alpha \in \mathbb{R}$, we have

$$\|\alpha x\|_2 = \sqrt{\sum_{j=1}^{n} (\alpha x_j)^2} = \sqrt{\sum_{j=1}^{n} \alpha^2 x_j^2} = \sqrt{\alpha^2 \sum_{j=1}^{p} x_j^2}.$$

Using the property $\sqrt{ab} = \sqrt{a}\sqrt{b}$ (for non-negative a, b) and $\sqrt{\alpha^2} = |\alpha|$, it follows that

$$\|\alpha x\|_2 = |\alpha| \sqrt{\sum_{j=1}^{n} x_j^2} = |\alpha| \|x\|_2.$$

Triangle Inequality: We must show $\|x+y\|_2 \leq \|x\|_2 + \|y\|_2$. We start by squaring the left side as follows.

$$\begin{aligned}\|x+y\|_2^2 &= (x+y)\cdot(x+y)\\ &= x\cdot x + 2(x\cdot y) + y\cdot y\\ &= \|x\|_2^2 + 2(x\cdot y) + \|y\|_2^2.\end{aligned}$$

By the *Cauchy-Schwarz inequality*, we know that $x\cdot y \leq |x\cdot y| \leq \|x\|_2\|y\|_2$. Substituting this upper bound for the middle term yields

$$\begin{aligned}\|x+y\|_2^2 &\leq \|x\|_2^2 + 2\|x\|_2\|y\|_2 + \|y\|_2^2\\ &= (\|x\|_2 + \|y\|_2)^2.\end{aligned}$$

Taking the square root of both sides (which preserves the inequality since both sides are non-negative) yields

$$\|x+y\|_2 \leq \|x\|_2 + \|y\|_2.$$

Definition 5.3 (Euclidean Distance)**.** The Euclidean distance between two vectors x and y is given by the norm of their difference:

$$d_2(x,y) = \|x-y\|_2 = \sqrt{\sum_{j=1}^{n}(x_j - y_j)^2}.$$

Remark 5.4 (Geometric Interpretation). The L_2 norm corresponds to our intuitive notion of distance in physical space, the *as the crow flies* or straight-line distance. It is a direct generalization of the Pythagorean theorem to p dimensions. In data science, it is the default measure of dissimilarity for many algorithms, including k-Nearest Neighbors (kNN) and k-Means clustering. Its mathematical properties, such as being rotationally invariant and having a smooth, easily computable derivative (for its square, $\|x\|_2^2 = x^T x$), make it highly suitable for optimization algorithms like gradient descent that are used to train machine learning models.

Example 5.5. In signal processing, the L_2 norm is often used to quantify the *energy* or strength of a discrete signal. Consider a short audio signal represented as a vector x with 3 time-steps:

$$x = \begin{bmatrix} 3 \\ -4 \\ 12 \end{bmatrix}$$

The *loudness* or magnitude of this signal is given by its Euclidean norm

$$\|x\|_2 = \sqrt{3^2 + (-4)^2 + 12^2} = \sqrt{9+16+144} = \sqrt{169} = 13.$$

If we were to normalize this signal (scale it to unit energy), we would divide the vector by this norm $\hat{x} = (1/13)\cdot x \approx [0.23, -0.31, 0.92]^T$.

Example 5.6. In a machine learning classification task, we often calculate the distance between an unknown observation and labeled training data to determine its class. Suppose we are classifying fruit based on two features: x_1 = Weight (g) and x_2 = Redness Index (1-10).

- **Unknown Fruit** (u): Weight = 150g, Redness = 8. $\rightarrow u = [150, 8]^T$.

- **Apple Prototype (a)**: Weight = 140g, Redness = 9. $\rightarrow a = [140, 9]^T$.
- **Banana Prototype (b)**: Weight = 100g, Redness = 1. $\rightarrow b = [100, 1]^T$.

We calculate the Euclidean distance from the unknown fruit to each prototype as follows.

$$\begin{aligned} d_2(u, a) &= \|u - a\|_2 \\ &= \sqrt{(150 - 140)^2 + (8 - 9)^2} \\ &= \sqrt{(10)^2 + (-1)^2} \\ &= \sqrt{101} \approx 10.05. \end{aligned}$$

$$\begin{aligned} d_2(u, b) &= \|u - b\|_2 \\ &= \sqrt{(150 - 100)^2 + (8 - 1)^2} \\ &= \sqrt{(50)^2 + (7)^2} \\ &= \sqrt{2500 + 49} \\ &= \sqrt{2549} \approx 50.49. \end{aligned}$$

Since $d_2(u, a) < d_2(u, b)$, the algorithm classifies the unknown fruit as an Apple.

An alternative and equally important measure of magnitude is the L_1 norm.

Example 5.7 (L_1 Norm)**.** For a vector $x \in \mathbb{R}^n$, the L_1 norm, also known as the Manhattan or Taxicab norm, is defined as

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|.$$

To establish that $\|\cdot\|_1$ is a norm, we verify the four defining axioms for arbitrary vectors $x, y \in \mathbb{R}^n$ and scalar $\alpha \in \mathbb{R}$.

Non-negativity: For every component x_i, the absolute value $|x_i|$ is by definition non-negative ($|x_i| \geq 0$). The sum of non-negative real numbers is always non-negative.

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \geq 0.$$

Positive Definiteness: We must show that $\|x\|_1 = 0$ if and only if $x = \mathbf{0}$.

- ($\Leftarrow$) If $x = \mathbf{0}$, then $x_i = 0$ for all $i = 1, \ldots, p$. Thus, $\|x\|_1 = \sum_{i=1}^{n} |0| = 0$.
- ($\Rightarrow$) Conversely, assume $\|x\|_1 = \sum_{i=1}^{n} |x_i| = 0$. Since each term $|x_i| \geq 0$, a sum of non-negative terms can only equal zero if *every individual term* is zero. Therefore, $|x_i| = 0$ implies $x_i = 0$ for all i. Consequently, the vector x must be the zero vector $\mathbf{0}$.

Absolute Homogeneity: For any scalar $\alpha \in \mathbb{R}$:

$$\begin{aligned}
\|\alpha x\|_1 &= \sum_{i=1}^{n} |\alpha x_i| \\
&= \sum_{i=1}^{n} |\alpha||x_i| \quad (\text{since } |ab| = |a||b|) \\
&= |\alpha| \sum_{i=1}^{n} |x_i| \quad (\text{factoring out the constant } |\alpha|) \\
&= |\alpha|\|x\|_1.
\end{aligned}$$

Triangle Inequality: We must show that $\|x+y\|_1 \leq \|x\|_1 + \|y\|_1$. Recall the standard triangle inequality for real numbers: for any $a, b \in \mathbb{R}$, $|a+b| \leq |a| + |b|$. applying this to each component i:

$$|x_i + y_i| \leq |x_i| + |y_i|.$$

Summing this inequality over all dimensions $i = 1$ to p yields

$$\begin{aligned}
\|x+y\|_1 &= \sum_{i=1}^{n} |x_i + y_i| \\
&\leq \sum_{i=1}^{n} (|x_i| + |y_i|) \\
&= \sum_{i=1}^{n} |x_i| + \sum_{i=1}^{n} |y_i| \\
&= \|x\|_1 + \|y\|_1.
\end{aligned}$$

Thus, the triangle inequality holds.

Definition 5.8 (Manhattan distance)**.** The Manhattan distance between two vectors x and y is $d_1(x, y) = \|x - y\|_1$.

Remark 5.9 (Geometric Interpretation)*.* The L_1 norm measures distance by summing the absolute differences along each coordinate axis. The name *Manhattan distance* provides the intuition: it is the distance a taxi would travel in a city grid, where movement is restricted to north-south and east-west blocks. Figure 6 illustrates the distance between two points $x = (1, 1)$ and $y = (4, 3)$ on a grid. Unlike the Euclidean distance (gray dashed line), which represents the direct path as the crow flies, the Manhattan distance (red solid line) is constrained to horizontal and vertical movements along the axes. The total distance is the sum of the absolute differences in coordinates ($|x_1 - y_1| + |x_2 - y_2| = 3 + 2 = 5$), demonstrating why this metric is often referred to as *taxicab geometry.*

In data science, the L_1 norm has two critical properties. First, it is more robust to outliers than the L_2 norm because it does not square the differences, giving large errors less weight. Second, and most importantly, it has a tendency to promote *sparsity*, which means it encourages solutions where many components are exactly zero. This property is fundamental to a technique called Lasso regression, which performs simultaneous model fitting and feature selection.

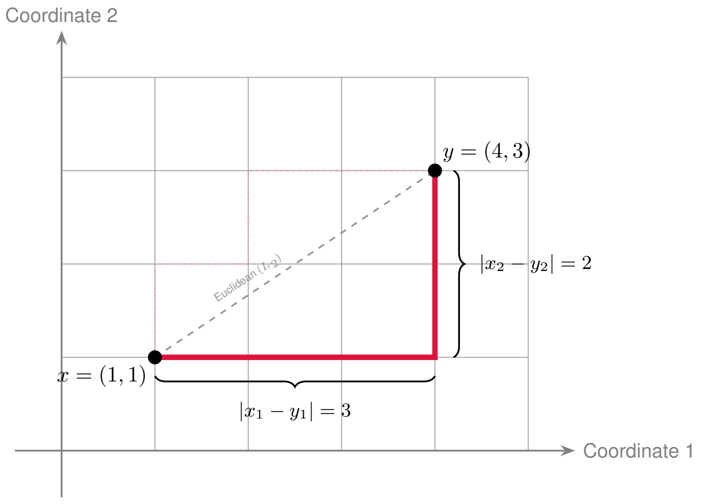


Figure 6: Geometric interpretation of the Manhattan (L_1) distance

Example 5.10. In machine learning, the L_1 norm is used as a regularization term (Lasso) to force the weights of irrelevant features to become exactly zero, effectively performing feature selection. Consider a model predicting house prices with a weight vector w, corresponding to three features [Square Footage, Number of Bedrooms, Owner's Zodiac Sign],

$$w = \begin{bmatrix} 50 \\ 20 \\ -0.05 \end{bmatrix}.$$

The *cost* added to the model by the L_1 norm is

$$\|w\|_1 = |50| + |20| + |-0.05| = 50 + 20 + 0.05 = 70.05.$$

Unlike the L_2 norm (which squares the 0.05, making it tiny), the L_1 norm applies a constant pressure that eventually drives the irrelevant "Zodiac Sign" weight (0.05) all the way to 0, simplifying the model.

Example 5.11. Robots in a warehouse move along a grid of aisles and shelves. They cannot move diagonally through the shelving units; they must move horizontally and vertically. Let the warehouse floor be a 2D grid.

- **Robot Position (r)**: Aisle 2, Bay 5. $\to r = [2,5]^T$
- **Target Item (t)**: Aisle 6, Bay 1. $\to t = [6,1]^T$

The Euclidean distance would assume a straight line through the shelves. The practical travel distance is the Manhattan distance

$$\begin{aligned} d_1(r,t) &= \|r - t\|_1 \\ &= |2-6| + |5-1| \\ &= |-4| + |4| \\ &= 4 + 4 = 8. \end{aligned}$$

The robot must travel 4 units east and 4 units south, for a total of 8 units.

The distinct properties of the L_1 and L_2 norms are best understood by visualizing their corresponding **unit balls**. The unit ball for a given norm is the set of all vectors whose norm is less than or equal to 1.

Definition 5.12 (Unit Ball)**.** The unit ball for a norm $\|\cdot\|$ in a vector space V is the set

$$B = \{x \in V : \|x\| \leq 1\}.$$

In $\mathbb{R}^2$, the unit balls for the L_2 and L_1 norms have strikingly different shapes, which explains their different behaviors in optimization.

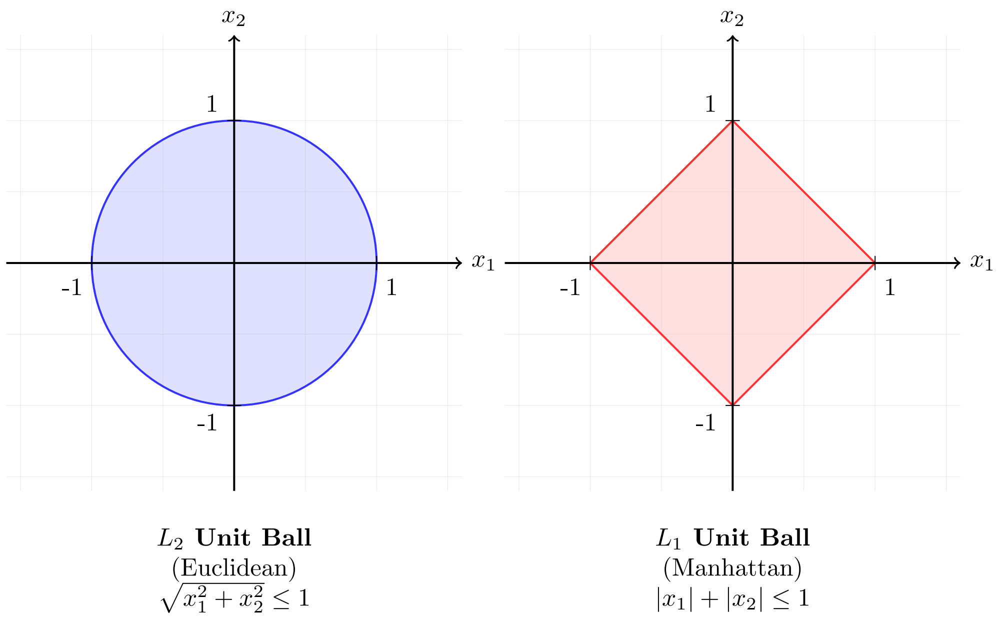


Figure 7: A comparison of the unit balls in $\mathbb{R}^2$. Left: The L_2 unit ball, defined by $\sqrt{x_1^2 + x_2^2} \leq 1$, is a circle. Right: The L_1 unit ball, defined by $|x_1| + |x_2| \leq 1$, is a diamond (a rotated square).

The geometric reason for the sparsity-inducing property of the L_1 norm can now be made clear. Many machine learning problems, particularly regularized regression which we will see later, can be framed as an optimization problem of the form:

$$\min_{\beta} \text{Error}(\beta) \quad \text{subject to} \quad \|\beta\|_p \leq t$$

This means we are searching for a parameter vector β that minimizes some error function (e.g., the sum of squared residuals) but is constrained to lie within a certain *budget* defined by its p-norm being less than some threshold t.

Geometrically, this is equivalent to finding the first point where the level sets of the error function (which are typically ellipses) expand from the unconstrained minimum to touch the boundary of the norm ball.

- With the L_2 norm, the constraint region is a circle (or a hypersphere in higher dimensions). Because this shape is smooth and rotationally symmetric, the point of tangency between an expanding ellipse and the circle can occur anywhere on its boundary. It is statistically very unlikely that this point will lie exactly on a coordinate axis (where one component of β would be zero).

- With the L_1 norm, the constraint region is a diamond (or a cross-polytope in higher dimensions). This shape has sharp *corners* or vertices that lie precisely on the coordinate axes. As the elliptical level sets of the error function expand, they are geometrically far more likely to make first contact with one of these corners than with one of the flat edges between them.

A solution that lies at a corner on an axis means that the coordinate corresponding to the other axis is zero. In a p-dimensional feature space, the L_1 ball has vertices where only one coordinate is non-zero, and edges/faces where many coordinates are zero. Therefore, minimizing an error function subject to an L_1 constraint naturally drives many components of the solution vector β to be exactly zero. This effect is sparsity, and it is a powerful form of automatic feature selection built directly into the optimization problem.

Example 5.13. The geometric difference between the unit balls implies that the L_2 ball covers more area than the L_1 ball. Let's verify this with a specific point. Consider the vector $v = [0.6, 0.6]^T$.

- **L_2 Norm Check:**
$$\|v\|_2 = \sqrt{0.6^2 + 0.6^2} = \sqrt{0.36 + 0.36} = \sqrt{0.72} \approx 0.85.$$
Since $0.85 \leq 1$, the point lies **inside** the L_2 unit ball (the circle).

- **L_1 Norm Check:**
$$\|v\|_1 = |0.6| + |0.6| = 1.2.$$
Since $1.2 > 1$, the point lies **outside** the L_1 unit ball (the diamond).

This numerical discrepancy explains why the diamond shape has flat edges and corners that are *pulled in* closer to the origin compared to the circle, creating the geometry necessary for the sparsity effect.

Example 5.14. Let us construct a simplified optimization problem to see how the L_1 norm forces a coordinate to zero. Suppose we want to maximize the function $f(x, y) = x + 0.5y$ (represented by linear level sets) subject to a unit norm constraint.

1. Subject to L_2 constraint ($\sqrt{x^2 + y^2} \leq 1$): The maximum occurs where the gradient vector $[1, 0.5]^T$ aligns with the radius. Normalizing this vector:
$$x^* = \frac{1}{\sqrt{1^2 + 0.5^2}} \approx 0.89, \quad y^* = \frac{0.5}{\sqrt{1^2 + 0.5^2}} \approx 0.45.$$
Result: Dense solution. Both x and y are non-zero $(0.89, 0.45)$.

2. Subject to L_1 constraint ($|x| + |y| \leq 1$): We evaluate the function at the vertices (corners) of the diamond:

- Corner $(1, 0) \to f(1, 0) = 1 + 0 = 1.0$
- Corner $(0, 1) \to f(0, 1) = 0 + 0.5 = 0.5$

Result: Sparse solution. The maximum is at $(1, 0)$, where the y component is forced exactly to zero.

Example 5.15. In a housing price prediction model, suppose we have two features:

- β_1: Square Footage (Highly relevant).

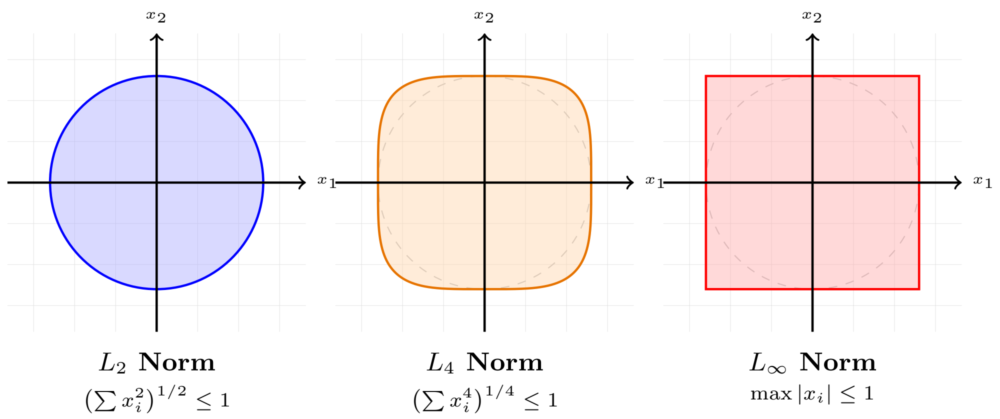


Figure 8: Evolution of the unit ball for $p = 2$ (Euclidean), $p = 4$, and $p = \infty$ (Max). As p increases, the unit ball bulges outward from the unit circle, eventually filling the bounding box defined by the L_∞ norm.

- β_2: The owner's favorite number (Completely irrelevant noise).

We want to minimize error while keeping the coefficients small (regularization).

- L_2 Penalty: The circular constraint effectively shrinks coefficients evenly. It might produce a result like $\hat{\beta} = [50.2, 0.15]^T$. It lowers the impact of the noise (β_2), but keeps it non-zero.
- L_1 Penalty: Because the elliptical error contours touch the *corner* of the L_1 diamond first, the optimization is likely to produce $\hat{\beta} = [50.3, 0.0]^T$.

The L_1 norm has performed *automatic feature selection* by identifying that β_2 contributes nothing and setting it to exactly zero.

The L_1 and L_2 norms are special cases of a broader family of norms.

Example 5.16 (L_p Norm)**.** For a real number $p \geq 1$, the L_p norm of a vector $x \in \mathbb{R}^n$ is defined as

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}.$$

Example 5.17 (L_∞ Norm)**.** The L_∞ norm (or max norm) of a vector $x \in \mathbb{R}^n$ is defined as the limit of the L_p norm as $p \to \infty$, which simplifies to

$$\|x\|_\infty = \max_{i=1,\dots,p} |x_i|.$$

The L_∞ norm is useful in applications where the goal is to minimize the maximum error, or where one is concerned only with the single largest-magnitude feature. The unit ball for the L_∞ norm in $\mathbb{R}^2$ is a square aligned with the axes, see Figure 8.

6 Conclusion: Setting the Stage for Modeling

6.1 Summary of Key Concepts

This lecture has established the foundational perspective for the entire course. We have made three fundamental conceptual shifts:

1. **Data as Geometry:** We have moved from viewing data as a table of numbers to seeing it as a collection of vectors residing in a high-dimensional feature space, $\mathbb{R}^p$. This collection is compactly represented by the data matrix, X.

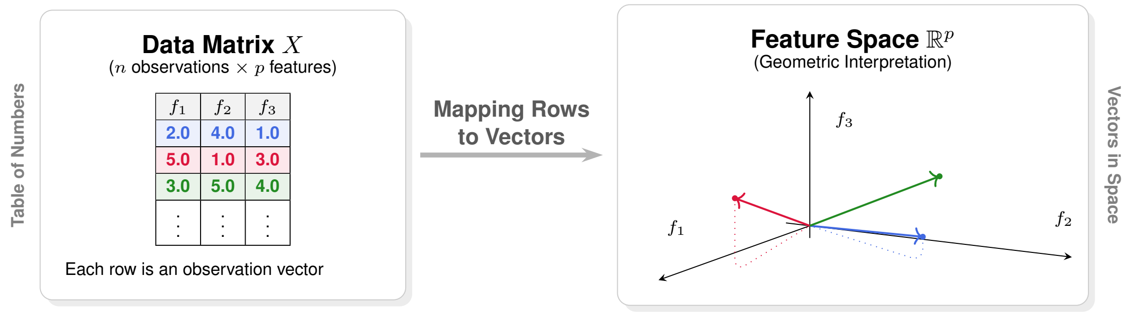
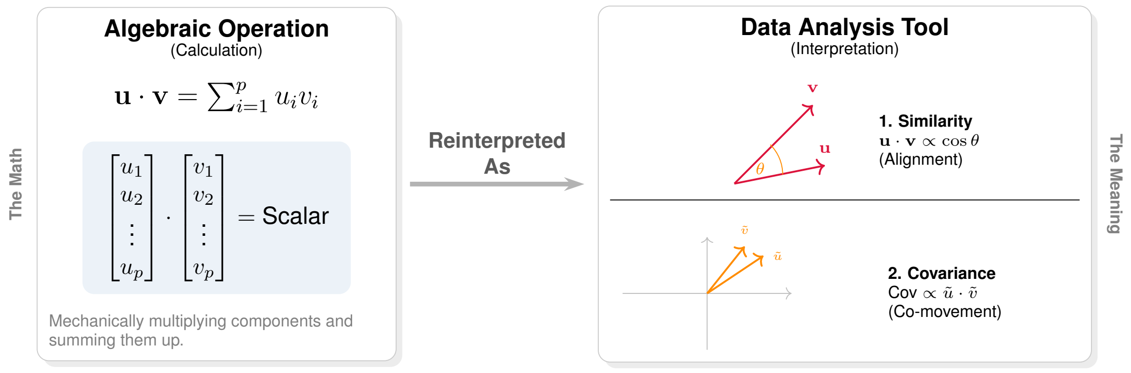


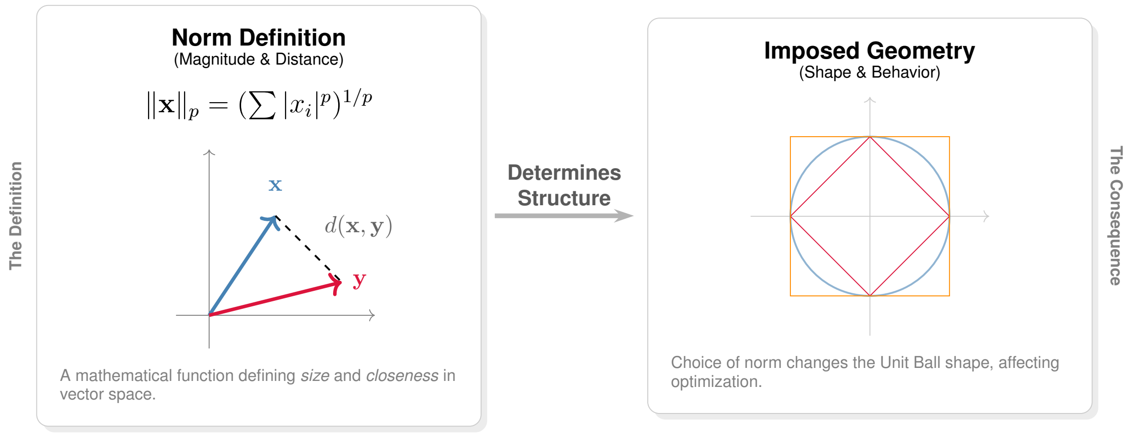
Table of Numbers
Data Matrix X
(n observations $\times$ p features)
Each row is an observation vector
Mapping Rows to Vectors
Feature Space $\mathbb{R}^p$
(Geometric Interpretation)
f_1
f_2
f_3
Vectors in Space

f_1	f_2	f_3
2.0	4.0	1.0
5.0	1.0	3.0
3.0	5.0	4.0
⋮	⋮	⋮

2. **Operations as Insights:** We have reinterpreted fundamental linear algebraic operations as data analysis tools. The dot product measures similarity and is deeply connected to statistical covariance.


The Math
Algebraic Operation
(Calculation)
$\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{p} u_i v_i$
$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_p \end{bmatrix} = \text{Scalar}$
Mechanically multiplying components and summing them up.
Reinterpreted As
Data Analysis Tool
(Interpretation)
$\mathbf{v}$
$\mathbf{u}$
θ
1. Similarity
$\mathbf{u} \cdot \mathbf{v} \propto \cos\theta$
(Alignment)
$\tilde{v}$
$\tilde{u}$
2. Covariance
Cov $\propto \tilde{u} \cdot \tilde{v}$
(Co-movement)
The Meaning

3. **Norms as Metrics:** We have understood vector norms as a way to define distance and magnitude within our feature space. The choice of norm (e.g., L_1 vs. L_2) imposes a specific geometry, with profound implications for algorithm behavior, such as the sparsity induced by the L_1 norm.


The Definition
Norm Definition
(Magnitude & Distance)
$\|\mathbf{x}\|_p = (\sum |x_i|^p)^{1/p}$
$\mathbf{x}$
$d(\mathbf{x}, \mathbf{y})$
$\mathbf{y}$
A mathematical function defining *size* and *closeness* in vector space.
Determines Structure
Imposed Geometry
(Shape & Behavior)
Choice of norm changes the Unit Ball shape, affecting optimization.
The Consequence

6.2 Bridge to the Workshop: Exploratory Data Analysis

These abstract geometric ideas have immediate, practical applications in the process of *exploratory data analysis* (EDA). When a data scientist creates a scatter plot of two features, they are performing a powerful geometric operation: they are creating an orthogonal projection of the p-dimensional data cloud onto a 2D subspace. This is a literal attempt to visualize the data's geometry. Similarly, when they compute summary statistics, they are quantifying geometric properties of this cloud. The mean vector, $\bar{x}$, is the cloud's *center of mass.* The covariance matrix, which we now understand is built from dot products of centered feature vectors, describes the cloud's shape, spread, and orientation. The workshop will focus on using these computational tools to explore the geometry we have discussed today.

6.3 Bridge to Week 2: Linear Regression

We conclude by posing the question that will motivate our next lecture, framed entirely in the geometric language we have just developed.

Suppose we have a data matrix of features $X \in \mathbb{R}^{n \times p}$ and a vector of outcomes $y \in \mathbb{R}^n$ that we wish to predict (e.g., the 'Charges' column). The goal of **linear regression** is to find a set of coefficients, a vector $\beta \in \mathbb{R}^p$, such that a linear combination of the features provides the best possible prediction of y. Our prediction, $\hat{y}$, is given by:

$$\hat{y} = X\beta = \beta_1 \cdot (\text{col}_1 \text{ of } X) + \cdots + \beta_p \cdot (\text{col}_p \text{ of } X)$$

Notice that this equation defines $\hat{y}$ as a linear combination of the columns of X. Therefore, by definition, $\hat{y}$ must lie in the column space of X, denoted $C(X)$, which is a subspace of $\mathbb{R}^n$.

The problem is that the true outcome vector y is almost certainly *not* in $C(X)$. If it were, we could find a perfect solution. Since it is not, we have an overdetermined system $X\beta = y$ with no exact solution. This leads us to the central question of linear regression, which is now a purely geometric one:

What is the vector $\hat{y}$ in the column space of X that is closest to the true outcome vector y?

As your intuition from linear algebra suggests, the answer is the **orthogonal projection** of y onto the subspace $C(X)$. Next week, we will use this geometric insight to derive the solution to the linear regression problem by solving for this projection, which will lead us directly to the famous Normal Equations: $X^T X\beta = X^T y$. The geometric framework we have built today provides the exact foundation needed to understand and solve this fundamental problem in data science.

Exercise

1. Consider the following two software systems used by a credit card company. Based on the distinctions provided in the lecture (Primary Goal, System Type, Mathematical Basis), determine which system constitutes "Data Science" and which does not. Justify your answer.

   (a) **System A:** A database trigger that automatically declines any transaction exceeding \$5,000 if the user resides in a specific list of countries.

   (b) **System B:** A fraud detection engine that analyzes transaction history, location, and time-of-day to generate a "suspicion score" for every swipe, flagging those in the top 1% of scores for review.

2. You are analyzing a dataset of 3 used cars ($n = 3$). The features recorded are: Age (years), Mileage (thousands of miles), and Price (thousands of dollars).

   - Car 1: 2 years old, 15k miles, \$20k.
   - Car 2: 5 years old, 60k miles, \$12k.
   - Car 3: 10 years old, 120k miles, \$5k.

   (a) Construct the Data Matrix $X \in \mathbb{R}^{3\times 3}$.

   (b) Identify the vector $X_{:,2}$ and explain its semantic meaning in this context.

   (c) If we visualize this data as a point cloud, what is the dimensionality of the feature space?

3. Two readers, User A and User B, rate two book genres: *Sci-Fi* and *Romance*. Their ratings are centered around their personal averages (positive means liked, negative means disliked).

$$u_A = \begin{bmatrix} 4 \\ -2 \end{bmatrix}, \quad v_B = \begin{bmatrix} -3 \\ 5 \end{bmatrix}$$

   (a) Calculate the dot product $u_A \cdot v_B$.

   (b) Based on the sign of the result, are the tastes of User A and User B aligned, opposing, or unrelated?

   (c) Explain the geometric relationship (angle θ) between these two vectors.

4. Let x be a vector representing study hours and y be a vector representing exam scores for $n = 3$ students.

$$x = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}, \quad y = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$$

   (a) Compute the sample means $\bar{x}$ and $\bar{y}$.

   (b) Construct the mean-centered vectors $\tilde{x}$ and $\tilde{y}$.

   (c) Calculate the sample covariance $\mathrm{Cov}(x, y)$ using the dot product of the centered vectors (as derived in the lecture).

5. A delivery robot operates on a grid-based city map. Its current location is at coordinates $p = [2, 1]^T$ and the destination is $q = [5, 5]^T$.

   (a) Calculate the Euclidean distance (L_2) between p and q.

(b) Calculate the Manhattan distance (L_1) between p and q.

(c) Which metric accurately reflects the distance the robot must travel if it cannot move diagonally through city blocks?

6. Consider the feature vector $w = [3, -4, 0]^T$.

   (a) Calculate $\|w\|_1$, $\|w\|_2$, and $\|w\|_\infty$.

   (b) Which norm gives the largest value and which gives the smallest? Is this ordering consistent with the general properties of L_p norms?

7. Consider the point $v = [0.7, 0.6]^T$ in $\mathbb{R}^2$.

   (a) Determine if v lies inside the L_2 unit ball ($B_2 = \{x : \|x\|_2 \leq 1\}$).

   (b) Determine if v lies inside the L_1 unit ball ($B_1 = \{x : \|x\|_1 \leq 1\}$).

   (c) Explain how the shape of the L_1 unit ball facilitates feature selection (forcing coefficients to zero) in optimization problems compared to the L_2 ball.

8. In an image recognition task, you are working with 20×20 pixel grayscale images.

   (a) To apply linear algebra methods, these images are flattened. What is the dimension p of the resulting observation vector?

   (b) If you have a dataset of 500 such images, what are the dimensions of the Data Matrix X?

   (c) If the dot product between two flattened image vectors is exactly 0, what does this imply about the pixels in the two images (assuming pixel values are non-negative)?

9. Given two vectors $a = [1, 2, 3]^T$ and $b = [1, 0, -1]^T$:

   (a) Compute the Euclidean norms $\|a\|_2$ and $\|b\|_2$.

   (b) Compute the dot product $a \cdot b$.

   (c) Using the geometric interpretation of the dot product, calculate the cosine of the angle θ between a and b.

   (d) Is the angle acute, obtuse, or orthogonal?

10. Suppose you have two centered feature vectors $\tilde{u}$ and $\tilde{v}$ such that $\tilde{u} \cdot \tilde{v} = 0$.

    (a) What is the sample covariance $\text{Cov}(u, v)$?

    (b) What statistical relationship corresponds to the geometric concept of orthogonality in the centered feature space?

Solution to exercise

1. **Classifying Data Science Systems**

   (a) **System A is NOT Data Science.** It is a deterministic system executing a known rule (Boolean logic). There is no probabilistic modeling or extraction of unknown insights; the outcome is fixed and certain based on the input.

   (b) **System B IS Data Science.** It models a stochastic system (fraud is uncertain and complex). It uses statistical inference to assign a probability (suspicion score) and extracts insight from historical patterns rather than following a hard-coded rule.

2. **The Data Matrix and Feature Space**

   (a) The matrix is formed by stacking the transposes of the observations:

$$X = \begin{bmatrix} 2 & 15 & 20 \\ 5 & 60 & 12 \\ 10 & 120 & 5 \end{bmatrix}$$

   (b) The vector $X_{:,2} = [15, 60, 120]^T$. This column vector represents the **Mileage** feature for the entire collection of cars.

   (c) The feature space is $\mathbb{R}^3$, corresponding to the 3 features (Age, Mileage, Price).

3. **Dot Product as Similarity**

   (a) $u_A \cdot v_B = (4)(-3) + (-2)(5) = -12 - 10 = -22$.

   (b) The tastes are **opposing**. The negative result indicates a negative correlation (User A likes what User B dislikes).

   (c) Since the dot product is negative, $\cos\theta < 0$, meaning the angle θ is **obtuse** (between 90° and 180°).

4. **Covariance Calculation**

   (a) Means: $\bar{x} = \frac{1+3+5}{3} = 3$. $\quad \bar{y} = \frac{2+4+6}{3} = 4$.

   (b) Mean-centered vectors:

$$\tilde{x} = \begin{bmatrix} 1-3 \\ 3-3 \\ 5-3 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 2 \end{bmatrix}, \quad \tilde{y} = \begin{bmatrix} 2-4 \\ 4-4 \\ 6-4 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 2 \end{bmatrix}$$

   (c) Using the formula $\text{Cov}(x, y) = \frac{1}{n-1}(\tilde{x} \cdot \tilde{y})$ with $n = 3$:

$$\tilde{x} \cdot \tilde{y} = (-2)(-2) + (0)(0) + (2)(2) = 4 + 0 + 4 = 8$$

$$\text{Cov}(x, y) = \frac{8}{3-1} = \frac{8}{2} = 4$$

5. **Distance Metrics (L1 vs L2)**

   (a) Euclidean (L_2): $\sqrt{(5-2)^2 + (5-1)^2} = \sqrt{3^2 + 4^2} = \sqrt{9 + 16} = \sqrt{25} = 5$.

   (b) Manhattan (L_1): $|5-2| + |5-1| = |3| + |4| = 3 + 4 = 7$.

(c) The **Manhattan distance (L_1)** accurately reflects the travel distance for a grid-constrained robot.

6. **Comparing Norms**

   (a) $L_1 = |3| + |-4| + |0| = 7$.
   $L_2 = \sqrt{3^2 + (-4)^2 + 0^2} = \sqrt{25} = 5$.
   $L_\infty = \max(|3|, |-4|, |0|) = 4$.

   (b) The L_1 norm is the largest (7) and the L_∞ norm is the smallest (4). Yes, this is consistent with the general property $\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1$.

7. **The Geometry of Sparsity**

   (a) $\|v\|_2 = \sqrt{0.7^2 + 0.6^2} = \sqrt{0.49 + 0.36} = \sqrt{0.85} \approx 0.92$. Since $0.92 \leq 1$, v is **inside** the L_2 ball.

   (b) $\|v\|_1 = |0.7| + |0.6| = 1.3$. Since $1.3 > 1$, v is **outside** the L_1 ball.

   (c) The L_1 ball is a polytope (diamond in 2D) with "corners" on the axes. In optimization, the error contours are statistically more likely to touch these corners first, where one coordinate is exactly zero, thus inducing sparsity. The L_2 ball is smooth, so solutions rarely land on an axis.

8. **High-Dimensional Operations**

   (a) $p = 20 \times 20 = 400$ dimensions.

   (b) $X \in \mathbb{R}^{500\times400}$ (500 rows for images, 400 columns for pixels).

   (c) Since pixel values are non-negative, a dot product of 0 implies that for every position where one image has a non-zero pixel, the other implies must have a zero (black) pixel. They share **no overlapping active pixels**.

9. **Calculating Angles in High Dimensions**

   (a) $\|a\|_2 = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{1 + 4 + 9} = \sqrt{14} \approx 3.74$.
   $\|b\|_2 = \sqrt{1^2 + 0^2 + (-1)^2} = \sqrt{1 + 0 + 1} = \sqrt{2} \approx 1.41$.

   (b) $a \cdot b = (1)(1) + (2)(0) + (3)(-1) = 1 + 0 - 3 = -2$.

   (c) Using $\cos\theta = \dfrac{a \cdot b}{\|a\|_2\|b\|_2}$:

   $$\cos\theta = \frac{-2}{\sqrt{14}\sqrt{2}} = \frac{-2}{\sqrt{28}} = \frac{-2}{2\sqrt{7}} = \frac{-1}{\sqrt{7}} \approx -0.378$$

   (d) Since $\cos\theta < 0$ and $\cos\theta \neq -1$, the angle is **obtuse**.

10. **Orthogonality and Independence**

    (a) Since $\mathrm{Cov}(u, v) \propto (\tilde{u} \cdot \tilde{v})$, if the dot product is 0, the covariance is $\mathbf{0}$.

    (b) Orthogonality in the centered feature space corresponds to the features being **uncorrelated** (linearly independent).