EE951: Linear Algebra **eMasters**

Chapter 4: Matrices

Author: Ketan Rajawat

Disclaimer: *These notes are taken from various sources, sometimes without proper citation. Do not distribute outside the class or upload on the Internet.*

Contents

4.1 Introduction

A matrix is a rectangular array of numbers arranged in rows and columns. It is denoted by a capital letter and its dimensions are specified by the number of rows and columns it contains. Let's consider an example of a 3x4 matrix:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 3 & 0 \\ 5 & 2 & 1 & 4 \\ -3 & 0 & 2 & 6 \end{bmatrix}$$

Here, the matrix has dimensions of 3×4, indicating it has three rows and four columns. Each individual number in the matrix is called an entry. In matrix notation, the entry at the i-th row and j-th column is denoted as $A_{i,j}$. For example, $A_{2,3}$ corresponds to the entry in the second row and third column of the matrix. As with vectors, the indexing for both rows and columns starts at 1.

We often use the terms "square matrix," "tall matrix," and "wide matrix" to describe different shapes of matrices. A square matrix is a matrix that has an equal number of rows and columns. In other words, the number of rows is equal to the number of columns. For example, a matrix with dimensions 3x3 or 4x4 is considered a square matrix. A tall matrix is a matrix that has more rows than columns, e.g., a matrix with dimensions 5x3 or 6x2. A wide matrix is a matrix that has more columns than rows, e.g., a matrix with dimensions 3x5 or 2x6.

The matrix notation subsumes the vector notation: an n-vector is simply an $n \times 1$ matrix. A matrix of size $1 \times n$ is also called a row vector, e.g., $\mathbf{a} = \begin{bmatrix} 1 & -1 & 2 \end{bmatrix}$, as opposed to the usual column vector.

4.2 Rows and Columns of a Matrix

Given an $m \times n$ matrix $\mathbf{A}$, we can use $\mathbf{a}_i$ to denote its ith column, where $\mathbf{a}_i \in \mathbb{R}^m$. Similarly, we can use $\mathbf{a}_{j\cdot}$ to denote its jth row, where $\mathbf{a}_{j\cdot}^\top \in \mathbb{R}^n$. For instance, given a matrix $\mathbf{A}$ with dimensions 3×2:

$$\mathbf{A} = \begin{bmatrix} 2 & 5 \\ -1 & 3 \\ 0 & 4 \end{bmatrix}$$

Using the notation, we can express the columns of $\mathbf{A}$ as $\mathbf{a}_1$ and $\mathbf{a}_2$, where:

$$\mathbf{a}_1 = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} 5 \\ 3 \\ 4 \end{bmatrix}$$

Similarly, we can express the rows of $\mathbf{A}$ as $\mathbf{a}_{1\cdot}$, $\mathbf{a}_{2\cdot}$, and $\mathbf{a}_{3\cdot}$, where:

$$\mathbf{a}_{1\cdot} = \begin{bmatrix} 2 & 5 \end{bmatrix} \qquad \mathbf{a}_{2\cdot} = \begin{bmatrix} -1 & 3 \end{bmatrix} \qquad \mathbf{a}_{3\cdot} = \begin{bmatrix} 0 & 4 \end{bmatrix}.$$

4.3 Block Matrices

Extending the idea of block vectors, we can similarly construct block matrices. Consider the following block matrix:

$$\mathbf{A} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{D} & \mathbf{E} \end{bmatrix}$$

In this example, we have divided the matrix $\mathbf{A}$ into four smaller submatrices or blocks: $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, and $\mathbf{E}$. To ensure that these blocks fit together properly within $\mathbf{A}$, the sizes of the blocks should be appropriate, that is,

1. The number of rows of $\mathbf{B}$ should be equal to the number of rows of $\mathbf{C}$;
2. The number of rows of $\mathbf{D}$ should be equal to the number of rows of $\mathbf{E}$;
3. The number of columns of $\mathbf{B}$ should be equal to the number of columns of $\mathbf{D}$;
4. The number of columns of $\mathbf{C}$ should be equal to the number of columns of $\mathbf{E}$.

For example,

$$\mathbf{A} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{D} & \mathbf{E} \end{bmatrix} = \left[\begin{array}{ccc|c} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \end{array}\right]$$

where the submatrices $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, and $\mathbf{E}$ are given by

1. **B** corresponds to the top-left block, which in this case is a 2×3 matrix: $\mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \end{bmatrix}$;
2. **C** corresponds to the top-right block, which is a 2×1 matrix: $\mathbf{C} = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$;
3. **D** corresponds to the bottom-left block, which is a 1×3 matrix: $\mathbf{D} = \begin{bmatrix} 9 & 10 & 11 \end{bmatrix}$;
4. **E** corresponds to the bottom-right block, which is a 1×1 matrix (or a scalar): $\mathbf{E} = \begin{bmatrix} 12 \end{bmatrix}$.

The colon notation, also known as indexing or slicing notation, can be used to extract submatrices from a general matrix $\mathbf{A}$. It provides a concise and flexible way to specify the desired rows and columns of the submatrix. The colon notation follows the format $\mathbf{A}_{p:q,r:s}$, where p, q, r, and s represent the indices or ranges of indices for rows and columns. Here's how the notation works:

1. If p and q are integers, $\mathbf{A}_{p:q,r:s}$ extracts the submatrix starting from row p to row q (inclusive) and column r to column s (inclusive). When $p = q$, we can leave one of them out, e.g., $\mathbf{A}_{p,r:s}$ is the same as $\mathbf{A}_{p:p,r:s}$; i.e.

$$\mathbf{A} = \begin{bmatrix} A_{11} & \dots & A_{1r} & \dots & A_{1s} & \dots & A_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{p1} & \dots & A_{pr} & \dots & A_{ps} & \dots & A_{pn} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{q1} & \dots & A_{qr} & \dots & A_{qs} & \dots & A_{qn} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{m1} & \dots & A_{mr} & \dots & A_{ms} & \dots & A_{mn} \end{bmatrix} \tag{4.1}$$

2. If either p or q is replaced by a colon, such as $\mathbf{A}_{:,r:s}$, it means all rows are included from the matrix $\mathbf{A}$;
3. Similarly, if either r or s is replaced by a colon, such as $\mathbf{A}_{p:q,:}$, it means all columns are included from the matrix $\mathbf{A}$; and
4. If both p and q (or r and s) are replaced by a colon, such as $\mathbf{A}_{:,:}$, it extracts the entire matrix $\mathbf{A}$.

In the example earlier, we can extract the submatrices **B**, **C**, **D**, and **E** using colon notation as follows:

$$\begin{aligned} \mathbf{B} &= \mathbf{A}_{1:2,1:3} & \qquad \mathbf{C} &= \mathbf{A}_{1:2,4} \\ \mathbf{D} &= \mathbf{A}_{3,1:3} & \qquad \mathbf{E} &= \mathbf{A}_{3,4} \end{aligned}$$

The block matrix notation also allows us to express an $m \times n$ matrix in terms of its columns $\mathbf{a}_i$ or rows $\mathbf{a}_{j\cdot}$, i.e.,

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 \; \mathbf{a}_2 \; \dots \; \mathbf{a}_n \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{1\cdot} \\ \vdots \\ \mathbf{a}_{m\cdot} \end{bmatrix}$$

We say that the matrix $\mathbf{A}$ is expressed as a concatenation of its columns or as a stack of its rows.

4.4 Interpretation of Matrices

We can interpret a matrix either as a table of values, a collection of vectors, or a representation of relationships.

4.4.1 Table

A matrix can be interpreted as a table of values, where the rows and columns correspond to different entities or variables. This tabular representation allows us to organize and analyze data in a structured manner. Here are a few examples to illustrate this interpretation:

Gradebook: Consider a gradebook for a class with five students and three assignments. We can represent this information using a matrix where the rows represent students and the columns represent assignments. Each entry in the matrix corresponds to the grade obtained by a student for a particular assignment.

Student	**Assignment 1**	**Assignment 2**	**Assignment 3**
Student 1	85	92	78
Student 2	76	80	88
Student 3	90	87	92
Student 4	82	78	84
Student 5	95	88	90

Sales Data: Imagine a retail store that tracks the sales of different products across various regions. We can represent this data using a matrix where the rows correspond to regions and the columns correspond to products.

Region	**Product 1**	**Product 2**	**Product 3**
Region 1	100	200	150
Region 2	250	180	120
Region 3	180	150	220

4.4.2 Collection of Vectors

A matrix can be interpreted as a collection of vectors, where each column may represent a feature or attribute. Such a representation is commonly used in machine learning and signal processing. For example, consider a 3xT matrix where each column represents the position or coordinate at a given time:

$$\mathbf{A} = \begin{bmatrix} x_1 & x_2 & \dots & x_T \\ y_1 & y_2 & \dots & y_T \\ z_1 & z_2 & \dots & z_T \end{bmatrix}$$

In this matrix, each column represents a vector in 3D space, where the first row corresponds to the x-coordinate, the second row corresponds to the y-coordinate, and the third row corresponds to the z-coordinate. Each column vector represents the position or coordinate at a specific time.

Another example can be found in the context of machine learning, where we have a dataset with input features and corresponding labels. Each row of the matrix represents a sample or instance, and the columns represent different features or attributes. For example, let us consider a dataset for classifying handwritten digits. Suppose we have n example images of digits, each of which can be represented as a d-vector, e.g., using pixel intensities of a $\sqrt{d} \times \sqrt{d}$ image. Each sample represents an image of a handwritten digit, and the features represent the pixel intensities of the image. The dataset can be represented as a matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}$$

In this matrix, each row $\mathbf{x}_i^\top$ represents a feature vector corresponding to a handwritten digit sample. Additionally, we may have a corresponding label vector representing the class or category of each digit:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

In this case, each element y_i represents the label or class of the corresponding digit sample. By interpreting the dataset as a matrix, we can leverage various machine learning algorithms to train models for digit classification. The matrix structure allows us to apply feature selection, feature extraction, and various classification algorithms that operate on the entire dataset or subsets of it.

4.4.3 Relationship Graph

Consider a scenario where we are given a set of n objects labeled from 1 to n. In this context, a relation $\mathcal{R}$ is defined on this set of objects. The relation $\mathcal{R}$ can be understood as a collection of ordered pairs of objects, indicating certain connections or associations between them. For instance, we can interpret $\mathcal{R}$ as a representation of a similarity relation among the n products. In this interpretation, if we have (i, j) belonging to R, it signifies that item i is considered similar to item j based on certain criteria, such as product category, customer reviews, or purchase history. Such relationships are widely used in various e-commerce applications, such as product recommendations, collaborative filtering, or market basket analysis.

A relationship can be viewed as a directed graph, where the objects or entities are represented as nodes, and the connections or associations between them are represented as directed edges. Each edge indicates a directed relationship from one node to another. This graph representation provides a visual and structural way to understand the relationship and analyze its properties.

Let us consider a relation $\mathcal{R}$ that represents the "is connected to" relationship among a group of people. Suppose we have five individuals labeled A, B, C, D, and E. The relation $\mathcal{R}$ can be defined as follows:

$$R = \{(A, B), (A, D), (B, C), (C, D), (D, E)\}$$

which is depicted in Fig. 4.4.3.

The relation can also be represented as the adjacency matrix of this graph, which is an $n \times n$ matrix defined as

$$A_{ij} = \begin{cases} 1 & (i, j) \in \mathcal{R} \\ 0 & (i, j) \notin \mathcal{R} \end{cases} \tag{4.2}$$

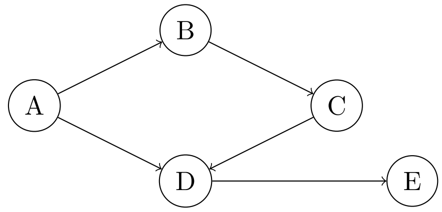

Figure 4.1: An example relation represented as a graph

For the earlier example, the adjacency matrix A is given by

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

In the adjacency matrix, each row and column represents a person, and the entry A_{ij} is 1 if there is a connection from person i to person j (i is connected to j), and 0 otherwise. The diagonal entries are typically zero in this representation since we assume no self-connections.

4.5 Special Matrices

In this section we discuss several special matrices which have their own names.

4.5.1 Zero matrix

The zero matrix, denoted as $\mathbf{0}_{m\times n}$, is the $m \times n$ matrix all of whose entries are zero. For example, $\mathbf{0}_{3\times 2}$ represents a zero matrix with 3 rows and 2 columns. In many cases, the size of the zero matrix can be inferred from the context. For instance, if we have a matrix equation where the dimensions of the zero matrix are determined by the dimensions of other matrices involved in the equation, we don't need to explicitly specify the size of the zero matrix. The context provides the necessary information.

4.5.2 Identity Matrix

The identity matrix, denoted as $\mathbf{I}_n$ or simply $\mathbf{I}$, is a square matrix with ones on the main diagonal (from the top left to the bottom right) and zeros everywhere else. The (i, j)-th entry of the identity matrix is defined as follows:

$$(\mathbf{I}_n)_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

In other words, the (i, j)-th entry is 1 when the row index i is equal to the column index j, and it is 0 otherwise. For example, the identity matrix I_3 is a 3x3 matrix defined as:

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The identity matrix $\mathbf{I}$ can also be expressed as a concatenation of standard unit vectors as:

$$\mathbf{I} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \dots & \mathbf{e}_n \end{bmatrix}$$

where $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ are the standard unit vectors.

The size subscript of the identity matrix is often omitted when it can be inferred from the context. For example, in a block matrix:

$$\begin{bmatrix} \mathbf{A} & \mathbf{I} \\ \mathbf{0} & \mathbf{C} \end{bmatrix}$$

we can infer that $\mathbf{I}$ is a square matrix with the same size as number of rows of $\mathbf{A}$ and the number of columns of $\mathbf{C}$.

4.5.3 Diagonal and Block Diagonal Matrices

A diagonal matrix is a square matrix in which all the elements outside the main diagonal (the diagonal from the top left to the bottom right) are zero. The elements on the main diagonal can be any real or complex numbers. A diagonal matrix is denoted as D or using the "Diag()" notation.

For example, consider the diagonal matrix D with elements 2, 4, and 6 on the main diagonal:

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

The "Diag()" notation is a concise way of representing a diagonal matrix. For a given vector $\mathbf{d} = [d_1, d_2, \dots, d_n]$, the diagonal matrix $\mathbf{D}$ with $\mathbf{d}$ as its main diagonal can be written as:

$$\mathbf{D} = \mathrm{Diag}(\mathbf{d}) = \begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_n \end{bmatrix}$$

In addition to diagonal matrices, we also have block diagonal matrices. A block diagonal matrix is a matrix that is composed of smaller square matrices along the main diagonal, with all other elements being zero. Each smaller square matrix is called a block. Block diagonal matrices are often denoted using the "blkdiag()" notation.

For example, consider the block diagonal matrix B formed by blocks B_1, B_2, and B_3:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & 0 & 0 \\ 0 & \mathbf{B}_2 & 0 \\ 0 & 0 & \mathbf{B}_3 \end{bmatrix}$$

The "blkdiag()" notation is used to represent a block diagonal matrix. For example, if we have matrices B_1, B_2, and B_3, the block diagonal matrix B can be written as:

$$\mathbf{B} = \mathrm{blkdiag}(\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)$$

The sizes of the blocks in a block diagonal matrix can vary, but each block must be a square matrix.

4.5.4 Triangular Matrices

An upper triangular matrix is a square matrix in which all the entries below the main diagonal are zero. The entries above the main diagonal can be any non-zero values. For example, consider the following upper triangular matrix $\mathbf{U}$:

$$\mathbf{U} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}$$

In this matrix, all the entries below the main diagonal are zero. Similarly, a lower triangular matrix is a square matrix in which all the entries above the main diagonal are zero. The entries below the main diagonal can be any non-zero values. For example, consider the following lower triangular matrix $\mathbf{L}$:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 4 & 5 & 6 \end{bmatrix}$$

In this matrix, all the entries above the main diagonal are zero. In an $n \times n$ triangular matrix, there are $n(n-1)/2$ zero entries.

4.6 Tensors or multi-dimensional arrays

Vectors and matrices can be generalized to tensors or mutli-dimensional arrays, which are elements from $\mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}$. An entry of a tensor $\mathbf{A}$ can be accessed as $A_{i_1 i_2 \ldots i_d}$.

Using the colon notation, $\mathbf{A}_{:,:,i_3,\ldots,i_d}$ is a $n_1 \times n_2 \times 1 \times \ldots \times 1$ tensor, but can also be interpreted as a matrix. However, $\mathbf{A}_{i_1,:,:,i_4,\ldots,i_d}$ would still be a $1 \times n_2 \times n_3 \times 1 \times \ldots \times 1$ tensor (with a third dimension which is not trivial) which would have to be reshaped before treating it as a matrix. In general, while the colon operator always returns a tensor, if the resulting tensor has all but the first two dimensions of size 1, then it can be interpreted as a (two-dimensional) matrix.

If we view a matrix as a two-dimensional array, then the transpose operation interchanges the first and second dimensions. More generally, we can can think of permuting the dimensions of a higher-dimensional arrays. For an n_d-array, there are $d!$ such permutations possible. For instance, a possible permutation of a $3 \times 2 \times 5 \times 4$ would yield a $2 \times 3 \times 4 \times 5$ array.