Matrices

Matrices are as fundamental as vectors in machine learning. With vectors, we can represent single variables as sets of numbers or instances. With matrices, we can represent sets of variables. In this sense, a matrix is simply an ordered collection of vectors.
More formally, we represent a matrix with a italicized upper-case letter like A. In two dimensions, we say the matrix A has m rows and n columns.Each entry of A is difined as $a_ij, i = 1, \ldots, m, and j = 1, \ldots, n$. A matrix $A \in \mathbb{R}^{m \times n}$ is defined as:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, a_{ij} \in \mathbb{R}$$

Basic Matrix operations

Matrix-matrix addition

We add matrices in a element-wise fashion. The sum of $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{m \times n}$ is defined as:

$$A + B := \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & \cdots & a_{2n} + b_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} + b_{m_1} & \cdots & a_{mn} + b_{mn} \end{bmatrix} \in \mathbb{R}$$

Matrix-scalar multipication

Matrix-scalar multiplication is an element-wise operation. Each element of the matrix A is multiplied by the scalar α. It is defined as:

$$a_{ij} \times \alpha, such\ that (\alpha A)_{ij} = \alpha (A)_{ij}$$

Matrix-vector multiplication: dot product

Matrix-vector multiplication equals to taking the dot product of each column n of a A with each element x resulting in a vector y. It is defined as:

$$A \cdot x := \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} a_{11} \\ \vdots \\ x_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ \vdots \\ x_{m2} \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_{1n} \\ \vdots \\ x_{mn} \end{bmatrix}$$

Matrix-matrix multiplication

Matrix-matrix multiplication is a dot product as well. To work, the number of columns in the first matrix A has to be equal to the number of rows in the second matrix B. Hence, $A \in \mathbb{R}^{m \times n}$ times $B \in \mathbb{R}^{n \times p}$ to be valid. One way to see matrix-matrix multiplication is by taking a series of dot products: the 1st row of A times the 1st column of B, the 2nd row of A times the 2nd column of B, until the n_{th} colum of A times the n_{th} row of B.
We define $A \in \mathbb{R}^{m \times n} \times B \in \mathbb{R}^{n \times p} = C \in \mathbb{R}^{m \times p}$

$$c_{ij} := \sum_{l=1}^{n} a_{il} b_{lj}, with\, i = 1, \dots m, and j = 1, \dots p$$

Matrix-matrix multiplication has a series of properties:
1. Associativity: $(AB)C = A(BC)$
2. Associativity with scalar multiplication:$\alpha(AB) = (\alpha A)B$
3. Distributivity with addtion: $(A(B + C) = AB + AC)$
4. Transpose of product:$(AB)^T = B^T A^T$

It's also important to remember that **matrix-matrix multiplication orders matter**, that is, it is **not commutative**. Hence, in general, $AB \neq BA$

Matrix identity

An identity matrix is a square matrix with ones on the diagonal from the upper left to the bottom right, and zeros everywhere else. We denote the identity matrix as I_n. We define $I \in \mathbb{R}^{m \times n}$ as:

$$I_n := \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{m \times n}$$

Matrix inverse

Consider the square matrix $A \in \mathbb{R}^{m \times n}$. We define A^{-1} as matrix with the property:

$$A^{-1}A = I_n = AA^{-1}$$

The main reason we care about the inverse, is because it allows to solve systems of linear equations in certain situations. Consider a system of linear equations as:

$$Ax = y$$

Assuming A has an inverse, we can multiply by the inverse as both sides:

$$A^{-1}Ax = A^{-1}y$$

And get:

$$x = A^{-1}y$$

This means that we just need to know the inverse of A, multiply by the target vector y, and we obtain the solution for our system. **This work only in certain situations: if and only if A happens to have an inverse**. Not a_l matrices have an inverse. When $A^{-1}exist, wesay$A$ is nonsingular or invertible, otherwise, we say it is noninvertible or singular.

The lingering question is how to find the inverse of a matrix. We can do it by reducing A to its reduced row echelon form by using Gauss-Jordan Elimination. If A has an inverse, we will obtain the identity matrix as the row echelon form of A.

Matrix transpose

Consider a matrix $A \in \mathbb{R}^{m \times n}$. The transpose of A is denoted as $A^T \in \mathbb{R}^{m \times n}$. We obtain A^T as:

$$(A^T)_{ij} = A_{ji}$$

In other words, we get the A^T by switching the columns by the rows of A.

Hadamard product

It is tempting to think in matrix-matrix multiplication as an element-wise operation, as multiplying each overlapping element of A abd B. *It is not.*. Such operation is called **Hadamard product**. The Hadamard product is defined as:

$$a_{ij}b_{ij} := c_{ij}$$

Special matrices

There are several matrices with special names that are commonly found in machine learning theory and application. Knowing these matrices beforhand can improve your linear algebra fluency, so we will briefly review a selection of 12 common matrices.

Rectangular matrix

Matrices are said to be rectangular when the number of rows is $\neq$ to the number of columns, i.e., $A^{m \times n}$ with $m \neq n$.

Square matrix

Matrices are said to be square when the number of rows = the number of columns.

Diagonal matrix

Square matices are said to be diagonal when each of its non-diagonal elemets is zero. For instance:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 9 \end{bmatrix}$$

Upper triangular matrix

Square matrices are sad to be upper triangular when the elements below the main diagonal are zero. For instance:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 6 \\ 0 & 0 & 9 \end{bmatrix}$$

Identity matrix

A diagonal matrix is said to be the identity when the elements along its main diagonal are equal to one. For instance:

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

Scalar matrix

Diagonal matrices are said to be scalar when all the elements along its main diagonal are equal. For instance:

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Null or zero matrix

Matrices are said to be null or zero matrices when all its elements equal to zero, which is denoted as 0_{m\times n}.

Echelon matrix

Matrices are said to be on echelon form when it has undergone the process of Gaussian elimination. More specifically:
1. Zero rows are at the bottom of the matrix.
2. The leading entry (pivot) of each nonzero row is to the right of the leading entry of the row above it.
3. Each leading entry is the only nonzero entry in its column.
For instance:

$$\begin{bmatrix} 1 & 3 & 5 \\ 2 & 2 & -1 \\ 1 & 3 & 2 \end{bmatrix}$$

In echelon form after Gaussian Elimination becomes:

$$\begin{bmatrix} 1 & 3 & 5 \\ 0 & -4 & -11 \\ 0 & 0 & -3 \end{bmatrix}$$

Antidiagonal matrix

Matrices are said to be antidiagonal when all the entries are zero but the antidiagonal (i.e., the diagonal starting from the bottom left corner to the upper right corner). For instance:

$$\begin{bmatrix} 0 & 0 & 3 \\ 0 & 5 & 0 \\ 7 & 0 & 0 \end{bmatrix}$$

Design matrix

Design matrix is a special name for matrices containing explanatory variables or features in the context of statistics and machine learning. Some authors favor this name to refer to the set of variables or features in a model.

Matrices as system of linear equations

Matrices are ideal to represent systems of linear equations. Consider the matrix M and vector w and y in $\mathbb{R}^3$. We can set up a system of linear equations as $Mw = y$ as:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

This is equivalent to:

$$\begin{aligned} m_{11}w_1 + m_{12}w_2 + m_{13}w_3 &= y_1 \\ m_{21}w_1 + m_{22}w_2 + m_{23}w_3 &= y_2 \\ m_{31}w_1 + m_{32}w_2 + m_{33}w_3 &= y_3 \end{aligned}$$

Geometrically, the solution for this representation equals to plot a set of planes in 3-dimensional space, one for each equation, and to find the segment where the plances intersect.

An alternative way, is to represent the system as a linear combination of the column vectors times a scaling term:

$$w_1 \begin{bmatrix} m_{11} \\ m_{23} \\ m_{31} \end{bmatrix} + w_2 \begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \end{bmatrix} + w_3 \begin{bmatrix} m_{13} \\ m_{23} \\ m_{33} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Geometrically, the solution for this representation equals to plot a set of vectors in 3-dimensional space, one for each column vector, then scale them by w_i and add them up, tip to tail, to find the resulting vector y.

The four fundamental matrix subspaces

Let's recall the definition of a subspace in the context of vectors:
1. Contains the zero vector, $0 \in S$
2. Closure under multiplication, $\forall \alpha \in \mathbb{R} \rightarrow \alpha \times s_i \in S$
3. Closure under addition, $\forall s_i \in S \rightarrow s_1 + s_2 \in S$

These conditions carry on to matrices since matrices are simply collections of vectors. Thus, now we can ask what are all possible subspaces that can be "covered" by a collection of vectorss in a matrix. Turns out, there are four fundamental subspaces that can be "covered" by a matrix of valid vectors: (1) the column space, (2) the row space, (3) the null space, and (4) the left null space or null space of the transpose.

These subspaces are considered fundamental because they express many important properties of matrices in linear algebra.

The column space

The column space of a matrix A is composed by all linear combinations of the column of A. We denote the column space as $C(A)$. In other words, $C(A)$ equals to the span of the columns of A.

For a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $v \in \mathbb{R}^m$, the column space is defined as:

$$C(A) := \{w \in \mathbb{R}^n | w = Av \; for \; some \; v \in \mathbb{R}^m\}$$

In words: all linear combinations of the column vectors of A and entries of an n dimensional vector v.

The row space

The row space of a matrix A is composed of all linear combinations of the rows of a matrix. We denote the row space as $R(A)$. In other words, $R(A)$ equals to the span of rows of A. Now, a different way to see the row space, is by transposing A^T. Now, we can define the row space simply as $R(A^T)$.

For a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $w \in \mathbb{R}^m$, the row sapce is defined as:

$$R(A) := \left\{v \in \mathbb{R}^m | v = Aw^T \; for \; some \; w \in \mathbb{R}^n\right\}$$

In words: all linear combinations of the row vectors of A and entries of an dimensional vector w.

The null space

The null space of a matrix A is composed of all vectors that are map into the zero vector when multiplied by A. We denote the null space as $N(A)$.

For a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $v \in \mathbb{R}^n$, the null space is defined as:

$$N(A) := \{v \in \mathbb{R}^m | Av = 0\}$$

The null space of the transpose

The left null space of a matrix A is composed of all vectors that are map into the zero vector when multiplied by A from the left. By "from the left", the vectors on the left of A. We denote that the left null space as $N(A^T)$

For a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $w \in \mathbb{R}^m$, the null space is defined as:

$$N(A^T) := \left\{w \in \mathbb{R}^n | v^T A = 0^T\right\}$$

Solving systems of linear equations with Matrices

Gaussian elimination

Gaussian elimination, is a robust algorithm to solve linear systems. We say it is robust, because it works in general, in all possible circumstances. It works by eliminating terms from a system of equations, such that it is simplified to the point where we obtain the row echelon form of the matrix. A matrix is in row echelon form when all rows contain zeros at the bottom left of the matrix. For example:

$$\begin{bmatrix} p_1 & a & b \\ 0 & p_2 & c \\ 0 & 0 & p_3 \end{bmatrix}$$

The p values along the diagonal are the pivots also know as basic variables of the matrix. An important remark about the pivots, is that they indicate which vectors are linearly independent in the matrix, onece the matrix has been reduced to the row echelon form.

There are three elementary transformations in Gaussian elimination that when combined, allow simplifying any system to its row echelon form:
1. Addition and substration of two equations (rows)

2. Multiplication of an euqation (row) by a number
3. Switching equations (rows)

Gauss-Jordan elimination
The only difference between Gaussian Elimination and Gauss-Jordan Elimination, is that this time we "keep going" with the elemental row operations until we obtain the reduced row echelon form. The reduced part means two additional things:
(1) The pivots must be 1
(2) The entries above the pivots must be 0.
This is simplest form a system of linear equations can take.

Matrix basis and rank

A set of n linearly independent column vectors with n elements form a basis. For instance, the column vectors of A are a basis:

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

"A basis for what?" You may be wondering. In the case of A, for any vector $y \in \mathbb{R}^2$. On the contrary, the column vectors for B do not form a basis for $\mathbb{R}^2$:

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

In the case of B, the third column vector is a linear combination of first and second column vectors. The definition of a basis depends on the independence-dimension inequality, which states that a linearly independent set of n vectors can have at most n elements. Alternatively, we say that any set of n vectors with n+1 elements is, necessarily, linearly dependent. Given that each vectors in a basis is linearly independent, we say that any vector y with n elements, can be generated in a unique linear combination of the basis vectors. Hence, any matrix more columns than rows (as in B) will have dependent vector. Basis are sometimes referred to as the minimal generating set.

An important question is how to find the basis for a matrix. Another way to put the same question is to found out which vectors are linearly independent of each other. Hence, we n_ed to solve:

$$\sum_{i=1}^{k} \beta_i \alpha_i = 0$$

Where α_i are the column vectors of A。 We can approach this by using Gaussian Elimination or Gauss-Jordan Eli\mination and reducing A to its row echelon form or reduced row echelon form. In either case, recall that the pivots of the echelon form indicate the set of linearly independent vectors in a matrix.

Now that we know about a basis and how to find it, understanding the concept of rank is simpler. Than rank of a matrix A is the dimensionality of the vector space generated by its number of linearly independent column vectors. This happens to be identical to the dimensionality of the vector space generated by its row vectors. We denote the rank of matrix as $rk(A)$ or $rank(A)$.

For an square matrix $\mathbb{R}^{m \times n}$ (i.e., $m = n$), we say is full rank when every column and/or row is linearly independent. For a non-square matrix with $m > n$ (i.e. more rows than columns), we say is full rank when every column is linearly independent. When $m < n$ (i.e., more columns than rows), we say is full rank when every row is linearly independent.

From an applied machine learning perspective, the rank of a matrix is relevant as a measure of the information content of the matrix.

Matrix norm

As with vectors, we can measure the size of a matrix by computing its norm. There are multiple ways to define the norm for a matrix, as long it satisfies the same properties defined for vector norms: (1) absolutely homogeneous (2) triangle inequality, (3) positive definite. Here are the three most commonly used norms in machine learning: (1) Frobenius norm, (2) max norm, (3) spectral norm.

Frobenius norm

The Frobenius norm is an element-wise norm named after the German mathematician Ferdinand Georg Frobenius. We denote this norm as $||A||_F$. You can think about this norm as flattening out the matrix into a long vector. For instance, a 3×3 matrix would become a vector with $n = 9$ entries. We define the Frobenius norm as:

$$||A||_F := \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$$

where A is a matrix with elements a_{ij}.
In words: square each entry of A, add them together, and then take the square root.

Max norm

The \max norm or infinity norm of a matrix equals to the largest sum of the absolute value of row vectors. We denote the max norm as $||A||_{max}$. Consider $A \in \mathbb{R}^{m \times n}$. We define the max norm for A as:

$$||A||_{max} := max_i \sum_{j=1}^{n} |a_{ij}|$$

This equals to go row by row, adding the absolute value of each entry, and then selecting the largest sum.

Spectral norm

To understand this norm, it's necessary to first learn about eigenvectors and eigenvalues. The spectral norm of a matrix equals to the largest singular value σ_1. We denote the spectral norm as:

$$||A||_2 := max_x \frac{||Ax||_2}{||x||_2}$$