The purpose of this post is to encapsulate the foundational linear algebra knowledge that I gained from working through the 18.06 course on MIT OCW (located [here]( https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall-2011/index.htm)). Gilbert Strang is an amazing instructor and I highly recommend this course for anyone wanting to explore the depths of linear algebra. This post was written to serve as a reference for the fundamental concepts of linear algebra.

# Introducing Vectors and Matrices

In this section, we will discuss definitions for vectors and matrices which are the foundational mathematical objects used in linear algebra.

## What is a Vector?

A source of confusion when learning about vectors is the abundance of differing definitions. For example, in high school physics, a vector is introduced as a quantity that has length and direction. While in computer science, a vector is defined as an ordered list of numbers.

So… what \**really*\* is a vector? For the purpose of this post and our development of linear algebra, it is useful to define a vector in its most general form, namely: \*\*a vector is simply an element of a vector space\*\*.

A vector space is defined as a set of objects \[V\] (elements of \[V\] are called vectors), and a field \[\user2{\mathbb{F}}\] (elements of \[\user2{\mathbb{F}}\] are called scalars). In general, there are several axioms that must be satisfied for \[V\] and \[\user2{\mathbb{F}}\] to be considered a vector space. Intuitively, these axioms largely boil down to the requirement that any linear combination of vectors in \[V\] (using scalar coefficients from \[\user2{\mathbb{F}}\]) must also be in \[V\] (i.e. if \[{\user1{v}\_1},{\user1{v}\_2} \in V\], then for \[{c\_1},{c\_2} \in \user2{\mathbb{F}}\], we have \[{c\_1}{\user1{v}\_1} + {c\_2}{\user1{v}\_2} \in V\]).

The advantage of a vector space is that we can now generalize the concept of a vector beyond the specific definitions given in high school physics and computer science. For illustration, the following are examples of vector spaces:

* \*\*Real Coordinate Space \[{\mathbb{R}^n}\]:\*\*: The real coordinate space \[{\mathbb{R}^n}\] over the field of real numbers is probably the most common and widely used vector space, where the vectors are n-tuples of real numbers. For example, a vector in this space could be \[\left[ {\begin{array}{\*{20}{c}}1\\2\\3\end{array}} \right] \in {\mathbb{R}^3}\]. In computer science, when we think about a “vector” as an ordered list of real numbers, we are really thinking about a coordinate vector in \[{\mathbb{R}^n}\].
* \*\*Complex Matrices \[{\mathbb{C}^{m \times n}}\]:\*\* The vector space \[{\mathbb{C}^{m \times n}}\] over the field of complex numbers consists of all \[m \times n\] matrices that have complex values. Notice how for this vector space, the vectors are actually matrices. For example, a vector in this space could be \[\left[ {\begin{array}{\*{20}{c}}i&2\\{3i}&{4i + 2}\end{array}} \right] \in {\mathbb{C}^{2 \times 2}}\].
* \*\*Polynomials \[{P\_n}\]:\*\* The vector space \[{P\_n}\] over the field of real numbers consists of all polynomials of degree \[n\] and smaller. For example, a vector in this space could be \[{x^3} + 2x + 5 \in {P\_3}\].
* \*\*Solutions to Linear Homogeneous Differential Equations:\*\* Suppose that \[{f\_1},{f\_2}\] are distinct solutions to the equation \[f'' = f\], such that \[{f''\_1} = {f\_1}\] and \[{f''\_2} = {f\_2}\]. Due to the linearity of the differentiation operator, it follows that any linear combination of \[{f\_1},{f\_2}\] will also be a solution to \[f'' = f\]. To see this, we note that \[{\left( {{c\_1}{f\_1} + {c\_2}{f\_2}} \right)^{\prime \prime }} = {c\_1}{f\_1}^{\prime \prime } + {c\_2}{f\_2}^{\prime \prime } = {c\_1}{f\_1} + {c\_2}{f\_2}\]. Therefore, the solutions to the differential equation \[f'' = f\] form a vector space over the field of real numbers and the vectors in this space are functions. For example, a vector in this space could be \[f\left( t \right) = \sin \left( t \right)\].
* \*\*Column Space of a Matrix in \[{\mathbb{R}^{m \times n}}\]:\*\* The column space of a matrix in \[{\mathbb{R}^{m \times n}}\] over the field of real numbers is defined by the set of all linear combinations of the columns in the matrix. Since the column vectors are in \[{\mathbb{R}^m}\], it follows that the column space will be a subspace of \[{\mathbb{R}^m}\], where the subspace dimension is equal to the number of independent columns in the matrix. The elements of this vector space are vectors in \[{\mathbb{R}^m}\]. For example, a vector in the column space of \[\left[ {\begin{array}{\*{20}{c}}1&3\\2&2\end{array}} \right]\] could be \[4\left[ {\begin{array}{\*{20}{c}}1\\2\end{array}} \right] + 2\left[ {\begin{array}{\*{20}{c}}3\\2\end{array}} \right]\].

To summarize, a vector space is defined by a set of vectors \[V\] and a field \[\user2{\mathbb{F}}\]. For \[\left( {V,\user2{\mathbb{F}}} \right)\] to be a valid vector space, it must be closed under linear combination. This means that if \[{\user1{v}\_1},{\user1{v}\_2} \in V\], then for \[{c\_1},{c\_2} \in \user2{\mathbb{F}}\], we have \[{c\_1}{\user1{v}\_1} + {c\_2}{\user1{v}\_2} \in V\]. You can check that this result holds for the example vector spaces shown above. Thus, our answer to “what is a vector” is simply this: a vector is an element of a vector space.

It is important to note that the most general vector space does not include a notion of distance. To include a metric of distance, we must upgrade our general vector space into a normed vector space. If we go further still, and define an inner product on our space to measure how orthogonal two vectors are (e.g. the inner product for \[{\mathbb{R}^n}\] is the dot product), then we upgrade our normed vector space into an inner product space. A great resource to see the relationships between mathematical spaces is [here](https://en.wikipedia.org/wiki/Space\_(mathematics)). For this post, we will be using the term \*vector space\* quite loosely and in most cases will assume that the vector space is an inner product space (i.e. has norm and inner product defined for the space).

The advantage of working with the generalized concept of a vector space is that any results proven for the general case can be applied equally well to any vector space regardless of whether it is composed vectors that are n-tuples, matrices, polynomials, or functions. If we can show that our data forms a valid vector space, then we can leverage the mathematics developed on generalized vector spaces to analyze our data. For example, consider the natural language processing application of determining word embedding vectors. For this task, it is assumed that words in a language form a vector space such that more similar words are closer in space, and the relationship between words can be represented by the additive properties of the vector space (e.g. king – man + woman = queen). The goal of word embedding models such as word2vec is to learn a representation of words as coordinate vectors in \[{\mathbb{R}^n}\]. Once we have a coordinate vector for each word, we can apply the properties of the \[{\mathbb{R}^n}\] vector space to geometrically visualize words as coordinates in space so that we can calculate distances between words, linear combinations of words, and the inner product between words.

The real power of linear algebra comes from being able to use it to computationally solve equations and process data on a computer. Since computers work with discrete numbers in memory, this raises a challenge when working with vector spaces whose vectors are objects such as continuous functions. For example, suppose that we are trying to solve the equation \[{x\_1}\left( {{t^2} + 4t + 1} \right) + {x\_2}\left( {2{t^2} + 5t + 2} \right) + {x\_3}\left( {4{t^2} + 6t + 3} \right) = 3{t^2} + 2t + 1\]. In this case, we are trying to find whether a vector in \[{P\_2}\] can be written as a linear combination of three other vectors from \[{P\_2}\]. To solve this problem on a computer, we must first answer an important question: how can we represent a polynomial vector from \[{P\_2}\] (e.g. \[{t^2} + 4t + 1\]) on a computer? This brings us to one of the most powerful attributes of vector spaces. Recall that any linear combination of vectors in a vector space is also contained in the vector space. Intuitively it makes sense that we should be able to have a minimal set of vectors whose linear combinations can describe every other vector in the space. As an informal proof of this, consider a set containing all the vectors in the vector space. Now follow an iterative pattern of: 1) taking each vector and checking whether it can be written as a linear combination of the other vectors in the set, 2) if a linear combination exists, then remove the vector from the set. If we continue this process, eventually we will get down to a minimal set of vectors whose linear combinations describe all vectors in the space. This minimal set is called a basis for the space, each vector in the basis is called a basis vector, and the number of vectors in the basis is called the dimension of the vector space. For example, a basis for \[{P\_2}\] could be the set \[\left\{ {1,t,{t^2}} \right\}\] which has dimension 3 and whose linear combinations could be used to describe any polynomial of degree 2 and smaller. Suppose that we want to represent the polynomial \[{t^2} + 4t + 1\] using this basis. We can write down the corresponding coefficients as an ordered list of numbers \[\left[ {\begin{array}{\*{20}{c}}1\\4\\1\end{array}} \right]\], such that we can express the polynomial as a linear combination of the basis vectors \[1 \* {t^2} + 4 \* t + 1 \* 1 = {t^2} + 4t + 1\]. Hence, to represent the polynomial vector \[{t^2} + 4t + 1\], we only need to know the corresponding coefficients of the basis vectors. This is critical because we can represent this list of coefficients easily on a computer. Therefore, instead of working directly with the polynomial vector, we can work with the coefficients defined for the basis we choose. Furthermore, notice that this ordered list of coefficients is itself a vector in \[{\mathbb{R}^3}\], since \[\left[ {\begin{array}{\*{20}{c}}1\\4\\1\end{array}} \right] \in {\mathbb{R}^3}\].

In general, we can describe any vector in a vector space as a linear combination of basis vectors. Therefore, for a given basis, each vector in the space is uniquely identified by a coordinate vector whose elements are the basis vector coefficients in the linear combination. Hence, we can use coordinate vectors to represent any general vector (e.g. function, polynomial, etc.) on a computer and apply linear algebra operations to these coordinate vectors.

In summary, to use vectors for computational linear algebra we can think about the following steps:

1. Define a vector space that represents the objects we are analyzing (e.g. tabular data as coordinates, polynomials, differentiable functions, language words, matrices, etc.)
2. Determine an appropriate basis, such that all basis vectors are independent (i.e. a minimal set), and every vector in the space can be written as a linear combination of the basis vectors
3. Perform a “coordinate transform” where we represent vectors in the space using coordinate vectors whose elements are the linear combination coefficients of the basis vectors (typically the coordinate vectors are in \[{\mathbb{R}^n}\])
4. Perform computational linear algebra procedures using the coordinate vector representation such as solving linear equations, change of basis (e.g. Fourier transform), or linear transformations

## What is a Matrix?

In its most general form, a matrix is simply a rectangular array of mathematical objects such as numbers or symbols. Depending on the context, matrices can represent different things, just like how the number “3” can mean different things in different contexts. The strength of matrices is that we can use results from matrix theory to utilize the structure of the matrix to help solve the specific problem at hand (e.g. solving a system of equations). Such techniques from matrix theory include eigenvectors/eigenvalues, different factorizations (QR, SVD, LR, etc.), positive definiteness, four fundamental subspaces, etc. Once we pose our problem in matrix form, we gain access to these powerful analytical tools from matrix theory to help understand and manipulate the structure of our data. Some examples of the data that matrices represent includes the following:

* \*\*Linear Equation Coefficients:\*\* Suppose that we have a system of linear equations. We can write coefficients of the equations in a matrix \[\user1{A}\] which enables us to write the entire system as \[\user1{Ax} = \user1{b}\]. When thinking about systems of linear equations, it is especially useful to interpret the matrix in terms of its four fundamental subspaces.
* \*\*Linear Transformations:\*\* Suppose we have a linear transformation \[f:V \to W\]from a vector space \[V\] to a vector space \[W\]. If \[\left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\] is a basis for \[V\], then every vector \[\user1{v} \in V\] is uniquely determined by the coordinates \[{c\_1}, \ldots ,{c\_n}\] in \[{\mathbb{R}^n}\] such that \[\user1{v} = {c\_1}{\user1{v}\_1} + \cdots {c\_n}{\user1{v}\_n}\]. Due to the linearity of \[f\], we can write \[f\left( \user1{v} \right) = f\left( {{c\_1}{\user1{v}\_1} + \cdots {c\_n}{\user1{v}\_n}} \right) = {c\_1}f\left( {{\user1{v}\_1}} \right) + \cdots + {c\_n}f\left( {{\user1{v}\_n}} \right)\]. This implies that the function \[f\] is completely determined by the transformed basis vectors \[f\left( {{\user1{v}\_1}} \right), \ldots ,f\left( {{\user1{v}\_n}} \right)\]. Now let \[\left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_m}} \right\}\] be a basis for \[W\]. Since\[f\left( {{\user1{v}\_j}} \right) \in W\], we can write each transformed vector in terms of the basis vectors of \[W\] as \[f\left( {{\user1{v}\_j}} \right) = {a\_{1,j}}{\user1{w}\_1} + \cdots {a\_{m,j}}{\user1{w}\_m}\]. Hence, the function \[f\] is completely determined by the coordinates \[{a\_{i,j}}\]. We can arrange these coordinates in a matrix such that\[\user1{A} = \left[ {\begin{array}{\*{20}{c}}{{a\_{1,1}}}& \cdots &{{a\_{n,1}}}\\ \vdots & \ddots & \vdots \\{{a\_{m,1}}}& \cdots &{{a\_{m,n}}}\end{array}} \right]\]. Therefore, the matrix \[\user1{A}\] completely determines the transformation \[f:V \to W\] using \[\left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\] and \[\left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_m}} \right\}\] as basis. \*\*Important insight:\*\* we can interpret each column of a general matrix \[\user1{A}\] as expressing the coordinates for a transformed input basis vector in terms of the output basis vectors. Therefore, for the transformation \[\user1{w = Av}\], we can interpret \[\user1{v}\] as representing an input coordinate vector relative to the basis \[\left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\], and \[\user1{w}\] as representing an output transformed coordinate vector relative to the basis \[\left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_m}} \right\}\]. The choice of basis for \[V\] and \[W\] is arbitrary but will change the matrix used to represent the linear transformation. For example, the \[{k^{th}}\] column of \[\user1{A}\] contains the coordinates \[\left[ {\begin{array}{\*{20}{c}}{{a\_{1,k}}}\\ \vdots \\{{a\_{m,k}}}\end{array}} \right]\] which is the transformed input basis vector \[f\left( {{\user1{v}\_k}} \right)\] written as a coordinate vector in terms of the output basis vectors as \[f\left( {{\user1{v}\_k}} \right) = {a\_{1,k}}{\user1{w}\_1} + \cdots {a\_{m,k}}{\user1{w}\_m}\]. For example, let us compute the linear transformation \[\user1{Ax} = \left[ {\begin{array}{\*{20}{c}}1&3\\2&3\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{{x\_1}}\\{{x\_2}}\end{array}} \right]\] where \[\left[ {\begin{array}{\*{20}{c}}{{x\_1}}\\{{x\_2}}\end{array}} \right]\] is a coordinate vector relative to the input basis. To visualize: 1) start by drawing a 2D Cartesian coordinate plane, 2) plot the columns of \[\user1{A}\] as coordinate vectors in this plane (these are the coordinates of the input basis vectors written in terms of the output basis vectors), 3) plot the linear combination of these vectors where \[\left[ {\begin{array}{\*{20}{c}}{{x\_1}}\\{{x\_2}}\end{array}} \right]\] are the scalar coefficients to get the transformed output as a coordinate vector relative to the output basis.
* \*\*Change of Basis:\*\* Suppose that we have a basis for our vector space \[{B\_{old}} = \left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\]. Now suppose that we have a vector \[\user1{v} = {c\_1}{\user1{w}\_1} + \cdots {c\_n}{\user1{w}\_n}\] that has been written in terms of a new basis \[{B\_{new}} = \left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_n}} \right\}\]. How can we write the coordinates for \[\user1{v}\] in terms of our old basis \[{B\_{old}} = \left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\]? Suppose that we can write each new basis vector in terms of the old basis vectors such that \[{\user1{w}\_j} = {a\_{1,j}}{\user1{v}\_1} + \cdots {a\_{n,j}}{\user1{v}\_n}\]. Then we can expand the vector as \[\user1{v} = {c\_1}\left( {{a\_{1,1}}{\user1{v}\_1} + \cdots {a\_{n,1}}{\user1{v}\_n}} \right) + \cdots {c\_n}\left( {{a\_{1,n}}{\user1{v}\_1} + \cdots {a\_{n,n}}{\user1{v}\_n}} \right)\]. Rearranging, we can put the entries \[{a\_{i,j}}\] into a matrix \[\user1{A}\] to write\[\user1{A}\left[ {\begin{array}{\*{20}{c}}{{c\_1}}\\ \vdots \\{{c\_n}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{{a\_{1,1}}}& \cdots &{{a\_{n,1}}}\\ \vdots & \ddots & \vdots \\{{a\_{n,1}}}& \cdots &{{a\_{n,n}}}\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{{c\_1}}\\ \vdots \\{{c\_n}}\end{array}} \right]\]. Therefore, the matrix \[\user1{A}\] takes a coordinate vector relative to the basis \[{B\_{new}} = \left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_n}} \right\}\] and produces a new coordinate vector relative to the original basis \[{B\_{old}} = \left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\]. Each column of \[\user1{A}\] provides the coordinates for each basis vector in \[{B\_{new}} = \left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_n}} \right\}\] written using the old basis vectors \[{B\_{old}} = \left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\]. Note that to represent a change of basis, \[\user1{A}\] must be a square invertible matrix. Therefore, we see that the matrix \[\user1{A}\] will change the basis of a coordinate vector relative to the “new” basis into a coordinate vector relative to the “old” basis. Likewise, the inverse \[{\user1{A}^{ - 1}}\] will change the basis of a coordinate vector relative to the “old” basis into a coordinate vector relative to the “new” basis. More generally, suppose that we have a linear transformation \[\user1{T}:V \to W\] which maps from a vector space \[V\] to a vector space \[W\] using \[\left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\] and \[\left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_m}} \right\}\] as basis for \[V\] and \[W\] respectively. Suppose that would like to represent the input and output coordinate vectors relative to other basis \[\left\{ {\user1{v}\_1^\*, \ldots ,\user1{v}\_n^\*} \right\}\] and \[\left\{ {\user1{w}\_1^\*, \ldots ,\user1{w}\_m^\*} \right\}\]. We can achieve this by first utilizing two change of basis matrices \[{\user1{P}\_{{\user1{v}^\*} \to \user1{v}}}\] and \[{\user1{Q}\_{\user1{w} \to {\user1{w}^\*}}}\], where \[{\user1{P}\_{{\user1{v}^\*} \to \user1{v}}}\] performs a change of basis from \[\left\{ {\user1{v}\_1^\*, \ldots ,\user1{v}\_n^\*} \right\}\] to \[\left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\], and \[{\user1{Q}\_{\user1{w} \to {\user1{w}^\*}}}\] performs a change of basis from \[\left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_m}} \right\}\] to \[\left\{ {\user1{w}\_1^\*, \ldots ,\user1{w}\_m^\*} \right\}\]. Then a coordinate vector \[\user1{x}\] written relative to \[\left\{ {\user1{v}\_1^\*, \ldots ,\user1{v}\_n^\*} \right\}\] can be transformed into a coordinate vector \[\user1{y}\] written relative to \[\left\{ {\user1{w}\_1^\*, \ldots ,\user1{w}\_m^\*} \right\}\] by the following \[\user1{y = }{\user1{Q}\_{\user1{w} \to {\user1{w}^\*}}}\user1{T}{\user1{P}\_{{\user1{v}^\*} \to \user1{v}}}\user1{x}\].
* \*\*Storing Data:\*\* We can use a matrix to store data such as pixel data from images. Matrix factorization methods can be used to perform dimensionality reduction.
* \*\*Difference Equations\*\* For some classes of difference equations and recurrence equations, the state transitions between steps can be represented using a matrix. General solutions to such equations can be found through eigenvalue analysis.
* \*\*Differential Equations:\*\* The relationship between the derivatives in some linear differential equations can be represented using a matrix. Eigenvalue analysis can be used to find solutions to these equations.
* \*\*Markov Chains:\*\* The state transition probabilities can be represented in a matrix. Eigenvalue analysis can be used to identify the long-term state probabilities
* \*\*Graphs:\*\* We can encode the structure of a graph in an adjacency matrix.

Now that we have seen some of the contexts in which matrices are used, an important question arises: how should we visualize the action of a matrix? Here are some principals to guide our intuition:

* \*\*SVD:\*\* One of the most informative ways to visualize the action of a matrix is to consider the singular value decomposition (SVD) theorem. This theorem states that any matrix can be written as a sequence of three operations: 1) rotation, 2) scaling, 3) rotation. Hence, we can interpret any linear transformation represented by a matrix as a sequence of these three composite transformations. For example, consider the transformation of a hypercube. The action of any matrix will first rotate the hypercube, then scale the hypercube which produces a stretched parallelotope, and then rotate the stretched parallelotope. Similarly, the action of any matrix on a hypersphere will first rotate the hypersphere, then scale it which will produce an ellipsoid, then rotate it again, producing a rotated (and potentially flattened) ellipsoid.
* \*\*Determinants:\*\* The determinant of a matrix tells us how much the linear transformation represented by a matrix changes volumes. As shown above with the SVD, the action of any matrix will take a hypersphere and produce a rotated ellipsoid. The ratio of the volume of the resulting ellipsoid to the original hypersphere is given by the determinant. Note that if the determinant is zero, it means that at least one of the dimensions of the ellipsoid has been collapsed. The sign of the determinant tells us whether the transformation preserves or reverses orientation.
* \*\*Four Fundamental Subspaces:\*\* Every \[m\,\, \times \,\,n\] matrix \[\user1{A}\] has four fundamental subspaces: 1) the column space defined as the subspace of \[{\mathbb{R}^m}\] spanned by the columns of the matrix, 2) the row space defined as the subspace of \[{\mathbb{R}^n}\] spanned by the rows of the matrix, 3) the nullspace defined as the subspace of \[{\mathbb{R}^n}\] orthogonal to the row space, and 4) the left nullspace defined as the subspace of \[{\mathbb{R}^m}\] orthogonal to the column space. Hence the combination of the column space and left nullspace is the complete \[{\mathbb{R}^m}\] and the combination of the row space and the nullspace is the complete \[{\mathbb{R}^n}\]. We can visualize each of these subspaces as being a linear manifold inside their respective parent space.
* \*\*Linear Transformations:\*\* See the section on linear transformations described above for more detail. Essentially, we can interpret each column of a general matrix \[\user1{A}\] as expressing the coordinates for a transformed input basis vector in terms of the output basis vectors. Therefore, for the transformation \[\user1{w = Av}\], we can interpret \[\user1{v}\] as representing an input coordinate vector relative to the basis \[\left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\], and \[\user1{w}\] as representing an output transformed coordinate vector relative to the basis \[\left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_m}} \right\}\]. The choice of basis for \[V\] and \[W\] is arbitrary but will change the matrix used to represent the linear transformation.

# Important Definitions

In this section, we will examine the definitions of several important concepts in linear algebra.

## Important Theorems

Gilbert Strang proposes that there are six great theorems of linear algebra (as seen [here]( https://math.mit.edu/~gs/linearalgebra/linearalgebra5\_6Great.pdf)). Some of these theorems are difficult to prove, so I will list the theorems without proof:

* \*\*Dimension Theorem:\*\* All bases for a vector space have the same number of vectors.
* \*\*Counting Theorem:\*\* Dimension of column space + dimension of nullspace = number of columns.
* \*\*Rank Theorem:\*\* Dimension of column space = dimension of row space. This is the rank.
* \*\*Fundamental Theorem:\*\* The row space and nullspace of an \[m\,\, \times \,\,n\] matrix \[\user1{A}\] are orthogonal complements in \[{\mathbb{R}^n}\].
* \*\*SVD:\*\* There are orthonormal basis (\[{\user1{u}\_1}, \ldots ,{\user1{u}\_r}\] for the column space and \[{\user1{v}\_1}, \ldots ,{\user1{v}\_r}\] for the row space) such that \[\user1{A}{\user1{v}\_i} = {\sigma \_i}{\user1{u}\_i}\].
* \*\*Spectral Theorem:\*\* If \[{\user1{A}^T} = \user1{A}\] then there are orthonormal \[{\user1{q}\_1}, \ldots ,{\user1{q}\_n}\] so that \[\user1{A}{\user1{q}\_i} = {\lambda \_i}{\user1{q}\_i}\] and \[\user1{A} = \user1{Q}\Lambda {\user1{Q}^T}\].

## Rank

The rank of a matrix \[\user1{A}\] is defined as the number of linearly independent columns of \[\user1{A}\]. In other words, the rank is the dimension of the column space of \[\user1{A}\]. To compute the rank of a matrix \[\user1{A}\], an effective strategy is to first transform \[\user1{A}\] into its reduced row echelon form \[\user1{R}\] and determine the number of pivot columns which corresponds to the rank. An interesting and nonobvious fact is that the column rank (number of independent columns) equals the row rank (number of independent rows) for any matrix \[\user1{A}\].

How can we show that the column rank equals the row rank for any matrix? Here are three different ways to see this:

1. \*\*Simplest Proof:\*\* The simplest proof to show that the row rank is equal to the column rank is the following. Suppose that we have any \[m\] by \[n\] matrix \[\user1{A}\], with column rank \[c\] and row rank \[r\]. The fact that the column rank equals \[c\] tells us that we can describe the \[n\] columns of \[\user1{A}\] using linear combinations of \[c\] vectors. If we put the \[c\] vectors in an \[m\] by \[c\] matrix \[\user1{C}\], and put the corresponding combination coefficients in a \[c\] by \[n\] matrix \[\user1{R}\], then we can write \[\user1{A}\] as \[\user1{A = CR}\]. The key insight for this proof is the fact that we can interpret the matrix multiplication \[\user1{CR}\] as a linear combination of the columns of \[\user1{C}\], OR we can interpret the matrix multiplication as a linear combination of the rows \[\user1{R}\]. Therefore, since \[\user1{R}\] is a \[c\] by \[n\] matrix, it follows that the rows of \[\user1{A}\] can be written as a linear combination of the rows in \[\user1{R}\]. Hence, the row rank of \[\user1{A}\] must be less than or equal to the number of rows in \[\user1{R}\], such that \[r \le c\]. We can repeat this process in the reverse order using the fact that the row rank equals \[r\] tells us that we can describe the \[m\] rows of \[\user1{A}\] using linear combinations of \[r\] vectors. We can put these \[r\] vectors in the rows of \[\user1{R}\] and repeat the proof process described above to show that \[c \le r\]. Therefore, we can equate these expressions to show that \[r = c\].
2. \*\*Information Theory Interpretation:\*\* Another intuitive “hand wavy” interpretation of why the dimensionality of the row space equals the column space can be framed in terms of information capacity. In this way, the rank tells you the information capacity of the channels (i.e. row space or column space) provided by the matrix. Simply put, if I give you an ordered list of rows in a matrix, you have all the information necessary to tell me what the columns are (here we are implicitly using the relationship between the rows and columns imposed by the definition of a matrix). Therefore, the columns are completely described by the information contained in the rows. Therefore, the column rank should be less than or equal to the row rank. Similarly, if I give you an ordered list of columns in a matrix, you have all the information necessary to tell me what the rows are. Therefore, the rows are completely described by the information contained in the columns. Therefore, the row rank should be less than or equal to the column rank. Hence, the column rank should equal the row rank. Continuing with this information theory interpretation, suppose that we have an \[m\] by \[n\] matrix \[\user1{A}\], where \[m > n\]. In this case, the vectors in the column space are in \[{\mathbb{R}^m}\] which have higher dimensionality than the vectors in the row space which are in \[{\mathbb{R}^n}\]. How is it possible for the row space and the column space to have the same information capacity if the vectors in their respective spaces have different dimensions? The key answer to this question is that the information capacity of the row space and column space depend on the dimensionality of the subspace, not the parent space. Hence, even though \[m > n\], the subspace dimensionality for the row space and column space are equal. This leads to a key insight into vector spaces, the information capacity of the space does not depend on whether the actual vectors in the space are in \[{\mathbb{R}^2}\], \[{\mathbb{R}^{100}}\], or some other abstract vector quantity. The information capacity depends on the subspace dimensionality. However, with this said, the subspace dimensionality has an upper bound described by the dimensionality of the parent space. Hence, a subspace of \[{\mathbb{R}^2}\] can have a maximum dimension of 2, while a subspace of \[{\mathbb{R}^{100}}\] can have a maximum dimension of 100.
3. \*\*Specific 2 by 2 Example:\*\* Suppose we ask the question, is it possible for a matrix to have more linearly independent columns than rows. Let us take the example of a 2 by 2 matrix \[\user1{A} = \left[ {\begin{array}{\*{20}{c}}a&b\\c&d\end{array}} \right]\]. Suppose that there exist scalars such that \[{x\_1}\left[ {\begin{array}{\*{20}{c}}a\\c\end{array}} \right] + {x\_2}\left[ {\begin{array}{\*{20}{c}}b\\d\end{array}} \right] = 0\]. Solving for these scalers gives \[{x\_2}\left( {\frac{{ad - bc}}{a}} \right) = 0\] which implies that \[ad - bc = 0\]. Now turning our attention to the rows, the question we are asking is whether we can find \[{y\_1},\,{y\_2}\] such that \[{y\_1}\left[ {\begin{array}{\*{20}{c}}a&b\end{array}} \right] + {y\_2}\left[ {\begin{array}{\*{20}{c}}c&d\end{array}} \right] = 0\]. We find that setting \[{y\_1} = d\] and \[{y\_2} = - b\] will solve this equation. Hence, if the columns are not linearly independent, then the rows cannot be linearly independent.

## Reduced Row Echelon Form

For any \[m\] by \[n\] matrix \[\user1{A}\] with rank \[r\], we can transform \[\user1{A}\] into what is called the \*reduced row echelon form\* such that \[{\user1{E}\_K} \cdots {\user1{E}\_1}\user1{A}{\user1{Q}\_1} \cdots {\user1{Q}\_L}\user1{ = R}\], where \[{\user1{E}\_K} \cdots {\user1{E}\_1}\] represent a series of invertible elementary row operations (i.e. swap rows, scale rows, add multiple of rows), and \[{\user1{Q}\_1} \cdots {\user1{Q}\_L}\] represent a series of column permutations. The reduced row echelon matrix takes the form \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}&{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,r}}}&{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\]. Where \[{\user1{I}\_{r,r}}\] is an \[r\,\, \times \,\,r\] identity matrix corresponding to the pivot columns, and \[{\user1{F}\_{r,n - r}}\] is an \[r\] by \[n - r\] matrix containing the free variable columns. The reduced row echelon matrix plays an important role in understanding the four fundamental subspaces for \[\user1{A}\].

## Linear Independence and Basis

What is linear independence? Suppose that we have vectors \[{\user1{x}\_1},{\user1{x}\_2}, \ldots ,{\user1{x}\_n}\]. We say that these vectors are independent if no linear combination gives the zero vector. In other words, it is impossible to pick some nontrivial (i.e. not all zero) coefficients to make the linear combination equal the zero vector \[{c\_1}{\user1{x}\_1}\user1{ + }{c\_2}{\user1{x}\_2}\user1{ + } \cdots \user1{ + }{c\_n}{\user1{x}\_n} \ne \user1{0}\].

How can we determine whether a set of vectors \[{\user1{x}\_1},{\user1{x}\_2}, \ldots ,{\user1{x}\_n}\] is independent? One way is to construct a matrix \[\user1{A}\] with \[{\user1{x}\_1},{\user1{x}\_2}, \ldots ,{\user1{x}\_n}\] as the columns, and then determine whether there are any nonzero entries in the nullspace. We can achieve this by computing the reduced row echelon form of \[\user1{A}\] and determining whether we have any free variables.

We say that vectors \[{\user1{x}\_1},{\user1{x}\_2}, \ldots ,{\user1{x}\_l}\] span a space if the space consists of all linear combinations of those vectors. In other words, if \[{\user1{x}\_1},{\user1{x}\_2}, \ldots ,{\user1{x}\_l}\] span a space \[S\], then \[S\] is the smallest space containing all linear combinations of those vectors. A basis for a vector space is a sequence of vectors \[{\user1{x}\_1},{\user1{x}\_2}, \ldots ,{\user1{x}\_d}\] with two properties: 1) \[{\user1{x}\_1},{\user1{x}\_2}, \ldots ,{\user1{x}\_d}\] are independent, and 2) \[{\user1{x}\_1},{\user1{x}\_2}, \ldots ,{\user1{x}\_d}\] span the vector space. The basis of a space provides everything we need to know about that space.

How can we test whether a set of vectors \[{\user1{x}\_1},{\user1{x}\_2}, \ldots ,{\user1{x}\_d}\] are a basis for \[{\mathbb{R}^d}\]? We can construct a matrix \[\user1{A}\] with \[{\user1{x}\_1},{\user1{x}\_2}, \ldots ,{\user1{x}\_d}\] as the columns and compute the reduced row echelon matrix. If all columns are independent, then we will have no free variables so the column rank \[r = d\]. To see this, remember that any matrix can be written in reduced row echelon form as \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}&{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,r}}}&{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\]. We clearly see that any column in \[\left[ {\begin{array}{\*{20}{c}}{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\] can be written as a linear combination of the columns in \[\left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}\\{{\user1{0}\_{m - r,r}}}\end{array}} \right]\] since we have the identity matrix \[{\user1{I}\_{r,r}}\] up top. Hence, we only have \[r\] linearly independent columns. Note also, that we only have \[r\] linearly independent rows, since rows filled with zeros are not independent.

Now, to ensure that the vectors span the complete space, we can look at whether \[\user1{Ay = b}\] is solvable for every \[\user1{b}\] in \[{\mathbb{R}^d}\]. To be solvable for every \[\user1{b}\], we cannot have a row of zeros at the bottom of the reduced row echelon matrix. Therefore, to be a basis, the reduced row echelon form must be the identity matrix.

Suppose we now want to test whether a set of vectors \[{\user1{x}\_1},{\user1{x}\_2}, \ldots ,{\user1{x}\_d}\] are a basis for a subspace \[{\mathbb{R}^n}\]. We can repeat each of the steps described above, the only difference is that when examining whether these vectors span the subspace, we are only interested in determining whether \[\user1{Ay = b}\] is solvable for \[\user1{b}\] in the subspace. For a given subspace, every basis for that space has the same number of vectors. We call this number the dimension of the subspace.

## Pseudoinverse

If we have a matrix \[\user1{A}\] that has full column rank but not full row rank, then we know that \[{\user1{A}^T}\user1{A}\] is invertible since the nullspace is empty (except for the zero vector). To see why, recall that full column rank implies that all columns are linearly independent. Therefore, by the definition of linear independence, \[\user1{Ax = 0}\] only has the trivial solution and the nullspace only contains the zero vector. Furthermore, we see that the nullspace of \[{\user1{A}^T}\user1{A}\] is equal to the nullspace of \[\user1{A}\] since \[{\user1{A}^T}\user1{Ax} = \user1{0} \Rightarrow {\user1{x}^T}{\user1{A}^T}\user1{Ax} = 0 \Rightarrow {\left\| {\user1{Ax}} \right\|^2} = 0\]. Hence it follows that the nullspace of \[{\user1{A}^T}\user1{A}\] is empty and \[{\user1{A}^T}\user1{A}\] is invertible since it is square and must have full rank for the nullspace of \[{\user1{A}^T}\user1{A}\] to be empty. Therefore, we see that \[{\left( {{\user1{A}^T}\user1{A}} \right)^{ - 1}}{\user1{A}^T}\] is a left inverse of \[\user1{A}\] since \[{\left( {{\user1{A}^T}\user1{A}} \right)^{ - 1}}{\user1{A}^T}\user1{A} = \user1{I}\].

Similarly, if we have a matrix \[\user1{A}\] that has full row rank but not full column rank, then we know that \[\user1{A}{\user1{A}^T}\] is invertible Therefore, we see that \[{\user1{A}^T}{\left( {\user1{A}{\user1{A}^T}} \right)^{ - 1}}\] is a right inverse of \[\user1{A}\] since\[\user1{A}{\user1{A}^T}{\left( {\user1{A}{\user1{A}^T}} \right)^{ - 1}} = \user1{I}\].

Now suppose that a matrix does not have full row rank nor full column rank. Since the column space and the row space have the same dimension (column rank equals row rank) there should be a mapping between the two. The pseudoinverse is the invertible transformation from the row space to the column space.

If \[\user1{x} \ne \user1{y}\] are in the row space, then \[\user1{Ax} \ne \user1{Ay}\]. To see this, suppose that \[\user1{Ax = Ay}\], then it follows that \[\user1{A}\left( {\user1{x} - \user1{y}} \right) = \user1{0}\] which implies that \[\user1{x} - \user1{y}\] is in the nullspace. However, vectors in the nullspace are orthogonal to vectors in the row space, so \[\user1{x} - \user1{y} = \user1{0}\]. However, we know that \[\user1{x} \ne \user1{y}\], thus, \[\user1{Ax} \ne \user1{Ay}\].

How can we find the pseudoinverse \[{\user1{A}^ + }\]? We can compute the SVD which gives \[\user1{A} = \user1{U}\Sigma {\user1{V}^T}\]. By taking the reciprocals of the nonzero elements of \[\Sigma \] we get \[{\Sigma ^ + }\] which enables the computation of the pseudoinverse as \[{\user1{A}^ + } = \user1{V}{\Sigma ^ + }{\user1{U}^T}\].

Let’s dig into why the formula for the pseudoinverse \[{\user1{A}^ + } = \user1{V}{\Sigma ^ + }{\user1{U}^T}\] makes sense. Suppose that the \[m\,\, \times \,\,n\] matrix \[\user1{A}\] has rank \[r\]. We can write the pseudoinverse as a sum of rank 1 matrices as \[{\user1{A}^ + } = \frac{1}{{{\sigma \_1}}}{\user1{v}\_1}\user1{u}\_1^T + \cdots + \frac{1}{{{\sigma \_r}}}{\user1{v}\_r}\user1{u}\_r^T + 0{\user1{v}\_{r + 1}}\user1{u}\_{r + 1}^T + \cdots + 0{\user1{v}\_M}\user1{u}\_M^T\] where \[M = \min \left\{ {m,n} \right\}\]. Suppose that we have a vector in \[{\mathbb{R}^m}\] which we can write as \[\user1{x} = {c\_1}{\user1{u}\_1} + \cdots + {c\_m}{\user1{u}\_m}\] (since the columns of \[\user1{U}\]provide an orthonormal basis for \[{\mathbb{R}^m}\]). Multiplying by the pseudoinverse and using orthonormality to simplify we get:

\[\begin{array}{l}{\user1{A}^ + }\user1{x} = \left( {\frac{1}{{{\sigma \_1}}}{\user1{v}\_1}\user1{u}\_1^T + \cdots + \frac{1}{{{\sigma \_r}}}{\user1{v}\_r}\user1{u}\_r^T + 0{\user1{v}\_{r + 1}}\user1{u}\_{r + 1}^T + \cdots + 0{\user1{v}\_M}\user1{u}\_M^T} \right)\left( {{c\_1}{\user1{u}\_1} + \cdots + {c\_m}{\user1{u}\_m}} \right)\\ = \frac{{{c\_1}}}{{{\sigma \_1}}}{\user1{v}\_1} + \cdots + \frac{{{c\_r}}}{{{\sigma \_r}}}{\user1{v}\_r}\end{array}\]

Since \[{\user1{v}\_1}, \ldots ,{\user1{v}\_r}\] provides an orthonormal basis for the row space, we can interpret the action of the pseudoinverse \[{\user1{A}^ + }\user1{x}\] as a two-step procedure: 1) computing the projection of \[\user1{x}\] onto the column space of \[\user1{A}\], and 2) mapping this column space projection to a vector in the row space. Hence, the pseudoinverse finds the vector in the row space \[\user1{v}\] which maps to the vector \[{\user1{x}^\*} = \user1{Av}\] in the column space such that \[{\user1{x}^\*}\] is closest to \[\user1{x}\]. To see why, consider computing \[\user1{A}{\user1{A}^ + }\user1{x}\]. If we expand \[\user1{A}\] using its rank-1 SVD representation we can write

\[\begin{array}{l}\user1{A}{\user1{A}^ + }\user1{x = }\left( {{\sigma \_1}{\user1{u}\_1}\user1{v}\_1^T + \cdots + {\sigma \_r}{\user1{u}\_r}\user1{v}\_r^T + 0{\user1{u}\_{r + 1}}\user1{v}\_{r + 1}^T + \cdots + 0{\user1{u}\_M}\user1{v}\_M^T} \right)\left( {\frac{{{c\_1}}}{{{\sigma \_1}}}{\user1{v}\_1} + \cdots + \frac{{{c\_r}}}{{{\sigma \_r}}}{\user1{v}\_r}} \right)\\ = {c\_1}{\user1{u}\_1} + \cdots + {c\_r}{\user1{u}\_r}\end{array}\]

Notice that the vector \[{c\_1}{\user1{u}\_1} + \cdots + {c\_r}{\user1{u}\_r}\] is the projection of \[\user1{x} = {c\_1}{\user1{u}\_1} + \cdots + {c\_m}{\user1{u}\_m}\] onto the column space of \[\user1{A}\]. To summarize: The pseudoinverse in enabled by the SVD theorem which tells us that there exist basis vectors for the row and column space of \[\user1{A}\] such that \[\user1{A}{\user1{v}\_k} = {\sigma \_k}{\user1{u}\_k}\]. The pseudoinverse \[{\user1{A}^ + }\user1{x}\] provides the coordinate vector in the row space of \[\user1{A}\] (relative to the basis \[{\user1{v}\_1}, \ldots ,{\user1{v}\_r}\]) that maps to the projection of \[\user1{x}\] onto the column space (relative to the basis \[{\user1{u}\_1}, \ldots ,{\user1{u}\_r}\]).

## Projections

We call the closest point to \[\user1{b}\] along the line \[c\user1{a}\] the projection of \[\user1{b}\] onto \[\user1{a}\]. To find the projection of a vector \[\user1{b}\] onto a vector \[\user1{a}\], we can compute the error term as being the difference between \[\user1{b}\] and the projected point written as \[\user1{e = b} - c\user1{a}\]. At the closest point, the error term will be orthogonal to \[\user1{a}\]. To see why, think about the error term as being composed of two components: 1) a component orthogonal to \[\user1{a}\], and 2) a component parallel to \[\user1{a}\]. To minimize the error term, we need to minimize the component parallel to \[\user1{a}\]. When the parallel component is 0, then what we have left is the orthogonal component. Therefore, at the closest point, the error term must be orthogonal to \[\user1{a}\] which gives us the equation \[{\user1{a}^T}\user1{e} = {\user1{a}^T}\left( {\user1{b} - c\user1{a}} \right) = 0\]. Solving for \[c\] we get \[c = \frac{{{\user1{a}^T}\user1{b}}}{{{\user1{a}^T}\user1{a}}}\]. Hence the projection of \[\user1{b}\] onto \[\user1{a}\] is \[c\user1{a = a}\frac{{{\user1{a}^T}\user1{b}}}{{{\user1{a}^T}\user1{a}}} = \frac{{\user1{a}{\user1{a}^T}}}{{{\user1{a}^T}\user1{a}}}\user1{b}\]. Therefore, we can think about the projection as being performed by a matrix where \[\user1{P} = \frac{{\user1{a}{\user1{a}^T}}}{{{\user1{a}^T}\user1{a}}}\]. We can multiply the matrix \[\user1{b}\] by \[\user1{P}\] to get the projection onto \[\user1{a}\].

Examining the projection matrix, we see that the column space of \[\user1{P}\] is the combination of columns of \[\user1{P} = \frac{{\user1{a}{\user1{a}^T}}}{{{\user1{a}^T}\user1{a}}}\]. Notice that \[\user1{a}{\user1{a}^T}\] is rank 1 since \[\user1{a}\] is a vector. Therefore, the columns of \[\user1{P}\] are multiples of \[\user1{a}\]. So the column space of \[\user1{P}\] can be visualized as multiples of the vector \[\user1{a}\]. We also see that \[{\user1{P}^T} = \user1{P}\], so the matrix is symmetric. Furthermore, since the projection matrix will project onto \[\user1{a}\], then it follows that if we project again onto \[\user1{a}\] then we will get the same point, so \[{\user1{P}^2} = \user1{P}\].

For many equations, \[\user1{Ax = b}\] there does not exist a solution. Therefore, our goal is to find the solution to \[\user1{A\hat x = p}\], where \[\user1{p}\] is a projection of \[\user1{b}\] onto the column space of \[\user1{A}\]. We can compute the projection of a point \[\user1{b}\] onto the column space of \[\user1{A}\] by first computing the error measured as the distance between \[\user1{b}\] and the projected point \[\user1{p}\]. We can compute the projected point as \[\user1{A\hat x = p}\]. Now for the error vector \[\user1{e = b} - \user1{p}\] to be orthogonal to the columns of \[\user1{A}\], the error vector must be in the left nullspace of \[\user1{A}\] such that \[{\user1{A}^T}\user1{e = 0}\]. We can expand this equation as \[{\user1{A}^T}\user1{A\hat x = }{\user1{A}^T}\user1{b}\]. Therefore, the solution is \[\user1{\hat x} = {\left( {{\user1{A}^T}\user1{A}} \right)^{ - 1}}{\user1{A}^T}\user1{b}\] and we can write the projection vector as \[\user1{p = A\hat x = A}{\left( {{\user1{A}^T}\user1{A}} \right)^{ - 1}}{\user1{A}^T}\user1{b}\]. Therefore, the projection matrix is \[\user1{P = A}{\left( {{\user1{A}^T}\user1{A}} \right)^{ - 1}}{\user1{A}^T}\]. Notice that if \[\user1{A}\] is a square invertible matrix, then the projection matrix becomes \[\user1{P = A}{\left( {{\user1{A}^T}\user1{A}} \right)^{ - 1}}{\user1{A}^T} = \user1{A}{\user1{A}^{ - 1}}{\user1{A}^{ - T}}{\user1{A}^T} = \user1{I}\]. We see that \[{\user1{P}^T} = \user1{P}\] and \[{\user1{P}^2} = \user1{P}\]. Note that if \[\user1{b}\] is in the column space then \[\user1{Pb = b}\], while if \[\user1{b}\] is orthogonal to the column space then \[\user1{Pb = 0}\].

If \[\user1{P}\] is a projection matrix onto the column space, then \[\user1{I} - \user1{P}\] is a projection matrix onto the left nullspace. This is because we can view any vector \[\user1{b}\] as being a combination of a component in the column space and a component in the left nullspace. Therefore, since \[\user1{Pb}\] gives us the column space component of \[\user1{b}\], then \[\user1{b} - \user1{Pb = }\left( {\user1{I} - \user1{P}} \right)\user1{b}\] gives us the projection onto the left nullspace.

In least squares the goal is to minimize \[{\left\| {\user1{Ax} - \user1{b}} \right\|^2} = {\left\| \user1{e} \right\|^2}\]. We can look at the least squares problem in two ways. The first way is to consider the errors produced by each equation which is contained in the individual elements of \[\user1{e}\]. For example, \[{\left\| \user1{e} \right\|^2} = e\_1^2 + \cdots + e\_m^2\]. These errors correspond to the squared difference between the prediction line and the training data. We can use calculus to minimize these errors by setting the partial derivatives to 0.

The second way to consider the column space as representing a space for the parameters. Each point in the column space represents a set of output predictions given the input data. Therefore, we can take our training data and project the desired outputs onto the column space to find the set of output predictions which are closest. Then we can determine the set of parameters which produce these predictions in the column space. The set of equations to solve to determine these parameters are called the normal equations written as \[{\user1{A}^T}\user1{Ax = }{\user1{A}^T}\user1{b}\].

When is \[{\user1{A}^T}\user1{A}\] invertible? If \[\user1{A}\] has independent columns, then \[{\user1{A}^T}\user1{A}\] is invertible. To show this, suppose that \[{\user1{A}^T}\user1{Ax = 0}\]. This implies that \[\user1{Ax}\] must be orthogonal to the columns of \[\user1{A}\]. But by definition, \[\user1{Ax}\] is a combination of the columns of \[\user1{A}\]. Therefore, we must have \[\user1{Ax = 0}\]. Since we postulate that the columns of \[\user1{A}\] are independent, then \[\user1{x = 0}\] which implies that \[{\user1{A}^T}\user1{A}\] is invertible. Another way to see this is to take the dot product with \[\user1{x}\] such that \[{\user1{x}^T}{\user1{A}^T}\user1{Ax = }{\left\| {\user1{Ax}} \right\|^2}\user1{ = 0} \Rightarrow \user1{Ax = 0}\].

# Four Fundamental Subspaces

A subspace is a vector space that is contained inside of another vector space. For example, a line in \[{\mathbb{R}^2}\] that passes through the origin is a vector space since any linear combination of vectors along this line will also be on the line. Furthermore, a line in \[{\mathbb{R}^2}\] is a subspace of the \[{\mathbb{R}^2}\] vector space since every vector on the line is also in \[{\mathbb{R}^2}\]. The total possible subspaces of \[{\mathbb{R}^2}\] are: 1) all of \[{\mathbb{R}^2}\], 2) any line in \[{\mathbb{R}^2}\] that passes through the origin, 3) the zero vector. Two subspaces \[S\] and \[T\] are said to be orthogonal if every vector in \[S\] is orthogonal to every vector in \[T\].

We have four fundamental subspaces associated with any \[m\] by \[n\] matrix \[\user1{A}\]:

1. \*\*The column space \[C\left( \user1{A} \right)\]:\*\* is a subspace of \[{\mathbb{R}^m}\] and contains all linear combinations of the columns of \[\user1{A}\]. The \[r\] pivot columns from the reduced row echelon matrix for \[\user1{A}\] form a basis for \[C\left( \user1{A} \right)\]. Therefore, the dimension of the subspace is \[\dim \,\,C\left( \user1{A} \right) = r\]. All vectors in \[C\left( \user1{A} \right)\] are orthogonal to vectors in the left nullspace \[N\left( {{\user1{A}^T}} \right)\]. As an example, suppose we have a 3 by 3 matrix \[\user1{A}\] where each of the column vectors is in \[{\mathbb{R}^3}\]. We can interpret each column vector as describing a line in \[{\mathbb{R}^3}\] that passes through the origin. Therefore, each column vector is itself a subspace of \[{\mathbb{R}^3}\]. If we consider the linear combination of all columns of \[\user1{A}\] then we get a subspace of \[{\mathbb{R}^3}\] that we call the column space of \[\user1{A}\], written as \[C\left( \user1{A} \right)\].
2. \*\*The nullspace \[N\left( \user1{A} \right)\]:\*\* is a subspace of \[{\mathbb{R}^n}\] and contains the vectors that are the solutions to the equation\[\user1{Ax = 0}\]. From the reduced row echelon form of \[\user1{A}\], written as \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}&{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,r}}}&{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\], we can solve for a basis of the nullspace as the columns of \[\user1{M} = \left[ {\begin{array}{\*{20}{c}}{ - {\user1{F}\_{r,n - r}}}\\{{\user1{I}\_{n - r,n - r}}}\end{array}} \right]\] where \[{\user1{F}\_{r,n - r}}\] are the free variable coefficients in the reduced row echelon form of \[\user1{A}\]. Hence, we see that the dimension of the null space is equal to the number of free variables \[\dim \,\,N\left( \user1{A} \right) = n - r\]. To illustrate why the dimension of the nullspace of \[\user1{A}\] must be \[n - r\], consider the following argument. If the rank of \[\user1{A}\] is \[r\], then there are \[r\] linearly independent rows of \[\user1{A}\]. Hence, to solve \[\user1{Ax} = 0\], we have \[r\] equations (one for each linearly independent row), and \[n\] variables in \[\user1{x}\]. Now consider the thought experiment of assigning arbitrary values to the first \[n - r\] variables in \[\user1{x}\]. Under these conditions, we have \[r\] variables remaining in \[\user1{x}\] that can be used to solve the system uniquely. Hence, we have \[n - r\]degrees of freedom in the nullspace which amounts to having dimensionality \[n - r\].
3. \*\*The row space \[C\left( {{\user1{A}^T}} \right)\]:\*\* is a subspace of \[{\mathbb{R}^n}\] and contains all linear combinations of the rows of \[\user1{A}\]. The reduced row echelon form of \[\user1{A}\] can be written as \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}&{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,r}}}&{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\]. Therefore, we see that there are \[r\] nonzero rows. Recall that we computed \[\user1{R}\] by performing invertible row and column operations on \[\user1{A}\] such that \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{AQ = R}\]. Hence we see that \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{A} = \user1{R}{\user1{Q}^{ - 1}}\] where \[\user1{Q = }{\user1{Q}^{ - 1}}\] is a column permutation operation matrix (i.e. the inverse of swapping two rows is to apply the same operation again and swap them back) and only affects the order of the elements in rows of \[\user1{R}\]. Hence, we see that by just performing elementary row operations \[{\user1{E}\_M} \cdots {\user1{E}\_1}\] we produced the matrix \[\user1{R}\] which has \[m - r\] rows filled with zeros. Therefore, we see that \[\user1{R}\] has \[r\] independent rows and thus, \[\user1{A}\] has \[r\] independent rows. Hence, the first \[r\] rows of \[\user1{R}\] form a basis for the row space of \[\user1{A}\].
4. \*\*The left nullspace \[N\left( {{\user1{A}^T}} \right)\]:\*\* is a subspace of \[{\mathbb{R}^m}\] and contains the vectors that are the solution to the equation \[{\user1{A}^T}\user1{x} = \user1{xA = 0}\]. Since the rank of \[{\user1{A}^T}\] is \[r\], then the number of free columns of \[{\user1{A}^T}\] must be \[m - r\]. Therefore, the dimension of the left nullspace must be \[\dim \,\,N\left( {{\user1{A}^T}} \right) = m - r\]. We can write the reduced row echelon matrix as \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{AQ = R}\]. Rearranging, we see that\[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{A = R}{\user1{Q}^{ - 1}}\]. Since the reduced row echelon matrix has the form \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}&{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,r}}}&{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\] and \[\user1{Q}\] is a column permutation matrix, \[\user1{Q}\] will have no effect on the \[m - r\] bottom rows filled with zeros. We can interpret \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{A}\] as a linear combination of the rows of \[\user1{A}\] where the combination coefficients are the values in \[{\user1{E}\_M} \cdots {\user1{E}\_1}\]. Hence, the coefficients that produce rows filled with zeros are the last \[m - r\] rows of \[{\user1{E}\_M} \cdots {\user1{E}\_1}\]. Additionally, we know the rows of \[{\user1{E}\_M} \cdots {\user1{E}\_1}\] are independent because the elementary matrices are invertible. Therefore, the last \[m - r\] rows of \[{\user1{E}\_M} \cdots {\user1{E}\_1}\] form a basis for the left nullspace.

## Finding Basis Vectors for the Four Fundamental Subspaces

Suppose we have an \[m\] by \[n\] matrix \[\user1{A}\]. Our goal in this section is to demonstrate how we can find a set of basis vectors for each of the four fundamental subspaces.

\*\*Column Space:\*\*

To determine a basis for the column space of \[\user1{A}\], we can first calculate the reduced row echelon form \[\user1{EA = R}\], where \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}&{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,r}}}&{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\] and \[\user1{E}\] represents a series of invertible elementary row operations. Note that for simplicity we are assuming that no column permutations are needed. We see that the first \[r\] columns of \[\user1{R}\] form a basis for the column space of \[\user1{R}\] (due to the identity matrix in the top left). We can write each vector in the column space of \[\user1{A}\] as a linear combination of the columns of \[\user1{A}\], written as \[{c\_1}{\user1{a}\_1} + \cdots + {c\_n}{\user1{a}\_n}\] where \[{c\_1}, \ldots ,{c\_n}\] are scalars and \[{\user1{a}\_1}, \ldots ,{\user1{a}\_n}\] are the columns of \[\user1{A}\]. Now using the relationship with the reduced row echelon matrix, we can write

\[\begin{array}{l}{c\_1}{\user1{a}\_1} + \cdots + {c\_n}{\user1{a}\_n} = {c\_1}{\user1{E}^{ - 1}}{\user1{r}\_1} + \cdots + {c\_n}{\user1{E}^{ - 1}}{\user1{r}\_n}\\ = {\user1{E}^{ - 1}}\left( {{c\_1}{\user1{r}\_1} + \cdots + {c\_n}{\user1{r}\_n}} \right)\end{array}\]

Where \[{\user1{r}\_1}, \ldots ,{\user1{r}\_n}\] are the columns of \[\user1{R}\]. Notice that the combination \[{c\_1}{\user1{r}\_1} + \cdots + {c\_n}{\user1{r}\_n}\] is in the column space of \[\user1{R}\] and therefore can be written using the first \[r\] columns as basis vectors as \[{c\_1}{\user1{r}\_1} + \cdots + {c\_n}{\user1{r}\_n} = {d\_1}{\user1{r}\_1} + \cdots + {d\_r}{\user1{r}\_r}\], where \[{d\_1}, \ldots ,{d\_r}\] are scalars. Hence, we can pop this back into the equation above to get

\[\begin{array}{l}{c\_1}{\user1{a}\_1} + \cdots + {c\_n}{\user1{a}\_n} = {\user1{E}^{ - 1}}\left( {{d\_1}{\user1{r}\_1} + \cdots + {d\_r}{\user1{r}\_r}} \right)\\ = {d\_1}{\user1{E}^{ - 1}}{\user1{r}\_1} + \cdots + {d\_r}{\user1{E}^{ - 1}}{\user1{r}\_r}\\ = {d\_1}{\user1{a}\_1} + \cdots + {d\_r}{\user1{a}\_r}\end{array}\]

Hence, we have shown that the columns of \[\user1{A}\] corresponding to the pivot columns of \[\user1{R}\] can be used to describe any vector in the column space of \[\user1{A}\] and hence span the column space.

To show that this is the minimal spanning set, we must show that all of the \[{\user1{a}\_1}, \ldots ,{\user1{a}\_r}\] are independent. We can show this by contradiction. Suppose that there did exist scalar coefficients such that \[{c\_1}{\user1{a}\_1} + \cdots + {c\_r}{\user1{a}\_r} = 0\]. This implies that \[{c\_1}\user1{E}{\user1{a}\_1} + \cdots + {c\_r}\user1{E}{\user1{a}\_r} = {c\_1}{\user1{r}\_1} + \cdots + {c\_r}{\user1{r}\_r} = 0\] which is impossible since \[{\user1{r}\_1}, \ldots ,{\user1{r}\_r}\] are all independent (they come from the identity matrix in \[\user1{R}\]).

Therefore, the columns of \[\user1{A}\] corresponding to the pivot columns of \[\user1{R}\] span the column space of \[\user1{A}\] and the set is minimal since all these vectors are independent. Hence, these vectors form a basis for the column space of \[\user1{A}\].

\*\*Row Space:\*\*

To determine a basis for the row space of \[\user1{A}\], we can first calculate the reduced row echelon form \[\user1{EA = R}\], where \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}&{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,r}}}&{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\] and \[\user1{E}\] represents a series of invertible elementary row operations. Thus, we can write, \[\user1{A} = {\user1{E}^{ - 1}}\user1{R}\]. We can interpret this matrix product as a linear combination of the rows of \[\user1{R}\]. Since rows filled with zeros do not contribute to a basis, and the first \[r\] rows of \[\user1{R}\] are all independent (due to the identity matrix in the top left), we see that the first \[r\] rows of \[\user1{R}\] form a basis for the row space of \[\user1{A}\].

\*\*Nullspace:\*\*

To determine a basis for the null space of \[\user1{A}\], we can first calculate the reduced row echelon form \[\user1{EA = R}\], where \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}&{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,r}}}&{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\] and \[\user1{E}\] represents a series of invertible elementary row operations. We can show that any solution \[\user1{x}\] to \[\user1{Ax = 0}\] will also be a be a solution to \[\user1{Rx = 0}\].

\[\begin{array}{l}\user1{Ax = 0}\\ \Rightarrow \user1{EAx = E0 = 0}\\ \Rightarrow \user1{Rx = 0}\end{array}\]

Therefore, every element in the nullspace of \[\user1{A}\] is also in the nullspace of \[\user1{R}\], such that \[N\left( \user1{A} \right) \subseteq N\left( \user1{R} \right)\]. We can also apply this logic in reverse:

\[\begin{array}{l}\user1{Rx = 0}\\ \Rightarrow {\user1{E}^{ - 1}}\user1{Rx = }{\user1{E}^{ - 1}}\user1{0 = 0}\\ \Rightarrow \user1{Ax = 0}\end{array}\]

Therefore, every element in the nullspace of \[\user1{R}\] is also in the nullspace of \[\user1{A}\], such that \[N\left( \user1{R} \right) \subseteq N\left( \user1{A} \right)\]. Therefore, the nullspace of \[\user1{R}\] equals the nullspace of \[\user1{A}\], \[N\left( \user1{R} \right) = N\left( \user1{A} \right)\].

Hence, to find a basis for the nullspace of \[\user1{A}\], we can instead find a basis for the nullspace of \[\user1{R}\]. We can view the pivot variables as being constrained for each equation in order to make the equation equal to 0. Therefore, the dimension of the nullspace is equal to the number of free variables which is \[n - r\]. We can find \[n - r\] independent solutions to the equation \[\user1{RM = 0}\] by noticing that \[\user1{RM = }\left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}&{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,r}}}&{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{ - {\user1{F}\_{r,n - r}}}\\{{\user1{I}\_{n - r,n - r}}}\end{array}} \right]\]. Therefore, \[\left[ {\begin{array}{\*{20}{c}}{ - {\user1{F}\_{r,n - r}}}\\{{\user1{I}\_{n - r,n - r}}}\end{array}} \right]\] forms a basis for the nullspace because it contains \[n - r\] independent vectors (due to the identity matrix).

\*\*Left Nullspace:\*\*

To determine a basis for the left nullspace of \[\user1{A}\], we can first calculate the reduced row echelon form \[\user1{EA = R}\], where \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}&{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,r}}}&{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\] and \[\user1{E}\] represents a series of invertible elementary row operations. We know that the row rank is \[r\], therefore, the number of free columns of \[{\user1{A}^T}\] must be \[m - r\] and the dimension of the left nullspace must be \[m - r\]. The left nullspace is defined as the solutions to \[{\user1{A}^T}\user1{x} = \user1{0}\] which is equivalent (by taking the transpose) to \[{\user1{x}^T}\user1{A} = \user1{0}\]. Notice that in the result of the multiplication \[\user1{EA = R}\], the bottom \[m - r\] rows of \[\user1{R}\]are filled with zeros. This means that the bottom \[m - r\] rows of \[\user1{E}\] must be solutions to \[{\user1{x}^T}\user1{A} = \user1{0}\]. Since \[\user1{E}\] is invertible, all the rows must be independent, otherwise there would be a linear combination of the rows such that \[{\user1{E}^T}\user1{y} = \user1{0}\] for nonzero \[\user1{y}\] which is impossible since \[\user1{E}\] is invertible and there is no transformation to transform \[\user1{0}\] back into \[\user1{y}\]. Hence the bottom \[m - r\] rows of \[\user1{E}\] must form a basis for the left nullspace of \[\user1{A}\].

# Solving the Equation \[\user1{Ax} = \user1{b}\]

One of the fundamental problems in linear algebra is how to find solutions to the equation \[\user1{Ax} = \user1{b}\]. In this section we will work through how to effectively visualize this problem and how to solve it.

## Interpreting the Product \[\user1{Ax}\]

Suppose we have a 3 x 3 matrix \[\user1{A}\] defined as

\[\user1{A} = \left[ {\begin{array}{\*{20}{c}}1&4&5\\3&2&5\\2&1&3\end{array}} \right]\]

and a 3 x 1 vector \[\user1{x}\] defined as

\[\user1{x = }\left[ {\begin{array}{\*{20}{c}}{{x\_1}}\\{{x\_2}}\\{{x\_3}}\end{array}} \right]\]

How should we interpret the matrix product \[\user1{Ax}\]? Let us consider a few powerful interpretations. Note that although our example is in \[{\mathbb{R}^3}\], these interpretations apply for any n-dimensional coordinate space.

1. \*\***Row Picture - Hyperplanes**:\*\* The row picture originates from expanding the matrix product using the rows of \[\user1{A}\] as \[\user1{Ax} = \left[ {\begin{array}{\*{20}{c}}1&4&5\\3&2&5\\2&1&3\end{array}} \right]\user1{x} = \left[ {\begin{array}{\*{20}{c}}{\left[ {\begin{array}{\*{20}{c}}1&4&5\end{array}} \right]\user1{x}}\\{\left[ {\begin{array}{\*{20}{c}}3&2&5\end{array}} \right]\user1{x}}\\{\left[ {\begin{array}{\*{20}{c}}2&1&3\end{array}} \right]\user1{x}}\end{array}} \right]\]. We see that the output elements are computed as the dot products between \[\user1{x}\] and the rows of \[\user1{A}\].The key idea for interpretability is that we can visualize each row of \[\user1{A}\] as describing a normal vector for a hyperplane. Hence, when we take the dot product of \[\user1{x}\] with a row of \[\user1{A}\], for example \[\left[ {\begin{array}{\*{20}{c}}1&4&5\end{array}} \right]\user1{x}\], we are effectively computing the closest distance from the origin to the hyperplane with normal vector \[\left[ {\begin{array}{\*{20}{c}}1&4&5\end{array}} \right]\] that contains \[\user1{x}\] (note that these distances will be scaled if the rows of \[\user1{A}\] are not unit length). Therefore, for a given \[\user1{x}\], the matrix product \[\user1{Ax}\] can be interpreted as computing the distances to hyperplanes containing \[\user1{x}\] and with normal vectors given by the rows of\[\user1{A}\].
2. \*\***Row Picture – Change of Coordinate System**:\*\* The row picture originates from expanding the matrix product using the rows of \[\user1{A}\] as \[\user1{Ax} = \left[ {\begin{array}{\*{20}{c}}1&4&5\\3&2&5\\2&1&3\end{array}} \right]\user1{x} = \left[ {\begin{array}{\*{20}{c}}{\left[ {\begin{array}{\*{20}{c}}1&4&5\end{array}} \right]\user1{x}}\\{\left[ {\begin{array}{\*{20}{c}}3&2&5\end{array}} \right]\user1{x}}\\{\left[ {\begin{array}{\*{20}{c}}2&1&3\end{array}} \right]\user1{x}}\end{array}} \right]\]. We see that the output elements are computed as the dot products between \[\user1{x}\] and the rows of\[\user1{A}\]. The key idea for interpretability is that we can visualize each row of \[\user1{A}\] as describing a basis vector for a new coordinate system. Hence, when we take the dot product of \[\user1{x}\] with a row of \[\user1{A}\], for example \[\left[ {\begin{array}{\*{20}{c}}1&4&5\end{array}} \right]\user1{x}\], we are effectively computing the projection of \[\user1{x}\] onto the new basis vector \[\left[ {\begin{array}{\*{20}{c}}1&4&5\end{array}} \right]\]. Therefore, for a given \[\user1{x}\], the matrix product \[\user1{Ax}\] can be interpreted as computing the projection of \[\user1{x}\] onto a new coordinate system described by the rows of \[\user1{A}\].
3. \*\***Column Picture**:\*\* The column picture originates from expanding the matrix product using the columns of \[\user1{A}\] as \[\user1{Ax} = \left[ {\begin{array}{\*{20}{c}}1&4&5\\3&2&5\\2&1&3\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{{x\_1}}\\{{x\_2}}\\{{x\_3}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}1\\3\\2\end{array}} \right]{x\_1} + \left[ {\begin{array}{\*{20}{c}}4\\2\\1\end{array}} \right]{x\_2} + \left[ {\begin{array}{\*{20}{c}}5\\5\\3\end{array}} \right]{x\_3}\]. We see that the output vector can be interpreted as a linear combination of the columns of \[\user1{A}\] where the combination coefficients are the elements of \[\user1{x}\]. Therefore, for a given \[\user1{x}\], the matrix \[\user1{Ax}\] is a linear combination of the columns.

## Interpreting the Equation \[\user1{Ax} = \user1{b}\]

Suppose that we are trying to solve the equation \[\user1{Ax} = \user1{b}\], where \[\user1{A}\] is an \[m\] by \[n\] matrix, and we have substantially more equations than unknowns so \[m > n\]. There will typically not exist a unique solution to this problem. What does it mean for there not to exist a unique solution? We can interpret this in two ways:

1. \*\*Row-centric:\*\* we can write the matrix product \[\user1{Ax} = \user1{b}\], as \[\user1{Ax} = \left[ {\begin{array}{\*{20}{c}}{{\user1{A}\_{1,:}}}\\ \vdots \\{{\user1{A}\_{m,:}}}\end{array}} \right]\user1{x} = \left[ {\begin{array}{\*{20}{c}}{{{\rm{b}}\_1}}\\ \vdots \\{{b\_m}}\end{array}} \right]\]. Therefore, we can interpret the matrix product as a system \[m\] equations, each of the form \[{\user1{A}\_{k,:}}\user1{x} = {b\_k}\], where \[{\user1{A}\_{k,:}}\] is the \[{k^{th}}\] row of \[\user1{A}\]. Each of these equations can be viewed as a hyperplane in \[{\mathbb{R}^n}\]. If an \[\user1{x}\] satisfies the equation \[{\user1{A}\_{k,:}}\user1{x} = {b\_k}\], then it implies that \[\user1{x}\] is located on the \[{k^{th}}\] hyperplane. To pick an \[\user1{x}\] which satisfies the equation \[\user1{Ax} = \user1{b}\], it follows that \[\user1{x}\] must be on each of the \[m\] hyperplanes, which is equivalent to \[\user1{x}\] being the intersection point of all hyperplanes. In the situation where we have more equations than unknowns such that \[m > n\], then we have more hyperplanes than the dimension of the space holding the hyperplanes and there will generally not exist an intersection point between these hyperplanes. To see why, consider the following example. Suppose that \[\user1{x}\] is a vector in \[{\mathbb{R}^2}\] such that we have 2 unknowns. If we have 1 equation, then any point on the hyperplane (in the case of \[{\mathbb{R}^2}\], the generalized “hyperplane” is a line) described by that equation will be a solution to the system. If we have 2 linearly independent equations, then there will be a unique intersection point of the hyperplanes and a unique solution to the system. Suppose we now take this 2-equation system with a unique solution and want to add another equation to the system. For this new system to remain consistent, we must ensure that the new hyperplane contains the unique solution from the 2-equation system. The two normal vectors for the hyperplanes in the 2-equation system describe a complete basis for \[{\mathbb{R}^2}\], therefore, any new hyperplane will have a normal vector that can be written as a linear combination of these basis vectors. Hence, any new equation will need to be linearly dependent on the existing equations from the 2-equation system to preserve consistency. Hence, if we have a new hyperplane that is not linearly dependent on the existing equations, we lose consistency and no longer have a solution. Another way to think about this situation is to consider that to add a new equation to the 2-equation system and preserve consistency, the new hyperplane must contain the unique solution to the 2-equation system, but is free to rotate around that point in space. A linear combination of the prior equations from the 2-equation system can be viewed as constructing a new hyperplane with a normal vector that is effectively a rotation along the axis between the two normal vectors from the 2-equation system.
2. \*\*Column-centric:\*\* we can write the matrix product \[\user1{Ax} = \user1{b}\], as \[\user1{Ax} = {\user1{A}\_{:,1}}{x\_1} + \cdots + {\user1{A}\_{:,n}}{x\_n} = \left[ {\begin{array}{\*{20}{c}}{{{\rm{b}}\_1}}\\ \vdots \\{{b\_m}}\end{array}} \right]\]. Therefore, we can interpret the matrix product as the linear combination of the columns of \[\user1{A}\]. Effectively the matrix product defines the column space of \[\user1{A}\]. A solution can be found if \[\user1{b}\] is in the column space of \[\user1{A}\]. When \[m > n\], the column space is a lower dimensional linear manifold in the space \[{\mathbb{R}^m}\]. As such, it becomes likely that there is a component of \[\user1{b}\] which is not in the column space.

Now suppose that we are given a 3 x 1 vector \[\user1{b}\] defined as

\[\user1{b = }\left[ {\begin{array}{\*{20}{c}}{{b\_1}}\\{{b\_2}}\\{{b\_3}}\end{array}} \right]\]

And we are asked to find \[\user1{x}\] such that \[\user1{Ax} = \user1{b}\]. How should we interpret the matrix equation \[\user1{Ax} = \user1{b}\]? Let us consider a few powerful interpretations.

1. \*\***Row Picture - Hyperplanes**:\*\* If we expand the matrix equation \[\user1{Ax} = \user1{b}\] using the rows of \[\user1{A}\], we can write an equation for each row, for example \[\left[ {\begin{array}{\*{20}{c}}1&4&5\end{array}} \right]\user1{x} = {b\_1}\]. To satisfy this equation, \[\user1{x}\] must lie on a hyperplane with normal vector \[\left[ {\begin{array}{\*{20}{c}}1&4&5\end{array}} \right]\] that is a distance of \[\frac{{{b\_1}}}{{\left\| {\left[ {\begin{array}{\*{20}{c}}1&4&5\end{array}} \right]} \right\|}}\] from the origin. For \[\user1{x}\] to satisfy the matrix equation \[\user1{Ax} = \user1{b}\], \[\user1{x}\] must simultaneously lie on each of the hyperplanes defined by the row equations. Therefore, we can visualize the solution space for \[\user1{x}\] as the set of points in the intersection of all these hyperplanes. We can visualize the effect of changing the components of \[\user1{b}\] as the process of moving the individual hyperplanes either closer or further from the origin along their respective normal vector axes defined by the rows of \[\user1{A}\].
2. \*\***Column Picture**:\*\* The column picture originates from expanding the matrix product using the columns of \[\user1{A}\] as\[\user1{Ax} = \left[ {\begin{array}{\*{20}{c}}1&4&5\\3&2&5\\2&1&3\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{{x\_1}}\\{{x\_2}}\\{{x\_3}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}1\\3\\2\end{array}} \right]{x\_1} + \left[ {\begin{array}{\*{20}{c}}4\\2\\1\end{array}} \right]{x\_2} + \left[ {\begin{array}{\*{20}{c}}5\\5\\3\end{array}} \right]{x\_3} = \user1{b}\]. To satisfy this equation, we must be able to express \[\user1{b}\] as a linear combination of the columns of \[\user1{A}\]. Any coefficients \[\user1{x}\] that express \[\user1{b}\] as a linear combination of the columns of \[\user1{A}\] is a valid solution. Any \[\user1{b}\] that exists in the column space of \[\user1{A}\], (i.e. the space defined by linear combinations of the columns) will have a solution \[\user1{x}\].

## Interpreting the Matrix Product \[\user1{A} = \user1{BC}\]

Now suppose that we have two matrices \[\user1{B} = \left[ {\begin{array}{\*{20}{c}}1&3\\5&2\end{array}} \right]\] and \[\user1{C} = \left[ {\begin{array}{\*{20}{c}}4&7\\1&6\end{array}} \right]\], and we want to calculate\[\user1{A = BC}\]. How can we interpret this multiplication? Here are a few different interpretation methods:

1. \*\***Individual Elements**:\*\* We can calculate the value of individual elements of \[\user1{A}\] by taking the dot product of a row of \[\user1{B}\] with a column of \[\user1{C}\]. For example, to calculate the \[{\user1{A}\_{1,2}}\] (first row, second column), we could take the dot product of the first row of \[\user1{B}\] and the second column of \[\user1{C}\], calculated as \[{\user1{A}\_{1,2}} = {\user1{B}\_{1, & :}}{\user1{C}\_{:,1}} = \left[ {\begin{array}{\*{20}{c}}1&3\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}7\\6\end{array}} \right] = 25\].
2. \*\***Combination of Columns**:\*\* We can calculate the value of a column of \[\user1{A}\] by taking a linear combination of the columns of \[\user1{B}\], where the combination coefficients come from a column of \[\user1{C}\]. For example, we can calculate the second column of \[\user1{A}\] as follows: \[{\user1{A}\_{:,2}} = \user1{B}{\user1{C}\_{:,2}} = \left[ {\begin{array}{\*{20}{c}}1\\5\end{array}} \right]7 + \left[ {\begin{array}{\*{20}{c}}3\\2\end{array}} \right]6 = \left[ {\begin{array}{\*{20}{c}}{25}\\{47}\end{array}} \right]\].
3. \*\***Combination of Rows**:\*\* We can calculate the value of a row of \[\user1{A}\] by taking a linear combination of the rows of \[\user1{C}\], where the combination coefficients come from a row of \[\user1{B}\]. For example, we can calculate the second row of \[\user1{A}\] as follows:\[{\user1{A}\_{2,:}} = {\user1{B}\_{2,:}}\user1{C} = 5\left[ {\begin{array}{\*{20}{c}}4&7\end{array}} \right] + 2\left[ {\begin{array}{\*{20}{c}}1&6\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{22}&{47}\end{array}} \right]\].
4. \*\***Sum of Rank-1 Matrices**:\*\* The matrix \[\user1{A}\] can be expressed as a sum of the outer products of the columns of \[\user1{B}\] and the rows of \[\user1{C}\]. Therefore, we can write: \[\user1{A} = \left[ {\begin{array}{\*{20}{c}}1\\5\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}4&7\end{array}} \right] + \left[ {\begin{array}{\*{20}{c}}3\\2\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}1&6\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}7&{25}\\{22}&{47}\end{array}} \right]\].
5. \*\***Decomposition into Blocks**:\*\* If we partition the matrix \[\user1{A}\] into blocks and cut the matrices \[\user1{B}\] and \[\user1{C}\] into appropriately sized blocks, then we can view the multiplication in terms of the multiplication of individual block components. For example, suppose that \[\user1{B}\] and \[\user1{C}\] were large matrices. if we partition the matrices such that \[\user1{A} = \left[ {\begin{array}{\*{20}{c}}{{\user1{A}\_1}}&{{\user1{A}\_2}}\\{{\user1{A}\_3}}&{{\user1{A}\_4}}\end{array}} \right]\], \[\user1{B} = \left[ {\begin{array}{\*{20}{c}}{{\user1{B}\_1}}&{{\user1{B}\_2}}\\{{\user1{B}\_3}}&{{\user1{B}\_4}}\end{array}} \right]\], \[\user1{C} = \left[ {\begin{array}{\*{20}{c}}{{\user1{C}\_1}}&{{\user1{C}\_2}}\\{{\user1{C}\_3}}&{{\user1{C}\_4}}\end{array}} \right]\], then we can write the equation for an individual block of \[\user1{A}\] as \[{\user1{A}\_1} = {\user1{B}\_1}{\user1{C}\_1} + {\user1{B}\_2}{\user1{C}\_3}\].

## Interpreting the Matrix Inverse

What is a matrix inverse? We can view the matrix product \[\user1{Ax} = \user1{y}\] as transforming the vector \[\user1{x}\] into \[\user1{y}\]. Hence, the inverse of this transformation, called \[{\user1{A}^{ - 1}}\], should be able to turn \[\user1{y}\] into \[\user1{x}\], such that \[{\user1{A}^{ - 1}}\user1{y = x}\]. Combining the equations, we see that \[{\user1{A}^{ - 1}}\user1{Ax} = \user1{x}\]. Since this equation must hold for any \[\user1{x}\], it follows that \[{\user1{A}^{ - 1}}\user1{A}\] must be the identity matrix such that and \[{\user1{A}^{ - 1}}\user1{A} = \user1{I}\].

Suppose that we are trying to find the inverse of a square matrix \[\user1{A}\]. From the definition of the inverse, we must solve for the matrix \[{\user1{A}^{ - 1}}\] such that \[\user1{A}{\user1{A}^{ - 1}} = \user1{I}\]. We can interpret this matrix multiplication using the interpretability methods in the section described above. Specifically, we can consider how the columns of \[\user1{A}{\user1{A}^{ - 1}}\] can be written as a linear combination of the columns of \[\user1{A}\]. As the columns of \[\user1{I}\] are all independent, they must span the entire space. As such, it follows that the columns of \[\user1{A}\] must also be independent and span the entire space. Another way to interpret this is as follows, for \[\user1{A}\] to have an inverse, it must have the columns of \[\user1{I}\] in its column space. Similarly, we can write the matrix product \[{\user1{A}^{ - 1}}\user1{A}\] as a linear combination of the rows of \[\user1{A}\]. Thus, in a likewise manner, for \[\user1{A}\] to have an inverse, its row space must contain the rows of \[\user1{I}\].

Another way to see when an inverse does not exist is to consider whether there exists a nonzero \[\user1{x}\] such that \[\user1{Ax} = \user1{0}\]. If such an \[\user1{x}\] exists, then the matrix \[\user1{A}\] is not invertible. To see why, suppose that an inverse \[{\user1{A}^{ - 1}}\] did exist. Then it would be possible to write \[{\user1{A}^{ - 1}}\user1{Ax = x}\], but we know that \[\user1{Ax} = \user1{0}\] which implies that \[{\user1{A}^{ - 1}}\user1{0} = \user1{x}\] which is impossible for nonzero \[\user1{x}\], therefore \[{\user1{A}^{ - 1}}\] cannot exist. To visualize this intuitively, when \[\user1{Ax} = \user1{0}\], we can think of \[\user1{A}\] as mapping points from the space onto the zero vector. The zero vector is like a black hole, once a vector enters it, no transformation will be able to get the original vector back.

Given an \[n\] by \[n\] square matrix \[\user1{A}\], how can we solve for the inverse? From the definition, our inverse is defined by the equation\[\user1{A}{\user1{A}^{ - 1}} = \user1{I}\]. Hence, we have \[{n^2}\] equations with \[{n^2}\] unknowns (i.e. the unknowns are all elements in \[{\user1{A}^{ - 1}}\]). We can solve these equations using Gauss-Jordan elimination. We can view each elimination step as applying an elementary operation applied to the left side of \[\user1{A}\] such that we eventually convert \[\user1{A}\] into \[\user1{I}\], with steps given as \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{A = I}\]. Therefore, we see that \[{\user1{A}^{ - 1}} = {\user1{E}\_M} \cdots {\user1{E}\_1}\]. Notice that \[{\user1{E}\_M} \cdots {\user1{E}\_1} = {\user1{E}\_M} \cdots {\user1{E}\_1}\user1{I}\], so we can set up an augmented matrix where on the left side we determine the row operations to transform \[\user1{A}\] into \[\user1{I}\], and on the right side we apply the same operations to \[\user1{I}\] to determine the inverse matrix \[{\user1{A}^{ - 1}} = {\user1{E}\_M} \cdots {\user1{E}\_1} = {\user1{E}\_M} \cdots {\user1{E}\_1}\user1{I}\].

Suppose that \[\user1{A}\] is a square matrix that has left inverse \[\user1{L}\] and right inverse \[\user1{R}\]. To show that \[\user1{L = R}\], consider the following proof. Using a clever rearrangement of parentheses, we can write \[\user1{L} = \user1{LI} = \user1{L}\left( {\user1{AR}} \right) = \left( {\user1{LA}} \right)\user1{R} = \user1{IR} = \user1{R}\]. Notice that this inequality only holds when \[\user1{A}\] is square, otherwise matrix multiplication is not possible due to the differing matrix sizes.

## When can we solve \[\user1{Ax} = \user1{b}\]?

An important question in linear algebra is to determine whether \[\user1{Ax} = \user1{b}\] has a solution, and how we can describe these solutions. A critical tool to understand the solution space is to write the matrix \[\user1{A}\] in reduced row echelon form. We can write the matrix \[\user1{A}\] in reduced row echelon form by left multiplying by a series of invertible elementary row operations \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{Ax = Rx = }{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{b}\], where \[\user1{R} = {\user1{E}\_M} \cdots {\user1{E}\_1}\user1{A}\] is the reduced row echelon form of \[\user1{A}\].

To assess whether the equation \[\user1{Ax} = \user1{b}\] can be solved for a given \[\user1{b}\], we can examine the rows of \[\user1{R}\]. If there exists a row of \[\user1{R}\] that is all zeros, and the corresponding element in \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{b}\] is nonzero, then the equation \[\user1{Ax} = \user1{b}\] has no solutions. We interpret this using our intuition for the row picture: every row equation in \[\user1{Ax} = \user1{b}\] describes a hyperplane. When we apply elementary row operations to this equation, we transform the hyperplanes, but preserve the set of points that lie on the intersection between hyperplanes used in the transformation. For example, when adding two rows together, the resulting hyperplane will preserve the set of points that lie on the intersection of the two hyperplanes being added. Hence, to produce a row of zeros in \[\user1{R}\], two rows must be combined that correspond to the same hyperplane normal vector. For there to be solutions, the distance to these hyperplanes from the origin (found in \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{b}\]) must be the same for both planes, otherwise they will not intersect, and no solutions exist.

If the equation \[\user1{Ax} = \user1{b}\] is solvable, then we can write the general set of solutions as\[{\user1{x}\_{complete}} = \left\{ {{\user1{x}\_p} + {\user1{x}\_n},{\user1{x}\_n} \in N\left( \user1{A} \right)} \right\}\], where \[{\user1{x}\_p}\]is a particular solution such that \[\user1{A}{\user1{x}\_p} = \user1{b}\], and \[{\user1{x}\_n}\] is a generic vector in the nullspace of \[\user1{A}\] such that \[\user1{A}{\user1{x}\_n} = \user1{0}\]. To show that \[{\user1{x}\_{complete}}\] contains all solutions to \[\user1{Ax} = \user1{b}\], let us consider any arbitrary solution \[{\user1{y}^\*}\]. If we calculate the matrix product of \[\user1{A}\] with the difference between \[{\user1{y}^\*}\] and \[{\user1{x}\_p}\], we get \[\user1{A}\left( {{\user1{y}^\*} - {\user1{x}\_p}} \right) = \user1{A}{\user1{y}^\*} - \user1{A}{\user1{x}\_p} = \user1{b} - \user1{b} = \user1{0}\]. This implies that \[{\user1{y}^\*} - {\user1{x}\_p}\] is in the nullspace of \[\user1{A}\] such that \[{\user1{y}^\*} - {\user1{x}\_p} \in N\left( \user1{A} \right)\]. Since \[{\user1{y}^\*} - {\user1{x}\_p} \in N\left( \user1{A} \right)\], we can let \[{\user1{x}\_n} = {\user1{y}^\*} - {\user1{x}\_p}\] and write \[{\user1{x}\_p} + {\user1{x}\_n} = {\user1{x}\_p} + {\user1{y}^\*} - {\user1{x}\_p} = {\user1{y}^\*}\]. Therefore, \[{\user1{y}^\*} = {\user1{x}\_p} + {\user1{x}\_n} \in {\user1{x}\_{complete}}\] and \[{\user1{x}\_{complete}}\] contains all solutions to \[\user1{Ax} = \user1{b}\]. To gain intuition behind this, we can visualize the solution space as being the intersection of hyperplanes which is a linear manifold. The points on this linear manifold can be described by the combination of a point on the linear manifold (\[{\user1{x}\_p}\]), and the direction vectors which are parallel to the linear manifold (the nullspace vectors \[{\user1{x}\_n}\]).

Hence to find all solutions to \[\user1{Ax} = \user1{b}\], we need to find: 1) a particular solution \[{\user1{x}\_p}\], and 2) a basis for the nullspace \[N\left( \user1{A} \right)\]. Since any particular solution will do, one way to find a particular solution is to take the reduced row echelon matrix \[\user1{R}\], set the free variables in \[\user1{x}\] to 0, and solve for the pivot variables to satisfy \[\user1{Ax} = \user1{b}\].

To find the nullspace, we need to find the solutions to the equation \[\user1{Rx} = \user1{0}\]. One way to do this is to capitalize upon the structure of the reduced row echelon matrix. We start by exchanging some columns in \[\user1{R}\] so that the pivot columns are clustered together on the left side of the matrix. We can write \[\user1{R}\] as a block matrix such that \[\user1{R = }\left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_r}}&\user1{F}\\\user1{0}&\user1{0}\end{array}} \right]\] where \[\user1{F}\]is the set of coefficients associated with the free variables and \[{\user1{I}\_r}\] is an identity matrix of size \[r\] by \[r\] where \[r\] is the rank of \[\user1{A}\]. To find vectors for the nullspace, we need to solve the equation \[\user1{Rx = 0}\]. Using the block representation for \[\user1{R}\], we can solve for a basis of the nullspace by recognizing that \[\user1{RN} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_r}}&\user1{F}\\\user1{0}&\user1{0}\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{ - \user1{F}}\\{{\user1{I}\_{n - r}}}\end{array}} \right] = \user1{0}\] where \[{\user1{I}\_{n - r}}\] is an identity matrix of size \[n - r\] by \[n - r\]. Therefore, the columns of the nullspace matrix \[\user1{N} = \left[ {\begin{array}{\*{20}{c}}{ - \user1{F}}\\{{\user1{I}\_{n - r}}}\end{array}} \right]\] provide a basis for the nullspace of \[\user1{A}\]. To summarize the operations done: we are trying to solve for the solutions to \[\user1{Ax = 0}\]. We can apply elementary row operations by left multiplying both sides of the equation with the elementary matrices such that \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{Ax} = {\user1{E}\_M} \cdots {\user1{E}\_1}\user1{0} = \user1{0}\]. Therefore, we see that applying row operations (that are invertible) does not change the nullspace since \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{Ax} = \user1{0}\]. Finally, we can right multiply the matrix with permutation matrix \[\user1{Q}\] to permute the columns, written as\[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{AQ}{\user1{Q}^{ - 1}}\user1{x} = \user1{R}{\user1{Q}^{ - 1}}\user1{x} = \user1{Ry} = \user1{0}\], where the final reduced row echelon matrix is \[\user1{R} = {\user1{E}\_M} \cdots {\user1{E}\_1}\user1{AQ}\], and the permuted vector \[\user1{x}\] is written as \[\user1{y} = {\user1{Q}^{ - 1}}\user1{x}\]. The permutation ensures that the pivot variable columns are on the left side of \[\user1{R}\] and the free variable columns are on the right side of \[\user1{R}\].

We can predict the type of solutions possible for the equation \[\user1{Ax} = \user1{b}\] by looking at the rank of \[\user1{A}\] which is an \[m\] by \[n\] matrix with rank \[r\]. Here we define \[\user1{R}\] to be the reduced row echelon form of \[\user1{A}\] with the pivot columns located to the left of the matrix. We have the following four potentials for the rank of \[\user1{A}\].

1. \[r = m = n\]: In this case, \[\user1{R} = \user1{I}\] since each row and column have a pivot. \[\user1{A}\] is invertible, and we have a unique solution to \[\user1{Ax} = \user1{b}\]. To see why \[\user1{A}\] is invertible. Recall that to construct \[\user1{R}\] we apply a series of invertible elementary row operations to \[\user1{A}\] such that \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{A} = \user1{R} = \user1{I}\]. Therefore, we see that \[{\user1{E}\_M} \cdots {\user1{E}\_1} = {\user1{A}^{ - 1}}\]
2. \[r = n < m\]: In this case, \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}\user1{I}\\\user1{0}\end{array}} \right]\]. Since we have no free variables, the nullspace only has the zero vector. Therefore, depending on the value of \[\user1{b}\] we will either have 0 or 1 solution to the equation \[\user1{Ax} = \user1{b}\]
3. \[r = m < n\]: In this case, \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}\user1{I}&\user1{F}\end{array}} \right]\]. Each row has a pivot, so we will always have a solution. Furthermore, we have free variables, therefore, the nullspace is a linear manifold and we will have infinitely many solutions
4. \[r < m,\,r < n\]: In this case, \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}\user1{I}&\user1{F}\\\user1{0}&\user1{0}\end{array}} \right]\] and we have free variables. Therefore, the nullspace is a linear manifold. Depending on the value of \[\user1{b}\] we will either have 0 or infinitely many solutions to the equation \[\user1{Ax} = \user1{b}\]

For \[\user1{Ax} = \user1{b}\] to be solvable, we have the following interpretations:

1. The vector \[\user1{b}\] must be in the column space of \[\user1{A}\].
2. We can write the reduced row echelon form of \[\user1{A}\] , denoted by \[\user1{R}\], by left multiplying both sides of the equation with elementary row operations such that\[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{Ax} = \user1{Rx} = {\user1{E}\_M} \cdots {\user1{E}\_1}\user1{b}\]. Hence, we have the transformed matrix equation \[\user1{Rx} = {\user1{E}\_M} \cdots {\user1{E}\_1}\user1{b}\]. If \[\user1{R}\] contains a row of zeros, then the matching entry in the transformed \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{b}\] vector must also be 0. If not, then no solution exists. We can interpret this using our intuition from the row picture of matrix multiplication. Each row equation of \[\user1{Ax} = \user1{b}\] represents a hyperplane. When we perform an elementary row operation \[{\user1{E}\_k}\] that adds a multiple of one row to another, we are transforming one of the hyperplanes into a new hyperplane while preserving the points that lie in the intersection of the hyperplanes represented by the two original rows. In other words, adding two rows together effectively constructs a new hyperplane that is rotated around the intersection set (in \[{\mathbb{R}^2}\] the intersection set would be a point, and in \[{\mathbb{R}^3}\] the intersection set would be a line). Hence, applying an elementary row operation modifies the hyperplane geometry but does not change the solution space (i.e. the set of intersections between all hyperplanes). Therefore, if we apply an elementary row operation that “adds a multiple of one row to another”, and this operation produces a row of zeros, then for a solution to exist, it must be that these two rows represent the same hyperplane, and therefore, the distance to the origin must be the same or else there is no intersections, and thereby, no solutions. Hence, when we subtract the distances to the origin for both rows (i.e. the corresponding values of the \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{b}\] vector), the value must be 0. We can view each invertible matrix product as an operation that transforms the hyperplanes in our equation while preserving the set of intersection points between the hyperplanes that are combined.

## How can we solve \[\user1{Ax} = \user1{b}\]?

Suppose we have an \[m\] by \[n\] matrix \[\user1{A}\] where \[m < n\]. How do we quickly show that for any solvable equation \[\user1{Ax} = \user1{b}\], we must have an infinite number of solutions? One way is to recognize this result is to use the structure of the reduced row echelon form. If we put the matrix \[\user1{A}\] in reduced row echelon form with the pivot columns on the left side, then it follows that the number of pivots must be less than or equal to the smaller dimension \[m\]. To understand why, remember that to compute the reduced row echelon form, we compute the following algorithm

1. Start with the element in the first row and first column. If this element is zero, then try to swap rows to make it nonzero. If the entire first row is zeros, then swap columns to make it nonzero.
2. Scale the first row to make the element in the first row, first column equal to 1
3. Add a multiple of the first row to all other rows to make the element in the first column equal 0. At this point, the only nonzero element should be the element in the first row, first column
4. Now proceed to complete steps 1-3 for the second row, second column, etc. Continue until there are no more nonzero entries in the remaining rows
5. The total computation of the reduced row echelon form for \[\user1{Ax} = \user1{b}\] can be written as \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{A}\left( {{\user1{Q}\_1} \cdots {\user1{Q}\_N}} \right)\left( {\user1{Q}\_N^{ - 1} \cdots \user1{Q}\_1^{ - 1}} \right)\user1{x} = {\user1{E}\_M} \cdots {\user1{E}\_1}\user1{b}\] where \[{\user1{E}\_M} \cdots {\user1{E}\_1}\] are the elementary row operations (i.e. scaling rows, swapping rows, adding a multiple of one row to another), and \[{\user1{Q}\_1} \cdots {\user1{Q}\_N}\] represent column permutation matrices used to swap the order of the columns. Recall that the inverse of a permutation matrix is the matrix itself. To understand this, consider that the inverse of swapping two columns is to simply apply the same operation again to swap the columns back. Hence by left multiplying \[\user1{x}\] with \[\user1{Q}\_N^{ - 1} \cdots \user1{Q}\_1^{ - 1}\], we are effectively performing row permutations on \[\user1{x}\] so that whatever columns we swapped in \[\user1{A}\], we swap the same rows in \[\user1{x}\] so that the result of the matrix product is unaffected as can be seen since \[\user1{A}\left( {{\user1{Q}\_1} \cdots {\user1{Q}\_N}} \right)\left( {\user1{Q}\_N^{ - 1} \cdots \user1{Q}\_1^{ - 1}} \right)\user1{x} = \user1{AIx = Ax}\]. Looking back at steps 1-3 described above, we see that during the computation of the reduced row echelon form for \[\user1{Ax} = \user1{b}\], we will apply a sequence of row and column operations. As a hypothetical example, we may swap two rows (represented by \[{\user1{E}\_1}\]), then add a multiple of one row to another (represented by \[{\user1{E}\_2}\]), then swap two columns (represented by \[{\user1{Q}\_1}\]), then add a multiple of one row to another (represented by \[{\user1{E}\_3}\]), etc. The key point is that it is intuitive to think of the row operations and column swaps as being intermingled (i.e. the order from the described example was \[{\user1{E}\_1},{\user1{E}\_2},{\user1{Q}\_1},{\user1{E}\_3}\]). Due to the matrix commutativity property, we can achieve this interpretation because we are “free” to compute the multiplication in the order that makes intuitive sense. To make this concrete when computing \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{A}{\user1{Q}\_1} \cdots {\user1{Q}\_N}\], the commutative properties tell us that we are free to put parentheses wherever we want. Hence, to represent the sequence described in the example above we have \[{\user1{E}\_M} \cdots \left( {{\user1{E}\_3}\left( {\left( {{\user1{E}\_2}\left( {{\user1{E}\_1}\user1{A}} \right)} \right){\user1{Q}\_1}} \right)} \right) \cdots {\user1{Q}\_N}\]. In this case we see that the order of multiplications is \[{\user1{E}\_1},{\user1{E}\_2},{\user1{Q}\_1},{\user1{E}\_3}\] which matches the example.
6. The final form of the reduced row echelon matrix produced by the above steps is a matrix with the general form \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}&{{\user1{F}\_{r,n - r}}}\\{{\user1{0}\_{m - r,r}}}&{{\user1{0}\_{m - r,n - r}}}\end{array}} \right]\], where \[r\] is the rank and is defined as the number of pivots. The subscripts of the block matrices in \[\user1{R}\] denote the row by column size of each of the blocks. It is important to note that depending on the shape and rank of the matrix, some of these blocks may be empty. For example, if \[r = n < m\] then the matrix reduces down to \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{{\user1{I}\_{r,r}}}\\{{\user1{0}\_{m - r,r}}}\end{array}} \right]\]. The block denoted by \[{\user1{F}\_{r,n - r}}\] refers to the coefficients for the non-pivot variables, called the “free variables”. Note that the partitioning of the variables into “free” vs. “pivot” is somewhat arbitrary because we could simply apply a column swap to turn a “free” variable into a “pivot” variable. Therefore, what is most important is the final block partitioned form of the matrix and the fact that we can write any matrix in the reduced row echelon form using the steps described above.

Returning to the central question: how can we solve \[\user1{Ax} = \user1{b}\]? One method is as follows:

1. Convert the system into reduced row echelon form
2. We can find a particular solution by setting the free variables to 0 and solve for the pivot variables
3. Next, we solve for the nullspace basis vectors by setting the free variables to the standard basis and solving for the pivot variables
4. Combine the particular solution and the linear combinations of the nullspace basis vectors to form the complete solution

Let us discuss the intuition for the method described above. We can visualize the solution space to \[\user1{Ax} = \user1{b}\] as being a linear manifold where the particular solution is a point on the linear manifold, and the nullspace basis describes a set of direction vectors that are parallel to the linear manifold. Hence, the particular solution plus a linear combination of the nullspace basis vectors defines all vectors on the linear manifold and ensures that \[\user1{Ax} = \user1{b}\].

To show that any solution to \[\user1{Ax} = \user1{b}\] can be written in terms of a particular solution \[{\user1{x}^\*}\] and vectors in the nullspace, let us postulate the potential of another solution \[{\user1{y}^\*}\] that cannot be written in terms of \[{\user1{x}^\*}\] and the vectors in the nullspace. If we take the difference between \[{\user1{y}^\*}\] and \[{\user1{x}^\*}\] we can write \[\user1{A}\left( {{\user1{y}^\*} - {\user1{x}^\*}} \right) = \user1{A}{\user1{y}^\*} - \user1{A}{\user1{x}^\*} = \user1{b} - \user1{b} = \user1{0}\]. Therefore, we see that \[{\user1{y}^\*} - {\user1{x}^\*}\] is in the nullspace of \[\user1{A}\], and we can write the solution \[{\user1{y}^\*}\] in terms of \[{\user1{x}^\*}\] and the vectors in the nullspace as \[{\user1{y}^\*} = \left( {{\user1{y}^\*} - {\user1{x}^\*}} \right) + {\user1{x}^\*}\]. Hence any solution of \[\user1{Ax} = \user1{b}\] can be written in terms of a particular solution \[{\user1{x}^\*}\] and vectors in the nullspace.

## Least Squares

In many practical situations we have noisy measurements with many measurements and few parameters. In this case, the chance of having the solution in the column space is very unlikely. Hence the equation \[\user1{Ax} = \user1{b}\] will not have a solution. If we cannot find a solution to the equation \[\user1{Ax} = \user1{b}\], then an alternative approach is to find the solution that minimizes the error \[{\left\| {\user1{b} - \user1{Ax}} \right\|^2} = {\left\| \user1{e} \right\|^2}\]. This can be interpreted as finding the point in the column space of \[\user1{A}\] that is closest to \[\user1{b}\]. Recall that we can visualize the column space as a linear manifold. The displacement from a linear manifold to a point in space can be written as a combination of a component parallel to the manifold, and a component perpendicular to the manifold. Hence, the distance from a linear manifold to a point in space can be minimized by finding the point on the linear manifold such that the displacement only consists of the perpendicular component. For linear manifolds, this will be a unique point in space. We can find this point by solving for the \[\user1{x}\] which produces a displacement vector \[\user1{b} - \user1{Ax}\] that is orthogonal to the column space. For a vector to be orthogonal to the column space, it must be orthogonal to all columns in \[\user1{A}\]. Therefore, it follows that \[{\user1{A}^T}\left( {\user1{b} - \user1{Ax}} \right) = \user1{0}\]. Using the QR factorization of \[\user1{A}\] we can write the equation as

\[\begin{array}{l}{\user1{A}^T}\left( {\user1{b} - \user1{Ax}} \right) = \user1{0}\\ \Rightarrow {\left( {\user1{QR}} \right)^T}\left( {\user1{b} - \user1{QRx}} \right) = \user1{0}\\ \Rightarrow {\user1{R}^T}\user1{Rx} = {\user1{R}^T}{\user1{Q}^T}\user1{b}\end{array}\]

To simplify this further, we would like to multiply both sides by \[{\left( {{\user1{R}^T}} \right)^{ - 1}}\]. However, we must first show that this inverse exists. The nullspace is defined by solutions to \[\user1{Ax = 0}\]. Substituting in the QR factorization we get \[\user1{Ax = 0} \Rightarrow \user1{QRx = 0}\]. Since the columns of \[\user1{Q}\] are orthonormal, it follows that the nullspace of \[\user1{Q}\] is only the zero vector. Hence, the nullspace of \[\user1{QR}\] must be equal to the nullspace of \[\user1{R}\] which in turn is equal to the nullspace of \[\user1{A}\]. If \[\user1{A}\] has full column rank (i.e. all its columns are independent) which is typically the case for tall skinny data matrices, then the nullspace of \[\user1{A}\] will only be the zero vector, which implies that the nullspace of \[\user1{R}\] will only be the zero vector. Hence, when \[\user1{A}\] has full column rank then \[{\left( {{\user1{R}^T}} \right)^{ - 1}}\] exists and we can write the equation from above as \[\user1{Rx} = {\user1{Q}^T}\user1{b}\]. Solving this equation will give us the least squares solution to the equation \[\user1{Ax = b}\].

# Determinants

Every square matrix has a specific number called the determinant which encodes information about the matrix. One useful interpretation of the determinant is as the volume of the parallelepiped described by the vectors of the matrix (either the rows or columns). The determinant can be used to find a closed form expression for the inverse of a matrix. Cramer’s Rule uses this expression for the inverse to produce a closed form expression for a solution to a system of linear equations.

Instead of simply writing the formula for the determinant, it is useful to describe the determinant in terms of its properties, and then use these properties to define the formula. Four axiomatic properties which form the basis for the determinant are the following:

1. The determinant of the identity matrix is \[\det \user1{I = }1\]
2. If you exchange two rows of a matrix, you reverse the sign of its determinant (i.e. positive to negative or vice versa)
3. The determinant behaves like a linear function on the rows of the matrix \[\left| {\begin{array}{\*{20}{c}}{a + a'}&{b + b'}\\c&d\end{array}} \right| = \left| {\begin{array}{\*{20}{c}}a&b\\c&d\end{array}} \right| + \left| {\begin{array}{\*{20}{c}}{a'}&{b'}\\c&d\end{array}} \right|\]
4. If we multiply one row of a matrix by \[t\], then the determinant is multiplied by \[t\] such that: \[\left| {\begin{array}{\*{20}{c}}{ta}&{tb}\\c&d\end{array}} \right| = t\left| {\begin{array}{\*{20}{c}}a&b\\c&d\end{array}} \right|\]

From these four properties we can deduce all other properties of the determinant. Some of these properties include the following

* \*\*Two Rows Equal:\*\* If two rows of a matrix are equal then the determinant is 0. To see this, suppose that we swap the two equal rows, then we can apply property 2 to see that the determinant must change sign. However, the elements of the swapped matrix have not changed since the two swapped rows are equal. Therefore, the determinant must be 0.
* \*\*Adding a Multiple of a Row:\*\* If we add a multiple of one row to another row, the determinant does not change. To see this, suppose we add a multiple of one row to another row. We can use property 3 to separate out the components of the first row, and then apply property 4 and the \*\*Two Rows Equal:\*\* properties to show the following\[\left| {\begin{array}{\*{20}{c}}{a + tc}&{b + td}\\c&d\end{array}} \right| = \left| {\begin{array}{\*{20}{c}}a&b\\c&d\end{array}} \right| + \left| {\begin{array}{\*{20}{c}}{tc}&{td}\\c&d\end{array}} \right| = \left| {\begin{array}{\*{20}{c}}a&b\\c&d\end{array}} \right| + t\left| {\begin{array}{\*{20}{c}}c&d\\c&d\end{array}} \right| = \left| {\begin{array}{\*{20}{c}}a&b\\c&d\end{array}} \right|\]
* \*\*Zero Row:\*\* If one of the rows is zero, then the determinant of the matrix is zero. To see this, apply property 4 where \[t = 0\].
* \*\*Triangular Matrix:\*\* The determinant of a triangular matrix is the product of its diagonal entries \[{d\_1}{d\_2} \cdots {d\_n}\]. To see this, we can use the \*\*Adding a Multiple of a Row\*\* property to convert the triangular matrix into a diagonal matrix without changing the determinant. Then property 4 tells us that the determinant of the diagonal matrix is equal to the product of the diagonal entries \[{d\_1}{d\_2} \cdots {d\_n}\] times the determinant of the identity matrix. From property 1 we know that the determinant of the identity matrix is 1. Note that if one of the \[{d\_k}\] is zero, then we can apply the \*\*Adding a Multiple of a Row:\*\* property to produce a row of zeros, and finally the \*\*Zero Row\*\* property to show that the determinant for this matrix is 0. Hence the determinant for any triangular matrix is \[{d\_1}{d\_2} \cdots {d\_n}\].
* \*\*Singular Matrix:\*\* If a matrix is singular, then its determinant will be zero. To see why, note that we can perform row operations on a singular matrix to produce a row of zeros. Therefore, from the \*\*Zero Row:\*\* property, the determinant will be 0.
* \*\*Product Rule:\*\* For any square matrices we have \[\left( {\det \user1{A}} \right)\left( {\det \user1{B}} \right) = \det \user1{AB}\]. This is harder to prove. See [here](https://proofwiki.org/wiki/Determinant\_of\_Matrix\_Product). A result from this property is that \[1 = \det \user1{I = }\det \user1{A}{\user1{A}^{ - 1}} = \left( {\det \user1{A}} \right)\left( {\det {\user1{A}^{ - 1}}} \right)\], therefore \[\det \user1{A} = \frac{1}{{\det {\user1{A}^{ - 1}}}}\].
* \*\*Transpose Rule:\*\* For any square matrix, \[\det {\user1{A}^T} = \det \user1{A}\]. To see why, use elimination to produce the LU factorization such that \[\user1{A} = \user1{LU}\]. Now taking the determinant of both sides gives \[\det \user1{A} = \det \user1{LU} = \left( {\det \user1{L}} \right)\left( {\det \user1{U}} \right)\]. Because \[\user1{L}\] has 1’s on the diagonal, then \[\det \user1{L} = \det {\user1{L}^T} = 1\]. Likewise, since \[\user1{U}\] is an upper triangular matrix, then the determinant is the product of the diagonals and since the diagonals stay the same for the transpose, we have \[\det \user1{U} = \det {\user1{U}^T}\]. Hence, we see that \[\det \user1{A} = \left( {\det \user1{L}} \right)\left( {\det \user1{U}} \right) = \left( {\det {\user1{U}^T}} \right)\left( {\det {\user1{L}^T}} \right) = \det {\user1{A}^T}\]

To understand how we can compute the determinant of a matrix, let us take the example of a 2 by 2 matrix

\[\user1{A} = \left[ {\begin{array}{\*{20}{c}}a&b\\c&d\end{array}} \right]\]

By applying property 3 we can write

\[\left| {\begin{array}{\*{20}{c}}a&b\\c&d\end{array}} \right| = \left| {\begin{array}{\*{20}{c}}0&b\\c&d\end{array}} \right| + \left| {\begin{array}{\*{20}{c}}a&0\\c&d\end{array}} \right| = \left| {\begin{array}{\*{20}{c}}a&0\\0&d\end{array}} \right| + \left| {\begin{array}{\*{20}{c}}a&0\\c&0\end{array}} \right| + \left| {\begin{array}{\*{20}{c}}0&b\\0&d\end{array}} \right| + \left| {\begin{array}{\*{20}{c}}0&b\\c&0\end{array}} \right|\]

Notice that for the determinants with a column filled with zeros, we can add a multiple of one row to another which would create a row of zeros. From the \*\*Zero Row\*\* property, we know that the value of these determinants is 0. Hence, we can write

\[\left| {\begin{array}{\*{20}{c}}a&b\\c&d\end{array}} \right| = \left| {\begin{array}{\*{20}{c}}a&0\\0&d\end{array}} \right| + \left| {\begin{array}{\*{20}{c}}0&b\\c&0\end{array}} \right|\]

From property 4 and property 1 we see that \[\left| {\begin{array}{\*{20}{c}}a&0\\0&d\end{array}} \right| = ad\left| {\begin{array}{\*{20}{c}}1&0\\0&1\end{array}} \right| = ad\] and from properties 1, 2, and 4 we see that \[\left| {\begin{array}{\*{20}{c}}0&b\\c&0\end{array}} \right| = - \left| {\begin{array}{\*{20}{c}}c&0\\0&b\end{array}} \right| = - cb\left| {\begin{array}{\*{20}{c}}1&0\\0&1\end{array}} \right| = - cb\]. Finally, we can combine these together to show that \[\left| {\begin{array}{\*{20}{c}}a&b\\c&d\end{array}} \right| = ad - bc\].

To calculate the determinants of higher dimensional matrices, we can extend the principal used in the 2 by 2 case, where we used property 3 to expand each row so that there is only one nonzero entry in each row. For the 2 by 2 case shown above, this was written as

\[\left| {\begin{array}{\*{20}{c}}a&b\\c&d\end{array}} \right| = \left| {\begin{array}{\*{20}{c}}0&b\\c&d\end{array}} \right| + \left| {\begin{array}{\*{20}{c}}a&0\\c&d\end{array}} \right| = \left| {\begin{array}{\*{20}{c}}a&0\\0&d\end{array}} \right| + \left| {\begin{array}{\*{20}{c}}a&0\\c&0\end{array}} \right| + \left| {\begin{array}{\*{20}{c}}0&b\\0&d\end{array}} \right| + \left| {\begin{array}{\*{20}{c}}0&b\\c&0\end{array}} \right|\]

Notice that if we expand the determinant of an \[n\] by \[n\] matrix in this way, then the number of “sub” determinants will be \[{n^n}\]. Thankfully, we can simplify this calculation since any determinant where two rows have the same column index that is nonzero will be zero (to see why, if two such rows exist, then we can add a multiple of one row to the other to create a row of zeros which means the determinant is zero). In this case the number of “sub” determinants reduces from \[{n^n}\] to \[n!\]. To see why it is \[n!\] we can ask how each “sub” determinant is constructed. We can construct a “sub” determinant by choosing which elements will be nonzero. We can think of this process as picking any one of the \[n\] elements in the first row, and then choosing one of the \[n - 1\] remaining columns in the second row, etc. Therefore, we have \[n!\] possible “sub” determinants of this form.

From this factorization, we see that if an element in row \[i\] and column \[j\] is nonzero, then every other element in row \[i\] and column \[j\] must be zero for the determinant to be nonzero. Hence, we can pose the determinant calculation as a recursive algorithm where we compute the determinant using the summation of cofactors which are the determinants of submatrices. To see more detail on the explicit calculation of the determinant, see [here](https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall-2011/least-squares-determinants-and-eigenvalues/determinant-formulas-and-cofactors/).

# Eigenvalues and Eigenvectors

We can think about a square matrix \[\user1{A}\] as a function that transforms a vector \[\user1{x}\] into a vector \[\user1{Ax}\]. The vectors which preserve their direction after the transformation (i.e. \[\user1{x}\] is parallel to \[\user1{Ax}\]) are called the eigenvectors of \[\user1{A}\]. Therefore, the eigenvectors satisfy \[\user1{Ax = }\lambda \user1{x}\]. If \[\user1{A}\] is singular, then \[\lambda = 0\] is an eigenvalue.

What is the importance of eigenvectors/eigenvalues? Here are some key insights:

* \*\*Linear Transformations:\*\* Recall that we can view a matrix as representing a linear transformation \[f:V \to W\]from a vector space \[V\] to a vector space \[W\] where we implicitly have a basis \[\left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\] for \[V\] and a basis \[\left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_m}} \right\}\] for \[W\]. Notice that when the transformation is noninvertible, then \[m\] will not equal \[n\]. We can interpret each column of a general matrix \[\user1{A}\] as expressing the coordinates for a transformed input basis vector in terms of the output basis vectors. Therefore, for the transformation \[\user1{w = Av}\], we can interpret \[\user1{v}\] as representing an input coordinate vector relative to the basis \[\left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\], and \[\user1{w}\] as representing an output transformed coordinate vector relative to the basis \[\left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_m}} \right\}\]. The choice of basis for \[V\] and \[W\] is arbitrary but will change the matrix used to represent the linear transformation. As such, notice that for eigenvectors written as \[\user1{Ax = }\lambda \user1{x}\], the input coordinates \[\user1{x}\] are relative to basis \[\left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\] and are equal to the scaled output coordinates \[\lambda \user1{x}\] which is relative to basis\[\left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_m}} \right\}\]. If \[\user1{A}\] is diagonalizable, then we have eigenvalues \[{\lambda \_1}, \ldots ,{\lambda \_n}\]and eigenvectors \[{\user1{e}\_1}, \ldots ,{\user1{e}\_n}\], which are a basis for the coordinate space \[{\mathbb{R}^n}\]. We can write any input coordinate vector in terms of these eigenvectors as: \[\user1{x} = {c\_1}{\user1{e}\_1} + \cdots + {c\_n}{\user1{e}\_n}\]. Computing the action of the transformation we see that \[\user1{Ax} = \user1{A}\left( {{c\_1}{\user1{e}\_1} + \cdots + {c\_n}{\user1{e}\_n}} \right) = {c\_1}{\lambda \_1}{\user1{e}\_1} + \cdots + {c\_n}{\lambda \_n}{\user1{e}\_n}\]. We can write this expression as \[\user1{APx} = \user1{PDx}\] where \[\user1{P} = \left[ {\begin{array}{\*{20}{c}}{{\user1{e}\_1}}& \cdots &{{\user1{e}\_n}}\end{array}} \right]\] and \[\user1{D} = \left[ {\begin{array}{\*{20}{c}}{{\lambda \_1}}& \cdots &0\\ \vdots & \ddots & \vdots \\0& \cdots &{{\lambda \_n}}\end{array}} \right]\]. Therefore, we see that \[\user1{A = PD}{\user1{P}^{ - 1}}\]. We can interpret this in the following way: the matrix \[{\user1{P}^{ - 1}}\] performs a change of basis on the input coordinate vector from \[\left\{ {{\user1{v}\_1}, \ldots ,{\user1{v}\_n}} \right\}\] to \[\left\{ {{\user1{e}\_1}, \ldots ,{\user1{e}\_n}} \right\}\], then using \[\user1{PD}\] to perform a scaling transformation that transforms coordinates from \[\left\{ {{\user1{e}\_1}, \ldots ,{\user1{e}\_n}} \right\}\] to \[\left\{ {{\user1{w}\_1}, \ldots ,{\user1{w}\_m}} \right\}\]. The key importance is the following: given a linear transformation represented by a square matrix \[\user1{A}\], we can find a change of basis that enables us to view the linear transformation as a scaling operation. The specific change of basis that enables this are the eigenvectors of \[\user1{A}\].
* \*\*Dynamical System:\*\* Many dynamical systems involve rules that govern the evolution of the current state into the next state. When the state update equations are linear, we can describe the state transition using a linear transformation represented as a matrix \[\user1{S}\]. Suppose we have some initial conditions for the system at \[t = 0\], given as \[{\user1{u}\_0}\]. To calculate the state of the system at \[t = n\], we must recursively apply the state transition transformation \[n\] times. Hence, the state of the system at \[t = n\] is \[{\user1{u}\_n} = {\user1{S}^n}{\user1{u}\_0}\]. Notice that this computation involves taking powers of the matrix \[\user1{S}\]. Taking large powers of a general matrix is computationally difficult but taking powers of a diagonal matrix is easy. Therefore, from the eigendecomposition of \[\user1{S}\] we can write \[{\user1{S}^n} = {\left( {\user1{PD}{\user1{P}^{ - 1}}} \right)^n} = \user1{P}{\user1{D}^n}{\user1{P}^{ - 1}}\]. Hence, the eigenvector basis is ideal when taking powers of a matrix.
* \*\*Differential Equations:\*\* Consider a system of linear differential equations. We can describe the system of equations using a linear transformation represented by matrix \[\user1{A}\]. It often becomes easy to solve the system if we can propose a change of variables that decouples the variables in the equations. Using the eigenvectors of \[\user1{A}\] as a new basis, we can decouple the variables in the system. See [here](https://math.stackexchange.com/questions/23312/what-is-the-importance-of-eigenvalues-eigenvectors) for a great example.
* \*\*Rank-1 Decomposition:\*\* For symmetric matrices such as covariance matrices, we can write the eigen-decomposition as \[\user1{A} = \user1{Q}\Lambda {\user1{Q}^T}\]. We can expand this factorization as a sum of rank-1 matrices such that \[\user1{A} = {\lambda \_1}{\user1{q}\_1}\user1{q}\_1^T + \cdots + {\lambda \_n}{\user1{q}\_n}\user1{q}\_n^T\]. This enables applications such as PCA.

We can find the eigenvalues and eigenvectors for a matrix \[\user1{A}\] by solving the equation \[\user1{Ax = }\lambda \user1{x}\]. We can rewrite this equation as \[\left( {\user1{A} - \lambda \user1{I}} \right)\user1{x} = \user1{0}\]. This implies that the matrix \[\user1{A} - \lambda \user1{I}\] must be singular, otherwise the only solutions are the trivial solutions. Therefore, since \[\user1{A} - \lambda \user1{I}\] is singular we know that \[\det \left( {\user1{A} - \lambda \user1{I}} \right) = 0\]. Once we have found the eigenvalues, then we can use elimination to find the nullspace of \[\user1{A} - \lambda \user1{I}\]. The vectors in the nullspace are eigenvectors of \[\user1{A}\] with eigenvalue \[\lambda \].

As an example, consider the projection matrix defined as \[\user1{P} = \user1{A}{\left( {{\user1{A}^T}\user1{A}} \right)^{ - 1}}{\user1{A}^T}\]. The projection matrix will project any vector onto the column space of \[\user1{A}\]. Therefore, any vector in the column space of \[\user1{A}\] will be an eigenvector of \[\user1{P}\]. We can see this since the column space of \[\user1{A}\] is defined by the columns of \[\user1{A}\], so \[\user1{PA} = \user1{A}{\left( {{\user1{A}^T}\user1{A}} \right)^{ - 1}}{\user1{A}^T}\user1{A} = \user1{A}\]. Therefore, we see that the columns of \[\user1{A}\] are the eigenvectors of \[\user1{P}\] with eigenvalues \[\lambda = 1\]. Also, any vector orthogonal to the column space will be an eigenvector of \[\user1{P}\] with eigenvalue \[\lambda = 0\]. This can be seen since if \[\user1{x}\] is orthogonal to the columns of \[\user1{A}\], then \[{\user1{A}^T}\user1{x} = \user1{0}\] and \[\user1{Px} = \user1{0}\].

Suppose that we have \[n\] linearly independent eigenvectors of \[\user1{A}\] and we put these vectors as the columns of \[\user1{S}\]. Then we can write\[\user1{S = }\left[ {\begin{array}{\*{20}{c}}{{\user1{x}\_1}}& \cdots &{{\user1{x}\_n}}\end{array}} \right]\]. Multiplying we can get \[\user1{AS = }\left[ {\begin{array}{\*{20}{c}}{\user1{A}{\user1{x}\_1}}& \cdots &{\user1{A}{\user1{x}\_n}}\end{array}} \right]\user1{ = }\left[ {\begin{array}{\*{20}{c}}{{\lambda \_1}{\user1{x}\_1}}& \cdots &{{\lambda \_n}{\user1{x}\_n}}\end{array}} \right] = \user1{S}\Lambda \], where \[\Lambda \] is a diagonal matrix with the eigenvalues on the diagonal. Then we can multiply by the inverse to get \[\user1{A} = \user1{S}\Lambda {\user1{S}^{ - 1}}\].

What can we say about the eigenvalues and eigenvectors of \[{\user1{A}^2}\]? If \[\user1{Ax = }\lambda \user1{x}\] then we can multiply both sides by \[\user1{A}\] to get \[{\user1{A}^2}\user1{x = }\lambda \user1{Ax = }{\lambda ^2}\user1{x}\]. Therefore, we see that \[{\user1{A}^2}\] has the same eigenvectors as \[\user1{A}\] and its eigenvalues are squared. We can also see this from \[{\user1{A}^2} = \user1{S}\Lambda {\user1{S}^{ - 1}}\user1{S}\Lambda {\user1{S}^{ - 1}} = \user1{S}{\Lambda ^2}{\user1{S}^{ - 1}}\]. This is an incredibly powerful property of the eigenvalue diagonalization factorization.

In order to diagonalize a matrix, we need to have \[n\] independent eigenvectors. When can we be certain that a matrix is diagonalizable? Two conditions that will guarantee that the matrix is diagonalizable are: 1) the matrix is symmetric (this comes from the spectral theorem), and 2) all eigenvalues of the matrix are different.

To see why \[n\] independent eigenvectors exist if all the eigenvalues are different, consider the following proof. Suppose that we have \[n\] different eigenvalues \[{\lambda \_1} \ldots ,{\lambda \_n}\], but a dependent eigenvector. Then there exists a linear combination of \[{\user1{x}\_k}\] in terms of the other independent eigenvectors such that \[{\user1{x}\_k} = {c\_1}{\user1{x}\_1} + \cdots + {c\_n}{\user1{x}\_n}\] . From the definition of an eigenvector, we know that \[\user1{A}{\user1{x}\_k} = {\lambda \_k}{\user1{x}\_k}\]. Substituting the linear combination on both sides gives:

\[\begin{array}{l}\user1{A}{\user1{x}\_k} = {\lambda \_k}{\user1{x}\_k}\\ \Rightarrow \user1{A}\left( {{c\_1}{\user1{x}\_1} + \cdots + {c\_n}{\user1{x}\_n}} \right) = {\lambda \_k}\left( {{c\_1}{\user1{x}\_1} + \cdots + {c\_n}{\user1{x}\_n}} \right)\\ \Rightarrow {c\_1}{\lambda \_1}{\user1{x}\_1} + \cdots + {c\_n}{\lambda \_n}{\user1{x}\_n} = {c\_1}{\lambda \_k}{\user1{x}\_1} + \cdots + {c\_n}{\lambda \_k}{\user1{x}\_n}\\ \Rightarrow {c\_1}\left( {{\lambda \_1} - {\lambda \_k}} \right){\user1{x}\_1} + \cdots + {c\_n}\left( {{\lambda \_n} - {\lambda \_k}} \right){\user1{x}\_n} = 0\end{array}\]

Since all the eigenvectors are different, we see that the \[{\lambda \_j} - {\lambda \_k} \ne 0\], hence the collection of vectors must be dependent which is a contradiction. Therefore, all the eigenvectors must be independent. If we have repeated eigenvalues, we may or may not have \[n\] independent eigenvectors. For example, the identity matrix has eigenvalues all equal to 1, but has \[n\] independent eigenvectors.

Some useful properties of eigenvalues are the following:

* The eigenvalues of \[\user1{AB}\] equals the eigenvalues of \[\user1{BA}\].
* The sum of the eigenvalues of \[\user1{A}\] is equal to the sum of the diagonal entries of\[\user1{A}\] (i.e. the trace).
* The product of the eigenvalues of \[\user1{A}\] is equal to the determinant of \[\user1{A}\].
* If we add a multiple of the identity matrix to \[\user1{A}\], then the eigenvectors will stay the same for all multiples, but the eigenvalues will increase by the scaling factor. To see this, suppose that \[\user1{x}\] is an eigenvector associated with an eigenvalue \[\lambda \], then \[\left( {\user1{A} + c\user1{I}} \right)\user1{x} = \user1{Ax + }c\user1{Ix = }\lambda \user1{x + }c\user1{x} = \left( {\lambda + c} \right)\user1{x}\].
* A matrix with real valued entries can have complex eigenvalues and eigenvectors. However, a symmetric matrix will always have real eigenvalues. On the other hand, antisymmetric matrices where \[{\user1{A}^T} = - \user1{A}\] have all imaginary eigenvalues.
* The eigenvalues of a triangular matrix are equal to the diagonal entries of the matrix.
* For a symmetric matrix, the signs of the pivots of the symmetric matrix in reduced row echelon form tell us the signs of the eigenvalues.

## Solving Difference Equations

Suppose that we have a difference equation such that \[{\user1{u}\_{k + 1}} = \user1{A}{\user1{u}\_k}\], then for some initial starting condition \[{\user1{u}\_0}\] we can write the \[{k^{th}}\] state as \[{\user1{u}\_k} = {\user1{A}^k}{\user1{u}\_0}\]. We can write the vector as a linear combination of the eigenvectors of \[\user1{A}\], such that \[{\user1{u}\_0} = {c\_1}{\user1{x}\_1} + \cdots + {c\_n}{\user1{x}\_n}\]. Then when we multiply by \[\user1{A}\] we get: \[\user1{A}{\user1{u}\_0} = {c\_1}{\lambda \_1}{\user1{x}\_1} + \cdots + {c\_n}{\lambda \_n}{\user1{x}\_n}\]. Therefore, the \[{k^{th}}\] state is \[{\user1{A}^k}{\user1{u}\_0} = {c\_1}\lambda \_1^k{\user1{x}\_1} + \cdots + {c\_n}\lambda \_n^k{\user1{x}\_n} = {\Lambda ^k}\user1{Sc}\].

As an application of the difference equations, we can examine the Fibonacci numbers. We can write the Fibonacci equation as \[{\user1{u}\_k} = \left[ {\begin{array}{\*{20}{c}}{{F\_{k + 1}}}\\{{F\_k}}\end{array}} \right]\] and \[{\user1{u}\_{k + 1}} = \left[ {\begin{array}{\*{20}{c}}1&1\\1&0\end{array}} \right]{\user1{u}\_k}\]. Finding the eigenvalues and eigenvectors of this matrix can allow us to calculate the Fibonacci numbers.

## Solving Differential Equations

We can visualize a matrix as a linear operator that acts on vectors, and we can visualize the derivative as a linear operator that acts on functions. An interesting connection between the two is the connection between the eigenvalues of a matrix and the solutions to a linear differential equation to a linear differential equation. For example, note that the solutions to the equation \[\user1{Ax = }\lambda \user1{x}\] are the eigenvectors (assuming that \[\user1{A}\] is a square matrix with independent eigenvectors). Likewise, for the differential equations, the solution to the equation \[{D\_\user1{x}}\,f\left( \user1{x} \right) = \lambda f\left( \user1{x} \right)\] are functions called eigenfunctions. For first order linear differential equations, the solutions are \[f\left( \user1{x} \right) = \left[ {\begin{array}{\*{20}{c}}{{c\_1}{e^{\lambda {x\_1}}}}\\ \vdots \\{{c\_n}{e^{\lambda {x\_n}}}}\end{array}} \right]\]. Now suppose that we have a system of differential equations such that \[{D\_t}\,\left[ {\begin{array}{\*{20}{c}}{{u\_1}\left( t \right)}\\{{u\_2}\left( t \right)}\end{array}} \right] = \user1{A}\left[ {\begin{array}{\*{20}{c}}{{u\_1}\left( t \right)}\\{{u\_2}\left( t \right)}\end{array}} \right]\]. In this case, we are now combining the differentiation operator with a vector of functions that is transformed by a matrix. We could write \[\left[ {\begin{array}{\*{20}{c}}{{u\_1}\left( t \right)}\\{{u\_2}\left( t \right)}\end{array}} \right]\] as a linear combination of the eigenvectors such that \[\left[ {\begin{array}{\*{20}{c}}{{u\_1}\left( t \right)}\\{{u\_2}\left( t \right)}\end{array}} \right] = {f\_1}\left( t \right){\user1{x}\_1} + {f\_2}\left( t \right){\user1{x}\_2}\] for some functions \[{f\_1}\left( t \right)\] and \[{f\_2}\left( t \right)\]. Substituting this into the equation above we get

\[\begin{array}{l}{D\_t}\,\left( {{f\_1}\left( t \right){\user1{x}\_1} + {f\_2}\left( t \right){\user1{x}\_2}} \right) = \user1{A}\left( {{f\_1}\left( t \right){\user1{x}\_1} + {f\_2}\left( t \right){\user1{x}\_2}} \right)\\ \Rightarrow {D\_t}\,{f\_1}\left( t \right){\user1{x}\_1} + {D\_t}\,{f\_2}\left( t \right){\user1{x}\_2} = {f\_1}\left( t \right){\lambda \_1}{\user1{x}\_1} + {f\_2}\left( t \right){\lambda \_2}{\user1{x}\_2}\end{array}\]

This implies that \[{D\_t}\,{f\_1}\left( t \right) = {f\_1}\left( t \right){\lambda \_1}\] and \[{D\_t}\,{f\_2}\left( t \right) = {f\_2}\left( t \right){\lambda \_2}\]. The general solutions to these equations are \[\left[ {\begin{array}{\*{20}{c}}{{f\_1}\left( t \right)}\\{{f\_2}\left( t \right)}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{{c\_1}{e^{{\lambda \_1}t}}}\\{{c\_2}{e^{{\lambda \_2}t}}}\end{array}} \right]\]. Therefore, we see that the general solution is \[\left[ {\begin{array}{\*{20}{c}}{{u\_1}\left( t \right)}\\{{u\_2}\left( t \right)}\end{array}} \right] = {f\_1}\left( t \right){\user1{x}\_1} + {f\_2}\left( t \right){\user1{x}\_2} = {c\_1}{e^{{\lambda \_1}t}}{\user1{x}\_1} + {c\_2}{e^{{\lambda \_2}t}}{\user1{x}\_2}\]

We can solve for the constants in this solution by using the initial conditions for the system. To generalize this idea for any system, suppose that we have the equation\[{D\_t}\,\user1{u}\left( t \right) = \user1{Au}\left( t \right)\]. We can write the values of \[\user1{u}\left( t \right)\] at any given time point as a linear combination of the eigenvectors of \[\user1{A}\] (assuming the eigenvectors are independent and span the complete space) such that \[\user1{u}\left( t \right) = \user1{Sv}\left( t \right)\]. We can write the equation as follows:

\[\begin{array}{l}{D\_t}\,\user1{Sv}\left( t \right) = \user1{ASv}\left( t \right)\\ \Rightarrow \user1{S}\left( {{D\_t}\,\user1{v}\left( t \right)} \right) = \user1{ASv}\left( t \right)\\ \Rightarrow {D\_t}\,\user1{v}\left( t \right) = {\user1{S}^{ - 1}}\user1{ASv}\left( t \right) = \Lambda \user1{v}\left( t \right)\end{array}\]

The general solution to this system is given as

\[\begin{array}{l}\user1{v}\left( t \right) = {e^{\Lambda t}}\user1{v}\left( 0 \right)\\ \Rightarrow {\user1{S}^{ - 1}}\user1{u}\left( t \right) = {e^{\Lambda t}}{\user1{S}^{ - 1}}\user1{u}\left( 0 \right)\\ \Rightarrow \user1{u}\left( t \right) = \user1{S}{e^{\Lambda t}}{\user1{S}^{ - 1}}\user1{u}\left( 0 \right) = {e^{\user1{A}t}}\user1{u}\left( 0 \right)\end{array}\]

What is a matrix exponential? We can write out the Taylor series for the exponential as

\[{e^x} = 1 + x + \frac{{{x^2}}}{2} + \frac{{{x^3}}}{6} + \cdots \]

We can define the matrix exponential by extending this formula such that

\[{e^{At}} = \user1{I} + \user1{A}t + \frac{{{{\left( {\user1{A}t} \right)}^2}}}{2} + \frac{{{{\left( {\user1{A}t} \right)}^3}}}{6} + \cdots \]

Using the eigenvalue decomposition for \[\user1{A}\], we can write

\[\begin{array}{l}{e^{At}} = \user1{S}{\user1{S}^{ - 1}} + \left( {\user1{S}\Lambda {\user1{S}^{ - 1}}} \right)t + \frac{{\user1{S}{\Lambda ^2}{\user1{S}^{ - 1}}}}{2}{t^2} + \frac{{\user1{S}{\Lambda ^3}{\user1{S}^{ - 1}}}}{6}{t^3} + \cdots \\ = \user1{S}{e^{\Lambda t}}{\user1{S}^{ - 1}}\end{array}\]

To calculate \[{e^{\Lambda t}}\] we recall that \[\Lambda \] is the diagonal matrix with the eigenvalues on the diagonal. When we substitute this into the Taylor series expansion, we see that the result of the sum is a diagonal matrix where the \[{k^{th}}\] diagonal entry is itself a Taylor series expansion for \[{e^{{\lambda \_k}t}}\]. Therefore, we see that

\[{e^{\Lambda t}} = \left[ {\begin{array}{\*{20}{c}}{{e^{{\lambda \_1}t}}}&0& \cdots &0\\0&{{e^{{\lambda \_2}t}}}&{}&0\\ \vdots &{}& \ddots & \vdots \\0& \cdots &0&{{e^{{\lambda \_n}t}}}\end{array}} \right]\]

We can understand the stability of a system by examining the eigenvalues of the system. We have three possibilities:

1. \*\*Transient:\*\* We see that the solution will go to zero if for all eigenvalues \[{\mathop{\rm Re}\nolimits} \left( \lambda \right) < 0\]. To see why, remember that if \[\lambda = a + bi\] we can write \[{e^{\lambda t}} = {e^{\left( {a + bi} \right)t}} = {e^{at}}{e^{bit}} = {e^{at}}\left( {\cos \,\,bt + i\sin \,bt} \right)\]. Hence, the imaginary part causes a rotational component with unit length. What controls the scale is the real part.
2. \*\*Steady State:\*\* If at least one eigenvalue is 0 and all other eigenvalues have negative real part then the system will converge to a steady state response
3. \*\*Explode:\*\* If \[{\mathop{\rm Re}\nolimits} \left( \lambda \right) > 0\] for any eigenvalue, then the system will explode

## Markov Chains

A Markov matrix is defined such that all entries are nonnegative and the sum of the entries in each column is 1. A Markov matrix \[\user1{A}\] will have one eigenvalue which equals 1, and the magnitude of all other eigenvalues will be less than 1. The way we can see that 1 is an eigenvalue of a Markov matrix is to recognize that since all the columns add to 1, then subtracting the identity matrix from the Markov matrix will make it such that the columns of \[\user1{A} - \user1{I}\] sum to 0. Therefore, the sum of the row vectors must be the zero vector which means that the rows of \[\user1{A} - \user1{I}\] are dependent. Hence, \[\user1{A} - \user1{I}\] is singular and has eigenvalue 0, and the matrix \[\user1{A}\] must have eigenvalue 1. If we have the difference equation \[{\user1{u}\_{k + 1}} = \user1{A}{\user1{u}\_k}\] and the matrix \[\user1{A}\] is Markov, then we can solve for the long-term state probabilities by using the eigenvalues and eigenvectors of \[\user1{A}\].

## Symmetric Matrices

Symmetric matrices are the most important class of matrices. Symmetric matrices have the property \[{\user1{A}^T} = \user1{A}\]. Symmetric matrices have the property that the eigenvalues are real, and the eigenvectors are orthogonal (given from the spectral theorem).

Usually, we can write the eigendecomposition of a matrix as \[\user1{A} = \user1{S}\Lambda {\user1{S}^{ - 1}}\]. However, when the matrix is symmetric, we can write \[\user1{A} = \user1{Q}\Lambda {\user1{Q}^T}\]. This is called the spectral theorem or principal axis theorem.

To show that the eigenvalues must be real for a symmetric matrix we can write the expression \[\user1{Ax = }\lambda \user1{x}\]. Regardless of the eigenvalues we can write the complex conjugate as

\[\begin{array}{l}\user1{\bar A\bar x = }\bar \lambda \user1{\bar x}\\ \Rightarrow \user1{A\bar x = }\bar \lambda \user1{\bar x}\end{array}\]

Which follows if the matrix has real components. If we transpose this equation, we get

\[\begin{array}{l}{{\user1{\bar x}}^T}{\user1{A}^T}\user1{ = }\bar \lambda {{\user1{\bar x}}^T}\\ \Rightarrow {{\user1{\bar x}}^T}\user1{A = }\bar \lambda {{\user1{\bar x}}^T}\\ \Rightarrow {{\user1{\bar x}}^T}\user1{Ax = }\bar \lambda {{\user1{\bar x}}^T}\user1{x}\\ \Rightarrow {{\user1{\bar x}}^T}\lambda \user1{x = }\bar \lambda {{\user1{\bar x}}^T}\user1{x}\end{array}\]

This implies that \[\lambda = \bar \lambda \] since the dot product \[{\user1{\bar x}^T}\user1{x}\] represents the length of the vector and will be nonzero since the eigenvectors are nonzero. Therefore, we see that the eigenvalues must be real. The equivalent to symmetric matrices for complex entries are called Hermitian matrices and \[{\user1{\bar A}^T} = \user1{A}\].

We can expand the eigenvalue factorization into rank 1 matrices such that

\[\user1{A} = \user1{Q}\Lambda {\user1{Q}^T} = {\lambda \_1}{\user1{q}\_1}{\user1{q}\_1}^T + \cdots + {\lambda \_n}{\user1{q}\_n}{\user1{q}\_n}^T\]

Recall that if we want to project a vector \[\user1{b}\] onto the vector \[\user1{q}\], then we have the normal equation for the error such that \[{\user1{q}^T}\left( {\user1{b} - x\user1{q}} \right) = 0\]. Solving for the coefficient we get \[x = \frac{{{\user1{q}^T}\user1{b}}}{{{\user1{q}^T}\user1{q}}}\]. Now computing the projection, we get \[x\user1{q} = \user1{q}\frac{{{\user1{q}^T}\user1{b}}}{{{\user1{q}^T}\user1{q}}} = \left( {\frac{{\user1{q}{\user1{q}^T}}}{{{\user1{q}^T}\user1{q}}}} \right)\user1{b}\]. Therefore, we see that the matrix \[\frac{{\user1{q}{\user1{q}^T}}}{{{\user1{q}^T}\user1{q}}}\] is the projection matrix to project onto the vector \[\user1{q}\]. Furthermore, if \[\user1{q}\] is unit length, then the projection matrix simply becomes \[\user1{q}{\user1{q}^T}\].

Another way to interpret this projection matrix is to consider what the expression \[\frac{{\user1{q}{\user1{q}^T}}}{{{\user1{q}^T}\user1{q}}}\user1{b}\] means. Rearranging and expanding using the definition of the dot product, we see:

\[\frac{{\user1{q}{\user1{q}^T}}}{{{\user1{q}^T}\user1{q}}}\user1{b = q}\frac{{{\user1{q}^T}\user1{b}}}{{{\user1{q}^T}\user1{q}}} = \user1{q}\frac{{\left\| \user1{q} \right\|\left\| \user1{b} \right\|\cos \theta }}{{{\user1{q}^T}\user1{q}}} = \frac{\user1{q}}{{\left\| \user1{q} \right\|}}\left\| \user1{b} \right\|\cos \theta \]

Therefore, the projection computes a scaling factor \[\left\| \user1{b} \right\|\cos \theta \] which we can see after drawing a triangle with \[\user1{b}\] as the hypotenuse and a ray along the vector \[\user1{q}\], represents the distance along \[\user1{q}\] where the perpendicular projection of \[\user1{b}\] occurs. Hence multiplying this scaling factor by the unit vector \[\frac{\user1{q}}{{\left\| \user1{q} \right\|}}\] gives the vector representing the projection of \[\user1{b}\] onto \[\user1{q}\].

Returning to the eigenvector decomposition written as \[\user1{A} = \user1{Q}\Lambda {\user1{Q}^T} = {\lambda \_1}{\user1{q}\_1}{\user1{q}\_1}^T + \cdots + {\lambda \_n}{\user1{q}\_n}{\user1{q}\_n}^T\], we see that each component \[{\lambda \_k}{\user1{q}\_k}{\user1{q}\_k}^T\] represents a perpendicular projection onto the unit eigenvector \[{\user1{q}\_k}\] where the eigenvalue \[{\lambda \_k}\] describes how much this projection contributes to the action of the matrix. Hence, every symmetric matrix can be interpreted as a combination of perpendicular projection matrices.

# Special Matrices

In this section we will discuss some classes of matrices which have special properties.

## Positive Definite Matrices

How do we know if a symmetric matrix is positive definite? There are a few different kinds of tests for positive definiteness: 1) are the eigenvalues all positive, 2) are the sub-determinants all positive, 3) are the pivots all positive, 4) \[{\user1{x}^T}\user1{Ax > 0}\] at all nonzero points.

Given that the eigenvalues of a positive definite matrix are positive, then we know that the inverse of the matrix must also be positive definite since the eigenvalues are reciprocals which preserves the sign. If \[\user1{A}\] and \[\user1{B}\] are positive definite, then we can easy show that \[{\user1{x}^T}\left( {\user1{A} + \user1{B}} \right)\user1{x} = {\user1{x}^T}\user1{Ax} + {\user1{x}^T}\user1{Bx} > 0\], so \[\user1{A} + \user1{B}\] is positive definite.

## Gram Matrices

Suppose we have an \[m\] by \[n\] matrix \[\user1{A}\]. We know that \[{\user1{A}^T}\user1{A}\] is square and symmetric, is it also positive definite? Looking at the expression \[{\user1{x}^T}{\user1{A}^T}\user1{Ax} = {\left\| {\user1{Ax}} \right\|^2} \ge 0\], therefore \[{\user1{A}^T}\user1{A}\] is positive semidefinite. For \[{\user1{A}^T}\user1{A}\] to be positive definite, we must ensure that \[\user1{Ax} \ne 0\] except for the zero vector. This means that the nullspace is empty and the column space must be full rank. Therefore, we can ensure that \[{\user1{A}^T}\user1{A}\] is positive definite if the rank of \[\user1{A}\] is \[n\]. The matrix \[{\user1{A}^T}\user1{A}\] is called the Gram matrix and is always semi-positive definite. If \[\user1{A}\] is invertible, then \[{\user1{A}^T}\user1{A}\] will be positive definite.

## Similar Matrices

Suppose that we have two square matrices \[\user1{A}\] and \[\user1{B}\]. These matrices are called similar if there exists a matrix \[\user1{M}\] such that \[\user1{B} = {\user1{M}^{ - 1}}\user1{AM}\]. From the eigenvalue decomposition, we can show that \[{\user1{S}^{ - 1}}\user1{AS = }\Lambda \] which implies that \[\user1{A}\] is similar to the eigenvalue matrix \[\Lambda \]. From here we can now define a family of similar matrices where we can transform the matrix such that for any invertible matrix \[\user1{D}\] we can write \[{\user1{D}^{ - 1}}{\user1{S}^{ - 1}}\user1{ASD = }{\user1{D}^{ - 1}}\Lambda \user1{D}\]. Notice that \[{\user1{D}^{ - 1}}\Lambda \user1{D}\] is another eigenvalue decomposition with the same eigenvalues but different eigenvectors. Hence \[\user1{A}\] is similar to all other matrices with the same eigenvalues.

Therefore, similar matrices must have the same eigenvalues. To show this, we can write the eigenvalue decomposition as \[\user1{Ax} = \lambda \user1{x}\]. Now suppose that we have a similar matrix \[\user1{B} = {\user1{M}^{ - 1}}\user1{AM}\]. We can write out the eigenvalue decomposition as

\[\begin{array}{l}\user1{Ax} = \lambda \user1{x}\\ \Rightarrow \user1{MB}{\user1{M}^{ - 1}}\user1{x = }\lambda \user1{x}\\ \Rightarrow \user1{B}{\user1{M}^{ - 1}}\user1{x = }\lambda {\user1{M}^{ - 1}}\user1{x}\\ \Rightarrow \user1{By = }\lambda \user1{y}\end{array}\]

Therefore, we see that the eigenvalues are preserved, but the eigenvectors have been transformed \[\user1{y} = {\user1{M}^{ - 1}}\user1{x}\].

Suppose that we now consider matrices with the same eigenvalues which may not be diagonalizable. Consider a matrix which is a multiple of the identity \[\user1{A} = c\user1{I}\] where all the eigenvalues equal \[c\]. We see that multiplying by any invertible matrix does not change the matrix since \[{\user1{M}^{ - 1}}\user1{AM} = c{\user1{M}^{ - 1}}\user1{IM} = c\user1{I}\]. Therefore, the matrix \[\user1{A} = c\user1{I}\] only has one matrix in its family.

If we now look at the matrix \[\left[ {\begin{array}{\*{20}{c}}4&1\\0&4\end{array}} \right]\], clearly the eigenvalues are both 4. However, this matrix has a larger number of similar matrices than \[\left[ {\begin{array}{\*{20}{c}}4&0\\0&4\end{array}} \right]\] because any matrix \[\left[ {\begin{array}{\*{20}{c}}4&c\\0&4\end{array}} \right]\] will be similar. The most diagonal representative from each family of similar matrices is called the Jordan normal form. This enables us to effectively “complete” the diagonalization process for matrices which cannot be diagonalized.

A Jordan block is a partitioning of the matrix where each block has one eigenvector. Every square matrix \[\user1{A}\] is similar to a Jordan matrix \[\user1{J}\], where \[\user1{J}\] is a partitioned matrix where each block has one associated eigenvector. To understand this, start with any \[\user1{A}\]. If the eigenvalues are distinct, then the matrix is diagonalizable and this matrix is similar to its diagonal eigenvector matrix. In this case, the Jordan form is \[\user1{J} = \Lambda \]. Now, if the eigenvalues are repeated and “missing” eigenvectors, then its Jordan matrix will have partitioned blocks that each have one associated eigenvector.

## Orthogonal Matrices

Consider that we have a projection (or expansion) onto an orthonormal basis defined by \[{\user1{q}\_1}, \ldots ,{\user1{q}\_n}\]. We can write this projection as \[\user1{v} = {x\_1}{\user1{q}\_1} + \cdots + {x\_n}{\user1{q}\_n}\]. An important question is how we find the corresponding coefficients. For an orthonormal basis, this process is very easy, we can just take the dot product of both sides of this equation by the corresponding basis. For example, for the \[{k^{th}}\] coefficient, the coefficient is defined as \[\user1{q}\_k^T\user1{v = }{x\_k}\]. We can write this in matrix notation as \[\user1{Qx = v}\]. Given that \[{\user1{Q}^{ - 1}} = {\user1{Q}^T}\], we can write \[\user1{x = }{\user1{Q}^T}\user1{v}\].

An orthogonal matrix is defined as a matrix where each column and row is perpendicular to every other column and row respectively. Specifically, a matrix \[\user1{Q}\]is called orthonormal if \[{\user1{Q}^T}\user1{Q} = \user1{I}\] where \[\user1{I}\] is the identity matrix. We can write this matrix in a rank-1 decomposition as \[{\user1{Q}^T}\user1{Q} = {\user1{q}\_1}\user1{q}\_1^T + \cdots + {\user1{q}\_n}\user1{q}\_n^T = \user1{I}\].

We can show that if \[\user1{Q}\] has orthonormal columns then it must have orthonormal rows as well. If \[\user1{Q}\] has orthonormal columns, then it follows that \[{\user1{Q}^T}\user1{Q} = \user1{I}\]. By left multiplying this expression by \[\user1{Q}\], then we can rearrange the expression as follows

\[\begin{array}{l}{\user1{Q}^T}\user1{Q} = \user1{I}\\\user1{Q}{\user1{Q}^T}\user1{Q} = \user1{Q}\\\left( {\user1{Q}{\user1{Q}^T} - \user1{I}} \right)\user1{Q} = \user1{0}\end{array}\]

Given that \[\user1{Q}\] has full rank (since all columns are mutually orthogonal and nonzero), it follows that any nonzero row in \[\user1{Q}{\user1{Q}^T} - \user1{I}\] can be written as a linear combination of the columns of \[\user1{Q}\]. As such, the product of a nonzero row of \[\user1{Q}{\user1{Q}^T} - \user1{I}\] with \[\user1{Q}\] must result in a nonzero vector. Therefore, since the product of all rows of \[\user1{Q}{\user1{Q}^T} - \user1{I}\] with \[\user1{Q}\] result in zero vectors (as seen from the equation \[\left( {\user1{Q}{\user1{Q}^T} - \user1{I}} \right)\user1{Q} = \user1{0}\]), it follows that \[\user1{Q}{\user1{Q}^T} - \user1{I} = 0\] which implies that \[\user1{Q}{\user1{Q}^T} = \user1{I}\]. From this result we see that \[\user1{Q}{\user1{Q}^T} = {\user1{Q}^T}\user1{Q} = \user1{I}\]. Therefore, \[{\user1{Q}^{ - 1}}\user1{ = }{\user1{Q}^T}\] and the rows of \[\user1{Q}\] must also be orthonormal.

## Graph Matrices

A graph is a set of nodes and edges connecting the nodes. In the incidence matrix, every row corresponds to an edge. Loops in the graph correspond to linearly dependent rows. If we have an incidence matrix \[\user1{A}\] , how can we interpret the \[\user1{x}\] in the product \[\user1{Ax}\]? We can think about the element \[{\user1{x}\_k}\] as representing the potential of the \[{k^{th}}\] node. Therefore, the product \[\user1{Ax}\] tells us the potential differences between the nodes. In this way, the nullspace of \[\user1{Ax}\] can be visualized as the settings of the potentials \[\user1{x}\] such that the potential differences are 0. Therefore, we can write the vectors in the nullspace as \[\user1{x} = c\left[ {\begin{array}{\*{20}{c}}1\\ \vdots \\1\end{array}} \right]\] where \[c\] is a constant. In terms of an electrical network, the potential difference is zero on each edge if each node has the same potential. We can’t tell what that potential is by observing the flow of electricity through the network, since only the potential differences correspond to flow. Therefore, only if one of the nodes is grounded to potential 0 can we determine the absolute potential of all other nodes in the graph. If we ground a node in the network, we are setting one of the nodes in \[\user1{x}\] to 0, which effectively removes one of the columns from \[\user1{A}\].

For an incidence matrix \[\user1{A}\] what does the nullspace of \[{\user1{A}^T}\user1{y = 0}\] correspond to? We can think about the element \[{\user1{y}\_l}\] as representing the current flowing along the \[{l^{th}}\] edge. Therefore, the sum of the currents into and out of every node must be 0. The matrix product \[{\user1{A}^T}\user1{y}\] tells us the net current at each node. Hence the solutions to \[{\user1{A}^T}\user1{y = 0}\] are the currents that follow Kirchhoff’s current law. The basis vectors for the nullspace \[{\user1{A}^T}\user1{y = 0}\] are the currents required for each of the loops to satisfy Kirchhoff’s current law. We can add current sources to \[{\user1{A}^T}\user1{y = f}\] where \[\user1{f}\] are the current sources.

# Change of Basis

A central question in linear algebra is the following: given a vector space \[V\], what is the best basis to choose to represent the vectors? For example, the vector space \[V\] could the space of all grayscale intensity images, where each image is a vector in this space. A grayscale intensity image by definition is a \[m\,\, \times \,\,n\] matrix in \[{\mathbb{R}^{m\,\, \times \,\,n\,\,}}\]. The most straightforward basis to choose is simply the standard basis where the \[\left( {i,j} \right)\] element of the coordinate vector describes the pixel intensity of the \[\left( {i,j} \right)\] element of the image. However, for specific applications such as image compression, this is not a very helpful basis, as we would prefer a basis where each basis vector describes global characteristics of the image. Several different basis representations have been researched include the following

* \*\*Fourier Basis:\*\* The Fourier basis is composed of complex exponentials and can be interpreted as representing a vector in “frequency” domain. This can be useful for a multitude of applications including filtering and compression
* \*\*Wavelet Basis:\*\* Wavelets are another orthogonal basis that like the Fourier basis, attempts to represent a vector in a “frequency” domain. Wavelets have several properties which are different than the Fourier basis.
* \*\*Eigenvalue Basis:\*\* If we represent a vector using the eigenvalue basis determined from a matrix, then the linear transformation is simply a scaling. This has great advantage when recurrently applying a transformation as in the application of difference equations.

## Fourier Basis

The goal of Fourier analysis is to represent a signal using a set of complex exponential basis functions. The primary reasons why complex exponentials are used as basis functions is because they have the following properties:

1. \*\*Periodic:\*\* A complex exponential basis function has a single associated frequency. Therefore, by using complex exponentials as basis functions, we can represent a signal in terms of composite frequency components. By computing the projection of the signal onto these basis functions, we can effectively transform a signal from the time/spatial domain into the frequency domain.
2. \*\*Orthogonal:\*\* Having orthogonal basis functions leads to the derivation of elegant and simple formulas to calculate the transform coefficients when projecting the signal onto the basis functions.

To begin our adventure into Fourier analysis, let us start with a question. Suppose we have a periodic real valued function \[f\left( t \right)\] that has period \[P\] such that \[f\left( t \right) = f\left( {t + P} \right)\]. Is it possible to find a set of coefficients \[{c\_k}\] such that we can represent the signal for \[N \to \infty \] as:

\[f\left( t \right) = \sum\limits\_{n = - N}^N {{c\_n}{e^{i\frac{{2\pi }}{P}nt}}} \]

To unpack this question, let us first examine the contents of the above equation. The complex exponential can be expanded as \[{e^{i\frac{{2\pi }}{P}nt}} = \cos \left( {\frac{{2\pi }}{P}nt} \right) + i\sin \left( {\frac{{2\pi }}{P}nt} \right)\]. Therefore, we see that the period of each complex exponential is \[\frac{P}{n}\]. A couple interesting things to note: 1) when \[n = 0\] the complex exponential reduces down to \[{e^{i\frac{{2\pi }}{P}0t}} = 1\] which is the constant function. As such, we can interpret the coefficient \[{c\_0}\] as specifying the average value of the function. 2) the smallest nonzero frequency we have in the sum is when \[n = \pm 1\] and the frequency is \[ \pm \frac{1}{P}\]. The reason why we don’t use a frequency smaller than \[ \pm \frac{1}{P}\] is that our signal is known to have a base frequency of \[\frac{1}{P}\], therefore, we don’t need any smaller frequencies to describe the signal. 3) The reason why we have negative values of \[n\] is so that the \[{c\_k}\] values can be complex conjugates for \[ \pm k\] so that the sum produces a real (potentially phase shifted) signal.

With this said, the central question becomes: how can we find the values for the coefficients \[{c\_k}\]? This is where we can exploit the orthogonality of the complex exponentials. We can compute the projection of the signal onto the \[{k^{th}}\] basis vector by multiplying by the \[{k^{th}}\] basis vector and integrating over the period \[P\] in the following way:

\[\begin{array}{l}f\left( t \right) = \sum\limits\_{n = - N}^N {{c\_n}{e^{i\frac{{2\pi }}{P}nt}}} \\ \Rightarrow \int\limits\_0^P {{e^{ - i\frac{{2\pi }}{P}kt}}f\left( t \right)dt} = \int\limits\_0^P {{e^{ - i\frac{{2\pi }}{P}kt}}\sum\limits\_{n = - N}^N {{c\_n}{e^{i\frac{{2\pi }}{P}nt}}} dt} \\ \Rightarrow \int\limits\_0^P {{e^{ - i\frac{{2\pi }}{P}kt}}f\left( t \right)dt} = \sum\limits\_{n = - N}^N {\int\limits\_0^P {{c\_n}{e^{ - i\frac{{2\pi }}{P}kt}}{e^{i\frac{{2\pi }}{P}nt}}dt} } \\ \Rightarrow \int\limits\_0^P {{e^{ - i\frac{{2\pi }}{P}kt}}f\left( t \right)dt} = P{c\_k}\\ \Rightarrow {c\_k} = \frac{1}{P}\int\limits\_0^P {{e^{ - i\frac{{2\pi }}{P}kt}}f\left( t \right)dt} \end{array}\]

We call this equation for the coefficient the “analysis” equation. The key to this result is the orthogonality of the basis vectors which tells us that

\[\int\limits\_0^P {{e^{ - i\frac{{2\pi }}{P}kt}}{e^{i\frac{{2\pi }}{P}nt}}dt} = \left\{ {\begin{array}{\*{20}{c}}{0\,\,\,k \ne n\,}\\{P\,\,k = n\,}\end{array}} \right.\]

Hence, we now have a formula for the coefficients of this series. We call this series the Fourier series of \[f\left( t \right)\].

We can write the series representation for \[f\left( t \right)\] as an equation known as the “synthesis” equation such that \[f\left( t \right) = \sum\limits\_{n = - N}^N {{c\_n}{e^{i\frac{{2\pi }}{P}nt}}} \].

A limitation of the Fourier series is that it assumes that \[f\left( t \right)\] is periodic with period \[P\]. How could we extend the concept of the Fourier series to handle nonperiodic functions? One approach is to continue assuming that \[f\left( t \right)\] is periodic, but let the period get very large such that \[P \to \infty \]. If we take the derivation that we developed for the coefficients of the Fourier series (written below with a shifted integral bound) and write out the synthesis equation for the Fourier series we get

\[f\left( t \right) = \sum\limits\_{n = - N}^N {\left( {\frac{1}{P}\int\limits\_{ - \frac{P}{2}}^{\frac{P}{2}} {{e^{ - i\frac{{2\pi }}{P}nt}}f\left( t \right)dt} } \right){e^{i\frac{{2\pi }}{P}nt}}} \]

We can define the frequency of the complex exponential as a function \[\omega \left( n \right) = \frac{n}{P}\]. We see that the difference between any two successive frequencies is \[\Delta \omega = \omega \left( {n + 1} \right) - \omega \left( n \right) = \frac{{n + 1}}{P} - \frac{n}{P} = \frac{1}{P}\]. Substituting into the expression, we see

\[f\left( t \right) = \sum\limits\_{n = - N}^N {\left( {\Delta \omega \int\limits\_{ - \frac{P}{2}}^{\frac{P}{2}} {{e^{ - i2\pi \omega t}}f\left( t \right)dt} } \right){e^{i2\pi \omega t}}} \]

Now, if we: 1) let \[P \to \infty \], so \[\Delta \omega \to d\omega \], 2) let \[N \to \infty \], and 3) swap the sum for an integral over \[\omega \], then we get the following

\[f\left( t \right) = \int\limits\_{ - \infty }^\infty {\left( {\int\limits\_{ - \infty }^\infty {{e^{ - i2\pi \omega t}}f\left( t \right)dt} } \right){e^{i2\pi \omega t}}d\omega } \]

The inner integral \[F\left( \omega \right) = \int\limits\_{ - \infty }^\infty {{e^{ - i2\pi \omega t}}f\left( t \right)dt} \] gives a function over \[\omega \] and this function is the Fourier transform of \[f\left( t \right)\]. The outer integral can be interpreted as the inverse Fourier transform of \[F\left( \omega \right)\].

An important question to understand is: what is the bound of \[\int\limits\_{ - \infty }^\infty {{e^{ - i2\pi \omega t}}f\left( t \right)dt} \]? We can safely assume that every real-life signal will have finite energy which means that\[\int\limits\_{ - \infty }^\infty {\left| {f\left( t \right)} \right|dt} = E\]. Now using the fact that \[{e^{ - i2\pi \omega t}}\] is periodic with amplitude 1, we see that \[\int\limits\_{ - \infty }^\infty {\left| {{e^{ - i2\pi \omega t}}f\left( t \right)} \right|dt} \le \int\limits\_{ - \infty }^\infty {\left| {f\left( t \right)} \right|dt} = E\].

Since any periodic signal can be written in terms of frequency components that are integer multiples of the base frequency of the signal, we only need a countably infinite number of complex exponentials to represent the periodic signal. This is the basis of the Fourier series. On the other hand, for infinite time signals that are aperiodic, we need to use all frequencies on the real number line to represent the signal. This leads to the Fourier integral transform where our integral produces a coefficient for each real frequency \[\omega \].

To develop intuition for the Fourier transform, we can think about the difference between the Fourier series and the Fourier transform, and the analogous relations between PMFs and PDFs from probability theory. For the Fourier series, we had a coefficient for each discrete frequency \[\frac{n}{P}\], with some of these frequencies having nonzero energy. This is analogous to the probability being located on discrete positions in the PMF.

Disclaimer: my knowledge of measure theory is minimal. Therefore, I have constructed this section mostly based on handwaving intuition. Please take it with a grain of salt.

Suppose that we want to represent the probability of landing anywhere on the real number line between \[\left( {0,1} \right]\]. Let us try to construct a random variable that can represent this distribution. Suppose that we have a random variable \[X\] associated with a uniform discrete distribution over the domain \[\left[ {\begin{array}{\*{20}{c}}{\frac{1}{N}}&{\frac{2}{N}}& \cdots &{\frac{N}{N}}\end{array}} \right]\]. The PMF associated with this variable is \[P\left( {X = \frac{k}{N}} \right) = \frac{1}{N}\] for any \[k \in \left\{ {1,2, \ldots ,N} \right\}\]. Consider what happens as we let \[N \to \infty \], we see that the PMF goes to zero for every point \[\mathop {\lim }\limits\_{N \to \infty } P\left( {X = \frac{k}{N}} \right) = \mathop {\lim }\limits\_{N \to \infty } \frac{1}{N} = 0\]. This poses a problem if we want the PMF to describe the entire real number line because all probabilities go to zero.

However, we have an even bigger problem. The fundamental problem is that we are trying to describe an interval on the real number line between \[\left( {0,1} \right]\] using a countably infinite set of points \[\left[ {\begin{array}{\*{20}{c}}{\frac{1}{N}}&{\frac{2}{N}}& \cdots &{\frac{N}{N}}\end{array}} \right]\]. As a result of measure theory, we know that the total measure of all the points in \[\left[ {\begin{array}{\*{20}{c}}{\frac{1}{N}}&{\frac{2}{N}}& \cdots &{\frac{N}{N}}\end{array}} \right]\] will be smaller than any interval of the real number line. Therefore, even if we let \[N \to \infty \], the domain \[\left[ {\begin{array}{\*{20}{c}}{\frac{1}{N}}&{\frac{2}{N}}& \cdots &{\frac{N}{N}}\end{array}} \right]\] cannot describe the complete interval \[\left( {0,1} \right]\].

Therefore, we need a new approach to represent probabilities on the real number line. One approach is to define a primitive measure, such that we can divide the real number line into a countable collection of these primitive measures. Then we can assign probability values to these primitive measures such that they sum to 1. A suitable primitive measure to represent an interval on the real number line is an infinitesimal interval since we can clearly represent any interval as a countable collection of smaller intervals.

This pattern of defining a primitive measure that we can use to use to represent the domain of our random variable as a countable set is key as it enables us to assign a nonzero probability to some of the primitive measures. This process can be generalized for different dimension spaces. For example: for a domain of discrete points, a primitive measure is a single point, for a domain of intervals of the real line, a primitive measure is an infinitesimal interval, for a domain of areas, a primitive measure is an infinitesimal area, for a domain of volumes, a primitive measure is an infinitesimal volume, etc..

Suppose that we represent the interval \[\left( {0,1} \right]\] as an ordered collection of intervals given as \[\left\{ {\left( {0,\frac{1}{N}} \right],\left( {0,\frac{2}{N}} \right], \ldots ,\left( {0,\frac{N}{N}} \right]} \right\}\]. Now suppose that each of these intervals has an associated probability value such that \[F\left( {\frac{k}{N}} \right)\] is the probability associated with interval \[\left( {0,\frac{k}{N}} \right]\]. Then we can write the sum of these probabilities as \[\sum\limits\_{n = 1}^N {F\left( {\frac{n}{N}} \right)} = 1\]. For a uniform distribution, the associated probability is \[F\left( {\frac{k}{N}} \right) = \frac{1}{N}\]. However, notice that as \[N \to \infty \], we have \[F\left( {\frac{k}{N}} \right) \to 0\]. Is there a nicer form that we can use to represent the probability that does not converge to zero? Here comes the key piece of innovation: suppose that we define another function that describes the probability density of the interval such that \[f\left( {\frac{k}{N}} \right)\frac{1}{N} = F\left( {\frac{k}{N}} \right)\] where \[\frac{1}{N}\] is the length of the interval. In the case of the uniform distribution, we see that as \[N \to \infty \], \[f\left( {\frac{k}{N}} \right) = 1\]. Hence, by dealing with the probability density, as \[N \to \infty \] the density will not converge to zero.

Suppose in the general case that our associated probability \[F\left( {\frac{k}{N}} \right)\] is not constrained to be uniform. In this case, \[F\left( {\frac{k}{N}} \right)\] can be any positive value such that the probabilities sum to 1, given by \[\sum\limits\_{n = 1}^N {F\left( {\frac{n}{N}} \right)} = 1\]. Now substitute in our expression for the density as \[\sum\limits\_{n = 1}^N {f\left( {\frac{n}{N}} \right)\frac{1}{N}} = 1\]. Taking the limit, we see that \[\mathop {\lim }\limits\_{N \to \infty } \sum\limits\_{n = 1}^N {f\left( {\frac{n}{N}} \right)\frac{1}{N}} = \int\limits\_0^1 {f\left( x \right)dx} = 1\]. Here we have substituted the limiting sum with a Riemann integral.

Therefore, an effective way to represent an arbitrary probability distribution is using the probability density which describes the probability per unit length. An intuitive way to think about the integral \[\int\limits\_0^1 {f\left( x \right)dx} \], is as follows: 1) this integral represents a countably infinite sum of primitive measures \[dx\] that have been scaled by \[f\left( x \right)\]. The interval from 0 to 1 is completely described by all the \[dx\] in the sum. 2) the function \[f\left( x \right)\] can be interpreted as the density of some quantity such as probability. The product \[f\left( x \right)dx\] computes the probability over an infinitesimal interval.

Returning to the Fourier transform, since we need all real frequencies \[\omega \] to describe an infinite time aperiodic signal, we need to consider the signal synthesis equation \[f\left( t \right) = \int\limits\_{ - \infty }^\infty {\hat f\left( \omega \right){e^{i2\pi \omega t}}d\omega } \] as being an integral over the domain of \[\omega \], where the primitive measure is an infinitesimal frequency interval of length \[d\omega \].

Now that we have seen the Fourier series for signals that are periodic, and the Fourier transform for signals that are aperiodic, what about signals that are discretely sampled? One way of modeling the process of sampling is to multiply a signal by an impulse train. An impulse train is defined as \[{\Psi \_T}\left( t \right) = T\sum\limits\_{n - \infty }^\infty {\delta \left( {t - nT} \right)} \] where \[T\] is the sampling interval.

To be able to use the Fourier series, we need to have a periodic signal. We can force our signal to be periodic by simply repeating it. Let us assume that our signal has length \[P\] and hence, by repeating the signal, the repeated signal has period \[P\]. We will assume that the sampling interval is a multiple of the period such that\[P = NT\]. We can write the Fourier series coefficients as follows

\[\begin{array}{l}{c\_k} = \frac{1}{P}\int\limits\_{ - \varepsilon }^{P - \varepsilon } {{e^{ - i\frac{{2\pi }}{P}kt}}T\sum\limits\_{n - \infty }^\infty {\delta \left( {t - nT} \right)} f\left( t \right)dt} \\ = \frac{T}{P}\sum\limits\_{n = 0}^{N - 1} {{e^{ - i\frac{{2\pi }}{P}knT}}f\left( {nT} \right)} \end{array}\]

Since \[P = NT\], we can rewrite the coefficient equation as: \[{c\_k} = \frac{1}{N}\sum\limits\_{n = 0}^{N - 1} {{e^{ - i2\pi \frac{{kn}}{N}}}f\left( {nT} \right)} \].

If we define \[{\hat c\_k} = N{c\_k}\] such that \[{\hat c\_k} = \sum\limits\_{n = 0}^{N - 1} {{e^{ - i2\pi \frac{{kn}}{N}}}f\left( {nT} \right)} \], then we can write the equations for all coefficients as:

\[\left[ {\begin{array}{\*{20}{c}}{{{\hat c}\_0}}\\ \vdots \\{{{\hat c}\_{N - 1}}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{{e^{ - i2\pi \frac{{\left( 0 \right)\left( 0 \right)}}{N}}}}& \cdots &{{e^{ - i2\pi \frac{{\left( 0 \right)\left( {N - 1} \right)}}{N}}}}\\ \vdots & \ddots & \vdots \\{{e^{ - i2\pi \frac{{\left( {N - 1} \right)\left( 0 \right)}}{N}}}}& \cdots &{{e^{ - i2\pi \frac{{\left( {N - 1} \right)\left( {N - 1} \right)}}{N}}}}\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{f\left( {0T} \right)}\\ \vdots \\{f\left( {\left( {N - 1} \right)T} \right)}\end{array}} \right]\]

If we let \[\omega = {e^{ - i\frac{{2\pi }}{N}}}\], then we can write the system of equations as:

\[\left[ {\begin{array}{\*{20}{c}}{{{\hat c}\_0}}\\ \vdots \\{{{\hat c}\_{N - 1}}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{{\omega ^{\left( 0 \right)\left( 0 \right)}}}& \cdots &{{\omega ^{\left( 0 \right)\left( {N - 1} \right)}}}\\ \vdots & \ddots & \vdots \\{{\omega ^{\left( {N - 1} \right)\left( 0 \right)}}}& \cdots &{{\omega ^{\left( {N - 1} \right)\left( {N - 1} \right)}}}\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{f\left( {0T} \right)}\\ \vdots \\{f\left( {\left( {N - 1} \right)T} \right)}\end{array}} \right]\]

The matrix \[{\user1{F}\_N} = \left[ {\begin{array}{\*{20}{c}}{{\omega ^{\left( 0 \right)\left( 0 \right)}}}& \cdots &{{\omega ^{\left( 0 \right)\left( {N - 1} \right)}}}\\ \vdots & \ddots & \vdots \\{{\omega ^{\left( {N - 1} \right)\left( 0 \right)}}}& \cdots &{{\omega ^{\left( {N - 1} \right)\left( {N - 1} \right)}}}\end{array}} \right]\] is called the N-point DFT matrix and will be unitary if scaled by \[\frac{1}{{\sqrt N }}\], such that \[{\left( {\frac{1}{{\sqrt N }}{\user1{F}\_N}} \right)^{ - 1}} = \frac{1}{{\sqrt N }}\user1{F}\_N^H\]. We can interpret this matrix as performing a change of basis from “time” domain to “frequency” domain.

# Matrix Factorizations

In this section we will discuss some of the important matrix factorizations.

## CR

Suppose we have a 3 x 3 matrix \[\user1{A}\] defined as

\[\user1{A} = \left[ {\begin{array}{\*{20}{c}}1&4&5\\3&2&5\\2&1&3\end{array}} \right]\]

We can write \[\user1{A}\] as the product of two matrices \[\user1{C}\] and \[\user1{R}\] such that

\[\user1{A} = \user1{CR} = \left[ {\begin{array}{\*{20}{c}}1&4\\3&2\\2&1\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}1&0&1\\0&1&1\end{array}} \right]\]

Where \[\user1{C}\] is constructed to contain the linearly independent columns of \[\user1{A}\], and \[\user1{R}\] contains the coefficients for the linear combinations of these columns to produce \[\user1{A}\]. This decomposition provides two fundamental and important interpretations of matrix multiplication:

1. Each column of \[\user1{A}\] can be viewed as a linear combination of the columns of \[\user1{C}\] where the combination coefficients are given by the columns of \[\user1{R}\]
2. Each row of \[\user1{A}\] can be viewed as a linear combination of the rows of \[\user1{R}\] where the combination coefficients are given by the rows of \[\user1{C}\]

We can define the column rank and row rank of \[\user1{A}\] to be the number of linearly independent columns and rows respectively. From the decomposition \[\user1{A} = \user1{CR}\] we can show that the column rank must equal the row rank of any matrix \[\user1{A}\] as follows:

1. Suppose that \[\user1{A}\] has \[r\] linearly independent columns, then \[\user1{C}\] has \[r\] columns and by the definition of its construction they are independent
2. Every column of \[\user1{A}\] can be written as a combination of the \[r\] columns of \[\user1{C}\] since \[\user1{A} = \user1{CR}\]
3. The \[r\] rows of \[\user1{R}\] must be independent since they contain the \[r\] by \[r\] identity matrix (this is because the \[r\] columns of \[\user1{C}\] are also columns of \[\user1{A}\] by construction, and hence, there must be a corresponding \[r\] by \[r\] identity submatrix in \[\user1{R}\]. Since \[\user1{R}\] has \[r\] rows and contains a \[r\] by \[r\] identity matrix, each row in \[\user1{R}\] contains a column which has value 1 for that row and 0 for all other rows)
4. Every row of \[\user1{A}\] is a combination of the \[r\] rows of \[\user1{R}\] since \[\user1{A} = \user1{CR}\], therefore the row rank of \[\user1{A}\] must be less than or equal to the column rank of \[\user1{A}\]
5. Now repeat steps 1-4 with \[{\user1{A}^T}\], we get the result that the column rank of \[\user1{A}\] is less than or equal to the row rank of \[\user1{A}\]
6. By combining the results from step 4 and step 5, we see that the row rank must equal the column rank

## LU

What is Lower Triangular – Upper Triangular (LU) factorization? The LU factorization states that we can factor any matrix \[\user1{A}\] with proper row and/or column permutations as \[\user1{A} = \user1{LU}\] where \[\user1{L}\] is a lower triangular matrix and \[\user1{U}\] is an upper triangular matrix. To develop intuition for this factorization, recall that we can transform a matrix \[\user1{A}\] into an upper triangular matrix by applying elementary row operations that consist of: 1) permuting two rows or columns, and 2) adding a multiple of one row to another row. If the matrix requires permutations, we can factor all the permutations into a matrix \[\user1{P}\] for row permutations, and a matrix \[\user1{Q}\] for column permutations such that \[\user1{PAQ} = \user1{LU}\]. By using these permutation matrices, we can ensure that the “add a multiple of one row to another row” operation only needs to add a multiple of an upper row of \[\user1{A}\] to a lower row of \[\user1{A}\]. This ensures that the matrix required to represent this operation is lower triangular. Hence, we can write the transformation of \[\user1{A}\] into an upper triangular matrix as \[{\user1{E}\_M} \cdots {\user1{E}\_1}\user1{PAQ} = \user1{U}\] where \[{\user1{E}\_M} \cdots {\user1{E}\_1}\] is the product of lower triangular matrices each describing an “add a multiple of one row to another row” operation. Therefore, to show that \[\user1{PAQ} = \user1{LU}\], we must show that \[{\left( {{\user1{E}\_M} \cdots {\user1{E}\_1}} \right)^{ - 1}}\] exists and is lower triangular. Consider the operation \[{\user1{E}\_1}\], which adds a multiple of one row to another row. The inverse of this operation is simply to subtract the same multiple of one row from the other row. Therefore, the inverse of \[{\user1{E}\_1}\] exists and furthermore is lower triangular since to compute the inverse of \[{\user1{E}\_1}\], we simply take the off diagonal component of \[{\user1{E}\_1}\] (which is below the diagonal) and invert the sign of the element (i.e. change from add to subtract). Hence, the inverse \[{\left( {{\user1{E}\_M} \cdots {\user1{E}\_1}} \right)^{ - 1}}\] exists and is the product of lower triangular matrices. It can be shown that the product of two lower triangular matrices is also lower triangular by examining the elements of the resultant matrix and showing that the elements above the diagonal must all be 0. Therefore, the matrix \[\user1{L} = {\left( {{\user1{E}\_M} \cdots {\user1{E}\_1}} \right)^{ - 1}}\] exists and is lower triangular. Hence, we have the factorization \[\user1{PAQ} = \user1{LU}\]. In the special case that \[\user1{A}\] already has proper row and/or column permutations, then \[\user1{P} = \user1{Q = I}\] and \[\user1{A} = \user1{LU}\].

How many operations are required to transform an \[n\] by \[n\] matrix \[\user1{A}\] into an upper triangular matrix \[\user1{U}\]? In the worst case, for each of the rows, we must perform a multiplication and a subtraction. Therefore, in the first step, we multiply the first row and add a copy of it to all other rows. This results in approximately \[{n^2}\] steps. As we repeat this process, we see that the number of operations is roughly \[{n^2} + {\left( {n - 1} \right)^2} + \cdots + {1^2} \approx \frac{{{n^3}}}{3}\].

Suppose we are trying to solve the system \[\user1{Ax} = \user1{b}\]. One way to approach this problem is to factor the matrix as \[\user1{A} = \user1{LU}\], where \[\user1{L}\] is a lower triangular matrix, and \[\user1{U}\] is an upper triangular matrix. The basic idea behind this factorization is the same as Gaussian elimination. We apply a series of lower triangular elementary operations to the matrix \[\user1{A}\] to form the upper triangular matrix \[\user1{U}\]. Upon determining the LU factorization, we can solve the system \[\user1{Ax} = \user1{LUx} = \user1{b}\] by first solving the lower triangular system \[\user1{Ly} = \user1{b}\] and then solving the upper triangular system \[\user1{Ux} = \user1{y}\].

## QR

Another important factorization is the QR factorization which states that any matrix \[\user1{A}\] can be decomposed as \[\user1{A} = \user1{QR}\] where \[\user1{Q}\] is an orthogonal matrix and \[\user1{R}\] is an upper triangular matrix. The Gram-Schmidt process is a way of computing the QR factorization, and proceeds by iteratively computing the component for the current vector that is orthogonal to each of the previously processed vectors. Using this process, an orthonormal basis can be built, and the resulting steps can be captured in the QR factorization.

A matrix with orthonormal columns is represented as \[{\user1{Q}^T}\user1{Q} = \user1{I}\] . If \[\user1{Q}\] is square, then \[{\user1{Q}^T}\user1{Q} = \user1{I}\] implies that \[{\user1{Q}^{ - 1}} = {\user1{Q}^T}\]. Suppose that \[\user1{Q}\] has orthonormal columns. If we want to project a vector onto its column space, then we can use the projection formula: \[\user1{P = Q}{\left( {{\user1{Q}^T}\user1{Q}} \right)^{ - 1}}{\user1{Q}^T} = \user1{Q}{\user1{Q}^T}\]. If \[\user1{Q}\] is square then \[\user1{Q}{\user1{Q}^T} = \user1{I}\]. For least squares, the normal equations for an orthonormal matrix reduces down to \[\user1{\hat x} = {\left( {{\user1{Q}^T}\user1{Q}} \right)^{ - 1}}{\user1{Q}^T}\user1{b = }{\user1{Q}^T}\user1{b}\].

Given an \[m\,\, \times \,\,n\] matrix \[\user1{A}\] with rank \[r\], how do we find the factorization \[\user1{A} = \user1{QR}\]? One method is to use Gram-Schmidt. The goal of Gram-Schmidt is to produce an orthonormal basis. The algorithm works as follows. Note that for simplicity we are assuming that the first \[r\] columns of \[\user1{A}\] are independent. We denote the column of \[\user1{A}\] as \[\user1{A} = \left[ {\begin{array}{\*{20}{c}}{{\user1{a}\_1}}& \cdots &{{\user1{a}\_n}}\end{array}} \right]\].

1. Pick the first column \[{\user1{a}\_1}\] and compute the normalized basis vector \[{\user1{q}\_1} = \frac{{{\user1{a}\_1}}}{{\left\| {{\user1{a}\_1}} \right\|}}\].
2. Select the next column \[{\user1{a}\_2}\] and compute the projection onto \[{\user1{q}\_1}\], written as \[{\user1{p}\_2} = {\user1{q}\_1}\user1{q}\_1^T{\user1{a}\_2}\]. Subtract the projection from \[{\user1{a}\_2}\] and normalize to get the next orthogonal basis vector \[{\user1{q}\_2} = \frac{{{\user1{a}\_2} - {\user1{p}\_2}}}{{\left\| {{\user1{a}\_2} - {\user1{p}\_2}} \right\|}}\].
3. Repeat this process for all \[r\] independent columns to establish an orthonormal basis \[\left\{ {{\user1{q}\_1}, \ldots ,{\user1{q}\_r}} \right\}\] for the column space of \[\user1{A}\].

Therefore, for a matrix with rank \[r\], the result of the Gram-Schmidt process will be an orthonormal basis for the column space \[\left[ {\begin{array}{\*{20}{c}}{{\user1{q}\_1}}& \cdots &{{\user1{q}\_r}}\end{array}} \right]\]. Since each column in \[\user1{A}\] is trivially in the column space. We can write each column as a linear combination \[{\user1{a}\_j} = {c\_1}{\user1{q}\_1} + \cdots + {c\_r}{\user1{q}\_r}\]. Using orthonormality, we can determine each of the coefficients as \[{c\_k} = \user1{q}\_k^T{\user1{a}\_j}\]. Substituting these coefficients back into the combination we get



\[\begin{array}{l}{\user1{a}\_j} = {c\_1}{\user1{q}\_1} + \cdots + {c\_r}{\user1{q}\_r} = {\user1{q}\_1}\user1{q}\_1^T{\user1{a}\_j} + \cdots + {\user1{q}\_r}\user1{q}\_r^T{\user1{a}\_j}\\ = \left[ {\begin{array}{\*{20}{c}}{{\user1{q}\_1}}& \cdots &{{\user1{q}\_r}}\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{\user1{q}\_1^T{\user1{a}\_j}}\\ \vdots \\{\user1{q}\_r^T{\user1{a}\_j}}\end{array}} \right]\end{array}\]

Therefore, we can represent the complete matrix as

\[\user1{A = }\left[ {\begin{array}{\*{20}{c}}{{\user1{a}\_1}}& \cdots &{{\user1{a}\_n}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{{\user1{q}\_1}}& \cdots &{{\user1{q}\_r}}\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{\user1{q}\_1^T{\user1{a}\_1}}& \cdots &{\user1{q}\_1^T{\user1{a}\_n}}\\ \vdots & \ddots & \vdots \\{\user1{q}\_r^T{\user1{a}\_1}}& \cdots &{\user1{q}\_r^T{\user1{a}\_n}}\end{array}} \right]\]

As a result of using Gram-Schmidt to construct the orthonormal basis, we have \[\user1{q}\_k^T{\user1{a}\_j} = 0\] when \[k > j\]. Therefore, we can simplify the second matrix in the product as

\[\left[ {\begin{array}{\*{20}{c}}{\user1{q}\_1^T{\user1{a}\_1}}& \cdots &{\user1{q}\_1^T{\user1{a}\_n}}\\ \vdots & \ddots & \vdots \\{\user1{q}\_r^T{\user1{a}\_1}}& \cdots &{\user1{q}\_r^T{\user1{a}\_n}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{\user1{q}\_1^T{\user1{a}\_1}}& \cdots &{\user1{q}\_1^T{\user1{a}\_n}}\\ \vdots & \ddots & \vdots \\0& \cdots &{\user1{q}\_r^T{\user1{a}\_n}}\end{array}} \right]\]

And this matrix is upper triangular. Note that this matrix will only be invertible if \[\user1{A}\] has full column rank and \[n = r\]. Hence, setting \[\user1{Q} = \left[ {\begin{array}{\*{20}{c}}{{\user1{q}\_1}}& \cdots &{{\user1{q}\_r}}\end{array}} \right]\], and \[\user1{R} = \left[ {\begin{array}{\*{20}{c}}{\user1{q}\_1^T{\user1{a}\_1}}& \cdots &{\user1{q}\_1^T{\user1{a}\_n}}\\ \vdots & \ddots & \vdots \\0& \cdots &{\user1{q}\_r^T{\user1{a}\_n}}\end{array}} \right]\] we see that any matrix can be factored as \[\user1{A} = \user1{QR}\] where \[\user1{Q}\] is an orthogonal matrix of size \[m\,\, \times \,\,r\] and \[\user1{R}\] is an upper triangular matrix or size \[r\,\, \times \,\,n\].

Some of the applications of QR factorization include:

* \*\*Solving Least Squares:\*\* The least squares solution \[{\user1{A}^T}\user1{Ax} = {\user1{A}^T}\user1{b}\] can be written using QR factorization as \[{\left( {\user1{QR}} \right)^T}\user1{QRx} = {\left( {\user1{QR}} \right)^T}\user1{b} \Rightarrow {\user1{R}^T}\user1{Rx = }{\user1{R}^T}\user1{Qb}\]. If \[\user1{A}\] has full column rank, then \[\user1{R}\] will be invertible and we can simplify the least squares equation to \[\user1{Rx = Qb}\].
* \*\*Pseudoinverse:\*\* Closely connected to least squares, the pseudoinverse of a matrix \[\user1{A}\] can be simplified using QR factorization as \[{\left( {{\user1{A}^T}\user1{A}} \right)^{ - 1}}{\user1{A}^T} = {\left( {{{\left( {\user1{QR}} \right)}^T}\user1{QR}} \right)^{ - 1}}{\user1{R}^T}{\user1{Q}^T}\user1{ = }{\left( {{\user1{R}^T}\user1{R}} \right)^{ - 1}}{\user1{R}^T}{\user1{Q}^T}\]. If \[\user1{A}\] has full column rank, then \[\user1{R}\] will be invertible and we can simplify this further to \[{\left( {{\user1{A}^T}\user1{A}} \right)^{ - 1}}{\user1{A}^T} = {\user1{R}^{ - 1}}{\user1{Q}^T}\].
* \*\*Projection onto the Column Space:\*\* Projection onto the column space of a matrix \[\user1{A}\] can be calculated using \[\user1{P = A}{\left( {{\user1{A}^T}\user1{A}} \right)^{ - 1}}{\user1{A}^T} = \user1{QR}{\left( {{{\left( {\user1{QR}} \right)}^T}\user1{QR}} \right)^{ - 1}}{\user1{R}^T}{\user1{Q}^T}\]. If \[\user1{A}\] has full column rank, then \[\user1{R}\] will be invertible and we can simplify this further to \[\user1{P = Q}{\user1{Q}^T}\].

## SVD

The singular value decomposition tells us that any \[m\] by \[n\] matrix \[\user1{A}\] with rank \[r\] can be factored as \[\user1{A = U}\Sigma {\user1{V}^T}\] where \[\user1{U}\] and \[\user1{V}\] are orthogonal matrices and \[\Sigma \] is a diagonal matrix.

By definition, a rank \[r\] matrix has a column space of dimension \[r\] which means that we need \[r\] vectors to describe a basis for the column space. Since the column rank equals the row rank, it follows that the row space also has dimension \[r\]. Since the dimensionality of these subspaces is the same, it makes intuitive sense that there should be a mapping between vectors in these subspaces. The process of finding such a mapping is the goal of the SVD.

The fundamental question that is answered by the SVD is the following: can we find an orthonormal basis for the row space \[{\user1{v}\_1}, \ldots ,{\user1{v}\_r}\], such that mapping these vectors to the column space \[\user1{A}\left[ {\begin{array}{\*{20}{c}}{{\user1{v}\_1}}& \cdots &{{\user1{v}\_r}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{{\sigma \_1}{\user1{u}\_1}}& \cdots &{{\sigma \_r}{\user1{u}\_r}}\end{array}} \right]\] produces an orthonormal basis for the column space \[{\user1{u}\_1}, \ldots ,{\user1{u}\_r}\].

Assuming that this factorization exists, then using matrices, we can write the mapping described above as

\[\user1{A}\left[ {\begin{array}{\*{20}{c}}{{\user1{v}\_1}}& \cdots &{{\user1{v}\_r}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{{\sigma \_1}{\user1{u}\_1}}& \cdots &{{\sigma \_r}{\user1{u}\_r}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{{\user1{u}\_1}}& \cdots &{{\user1{u}\_r}}\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{{\sigma \_1}}&{}&{}\\{}& \ddots &{}\\{}&{}&{{\sigma \_r}}\end{array}} \right]\]

We can go further than this by including the left and right nullspace into this equation. Given that the rank is \[r\], the dimension of the nullspace will be \[n - r\] which means that we can describe a basis for the nullspace using \[n - r\] vectors. We can easily find orthonormal vectors which are orthogonal to \[{\user1{v}\_1}, \ldots ,{\user1{v}\_r}\] by using a process such as Gram-Schmidt. Let us call these orthonormal basis vectors for the nullspace \[{\user1{v}\_{r + 1}}, \ldots ,{\user1{v}\_n}\]. Then we can write

\[\user1{A}\left[ {\begin{array}{\*{20}{c}}{{\user1{v}\_1}}& \cdots &{{\user1{v}\_r}}& \cdots &{{\user1{v}\_n}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{{\user1{u}\_1}}& \cdots &{{\user1{u}\_r}}\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{{\sigma \_1}}&{}&{}&0& \cdots &0\\{}& \ddots &{}&0& \cdots &0\\{}&{}&{{\sigma \_r}}&0& \cdots &0\end{array}} \right]\]

Likewise, we can add vectors for the left nullspace. Given that the rank is \[r\], the dimension of the left nullspace will be \[m - r\] . In a similar process to that described above, we can find an orthonormal basis for the left nullspace \[{\user1{u}\_{r + 1}}, \ldots ,{\user1{u}\_m}\]. Then we can write

\[\user1{A}\left[ {\begin{array}{\*{20}{c}}{{\user1{v}\_1}}& \cdots &{{\user1{v}\_r}}& \cdots &{{\user1{v}\_n}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}{{\user1{u}\_1}}& \cdots &{{\user1{u}\_r}}& \cdots &{{\user1{u}\_m}}\end{array}} \right]\left[ {\begin{array}{\*{20}{c}}{{\sigma \_1}}&{}&{}&0& \cdots &0\\{}& \ddots &{}&0& \cdots &0\\{}&{}&{{\sigma \_r}}&0& \cdots &0\\0&0&0&0& \cdots &0\\ \vdots & \vdots & \vdots & \vdots & \ddots &0\\0&0&0&0&0&0\end{array}} \right]\]

We can now define:

\[\user1{V} = \left[ {\begin{array}{\*{20}{c}}{{\user1{v}\_1}}& \cdots &{{\user1{v}\_r}}& \cdots &{{\user1{v}\_n}}\end{array}} \right]\]

\[\user1{U} = \left[ {\begin{array}{\*{20}{c}}{{\user1{u}\_1}}& \cdots &{{\user1{u}\_r}}& \cdots &{{\user1{u}\_m}}\end{array}} \right]\]

\[\Sigma = \left[ {\begin{array}{\*{20}{c}}{{\sigma \_1}}&{}&{}&0& \cdots &0\\{}& \ddots &{}&0& \cdots &0\\{}&{}&{{\sigma \_r}}&0& \cdots &0\\0&0&0&0& \cdots &0\\ \vdots & \vdots & \vdots & \vdots & \ddots &0\\0&0&0&0&0&0\end{array}} \right]\]

Then our expression becomes \[\user1{AV = U}\Sigma \]. Given that \[\user1{V}\] is composed of orthonormal columns we know that \[{\user1{V}^T}\user1{V} = \user1{I}\]. Multiplying both sides of this equation by \[{\user1{V}^T}\] gives

\[\begin{array}{l}{\user1{V}^T}\user1{V}{\user1{V}^T} = \user1{I}{\user1{V}^T} = {\user1{V}^T}\\ \Rightarrow {\user1{V}^T}\left( {\user1{V}{\user1{V}^T}} \right) = {\user1{V}^T}\\ \Rightarrow \user1{V}{\user1{V}^T} = \user1{I}\end{array}\]

Therefore, we see that \[{\user1{V}^T} = {\user1{V}^{ - 1}}\]. Hence, we can write the above factorization as \[\user1{A = U}\Sigma {\user1{V}^T}\]. Before diving into how to find these matrices, let us first describe what each of them represent:

* \[\user1{A}\] - This is the matrix we are trying to factor. \[\user1{A}\] can be any \[m\] by \[n\] matrix with rank \[r\].
* \[\user1{V}\] - This is an orthonormal matrix of size \[n\] by \[n\]. The first \[r\] columns represent an orthonormal basis for the row space of \[\user1{A}\]. To see why, consider rearranging the factorization as \[\user1{A = U}\Sigma {\user1{V}^T} = \left( {\user1{U}\Sigma } \right){\user1{V}^T}\]. From our interpretations of matrix multiplication, we can interpret the rows of the product \[\left( {\user1{U}\Sigma } \right){\user1{V}^T}\] as being linear combinations of the first \[r\]rows of \[{\user1{V}^T}\]. Note that only the first \[r\] rows of \[{\user1{V}^T}\] contribute to the final product since the last \[n - r\] rows of \[\Sigma \] are filled with zeros. Therefore, the rows of \[\user1{A}\] are linear combinations of the first \[r\]rows of \[{\user1{V}^T}\], or equivalently, the first \[r\] columns of \[\user1{V}\]. Hence, the first \[r\] columns represent an orthonormal basis for the column space of \[\user1{A}\]. The last \[n - r\] columns of \[\user1{V}\] represent an orthonormal basis for the nullspace of \[\user1{A}\]. To see why, we notice that the last \[n - r\] columns of \[\Sigma \] are filled with zeros, which implies that \[\user1{A}\left[ {\begin{array}{\*{20}{c}}{{\user1{v}\_{r + 1}}}& \cdots &{{\user1{v}\_n}}\end{array}} \right] = \left[ {\begin{array}{\*{20}{c}}\user1{0}& \cdots &\user1{0}\end{array}} \right]\]. Therefore, we see that \[{\user1{v}\_{r + 1}}, \ldots ,{\user1{v}\_n}\] are all in the nullspace of \[\user1{A}\], and since they are orthonormal they form a basis for the nullspace.
* \[\user1{U}\]- This is an orthonormal matrix of size \[m\] by \[m\]. The first \[r\] columns represent an orthonormal basis for the column space of \[\user1{A}\]. The last \[m - r\] columns of \[\user1{U}\] represent an orthonormal basis for the left nullspace of \[\user1{A}\].
* \[\Sigma \] - This is an \[m\] by \[n\] diagonal matrix where the first \[r\] diagonal entries of the matrix are nonzero and called singular values and the rest of the entries are 0.

At this point, the question becomes: how do we find the SVD factorization for a matrix \[\user1{A = U}\Sigma {\user1{V}^T}\]? Well, we can start by assuming that the factorization exists. Then multiplying on the left by \[{\user1{A}^T}\] we get:

\[{\user1{A}^T}\user1{A = }{\left( {\user1{U}\Sigma {\user1{V}^T}} \right)^T}\user1{U}\Sigma {\user1{V}^T} = \user1{V}{\Sigma ^T}\Sigma {\user1{V}^T}\]

Therefore, we see that \[\user1{V}\] must be the eigenvector matrix for \[{\user1{A}^T}\user1{A}\], and \[{\Sigma ^T}\Sigma \] must be the eigenvalue matrix of \[{\user1{A}^T}\user1{A}\]. Given that \[{\user1{A}^T}\user1{A}\] is a symmetric matrix, the spectral theorem tells us that there exists an orthonormal basis for its eigenvectors. Hence deriving \[\user1{V}\] from the eigenvectors of \[{\user1{A}^T}\user1{A}\] ensures that it is orthonormal.

To get \[\Sigma \], we notice that \[{\Sigma ^T}\Sigma \] is the product of two diagonal matrices which will also be diagonal. Therefore, the entries along the diagonal of \[\Sigma \] are simply the square roots of the corresponding diagonal entries of \[{\Sigma ^T}\Sigma \] which are the eigenvalues of \[{\user1{A}^T}\user1{A}\].

Finally, to get the matrix \[\user1{U}\] we can apply the same technique, but in the opposite order. Multiplying on the right by \[{\user1{A}^T}\] gives:

\[\user1{A}{\user1{A}^T}\user1{ = U}\Sigma {\user1{V}^T}{\left( {\user1{U}\Sigma {\user1{V}^T}} \right)^T} = \user1{U}{\Sigma ^T}\Sigma {\user1{U}^T}\]

Therefore, we see that \[\user1{U}\] must be the eigenvector matrix for\[\user1{A}{\user1{A}^T}\], and \[{\Sigma ^T}\Sigma \] must be the eigenvalue matrix of \[\user1{A}{\user1{A}^T}\]. Similar to above, as a product of the spectral theorem, we see that deriving \[\user1{U}\] from the eigenvectors of \[\user1{A}{\user1{A}^T}\] ensures that it is orthonormal.

Therefore, we have shown how to derive the SVD for a general matrix \[\user1{A}\]. The beauty of the SVD is that it selects the best basis (i.e. orthonormal) for the four subspaces associated with \[\user1{A}\], and describes a scaling mapping between the row and column space basis. The SVD enables us to describe the action of any matrix as three steps: 1) rotation, 2) scaling, 3) rotation.