CS 596: Linear Algebra Notes

16:198:596

As discussed in the introductory lecture, we tend to view data as large collections of vectors - each vector is a record, and each component of the vector corresponds to some feature, attribute, or measurement of that record. Hence, if we want to understand data and the shape of data, we need to have a good mathematical grounding in understanding collections of vectors. Similarly, models can be thought of as transformations from one data space to another, or similarly from one vector space to another. These can vary from the simple linear models of linear regression, to much more complex non-linear models of neural networks. But again, if we want to understand models and the analysis and properties thereof, we need to understand mappings or transformations between data space.

As a motivating example for some of the discussion to follow, suppose that we have a data matrix X that is $N \times D$ dimensional - N records, each of D features or attributes. We might very well want to understand the relationship between records or the *row space* of X, or between different features in the *column space* of X. Ultimately we'll connect the matrix $X^{\mathrm{T}}X$ back to important statistical properties such as the covariance between features in a population.

Vectors

The sort of classical view of vectors is as a D-component ordered list of real numbers (though we might extend this to consider discrete or boolean data, for instance taking **True/False** as 1/0). This can be thought of as a consequence of the *standard basis* (discussed shortly), but more abstractly a **Vector Space** V is a set of objects (vectors) such that

- For any $\underline{v}, \underline{u} \in V$, $\underline{v} + \underline{u} \in V$. That is, any two vectors can be added to produce a new vector in the space.
- For any $\underline{v} \in V$ and real number α, $\alpha V \in V$. That is, any vector can be scaled by a real number (though we might generalize this as well) to produce a new vector in the space.

It's relatively immediate that our usual interpretation of vectors satisfies this. Taking $V = \mathbb{R}^D$, we typically represent each vector as D-many ordered values, where addition is defined element by element, and scalar multiplication is defined similarly. But one of the values of this abstract definition is that we can apply it to many other things as well - we might think about adding or superimposing images, or scaling them up by some intensity, etc. Images do not immediately fit the usual view of vectors (though we could definitely take the pixel matrix data and stretch it out to form a very long vector in the more usual way). Similarly, we can actually define vector spaces of functions as well - addition and scaling is well defined over sets of functions.

But this provides a very flexible, general framework for understanding vectors and data as a collection of vectors. Note that a vector space V does not demand a specific representation of vectors in it (such as the $(x_1, x_2, \ldots, x_D)$ representation), but we will explore that more when we get to geometry and bases.

As noted, the key operations that make a vector space are addition and scalar multiplication. Another common operation we define is the notion of the *inner product* or *dot product*:

$$\underline{x}.\underline{y} = x_1 * y_1 + x_2 * y_2 + \ldots x_D * y_D = \sum_{i=1}^{D} x_i y_i. \tag{1}$$

The inner product has a number of geometric interpretations, but roughly speaking it starts to quantify the relationships (covariance) between the features of $\underline{x}$ and $\underline{y}$. The dot product of a vector with itself has a particular

interpretation geometrically:

$$\underline{x}.\underline{x} = \sum_{i=1}^{D} x_i^2 = ||\underline{x}||_2^2, \tag{2}$$

where $||\underline{x}||_2$ is the *2-norm* of the vector $\underline{x}$, or the Euclidean distance from the 'origin' to a point displaced from the origin by $\underline{x}$. This is typically thought of as the 'length' of the vector $\underline{x}$. Note, this gives us a couple of ways to consider the distance from $\underline{x}$ to $\underline{y}$. In particular,

$$||\underline{x} - \underline{y}||_2 = \sqrt{\sum_{i=1}^{D}(x_i - y_y)^2} = (\underline{x} - \underline{y}).(\underline{x} - \underline{y}). \tag{3}$$

There are a number of ways we might define the 'length' or the size of the vector $\underline{x}$. One of them is a flexible notion of the p-norm:

$$||\underline{x}||_p = \left(\sum_{i=1}^{D} |x_i|^p\right)^{1/p}, \tag{4}$$

where the Euclidean norm corresponds to taking $p = 2$. Another commonly used norm is taking $p = 1$, giving

$$||\underline{x}||_1 = \sum_{i=1}^{D} |x_i|, \tag{5}$$

commonly referred to as the *1-norm* or the Manhattan distance. Another, slightly less common norm is defined by taking $p \to \infty$, giving the *infinity norm* or

$$||\underline{x}||_\infty = \max_i |x_i|. \tag{6}$$

All of these give different notions of the 'size' of $\underline{x}$, quantifying how different it is from the zero vector $\underline{0}$.

One thing worth mentioning here is that this allows us (at least in a sort of Euclidean sense) to define a notion of *direction*. From a geometric perspective, vectors have two things - a length, and a direction (pointing potentially in some high dimensional space). We can define that direction in terms of a *unit vector*, a normalized vector pointing in the same direction. Given a vector $\underline{x}$, the unit vector (usually denoted $\hat{\underline{x}}$) is defined as

$$\hat{\underline{x}} = \frac{1}{||\underline{x}||_2}\underline{x}. \tag{7}$$

We can see from this that $||\hat{\underline{x}}||_2^2 = \hat{\underline{x}}.\hat{\underline{x}} = (1/||\underline{x}||_2^2)||\underline{x}||_2^2 = 1$, hence the unit vector is *normalized* to length 1. This is convenient, as stated, for expressing directions in some kind of data space.

We can now discuss more clearly the intuitions behind the dot product, $\underline{u}.\underline{v}$. In particular, if you consider $\underline{u}, \underline{v}$ as vectors pointing from the origin to some points in space, they define some angle θ between them. It can be shown geometrically that

$$\underline{u}.\underline{v} = ||\underline{u}||_2||\underline{v}||_2 \cos(\theta). \tag{8}$$

As such, the dot product gives a very easy means of computing the angle between two vectors. Additionally, this is important when thinking about *projection* - just because a vector points in a certain direction, it can frequently be decomposed into projections in different directions (for example, north west is a combination of projection in the north direction and in the west direction). Given a vector $\underline{v}$ and a unit vector or direction $\hat{\underline{u}}$, the 'projection of $\underline{v}$ in the direction of $\hat{\underline{u}}$' is given by

$$\text{proj}_{\hat{\underline{u}}}(\underline{v}) = \underline{v}.\hat{\underline{u}}. \tag{9}$$

This gives rise to the notion of orthogonal vectors - vectors that point in 90-degree angles to one another. Similar to the idea that in 3-dimensions, the vectors $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$ point at right angles to one another, $\underline{u}$ and $\underline{v}$

are orthogonal if $\underline{u}.\underline{v} = 0$, since $\cos(90\text{deg}) = 0$. A set of vectors $\underline{v}_1, \ldots, \underline{v}_N$ are *mutually orthogonal* if $\underline{v}_i.\underline{v}_j = 0$ for $i \neq j$. A set of vectors is mutually *orthonormal* if they are mutually orthogonal and $||\underline{v}_i||_2^2 = \underline{v}_i.\underline{v}_i = 1$.

To close this discussion of vectors and geometry of vectors, it's worth discussing the *Cauchy-Schwartz* inequality. In the notation we've established here, it says that

$$|\underline{u}.\underline{v}| \leq ||\underline{u}||_2||\underline{v}||_2. \tag{10}$$

This can be very easily seen as true utilizing the geometric view of the dot product above, since $|\cos| \leq 1$. But this can be extended to apply to general vector spaces as well. One of the values of it is that it can essentially decompose relationships between $\underline{u}$ and $\underline{v}$ to treat the two vectors separately. A good example of this is the *triangle inequality*, which says that for any vectors $\underline{x}, \underline{y}$ we have

$$||\underline{x} + \underline{y}||_2 \leq ||\underline{x}||_2 + ||\underline{y}||_2. \tag{11}$$

Where we see that the interactions between $\underline{x}$ and $\underline{y}$ in the sum have been decomposed to treat $\underline{x}$ and $\underline{y}$ separately. To see this, note that (as an application of the Cauchy-Schwartz inequality)

$$\begin{aligned} ||\underline{x} + \underline{y}||_2^2 &= (\underline{x} + \underline{y}).(\underline{x} + \underline{y}) \\ &= \underline{x}.\underline{x} + \underline{x}.\underline{y} + \underline{y}.\underline{x} + \underline{y}.\underline{y} \\ &= ||\underline{x}||_2^2 + 2\underline{x}.\underline{y} + ||\underline{y}||_2^2 \\ &\leq ||\underline{x}||_2^2 + 2|\underline{x}.\underline{y}| + ||\underline{y}||_2^2 \\ &\leq ||\underline{x}||_2^2 + 2||\underline{x}||_2||\underline{y}||_2 + ||\underline{y}||_2^2 \\ &= \left(||\underline{x}||_2 + ||\underline{y}||_2\right)^2 \end{aligned} \tag{12}$$

which, taking the square root of both sides, gives $||\underline{x} + \underline{y}||_2 \leq ||\underline{x}||_2 + ||\underline{y}||_2$.

A Brief Aside on Matrices

We typically think of a matrix A as an r-row by c-column (dimension (r, c), rows usually coming first) table of values. We can think of it as belonging to a space $A \in \mathbb{R}^{r \times c}$ - that is, while a matrix A is decidedly non-flat, we can think of it equivalently as a vector of length $r * c$. As such, we can think of matrix addition and scalar multiplication exactly as we do on vectors, operating element by element.

To discuss matrices, it is convenient to introduce the distinction between row vectors and column vector; a row vector is denoted in the usual way, $(x_1, \ldots, x_D)$, whereas a column vector's values are distributed vertically. In this way, a row vector can be thought of as a $1 \times D$ matrix, and a column vector a $D \times 1$ matrix. For a given matrix A, we might indicate the i-th row of A by the row vector $\underline{A}_i$, and the j-th entry of the i-th row as $A_{i,j}$.

Under this convention, we can denote the matrix product of an $r \times c$ dimensional matrix A and a c-entry column vector $\underline{v}$ as the r-dimensional column vector $A\underline{v}$, where the i-th entry is given as the dot product $\underline{A}_i.\underline{v}$. In this way, the resulting vector $A\underline{v}$ can be thought of either as the set of inner product results, or it can additionally be thought of as a linear combination of the columns of A.

Linear Transformations and Matrices

Typically, the first model we consider in machine learning is a linear model. That is, given features $x_1, \ldots, x_D$, we want to model some output value y as a linear combination of the input vectors:

$$f(x_1, \ldots, x_D) = w_1x_1 + w_2x_2 + \ldots + w_Dx_D + b \tag{13}$$

Note that we can express this more compactly as $f(\underline{x}) = \underline{w}.\underline{x} + b$, for some vector of weights $\underline{w}$ and some intercept b. We could even expand this to a higher dimensional (assuming the output we want to predict is multi-dimensional) $f(\underline{x}) = W\underline{x} + \underline{b}$, where W is some weight matrix (each row a vector of weights according to the 1-D model), and $\underline{b}$ a vector of intercepts.

We can generalize this somewhat to the concept of a **linear transform**. A linear transform is a map F between vector spaces (i.e., data spaces) where the following relations are maintained: for any α, β in the reals, and $\underline{u}, \underline{v}$ in the initial vector space, we have

$$F(\alpha\underline{v} + \beta\underline{u}) = \alpha F(\underline{v}) + \beta F(\underline{v}). \tag{14}$$

It is straightforward to show for instance that multiplication by a matrix is a linear transform. If $F(\underline{v}) = M\underline{v}$, then we have that

$$F(\alpha\underline{v} + \beta\underline{u}) = M(\alpha\underline{v} + \beta\underline{u}) = \alpha M\underline{v} + \beta M\underline{u} = \alpha F(\underline{v}) + \beta F(\underline{u}). \tag{15}$$

Subject to some discussion in the next section on basis sets, *any* linear transform can be expressed as multiplication by some matrix. Hence in order to understand linear transforms and linear models, it's worth spending some time unpacking and analyzing matrices and matrix multiplication.

A keen eyed observer will note, however, that the general linear model presented is in fact *not* a linear transform. Note that if $f(\underline{x}) = W.\underline{x} + \underline{b}$, then we have that

$$F(\alpha\underline{v} + \beta\underline{u}) = W(\alpha\underline{v} + \beta\underline{u}) + \underline{b} = \alpha(W\underline{v} + \underline{b}) - \alpha\underline{b} + \beta(W\underline{u} + \underline{b}) - \beta\underline{b} + \underline{b} = \alpha F(\underline{v}) + \beta F(\underline{u}) + (1 - \alpha - \beta)\underline{b}. \tag{16}$$

Hence, unless $\alpha + \beta = 1$, or $\underline{b} = 0$, the necessary relation will not hold. As such, the general linear model is not in fact a linear transform.

However we can recover it in the following way: consider augmenting the data vector $\underline{x}$ to include a 0th component, which is fixed to be 1:

$$\underline{x}' = (1, x_1, x_2, \ldots, x_D). \tag{17}$$

Additionally, consider taking the vector $\underline{b}$ and prepending it to the weight matrix W as a 0th column - call this augmented matrix W'. In this case, we would get that

$$W\underline{x} + \underline{b} = W'\underline{x}', \tag{18}$$

and we can see that the general linear model is in fact a linear transform applied to a higher dimensional vector space (of which our data space can be thought of as a lower-dimensional slice). In this way, the analysis of linear transforms carries over to general linear models. The properties of linear models are then tied directly to the properties of matrices and matrix multiplication.

As discussed previously, matrices are little more than malformed vectors and as a result form a vector space, where addition and scaling by a constant functions element-wise as you would expect. Additionally, we defined multiplication of a matrix by a vector, and we can extend this to multiplication of two matrices: the matrix product $C = AB$ is taken to be the matrix where $C_{i,j}$ is the inner product of the i-th row of A and the j-th column of B, or in other words the j-th column of C is the product of A and the j-th column of B. Note that in order for matrix multiplication to be well-defined, there must be some agreement between the dimensions of A and B: if A is of dimension (r, c), then B must be of dimension (c, d) for some number of columns d; that is, the number of columns of A must match the number of rows of B. The product will then be a matrix C of dimension (r, d). It is interesting to observe here that while matrix multiplication is associative, i.e., $(A * B) * C = A * (B * C)$, it is not generally the case that matrices commute, i.e., it is not generally the case that $A * B = B * A$.

A Note on Computation and Complexity: This has a couple of interesting implications for the computation of a matrix product. Consider two $n \times n$ matrices A, B. To compute the product in the naive way (taking all of the

relevant dot products), would require n^2 dot product operations, each a complexity of order $O(n)$, to give a total complexity of $O(n^3)$. A number of matrix multiplication algorithms exist to try to simplify this, and the exponent has gotten down quite low, but can't really theoretically be any lower than $O(n^2)$ *(why?)*. As such, multiplying large matrices is potentially an expensive operation. Additionally, for non-uniformly sized matrices, the associative property can be used to try to cut down on the total number of operations. *For various matrices A, B, C, when might it be beneficial to compute $A * B * C$ as $(A * B) * C$ and when might it be better to compute it as $A * (B * C)$?*

An additional matrix operation that is useful to define is the *transpose*: M^{T} is taken to be the matrix M with the rows and columns inverted. If M is of dimension (r, c), then M^{T} will be of dimension (c, r), where the first column of M^{T} corresponds to the first row of M. Note, this allows us to unify our discussion of row and column vectors somewhat, taking any row vector to be the transpose of some column vector. In this way, we will (unless otherwise specified) take vectors to be referring to column vectors, and use the transpose operator to refer to the corresponding row vector. Additionally, we can now indicate dot products with the more standardized notation, $\underline{u}^{\mathrm{T}}\underline{v}$.

An important property of the transpose operation, connecting it to matrix multiplication, is that $(AB)^{\mathrm{T}} = B^{\mathrm{T}}A^{\mathrm{T}}$ - that is, transposition reverses the order of multiplication while taking transposes. Can you show that this is true?

A Brief Aside on Rotations

A good example of a linear transformation is that of *rotation*. Any computation on points in a vector space, rotated, will yield the same result as that computation on the original points rotated. One way of understanding rotation is as a rigid linear transformation that preserves distance (note, a consequence of it being a linear transform is no translation!). This can be modeled in the general case as multiplication by a matrix where the columns of the matrix form an orthonormal set. Let M be a matrix where the columns $\{\underline{M}_i\}$ (note we are referring to the columns here rather than the rows) represent an orthonormal set. This distance preservation property can be thought of as the claim that for any $\underline{x}, \underline{y}$, we have that

$$||M\underline{x} - M\underline{y}||_2 = ||\underline{x} - \underline{y}||_2, \tag{19}$$

or equivalently that for any vector $\underline{v}$, $\underline{v}$ and $M\underline{v}$ have the same norm. We can verify this in the following way, utilizing the orthonormality of the rows of M and the definition of the dot product and the properties of the transpose. Note that

$$||M\underline{v}||_2^2 = [M\underline{v}]^{\mathrm{T}}[M\underline{v}] = \underline{v}^{\mathrm{T}}M^{\mathrm{T}}M\underline{v}. \tag{20}$$

However, because the columns of M form an orthonormal set, any column of M multiplied by the rows of M^{T} will be 0, except for the row corresponding with that column, which will give an inner product of 1. As a result, $M^{\mathrm{T}}M$ is the matrix of all 0s, except for the diagonal which is all 1s - the so called Identity Matrix. Hence we have

$$||M\underline{v}||_2^2 = \underline{v}^{\mathrm{T}}M^{\mathrm{T}}M\underline{v} = \underline{v}^{\mathrm{T}}I\underline{v} = \underline{v}^{\mathrm{T}}\underline{v} = ||\underline{v}||_2^2, \tag{21}$$

and we see that the norm is preserved.

Matrix Inverse and Its Properties

If any linear transform is a matrix multiplication, the *inverse* transform (should it exist) should also be a matrix multiplication. That is, if $\underline{y} = A\underline{x}$ and the map is invertible, we should also get that $\underline{x} = B\underline{y}$ for some matrix B. This is the *multiplicative inverse* of A, usually denoted $B = A^{-1}$. For reasons that will become more clear, only square matrices can be invertible, and their inverses will be square matrices of the same dimension. The inverse matrix satisfies the relation

$$AA^{-1} = I, \tag{22}$$

where I is the identity matrix, a square matrix of all zeroes except for 1s on the diagonal. Note that the identity matrix functions much like the number 1 multiplicatively, in that for every square matrix A, we have that $AI = IA = A$.

It's interesting to note here that while matrix multiplication is typically non-commutative, every invertible matrix commutes with its inverse. To see this suppose that A has a 'right inverse' B such that $AB = I$ and a 'left inverse' C such that $CA = I$. In this case, we have that

$$C = CI = C(AB) = (CA)B = IB = I. \tag{23}$$

And we establish the matrices are in fact identical.

It's worth repeating, not all matrices are invertible however. For a 2×2 matrix

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \tag{24}$$

the inverse is given explicitly by

$$M^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}, \tag{25}$$

which exists as long as $ad - bc \neq 0$ - a condition that we will unpack more in a moment. We can multiply this out to verify it directly, and we can derive it by considering a generic matrix A and solving for the individual terms to make $AM = I$. Other inverses can be computed similarly, though we will discuss that more later.

For the moment, we consider the problem of when a matrix is invertible in the following way: recalling the relation between matrix multiplication and linear transformations, a linear transformation $A\underline{x}$ would *fail* to be invertible if there were two distinct $\underline{x}, \underline{x}'$ such that $A\underline{x} = A\underline{x}'$. If this held, it would be impossible to identify a unique pre-image of $\underline{y} = A\underline{x}$.

Manipulating this, we see that it would imply that $A(\underline{x} - \underline{x}') = 0$, or in other words that there is some non-zero vector $\underline{v}$ such that $A\underline{v} = 0$. This gives us a necessary and in fact sufficient condition for invertibility of a matrix:

> A matrix A is invertible if and only if $A\underline{v} \neq 0$ for all non-zero vectors $\underline{v}$.

As an end note, it's worth considering the properties of the matrix inverse and how it relates to the other operations we've discussed. In particular, you can show the following:

- $(AB)^{-1} = B^{-1}A^{-1}$
- $(A^{\mathrm{T}})^{-1} = (A^{-1})^{\mathrm{T}}$.

And it's worth mentioning as well that much as there is no obvious connection between $1/(x + y)$ and $1/x, 1/y$, in general there is no nice expression for $(A + B)^{-1}$ in terms of A^{-1} and B^{-1}, and it is certainly **not** generally true that $(A + B)^{-1} = A^{-1} + B^{-1}$.

A Brief Aside on Inverting Matrices

In general, for an invertible matrix A, the inverse matrix A^{-1} can be difficult to compute (complexity on the order of $O(n^3)$, etc). On occasion this can be simplified somewhat, such as in the following lemma: suppose that $M = A + \underline{u}\underline{v}^T$, where A is a given matrix whose inverse is known, and $\underline{u}, \underline{v}$ are vectors. Note that multiplying $\underline{u}$ and $\underline{v}$ in this way will yield a square matrix, the same dimension as A.

We would like to compute the inverse of M, without too much more additional work than what has been done. The Matrix Inversion Lemma states the following: If $1 + \underline{v}^{\mathrm{T}} A^{-1} \underline{u} \neq 0$, the inverse of M exists and is given by

$$M^{-1} = A^{-1} - \left(\frac{1}{1 + \underline{v}^{\mathrm{T}} A^{-1} \underline{u}} \right) A^{-1} \underline{u}\underline{v}^{\mathrm{T}} A^{-1}. \tag{26}$$

To see this, we can verify it directly in the following way:

$$\begin{aligned} MM^{-1} &= (A + \underline{u}\underline{v}^{T}) \left(A^{-1} - \left(\frac{1}{1 + \underline{v}^{\mathrm{T}} A^{-1} \underline{u}} \right) A^{-1} \underline{u}\underline{v}^{\mathrm{T}} A^{-1} \right) \\ &= (AA^{-1} + \underline{u}\underline{v}^{T} A^{-1}) \left(I - \left(\frac{1}{1 + \underline{v}^{\mathrm{T}} A^{-1} \underline{u}} \right) \underline{u}\underline{v}^{\mathrm{T}} A^{-1} \right) \\ &= (I + \underline{u}\underline{v}^{T} A^{-1}) \left(I - \left(\frac{1}{1 + \underline{v}^{\mathrm{T}} A^{-1} \underline{u}} \right) \underline{u}\underline{v}^{\mathrm{T}} A^{-1} \right) \\ &= I - \left(\frac{1}{1 + \underline{v}^{\mathrm{T}} A^{-1} \underline{u}} \right) \underline{u}\underline{v}^{\mathrm{T}} A^{-1} + \underline{u}\underline{v}^{T} A^{-1} - \left(\frac{1}{1 + \underline{v}^{\mathrm{T}} A^{-1} \underline{u}} \right) \underline{u}\underline{v}^{T} A^{-1} \underline{u}\underline{v}^{\mathrm{T}} A^{-1} \end{aligned} \tag{27}$$

Let α be the scalar value $\underline{v}^{\mathrm{T}} A^{-1} \underline{u}$. The above is rewritten as

$$\begin{aligned} MM^{-1} &= I - \left(\frac{1}{1 + \alpha} \right) \underline{u}\underline{v}^{\mathrm{T}} A^{-1} + \underline{u}\underline{v}^{T} A^{-1} - \left(\frac{1}{1 + \alpha} \right) \underline{u}\alpha\underline{v}^{\mathrm{T}} A^{-1} \\ &= I - \left(\frac{1}{1 + \alpha} \right) \underline{u}\underline{v}^{\mathrm{T}} A^{-1} + \underline{u}\underline{v}^{T} A^{-1} - \left(\frac{\alpha}{1 + \alpha} \right) \underline{u}\underline{v}^{\mathrm{T}} A^{-1} \\ &= I + \left[- \left(\frac{1}{1 + \alpha} \right) + 1 - \left(\frac{\alpha}{1 + \alpha} \right) \right] \underline{u}\underline{v}^{\mathrm{T}} A^{-1} \\ &= I + [0]\, \underline{u}\underline{v}^{\mathrm{T}} A^{-1} \\ &= I \end{aligned} \tag{28}$$

Hence the formula for the inverse is verified.

This gets generalized into the following formula:

$$(A + UCV)^{-1} = A^{-1} - A^{-1} U (C^{-1} + VA^{-1} U)^{-1} VA^{-1}, \tag{29}$$

But the utility of this depends on having the inverses of A and C, as well as being able to easily invert $C^{-1} + VA^{-1}U$.

Interesting Classes of Matrices

There are a couple of classes or forms of matrix that recur in practice and are worth being familiar with.

Diagonal Matrix

A diagonal matrix is a matrix D that is everywhere 0 except for the entries along the diagonal starting from the upper left. The classical example of this is of course the identity matrix. Note that multiplying a vector $\underline{x}$ by a square diagonal matrix D, the result is a vector where the i-th component is given by $D_{i,i} * x_i$, i.e., all elements of $\underline{x}$ are individually scaled by the corresponding element of the diagonal in D.

As a result, diagonal matrices are exceptionally easy to invert - as long as the entries of the diagonal are non-zero, the inverse D^{-1} is given by the diagonal matrix where $\left[D^{-1}\right]_{i,i} = 1/D_{i,i}$.

Symmetric Matrices

A symmetric matrix A is one for which $A^{\mathrm{T}} = A$, i.e., the corresponding rows and columns are identical. As a result, from a purely dimensional standpoint, symmetric matrices must be square.

Symmetric matrices hold a special place, in the following way. For a general linear form, such as $3x_1 + 2x_2$, we could think of this as the dot product of a weight vector $\underline{w}$ and the vector $\underline{x} = (x_1, x_2)$. Similarly, we can think of a *quadratic* form as associated with a symmetric weight matrix. Consider for example the quadratic form $x_1^2 + 4x_1x_2 - x_2^2$ - note that every term here is of degree 2. Defining the matrix A such that

$$A = \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix}, \tag{30}$$

we can express this quadratic form more compactly as $\underline{x}^{\mathrm{T}} A \underline{x}$. As such, a general D-variable quadratic polynomial might be written as

$$\underline{x}^{\mathrm{T}} A \underline{x} + \underline{w}^{\mathrm{T}} \underline{x} + b, \tag{31}$$

for some symmetric matrix A and weight vector $\underline{w}$ and scalar intercept b.

Triangular Matrices and Nilpotent Matrices

An *upper* triangular matrix M is one where the entries are zero below the diagonal, and the only non-zero entries are on or above the diagonal. A *strictly* upper triangular matrix is one in which the diagonal is zero as well. Triangular matrices frequently arise in conjunction with analyzing systems of linear equations, discussed in a section to follow. Lower triangular matrices are defined similarly.

It's worth observing here that a strictly triangular matrix is an example of a **nilpotent** matrix, i.e., a matrix such that $A^k = A * \ldots * A = 0$ for some k. Note that nilpotent matrices cannot be invertible *(why?)*. Additionally, note that if A is a square, strictly triangular matrix of dimension $D \times D$, it must be that $A^D = 0$. *Why?*

Permutation Matrices

A permutation matrix P is a square matrix where, if the dimension is D, there are D-many 1s, and the other entries are all zero - the ones being placed such that every row and column has a single 1. *There are $D!$ permutation matrices. Why?* Note that the effect of multiplying such a matrix by a vector $\underline{x}$ is to reorder the entries of $\underline{x}$. Permutation matrices are necessarily invertible. *Why? Give two reasons.*

The Geometry of Vector Spaces

Given two vectors $\underline{u}, \underline{v}$, one of the geometric relationships we typically discuss is the midpoint or average, the point exactly half way between both vectors. This is typically given as $\underline{x} = (1/2)(\underline{u} + \underline{v})$, though I will write it more deliberately as $\underline{x} = (1/2)\underline{u} + (1/2)\underline{v}$, a weighted linear combination of $\underline{u}$ and $\underline{v}$. Note that regardless of dimension, we have that

$$||\underline{u} - \underline{x}||_2 = ||\underline{v} - \underline{x}||_2. \tag{32}$$

We can extend this idea, and parameterize the entire line segment between the two points as

$$\{\lambda \underline{v} + (1 - \lambda)\underline{u} : 0 \leq \lambda \leq 1\}. \tag{33}$$

This is a 1-dimensional parameterization of all the points between $\underline{u}$ and $\underline{v}$.

We can extend this idea to that of the **convex hull** - given a set of points $\underline{v}_1, \ldots, \underline{v}_N$, the convex hull is the smallest convex shape that contains all N points. Recall the definition of a convex shape in this case as one that includes all line segments connecting points within it. The convex hull is frequently visualized as a rubber band or rubber bubble, snapped around the set of points (in whatever dimension the problem is stated in).

We can specify the convex hull of a set of points in the following way:

$$\text{ConvexHull}(\underline{v}_1, \ldots, \underline{v}_N) = \left\{ \sum_{i=1}^{N} \lambda_i \underline{v}_i : \sum_{i=1}^{N} \lambda_i = 1, \lambda_i \geq 0 \right\}. \tag{34}$$

We expand this further with the notion of the **span**: the span of a set of vectors is the set of all points expressible as a linear combination of those points, i.e.,

$$\text{Span}(\underline{v}_1, \ldots, \underline{v}_N) = \left\{ \sum_{i=1}^{N} \lambda_i \underline{v}_i : \lambda_1, \ldots, \lambda_N \in \mathbb{R} \right\}. \tag{35}$$

Note, the span of a set of points is necessarily a *vector space*, as adding two points in the span will result in another linear combination of $\underline{v}_1, \ldots, \underline{v}_N$, as will scaling a linear combination.

Recall the idea that when multiplying a matrix A by a vector $\underline{v}$, the result can be interpreted as a linear combination of the columns of A, where the weights are given by the components of $\underline{v}$. In this way, we can think of the *Image* of A, the set of all possible vectors that might be produced, as the span of the columns of A. This is frequently referred to as the *column space* of A.

Similarly, we might define the *row space* of A as the span of the rows of A.

We can also define the *null space* of A, the set of all vectors $\underline{v}$ such that $A\underline{v} = 0$. Note that NullSpace(A) necessarily includes the 0-vector. In fact, the nullspace of a matrix is itself a vector space. Suppose that $\underline{u}, \underline{v}$ are in the nullspace of A. Then

$$A(\alpha\underline{u} + \beta\underline{v}) = \alpha A\underline{u} + \beta A\underline{v} = \alpha * 0 + \beta * 0 = 0, \tag{36}$$

and the linear combination lies in the nullspace as well.

Recall the previous discussion of invertibility of matrices, that conclude that a square matrix A is invertible if and only if $A\underline{x} \neq 0$ for any non-zero vector $\underline{x}$. Another way to state this is that a square matrix A is invertible if the nullspace of A is trivial, or if there is a non-trivial linear combination of the columns that yields the zero vector.

We can use this to motivate the following definition: a set of vectors $\underline{v}_1, \ldots, \underline{v}_N$ is **linearly dependent** if there is a set of $\alpha_1, \ldots, \alpha_N$, not all zero, such that $\alpha_1\underline{v}_1 + \ldots + \alpha_N\underline{v}_N = 0$. Equivalently, a set of vectors is linearly dependent if one of them can be expressed as a non-trivial linear combination of the others, or if one of them lies (non-trivially) in the span of the others.

A set of vectors is **linearly independent** of none of them can be expressed as a linear combination of the others, or equivalently that the only solution to $\alpha_1\underline{v}_1 + \ldots + \alpha_N\underline{v}_N = 0$ is $\alpha_1 = \alpha_2 = \ldots = \alpha_N = 0$.

Note the following useful result: two vectors are linearly independent iff one is not a scalar multiple of the other.

Tying a number of threads together, we may state:

> A square matrix A is invertible if and only if the columns of A are linearly independent.

Basis Sets, Transformations, Dimension, Rank

To get deeper into the structure of vector spaces and collections of vectors, we introduce the idea of a **basis set**: a set $\underline{v}_1, \ldots, \underline{v}_N \in V$ is a **basis set** of V iff $V = \text{span}(\underline{v}_1, \ldots, \underline{v}_N)$ and $\underline{v}_1, \ldots, \underline{v}_N$ are linearly independent.

The practical intuition here is that the vectors $\underline{v}_1, \ldots, \underline{v}_N$ are each distinct (cannot be made from one another) but that *any* vector in V can be made as a linear combination of $\underline{v}_1, \ldots, \underline{v}_N$. The *standard basis* in D dimensions is the collection $\underline{e}_1, \ldots, \underline{e}_D$, where $\underline{e}_i$ is the vector of all 0s, except for ith component which is a 1. Note that any vector $\underline{v} = (v_1, v_2, \ldots, v_D)$ can be represented in terms of this basis set as

$$\underline{v} = v_1\underline{e}_1 + v_2\underline{e}_2 + \ldots + v_D\underline{e}_D. \tag{37}$$

Additionally, the $\{\underline{e}_i\}$ are clearly linearly independent. *Why?*

Building on this idea of representation - given a vector $\underline{v} \in V$ and a basis set, there is a representation of $\underline{v}$ of the form

$$\underline{v} = \lambda_1\underline{v}_1 + \lambda_2\underline{v}_2 + \ldots + \lambda_N\underline{v}_N, \tag{38}$$

and we can identify $\underline{v}$ with respect to this representation, as the vector $\underline{v} = (\lambda_1, \ldots, \lambda_N)$. And in fact, this representation is unique. Suppose that it were not, and we had

$$\underline{v} = \sum_{i=1}^{N} \lambda_i\underline{v}_i = \sum_{i=1}^{N} \alpha_i\underline{v}_i, \tag{39}$$

where the $\{\lambda_i\}, \{\alpha_i\}$ are not all equal (i.e., we have distinct representations). In this case, we would have

$$0 = \sum_{i=1}^{N} (\lambda_i - \alpha_i)\underline{v}_i, \tag{40}$$

and since the $\underline{v}_i$ are linearly independent, we must have that all $\lambda_i - \alpha_i = 0$, or $\lambda_i = \alpha_i$ - i.e., the representations are identical.

Hence, specifying a basis for a vector space V, we can represent any vector in V with respect to that basis. One value of this is the expression of linear transformations as multiplying by a matrix: suppose that V is represented with a basis set $\underline{v}_1, \ldots, \underline{v}_D$; if $\underline{v} = (\lambda_1, \ldots, \lambda_D)$ is represented by its coefficients with respect to this basis set, and then the action of a linear transform can be expressed as

$$F(\underline{v}) = F\left(\sum_{i=1}^{D} \lambda_i\underline{v}_i\right) = \sum_{i=1}^{D} \lambda_i F(\underline{v}_i). \tag{41}$$

Defining the column vector $\underline{A}_i = F(\underline{v}_i)$ (i.e., the $\underline{A}_i$ are the results of transforming the basis specifically), then taking A as the matrix of columns given by $\underline{A}_i$, we have

$$F(\underline{v}) = \sum_{i=1}^{D} \lambda_i\underline{A}_i = A\underline{v}. \tag{42}$$

That is, the effect of a linear transform can be expressed as multiplication by a matrix, where the columns of the matrix are the transformed basis vectors the vector space is expressed with respect to. This solidifies the earlier claim, that linear transformations can be expressed as multiplication by a suitably constructed matrix.

As a final couple of observations: basis sets allow us to define a notion of *dimension*. The **dimension** of V is the size of any basis set, the size of any maximally linearly independent set of vectors. Note that for a given vector space V there will be many (infinitely many) possible basis sets for that vector space, but they will all be the same size.

This allows us to give another condition for invertibility:

> A $D \times D$ dimension matrix A is invertible iff the dimension of the column space of A is D.

The column space is spanned by the columns of A. If the columns are additionally linearly independent (and therefore the matrix is invertible), then the columns form a basis of the column space of A.

The dimension of the column space is the **rank** of the matrix. An important property of matrices (the proof is too long to include here) is that the dimension of the column space of a matrix is always equal to the dimension of the row space - in this case, rank can be applied to both the row and column spaces. We can denote this as

$$\dim(\text{row space } A) = \dim(\text{col space } A) \tag{43}$$

or

$$\text{rank } A = \text{rank } A^{\mathrm{T}}. \tag{44}$$

A similarly useful property (that will become more useful when discussion linear systems of equations or regression) connects the rank of the matrix to the null space of the matrix. If A is a matrix operating on a vector space V, the following relation holds:

$$\text{rank } A + \dim(\text{null space } A) = \dim V. \tag{45}$$

Determinants

One last way we discuss the geometry of vectors, spaces, and matrices, has to do with **determinants**. The determinant of a $D \times D$ matrix A is defined to be the volume of the D-dimensional polytope formed by the columns of A. Consider as an example, $A = I$, where the columns are simply the standard unit basis vectors. This traces out a very simple unit cube (in whatever dimension), and hence we would have that $\det(I) = 1$. For a 2×2 matrix $[[a, b], [c, d]]$, the area of the shape traced out by the vectors is given by $ad - bc$ (signed, depending on the orientation of the vectors). There are explicit formulas for 3×3 matrices, but computing determinants for higher dimensional matrices becomes increasingly complex.

Taking the view of a matrix as acting on vectors when multiplied, the columns of A can be thought of as the 'result' of applying A to the standard basis vectors - stretching, rotating them as appropriate. Hence we can consider $\det(A)$ as effect on the volume of the standard basis cube under the transformation of A. A general shape, linearly transformed by A, will have its volume scaled by $\det(A)$.

Note one result of this volume interpretation of the determinant is this: if the columns of A are linearly *dependent*, then the dimension of the shape formed by the columns will actually be lesser than the dimension of the whole space, i.e., flattened, and have a volume of 0. Summarizing this, $\det(A) = 0$ iff the columns of A are linearly dependent - i.e., we have another condition for invertibility,

> A matrix A is invertible if and only if $\det(A) \neq 0$.

Other properties of the determinant include:

- $\det(I) = 1$ - as stated above.
- $\det(A) = \det(A^{\mathrm{T}})$
- $\det(A^{-1}) = 1/\det(A)$ - if applying the matrix A scales volume by a factor of $\det(A)$, then the *inverse* transformation, multiplying by A^{-1}, shrinks volume by a factor of $1/\det(A)$.

- $\det(AB) = \det(A)\det(B)$ - applying A and B sequentially scales first by a factor of $\det(B)$ then by a factor of $\det(A)$.
- $\det(cA) = c^D \det(A)$ - stretching the columns of A by a factor of c will inflate the volume by a factor of c^D.
- For a triangular matrix A, $\det(A)$ is given by the product of the diagonal entries of A.

Linear Systems, Linear Regression

One application of all this is to systems of linear equations, things like

$$\begin{aligned} 3x + 2y + z &= 1 \\ x + y - z &= 2 \\ x + y + z &= 3. \end{aligned} \tag{46}$$

Defining the matrix M as

$$M = \begin{pmatrix} 3 & 2 & 1 \\ 1 & 1 & -1 \\ 1 & 1 & 1 \end{pmatrix} \tag{47}$$

and the vectors $\underline{x} = (x, y, z)$, $\underline{y} = (1, 2, 3)$, we can describe the above through the matrix equation

$$M\underline{x} = \underline{y}. \tag{48}$$

Note that what this is asking, essentially, is to find a linear combination of the columns of M that yields $\underline{y}$. Taking this column-space centric view of the problem, we could solve it by inverting M, giving the solution

$$\underline{x} = M^{-1}\underline{y}. \tag{49}$$

Note that if M is invertible, this implies that there is a unique solution for $\underline{x}$ - this ties directly into the previous discussions of the columns of M being linearly independent and therefore a basis (giving a unique representation of any other vector) in the case that M is invertible.

If M is not invertible, this implies that either there is no solution to the system or that the solution is not unique.

We can also take a row-space centric view of the problem, noting that every equation is essentially modeled as a row vector, we can think of the entire system with respect to the augmented matrix

$$M' = \begin{pmatrix} 3 & 2 & 1 & 1 \\ 1 & 1 & -1 & 2 \\ 1 & 1 & 1 & 3 \end{pmatrix} \tag{50}$$

In the usual algebraic way, we can scale and add equations to produce new equations (implying that the space of equations is a vector space already), but this amounts to exploring the row space of the above matrix. In particular, subtracting the 2nd from the 3rd, this yields $2z = 1$ or $z = 1/2$. Hence the row space of the above matrix is the same as the row space of

$$\begin{pmatrix} 3 & 2 & 1 & 1 \\ 1 & 1 & -1 & 2 \\ 0 & 0 & 1 & 0.5 \end{pmatrix} \tag{51}$$

At this point, having reduced one of the rows in this way, we can use it to 'clear out' the corresponding column, adding it to the second row and subtracting it from the first to yield

$$\begin{pmatrix} 3 & 2 & 0 & 0.5 \\ 1 & 1 & 0 & 2.5 \\ 0 & 0 & 1 & 0.5 \end{pmatrix} \tag{52}$$

We continue in this way, trying to 'clear out' columns, finding as simple a representation or basis for the row space that we can. Note that at every step here, based on the elementary operations we are doing, the row space of the above matrix has not changed. Continuing on, subtracting twice the second from the first

$$\begin{pmatrix} 1 & 0 & 0 & -4.5 \\ 1 & 1 & 0 & 2.5 \\ 0 & 0 & 1 & 0.5 \end{pmatrix} \tag{53}$$

At this point, we can subtract the first from the second to reveal the final simplified form:

$$\begin{pmatrix} 1 & 0 & 0 & -4.5 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 1 & 0.5 \end{pmatrix} \tag{54}$$

This has the same row space as the original matrix, meaning that from the original equations we could just as easily derive the equations $x = -4.5, y = 7, z = 0.5$. The above process, identifying, isolating, and eliminating one variable at a time through thee elementary operations, is known as Gaussian elimination. From an initial matrix it yields what is known as the 'reduced row-eschelon form' of the matrix, a matrix with an equivalent row space where all entries below the diagonal are zero, and as many above the diagonal have been eliminated as possible. This amounts to a solution to the original system, having decoupled the variables as much as is linearly possible.

Underspecification - The Utility of Gaussian Elimination and the Null Space

Note that if M is not invertible, the inverse method above will not work, but the Gaussian elimination procedure can proceed relatively unharmed. In particular, consider the system

$$\begin{aligned} 3x + 2y + z &= 1 \\ x + y - z &= 2. \end{aligned} \tag{55}$$

This is an example of an *underspecified system*, where there are not enough equations to determine the values of all the variables.

In this case, the weight matrix is not square and hence cannot be inverted, but we can proceed through our exploration of the row space as before:

$$M' = \begin{pmatrix} 3 & 2 & 1 & 1 \\ 1 & 1 & -1 & 2 \end{pmatrix} \tag{56}$$

Adding the two equations, we get $(4, 3, 0, 3)$ or the equation $4x + 3y = 3$ or $x + 3/4y = 3/4$. Exchanging the first row for this, we have

$$\begin{pmatrix} 1 & 3/4 & 0 & 3/4 \\ 1 & 1 & -1 & 2 \end{pmatrix} \tag{57}$$

Subtracting the first from the second, we get

$$\begin{pmatrix} 1 & 3/4 & 0 & 3/4 \\ 0 & 1/4 & -1 & 5/4 \end{pmatrix} \tag{58}$$

This would be the row-eschelon form of the original matrix, giving the system of equations $x + 3/4y = 3/4$ and $1/4y - z = 5/4$. Simplifying it further to the reduced row eschelon form, we can deduct three times the second from the first

$$\begin{pmatrix} 1 & 0 & 3 & -12/4 \\ 0 & 1/4 & -1 & 5/4 \end{pmatrix} \tag{59}$$

And for simplicity, scaling the second row by 4,

$$\begin{pmatrix} 1 & 0 & 3 & -3 \\ 0 & 1 & -4 & 5 \end{pmatrix} \tag{60}$$

Hence we see that the final 'solution' yielded is the simplified system $x + 3z = -3$ and $y - 4z = 5$. This suggests a 1-dimensional solution parameterized by z: any solution (x, y, z) is of the form $(-3 - 3z, 5 + 4z, z)$. It is convenient to express this as $(-3, 5, 0) + z * (-3, 4, 1)$.

This agrees with the geometric intuition: the two initial equations describe a pair of planes, and if the planes intersect, they will intersect at a line, a one dimensional object through the 3-space.

It's worth noting that for the original underspecified system,

$$\begin{aligned} 3x + 2y + z &= 1 \\ x + y - z &= 2. \end{aligned} \tag{61}$$

if we define

$$M = \begin{pmatrix} 3 & 2 & 1 \\ 1 & 1 & -1 \end{pmatrix} \tag{62}$$

so that we are trying to solve $M\underline{x} = \underline{y}$, where $\underline{y} = (1, 2)$, if we have any two solutions $\underline{x}_1$ and $\underline{x}_2$, it must be that

$$M(\underline{x}_2 - \underline{x}_1) = M\underline{x}_1 - M\underline{x}_2 = \underline{y} - \underline{y} = 0, \tag{63}$$

i.e., the difference between any two solutions must be in the null-space of M. Similarly, we can invert this reasoning to argue that any solution must be some initial solution, call it $\underline{x}_0$, plus some element $\underline{\Delta}$ in the null-space of M. So what does the null-space look like?

We want to find the $\underline{\Delta}$ such that $M\underline{\Delta} = 0$, or

$$\begin{aligned} 3x + 2y + z &= 0 \\ x + y - z &= 0. \end{aligned} \tag{64}$$

The second suggests that $z = x+y$, and substituting this into the first gives $4x+3y = 0$, or $y = -(4/3)x$. Substituting in gives us that $z = x - (4/3)x$ or $z = -(1/3)x$, or $x = -3z$, which gives in turn that $y = -(4/3)(-3)z = 4z$. This suggests that anything in the null-space is defined by $(x, y, z) = (-3z, 4z, z) = z * (-3, 4, 1)$, exactly the parameterization of the extra solutions given by the above method using Gaussian elimination. It all comes together.

Over Specification - The Case for Linear Regression

It can also happen that a linear system might be *over specified* - we have more equations than can actually be satisfied. In this case, we typically are interested in as close a solution as we can get. This is frequently the case in terms of linear regression. Suppose we have a number of data vectors $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N$ (each of $D + 1$ dimensions (where the first component is taken to be 1). Each data vector has some corresponding output $y_1, y_2, \ldots, y_N$. We

would like to try to find a linear model to fit this data, i.e., a vector of unknown weights $\underline{w}$ such that $\underline{x_i}.\underline{w} = y_i$ for each i, or

$$X\underline{w} = \underline{y}. \tag{65}$$

However, if we have more equations than variables, it is frequently the case that we will have no exact solution to the above. In this case, we are more interested in the closest solution, or the weight vector $\underline{w}$ that minimizes the error between $X\underline{w}$ and $\underline{y}$, or minimizes

$$||X\underline{w} - \underline{y}||_2^2. \tag{66}$$

This will recur and is frequently of interest in machine learning, so it's worth unpacking some here. In particular:

$$\begin{aligned} ||X\underline{w} - \underline{y}||_2^2 &= \left(X\underline{w} - \underline{y}\right)^{\mathrm{T}} \left(X\underline{w} - \underline{y}\right) \\ &= \left(\underline{w}^{\mathrm{T}}X^{\mathrm{T}} - \underline{y}^{\mathrm{T}}\right) \left(X\underline{w} - \underline{y}\right) \\ &= \underline{w}^{\mathrm{T}}X^{\mathrm{T}}X\underline{w} - \underline{w}^{\mathrm{T}}X^{\mathrm{T}}\underline{y} - \underline{y}^{\mathrm{T}}X\underline{w} - \underline{y}^{\mathrm{T}}\underline{y} \\ &= \underline{w}^{\mathrm{T}} \left[X^{\mathrm{T}}X\right] \underline{w} - 2\underline{w}^{\mathrm{T}}X^{\mathrm{T}}\underline{y} + ||\underline{y}||_2^2. \end{aligned} \tag{67}$$

Note the last line relies on the fact that $\underline{w}^{\mathrm{T}}X^{\mathrm{T}}\underline{y} = \underline{y}^{\mathrm{T}}X\underline{w}$, since they are scalars and transposes of each other.

Viewing this as a general quadratic function in terms of the unknown entries of $\underline{w}$, it follows that the minimum occurs when all the derivatives are zero or, in terms of the gradient, $\nabla_{\underline{w}} = 0$, which gives

$$2\left[X^{\mathrm{T}}X\right]\underline{w} - 2X^{\mathrm{T}}\underline{y} = 0, \tag{68}$$

or

$$\left[X^{\mathrm{T}}X\right]\underline{w} = X^{\mathrm{T}}\underline{y}. \tag{69}$$

We will see the matrix $X^{\mathrm{T}}X$ again - the *similarity matrix*, it quantifies the covariances between the elements of the data. If this matrix is invertible, this suggests a unique solution for the minimizer $\underline{w}$, giving us our formula for the solution to least squares linear regression:

$$\underline{w}^* = \left[X^{\mathrm{T}}X\right]^{-1} X^{\mathrm{T}}\underline{y}. \tag{70}$$

It is worth considering what happens if the matrix $X^{\mathrm{T}}X$ is *not* invertible, however.