Linear Algebra for Dummies

Jorge A. Menendez

October 6, 2017

Contents

1 Matrices and Vectors

An $m \times n$ *matrix* is simply an array of numbers:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

where we define the indexing $\mathbf{A}_{ij} = a_{ij}$ to designate the component in the ith row and jth column of $\mathbf{A}$. The *transpose* of a matrix is obtained by flipping the rows with the columns:

$$\mathbf{A}^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \vdots & & & \vdots \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{bmatrix}$$

which evidently is now an $n \times m$ matrix, with components $\mathbf{A}^T_{ij} = \mathbf{A}_{ji} = a_{ji}$. In other words, the transpose is obtained by simply flipping the row and column indeces.

One particularly important matrix is called the *identity matrix*, which is composed of 1's on the diagonal and 0's everywhere else:

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

It is called the identity matrix because the product of any matrix with the identity matrix is identical to itself:

$$\mathbf{AI} = \mathbf{A}$$

In other words, $\mathbf{I}$ is the equivalent of the number 1 for matrices.

For our purposes, a *vector* can simply be thought of as a matrix with one column[1]:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

We say that such a vector lives in $n-$dimensional space, since it has n components that determine it. This is written as:

$$\mathbf{a} \in \mathbb{R}^n$$

since its components are all real ($a_i \in \mathbb{R}, i = 1, \ldots, n$). The transpose of a vector yields a row vector:

$$\mathbf{a}^T = \begin{bmatrix} a_1 & a_2 & \ldots & a_n \end{bmatrix}$$

A pair of vectors $\mathbf{a}_1, \mathbf{a}_2$ are said to be *linearly independent* if and only if there is no *linear combination* of them that yields zero, i.e. there exists no pair of non-zero scalars c_1, c_2 such that

$$c_1\mathbf{a}_1 + c_2\mathbf{a}_2 = \mathbf{0}$$

where $\mathbf{0}$ is a vector of 0's. If there were, then one would simply be a scaled version of the other:

$$\Leftrightarrow \mathbf{a}_1 = -\frac{c_2}{c_1}\mathbf{a}_2$$

meaning that $\mathbf{a}_1$ and $\mathbf{a}_2$ are parallel and linearly dependent. In other words, two linearly independent vectors are simply two vectors pointing in different directions (i.e. not parallel). Because they are pointing in different directions, the (infinite) set of all possible linear combinations of $\mathbf{a}_1, \mathbf{a}_2$ forms a two-dimensional plane in $\mathbb{R}^n$. If $n > 2$, this plane only contains a subset of all the vectors that exist in $\mathbb{R}^n$ and is thus called a *subspace* of $\mathbb{R}^n$[2] If $n = 2$, then the set of all linear combinations of $\mathbf{a}_1, \mathbf{a}_2$ contains all the vectors in $\mathbb{R}^n$. It is then said that $\mathbf{a}_1, \mathbf{a}_2$ *span* the space $\mathbb{R}^2$, or that they form a *basis* for $\mathbb{R}^2$. In general, any set of n linearly independent vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^n$ will span $\mathbb{R}^n$. Any such set is a basis for $\mathbb{R}^n$: you can express any vector in $\mathbb{R}^n$ as a linear combination of $\mathbf{a}_1, \ldots, \mathbf{a}_n$.

2 Matrix Multiplication

The product of an $m \times n$ matrix $\mathbf{A}$ and a $n \times p$ matrix $\mathbf{B}$ is given by:

$$\mathbf{AB} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & & & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1p} \\ b_{21} & b_{22} & \ldots & b_{2p} \\ \vdots & & & \vdots \\ b_{n1} & b_{n2} & \ldots & b_{np} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n a_{1i}b_{i1} & \sum_{i=1}^n a_{1i}b_{i2} & \ldots & \sum_{i=1}^n a_{1i}b_{ip} \\ \sum_{i=1}^n a_{2i}b_{i1} & \sum_{i=1}^n a_{2i}b_{i2} & \ldots & \sum_{i=1}^n a_{2i}b_{ip} \\ \vdots & & & \vdots \\ \sum_{i=1}^n a_{mi}b_{i1} & \sum_{i=1}^n a_{mi}b_{i2} & \ldots & \sum_{i=1}^n a_{mi}b_{ip} \end{bmatrix}$$

This component-wise definition of a product of matrices is actually the absolute worst way to think about matrix multiplication, but it clearly shows that $(\mathbf{AB})_{ij} = \sum_{k=1}^n \mathbf{A}_{ik}\mathbf{B}_{kj} = \sum_{k=1}^n a_{ik}b_{kj}$. It is

[1] I will always treat a vector as a column, although others are more loose with this, treating a vector as a more general one-dimensional array of numbers. Because vectors will always be columns for me, inner and outer products (see below) are always expressed as $\mathbf{a}^T\mathbf{b}$ and $\mathbf{ab}^T$, respectively, whereas others (e.g. Peter L) would use $\mathbf{a} \cdot \mathbf{b}$ and $\mathbf{ab}$.

[2] Strictly, a subspace $\mathcal{S}$ must obey the following three properties:

- $\mathbf{0} \in \mathcal{S}$
- if $\mathbf{u}, \mathbf{v} \in \mathcal{S}$ then $\mathbf{u} + \mathbf{v} \in \mathcal{S}$
- if $\mathbf{u} \in \mathcal{S}$ then $c\mathbf{u} \in \mathcal{S}$ for any scalar c

These latter two requirements are also expressed as: the set of vectors $\mathcal{S}$ is *closed under vector addition and scalar multiplication*.

thus evident that two matrices can only be multiplied if their inner dimensions agree. In this case, $\mathbf{A}$ has n columns and $\mathbf{B}$ has n rows, so they can be multiplied. Conversely, $\mathbf{BA}$ is not a valid product, since $\mathbf{B}$ has p columns and $\mathbf{A}$ has m rows - their inner dimensions don't agree! This illustrates the important fact that matrix multiplication, unlike scalar multiplication, is not *commutative*: in general, $\mathbf{AB} \neq \mathbf{BA}$. That said, matrix multiplication is associative ($\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$) and distributive ($\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB} + \mathbf{AC}$). Note as well that the dimensionality of a matrix product is given by the outer dimensions: if $\mathbf{A}$ is $m \times n$ and $\mathbf{B}$ is $n \times p$, then $\mathbf{AB}$ is $m \times p$. These are important facts to remember when doing matrix algebra.

Another much more useful way of thinking about matrix multiplication is illustrated by considering the product of a matrix with a vector:

$$\begin{aligned}
\mathbf{Ax} &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 a_{11} + x_2 a_{12} + \dots + x_n a_{1n} \\ x_1 a_{21} + x_2 a_{22} + \dots + x_n a_{2n} \\ \vdots \\ x_1 a_{m1} + x_2 a_{m2} + \dots + x_n a_{mn} \end{bmatrix} \\
&= \begin{bmatrix} x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \dots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} \end{bmatrix} \\
&= \begin{bmatrix} x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \dots + x_n \mathbf{a}_n \end{bmatrix}
\end{aligned}$$

Noting that $\mathbf{A}$ is $m \times n$ and $\mathbf{x}$ is $n \times 1$, we know that the product $\mathbf{Ax}$ must be $m \times 1$: it is a vector. What this example illustrates is that this m-dimensional vector $\mathbf{Ax}$ is a linear combination of the columns of $\mathbf{A}$, given by the m-dimensional vectors $\mathbf{a}_1, \dots, \mathbf{a}_n$. It is thus often useful to think of an $m \times n$ matrix as a collection of vectors $\mathbf{a}_1, \dots, \mathbf{a}_n \in \mathbb{R}^m$ placed side by side. For any n-dimensional vector $\mathbf{x}$, $\mathbf{Ax}$ is then a linear combination of these vectors. Note that this implies that $\mathbf{Ax}$ lives in the (sub)space spanned by $\mathbf{a}_1, \dots, \mathbf{a}_n$. We call this (sub)space the *column space* of $\mathbf{A}$. The notion of the column space of a matrix is extremely useful in linear algebra, and taking linear combinations of the columns of $\mathbf{A}$ is a much easier and intuitive way of understanding what matrix multiplication is.

Let's extend this to multiplying two bona fide matrices together rather than a matrix with a vector. The easiest way of looking at this now is by treating $\mathbf{A}$ and $\mathbf{B}$ as two collections of vectors:

$$\mathbf{AB} = \begin{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} & \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} & \dots & \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{n1} \end{bmatrix} & \begin{bmatrix} b_{12} \\ b_{22} \\ \vdots \\ b_{n2} \end{bmatrix} & \dots & \begin{bmatrix} b_{1p} \\ b_{2p} \\ \vdots \\ b_{np} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_p \end{bmatrix}$$

where $\mathbf{a}_i \in \mathbb{R}^m, \mathbf{b}_i \in \mathbb{R}^n$. We then perform the full matrix multiplication by simply performing a series of matrix-vector products: the jth column of $\mathbf{AB}$ is given by $\mathbf{Ab}_j = \sum_{k=1}^n \mathbf{a}_k b_{kj}$:

$$\mathbf{AB} = \begin{bmatrix} \sum_{k=1}^n \mathbf{a}_k b_{k1} & \sum_{k=1}^n \mathbf{a}_k b_{k2} & \dots & \sum_{k=1}^n \mathbf{a}_k b_{kp} \end{bmatrix}$$

In other words, the columns of $\mathbf{AB}$ are different linear combinations of the columns of $\mathbf{A}$. It is easy to verify that this is equivalent to our above equation $\mathbf{AB}_{ij} = \sum_k a_{ik} b_{kj}$.

In this view of matrices as collections of vectors, we can easily write the matrix transpose as a collection of row vectors:

$$\mathbf{A}^T = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_n^T \end{bmatrix}$$

If we instead start with viewing matrices as collections of row vectors, we can reinterpret matrix multiplication as summing rows rather than columns:

$$\mathbf{AB} = \begin{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \end{bmatrix} \\ \begin{bmatrix} a_{21} & a_{22} & \dots & a_{2n} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1p} \end{bmatrix} \\ \begin{bmatrix} b_{21} & b_{22} & \dots & b_{2p} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} b_{n1} & b_{n2} & \dots & b_{np} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix} \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \vdots \\ \mathbf{b}_n^T \end{bmatrix}$$

where $\mathbf{a}_i \in \mathbb{R}^n, \mathbf{b}_i \in \mathbb{R}^p$. The ith row of $\mathbf{AB}$ is now given by $\mathbf{a}_i^T\mathbf{B} = \sum_{k=1}^n a_{ik}\mathbf{b}_k^T$:

$$\mathbf{AB} = \begin{bmatrix} \sum_{k=1}^n a_{1k}\mathbf{b}_k^T \\ \sum_{k=1}^n a_{2k}\mathbf{b}_k^T \\ \vdots \\ \sum_{k=1}^n a_{mk}\mathbf{b}_k^T \end{bmatrix}$$

Again, this is simply another equivalent way of looking at matrix multiplication.

3 Matrix Inverse, Pseudo-inverse

Consider now the equation

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A}$ is $m \times n$ and, accordingly, $\mathbf{b} \in \mathbb{R}^m, \mathbf{x} \in \mathbb{R}^n$. As we saw above, this equation implies that $\mathbf{b} \in \mathbb{R}^m$ lives in the column space of $\mathbf{A}$. Suppose we want to solve it for an unknown $\mathbf{x}$, given some $\mathbf{A}, \mathbf{b}$. We consider four possible cases:

1. $\mathbf{A}$ is square ($m = n$) and its columns are all linearly independent: if the n columns of $\mathbf{A}$ are linearly independent, they span $\mathbb{R}^m = \mathbb{R}^n$, and the column space of $\mathbf{A}$ is $\mathbf{R}^m$. In this case, there is a unique set of scalars $x_1, \ldots, x_n$ such that $x_1\mathbf{a}_1 + \ldots x_n\mathbf{a}_n = \mathbf{b}$, i.e. there is only one linear combination of $\mathbf{a}_i$'s that is equal to $\mathbf{b}$. $\mathbf{b}$ and $\mathbf{A}$ therefore uniquely determine $\mathbf{x}$[3] and we can write

   $$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

   where $\mathbf{A}^{-1}$ is called the *matrix inverse* of $\mathbf{A}$, defined by the identity

   $$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

   Note that for this to be true, $\mathbf{A}$ must be square: only square matrices have inverses.

2. $\mathbf{A}$ is square ($m = n$) but only $k < n$ of its columns are linearly independent: in this case, the column space of $\mathbf{A}$ is a subspace of $\mathbf{R}^m$, and the equation will only hold for $\mathbf{b}$'s that live in this k-dimensional subspace. For any such $\mathbf{b}$, however, there are many $\mathbf{x}$'s that can satisfy the equation, since any set of k linearly independent vectors can be linearly combined to produce $\mathbf{b}$ (so one solution would be the $\mathbf{x}$ that combines the first k columns and ignores the others, $x_{i>k} = 0$, another would be the $\mathbf{x}$ that combines the last k columns and ignores the others, $x_{i<(k-1)} = 0$, etc.). Therefore, the equation can't be solved (the mapping through $\mathbf{A}$ is not invertible), implying that $\mathbf{A}^{-1}$ doesn't exist: $\mathbf{A}$ is not an invertible matrix. We call such a matrix a *singular* matrix. Note that our analysis crucially involved knowing how many columns of $\mathbf{A}$ are linearly independent - this is called the *rank* of $\mathbf{A}$. When the rank of a square matrix is equal to the number of columns, we say it is full rank. If it is not, then the matrix will necessarily be singular. The rank of a matrix also tells you the dimensionality of its column space. Importantly, it turns out that the number of linearly independent columns of a matrix is always equal to the number of linearly independent rows, so the dimensionality of the row space (the space of all vectors obtained by linear combinations of the rwos of the matrix) is always equal to the dimensionality of the column space (even though these respective subspaces may reside within two different vectors spaces $\mathbb{R}^n, \mathbb{R}^m$)

3. $\mathbf{A}$ is skinny ($m > n$): if $m > n$, then the column space of $\mathbf{A}$ is necessarily a subspace of $\mathbb{R}^m$ and the equation only holds if $\mathbf{b}$ lives within this subspace. If it does, if the columns of $\mathbf{A}$ are independent (i.e. $\mathbf{A}$ is rank n), then there is a unique linear combination of them that yields $\mathbf{b}$, so we can determine $\mathbf{x}$. We can see this algebraically by multiplying both sides of the equation by the *pseudo-inverse* of $\mathbf{A}$:

   $$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

   The pseudo-inverse is the closest thing we have to a matrix inverse for rectangular matrices. Crucially, it exists in this case only because $\mathbf{A}$ is rank n, so by the property of matrix ranks

   $$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^T) = \text{rank}(\mathbf{A}\mathbf{A}^T) = \text{rank}(\mathbf{A}^T\mathbf{A})$$

[3] In other words: the *mapping* from $\mathbf{x}$ to $\mathbf{b}$ through $\mathbf{A}$ is *invertible*

we know that the square $n \times n$ matrix $\mathbf{A}^T\mathbf{A}$ is full rank and therefore invertible. If $\mathbf{A}$ were rank $k < n$, then by the same argument as a above there would be many different $\mathbf{x}$'s providing linear combinations equalling $\mathbf{b}$ (which must now live in the k-dimensional column space of $\mathbf{A}$ for the equality to hold), and $\mathbf{A}^T\mathbf{A}$ would be singular.

4. $\mathbf{A}$ is fat ($m < n$): in this case, the columns of $\mathbf{A}$ form an *overcomplete* basis for its column space, since you have more vectors than necessary to linearly combine to obtain any vector in $\mathbb{R}^m$. Thus, there are many different possible linear combinations of the columns of $\mathbf{A}$ that could yield $\mathbf{b}$ and $\mathbf{x}$ is impossible to recover. Another way of looking at it is like this: $\mathbf{Ax} = \mathbf{b}$ is a set of m equations of the form $x_1a_{i1} + x_2a_{i2} + \ldots + x_na_{in} = b_i, i = 1, \ldots, m$ with $n < m$ unkowns, which is an underdetermined system. Algebraically, we would again try to solve this using the matrix psuedoinverse $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$, but it doesn't exist since $\mathbf{A}$ has maximum rank $m < n$ so the $n \times n$ matrix $\mathbf{A}^T\mathbf{A}$ is rank $m < n$ and therefore singular.

4 Outer products

An outer product between two vectors is just another matrix product, but between an $m \times 1$ matrix (i.e. a vector) and a $1 \times n$ matrix (i.e. a row vector, or vector transposed):

$$\mathbf{u}\mathbf{v}^T = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix} = \begin{bmatrix} u_1v_1 & u_1v_2 & \ldots & u_1v_n \\ u_2v_1 & u_2v_2 & \ldots & u_2v_n \\ \vdots & \vdots & & \vdots \\ u_mv_1 & u_mv_2 & \ldots & u_mv_n \end{bmatrix} = \begin{bmatrix} v_1\mathbf{u} & v_2\mathbf{u} & \ldots & v_n\mathbf{u} \end{bmatrix} = \begin{bmatrix} u_1\mathbf{v}^T \\ u_2\mathbf{v}^T \\ \vdots \\ u_m\mathbf{v}^T \end{bmatrix}$$

where I have given the component-wise view, the column-wise view, and the row-wise view as I did above with matrix multiplication.

Outer products are particularly useful because they give us a third way of expressing matrix multiplication. It turns out that, for $m \times n\mathbf{A}$ and $n \times p\mathbf{B}$,

$$\mathbf{AB} = \sum_{k=1}^{n} \mathbf{a}_k\mathbf{b}_k^T$$

where $\mathbf{a}_k \in \mathbb{R}^m$ is the kth column of $\mathbf{A}$ and $\mathbf{b}_k^T \in \mathbb{R}^p$ is the kth row of $\mathbf{B}$.

Note that an outer product is a rank 1 matrix, since each column (row) is simply a scaled version of the first (second) vector in the product. In fact, any rank k matrix can be expressed as a sum of k rank 1 matrices or outer products.

5 Inner Products

The Euclidean inner product (also called dot product) is again another matrix product, but between a $1 \times n$ matrix (i.e. a row vector) and an $n \times 1$ matrix (i.e. a vector):

$$\mathbf{u}^T\mathbf{v} = \begin{bmatrix} u_1 & u_2 & \ldots & u_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \sum_{k=1}^{n} u_kv_k$$

More generally, an inner product maps two vectors to a scalar[4]. Other non-Euclidean abstract spaces can be defined by defining alternative inner products that don't directly correspond to row

[4]Strictly, an inner product $\langle \cdot, \cdot \rangle$ must satisfy the following three porperties:

- Conjugate symmetry: $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \bar{\mathbf{v}}, \mathbf{u} \rangle$
- Linearity: $\langle a\mathbf{u}, \mathbf{v} \rangle = a\langle \mathbf{u}, \mathbf{v} \rangle$, $\langle \mathbf{u} + \mathbf{w}, \mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle + \langle \mathbf{w}, \mathbf{v} \rangle$
- Positive-definiteness: $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$, $\langle \mathbf{u}, \mathbf{u} \rangle = 0 \Leftrightarrow \mathbf{u} = \mathbf{0}$

It is easy to show that $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T\mathbf{v}$ satsifies these.

times vector matrix multiplication as I have discussed it here. The reason this particular inner product is called "Euclidean" is because it gives you the Euclidean length, or *norm*, of a vector:

$$\|\mathbf{u}\|_2 := \sqrt{\mathbf{u}^T\mathbf{u}} = \sqrt{\sum_{k=1}^{n} u_k^2}$$

which, by Pythagoras' theorem, is the length of vector $\mathbf{u}$. The $:=$ symbol is read "is defined as". This norm $\|\cdot\|_2$ is naturally called the Euclidean norm (or L2 norm), but other non-Euclidean norms $\|\cdot\|_p$ exist which define abstract non-Euclidean spaces (e.g. the L1 norm $\|\mathbf{u}\|_1 = \sum_k u_k$).

The best part of the Euclidean inner product is that it can actually be interpreted in terms of the angle between two vectors. It turns out that the following is true:

$$\mathbf{u}^T\mathbf{v} = \sum_k u_k v_k = \|\mathbf{u}\|\|\mathbf{v}\|\cos\theta$$

where θ is the angle between $\mathbf{u}$ and $\mathbf{v}$, and I have used the shorthand $\|\cdot\| = \|\cdot\|_2$ for the Euclidean norm (I will use this shorthand henceforth). This means that when $\|\mathbf{u}\| = \|\mathbf{v}\| = 1$, we can directly interpret the inner product as a measure of similarity: if $\mathbf{u}$ and $\mathbf{v}$ line up and are identical, $\theta = 0^o$ and $\mathbf{u}^T\mathbf{v} = \cos(0) = 1$. On the other hand, if $\mathbf{u}$ and $\mathbf{v}$ are orthogonal (perpendicular), $\theta = 90^o$ and $\mathbf{u}^T\mathbf{v} = \cos(\pi/2) = 0$. It follows that two vectors $\mathbf{u}, \mathbf{v}$ are orthogonal if and only if $\mathbf{u}^T\mathbf{v} = 0$.

The Euclidean inner product also gives us a final fourth way of viewing matrix multiplication:

$$\mathbf{AB} = \begin{bmatrix} \sum_{i=1}^n a_{1i}b_{i1} & \sum_{i=1}^n a_{1i}b_{i2} & \dots & \sum_{i=1}^n a_{1i}b_{ip} \\ \sum_{i=1}^n a_{2i}b_{i1} & \sum_{i=1}^n a_{2i}b_{i2} & \dots & \sum_{i=1}^n a_{2i}b_{ip} \\ \vdots & & & \vdots \\ \sum_{i=1}^n a_{mi}b_{i1} & \sum_{i=1}^n a_{mi}b_{i2} & \dots & \sum_{i=1}^n a_{mi}b_{ip} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T\mathbf{b}_1 & \mathbf{a}_1^T\mathbf{b}_2 & \dots & \mathbf{a}_1^T\mathbf{b}_p \\ \mathbf{a}_2^T\mathbf{b}_1 & \mathbf{a}_2^T\mathbf{b}_2 & \dots & \mathbf{a}_2^T\mathbf{b}_p \\ \vdots & & & \vdots \\ \mathbf{a}_m^T\mathbf{b}_1 & \mathbf{a}_m^T\mathbf{b}_2 & \dots & \mathbf{a}_m^T\mathbf{b}_p \end{bmatrix}$$

where $\mathbf{a}_k^T \in \mathbb{R}^n$ is the kth row of $\mathbf{A}$ and $\mathbf{b}_k \in \mathbb{R}^n$ is the kth column of $\mathbf{B}$. Another way of writing this is: $(\mathbf{AB})_{ij} = \mathbf{a}_i^T\mathbf{b}_j$. Viewing matrix multiplication in this way illustrates two important concepts.

Firstly, consider an $n \times n$ square matrix $\mathbf{A}$ with columns $\mathbf{a}_1, \dots, \mathbf{a}_n$ that are all orthogonal to each other and unit length ($\|\mathbf{a}_i\| = 1$). Such a matrix is called an *orthogonal matrix*, and it has the following important properties:

- Since $\mathbf{a}_1, \dots, \mathbf{a}_n$ are orthogonal, they are all linearly independent of each other so $\mathbf{A}$ is full rank and therefore invertible ($\mathbf{A}^(- 1)$ exists)
- Since $\mathbf{a}_1, \dots, \mathbf{a}_n$ are orthogonal and unit length, $\mathbf{a}_i^T\mathbf{b}_j = \delta_{ij}$[5]. So, $(\mathbf{A}^T\mathbf{A})_{ij} = \mathbf{a}_i^T\mathbf{b}_j = \delta_{ij}$, meaning that $\mathbf{A}^T\mathbf{A} = \mathbf{I}$.
- It turns out that $\mathbf{AA}^T = \mathbf{I}$ as well
- By the definition of the matrix inverse, the above two points imply that $\mathbf{A}^{-1} = \mathbf{A}^T$

Second, consider the equation

$$\mathbf{Ax} = \mathbf{0}$$

We can now write this as:

$$\mathbf{Ax} = \begin{bmatrix} \mathbf{a}_1^T\mathbf{x} \\ \mathbf{a}_2^T\mathbf{x} \\ \vdots \\ \mathbf{a}_m^T\mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where $\mathbf{A}$ is $m \times n$ and $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{a}_k^T \in \mathbb{R}^n$ giving the kth row of $\mathbf{A}$. As we saw above, $\mathbf{a}_k^T\mathbf{x} = 0$ implies that $\mathbf{x}$ is orthogonal to $\mathbf{a}_k$, meaning that this equation tells us that $\mathbf{x}$ is orthogonal to all the rows of $\mathbf{A}$: it is orthogonal to the entire subspace spanned by the rows of $\mathbf{A}$ (the row space of $\mathbf{A}$). The space of all such vectors $\mathbf{x}$ orthogonal to the row sapce of $\mathbf{A}$ is called the *nullspace* of $\mathbf{A}$. Note that

[5] This is called the *Kronecker delta*:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

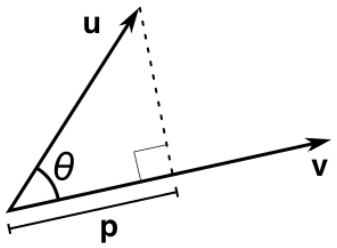

Figure 1: $\mathbf{p}$ is the projection of $\mathbf{u}$ onto $\mathbf{v}$, the scalar projection is $\|\mathbf{p}\|$

the above equation implies the columns of $\mathbf{A}$ are not linearly independent, since it directly tells us there is a linear combination of them that equals $\mathbf{0}$. This also means that $\mathbf{A}$ is not full rank. Indeed, the dimensionality of the nullspace is given by $n - k$, where n is the dimensionality of the rows of $\mathbf{A}$ and k is its rank, which is equal to the dimensionality of the subspace spanned by the rows (or columns) of $\mathbf{A}$. There are a lot of deep connections here...

Inner products are also useful for computing the *projection* of a vector onto another vector, illustrated in figure 1. In this case, $\mathbf{p}$ is the projection of $\mathbf{u}$ onto $\mathbf{v}$. Importantly, $\mathbf{p}$ is the vector in the subspace spanned by $\mathbf{v}$ (i.e. the one-dimensional subspace consisting of all scaled versions of $\mathbf{v}$) closest to $\mathbf{u}$. It is easy to prove this formally, but the intuition is evident in figure 1: if I make $\mathbf{p}$ longer or shorter along $\mathbf{v}$ (i.e. move it along the subspace spanned by $\mathbf{v}$), the distance between $\mathbf{u}$ and $\mathbf{p}$ (the length of the dotted line) will only get longer. By definition, $\mathbf{p}$ and $\mathbf{u}$ form a right triangle, so we can use our standard trigonometric rules to show that

$$\|\mathbf{p}\| = \|\mathbf{u}\| cos\theta = \frac{\mathbf{u}^T\mathbf{v}}{\|\mathbf{v}\|}$$

This quantity is called the *scalar projection* of $\mathbf{u}$ onto $\mathbf{v}$. To find the actual projection vector $\mathbf{p}$ we first note that, since $\mathbf{p}$ lies in the subspace spanned by $\mathbf{v}$, $\mathbf{p} = a\mathbf{v}$ for some real scalar a. We then solve for a by equating the norm of $\mathbf{p} = a\mathbf{v}$ to the scalar projection $\frac{\mathbf{u}^T\mathbf{v}}{\|\mathbf{v}\|}$:

$$\|\mathbf{p}\| = \sqrt{\mathbf{p}^T\mathbf{p}} = \sqrt{a^2\mathbf{v}^T\mathbf{v}} = a\|\mathbf{v}\| = \frac{\mathbf{u}^T\mathbf{v}}{\|\mathbf{v}\|} \Leftrightarrow a = \frac{\mathbf{u}^T\mathbf{v}}{\|\mathbf{v}\|^2} = \frac{\mathbf{u}^T\mathbf{v}}{\mathbf{v}^T\mathbf{v}}$$

We thus have that the projection of $\mathbf{u}$ onto $\mathbf{v}$ is given by

$$\mathbf{p} = \frac{\mathbf{u}^T\mathbf{v}}{\mathbf{v}^T\mathbf{v}}\mathbf{v}$$

Because $\mathbf{p}$ is the vector along $\mathbf{v}$ closest to $\mathbf{u}$, projections will turn out to be an extremely useful quantity for solving least-squares problems where you want to minimize the distance between a vector and a subspace. One such problem is linear regression.

6 Example: Linear Regression

In linear regression, we want to estimate the linear relationship between a dependent variable y (e.g. IQ) and a set of independent features $x_1, \ldots, x_k$ (e.g. height, weight, income). Mathematically, we translate this to finding the set of weights $w_1, \ldots, w_k$ such that

$$y = w_1x_1 + w_2x_2 + \ldots + w_kx_k$$

But the relationship will rarely be exactly linear, so we want to find the weights that make the right hand side as close as possible to the left hand side of the equation. To do so, we obtain a set of observations y_i with corresponding features $x_{i1}, \ldots, x_{ik}$ to get the best possible estimate of $w_1, \ldots, w_k$. Arranging all n observations into a vector $\mathbf{y}$ and the corresponding features into a matrix $\mathbf{X}$, applying the above equation to each of the n data points gives us the following system

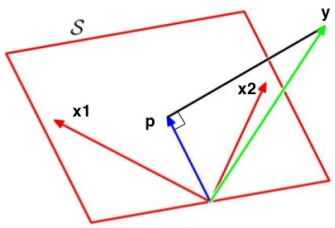

Figure 2: $\mathbf{x}_1, \mathbf{x}_2$ are the two n-dimensional columns of $\mathbf{X}$, providing the basis of the two-dimensional column space $\mathcal{S}$ (assuming $\mathbf{x}_1, \mathbf{x}_2$ are linearly independent). $\mathbf{p}$ is the projection of $\mathbf{y}$ onto $\mathcal{S}$.

of equations, naturally expressed as a matrix equation:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} w_1 x_{11} + w_2 x_{12} + \ldots + w_k x_{1k} \\ w_1 x_{21} + w_2 x_{22} + \ldots + w_k x_{2k} \\ \vdots \\ w_1 x_{n1} + w_2 x_{n2} + \ldots + w_k x_{nk} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1k} \\ x_{21} & x_{22} & \ldots & x_{2k} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{nk} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix} = \mathbf{X}\mathbf{w}$$

Seen now as a matrix equation, the goal of linear regression is to find the linear combination of columns of $\mathbf{X}$ (i.e. n-dimensional vectors containing a set of measurements of a single feature) that is closest to $\mathbf{y}$. In other words, if $\mathcal{S}$ is the column space of $\mathbf{X}$, we want to find the vector in $\mathcal{S}$ that is closest to $\mathbf{y}$. As in our one-dimensional example above (i.e. figure 1), this is given by the projection of $\mathbf{y}$ onto $\mathcal{S}$. The case of $k = 2$ is illustrated in figure 2.

As long as the columns of $\mathbf{X}$ are linearly independent, we can compute this projection by simply projecting $\mathbf{y}$ onto each of the columns of $\mathbf{X}$, i.e. each of the vectors in the basis of $\mathcal{S}$. This can be done simply using our projection formula from above. The solution is:

$$\mathbf{p} = \sum_{i=1}^{k} \frac{\mathbf{x}_i^T \mathbf{y}}{\mathbf{x}_i^T \mathbf{x}_i} \mathbf{x_i}$$

In other words, the set of weights $\mathbf{w}$ that make $\mathbf{X}\mathbf{w}$ as close as possible to $\mathbf{y}$ are given by

$$w_i = \frac{\mathbf{x}_i^T \mathbf{y}}{\mathbf{x}_i^T \mathbf{x}_i}, \quad i = 1, \ldots, k$$

An algebraically simpler but less intuitive way to approach this problem is to simply solve for $\mathbf{w}$ using the pseudo-inverse of $\mathbf{X}$:

$$\mathbf{y} = \mathbf{X}\mathbf{w} \Leftrightarrow (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{w}$$

which exists only if the columns of k are linearly independent. This will generally be true if the k features are uncorrelated. This is in fact equivalent to our earlier solution.

7 Eigenstuff

I start by first defining eigenvectors and eigenvalues, and then show why they are useful and important.

The best way to think about an eigenvector of a square $n \times n$ matrix $\mathbf{A}$ is by thinking about $\mathbf{A}$ as defining a mapping $\mathcal{T}$:

$$\mathcal{T} : \mathbf{x} \mapsto \mathbf{A}\mathbf{x}$$

In English, $\mathbf{A}$ maps the vector $\mathbf{x}$ to the new vector $\mathbf{A}\mathbf{x}$. An *eigenvector* of $\mathbf{A}$ is a vector that maintains its direction through this mapping:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

where λ is just a scalar. In other words, when passed through the mapping $\mathbf{A}$, an eigenvector of $\mathbf{A}$ is simply rescaled, keeping its same orientation as before. The rescaling factor λ is called the

eigenvalue associated with that particular eigenvector $\mathbf{v}$, which we always assume to have unit length ($\|\mathbf{v}\| = 1$). Note that for this to be possible, $\mathbf{Av}$ has to have the same dimensions as $\mathbf{v}$ - in other words, $\mathbf{A}$ must be square. Only square matrices have eigenvectors and eigenvalues[6]. It turns out an $n \times n$ matrix of rank k will always have n eigenvectors with k non-zero associated eigenvalues.

One of the most useful applications of eigenvectors and eigenvalues is for decomposing a matrix into a form that is often easier to work with. Let $\mathbf{V} = \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n\end{bmatrix}$ be a matrix containing all n eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^n$ of an $n \times n$ matrix $\mathbf{A}$ as columns. Using our summing columns view of matrix multiplication, we first note that

$$\mathbf{AV} = \begin{bmatrix}\mathbf{A}\mathbf{v}_1 & \mathbf{A}\mathbf{v}_2 & \dots & \mathbf{A}\mathbf{v}_n\end{bmatrix} = \begin{bmatrix}\lambda_1\mathbf{v}_1 & \lambda_2\mathbf{v}_2 & \dots & \lambda_n\mathbf{v}_n\end{bmatrix}$$

which (again using our summing columns view) can be rewritten as:

$$\mathbf{AV} = \begin{bmatrix}\lambda_1\mathbf{v}_1 & \lambda_2\mathbf{v}_2 & \dots & \lambda_n\mathbf{v}_n\end{bmatrix} = \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n\end{bmatrix} \begin{bmatrix}\lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \lambda_n\end{bmatrix} = \mathbf{V\Lambda}$$

where we call $\mathbf{\Lambda}$ the diagonal matrix with the eigenvalues on the diagonal and 0's everywhere else. Now if the eigenvectors of $\mathbf{A}$ are linearly independent (which necessitates that $\mathbf{A}$ be full rank), $\mathbf{V}$ is invertible and we can write two crucial properties of matrices with linearly independent eigenvectors:

$$\begin{aligned}\mathbf{AV} &= \mathbf{V\Lambda} \\ \Leftrightarrow \mathbf{A} &= \mathbf{V\Lambda V}^{-1} \\ \Leftrightarrow \mathbf{V}^{-1}\mathbf{AV} &= \mathbf{\Lambda}\end{aligned}$$

The first one yields the *eigendecomposition* of $\mathbf{A}$. Using the sum of outer products view of matrix multiplication, it also gives us

$$\mathbf{A} = \sum_{k=1}^{n} \lambda_n \mathbf{v}_k \mathbf{v}_k^\dagger$$

where $\mathbf{v}_k^\dagger$ is the kth row of $\mathbf{V}^{-1}$. This is famously allegedly the only linear algebra fact that Peter Latham knows (which is obviously not true - but it does mean he will use this all the time), so we cll it PEL's rule. The second property is called *diagonalization*: any matrix with linearly independent eigenvectors can be transformed into a diagonal matrix in this way. In other words, a matrix is *diagonalizable* if and only if it has linearly inedependent eigenvectors (i.e. these two statements are equivalent).

Why is this useful? Consider computing the powers of a square matrix:

$$\mathbf{A}^k = \mathbf{AAA}\dots\mathbf{A} = \mathbf{V\Lambda V}^{-1}\mathbf{V\Lambda V}^{-1}\mathbf{V\Lambda V}^{-1}\mathbf{V}\dots\mathbf{V}^{-1}\mathbf{V\Lambda V}^{-1}$$

By definition of the inverse, $\mathbf{V}^{-1}\mathbf{V} = \mathbf{I}$, so

$$\mathbf{A}^k = \mathbf{V\Lambda I\Lambda I\Lambda}\dots\mathbf{I\Lambda V}^{-1} = \mathbf{V\Lambda}^k\mathbf{V}^{-1}$$

since by definition of the identity matrix (above) $\mathbf{\Lambda I} = \mathbf{\Lambda}$. It turns out this makes the computation of powers of $\mathbf{A}$ efficient on a computer since multiplying diagonal matrices is computationally cheap, but more importantly it can give us some intuitions in other settings. Consider, for example, a discrete-time linear dynamical system:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1}$$

where the state at time t is linear transformation of the previous state at time $t-1$. Assuming $\mathbf{A}$ is diagonalizable (i.e. its eigenvectors are all linearly independent), we can write:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} = \mathbf{AA}\mathbf{x}_{t-2} = \dots = \mathbf{A}^t\mathbf{x}_0 = \mathbf{V\Lambda}^t\mathbf{V}^{-1}\mathbf{x} = \sum_{k=1}^{n} \lambda^t \mathbf{v}_k \mathbf{v}_k^\dagger \mathbf{x}$$

[6]There are analogs for rectangular matrices but they are not discussed here, cf. singular value decomposition

where in the last equality we used the fact that $\mathbf{\Lambda}^t$ is a diagonal matrix just like $\mathbf{\Lambda}$ but with λ_k^t's on the diagonal. Letting $c_k = \mathbf{v}^\dagger \mathbf{x}$, we note that the state at time t is just a linear combination of the eigenvectors of $\mathbf{A}$, weighted by their associated eigenvalues to the power of t:

$$\mathbf{x}_t = \sum_k c_k \lambda_k^t \mathbf{v}_k$$

So if the largest eigenvalue λ_1 is greater than 1, λ_1^t will quickly diverge over time and dominate all the other terms in the sum so that, as t gets big, $\mathbf{x}_t$ approaches $\lambda_1^t \mathbf{v}_1$. On the other hand, if all the eigenvalues are between 0 and 1, λ_k^t will quickly decay to 0 for all k, so that the system eventually settles at $\mathbf{x}_t = \mathbf{0}$ for large t. This illulstrates how important and meaningful the eigenvectors of a matrix are: if you repeat the transformation implied by the matrix over and over again, the limit over repetitions is determined by the eigenvectors and values.

This happens to be true for a continuous time dynamical system as well. Consider first the *matrix exponential*, defined just as your vanilla scalar exponential by the power series[7]

$$\begin{aligned}
e^{\mathbf{A}} &= \mathbf{I} + \mathbf{A} + \frac{1}{2!}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \dots \\
&= \mathbf{V}\mathbf{V}^{-1} + \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1} + \frac{1}{2!}\mathbf{V}\mathbf{\Lambda}^2\mathbf{V}^{-1} + \frac{1}{3!}\mathbf{V}\mathbf{\Lambda}^3\mathbf{V}^{-1} + \dots \\
&= \mathbf{V}(\mathbf{I} + \mathbf{\Lambda} + \frac{1}{2!}\mathbf{\Lambda}^2 + \frac{1}{3!}\mathbf{\Lambda}^3 + \dots)\mathbf{V}^{-1} \\
&= \mathbf{V}e^{\mathbf{\Lambda}}\mathbf{V}^{-1}
\end{aligned}$$

which is an easy expression to work with since taking the matrix exponential of a diagonal matrix is the same as exponentiating each of its diagonal components:

$$(e^{\mathbf{\Lambda}})_{ii} = e^{\lambda_i}$$

Just like in the scalar case, it also holds that

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} e^{\mathbf{A}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\left[\mathbf{I} + \mathbf{A}t + \frac{1}{2!}\mathbf{A}^2t^2 + \frac{1}{3!}\mathbf{A}^3t^3 + \dots\right] \\
&= \mathbf{0} + \mathbf{A} + \mathbf{A}^2t + \frac{1}{2!}\mathbf{A}^3t^2 + \dots \\
&= \mathbf{A}\left(\mathbf{I} + \mathbf{A}t + \frac{1}{2!}\mathbf{A}^2t^2 + \dots\right) \\
&= \mathbf{A}e^{\mathbf{A}t}
\end{aligned}$$

We can use this to solve a system of linear differential equations:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \begin{bmatrix} \frac{\mathrm{d}x_1}{\mathrm{d}t} \\ \frac{\mathrm{d}x_2}{\mathrm{d}t} \\ \vdots \\ \frac{\mathrm{d}x_n}{\mathrm{d}t} \end{bmatrix} = \mathbf{A}\mathbf{x} \Leftrightarrow \mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0)$$

Using the eigendecomposition and outer product view of matrix multiplication (i.e. using PEL's rule), as well as the fact that the matrix exponential of a diagonal matrix yields the exponentials of its diagonal components, we have

$$\mathbf{x}(t) = \sum_{k=1}^{n} e^{\lambda_k t}\mathbf{v}_k\mathbf{v}_k^\dagger\mathbf{x}(0)$$

Again, the long run behavior of $\mathbf{x}(t)$ is determined by its eigenvalues and eigenvectors: if the largest eigenvalue $\lambda_1 > 0$, $e^{\lambda_1 t}$ will grow faster than any of the other terms in the sum and eventually dominate them, aligning $\mathbf{x}(t)$ with its associated eigenvector $\mathbf{v}_1$ as it grows to infinity. If $\lambda_1 < 0$, on the other hand, $e^{\lambda_1 t}$ will go to zero as t gets big and $\mathbf{x}(t)$ will go to $\mathbf{0}$ in the long run.

[7] https://en.wikipedia.org/wiki/Exponential_function#Formal_definition

8 Example: Covariance Matrices

The *covariance* between two random variables X_i, X_j is defined as

$$\text{cov}[X_i, X_j] = \mathbb{E}\left[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])\right]$$

where $\mathbb{E}[X_i]$ is the expectated value of random variable X_i. A multivariate n-dimensional random variable is simply a vector composed of a collection of n random variables $X_1, \ldots, X_n$:

$$\mathbf{x} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}$$

The variance of this vector over n-dimensional space is described by its *covariance matrix*:

$$\mathbf{\Sigma} = \begin{bmatrix} \text{cov}[X_1, X_1] & \text{cov}[X_1, X_2] & \ldots & \text{cov}[X_1, X_n] \\ \text{cov}[X_2, X_1] & \text{cov}[X_2, X_2] & \ldots & \text{cov}[X_2, X_n] \\ \vdots & \vdots & & \vdots \\ \text{cov}[X_n, X_1] & \text{cov}[X_n, X_2] & \ldots & \text{cov}[X_n, X_n] \end{bmatrix}$$

where $\text{cov}[X_i, X_i] = \mathbb{E}\left[(X_i - \mathbb{E}[X_i])^2\right] = \text{var}[X_i]$.

Note that since $\text{cov}[X_i, X_j] = \text{cov}[X_j, X_i]$, $\mathbf{\Sigma}_{ij} = \mathbf{\Sigma}_{ji}$ and therefore $\mathbf{\Sigma} = \mathbf{\Sigma}^T$. This is a very special property, and we such matrices *symmetric matrices*. Symmetric matrices all inherit the following properties:

- The eigenvalues of $\mathbf{\Sigma}$ are all real
- The eigenvectors of $\mathbf{\Sigma}$ are orthogonal

This latter point implies that when we eigendecompose a symmetric matrix $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, $\mathbf{V}$ is an orthonormal matrix since its columns are unit length (all eigenvectors always are) and orthogonal (since $\mathbf{A}$ is symmetric). This means that $\mathbf{V}^{-1} = \mathbf{V}^T$, so that $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$.

Covariance matrices in particular hold the additional property that they are *positive semi-definite*, meaning that their eigenvalues are all greater than or equal to 0. Using PEL's rule, we note that this implies that for any vector $\mathbf{u}$ and $n \times n$ covariance matrix $\mathbf{\Sigma}$ with eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and associated eigenvalues $\lambda_1, \ldots, \lambda_n \geq 0$,

$$\mathbf{u}^T\mathbf{\Sigma}\mathbf{u} = \sum_k \mathbf{u}^T\lambda_k\mathbf{v}\mathbf{v}^T\mathbf{u} = \sum_k \lambda_k(\mathbf{u}^T\mathbf{v})^2 \geq 0$$

This is in fact the defining statement of a positive semi-definite symmetric matrix, and it holds if and only if the matrix has eigenvalues greater than or equal to 0.

The covariance of a sample of n d-dimensional data points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)} \in \mathbb{R}^d$ with mean $\bar{\mathbf{x}}$ can be computed as a sum of outer products:

$$\begin{aligned}\mathbf{\Sigma} &= \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n (x_1^{(i)} - \bar{x}_1)(x_1^{(i)} - \bar{x}_1) & \frac{1}{n}\sum_{i=1}^n (x_1^{(i)} - \bar{x}_1)(x_2^{(i)} - \bar{x}_2) & \ldots & \frac{1}{n}\sum_{i=1}^n (x_1^{(i)} - \bar{x}_1)(x_d^{(i)} - \bar{x}_d) \\ \frac{1}{n}\sum_{i=1}^n (x_2^{(i)} - \bar{x}_2)(x_1^{(i)} - \bar{x}_1) & \frac{1}{n}\sum_{i=1}^n (x_2^{(i)} - \bar{x}_2)(x_2^{(i)} - \bar{x}_2) & \ldots & \frac{1}{n}\sum_{i=1}^n (x_2^{(i)} - \bar{x}_2)(x_d^{(i)} - \bar{x}_d) \\ \vdots & \vdots & & \vdots \\ \frac{1}{n}\sum_{i=1}^n (x_d^{(i)} - \bar{x}_d)(x_1^{(i)} - \bar{x}_1) & \frac{1}{n}\sum_{i=1}^n (x_d^{(i)} - \bar{x}_d)(x_2^{(i)} - \bar{x}_2) & \ldots & \frac{1}{n}\sum_{i=1}^n (x_d^{(i)} - \bar{x}_d)(x_d^{(i)} - \bar{x}_d) \end{bmatrix} \\ &= \frac{1}{n}\sum_{i=1}^n (\mathbf{x}^{(i)} - \bar{\mathbf{x}})(\mathbf{x}^{(i)} - \bar{\mathbf{x}})^T\end{aligned}$$

In many cases, we are interested in knowing how much such multivariate data varies along a particular direction $\mathbf{v}$. Rather than averaging data points' squared deviations from the mean, in this case we take the squared scalar projections of the deviations onto $\mathbf{v}$. Since we are only interested in the direction of $\mathbf{v}$, we set it to be a unit vector with length $\|\mathbf{v}\| = 1$, which means the scalar projection of any vector $\mathbf{u}$ onto $\mathbf{v}$ is given by their inner product $\mathbf{u}^T\mathbf{v}$ (since $\mathbf{v}^T\mathbf{v} = 1$).

Taking the average over all data points then gives us the following expression for the variance along direction $\mathbf{v}$:

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}((\mathbf{x}^{(i)}-\bar{\mathbf{x}})^T\mathbf{v})^2 &= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)}-\bar{\mathbf{x}})^T\mathbf{v}(\mathbf{x}^{(i)}-\bar{\mathbf{x}})^T\mathbf{v} \\ &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{v}^T(\mathbf{x}^{(i)}-\bar{\mathbf{x}})(\mathbf{x}^{(i)}-\bar{\mathbf{x}})^T\mathbf{v} \\ &= \mathbf{v}^T\left(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)}-\bar{\mathbf{x}})(\mathbf{x}^{(i)}-\bar{\mathbf{x}})^T\right)\mathbf{v} \\ &= \mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v}\end{aligned}$$

where in the second line we used the fact that for any two vectors $\mathbf{u}, \mathbf{v}$, $\mathbf{u}^T\mathbf{v} = \mathbf{v}^T\mathbf{u}$, and in the third line we used the fact that matrix multiplication is distributive ($\mathbf{AB} + \mathbf{AC} = \mathbf{A}(\mathbf{B} + \mathbf{C})$)

9 Example: PCA

The point of *Principal Components Analysis* (PCA), is to find the directions in data space along which a given data sample $\mathbf{x}_1, \ldots, \mathbf{x}_n$ varies the most. In other words, we want to find the direction $\mathbf{v}$ along which the data has most variance. As we saw above, this translates to the following optimization problem:

$$\begin{aligned}\max_{\mathbf{v}} \quad & \mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v} \\ \text{subject to} \quad & \|\mathbf{v}\| = 1\end{aligned}$$

where $\boldsymbol{\Sigma} = \frac{1}{n}\sum_i(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$ is the sample covariance matrix. The second line gives us the constraint that the length of $\mathbf{v}$ be 1, since this is necessary for the expression $\mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v}$ to be equal to the variance of the data along the direction of $\mathbf{v}$ (see the derivation of this expression in the end of last section).

We proceed to solve this optimization problem by using a Lagrange multiplier λ to implement the constraint $\mathbf{v}^T\mathbf{v} - 1 = 0$, which is equivalent to our constraint $\|\mathbf{v}\| = 1$ (doing it this way just makes the algebra easier). The solution is then given by the equation

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{v}}\left[\mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v} - \lambda(\mathbf{v}^T\mathbf{v} - 1)\right] = \mathbf{0}$$

Using the fact that $\frac{\mathrm{d}}{\mathrm{d}\mathbf{v}}\mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v} = 2\boldsymbol{\Sigma}\mathbf{v}$ and $\frac{\mathrm{d}}{\mathrm{d}\mathbf{v}}\mathbf{v}^T\mathbf{v} = 2\mathbf{v}$, we have:

$$\begin{aligned}2\boldsymbol{\Sigma}\mathbf{v} - 2\lambda\mathbf{v} &= 0 \\ \Leftrightarrow \boldsymbol{\Sigma}\mathbf{v} &= \lambda\mathbf{v}\end{aligned}$$

In other words, the optimal direction $\mathbf{v}$ is an eigenvector of the covariance matrix $\boldsymbol{\Sigma}$! Plugging this back into our objective function we want to maximize, we have that the variance along $\mathbf{v}$ is given by

$$\mathbf{v}^T\boldsymbol{\Sigma}\mathbf{v} = \lambda\mathbf{v}^T\mathbf{v} = \lambda$$

So the direction along which the data has most variance is given by the eigenvector of $\boldsymbol{\Sigma}$ with largest eigenvalue. And the $k < n$-dimensional subspace that contains the most variance of the data is given by the subspace spanned by the k eigenvectors of $\boldsymbol{\Sigma}$ with the k largest eigenvalues.

10 Useful resources

- MIT OpenCourseWare "Linear Algebra" course by Gilbert Strang (https://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/video-lectures/)
- 3Blue1Brown channel on YouTube (https://www.youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab)