8

Eigenvalues and Eigenvectors

Consider the matrix

$$\mathbf{A} = \begin{pmatrix} 2 & 7 \\ -1 & -6 \end{pmatrix}$$

Multiplying $\mathbf{A}$ by the vector $\mathbf{x}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ gives an interesting result.

$$\mathbf{A}\mathbf{x}_1 = \begin{pmatrix} 2 & 7 \\ -1 & -6 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ -5 \end{pmatrix} = -5 \begin{pmatrix} -1 \\ 1 \end{pmatrix} = -5\mathbf{x}_1$$

Similarly, with $\mathbf{x}_2 = \begin{pmatrix} -7 \\ 1 \end{pmatrix}$:

$$\mathbf{A}\mathbf{x}_2 = \begin{pmatrix} 2 & 7 \\ -1 & -6 \end{pmatrix} \begin{pmatrix} -7 \\ 1 \end{pmatrix} = \begin{pmatrix} -7 \\ 1 \end{pmatrix} = \mathbf{x}_2$$

In both cases, multiplication by $\mathbf{A}$ returned a scalar multiple of the vector (-5 for $\mathbf{x}_1$ and 1 for $\mathbf{x}_2$). This is not a property of solely the matrix $\mathbf{A}$, since the vector $\mathbf{x}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ is not transformed by a single scalar.

$$\mathbf{A}\mathbf{x}_3 = \begin{pmatrix} 2 & 7 \\ -1 & -6 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 9 \\ 5 \end{pmatrix} \neq \lambda\mathbf{x}_2$$

Similarly, the results we are seeing are not properties of the vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, since they do not become scalar multiples of themselves when multiplied by other matrices.

$$\mathbf{B} = \begin{pmatrix} 2 & 1 \\ -3 & 0 \end{pmatrix}$$

$$\mathbf{B}\mathbf{x}_1 = \begin{pmatrix} 2 & 1 \\ -3 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \end{pmatrix} \neq \lambda\mathbf{x}_1$$

$$\mathbf{B}\mathbf{x}_2 = \begin{pmatrix} 2 & 1 \\ -3 & 0 \end{pmatrix} \begin{pmatrix} -7 \\ 1 \end{pmatrix} = \begin{pmatrix} -13 \\ 21 \end{pmatrix} \neq \lambda\mathbf{x}_2$$

The phenomena we're observing is a result of the paring between the matrix $\mathbf{A}$ and the vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. In general, we see that multiplying a vector by a matrix returns a scalar multiple of the vector, or

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

Any vector $\mathbf{x}$ that obeys the above relationship is called an *eigenvector* of the matrix $\mathbf{A}$. The scalar λ is called the *eigenvalue* associated with the eigenvector $\mathbf{x}$. The vector $\mathbf{x}$ is an eigenvector of the matrix $\mathbf{A}$; it is not generally an eigenvector of other matrices.

Eigenvectors were originally called *characteristic* vectors, as they describe the character of the matrix. German mathematicians dropped this nomenclature in favor of the German prefix "eigen-", which mean "own". An eigenvector can be viewed as one of a matrix's "own" vectors since it is not rotated when transformed by multiplication.

In the example above, the matrix $\mathbf{A} = \begin{pmatrix} 2 & 7 \\ -1 & -6 \end{pmatrix}$ has two eigenvectors, $\mathbf{v}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ with eigenvalue $\lambda_1 = -5$, and $\mathbf{v}_2 = \begin{pmatrix} -7 \\ 1 \end{pmatrix}$ with eigenvector $\lambda_2 = 1$.

8.1 Properties of Eigenvectors and Eigenvalues

Only square matrices have eigenvectors and eigenvalues. An n by n matrix of real numbers can have up to n distinct eigenvectors. Each eigenvector is associated with an eigenvalue, although the eigenvalues can be duplicated. Said another way, two eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ of a matrix will never be the same, but the corresponding eigenvalues λ_1 and λ_2 can be identical.

To understand why the matrix must be square, remember that a non-square matrix with m rows and n columns transforms an n-dimensional vectors into an m-dimensional vector. Clearly, the m-dimensional output cannot be the n-dimensional input multiplied by a scalar!

Although the number of eigenvectors may vary, all eigenvectors for a matrix are linearly independent. Thus, if an n by n matrix has n eigenvectors, these vectors can be used as a basis (called an *eigenbasis*). If an eigenbasis exists for a matrix, decomposing vectors over this basis simplifies the process of matrix multiplication. To illustrate, imagine we decompose the vector $\mathbf{x}$ over a set of eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$. Decomposing $\mathbf{x}$ means we can find coefficients $a_1, \ldots, a_n$ such that

An n by n matrix with n eigenvectors and n distinct eigenvalues is called a *perfect matrix*. As the name implies, perfect matrices are great to find, but somewhat uncommon.

$$\mathbf{x} = a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n$$

Now let's compute the product $\mathbf{A}\mathbf{x}$. We multiply both sides of the decomposition by $\mathbf{A}$.

$$\mathbf{A}\mathbf{x} = \mathbf{A}\left(a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n\right)$$

We distribute the matrix $\mathbf{A}$ into the sum on the right hand side and note that the constants a_i can be moved in front of the matrix multiplication.

$$\mathbf{A}\mathbf{x} = a_1\mathbf{A}\mathbf{v}_1 + \cdots + a_n\mathbf{A}\mathbf{v}_n$$

Remember that $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are eigenvectors of $\mathbf{A}$, so $\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i$. We can simplify the previous expression to

$$\mathbf{A}\mathbf{x} = a_1\lambda_1\mathbf{v}_1 + \cdots + a_n\lambda_n\mathbf{v}_n$$

We don't need to perform the multiplication at all! Instead, we can scale each eigenvector by the eigenvalue. Multiplying again by the matrix $\mathbf{A}$ multiplies each eigenvector by its eigenvalue.

We use the notation $\mathbf{A}^2$ to denote $\mathbf{AA}$, $\mathbf{A}^3$ for $\mathbf{AAA}$, and $\mathbf{A}^k$ for the product of k matrices $\mathbf{A}$.

$$\mathbf{A}^2\mathbf{x} = a_1\lambda_1^2\mathbf{v}_1 + \cdots + a_n\lambda_n^2\mathbf{v}_n$$

$$\mathbf{A}^k\mathbf{x} = a_1\lambda_1^k\mathbf{v}_1 + \cdots + a_n\lambda_n^k\mathbf{v}_n$$

8.2 Computing Eigenvectors and Eigenvalues

We can use the relationship between matrix multiplication and eigenvalues to find eigenvectors for any matrix. Our computational approach is based on the following theorem.

Theorem. *Given any (random) vector* $\mathbf{b}$*, repeated multiplication by the matrix* $\mathbf{A}$ *will converge to the eigenvector of* $\mathbf{A}$ *with the largest magnitude eigenvalue – provided the largest eigenvalue is unique. Said another way,*

$$\lim_{k\to\infty} \mathbf{A}^k\mathbf{b} = \mathbf{v}_{\max}$$

Proof. We know that the product $\mathbf{Ax}$ can be expressed as a linear combination of the eigenvectors and eigenvalues of $\mathbf{A}$, i.e. $\mathbf{Ax} = a_1\lambda_1\mathbf{v}_1 + \cdots + a_n\lambda_n\mathbf{v}_n$. Thus

$$\lim_{k\to\infty} \mathbf{Ab} = \lim_{k\to\infty} \left(a_1\lambda_1^k\mathbf{v}_1 + \cdots + a_n\lambda_n^k\mathbf{v}_n\right)$$

As k increases, the values λ_i^k grow very large. However, the λ_i to not grow at the same rate. The largest eigenvalue will grow the fastest. At very large values of k, the term associated with the largest eigenvalue will dominate the entire sum, so the result will point in only the direction of the associated eigenvector. Note that convergence to a single eigenvector requires that the largest eigenvalue be distinct. If two eigenvectors have the same (largest) eigenvalue, both terms in the above sum would "blow up" at the same rate. Repeated multiplications by $\mathbf{A}$ would then converge to the sum of the two associated eigenvectors. □

The above theorem allows us to find the eigenvector paired with the largest eigenvalue. While the direction of the eigenvector doesn't change, its magnitude grows as the number of multiplication of $\mathbf{A}$ increases. If convergence is slow, we might need to work with numbers before finding the eigenvector. To avoid numerical difficulties, we renormalize the vector after every multiplication by $\mathbf{A}$. This algorithm is called the Power Iteration method, which proceeds as follows:

1. Choose a random vector $\mathbf{b}_0$. For fastest convergence, it helps to choose a vector close to $\mathbf{v}_{\max}$ if possible. Normalize this vector to product $\hat{\mathbf{b}}_0 = \mathbf{b}_0 / \|\mathbf{b}_0\|$.

2. Compute vector $\mathbf{b}_1 = \mathbf{A}\hat{\mathbf{b}}_0$. Normalize this vector to give $\hat{\mathbf{b}}_1 = \mathbf{b}_1 / \|\mathbf{b}_1\|$.

3. Repeat step 2 to product $\hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3, \ldots, \hat{\mathbf{b}}_k$. Stop when all entries of $\hat{\mathbf{b}}_k$ do not change from the entries in $\hat{\mathbf{b}}_{k-1}$. The vector $\hat{\mathbf{b}}_k$ is an eigenvector of $\mathbf{A}$.

The eigenvector associated with the largest magnitude eigenvalue is called the *leading eigenvector*.

Now that we have the eigenvector $\mathbf{v}_{\max}$, how do we find the associated eigenvalue $\lambda_{\max}$? We know that $\mathbf{v}_{\max}$ is an eigenvector of $\mathbf{A}$, to $\mathbf{A}\mathbf{v}_{\max} = \lambda_{\max}\mathbf{v}_{\max}$. The ith element in $\mathbf{A}\mathbf{v}_{\max}$ should be equal to $\lambda_{\max}$ times the ith element in $\mathbf{v}_{\max}$. However, since we only found a numerical approximation to the $\mathbf{v}_{\max}$, the estimate for $\lambda_{\max}$ from each element in $\mathbf{v}_{\max}$ might differ slightly. To "smooth out" these variations, compute the eigenvalue using the Rayleigh quotient:

$$\lambda_{\max} = \frac{\mathbf{v}_{\max} \cdot \mathbf{A}\mathbf{v}_{\max}}{\mathbf{v}_{\max} \cdot \mathbf{v}_{\max}}$$

To see why the Raleigh quotient works, consider an eigenvector $\mathbf{v}$ for matrix $\mathbf{A}$ with associated eigenvalue λ. Then

$$\frac{\mathbf{v} \cdot \mathbf{A}\mathbf{v}}{\mathbf{v} \cdot \mathbf{v}} = \frac{\mathbf{v} \cdot (\lambda\mathbf{v})}{\mathbf{v} \cdot \mathbf{v}} = \lambda\frac{\mathbf{v} \cdot \mathbf{v}}{\mathbf{v} \cdot \mathbf{v}} = \lambda$$

The dot product in the Rayleigh quotient averages out all of the small discrepancies between our estimate $\mathbf{v}_{\max}$ and the true largest eigenvector. The Rayleigh quotient provides a numerically stable estimate of the largest eigenvalue.

Now that we've found the first eigenvector, how do we find the others? If we start the Power Iteration method over again using the matrix $(\mathbf{A} - \lambda_{\max}\mathbf{I})$ instead of $\mathbf{A}$, the algorithm will converge to the eigenvector associated with the second largest eigenvalue. We can subtract this eigenvalue from $\mathbf{A}$ and repeat to find the third eigenvector, and so on. Proving Power Iteration is able to find subsequent eigenvectors is beyond the scope of this course. However, as we'll see later, finding only the first eigenvector is sufficient for addressing a number of interesting problems.

8.2.1 Eigenvalues and Eigenvectors in MATLAB

The MATLAB function `eig` computes eigenvalues and eigenvectors. The statement `[V,L] = eig(A)` involving an n by n matrix `A` returns two n by n matrices:

- Each column of the matrix `V` is an eigenvector `A`.
- The matrix `L` is a diagonal matrix. The ith entry on the diagonal is the eigenvalue associated with the ith column in `V`.

Remember that any vector that points in the same direction as an eigenvector of a matrix is also an eigenvector of that matrix. If the eigenvectors returned by computational systems like MATLAB are not what you expect, remember that they may be normalized or scaled – but still point along the same direction.

8.3 Applications

Eigenvalue and eigenvectors can be used to solve a number of interesting engineering and data science problems.

8.3.1 Solving Systems of ODEs

Consider the linear system of ODEs

$$\frac{dx_1}{dt} = x_1 + 2x_2$$
$$\frac{dx_2}{dt} = 3x_1 + 2x_2$$

with initial conditions $x_1(0) = 0$ and $x_2(0) = -4$. We can write this system using vectors and matrices as

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x}, \quad \mathbf{x}(0) = \mathbf{x}_0$$

where for the example above

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix}, \quad \mathbf{x}_0 = \begin{pmatrix} 0 \\ -4 \end{pmatrix}$$

If we know the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and eigenvalues $\lambda_1, \ldots, \lambda_n$ for the matrix $\mathbf{A}$, we can compute the solution as

$$\mathbf{x}(t) = c_1\mathbf{v}_1 e^{\lambda_1 t} + c_2\mathbf{v}_2 e^{\lambda_2 t} + \cdots + c_n\mathbf{v}_n e^{\lambda_n t}$$

This solution requires the matrix $\mathbf{A}$ be perfect and therefore have a complete set of eigenvectors.

The scalars $c_1, \ldots, c_n$ are the constants of integration. To find these values, notice what happens to our solution at time $t = 0$:

$$\mathbf{x}(0) = \mathbf{x}_0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n$$

The function $f(t) = e^{\lambda t}$ is an *eigenfunction* of the derivative operator, i.e.

$$\frac{d}{dt}f(t) = \lambda e^{\lambda t} = \lambda f(t)$$

. The solution of a system of linear ODEs is the product of the eigenvectors of $\mathbf{A}$ and the eigenfunctions of $\frac{d\mathbf{x}}{dt}$.

At $t = 0$, the right hand side is a decomposition of the initial conditions $\mathbf{x}_0$. If we collect the eigenvectors as columns of a matrix $\mathbf{V} = (\mathbf{v}_1\mathbf{v}_2 \ldots \mathbf{v}_n)$, we can find the constants $c_1, \ldots, c_n$ by solving the linear system

$$\mathbf{V}\begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} = \mathbf{x}_0$$

Returning to our original example, the matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix}$$

has eigenvalue/eigenvector pairs

$$\lambda_1 = -1, \quad \mathbf{v}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \qquad \text{and} \qquad \lambda_2 = 4, \quad \mathbf{v}_2 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

The integration constants c_1 and c_2 are defined by the system $\mathbf{Vc} = \mathbf{x}_0$, which for this example is

$$\begin{pmatrix} -1 & 2 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 0 \\ -4 \end{pmatrix}$$

Solving the above equations reveals $c_1 = -4/5$ and $c_2 = -8/5$. The final solution to this systems of ODEs is

$$\mathbf{x}(t) = -\frac{8}{5} \begin{pmatrix} -1 \\ 1 \end{pmatrix} e^{-t} - \frac{4}{5} \begin{pmatrix} 2 \\ 3 \end{pmatrix} e^{4t}$$

8.3.2 Stability of Linear ODEs

The eigenvalues of $\mathbf{A}$ are sufficient to tell if the system $\frac{d\mathbf{x}}{dt} = \mathbf{Ax}$ is stable. For a system of linear ODEs to be stable, all eigenvalues of $\mathbf{A}$ must be nonpositive. If the eigenvalues are all negative, each term $e^{\lambda_i t}$ goes to zero at long times, so all variables in the system to go zero. If any of the eigenvalues are zero, the system is still stable (provided all other eigenvalues are negative), but the system will go to a constant value $c_i \mathbf{v}_i$, where $\mathbf{v}_i$ is the eigenvector associated with the zero eigenvalue.

8.3.3 Positive Definite Matrices

A symmetric matrix $\mathbf{A}$ is *positive definite* (p.d.) if $\mathbf{x}^\mathrm{T}\mathbf{Ax} > 0$ for all nonzero vectors $\mathbf{x}$. If a matrix $\mathbf{A}$ satisfies the slightly relaxed requirement that $\mathbf{x}^\mathrm{T}\mathbf{Ax} \geq 0$ for all nonzero $\mathbf{x}$, we say that $\mathbf{A}$ is *positive semi-definite* (p.s.d.).

Remember that a matrix $\mathbf{A}$ is symmetric if $\mathbf{A} = \mathbf{A}^\mathrm{T}$.

Knowing that a matrix is positive (semi-)definite is useful for quadratic programming problems like the Support Vector Machine classifier. The quadratic function $f(\mathbf{x}) = \mathbf{x}^\mathrm{T}\mathbf{Qx}$ is convex if and only if the matrix $\mathbf{Q}$ is positive semi-definite. For optimization problems like quadratic programs, the convexity of the objective function has enormous implications. Convex quadratic programs must only have global optima, making them easy to solve using numerical algorithms.

If $\mathbf{Q}$ is positive definite (rather than just positive semi-definite) then $\mathbf{x}^\mathrm{T}\mathbf{Qx}$ is strictly convex.

Determining if a matrix is positive (semi-)definite can be difficult unless we use eigenvalues. Any matrix with only positive eigenvalues is positive definite, and any matrix with only nonnegative eigenvalues is positive semi-definite. For example, consider the 2×2 identity matrix

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The product $\mathbf{x}^\mathrm{T}\mathbf{Ix}$ is

$$\begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_1^2 + x_2^2$$

Since $x_1^2 + x_2^2$ is greater than zero for all nonzero inputs x_1 and x_2, the matrix $\mathbf{I}$ is positive definite and all its eigenvalues should be positive. Indeed, the eigenvalues for the identity matrix are $\lambda_1 = \lambda_2 = 1$.

As another example, consider the matrix

$$\mathbf{A} = \begin{pmatrix} 1 & -2 \\ -2 & 1 \end{pmatrix}$$

The product $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} = x_1^2 - 4x_1x_2 + x_2^2$, which is not always positive. When $x_1 = x_2 = 1$, we see that $x_1^2 - 4x_1x_2 + x_2^2 = -2$. We know that $\mathbf{A}$ is not positive definite (or even positive semi-definite), so $\mathbf{A}$ should have at least one negative eigenvalue. As expected, the eigenvalues for $\mathbf{A}$ are $\lambda_1 = 3$ and $\lambda_2 = -1$.

8.3.4 Network Centrality

Networks are represented by collections of *nodes* connected by *edges*. When analyzing a network, it is common to ask which node occupies the most important position in the network. For example, which airport would cause the most problems if closed due to weather? In biological networks, the importance or *centrality* of an enzyme is used to prioritize drug targets.

We can quantify the centrality of each node in a network using random walks. We start by choosing a random node in the network. Then we randomly choose one of the edges connected to the node and travel to a new node. This process repeats again and again as we randomly traverse nodes and edges. The centrality of each node is proportional to the number of times we visit the node during the random walk. In the airport analogy, randomly traveling to cities across the country means you will frequently visit major hub airports.

Measuring centrality by random walks is easy to understand but impractical for large networks. It could take millions of steps to repeatedly reach all the nodes in networks with only a few thousand of nodes. In practice, we use eigenvectors to calculate centrality for networks. We begin by constructing an *adjacency matrix* for the network. The adjacency matrix encodes the connections (edges) between the nodes. The adjacency matrix is square with rows and columns corresponding to nodes in the network. The (i,j) entry in the network is set to 1 if there is an edge connecting node i with node j. Otherwise, the entries are zero. Note that we only consider direct connections. If node 1 is connected to node 2 and node 2 is connected to node 3, we do not connect nodes 1 and 3 unless there is a direct edge between them. Also, no node is connected to itself, so the diagonal elements are always zero.

Consider the four node network shown in Figure 8.1. The four node

network has the following adjacency matrix.

$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

To identify the most central node in the network, we find the eigenvector that corresponds to the largest eigenvalue ($\lambda_{\max}$). For the above adjacency matrix, $\lambda_{\max} \approx 2.2$, and the associated eigenvector is

$$\mathbf{v}_{\max} = \begin{pmatrix} 0.52 \\ 0.61 \\ 0.52 \\ 0.28 \end{pmatrix}$$

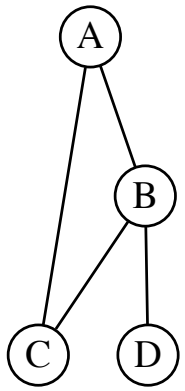

Figure 8.1: Sample network with degenerate flux distributions.

The entries in the eigenvector $\mathbf{v}_{\max}$ are called the *eigencentrality* scores. The largest entry corresponds to the most central node, and the smallest entry is associated with the least central node. We see that node B (entry 2) is most central in Figure 8.1 and node D (entry 4) is least central.

Centrality requires only the leading eigenvector of a network's adjacency matrix. Power Iteration (section 8.2) can find the leading eigenvector efficiently for large networks.

In simple networks like Figure 8.1, the most central node also has the most direct connections. This is not always the case. Eigencentrality considers not only the number of connections but also their importance. Each edge is weighted by the centrality of the nodes it connect. Connections from more central nodes are more important, just as flights between major hub cities usually have the highest passenger volumes. Eigencentrality has numerous applications including web searching. Google uses a modified version of centrality (called PageRank) to determine which results should be displayed first to users.

8.4 Geometric Interpretation of Eigenvalues

Consider a matrix $\mathbf{A} \in \mathbb{R}^2$ with eigenvalues λ_1 and λ_2 and corresponding eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$. Let's take a vector $\mathbf{x}$ and decompose it over the eigenvectors.

$$\mathbf{x} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2$$

We can represent the vector $\mathbf{x}$ by plotting it; however, instead of using the normal Cartesian unit vectors as axes, we will use the eigenvectors. The values of $\mathbf{x}$ along the "eigenaxes" are a_1 and a_2. What happens when we multiply $\mathbf{x}$ and $\mathbf{A}$?

$$\mathbf{A}\mathbf{x} = \mathbf{A}(a_1\mathbf{v}_1 + a_2\mathbf{v}_2) = \lambda_1 a_1\mathbf{v}_1 + \lambda_2 a_2\mathbf{v}_2$$

Visually, multiplying by $\mathbf{A}$ scales the values along the eigenvector axes. The scaling factor along each axis is the corresponding eigenvalue. To quantify the overall effect of multiplying by the matrix $\mathbf{A}$, we can compare the areas swept out by the vector $\mathbf{x}$ before and after multiplication. The area is simply the product of the values along each axis, so the ratio becomes

$$\frac{\text{area}(\mathbf{Ax})}{\text{area}(\mathbf{x})} = \frac{\lambda_1 a_1 \lambda_2 a_2}{a_1 a_2} = \lambda_1 \lambda_2$$

We can repeat the same calculation in three dimensions by looking at the ratio of the volume before and after multiplying by the matrix $\mathbf{A}$.

$$\frac{\text{volume}(\mathbf{Ax})}{\text{volume}(\mathbf{x})} = \frac{\lambda_1 a_1 \lambda_2 a_2 \lambda_3 a_3}{a_1 a_2 a_3} = \lambda_1 \lambda_2 \lambda_3$$

In general, the product of the eigenvalues of a matrix describe to overall effect of multiplying a vector by the matrix. The product of the eigenvalues of a matrix $\mathbf{A}$ is called the *determinant* of $\mathbf{A}$, or $\det(\mathbf{A})$.

$$\det(\mathbf{A}) = \lambda_1 \lambda_2 \cdots \lambda_n$$

If the determinant of a matrix is large, multiplying a vector by the matrix enlarges the volume swept out by the vector. If the determinant is small, the volume contracts.

Remember that the volume we're discussing here is the volume when a vector is plotted using the matrix's eigenvectors as axes.

8.5 Properties of the Determinant

The determinant is a powerful property of a matrix. Determinants can be easily calculated for a matrix and contain useful information about the matrix.

In MATLAB, the function `det` calculates the determinant of a matrix.

Let's say a vector $\mathbf{y} = \mathbf{Ax}$. We know the determinant of the matrix $\mathbf{A}$ is the ratio of the volumes of $\mathbf{x}$ and $\mathbf{y}$.

$$\det(\mathbf{A}) = \frac{\text{volume}(\mathbf{Ax})}{\text{volume}(\mathbf{x})} = \frac{\text{volume}(\mathbf{y})}{\text{volume}(\mathbf{x})}$$

If the inverse of $\mathbf{A}$ exists, we know that $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$. Thus, the determinant of the inverse matrix $\mathbf{A}^{-1}$ is

$$\det(\mathbf{A}^{-1}) = \frac{\text{volume}(\mathbf{A}^{-1}\mathbf{y})}{\text{volume}(\mathbf{y})} = \frac{\text{volume}(\mathbf{x})}{\text{volume}(\mathbf{y})} = \frac{1}{\det(\mathbf{A})}$$

The determinant of $\mathbf{A}^{-1}$ is the inverse of the determinant of $\mathbf{A}$. If the determinant of a matrix is zero, this property indicates there is a problem with the inverse of the matrix.

$$\det(\mathbf{A}) = 0 \Rightarrow \det(\mathbf{A}^{-1}) = \frac{1}{\det(\mathbf{A})} = \frac{1}{0} \rightarrow \text{undefined}$$

Although we won't prove it in this course, **a matrix has an inverse if and only if the determinant of the matrix is nonzero**. The following statements are, in fact, equivalent for a square matrix $\mathbf{A}$:

- $\mathbf{A}$ can be transformed into the identity matrix by elementary row operations.
- The system $\mathbf{Ax} = \mathbf{y}$ is solvable and has a unique solution.
- $\mathbf{A}$ is full rank.
- $\mathbf{A}^{-1}$ exists.
- $\det(\mathbf{A}) \neq 0$.

A matrix with a determinant equal to zero has a geometric interpretation. Remember that the determinant is the product of the eigenvalues. It is the product of the scaling factors of the matrix along each eigenvector. If one of the eigenvalues is zero, we are missing information about how the matrix scales along at least one eigenvector. Our knowledge of the transformation is incomplete, which is why we cannot find a unique solution for the corresponding linear system.

Using the determinant we can concisely state our last field axiom. Recall that for scalars we required a multiplicative inverse exist for any nonzero member of the field, i.e.

$$\text{For all scalars } a \neq 0 \text{ there exists } a^{-1} \text{ such that } aa^{-1} = 1.$$

For vector spaces, we have the following:

$$\text{For all square matrices } \mathbf{A} \text{ where } \det(\mathbf{A}) \neq 0,$$
$$\text{there exists } \mathbf{A}^{-1} \text{ such that } \mathbf{AA}^{-1} = \mathbf{I} = \mathbf{A}^{-1}\mathbf{A}.$$