A Linear Algebra Refresher

Vincent A. Voelz
E-mail: vvoelz@gmail.com
† *Math Bio Boot Camp 2006*
University of California at San Francisco, San Francisco, CA 94143

August 29, 2006

1 Linear Algebra

1.1 Matrices

An $(m \text{ x } n)$ **matrix** is a rectangular array of numbers with m rows and n columns.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & ... & a_{1n} \\ a_{21} & a_{22} & a_{23} & ... & a_{2n} \\ a_{31} & a_{32} & a_{33} & ... & a_{3n} \\ ... & ... & ... & ... & ... \\ a_{m1} & a_{32} & a_{33} & ... & a_{mn} \end{pmatrix}$$

Each entry in the matrix is denoted a_{ij}, where the subscript denotes the i^{th} row and the j^{th} column.

The **transpose** of an $(m \text{ x } n)$ matrix $\mathbf{A}$ is an $(n \text{ x } m)$ matrix $\mathbf{A^T}$ flipped along its diagonal, so the rows are now columns, and vice versa.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad \mathbf{A^T} = \begin{pmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{31} & a_{23} & a_{33} \end{pmatrix}$$

1.2 Vectors

A **vector** is simply a list of numbers, represented as a one-dimensional array, for example $\mathbf{x} = (x_1, x_2, x_3)$. The "standard" representation of a vector is in column form as an (m x 1) matrix:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

The row form of a vector is the **transpose** of the standard column form, a (1 x m) matrix:

$$\mathbf{x^T} = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}$$

The entries of vectors and matrices are sometimes called *components* or *elements*.

Inner and outer products. The **inner product** of (m x 1) vectors $\mathbf{x}$ and $\mathbf{y}$ can be written as $\mathbf{x^T y} = \mathbf{y^T x}$:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$\mathbf{x^T y} = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \sum_i x_i y_i = \begin{pmatrix} y_1 & y_2 & y_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{y^T x}$$

The product of a (1 x m) matrix and an (m x 1) matrix, results in a (1 x 1) matrix, i.e. a *scalar.*

The **outer product** of $\mathbf{x}$ and $\mathbf{y}$ can be written as $\mathbf{xy^T}$:

$$\mathbf{xy^T} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \begin{pmatrix} y_1 & y_2 & y_3 \end{pmatrix} = \begin{pmatrix} x_1y_1 & x_1y_2 & x_1y_3 \\ x_2y_1 & x_2y_2 & x_2y_3 \\ x_3y_1 & x_2y_2 & x_3y_3 \end{pmatrix}$$

In this case, the product of a (m x 1) matrix and a (1 x m) matrix results in an (m x m) matrix.

Note also that $\mathbf{xy^T} \neq \mathbf{yx^T}$:

$$\mathbf{yx^T} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} = \begin{pmatrix} y_1x_1 & y_1x_2 & y_1x_3 \\ y_2x_1 & y_2x_2 & y_2x_3 \\ y_3x_1 & y_2x_2 & y_3x_3 \end{pmatrix} \neq \mathbf{xy^T}$$

So, the outer product is not (in general) commutative.

1.3 Matrix multiplication

The procedure for multiplying two matrices $\mathbf{A}$ and $\mathbf{B}$ is to produce a new matrix $\mathbf{C}$ whose entries c_{ij} are the **inner product** of the i^{th} row of $\mathbf{A}$ and the j^{th} row of $\mathbf{B}$:

$$c_{ij} = \sum_k a_{ik} b_{kj}$$

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}$$

$$\mathbf{AB} = \mathbf{C} = \begin{pmatrix} (a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31}) & ... & ... \\ ... & ... & ... \\ ... & ... & ... \end{pmatrix}$$

Matrix multiplication is not commutative. Because the inner product (also called a *dot product*) demands that the i^{th} row and j^{th} column must have the same number of elements, matrices and vectors can only be multiplied if the left matrix has the same number of rows as the right matrix has columns. For example, an (m x p) matrix can always be multiplied by a (p x n) matrix, to produce an (m x n) matrix, but not the other way around is $m \neq n$.

$$\text{RIGHT!} \begin{pmatrix} a_{11} & a_{12} & a_{13} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix} = 1 \text{ x } 2 \text{ matrix}$$

$$\text{WRONG!} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \end{pmatrix} = ?$$

So, by definition, matrix multiplication is not, in general, *commutative*, i.e. $\mathbf{AB} \neq \mathbf{BA}$.

Transposition chain rule. Although the outer product is not in general commutative,, inspection of the above results shows that $\mathbf{xy^T}$ is the transpose of $\mathbf{yx^T}$, which is true in general due to the following chain rule for vectors and matrices:

$$(\mathbf{AB})^\mathbf{T} = (\mathbf{B^T A^T})$$
$$(\mathbf{ABC})^\mathbf{T} = (\mathbf{C^T B^T A^T})$$
$$(\mathbf{xy^T})^\mathbf{T} = (\mathbf{y^T})^\mathbf{T}\mathbf{x^T} = \mathbf{yx^T}$$

Like the inner and outer product examples, you can demonstrate this for yourself.

1.4 Square Matrices

From now on, we will restrict our discussion to only *square matrices*, those with the same number of rows as columns.

1.4.1 Matrix Inverse

An $(m \text{ x } m)$ matrix $\mathbf{A}$ has an inverse if there exists a matrix $\mathbf{A^{-1}}$ such that:

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

where $\mathbf{I}$ is the **identity matrix**, which is an $(m \text{ x } m)$ matrix with diagonal entries $\mathbf{I}_{ii} = 1$ and all other entries $\mathbf{I}_{ij} = 0$, $i \neq j$.

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

A proof of how to find the inverse of a matrix $\mathbf{A}$ is rather involved, so we won't discuss it here. There are several ways to do it (like Gaussian elimination), and most involve some kind of recursive method to mechanically solve the system of m^2 linear equations that result from the equation $\mathbf{AX} = \mathbf{I}$. We will give the general formula for the matrix inverse, but first we need to introduce determinants.

1.4.2 Determinants

When is a given matrix $\mathbf{A}$ invertible? To answer this question, we need the **determinant**, which is a scalar function of a matrix that turns out to characterize many useful properties of a matrix. The determinant of $\mathbf{A}$ is abbreviated $\det(\mathbf{A})$. It can also be uniquely defined by three main properties:

1. $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$
2. $\det(\mathbf{1}) = 1$
3. $\det(\mathbf{A}) \neq 0$ if and only if $\mathbf{A}$ has an inverse.

   Other properties include:

4. $\det(\mathbf{A^T}) = \det(\mathbf{A})$
5. $\det(\mathbf{A^{-1}}) = 1/\det(\mathbf{A})$

The formula for the determinant can be calculated by a recursive formula:

$$\det(a) = a$$

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

For $m \geq 3$, the $\det(\mathbf{A})$ can be best expressed as a function of the determinants of $(m-1)$ x $(m-1)$ sub-matrices of $\mathbf{A}$. First, as a reference, we pick a particular row i of $\mathbf{A}$ to expand. Then,

$$\det(\mathbf{A}) = \sum_j (-1)^{i+j} a_{ij} \det(\mathbf{M}(i,j))$$

where $\mathbf{M}(i,j)$ is the matrix formed from $\mathbf{A}$ when rows i and column j are removed.

Example. If we pick the first row $i = 1$ to expand a (3 x 3) matrix:

$$\det \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = a \det \begin{pmatrix} e & f \\ h & i \end{pmatrix} - b \det \begin{pmatrix} d & f \\ g & i \end{pmatrix} + c \det \begin{pmatrix} d & e \\ g & h \end{pmatrix}$$

1.4.3 General Formula for Matrix Inverse

The general expression for computing $\mathbf{A^{-1}}$ from $\mathbf{A}$ involves a similar recursive formula:

$$\mathbf{A^{-1}} = \frac{1}{\det(\mathbf{A})}\mathbf{C^T}$$

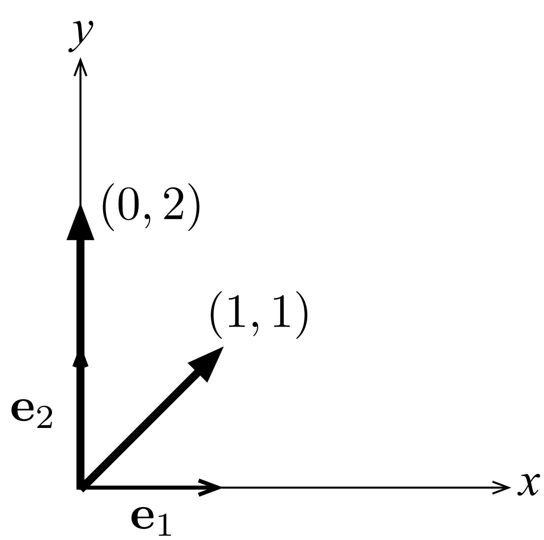
where $\mathbf{C}$ is the *cofactor matrix* defined by $C_{ij} = (-1)^{i+j} \det(\mathbf{M}(i,j))$.

1.5 Coordinate systems

Consider the (2 x 2) matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}$$

The columns of this matrix (lets call them vectors $\mathbf{a_1} = (1, 1)$ and $\mathbf{a_2} = (0, 2)$ live in two dimensions, and we can plot them on an x-y plot:



In this coordinate system, the axes are defined by unit vectors $\mathbf{e}_1 = (1, 0)$ for the x-axis, and $\mathbf{e}_2 = (0, 1)$ for the y-axis.

1.5.1 Basis sets

Here, $\mathbf{e}_1 = (1, 0)$ and $\mathbf{e}_2 = (0, 1)$ form a complete **basis**. A basis is a set of *linearly independent* vectors $\mathbf{v_1}, \mathbf{v_2}, ...\mathbf{v_n}$, such that any vector $\mathbf{w}$ in the space can be expressed as a linear combination of them:

$$\mathbf{w} = c_1\mathbf{v_1} + c_2\mathbf{v_2} + ... + c_n\mathbf{v_n}$$

where c_i are scalar constants. Obviously, any vector in our x-y plot can be expressed as a linear combination of $\mathbf{e}_1$ and $\mathbf{e}_2$:

$$\begin{pmatrix} x \\ y \end{pmatrix} = x\begin{pmatrix} 1 \\ 0 \end{pmatrix} + y\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

For this 2-dimensional space, at least 2 linearly independent basis vectors are needed to fully *span* the space. For an n-dimensional space, at least n linearly independent basis vectors are needed.

In this example, $\mathbf{e}_1$ and $\mathbf{e}_2$ are also **orthonormal**. A set of basis vectors $\mathbf{e}_i$ are orthonormal if:

1. $\mathbf{e_i^T e_i} = 1$ for all i
2. $\mathbf{e_i^T e_j} = 0$ for all $i \neq j$

This means that all of the basis vectors are of unit length and are all mutually perpendicular.

1.5.2 Rigid coordinate system changes

For rigid coordinate changes from one orthonormal basis to another, we can project vectors in one basis onto another basis.

Projection. The operation of projecting one vector $\mathbf{a}$ onto another $\mathbf{b}$ is defined as

$$\mathbf{a^T}\frac{\mathbf{b}}{|\mathbf{b}|}$$

where $|\mathbf{b}| = \sqrt{\mathbf{b^T b}}$ is the *length* of $\mathbf{b}$. Thus, $\mathbf{b}/|\mathbf{b}|$ is a vector of unit length.

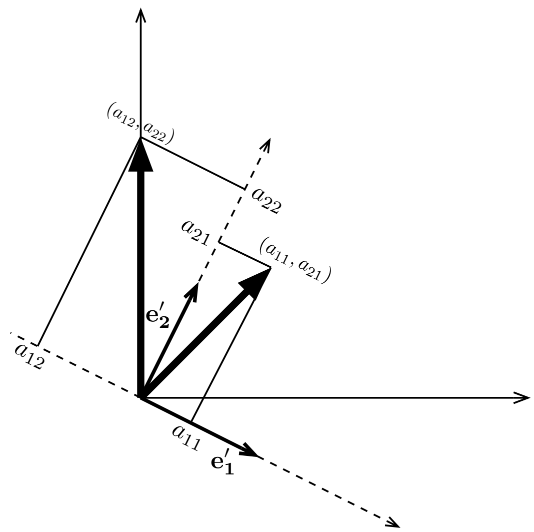
The x-coordinate of $\mathbf{a_1}$ is found by projection onto $\mathbf{e}_1$ (which is already of unit length):

$$\mathbf{a_1^T e_1} = \begin{pmatrix} 1 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1$$

and the y-coordinate of $\mathbf{a_1}$ is found by projection onto $\mathbf{e}_2$:

$$\mathbf{a_1^T e_2} = \begin{pmatrix} 1 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = 1$$

Now suppose we want to find the coordinates of $\mathbf{a_1}$ and $\mathbf{a_2}$ along a different set axes that have been rigidly rotated or reflected, defined by $\mathbf{e}'_1$ and $\mathbf{e}'_2$. The procedure is the same. The $\mathbf{e}'_1$-$\mathbf{e}'_2$-coordinates of $\mathbf{a_1}$ and $\mathbf{a_2}$ are found by projection:



$$\mathbf{a'_1} = \begin{pmatrix} \mathbf{a_1^T e'_1} \\ \mathbf{a_1^T e'_2} \end{pmatrix} = \begin{pmatrix} \mathbf{e'_1}^{\mathbf{T}} \mathbf{a_1} \\ \mathbf{e'_2}^{\mathbf{T}} \mathbf{a_1} \end{pmatrix}$$

$$\mathbf{a'_2} = \begin{pmatrix} \mathbf{a_2^T e'_1} \\ \mathbf{a_2^T e'_2} \end{pmatrix} = \begin{pmatrix} \mathbf{e'_1}^{\mathbf{T}} \mathbf{a_2} \\ \mathbf{e'_2}^{\mathbf{T}} \mathbf{a_2} \end{pmatrix}$$

The matrix $\mathbf{A'}$ which has coordinate-transformed column vectors $\mathbf{a'_1}$ and $\mathbf{a'_2}$, can thus be written

$$\mathbf{A'} = \begin{pmatrix} \mathbf{e'_1}^{\mathbf{T}} \\ \mathbf{e'_2}^{\mathbf{T}} \end{pmatrix} \begin{pmatrix} \mathbf{a_1} & \mathbf{a_2} \end{pmatrix} = \mathbf{U^T A}$$

where $\mathbf{U} = \begin{pmatrix} \mathbf{e'_1} & \mathbf{e'_2} \end{pmatrix}$ is the matrix whose columns are the new basis vectors. We can transform $\mathbf{A'}$ back into $\mathbf{A}$, in our original coordinate system, by finding the inverse of $\mathbf{U^T}$:

$$\mathbf{A} = (\mathbf{U^T})^{-1}\mathbf{A'}$$

Now we use a very important fact to remember: *If* $\mathbf{U}$ *has orthonormal columns (or rows),* $\mathbf{U^{-1}} = \mathbf{U^T}$. This is true for our matrix $\mathbf{U}$. So the transformation back to the original coordinate system is simply:

$$\mathbf{A} = \mathbf{UA'}$$

You can verify that $\mathbf{U^{-1}} = \mathbf{U^T}$ by using the properties of orthonormal vectors defined above, i.e.:

$$\begin{pmatrix} \mathbf{e'_1}^{\mathbf{T}} \\ \mathbf{e'_2}^{\mathbf{T}} \end{pmatrix} \begin{pmatrix} \mathbf{e'_1} & \mathbf{e'_2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

1.5.3 Non-rigid coordinate transformations

What happens when we want to to change to a coordinate system where the basis is *non-orthogonal*? In order to to find the new components $\mathbf{w}' = (x', y')$ of a vector $\mathbf{w} = (x, y)$, in the $\mathbf{e_1'}$-$\mathbf{e_2'}$ coordinate system, we need to satisfy

$$\mathbf{w} = \begin{pmatrix} x \\ y \end{pmatrix} = x' \begin{pmatrix} \mathbf{e_1'} \end{pmatrix} + y' \begin{pmatrix} \mathbf{e_2'} \end{pmatrix} = \begin{pmatrix} \mathbf{e_1'} & \mathbf{e_2'} \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{U}\mathbf{w}'$$

which we can solve by

$$\mathbf{w}' = \mathbf{U^{-1}}\mathbf{w}$$

This is the recipe to translate to the new basis set coordinates. Note that because the column vectors of $\mathbf{U}$ are not orthonormal, we must use $\mathbf{U^{-1}}$ rather than $\mathbf{U^T}$. This is because in a non-orthogonal basis, simple projection will not suffice. This is illustrated in the next two figures.

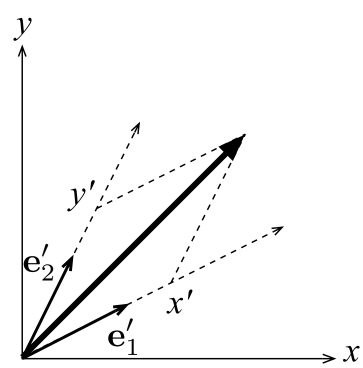


Figure 1: A vector can be specified in a non-orthogonal coordinate system as a linear combination of basis vectors.

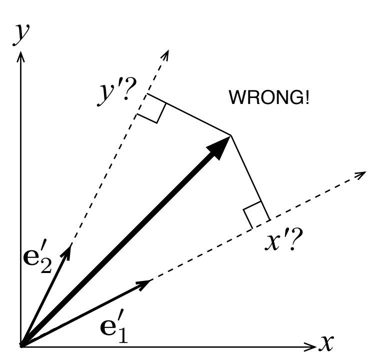


Figure 2: Coordinate changes by simple projection does not work for non-orthogonal coordinate systems.

Coordinate transformations like the kind shown above can be reversible only if $\mathbf{e}_1'$ and $\mathbf{e}_2'$ form a complete basis. An example of an incomplete basis set would be if $\mathbf{e}_1' = \mathbf{e}_2'$. In this case, the columns of $\mathbf{A}$ are transformed to the same vector space, so we lose some of the information about how to specify the coordinates of each column in 2-dimensional space. For this same reason, a matrix is only invertible if its columns (or rows) are linearly independent, providing a complete basis set.

1.6 Eigenvalues and Eigenvectors

As far as basis sets go, there is one set of basis vectors that is particularly useful: the so-called **eigenbasis**. For every matrix $\mathbf{A}$ there is a set of (non-trivial) **eigenvectors** $\mathbf{e_i}$ and scalar **eigenvalues** λ_i satisfying the *characteristic equation*:

$$\mathbf{A}\mathbf{e_i} = \lambda_i \mathbf{e_i}$$

Each eigenvector $\mathbf{e_i}$ has the special property that multiplying it by $\mathbf{A}$ results in the same eigenvector, scaled by a constant λ_i. The German word *eigen* means "own" or "characteristic", descriptive of the fact that every matrix has a unique set of eigenvalues and eigenvectors that characterizes it.

Eigenvectors are the basis in which a matrix is diagonal. What happens when we change coordinates to the eigenbasis? For ease, let's again define $\mathbf{U}$ to be the matrix consisting of column eigenvectors:

$$\mathbf{U} = \begin{pmatrix} \mathbf{e_1} & \mathbf{e_2} & ... & \mathbf{e_n} \end{pmatrix}$$

From the characteristic equation, we have that

$$\mathbf{AU} = \begin{pmatrix} \lambda_1\mathbf{e_1} & \lambda_2\mathbf{e_2} & ... & \lambda_n\mathbf{e_n} \end{pmatrix} = \begin{pmatrix} \mathbf{e_1} & \mathbf{e_2} & ... & \mathbf{e_n} \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & ... & \\ & & & \lambda_n \end{pmatrix} = \mathbf{U\Lambda}$$

$\mathbf{\Lambda}$ is a *diagonal* matrix with eigenvalues as entries. If we transform $\mathbf{AU}$ into eigenbasis coordinates, we find that:

$$\mathbf{U^{-1}}(\mathbf{AU}) = \mathbf{U^{-1}U\Lambda} = \mathbf{\Lambda}$$

and rearranging,

$$\mathbf{A} = \mathbf{U\Lambda U^{-1}}$$

This expression has a very nice geometrical interpretation, where it is easy to understand transformations by matrix multiplication. Suppose we multiply a vector $\mathbf{x}$ by the matrix $\mathbf{A}$:

$$\mathbf{Ax} = \mathbf{U}(\mathbf{\Lambda}(\mathbf{U^{-1}x}))$$

In this form, we can describe the transformation in three steps:

1. $\mathbf{U^{-1}}$ brings $\mathbf{x}$ into the eigenbasis coordinate system
2. $\mathbf{\Lambda}$ stretches each coordinate by its correpsonding eigenvalue
3. $\mathbf{U}$ brings the result back into our original coordinate system

The figure below illustrates this process:

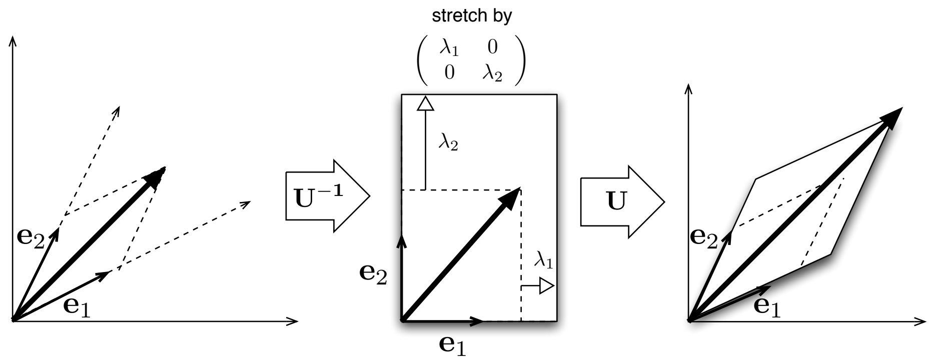


Figure 3: The action of a matrix $\mathbf{A} = \mathbf{U\Lambda U^{-1}}$ is equivalent to stretching the coordinates along its eigenbases.

1.6.1 More on matrix determinants

Now that we have the eigenbasis decomposition, we derive some handy facts about determinants (recall the earlier section on determinant properties):

$$\begin{aligned} \det(\mathbf{A}) &= \det(\mathbf{U\Lambda U^{-1}}) && (1)\\ &= \det(\mathbf{U})\det(\mathbf{\Lambda})\det(\mathbf{U^{-1}}) && (2)\\ &= \det(\mathbf{\Lambda})\det(\mathbf{U})\det(\mathbf{U^{-1}}) && (3)\\ &= \det(\mathbf{\Lambda})\det(\mathbf{UU^{-1}}) && (4)\\ &= \det(\mathbf{\Lambda}) && (5)\\ \det(\mathbf{A}) &= \lambda_1\lambda_2...\lambda_n && (6) \end{aligned}$$

This is an invariant property of any matrix $\mathbf{A}$. The geometric interpretation of the determinant can be thought of as a scaling factor defined by how much the n-dimensional volume defined by the eigenbasis is stretched.

1.6.2 How to solve for the eigenvalues and eigenvectors of a matrix

There are many sophisticated numerical algorithms for *diagonalizing* a matrix, that is, finding the complete set of eigenvalues and eigenvectors. For extremely large data sets, it if often desirable to only calculate a subset of eigenvectors corresponding to the the largest-magnitude eigenvalues. We won't go into these techniques, but the literature is vast. (MATLAB's help pages and references [1], [2] and [3] are a good place to start.)

For simple 2x2 or 3x3 matrices, however, eigenvalues and eigenvectors can be computed by hand. The first step is to find the eigenvalues using the *characteristic equation*:

$$\begin{aligned} \mathbf{Ae} &= \lambda\mathbf{e} && (7)\\ (\mathbf{A} - \lambda\mathbf{I})\mathbf{e} &= \mathbf{0} && (8)\\ & && (9) \end{aligned}$$

Using the properties of determinants, it follows that

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

For an n x n matrix, this expression turns out to be, in general, a n^{th}-degree polynomial that can be solved for n different values of λ. This is called the *characteristic polynomial.* (Note: *It is possible that the solutions to this polynomial are complex. If this is the case, the complex eigenvalues must come in conjugate pairs, as demanded by the characteristic polynomial. The geometrical interpretation is that in addition to stretching along the eigenbases, there is rotation as well.*)

Once the eigenvalues are known, each can be inserted into the characteristic equation to solve for the corresponding eigenvector.

Example. Find the eigenvalues and eigenvectors of $\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}$.

$$\det(\mathbf{A}) = (1 - \lambda)(2 - \lambda) = 0$$

So the two eigenvalues of $\mathbf{A}$ must be $\lambda_1 = 1$ and $\lambda_2 = 2$. For the eigenvector belonging to λ_1, we solve:

$$\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 1 \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

which leads to a class of solutions lying on the line $x + y = 0$. To constrain the eigenvector to unit length, we choose $\mathbf{e_1} = (1/\sqrt{2}, -1/\sqrt{2})$. For the eigenvector belonging to λ_2, we solve:

$$\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 2 \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

This leads to a class of solultions lying on the line $x = 0$. Again, to constrain the eigenvector to unit length, we choose $\mathbf{e_2} = (0, 1)$.

Exercise. Check for yourself that these eigenvalues and eigenvectors satisfy the characteristic equation by working it out with pencil and paper. Are $\mathbf{e_1}$ and $\mathbf{e_2}$ orthonormal?

1.7 Symmetric matrices

Symmetric matrices have some properties that we will readily exploit in the section on data analysis. Keep the folowing theorems in mind for the next section.

1.7.1 Symmetric matrices have real eigenvalues

Proof. Consider a symmetric matrix $\mathbf{A}$ that has all real (non-complex) entries. The characteristic equation says:

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

Taking the *complex conjugate* of both sides, we get

$$\mathbf{A}\bar{\mathbf{e}} = \bar{\lambda}\bar{\mathbf{e}}$$

Next, we multiply the first equation by $\bar{\mathbf{e}}^{\mathbf{T}}$, and the second by $\mathbf{e}^{\mathbf{T}}$. Then subtract the transpose of the second from the first. Using the transposition chain rule, and using the fact that $\mathbf{A}$ is symmetric, the left side of this equation is zero.

$$\bar{\mathbf{e}}^{\mathbf{T}}\mathbf{A}\mathbf{e} - (\mathbf{e}^{\mathbf{T}}\mathbf{A}\bar{\mathbf{e}})^{\mathbf{T}} = \lambda\bar{\mathbf{e}}^{\mathbf{T}}\mathbf{e} - (\bar{\lambda}\mathbf{e}^{\mathbf{T}}\bar{\mathbf{e}})^{\mathbf{T}}$$

$$0 = (\lambda - \bar{\lambda})\bar{\mathbf{e}}^{\mathbf{T}}\mathbf{e}$$

$\mathbf{e}^{\mathbf{T}}\bar{\mathbf{e}}$ must be real and non-zero. Thus, any eigenvalue λ must be equal to its own complex conjugate, and therefore must be *real*.

1.7.2 Symmetric matrices have orthonormal eigenvectors

Proof. Again we consider a symmetric matrix $\mathbf{A}$ that has all real (non-complex) entries and start with the characteristic equation for two different eigenvectors $\mathbf{e_i}$ and $\mathbf{e_j}, i \neq j$:

$$\mathbf{A}\mathbf{e_i} = \lambda_i \mathbf{e_i}$$

$$\mathbf{A}\mathbf{e_j} = \lambda_j \mathbf{e_j}$$

Multiply the first equation by $\mathbf{e_j^T}$ and the second by $\mathbf{e_i^T}$, and subtract the two equations. By similar reasoning to the last theorem, we arrive at the equation:

$$0 = (\lambda_i - \lambda_j)\mathbf{e_i^T}\mathbf{e_j}$$

This is only true if $\mathbf{e_i^T}\mathbf{e_i} = 1$, and $\mathbf{e_i^T}\mathbf{e_j} = 0$. In otherwords, the $\mathbf{e_i}$ must be orthnormal.

2 Singular Value Decomposition (SVD)

It should be briefly mentioned that the eigenbasis coordinate change we have outlined above can be generalized to non-square matrices, and is called the *singular value decomposition*, or SVD. The basic idea is that every M x N matrix $\mathbf{A}$, $(M > N)$, can be decomposed into the product:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}}$$

where $\mathbf{U}$ is an M x M matrix, $\mathbf{V}$ is an N x N matrix, both with orthogonal column vectors. $\mathbf{\Sigma}$ is an M x N matrix that only has entries along the diagonal:

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1 & & \\ & \cdots & \\ & & \sigma_N \\ & \mathbf{0} & \end{pmatrix}$$

The geometrical interpretation of the SVD is that multiplying a N-dimensional vector $\mathbf{x}$ by an M x N matrix $\mathbf{A}$ is equivalent to taking the vector to a N-dimensional subspace, and then transforming again to a larger M-dimensional subspace.

In practice, singular vales can be found by zero-padding $\mathbf{A}$ to be a square matrix, and finding the (incomplete) set of eigenvectors and eigenvalues.

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[2] Lloyd N. Trefethen and III David Bau. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.

[3] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical recipes in C: The art of scientific computing. Chapter 12*. Cambridge University Press., Cambridge, 2nd edition, 1989.