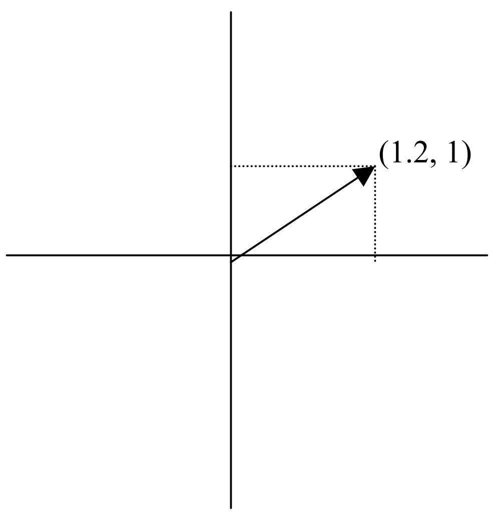
Vectors, Matrices, and Linear Algebra 101

First, vectors. Rigorously, vectors are rather abstract objects and defined by their so-called algebraic properties. Most of you have learned drawing pictures with vectors and how to add two vectors (remember parallelograms) or how to multiply a vector with a "scalar" (i.e., a number) to stretch or shrink the vector. However, for our purposes we will simply say vectors are equivalent to a *n*-tuple of numbers, like (*x,y,z*). These *n*-tuple numbers correspond to the "coordinates" of the vectors. We can think of the coordinates just like how we normally think of coordinates, a set of numbers that designate a point in space. In the case of vectors, we imagine the vectors to be an arrow with the tail fixed at the origin and the head at the point of the coordinate. For each *n*-tuple, there is a vector corresponding to it, so we will equate the vector with that n-tuple numbers. Conversely, for each vector we may have a method to compute its coordinates (e.g., draw a perpendicular line and read off the corresponding number). (We will study coordinates in more detail in a later lecture.)



As mentioned, we can equate a vector with a *n*-tuple of numbers that we list like this *(x, y, z).* We will call this a row vector. We can also try to list it "column-wise" like this:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Suppose we use the variable **v** to designate the vector *(x, y, z)* and **w** to designate the column vector above. Then we write $\mathbf{v}^{\mathbf{t}} = \mathbf{w}$ and read "**v** transpose equals **w**", where transpose means to exchange rows and columns. Sometimes the notation **v'** is used instead of $\mathbf{v}^{\mathbf{t}}$ to denote transpose. So a vector can be denoted by either a row of numbers or a column of numbers. Typically, if nothing is explicitly said, it is convention to consider vectors a column of numbers rather than a row of numbers. To write a column vector "inline", I will often use notations like $\mathbf{w} = (x, y, z)^t$.

Now matrices. An m by n matrix is a collection of mn numbers arranged in a rectangular array with m rows and n columns like this:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \\ \vdots & & & \\ a_{m1} & \cdots & & a_{nm} \end{pmatrix}$$

The elements of the matrix are scalars (i.e., numbers) and indexed with a i,j subscript. I will interchangeably use $a(i,j)$ and a_{ij} to denote the element in the ith row and jth column. If we aren't too picky, a m by n matrix can be seen as a collection of n column vectors each of length m. (Conversely, m row vectors each of length n.) We can also consider a vector of length m as a m by 1 column matrix (or 1 by m row matrix).

The sum of two matrices is defined as the direct sum of elements. For example,

$$\begin{pmatrix} 2 & 1 & 0 \\ -1 & 2 & 3 \\ 1 & 1 & 4 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 3 & 3 \\ 1 & 1 & 5 \end{pmatrix}$$

By this definition you can see the addition operation is defined only if you have same shape matrices (that is, both of them are m by n). Subtraction is similarly defined.

We will often denote entire matrix by a single symbol (variable) like,

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

We can also take transposes of matrices. For example, for the matrix **A** above,

$$\mathbf{A^t} = \begin{pmatrix} 2 & 0 & 2 \\ 3 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix}$$

Note, how what used to be columns of **A** became rows of $\mathbf{A^t}$.
A matrix **A** can be multiplied by a scalar, say c. In this case the product is defined as by multiplying the scalar to every element of the matrix. For example, for the matrix **A** above,

$$2\mathbf{A} = \begin{pmatrix} 4 & 6 & 0 \\ 0 & 2 & 4 \\ 4 & 4 & 2 \end{pmatrix}$$

More complicated is matrix multiplication. We will first define vector multiplication between a row vector and a column vector. The product is defined only if they are the same size. Suppose we have the following row and column vectors

$$\mathbf{x} = (x_1, x_2, \cdots, x_n) \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Then we define:

$$\mathbf{xy} = (x_1, x_2, \cdots, x_n) \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n = \sum_{i=1}^{n} x_i y_i$$

Note that we have taken a pair of *n* numbers and turned it into a single number. The vector product defined in this manner is often called the "dot product" or the "scalar product" (because it creates a scalar from two vectors) or sometimes the "usual inner product".

Given this definition for vector products and viewing a *n* by *m* matrix as a collection *m* column vector of size *n* you see how a matrix product can be defined. The product of *n* by *m* matrix, **A**, and *m* by *k* matrix, **B**, is a *n* by *k* matrix, **C**. The ith row and jth column element of **C**, that is C(i,j), is defined as the vector product of ith row vector from **A** and jth column vector from **B**. For example,

$$\begin{pmatrix} 2 & -3 \\ 1 & 3 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 3 \\ 2 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2\cdot 1 + -3\cdot 2 & 2\cdot 0 + -3\cdot 1 & 2\cdot 3 + -3\cdot 0 \\ 1\cdot 1 + 3\cdot 2 & 1\cdot 0 + 3\cdot 1 & 1\cdot 3 + 3\cdot 0 \\ -1\cdot 1 + 0\cdot 2 & -1\cdot 0 + 0\cdot 1 & -1\cdot 3 + 0\cdot 0 \end{pmatrix} = \begin{pmatrix} -4 & -3 & 6 \\ 7 & 3 & 3 \\ -1 & 0 & -3 \end{pmatrix}$$

Here we multiplied a 2 by 3 matrix with a 3 by 2 matrix and obtained a 3 by 3 matrix. Note that since we are multiplying the row vectors of **A** with the column vectors of **B**, the number of columns in **A** must equal the number of rows in **B**, otherwise the multiplication is not defined.

Except for special cases the matrix product **AB** does not equal **BA**. (Such algebraic objects are called non-commutative or non-Abelian algebra.). Division of one matrix by another, **A/B**, is not defined also except for special cases. The special cases are when the inverse of the matrix **B** (which I will denote by $\mathbf{B}^{-1}$) is defined. We will see this below.

Square matrices are often of interest. That is, matrices that are *n* by *n*. This is because if you multiply a *n* by *n* matrix with a vector of size *n*, you end up with a vector of size *n* again. For example,

$$\begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix}\begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 2\cdot 3 + 1\cdot 1 \\ 3\cdot 3 + 2\cdot 1 \end{pmatrix} = \begin{pmatrix} 7 \\ 11 \end{pmatrix}$$

This is nice because we can imagine that the matrix "transforms" *n* numbers to another set of *n* numbers. I won't go into too much detail on this "transformation" view of matrices but you should look up a good linear algebra book. However, we will deal a lot with square matrices. For example, transition matrices for finite state Markov chain are square matrices, and when multiplied to a vector of marginal probabilities of each state, it transforms it into a new set of marginal probabilities.

A special type of square matrix is a matrix with "1" in the diagonals and "0" elsewhere. Like,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

These kind of matrices are called identity matrix and often denoted by **I**. Verify that **IA** = **AI** = **A** for any square matrix **A**. In terms of Markov Processes, an identity matrix would represent a model in which nothing is allowed to change.

The matrix notation was originally introduced as a shorthand for writing linear equations. For example, if we have a system of equations,

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & +\cdots+ & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & +\cdots+ & a_{2n}x_n & = & b_2 \\ \vdots & & & & & \vdots & \\ \vdots & & & & & \vdots & \\ a_{m1}x_1 & + & a_{m2}x_2 & +\cdots+ & a_{mn}x_n & = & b_m \end{array}$$

We can write this as $\mathbf{Ax} = \mathbf{b}$, where **A** is a matrix, and **x** and **b** are vectors. (You should write this out.) There are many different algorithms for solving this kind of problem, which we will not cover in this class.

We talked briefly about inverses above. Now we define them.

A matrix **B** is inverse of **A** if and only if **AB** = **BA** = **I** (identity matrix). Notice, that for us to multiply from right or left we need both **A** and **B** to be a square matrix. For any given square matrix **A**, an inverse may not exist. If it exists we denote it by $\mathbf{A}^{-1}$ and we have $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$. A great deal of linear algebra is devoted to defining conditions under which matrix inverses exist. We won't need to go into such details in this class, but the important thing is to remember the computational definition of the matrix inverse.

If a matrix is square, we can continue to multiply it by itself. (Why not for non-square matrices?) We denote this by the usual "power" notation. For example, $\mathbf{A} \cdot \mathbf{A} = \mathbf{A}^{\mathbf{2}}$ or $\mathbf{A} \cdot \mathbf{A} \cdot \mathbf{A} = \mathbf{A}^{\mathbf{3}}$. Sometimes, when we take powers of matrices continuously, the values may converge to some fixed matrix. That is, $\mathbf{A}^{\mathbf{n}} \to \mathbf{B}, n \to \infty$. This often happens with transition matrices for Markov Processes. Finally, sometimes we will need to take matrix powers of the number *e*. That is, compute $e^{\mathbf{A}}$. Some of you may know that *x* powers of the number *e* can be defined as the following infinite sum.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

We can do an analogous series for matrix powers and define:

$$e^{\mathbf{A}} = \mathbf{I} + \mathbf{A} + \frac{\mathbf{A}^2}{\mathbf{2!}} + \frac{\mathbf{A}^3}{\mathbf{3!}} + \cdots = \sum_{n=0}^{\infty} \frac{\mathbf{A}^n}{n!}$$

(where $\mathbf{A}^{\mathbf{0}}$ is defined to be the identity matrix). We will take for granted that this series converges. Since this is sum of matrices, it will be another matrix. So, taking *e* to the power of a matrix is another matrix. These kinds of powers will be used in defining continuous time Markov Processes.