Basic Linear Algebra Operations and Definitions

Here I want to introduce some notation that I use (which may be non-standard) and some ideas which are basic and well known, but helpful to keep in mind when reading the ML literature. The vector $\mathbf{u}$ with n elements is either a horizontal or vertical collection of n numbers. In the former $\mathbf{u}$ is *row* vector (and I note its dimension $\mathbf{u} \equiv (1 \times n)$) and the latter it is a *column* vector ($\mathbf{u} \equiv (n \times 1)$). By default all vectors are column vectors. A column vector $\mathbf{u}$ can be written as a row by tranposing (ie $\mathbf{u}^T$). We define the two vector-vector products:

Vector Products

The *dot product* of two vectors $\mathbf{u}$ and $\mathbf{v}$ is the scalar

$$\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} = \sum \mathbf{u}_i \mathbf{v}_i \tag{1}$$

The *vector product* of two vectors $\mathbf{u}$ and $\mathbf{v}$ is the matrix

$$\mathbf{u}\mathbf{v}^T = \begin{pmatrix} u_1 v_1 & u_1 v_2 & \dots & u_1 v_n \\ u_2 v_1 & u_2 v_2 & \dots & u_2 v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_n v_n & u_1 v_n & s & u_n v_n \end{pmatrix} \tag{2}$$

The vectors $\mathbf{u}$ A $m \times n$ matrix A, where

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{pmatrix} \tag{3}$$

is often decomposed into its m rows

$$A = \begin{pmatrix} \mathtt{r}(A)_1 \\ \mathtt{r}(A)_2 \\ \vdots \\ \mathtt{r}(A)_m \end{pmatrix} \tag{4}$$

or n columns

$$A = \begin{pmatrix} \mathtt{c}(A)_1 & \mathtt{c}(A)_2 & \dots & \mathtt{c}(A)_n \end{pmatrix} \tag{5}$$

so as to better represent matrix operations. With this notation we view the matrix product AB where $A \equiv (m \times n)$ and $B \equiv (n \times m)$ as a matrix of dot products

$$AB = \begin{pmatrix} \mathtt{r}(A)_1 \cdot \mathtt{c}(B)_1 & \mathtt{r}(A)_1 \cdot \mathtt{c}(B)_2 & \cdots & \mathtt{r}(A)_1 \cdot \mathtt{c}(B)_m \\ \mathtt{r}(A)_2 \cdot \mathtt{c}(B)_1 & \mathtt{r}(A)_2 \cdot \mathtt{c}(B)_2 & \cdots & \mathtt{r}(A)_2 \cdot \mathtt{c}(B)_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathtt{r}(A)_m \cdot \mathtt{c}(B)_m & \mathtt{r}(A)_1 \cdot \mathtt{c}(B)_m & \cdots & \mathtt{r}(A)_m \cdot \mathtt{c}(B)_m \end{pmatrix} \tag{6}$$

The matrix-vector product $A\mathbf{x}$ is the weighted sum of the columns of A ie

$$A\mathbf{x} = \sum x_i \mathtt{c}(A)_i \tag{7}$$

The set of all possible combinations of the columns defines the *column space* of A. This idea is couple to the idea of a *vector space*. More specifically, the *vector space* generated by a set of n vectors

$\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ is the set of all possible linear combinations of the $\mathbf{u}_i$. If we combine these vectors to form the columns of a matrix A, the we get the linear combination from the above matrix-vector product $A\mathbf{x}$. The column space of a matrix A then is the *vector space* generated by its columns. This *vector space*, which is the *column space* of A is often also called the *range* of A, ie

$$\text{range}\,(A) = \{\mathbf{u} \in \mathcal{R}^n \text{ where there exists } \mathbf{x} \text{ with } A\mathbf{x} = \mathbf{u}\} \tag{8}$$

since it defines which vectors can be produced by summing the columns in A. The *null space* is a particular set of non-zero weights $\mathbf{x}$ which produce the zero vector, ie

$$\text{null}\,(A) = \{\text{all } \mathbf{u} \text{ where } A\mathbf{u} = 0\} \tag{9}$$

If such an $\mathbf{u}$ exists, then one column can be written as a linear combination of at least one of the others. This has significant implications - see below. If no such combination exists, then the columns are *linearly independant* - one cannot be written as a combination of the others. This concept is again common to sets of vectors - ie a set of vectors is linearly independant if no linear combination of the vectors produces a zero vector. Spaces which are *not* linearly independant include redundant vectors.

The *column rank* of matrix A is the largest number of independant columns in A. If all columns are independant, the matrix has *full rank*. It is denoted colrank (A). The *row rank* is defined similarly, but is used much less frequently - the *rank* of a matrix rank (A) usually refers to the column rank. Some examples would be good.

Consider a matrix A and some set of vectors $\begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_n\end{pmatrix}$. For every possible combination of the columns of A (ie every $\mathbf{v} = A\mathbf{u}$), it may be possible to choose some combination of the vectors $\mathbf{u}_i$ that produces the same $\mathbf{v}$, and vica versa. In this sense, the set of vectors $\begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_n\end{pmatrix}$ can substitute for the matrix A, since they can both generate the exact same output (ie they have the same range). If we want to do this substitution, we shouldnt choose any redundant vectors in the $\mathbf{u}_i$ - we should only use *linearly independant* vectors. Such a set of vectors is said to form a *basis* for A.

Elementary Matrices and Elementary Matrix operations

Here a section on elementary matrices, and elementary operations.

Consider two types of elementary matrices:

- Diagonal or Scaling matrices
- Permutation matrices

Consider a matrix A. Multiplication by an elementary matrix E transforms either rows or columns of its operand depending on whether we pre or post multiply ie EA or AE.

Operation	Effect
pre-multiply (ie EA)	row transformation
post-multiply (ie AE)	column transformation

Non-Singular Matrices

Consider the inverse problem. Given a set of vectors $\begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_n\end{pmatrix}$, or a matrix A, is it possible to compute out which combination produces a particular vector $\mathbf{v}$? That is, can we solve $A\mathbf{x} = \mathbf{v}$?

Clearly, $\mathbf{v}$ should be in column space of A, since the column space contains all possible linear combinations of the columns in A. Under what conditions can we find this $\mathbf{x}$?

The square matrix A is *non-singular* if a unique solution $\mathbf{x}$ to $A\mathbf{x} = \mathbf{v}$ exists. The solution is denoted $\mathbf{x} = A^{-1}\mathbf{v}$. If A^{-1} exists, then it is unique. To see this, assume instead that $A^{-1} = B_1$ and $A^{-1} = B_2$ where $B_1 \neq B_2$ - that is, there are two different matrices which are inverses of A. By definition

$$AB_1 = I = B_1A$$
$$AB_2 = I = B_2A$$

and by multiplying the second equation by B_1 we see

$$(B_1A)\, B_2 = B_1 I \tag{10}$$

where the bracketed term is I and we have $B_2 = B_1$. The previous definitions of *rank*, *range* and *null space* are usefull here. When determing if A^{-1} exists, remember

$$A^{-1} \iff \text{null}\,(A) = \{0\}$$
$$A^{-1} \iff \text{range}\,(A) = \mathcal{R}^n$$
$$A^{-1} \iff \text{rank}\,(A) = n$$

Some important identities that are used regularly are

$$(AB)^{-1} = B^T A^T$$
$$\left(A^T\right)^{-1} = \left(A^{-1}\right)^T$$

To check the first, let $M = AB$, and note that

$$B^{-1}A^{-1}M = B^{-1}\left(A^{-1}A\right)B = I$$

so that $M^{-1} = B^{-1}A^{-1} = (AB)^{-1}$. For the second, consider $\left(A^{-1}\right)^T A^T = \left(AA^{-1}\right)^T = I$.

Some examples would be good here, as well as elementary matrix operations section. Also, a section on why we do not computing inverse matrices directly. Conditioning etc.

Some Important Matrices

In these sections, I need to flesh out the proofs and provide some examples. How do I nicely input definitions and proofs into this?

Diagonal Matrices

Diagonal matrices D are matrices where all off-diagonal elements are zero ie $D_{ij} \neq 0$ only for $i = j$. In these notes assume unless otherwise specified that D is diagonal. If D is diagonal, then $D^T = D$ since it is symetric. Also, both $D_1 and D_2$ and D_1D_2 are diagonal. To see that the product is diagonal, Let A and B be diagonal. Then $(AB)_{ij} = \mathtt{r}\,(A)_i\, \mathtt{c}\,(B)_j$. Only the i^{th} element in $\mathtt{r}\,(A)_i \neq 0$, and the j^{th} element in $\mathtt{c}\,(B)_j \neq 0$. Therefore a dot product can only be non-zero when $i = j$.

Diagonal matrices are a scaling operation. As with other *elementary matrices*, pre-multiplying is a row transformation, post-multiplying is a column transformation. To see this, recognise the matrix

product as a matrix of dot products. In the case of pre-multiplication, where D is diagonal with $D_{ii} = di$

$$\begin{aligned}(DA)_{ij} &= \mathtt{r}(D)_i \cdot \mathtt{c}(A)_j \\ &= D_{ii}A_{ij} \\ &= d_i A_{ij}\end{aligned}$$

where the second line occurs because all elements in $\mathtt{r}(D)_i = 0$ except the i^{th} element, so only the i^{th} element in $\mathtt{c}(A)_i$ (ie A_{ij}), survives the dot product. The result is the whole of the i^{th} row of A is multiplied through by d_i. The analgous result happens for post-multiplication. This has important implications - if D is diagonal and $D_{ii} \neq 0 \; \forall \; i$ then D^{-1} exists, and each element of D^{-1} is $1/D_{ii}$. Note too the the determinant of a diagonal matrix is

$$\mathtt{det}(D) = \prod D_{ii} \tag{11}$$

which is clearly zero if any diagonal element is zero.

Triangular Matrices

Triangular matrices are ubiquitious in computation linear algebra, because they can be solved with little effort. Matrices can be upper or lower triangular:

- *Upper Triangular:* U is upper triangular if all $U_{ij} = 0$ when $i < j$
- *Lower Triangular:* L is lower triangular if all $L_{ij} = 0$ when $i > j$

I often refer to lower and upper as being types of triangular matrices. Assume everywhere that T is a triangular matrix of some type, and U/L are upper/lower triangular matrices unless otherwise specified. There are some interesting properties of triangular matrices.

- If T_1, T_2 have the same type, then the sum $T = T_1 + T_2$ also has the same type.
- If T_1, T_2 have the same type, then the product $T = T_1 T_2$ also has the same type.
- T^{-1} has the same type as T.
- $\mathtt{det}(T) = \prod T_{ii}$, the product of its diagonal.

Solving Triangular Matrices

Triangular matrices can be solved very quickly using backward/forward substitution. Consider solving $U\mathbf{x} = \mathbf{y}$. The bottom row only has one one non-zero element $U_{nn}x_n = y_n$, so

$$x_n = y_n / U_{nn} \tag{12}$$

. The next row has two:

$$\begin{aligned}U_{n-1,n-1}x_{n-1} + U_{n,n}x_n &= y_{n-1} \\ x_{n-1} &= \left(y_{n-1} - U_{n,n}x_n\right)/U_{n-1,n-1}\end{aligned}$$

and we know x_n from the previous step. We continue on recursively. An upper triangular matrix is solved the identical way except starting at the top row instead of the bottom. Clearly the process breaks down if T has a zero element on the diagonal, but we expected that since this would cause $\mathtt{det}(T) = 0$. A triangular matrix of size N will take $order N^2$ operations to solve in this way.

Consider a particular row in the algorithm described above. The operations are

$$\begin{aligned} y_i &= \sum_{j=1}^{N} U_{ij} x_j \\ &= \sum_{j=i}^{N} U_{ij} x_j \\ &= U_{ii} x_i + \sum_{j=i+1}^{N} U_{ij} x_j \\ x_i &= \left(y_i - \sum_{j=i+1}^{N} U_{ij} x_j \right) / U_{ii} \end{aligned}$$

where the second line follows since the first elements U_{i1} to $U_{i,i-1}$ are all zero, and in last last recognise that we have already found the next elements we are solving for, so that for x_i we already have found $x_{i+1}, x_{i+2}, \ldots, x_N$. Computation of x_i involves the summation of $U_{ij} x_j$ for $j = i+1, \ldots, N$, which is $\mathcal{O}(N)$. This calculation is performed N times, once for each of the N rows, so in total we require $N\mathcal{O}(N) = \mathcal{O}\left(N^2\right)$ operations.

Creating Triangular Matrices with LU decompositions

The LU decomposition of a matrix A are the upper and lower triangular matrices U and L where $A = LU$. Conventionally, the main diagonal of L is $L_{ii} = 1$. This ensures uniqueness, since A has n^2 elements, while L has

$$\begin{aligned} \text{Elements} &= n + (n-1) + \ldots + 2 + 1 \\ &= \sum_{i=1}^{n} i \\ &= n(n+1)/2 \end{aligned}$$

non-zero elements, as does U. So we have $n(n+1)$ non-zero variables in total to set using the n^2 elements of A. To make this match set n of them, the diagonal of U, to be all one. When computing L and U analytically, we use sequences of transformations via elementary matrices. In practice LU are computed numerically in $\mathcal{O}\left(()\, N^3\right)$ operations. Here, I should add both numeric and analytic examples of LU decompositions. Note that many authors refer to decompositions where rows have been swapped (ie *pivoting*) as a *LU decomposition with pivoting*. If a LU decomposition is attempted directly without any pivoting the algorithm will fail anywhere with a zero element on the main diagonal. For example the algorithm will fail to decompose

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{13}$$

where it is obvious that exchanging the rows leads to an easy solution. If the pivot operations on A are collected in the matrix P, the the decomposition is $PA = LU$. Here P is *not*-unique, but once P is specified LU is.

The benefit of the LU decomposition $PA = LU$ is that we can use forward and backward substitution to very quickly solve linear systems. Consider the system $A\mathbf{x} = \mathbf{y}$. To solve, find P which decomposes $PA = LU$, and solve $PA\mathbf{x} = LU\mathbf{x} = P\mathbf{y}$ in two steps. The trick is to let $U\mathbf{x} = \mathbf{v}$,

and solve in two steps as follows

$$\begin{aligned} A\mathbf{x} &= \mathbf{y} \\ PA\mathbf{x} &= P\mathbf{y} \\ L\left(U\mathbf{x}\right) &= P\mathbf{y} \\ L\mathbf{v} &= P\mathbf{y} \end{aligned}$$

for $\mathbf{v}$, and then solve

$$U\mathbf{x} = \mathbf{v} \tag{14}$$

for $\mathbf{x}$, as required.

Eigenvalues and Eigenvectors

Eigenvectors and *Eigenvalues* of a matrix A are fundamental quantities used regularly in matrix operations. The Eigenvector/value pair corresponding to a matrix A is the pair $\mathbf{v}$,λ with the property that

$$A\mathbf{v} = \lambda\mathbf{v} \tag{15}$$

The geometric interpretation of this is nice, but I've never found it particularly usefull. In two dimensions $\mathbf{v}$ is a line in the $x-y$ plane, and $A\mathbf{x}$ is a linear transformation that changes the direction $\mathbf{v}$ points, as well as its length. If $\mathbf{v}$ is an is an eigenvector of A, then the transformation $A\mathbf{v}$ only changes the length of the line $\mathbf{v}$, and not its direction. What is the practical importance of that? I'm not really sure. It is quite tedious to find eigenvalue/vector pairs analytically (include a few examples). Computing eigenvectors/values numerically is not as simple as the LU decomposition described earlier either. Since they are so critically important in all areas of applied maths/statistics, the computation of eigenvalues is a huge field of study. Note that if λ is an eigenvalue with a corresponding eigenvector $\mathbf{v}$, then any multiple of λ is also an eigenvalue. As such two vectors $\mathbf{u}$ and $\mathbf{v}$ with the above property, where $\mathbf{u} = c\mathbf{v}$ (ie one is just a multiple of the other) do not count as separate eigenvectors. To see this note that

$$\begin{aligned} A\mathbf{v} &= \lambda\mathbf{v} \\ A\left(c\mathbf{v}\right) &= \left(c\lambda\right)\mathbf{v} \\ A\mathbf{u} &= \lambda_u\mathbf{u} \end{aligned}$$

where $\mathbf{u} = c\mathbf{v}$ and $\lambda_u = c\lambda$.

To finding eigenvalue/vector pairs, note that if λ is an eigenvalue of A, then

$$\begin{aligned} A\mathbf{v} &= \lambda\mathbf{v} \\ \left(A - \lambda I\right)\mathbf{v} &= 0 \end{aligned}$$

which means the vector $\mathbf{v}$ is in the null-space of the matrix $A - \lambda I$. If such a vector exists, so that the null-space is non-empty, we know $\left(A - \lambda I\right)$ is a singular matrix. Since the purpose is to find such a $\mathbf{v}$, we need to find the conditions for which $\left(A - \lambda I\right)$ is singular. The most obvious is that it has a zero determinant; ie

$$\texttt{det}\left(A - \lambda I\right) = 0 \tag{16}$$

This results in a polynomial equation for λ, which may have multiple solutions, some of which might be repeated. This equation is refered to as the *characteristic polynomial* of A. Once we have all the solutions λ_i, we can work out the corresponding $\mathbf{v}_i$. These are the eigenvalue/vector pairs.

This is quite a tedious process! I should put in an example or two here.

Since computing the λ_i reduces to working with determinants, the special structure of some matrices means that computing the λ_i is straight forward. The subtraction $A - \lambda I$ only modifies the diagonal of A, many matrices retain the special structure after the subtraction. Most importantly:

- **Diagonal Matrices** If D is diagonal, then $A - \lambda I$ is diagonal too. The determinant of a diagonal matrix is $\texttt{det}(D) = \prod D_{ii}$, so the characterist polynomial of $A - \lambda I$ is $\texttt{det}(A - \lambda I) = \prod (D_{ii} - \lambda) = 0$. As such the diagonal elements D_{ii} are themselves the eigenvalues of D.
- **Triangular Matrices** If T is triangular, then $A - \lambda I$ is triangular too. The determinant of a triangular matrix is $\texttt{det}(T) = \prod T_{ii}$, so the characterist polynomial of $A - \lambda I$ is $\texttt{det}(A - \lambda I) = \prod (T_{ii} - \lambda) = 0$. As such the diagonal elements T_{ii} are themselves the eigenvalues of T.

Eigenvector/value pairs give a great deal of information about the matrix they were computed from. Let $(\lambda_i, \mathbf{v}_i)$ be the eigenvalue/vector pairs of the matrix A.

- If any $\lambda_i = 0$, then A is singular. Remember that if A is non-singular, then it has an empty null-space. That is, there can be no non-trivial vector $\mathbf{v}$ where $A\mathbf{v} = 0$.
  However, if $\lambda = 0$ is an eigenvalue, and $\mathbf{v}$ its corresponding eigenvector, then $A\mathbf{v} = \lambda\mathbf{v} = 0$.
  This means the null-space of A is not empty (ie the eigenvector $\mathbf{v}$ is in it), so A is singular.
- An $(n \times n)$ matrix A has exactly n eigenvalues. Some may be repeated, and some may be complex.
- Every eigenvalue has at least one *eigenvector*.
- If $\lambda_1 \neq \lambda_2$ then their corresponding eigenvectors are linearly independant.
- If A has an eigenvalue with multiplicity m, it has at least one eigenvector, but may have less than m eigenvectors. This means that A may have n eigenvectors which are all linearly independant, but it may have less if any eigenvalues were repeated.