# Chem237: Lecture 13

Shane Flynn

5/3/18

## 1 Note

9-9-19; Just copied the lecture notes I have over, this has not been reviewed/edited/made ready for the world.

## Gaussian Elimination

Reduce your matrix to an upper-triangular (or lower triangular) form by performing operations such that the determinant doesn't change.

Consider the square N by N matrix $\mathbf{A}$, which has column vectors labeled $\vec{a}_i$

$$\mathbf{A} = \begin{bmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{1N} & \cdots & A_{NN} \end{bmatrix} = \begin{bmatrix} \vec{a}_1 & \dots & \vec{a}_N \end{bmatrix} \tag{1}$$

To compute the determinant of $\mathbf{A}$ we know the following

$$\det(\mathbf{A}) = \det(\vec{a}_1 \cdots \vec{a}_N) = \det(\vec{a}_1 - \lambda_{a2}, \vec{a}_2 \cdots \vec{a}_N) \tag{2}$$

Let $\lambda_1 = \frac{A_{N1}}{A_{N2}}$ then we have

$$\begin{bmatrix} \star & A_{12} & \cdots & A_{1N} \\ \star & A_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & A_{N2} & \cdots & A_{NN} \end{bmatrix} \tag{3}$$

Next step set $\lambda_2 = \frac{A_{N2}}{A_{N3}}$ then we have

$$\begin{bmatrix} \star & \star & \cdots & \vdots \\ \star & \vdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{NN} \end{bmatrix} \tag{4}$$

In this way we can reduce every element of the Nth row to 0 except for the final term $A_{NN}$. We can then repeat this process for the N-1 row (Second to last row), which will convert every element up to N-1 in the row to 0.

$$\begin{bmatrix} \star & \star & \cdots & \vdots \\ \star & \vdots & \cdots & \vdots \\ 0 & \cdots & 0 & \star \\ 0 & \cdots & 0 & A_{NN} \end{bmatrix} \tag{5}$$

Repeating the process iteratively eventually ends in

$$\begin{bmatrix} \star & \star & \star & \star \\ 0 & \star & \star & \star \\ 0 & 0 & \star & \star \\ 0 & 0 & 0 & \star \end{bmatrix} \tag{6}$$

What happens if you have a 0 at some point during the iterations? Just move it and keep going.

### 1.1 Numerical Linear Algebra

Using the definition of the determinant the scaling would be N! which is very bad. Using the algorithm from above leads to N$^3$ scaling, which is a common result for numerical linear algebra algorithms. $a_1$ - $\lambda_{a2}$ = 2 operations scaling as N, Appy to row = N, apply to other ros = N.

### 1.2 Linear Indepedence

Consider a set of n vectors $\vec{a}_1 \cdots \vec{a}_N$. These vectors are linearly indepedent if for any $\lambda_1 \cdots \lambda_N$

$$\sum_{i=1}^{N} \lambda_i \vec{a}_i \neq 0 \tag{7}$$

and vice versa linearly depedent if

$$\sum_{i=1}^{N} \lambda_i \vec{a}_i = 0 \tag{8}$$

Theorem: determinate matrix formed by N vectors $\det(\vec{a}_1 \cdots \vec{1}_N) = 0$ iff $\vec{a}_1 \cdots \vec{a}_N$ are linearly depedent, and vice versa linearly indepedent vectors produce a determinate $\neq 0$.

If your vectors are linearly indepedent can't ever get a digonal element = 0, if 1 element on the diagonal is 0 the determinent is 0, for a linearly depedent set of vectors you can get 1 column being all 0.

## 2 Linear Systems

Consider a linear system of N equations

$$A_{11}x_1 + \cdots + A_{1N}x_N = b_1 \tag{9}$$

$$\cdots = \vdots \tag{10}$$

$$A_{N1}x_1 + \cdots + A_{NN}x_N = b_N \tag{11}$$

Where the vector $\vec{x}$ contains our N unknown elements.

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \tag{12}$$

For this system matrix A is referred to as the coefficient matrix, and the column vector b is the right hand side (RHS), so we have the matrix equation $bA\vec{x} = \vec{b}$. In general A can be rectangular, leading to differnt types of systems, namely overdtetermined systems, and underdetermined systems, corresponding to no solution in general and an infinite number of solutions in general respectively.

Consider the case of a square matrix $\mathbf{A}$, if the system is $\mathbf{A}\vec{x} = 0$ then the linear system is termed homogenous, if $vecb \neq =$ the system is inhomogenous.

Algebraically speaking if $\mathbf{A}^{-1}$ exists then $\vec{x} = \mathbf{A}^{-1}\vec{b}$, however you would need the inverse of a matrix which can be non-trivial. The inverse of a matrix existing is related to the determijnate of the matrix not being 0, if the determinate is 0 than the inverse does not exist.

Consider a matrix A of the form

$$\begin{bmatrix} \star & \star & \star & \star \\ 0 & \star & \star & \star \\ 0 & 0 & \star & \star \\ 0 & 0 & 0 & \star \end{bmatrix} \tag{13}$$

The solution to the associated matrix equation is trivial, simply use recursion to solve the problem.

$$\begin{bmatrix} A_{11} & \star & \dots & \star \\ 0 & A_{22} & \dots & \star \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_{NN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{14}$$

The last equation (bottom row) gives (remember **A** is a coefficient matrix, we know the values).

$$A_{NN}x_N = b_N \Rightarrow x_N = \frac{b_N}{A_{NN}} \tag{15}$$

Using $x_N$ we can solve for $x_{N-1}$ using the second to last row, and clearly we can solve the problem recursively. The solution to a trianglular matrix (upper or lower triangle doesn't matter) can always be solvef recursively.

We can always add rows together (NOT COLUMNS), adding rows wont change anything. So start by reducing the matrix to an upper-triangular form with Gaussian elimination. The you can use recursion to solve the system. This is analagous to the determinate problem, if $\det(\mathbf{A}) \neq 0$ then you can solve the linear system with recursion to find a unique solutiojn.

### 2.1 Determinate = 0

In the case of $\det(\mathbf{A}) = 0$ the rows are linearly depedent, meaning 1 equation will have a LHS where all terms are 0. In general $b_i$ does not have to be 0 therefore we find 0=$b_i$ as one of our equaitons, meaning no solution unless $b_i$=0.

In general if the determinate of a matrix is 0 than if $\vec{x}$ is a solution to $\mathbf{A}\vec{x} = 0$, $\lambda\vec{x}$ must also be a solution, leading to an infinite number of solutions. A homogenous system has an infinit number of solutions if the system is linearly depedent. If you try removing an equaiton (to remove linear depedence) you will have more unknowns than equations and again get an infinite number of solutions in general.

## 3 Matrix Inverses

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \tag{16}$$

How do we compute matrix B ($\mathbf{B} = \mathbf{A}^{-1}$)? Consider a cofactor matrix $\mathbf{A}_c$, every element of this matrix is a cofactor.

$$\mathbf{A}_c = \begin{bmatrix} A_{c,11} & \cdots & A_{c,1N} \\ \vdots & \ddots & \vdots \\ A_{c,1N} & \cdots & A_{c,NN} \end{bmatrix} \tag{17}$$

In the context of square matricies the ith-jth cofactor ($\mathbf{A}_{c.ij}$) is computed by removing the ith and jth row and column from the original matrix **A** and then computing the subsequent determinate of the (N-1 × N-1) resulting matrix. Finally multiply by $(-1)^{i+j}$ and you have the cofactor.

### Adjoint Method

It turns out that we can compute the inverse of a matrix using the cofactor matrix transpose.

$$\mathbf{A}^{-1} = \frac{\mathbf{A}_c^T}{\det(\mathbf{A})} \tag{18}$$

Clearly this is very expensive (computing the determinate for $\mathrm{N}^2$ elements, however, it does give an explicit expression for the inverse of a matrix. It also shows that the determinate must not be 0, consistent with our claims before.

Consider a different approach for finding the inverse of a matrix.

$$\mathbf{AB} = \mathbf{I} \tag{19}$$

Where **B** is the unknow inverse of a matrix we want to compute.

Consider the first term in this system.

$$\mathbf{A}\vec{b}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{20}$$

It is convention to define the columns of the identity matrix as $\vec{e}$.

$$\mathbf{I} = \begin{bmatrix} \vec{e}_1 & \ldots & \vec{e}_N \end{bmatrix} \tag{21}$$

Where $\vec{e_i}$ refers to a column vector containing all 0 terms, except a 1 at the i'th row. Our system can therefore be written as

$$\mathbf{A}\vec{b}_1 + = \vec{e}_1 \tag{22}$$

$$\mathbf{A}\vec{b}_N + = \vec{e}_N \tag{23}$$

$$\tag{24}$$

Which can be solved indepedently

# 4 Solving Systems

You can easily solve a system of equations using Gaussian Elimination. Consider applying Gauss Elimination to

$$\mathbf{AI} \tag{25}$$

# 5 Matrix Inverses

Some useful properties of matrix inverse, consider matricies A and B

$$\begin{aligned} (\mathbf{AB})^{-1} &= \mathbf{B}^{-1}\mathbf{A}^{-1} \\ \left(\mathbf{A}^{-1}\right)^{-1} &= \mathbf{A} \\ \left[\mathbf{A}^{-1}, \mathbf{A}\right] &= 0 \end{aligned} \tag{26}$$

It is useful to think of the inverse of a matrix as simply some function acting upon a matrix.

# 6 Unitary Matrix

Unitary matricies are defined as a matrix whose inverse equals its conjugate transpose

$$\begin{aligned} \mathbf{U}^{\dagger}\mathbf{U} &= \mathbf{I} \\ \mathbf{U}^{\dagger} &= \mathbf{U}^{-1} \end{aligned} \tag{27}$$

This means

$$\left(\mathbf{U}^{\dagger}\right)_{mn} = \left(\mathbf{U}^{\star}\right)_{nm} \tag{28}$$

Unitary matricies have nice properties, and the associated unitary transformations have nice properties.

$$\begin{aligned} \mathbf{U}\mathbf{U}^{\star} &= 1 \\ |\mathbf{U}| &= 1 \\ \mathbf{U} &= e^{i\theta} \end{aligned} \tag{29}$$

These numbers are special, all unitary matricies eigenvales have this property.

Orthogonal vectors also have the Hermitian Inner Product (the final line below).

$$\begin{aligned} \mathbf{U} &= \begin{bmatrix} \mathbf{U}_1 & \dots & \mathbf{U}_N \end{bmatrix} \\ \mathbf{U}_i^{\dagger}\mathbf{U}_j &= 1 \\ \sum_k \mathbf{U}_{jk}\mathbf{U}_{ik}^{\star} &= \delta_{ij} \end{aligned} \tag{30}$$

### 6.1 Unitary Transformation

$$\begin{aligned} \mathbf{A}' &= \mathbf{U}\mathbf{A}\mathbf{U}^\dagger \\ \det(\mathbf{A}) &= \det(\mathbf{A}) \end{aligned} \tag{31}$$

This last line comes from $\det(\mathbf{U}) = e^{i\theta}, \det(\mathbf{U}^\dagger) = e^{-i\theta}$.

The fourier transform is related to unitary transforms in special cases, all of which have special properties.

$$\begin{aligned} \mathbf{A}\Psi_k &= a_k\Psi_k \\ \mathbf{A}'\Psi_k &= a_k\Psi_k \end{aligned} \tag{32}$$

The unitary transformation preserves teh determinant eigenvalues and eigenvectors.

## Orthogonal Matrix

Orthogonal matricies are a special cas of the unitary matrix

$$\begin{aligned} \mathbf{W}\mathbf{W}^T &= \mathbf{I} \\ \mathbf{W} &= \begin{bmatrix} \vec{e}_1 & \dots & \vec{e}_N \end{bmatrix} \\ \vec{w}_i\vec{w}_j &= \delta_{ij} \end{aligned} \tag{33}$$

One example of an orthogonal matrix is teh rotation matrix

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{34}$$

Gaussian elimination preserves the determinate of a matrix just like the unitary transformation. Unitary is better though, in gaussian elimination you divide (first step) which can generate numerical instability.

In numerical linear algebra using gaussian elimination is not feasible (even for well behaved matricies) because of this division instability and associated rounding errors. Gaussian elimination is unstable.

All good linear algebra is based on unitary matricies, much more stable. A unitary transformation is a non-trivial transformation, you performa transformation that magically does not change the determinate or eigenvalues and eigenvectors, it is very powerful.

Many methods in numerical linear algebra combine LU decomposition with unitary transformations.