211 LECTURE 15

Gauss-Jordan Elimination

Today we study an efficient method for solution of systems of linear equations. It is called *Gauss-Jordan Elimination*.

We begin with a system of m equations in n unknowns. We will write this in matrix form as $A \cdot \mathbf{x} = \mathbf{b}$, where A is an $m \times n$ matrix, $\mathbf{x}$ is a column vector of size n and $\mathbf{b}$ is a column vector of size m. The process is summarized as follows:

Step One: Write the equations in augmented matrix form.

Step Two: Use row operations to put this matrix into *reduced row echelon form*. This is sometimes called row reduction.

Step Three: Assign some arbitrary constants for the free variables, and write out the solutions.

Step one is one we've covered already. We focus now on step two. A matrix is said to be in row echelon form if it looks like

$$\begin{pmatrix} x & * & * & * & * & * & * \\ 0 & x & * & * & * & * & * \\ 0 & 0 & 0 & 0 & x & * & * \\ 0 & 0 & 0 & 0 & 0 & x & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Here by x we mean some non-zero number, and by $*$ we mean some arbitrary number. The x entries are called *pivots*. a column without a pivot is called a *free column* and the corresponding variable is called a *free variable*. In the matrix above the third, fourth and seventh columns are free, so we get three free variables x_3, x_4, x_7. (well, in terms of our augmented matrices, the last column should be $\mathbf{b}$ essentially–so perhaps we shouldn't count x_7.)

The idea is that to get to this form one needs only the operations of interchanging rows and adding a multiple of one row to another. We apply the process by making the pivots move to the right as we move down. First, zero out as much of the first column as you can-but at least everything below the diagonal. Then you move to the second column and eliminate everything below the diagonal. This continues till you get the form above.

Note that each row can contain only one pivot. It is possible to get rows without pivots–they should all be collected together at the bottom of the matrix. These rows require special care in the final steps.

This form is a very good start, and many texts use this as the end of the main process. At this point one can usually take the new (equivalent) system and solve by hand. This is fine for small systems when working by hand; however, I advocate continuing on to put the system into reduced row echelon form:

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

To do so, we multiply rows by constants to get pivots equal to one, and then add multiples of rows to cancel things above the pivot points. This time work right to left. This is sometimes called the

"backward pass".

Now we are ready for the third step. We assign some arbitrary constants to the value of the free variables, say $s, t, \ldots$, and then we can simply rearrange the rows to write down the solutions. This has happened because columns which are not free have only one non-zero element, and this entry is equal to 1. Thus the corresponding variable is easy to solve for as it now appears in our new version of the system only once!

To clarify the process, we shall do some small variable examples in detail, and use it to introduce the relevant terminology.

Example The first thing to be aware of: some systems do not have a solution. Such a system is called *inconsistent*.

$$\begin{aligned} x+2y\ +z &= 3 \\ y+6z &= 2 \\ x\ +y-5z &= 0 \end{aligned}$$

Example In this example, things work out just right.

$$\begin{aligned} 2x\ +y+3z &= 1 \\ x-3y\ +z &= 2 \\ x+7y\ +z &= 3 \end{aligned}$$

Example Free column.

$$\begin{aligned} 7x+2y\ -3z &= \ \ 0 \\ x\ -y\ +2z &= \ \ 1 \\ 9x+9y-16z &= -5 \end{aligned}$$

Example another one for good measure.

$$\begin{aligned} -2x+3y+6z-7u\ -v &= -1 \\ -x-6y\ +z-6u-8v &= -8 \\ -9y-5z-1u+9v &= \ \ 9 \end{aligned}$$

Discuss the picture of the solution set from the outcome of the process and how to write out solution as parameterized set.

There is an interpretation of this process that makes everything sit together really well. Applying a row operation is the same as multiplying our matrix equation on the left by an *elementary matrix*. An elementary square matrix of size m is one that fits one of the following descriptions. Let E_{ij} be the $m \times m$ square matrix which has all entries equal to zero except for a 1 in the ij spot.

First kind: a matrix of the form $\mathbf{I}_m - E_{ii} - E_{jj} + E_{ij} + E_{ji}$.
Second kind: a matrix of the form $\mathbf{I}_m + (x-1) \cdot E_{ii}$.
Third kind: a matrix of the form $\mathbf{I}_m + x \cdot E_{ij}$.

(actually, the first kind can be viewed a special type of the third, but it is helpful to separate them in your head.)

Suppose that $A \cdot \mathbf{x} = \mathbf{b}$ is associated to a system of m equations in n unknowns. Left multiplying by an elementary matrix of size m as above has the following effect on your equations:

first kind: switching the rows i and j.
second kind: multiplying the ith row by x.
third kind: adding x times the jth row to the ith row.

These are easy to see if you use the 2×2 case as examples.

Now we can interpret our operations properly. Given a matrix equation $A \cdot \mathbf{x} = \mathbf{b}$ applying row operations is the same as using a sequence of multiplications (on the left) by elementary matrices. That is, after k steps, we have found k elementary matrices $M_1, M_2, \ldots, M_k$ and multiplied on the right to have the new equivalent system

$$M_k \cdot M_{k-1} \ldots M_2 \cdot M_1 \cdot A \cdot \mathbf{x} = M_k \cdot M_{k-1} \ldots M_2 \cdot M_1 \cdot \mathbf{b}.$$

What we are hoping for is that the matrix product on the left hand side eventually becomes something like a reduced row echelon form matrix (this time without the augmentation). In the mean time the right hand side is being slowly transformed into a solution vector!

More on the square case

In the square case, we get more out of this. If A is square of size n, then the appropriate elementary matrices to use have size n and we are simply aiming for the relation

$$M_k \cdot M_{k-1} \ldots M_2 \cdot M_1 \cdot A = \mathbf{I}$$

That is, we are slowly computing the inverse of A in the form of

$$A^{-1} = M_k \cdot M_{k-1} \ldots M_2 \cdot M_1.$$

Recall that if A is invertible, then our equation will have a unique solution given by $x = A^{-1} \cdot \mathbf{b}$.

A square matrix A of size n is called *non-singular* when for every $\mathbf{b}$ the system of equations $A \cdot \mathbf{x} = \mathbf{b}$ has a solution. It is *singular* otherwise.

Proposition: A square matrix A is non-singular if and only if it is invertible.

Give simple proof.

Discuss finding the inverse of a matrix by using row operations/elementary matrices on a big augmented matrix $[A, \mathbf{I}]$. If it works, we get an output of $[\mathbf{I}, A^{-1}]$.

This process is a lot of work. it would be nice to know that it would be worth it before we start.

Of course, if A is singular, we find this out because either there will be a free column during the process, or the system will be inconsistent. Then we have lots of solutions, and they will need to be parameterized like the above examples, or we have no solutions. These are often referred to as *degenerate cases* for the square system.

Coming Soon. A way to test in advance if a matrix is invertible. A way to handle systems that aren't, and some of the complications.