Lecture 11: Graphs and their Adjacency Matrices

Vidit Nanda

The class should, by now, be relatively comfortable with Gaussian elimination. We have also successfully extracted bases for null and column spaces using Reduced Row Echelon Form (RREF). Since these spaces are defined for the transpose of a matrix as well, we have four fundamental subspaces associated to each matrix.

Today we will see an interpretation of all four towards understanding *graphs and networks*.

1. Recap

Here are the four fundamental subspaces associated to each $m \times n$ matrix $A : \mathbb{R}^n \to \mathbb{R}^m$:

(1) The *null space* $N(A)$ is the subspace of $\mathbb{R}^n$ sent to the zero vector by A,
(2) The *column space* $C(A)$ is the subspace of $\mathbb{R}^m$ produced by taking all linear combinations of columns of A,
(3) The *row space* $C(A^T)$ is the subspace of $\mathbb{R}^n$ consisting of linear combinations of rows of A, or the columns of its transpose A^T, and finally,
(4) The *left nullspace* $N(A^T)$ is the subspace of $\mathbb{R}^m$ consisting of all vectors which A^T sends to the zero vector.

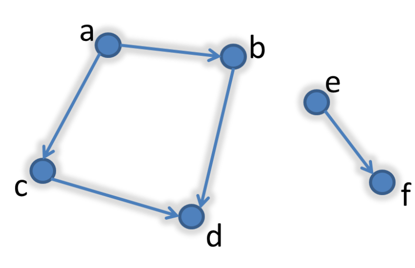
By the **Fundamental Theorem of Linear Algebra** from Lecture 9 it turns out that knowing the dimension of one of these spaces immediately tells us about the dimensions of the other three. So, if the *rank* – or $\dim C(A)$ – is some number r, then immediately we know:

- $\dim N(A) = n - r$,
- $\dim C(A^T) = r$, and
- $\dim N(A^T) = m - r$.

And more: we can actually extract bases for each of these subspaces by using the RREF as seen in Lecture 10.

2. Graphs

A **graph** – not to be confused with the graph of a function – is a collection of *nodes* and *edges* (ar arrows) between nodes; here's an example:



Let's call this graph $\mathcal{G}$ and note that it has 6 nodes (labeled a throught f) and 5 edges. Convention dictates that we denote an edge from a to b simply by ab. So, the edges of $\mathcal{G}$ are ab, ac, bd, cd and ef. If we stare at the graph a little longer, we might even notice that it has two "pieces" and one "loop".

It turns out that graphs have infested just about every field of science and engineering these days: they are extremely convenient and intuitive models for encoding various structures, data types and their interrelations. Once you start thinking in terms of graphs, you start seeing them everywhere.

Here are a few concrete examples of graph-based models that are *all actually used* in modern research:

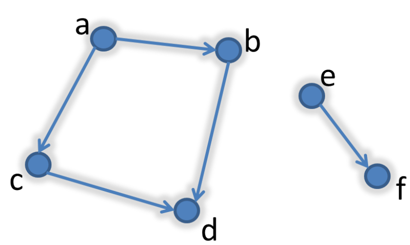
(1) **Computer Science**: There's always the internet! Give each website a node, and throw in an edge from one website to another whenever the first links to the second.
(2) **Electrical Engineering**: It doesn't take too much imagination to see that any electrical circuit is a graph.
(3) **Finance and Economics**: Each company is a node, and an edge from one to another indicates that the first company bought products from the second one.
(4) **Biology**: Nodes represent habitats while edges are migration paths of a given species.
(5) **Navigation**: Nodes are major cities while edges are the interconnecting roads.

Okay, so this list could get much longer. Let's try to see what on earth linear algebra has to do with graphs, and in particular, how the four fundamental subspaces show up.

3. Adjacency Matrices

When a graph is tiny (like our friend $\mathcal{G}$ with only 6 nodes and 5 edges), it is really easy to visualize. Let's say it was the graph of the internet: you'd know immediately that there are two pages (e and f) that would be impossible to reach from $\mathsf{a}, \mathsf{b}, \mathsf{c}$ and d if all you were allowed to do was click links and the back button on your browser. Unfortunately for those wishing to analyze the structure of internet[1], the actual graph corresponding to the internet is enormous and can barely be stored on a single machine, let alone be visualized. It is in the context of massive graphs that linear algebra becomes extremely useful.

So here's the first bit of "cleverness". Let's see $\mathcal{G}$ again:



We will construct (surprise, surprise) a matrix $A_{\mathcal{G}}$ which faithfully encodes the structure of $\mathcal{G}$ as follows: each column is a node, each row is an edge; the entry in the i-th row and j-th column is given as follows. If the corresponding node is the source of the edge, then we put in -1. If it is the target, we put in $+1$. And if that node misses the edge completely, we put in a zero.

Maybe the easiest way to get a handle on this stuff is to see a toy example. Here is $A_{\mathcal{G}}$ in all its glory, with rows and columns decorated by corresponding edges and nodes respectively:

$$A_{\mathcal{G}} = \begin{array}{c} \\ \mathsf{ab} \\ \mathsf{ac} \\ \mathsf{bd} \\ \mathsf{cd} \\ \mathsf{ef} \end{array} \begin{array}{c} \begin{array}{cccccc} \mathsf{a} & \mathsf{b} & \mathsf{c} & \mathsf{d} & \mathsf{e} & \mathsf{f} \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \end{array}$$

This is *not* a square matrix in general! There are as many columns as nodes and as many rows as edges, so we only get a square matrix when the number of nodes equals the number of edges.

Before moving on to the four subspaces of $A_{\mathcal{G}}$, I want to re-emphasize: *a lot of what we conclude today will be obvious* from just a simple glance at the graph $\mathcal{G}$. But the whole point of developing these tools is

[1] but fortunately for millions of cat video afficionados

that there are tons of interesting examples (see above) of graphs that we have absolutely no hope of being able to visualize with any degree of efficiency. On the other hand, the linear algebra still works!

4. Null Spaces of the Adjacency Matrix

We begin with the two null spaces $N(A_{\mathcal{G}})$ and $N(A_{\mathcal{G}}^T)$: these two are the easiest to interpret in the context of graphs. At the end of each calculation, I will place a **moral** which explains precisely the connection between a fundamental subspace of the adjacency matrix and its interpretation in the world of graphs.

4.1. The Null Space. $N(A_{\mathcal{G}})$ is precisely the subspace of $\mathbb{R}^5$ consisting of precisely those vectors which $A_{\mathcal{G}}$ annihilates[2]. Without resorting to any Gaussian elimination, we can already tell that the null space will at least have dimension 1: since any given edge contributes a -1 and a $+1$ to each row, adding up all the columns produces the zero vector, so there is definitely some linear dependence here!

But let's be more systematic and get as much of the full picture as possible: we should solve for $x_1, x_2, \ldots, x_6$ in the following linear system:

$$\begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Look carefully at how matrix multiplication is working! Each row (i.e., edge in our graph) multiplies across our variables, picking out two with opposite signs and zeroing out everything else. We now want to solve the really easy system of equations:

$$\begin{pmatrix} -x_1 + x_2 \\ -x_1 + x_3 \\ -x_2 + x_3 \\ -x_3 + x_4 \\ -x_5 + x_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

The first component forces $x_1 = x_2$, the second forces $x_2 = x_3$ and so forth. If we consider the obvious correspondence of nodes in our graph and these variables – a and x_1, b and x_2, etc., up to x_6 and f – then we see that *whenever there is a path connecting one node to another (regardless of the arrow directions), their corresponding variables must be equal.*

In our case, we have $x_1 = x_2 = x_3 = x_4$ coming from the first four components of the vector equation above (i.e., the first piece of the graph consisting of nodes a through d). Separately, we also get $x_5 = x_6$ from the fifth component of the vector equation which corresponds to the second piece of the graph containing e and f. Of course, this describes the entire null space $N(A_{\mathcal{G}})$: it is *two* dimensional and the following two vectors in $\mathbb{R}^6$ form a basis:

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

This is cool: the fact that the first basis vector assigns equal nonzero values to the first four components means that a, b, c and d form a single piece of $\mathcal{G}$ and also that this piece does not contain e and f! Similarly, the second basis vector tells us that e and f are in the same piece as each other.

Moral: The dimension of the null space of an adjacency matrix counts the number of pieces in the underlying graph. Even better, if you have an explicit basis for the null space as shown above then you can also immediately tell which nodes belong to which piece.

---

[2]All jokes aside, this is actually a technical term in abstract algebra, meaning *sends to zero*. Sometimes mathematicians like to pretend that they can be menacing: see also *Killing form* and *monster group*.

4.2. The Left Nullspace. Again, in order to compute the left nullspace $\mathsf{N}(A_{\mathcal{G}}^{\mathsf{T}})$ of A one might be tempted to start quickly doing row operations on $A_{\mathcal{G}}$ (or $A_{\mathcal{G}}^{\mathsf{T}}$) in RREF. And for the types of gigantic graphs which arise in actual scientific models, this sort of brute-fource computation is the only strategy! But our toy example $\mathcal{G}$ is small enough that we can actually see *what the row space means*! In order to see this, we take a short walk.

So, imagine you are at node $\mathfrak{a}$ in $\mathcal{G}$, and you are allowed to start walking on the graph along or against the edges. You can never reach $\mathfrak{e}$ and $\mathfrak{f}$ this way, they are not connected to $\mathfrak{a}$ by a path consisting of edges, but you can reach $\mathfrak{b}, \mathfrak{c}$ and $\mathfrak{d}$. Let's go clockwise around the loop: $\mathfrak{a}$ to $\mathfrak{b}$ to $\mathfrak{d}$ to $\mathfrak{c}$ and then back to $\mathfrak{a}$. In order to keep track of whether the edge is going with us or against us, we will assign a negative sign to any edge that is pointed against us in our walk: so $\mathfrak{ab}$ and $\mathfrak{bd}$ gets plus signs since they are aligned with us on the clockwise walk, but $\mathfrak{cd}$ and $\mathfrak{ac}$ are pointed against us so we get zeros. Here is the result of our clockwise walk which takes us from $\mathfrak{a}$ to $\mathfrak{a}$ clockwise around the loop:

$$\mathfrak{ab} + \mathfrak{bd} - \mathfrak{cd} - \mathfrak{ac}.$$

If we decided to walk counterclockwise, we'd get the same thing but with all the signs flipped.

Our walk on $\mathcal{G}$ actually becomes a row operation on $A_{\mathcal{G}}$! Remember,

$$A_{\mathcal{G}} = \begin{array}{c} \\ \mathfrak{ab} \\ \mathfrak{ac} \\ \mathfrak{bd} \\ \mathfrak{cd} \\ \mathfrak{ef} \end{array} \begin{array}{c} \begin{array}{cccccc} \mathfrak{a} & \mathfrak{b} & \mathfrak{c} & \mathfrak{d} & \mathfrak{e} & \mathfrak{f} \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \end{array}$$

So the clockwise walk corresponds to a linear combination of the first four rows: Row 1 plus Row 3 minus Row 4 minus Row 2. You can crunch through the computation and check that this combination is in fact the zero row. In fact, there's no need to compute anything – look at node $\mathfrak{b}$ for instance. It shows up twice in this row sum (once in $\mathfrak{ab}$ and once in $\mathfrak{bd}$). When we enter $\mathfrak{b}$ via $\mathfrak{ab}$, it has a coefficient of $+1$ and when we leave it via $\mathfrak{bd}$ we get -1, so the column corresponding to $\mathfrak{b}$ cancels out to give 0. Similar arguments show that the columns corresponding to $\mathfrak{a}$, $\mathfrak{c}$ and $\mathfrak{d}$ are also zero.

It is important to note that *the zeroing out happens only for loops*. If instead of the entire loop we only traverse along $\mathfrak{ab}$, $\mathfrak{bd}$ and then against $\mathfrak{cd}$, the coefficients for $\mathfrak{a}$ and $\mathfrak{d}$ do not zero out. In any case, we have found a linear dependence among the rows so the left nullspace is at least one dimensional, since it includes the span of the following vector in $\mathbb{R}^5$

$$\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 0 \end{pmatrix}.$$

Thus, $\dim \mathsf{N}(A_{\mathcal{G}}^{\mathsf{T}})$ *equals the number of genuine loops in* $\mathcal{G}$. If you go through the trouble of producing an RREF of $A_{\mathcal{G}}$, then you'd discover that in fact $\mathsf{N}(A_{\mathcal{G}}^{\mathsf{T}})$ has dimension one[3]. So the vector mentioned above forms a complete basis for this space, and more importantly, by looking at the columns of this vector as representatives of the edges, you can actually extract information about the loop: the unique loop can be produced by going forwards (i.e., coefficient $+1$) along $\mathfrak{ab}$ and $\mathfrak{bd}$, then backwards (coefficient -1) along $\mathfrak{cd}$ and $\mathfrak{ac}$.

Moral: The dimension of the left nullspace of an adjacency matrix counts the number of loops in the underlying graph. And if you produce a basis for this subspace using the method above, you can actually identify the relevant loops in your graph!

[3]Of course, there are two easier ways of checking this when the graph is $\mathcal{G}$: we already know that $\mathsf{N}(A_{\mathcal{G}})$ has dimension 2, so by the fundamental theorem we know that the rank of $A_{\mathcal{G}}$ is $6-2=4$, and so the dimension of the left nullspace is $5-4=1$. Even better, you can look at $\mathcal{G}$ and easily check that it has only one loop!

5. The Column and Row Spaces of the Adjacency Matrix

What we should already know by the fundamental theorem is that both Column and Row spaces of $A_{\mathcal{G}}$ have dimension 4. But these subspaces are slightly trickier to interpret than the null spaces! The null spaces of $A_{\mathcal{G}}$ gave us geometric information about $\mathcal{G}$ (i.e., number of pieces and number of loops) but the column and row space are more about *structures built on top of* $\mathcal{G}$.

For now, the easiest structure to keep in mind is an **electric circuit**. So at each node, we have a *voltage* and across each edge we have a *current*. We will see that the row and column spaces of $A_{\mathcal{G}}$ give us information about such circuits built on top of $\mathcal{G}$. The input space $\mathbb{R}^6$ of $A_{\mathcal{G}}$ now consists of vectors $x = (x_1, \ldots, x_6)$, where the components are assignments of voltages to $a, b, \ldots, f$. The output space $\mathbb{R}^5$ will now be thought of as assignments of currents $b = (b_1, \ldots, b_5)$ to the edges $ab, ac, \ldots, ef$. The adjacency matrix now takes voltage-space to current-space. It's transpose does the opposite thing: it takes in currents and spits out voltages.

The basic physical principles governing electrical circuits are the entirely reasonable **Kirchoff's Laws**. The *net current* across each node is the sum of outgoing currents minus the sum of incoming currents. Similarly, the *voltage differential* across each edge equals the voltage at its target minus the voltage across its source. Kirchoff's laws are about these two quantities:

(1) Kirchoff's **Current** law (abbreviated KCL) requires the net current across each node to be zero.
(2) Kirchoff's **Voltage** law (henceforth KVL) requires that the sum of voltage differentials of edges in a loop is zero.

Even if you have never heard of these laws before, they really *should* appeal to your physical intuition. If the KCL doesn't hold at some node, then that node will either be accumulating or leaking current without restraint. And if the KVL doesn't hold across some loop, then the total energy (change in voltage per unit charge) is not conserved.

5.1. The Column Space.

Going back to the definition of column space, we'd like to somehow interpret the set of all current assignments b in $\mathbb{R}^5$ which can be the output of some voltage assignment x in $\mathbb{R}^6$ when we multiply by the adjacency matrix $A_{\mathcal{G}}$. This means, we must have

$$\begin{pmatrix} -x_1 + x_2 \\ -x_1 + x_3 \\ -x_2 + x_3 \\ -x_3 + x_4 \\ -x_5 + x_6 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix}.$$

So the basic translation of the column space $C(A_{\mathcal{G}})$ in circuit-speak becomes: *which assignment of edge-currents can arise as differences of node-voltages*? Actually, we've kind of already solved this problem when dealing with the left nullspace $N(A_{\mathcal{G}}^T)$! The relation that the edge currents must satisfy is precisely the one which comes from the basis vector $(1, -1, 1, -1, 0)$ of $N(A_{\mathcal{G}}^T)$: all we really need is $b_1 - b_2 + b_3 - b_4 = 0$. This is one equation among 5 variables, so it defines a 4-dimensional subspace of the current-space $\mathbb{R}^5$. But look: b_1 is just $-x_1 + x_2$, which equals the voltage drop across ab as we move from a to b! Similarly, b_2 is just the voltage drop across edge bc, and so forth. So, *the column space of* $A_{\mathcal{G}}$ *consists of all current assignments which respect KVL.*

Moral: The column space of $A_{\mathcal{G}}$ consists precisely of those current assignments which come from voltages satisfying Kirchoff's Voltage Law!

5.2. The Row Space.

The row space $C(A_{\mathcal{G}}^T)$ of $A_{\mathcal{G}}$ can be determined by looking directly at the transpose:

$$A_{\mathcal{G}}^T = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{ccccc} ab & ac & bd & cd & ef \end{array} \\ \begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \end{array}$$

This matrix takes currents as inputs and produces voltages as their linear combinations, so of course we now want to ask which voltages one might expect to see in the output space. More precisely, we want to see which voltage vectors $x = (x_1, \ldots, x_6)$ in $\mathbb{R}^6$ can be produced by the matrix product $A_\mathcal{G}^\mathsf{T} b$ where $b = (b_1, \ldots, b_5)$ is a vector of current assignments in $\mathbb{R}^5$. Going through the matrix multiplication gives us the following system of equations:

$$\begin{pmatrix} -b_1 - b_2 \\ b_1 - b_3 \\ b_2 - b_4 \\ b_3 + b_4 \\ b_5 \\ -b_5 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}.$$

Let's examine node b more closely: it is assigned voltage x_2, and the net current across node b is given by $b_3 - b_1$. The second components of the vector equation above say that these two quantities add to zero: $x_2 + b_3 - b_1 = 0$. This should not be surprising: it is just KCL in action! If x_2 was zero, then we'd be forced to have $b_1 = b_3$ by KCL applied to node b. So, *the row space of $A_\mathcal{G}$ consists of all node-voltages which arise from edge-currents which respect KCL.*

Why is this a four dimensional subspace of $\mathbb{R}^6$, by the way? Again, the basic insight comes from the two basis vectors of $N(A_\mathcal{G})$ that we have already computed. Here they are:

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

These vectors give us those linear combinations of our voltages which must give zero: the sums $x_1+x_2+x_4+x_4$ (from the first vector) and x_5+x_6 (from the second vector) must equal zero, and hence there are two equations in six variables. These equations must be independent (they have no variables in common!), so the row space is 4-dimensional.

Moral: The row space of the adjacency matrix corresponds to those voltage assignments at nodes which come from edge currents which respect KCL.