MODULE 2: SOLVING LINEAR SYSTEM OF EQUATIONS

1. What does linear system of equations have to do with Data Analytics?

An important class of problems that we encounter in several machine learning settings is solving a system of equations. Let's assume that you are building a predictive model of likely purchasers of a product for an e-commerce website. You have several data points about each visitor to the site. This can be age, household income, past purchase history, postal code, etc. These data points together can be considered as a vector with n components, or an n-dimensional vector. Now, if you have m such visitors – you can pack all this information in an mxn matrix. For each of these m visitors, let's assume further that we know if they are loyal users or non buyers. That is, you have a m-dimensional vector of 1s and 0s with 1 denoting a loyal user and a 0 denoting the non buyer. When you are building a model, you are finding the best m-dmensional vector x that minimizes the magnitude of following vector: Ax - b. You'll recognize this as nothing but a system of equations and a constraint to solve them.

This is what makes linear algebra so useful for all sorts machine learning and data analytics problems. Many of the tasks can be reasoned and broken down into compact linear algebra forms and familiar techniques can be applied to solve such problems.

2. Linear System of Equations

The central problem in linear algebra is to solve systems of equations. It is called linear algebra because the equations that you encounter have unknowns that are only multiplied by co-efficients and not other unknown themselves. Youll never see x times y. You may think that this is a big limitation and that these equations can not be very useful but youll be surprised how many problems can be broken down in terms of a linear system of equations and solved effectively.

Linear systems of equations extend the high-school algebra concept of several variables and equations relating them. Assume you have n variables and n equations – This is the good case and you are probably familiar from high-school algebra that these set of equations can be solved to produce values for variables that satisfy each of the n equations. A useful way to view this is to consider the equations as constraints on the variables. Each equation imposes a constraint on a set of variables. A system of equations imposes a collection of constraints that need to be solved together in order to find a solution that fits.

Perhaps an example from television marketing might be useful here. Let there be a new television series coming out in the Fall and the producer wants to target fashion-forward young women living in metros with a moderately high disposable income. You can recognize this as an SQL query from a database of TV viewers where the constraints are specified in the WHERE clause. This works reasonably well as long as the number of constraints

are small enough that they can be conveniently expressed in a WHERE clause and the data has been scrubbed properly so that TV viewers are categorized as "fashion-forward = T—F", "metro=New York, Boston, Los Angeles, San Francisco, ...".

However, often times there are many more columns per TV viewer and values are neither boolean nor categorical. Assume that the database also includes a survey on the top 50 TV series they watch with ratings, some recent clothing brands they shopped with dollar amounts. In such a situation, the constraints are not as easy to specify using SQL and there are too many of them to specify in a compact statement. However, linear system of equations can be used with a great deal of effectiveness in such situations to find and score the right cohort of TV viewers that need to be identified for marketing.

For now, lets start with two equations and two unknowns the good case. When you have as many equations as there are unknowns, it is usually a good sign, unless one of the equations is degenerate - but well get to that later. You can view each equation as imposing a constraint of the set of variables.

$$\begin{aligned} x + y &= 1 \\ 2x - y &= 2 \end{aligned} \tag{1}$$

Figure 1 shows a graphical representation of the solution for these two equations. Each of the equation defines a line in the 2-D co-ordinate system. The point at which these two lines intersect represents the solution where both these equations are satisfied. This approach is also referred to as the row-representation of the system of equations. This approach works great for 2-D but it gets tricky to visualize planes or hyperplanes in higher dimensions.

3. Linear combination of columns

Another completely equivalent representation of the above system of equations is the matrix form:

$$\begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \tag{2}$$

You can see this as a linear combination of two column vectors.

$$x \begin{bmatrix} 1 \\ 2 \end{bmatrix} + y \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \tag{3}$$

You can inspect the equation and see that we need 1 times the first column vector and 0 times the second column vector to get to our solution. Both the column-picture representation and the row-picture are completely equivalent, though you might find it easier to visualize the solution as a linear combination of column vectors rather than a point specified by a set of intersecting hyper-planes.

The above equations can be represented in the classical form: $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A}$ is the coefficient matrix and $\mathbf{b}$ is the set of constraints that are imposed on it. When you multiply a matrix with a vector, you need to take special care. You can only multiply a matrix with a vector is their dimensions are compatible. An mxn matrix A can only be multiplied with

a nx1 column vector x if the number of columns of the matrix A match the number of rows in the vector x. This is because, we are repeatedly taking dot-products between the rows of A with the column-vector x. So, the number of columns of A have to match the number of rows of x, or the dimension of vector x.

4. Solving by Elimination

When solving a system of equations, you'll have the co-efficient matrix A and the constraint vector b. You'll need to find an x that satisfies the constraints. If a solution exists, let's now find out how we can compute it.

Let's look at Eq. 1. If we multiply the first set of constraints by 2 and then subtract the 2nd equation, we get the following

$$\begin{aligned} x + y &= 1 \\ y &= 0 \end{aligned} \tag{4}$$

which immediately tells us that the solution is

$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{5}$$

The elimination procedure resulted in a new, modified A' and b' as shown below:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{6}$$

The modified A' is called the *Upper Triangular Form*. Once we have the co-efficient matrix in this form, we can systematically substitute one variable at a time and get the final solution. We can build a simple iterative procedure to systematically convert the co-efficient matrix A into the *Upper Triangular Form*. We will call this procedure *Pivot & Multiply*.

(1) Start with row 1 of the co-efficient matrix
(2) **Pivot**: The first non-zero element in the row being evaluated
(3) **Multiplier**: The element being eliminated divided by the Pivot
(4) Subtract **Multiplier** times row n from row $n+1$
(5) Advance to the next row and repeat

This works remarkably well except when you run into problems. For instance, when you advance from one row to the next, your Pivot might be zero. If so, no problem. Just simply exchange the rows (and exchange the corresponding constraint rows as well) and continue. If there is no solution to the problem, you'll end up with one or more equations that are completely eliminated and have no elements left for pivoting.

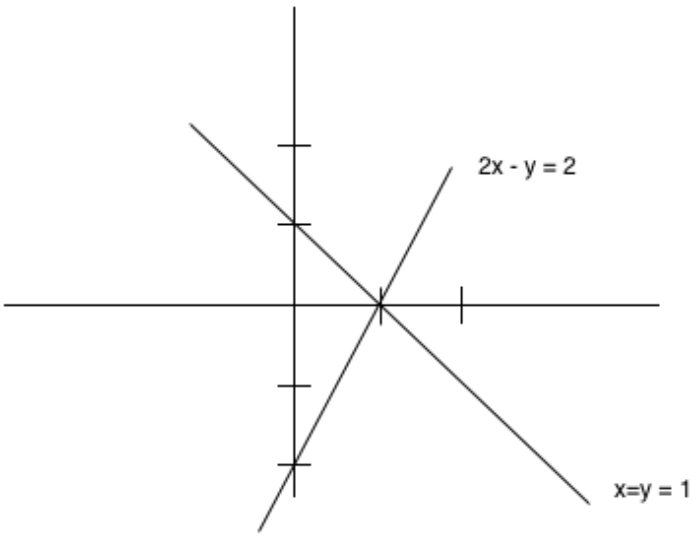

FIGURE 1. The two lines representing equations above. Point (1,0) is a solution as both lines intersect there