EECS 16A Designing Information Devices and Systems I
Spring 2015 Note 7

Lecture notes by Zhongxia Yan (02/10/2015)

Inversion

Inversion is a useful tool to solve systems of linear equations.
For example,

$$x + y = 3$$

$$2x + y = 4$$

can be rewritten as the matrix equation $\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$ by putting the coefficients of x and y (the left hand side of the equations) into the first matrix and putting the constants (the right hand side of the equations) into the last matrix.

In general, if $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and $AB = C$, then $B = A^{-1}C$, where $A^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -c \\ -b & a \end{bmatrix}$.
However, the inverse for larger matrices is much harder to calculate, so other methods for finding the inverse are needed. One such method is Gaussian Elimination, as discussed below.

Gaussian Elimination

Gaussian Elimination is a strategy to solve systems of linear equations in the form of matrices. The goal of Gaussian Elimination is to reduce a system of linear equations such as $\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$ to $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$ using valid row operations. The left-most numerical matrix must become the identity matrix, so multiplying the matrix out results in $x = a$ and $y = b$, and thus $\begin{bmatrix} a \\ b \end{bmatrix}$ is the solution to the system of linear equations.

Row operations are transformations that can be applied to any number of any row in the system of equations, one row at a time, in order to step-by-step transform the left numerical matrix to the identity matrix. Row operation only affects the numerical matrices and does **not** affect the variable matrix $\begin{bmatrix} x \\ y \end{bmatrix}$. There are three basic types of row operations: row addition, row scaling, and row exchange. Consider the general system of linear equation $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix}$, where a, b, c, d, e, and f are substitutes for **numerical** constants.

Row addition:

The first row R_1 of the system of equations is $\left[\, a\ b \,\right]\begin{bmatrix} x \\ y \end{bmatrix} = e$ and the second row R_2 is $\left[\, c\ d \,\right]\begin{bmatrix} x \\ y \end{bmatrix} = f$. The row addition operation $R_1 + R_2$ replaces row R_1 with a new row containing the sum of the numerical components of R_1 and R_2: $\left[\, a+c \quad b+d \,\right]\begin{bmatrix} x \\ y \end{bmatrix} = e+f$. The system of equations would now be:

$$\begin{bmatrix} a+c & b+d \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e+f \\ f \end{bmatrix}$$

Note that $\left[\, x\ y \,\right]^T$ does not change, but $\left[\, a\ b \,\right]$ becomes $\left[\, a+c \quad b+d \,\right]$ and e becomes $e+f$. Conversely, the operation $R_2 + R_1$ would replace R_2 and result in the system of linear equations:

$$\begin{bmatrix} a & b \\ c+a & d+b \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e \\ f+e \end{bmatrix}$$

Intuitively, one can view R_1 as $ax+by=e$ and R_2 as $cx+dy=f$. Thus, R_1+R_2 adds each side of R_2 to the corresponding side of R_1, so R_1 is now $(ax+by)+(cx+dy)=e+f$. Thus, R_1 can be written as $(a+c)x+(b+d)y=e+f$, which is equivalent to $\left[\, a+c \quad b+d \,\right]\left[\, x\ y \,\right]^T = e+f$.

Addition can also be performed with one of the rows multiplied by a scalar factor. For example, R_1+2R_2 would result in the system of equations:

$$\begin{bmatrix} a+2c & b+2d \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e+2f \\ f \end{bmatrix}$$

Again, $\left[\, x\ y \,\right]^T$ is not changed when R_2 is scaled by a factor of 2.

Row scaling:

The first row R_1 of the system of equations is $\left[\, a\ b \,\right]\begin{bmatrix} x \\ y \end{bmatrix} = e$ and the second row R_2 is $\left[\, c\ d \,\right]\begin{bmatrix} x \\ y \end{bmatrix} = f$. The row scaling operation αR_1 replaces row R_1 with a new row containing the result of the scalar multiplication of R_1 by α: $\left[\, \alpha a\ \ \alpha b \,\right]\begin{bmatrix} x \\ y \end{bmatrix} = \alpha e$. The system of equations would now be:

$$\begin{bmatrix} \alpha a & \alpha b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \alpha e \\ f \end{bmatrix}$$

Note that $\begin{bmatrix} x \\ y \end{bmatrix}$ still does not change, but $\left[\, a\ b \,\right]$ becomes $\left[\, \alpha a\ \ \alpha b \,\right]$ and e becomes αe. Also, row scaling affects only **one** row at a time; here, R_2 is not affected even thought R_1 is scaled by α. Intuitively, row scaling multiplies both sides of an equation by a scalar constant. One can view R_1 as $ax+by=e$ and αR_1 as $\alpha ax + \alpha dy = \alpha f$.

Row exchange:

The first row R_1 of the system of equations is $\left[\, a\ b \,\right]\begin{bmatrix} x \\ y \end{bmatrix} = e$ and the second row R_2 is $\left[\, c\ d \,\right]\begin{bmatrix} x \\ y \end{bmatrix} = f$. The row exchange operation $R_1 \leftrightarrow R_2$ swaps the position of the two rows in the numerical matrices. This is equivalent to setting R_1 to R_2 and R_2 to R_1. The system of equations $R_1 \leftrightarrow R_2$ after would now be:

$$\begin{bmatrix} c & d \\ a & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ e \end{bmatrix}$$

Note that $\begin{bmatrix} x \\ y \end{bmatrix}$ still does not change, but $[\ a\ b\ ]$ and $[\ c\ \ d\ ]$ switch locations, while e and f switch positions. Intuitively, the row exchange operation is equivalent to changing the physical locations of two equations on paper. For example, if equation X is above of equation Y on paper originally, equation Y is now written above equation X. Of course, changing the locations does not change the result of solving this system of equations.

Example: Using Gaussian Elimination to solve $\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$.

Our goal is to transform $\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$ into the 2 by 2 identity matrix using row operations. A good strategy is to gradually transform all of the elements below the top-left to bottom-right diagonal to 0, then transform the elements above the diagonal to 0.

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \Rightarrow R_2 - 2R_1 \Rightarrow \begin{bmatrix} 1 & 1 \\ 2-2(1) & 1-2(1) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 2-2(3) \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$$

Here we used row addition $R_2 - 2R_1$to transform the bottom left element to 0, and we will proceed to transforming the rest of the matrix to the identity matrix.

$$\Rightarrow -1(R_2) \Rightarrow \begin{bmatrix} 1 & 1 \\ 0 & -1(-1) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ -1(-2) \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

Here we have scaled R_2 by -1 to transform the bottom row of the left numerical matrix to the bottom row of the identity matrix.

$$\Rightarrow R_1 - R_2 \Rightarrow \begin{bmatrix} 1 & 1-1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3-2 \\ 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Here we have used row addition $R_1 - R_2$ to transform the entire left numerical matrix to the identity matrix. Therefore we are done, and $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

Example 2: Alternative method using row exchange to solve $\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$.

Our goal is the same. But this time we will demonstrate row exchange in the example to show how strategic row exchange may simplify calculations. This deviates from the strategy above and is purely a demonstration of row exchange.

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \Rightarrow R_1 \leftrightarrow R_2 \Rightarrow \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

Here we used row exchange $R_1 \leftrightarrow R_2$ to switch R_1 and R_2 notice that the x and y do not switch positions.

$$\Rightarrow R_1 - R_2 \Rightarrow \begin{bmatrix} 2-1 & 1-1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4-3 \\ 3 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

Here we have used row addition $R_1 - R_2$ by to transform the top row of the left numerical matrix to the top row of the identity matrix.

$$\Rightarrow R_1 - R_2 \Rightarrow \begin{bmatrix} 1 & 0 \\ 1-1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 3-1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Here we have used row addition $R_1 - R_2$ to transform the entire left numerical matrix to the identity matrix. Again, $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

Inversion with Gaussian Elimination

A square matrix M and its inverse M^{-1} will always satisfy the following conditions $MM^{-1} = I$ and $M^{-1}M = I$, where I is the identity matrix.
Let $M = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$ and $M^{-1} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$, and we want to find the values of b_{ij} to find the identity matrix.
Therefore
$MM^{-1} = I$

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Since we mathematicians are lazy, we can write the above as an **augmented matrix**, which joins the left and right numerical matrices together and hides the variable matrix, as shown below.

$$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{array}\right]$$

Now, to find the inverse matrix M^{-1} using Gaussian Elimination, we have to transform the left numerical matrix (left half of the augmented matrix) to the identity matrix, then the right numerical matrix (right half of the augmented matrix) becomes our solution. In equation form $MM^{-1} = I$, we are transforming M and I simultaneously using row operations so that the equation becomes $IM^{-1} = A$, where A is the resulting numerical matrix from the Gaussian Elimination. Since M^{-1} is multiplied by the identity matrix I, the resulting numerical matrix A must equal to M^{-1}, and we have the values for the elements in our inverse matrix. We will now do the actually computation below:

$$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{array}\right] \Rightarrow R_2 - 2R_1 \Rightarrow \left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 0 & -1 & -2 & 1 \end{array}\right] \Rightarrow -1(R_2) \Rightarrow \left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & -1 \end{array}\right]$$

$$\Rightarrow R_1 - R_2 \Rightarrow \left[\begin{array}{cc|cc} 1 & 0 & -1 & 1 \\ 0 & 1 & 2 & -1 \end{array}\right]$$

M^{-1} is the right half of the augmented matrix. Therefore $M^{-1} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix}$.

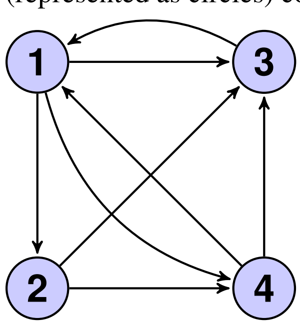
Introduction to Eigenvalues

How does Google work? Google uses a page-rank algorithm to order websites by popularity. There search engine does two things: crawling through the web of websites on the Internet and rating them according to importance of the website.

Google needs to find the most popular websites and make sure they show up first in a search by a user. Such an algorithm is integral to Google's success an popularity. One measure of popularity of a website is the number of other websites that point to the website. Not surprisingly, the page-rank algorithm utilizes linear algebra techniques to analyze these website linkages and determine which webpages are most popular.
Ideally, Google would directly measure the traffic on each website to be able to understand the most popular websites. However, since Google doesn't own all the servers that host all the websites, it can't do this. So it has to come up with a different method to *approximate* how popular different websites are. For this, it develops a model and analyzes this to generate estimated popularity.

Today, we will briefly explore a simple version of this model.

Assume that an arrow represents a link from one website to another. Suppose there are websites 1, 2, 3, 4 (represented as circles) connected to each other in the following graph:



Intuitive, the more incoming links to a website, the more popular the website is. Therefore, to rank the websites by popularity, we must find the relative number of incoming links of each website. We will use the number of people who are on a website at any given time to try to estimate the popularity of the website.

So let us attempt to rank websites by following the path of 100 users that start on page 1. We assume that every second, each user clicks on a link to jump to a new page. Let us see where these users end up.

At T_0, there are 100 people at website 1 and 0 at website 2,3, and 4.
At T_1, all users click a random link. Since website 1 has 1 link to each of the 3 other websites, the expected number of users at each of the other websites is about 33. Thus, website 1 has 0 people and website 2, 3, and 4 each has 33 people.
At T_2, all users click a random link. Half of the users at website 2 goes to 3, and the other half go to 4. Half of the users at website 4 goes to 3, and the other half goes to 1. All of the users at 3 goes to 1. Thus, website 1 has 50 people, 2 has 0 people, 3 has about 33 people, and 4 has about 17 people.

If this strategy is allowed to run ad infinitum, then the number of people at each website can reach an equilibrium. That is, the expected number of people leaving will equal the number of people entering, and the total number does not change. The amount of people who end up at each website determines the relative "importance score" of website. For example, in our run with 3 clicks, website 1 > website 3 > website 4 > website 2 in number of people at the end. Although more clicks would be needed to attain accurate results, the simulated number of people that end up at each website already begins to reflect the actual number of links to the website. Website 1, 3, and 4 all have more people than website 2, which has the least number of people and the least number of incoming links.

This strategy can be represented as a matrix multiplication. More specifically, a probability matrix can be made containing the chances of reaching every website from every other website. One such probability matrix (the probability matrix for the simulation above) is demonstrated below:

$$\begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} p_{1i} \\ p_{2i} \\ p_{3i} \\ p_{4i} \end{bmatrix} = \begin{bmatrix} p_{1f} \\ p_{2f} \\ p_{3f} \\ p_{4f} \end{bmatrix}$$

The probability matrix is the left-most 4 x 4 matrix. Column i represents the probability of click to each of the other websites from website i. For example, the first column represents the probability of reaching other websites from website 1: 0 to itself and $\frac{1}{3}$ to 2, 3, and 4 each. On the other hand, row j represents the probability of website j being reached from the other websites. For example, the third row implies that website 3 can be reached from website 1 one third ($\frac{1}{3}$) of the time, from website 2 one half ($\frac{1}{2}$) of the time, from itself 0 of the time, and from website 4 one half ($\frac{1}{2}$) of the time.

The middle matrix represents the percentage of people at each website before the matrix multiplication. For example, we could set each of p_{1i}, p_{2i}, p_{3i}, p_{4i} to $\frac{1}{4}$ on the first turn, before any user clicked any websites. The i represents initial.This is equivalent to putting $\frac{1}{4}$the of the users on each website initially.

The right-most matrix represents the percentage of people at each website after the matrix multiplication (equivalent to the result of random clicking). This depends on the values within the probability matrix and the initial percentage of users at each website. The f in the subscript represents final.

To perform the simulation like before, each multiplication is equivalent to a round of clicking for the users. After each round, the percentage of users change. To continue to the next round, we merely take the final user percentage matrix (right-most matrix) and use it as the initial user percentage matrix (middle matrix) for the next round. Repeated multiplication of these matrices is a fast way to calculate the equilibrium percentage of people at each website. This will allow the page-rank algorithm to rank the websites from most popular to least popular.