Using the concept of Google's PageRank to enhance student motivation in mathematics

Francisco Pedroche Sánchez
Department of Applied Mathematics,
Universitat Politècnica de València,
Camí de Vera s/n 46022, Valencia, Spain
pedroche@mat.upv.es

Abstract - **The fact that the search engine Google™ is a popular tool known by our students can be used to catch their attention in the classroom. Google recognizes that the PageRank™ algorithm is still the basic tool of the engine. In fact, this algorithm has already caught the interest of the researchers in mathematics and it is an active field of investigation. In this paper it is shown how to use the key-concept of the search engine Google, namely the PageRank, to illustrate some topics of mathematics. In order to explain how this algorithm works one has to deal with the following items: random process, random walk, conditional probability, total probability theorem, Markov chain, incidence matrix, stochastic matrix, eigenvalues, eigenvectors, vector space, spectral radius, matrix product, directed graph, stationary state, numerical methods and iterative methods for linear systems of equations. This communication describes how to introduce these concepts in a very suggestive way and connected with the topic of Google's PageRank.**

Index Terms – Iterative methods, matrix analysis, mathematical modeling, PageRank, probability.

INTRODUCTION

The concept of crawling the internet is a common term for our students. For this reason, it is normal that when we use this term in the classroom, the students feel that the teacher is going to explain something familiar for them; something connected with their world made up with internet-related terms like myspace.com, messenger.com, secondlife.com or popomundo.com. When you talk about Google, they use to think: this person is going to say something interesting! In fact, some of us have first noticed the existence of Google in the late 90's when our students told us that there was a fantastic searcher called that way. Some years later, in the course of a research project about Markov chains, we became in contact with the theoretical foundations of Google and begun to incorporate some of the concepts of Google as a part of the material used as examples in the classroom. In this communication we try to put in order some of the ideas that we have been using in the classroom for some years. The aim of this paper is to show how these ideas can be used to illustrate real applications of mathematics to students of mathematics, computer science or engineering programs. Since the purpose of this paper is to focus on Google's concepts we have omitted rigorous definitions of the terms involved letting this task to the teachers interested in using these notes in their classrooms. The paper is structured as follows. Each section is named after the main concept that it is treated therein. The aim in each section is to show the connection of each topic with the main algorithm of Google: the PageRank algorithm [1] that computes a vector holding a global measure of importance for each web page.

SYSTEMS OF LINEAR EQUATIONS

Let us consider a system of n linear equations with n unknowns of the form

$$\left.\begin{array}{cccc} a_{11}x_1 + & a_{12}x_2 + & \cdots & +a_{1n}x_n = b_1 \\ a_{21}x_1 + & a_{22}x_2 + & \cdots & +a_{2n}x_n = b_2 \\ \vdots & \vdots & & \vdots \\ a_{n1}x_1 + & a_{n2}x_2 + & \cdots & +a_{nn}x_n = b_n \end{array}\right\} \quad (1)$$

where a_{ij}, x_i and b_i are real numbers for all the values of the indices i and j.

The engineers at Google have to deal with a system of linear equations which size is that of the web that is crawled by Google. That is to say they have to handle systems of the order of billions of equations with billions of unknowns. How do they manage to handle this amazing number of equations? Indeed, this problem has been addressed as the largest matrix computation ever made [2]. Obviously, the first thing that we need to do is to use an easy notation. Try to imagine that we have to write explicitly (1) with billions of equations! That is nonsense. With the aid of the concepts of matrix, column vector (a particular type of matrix) and product of two matrices we can write (1) in the very short form

$$Ax = b\,, \quad (2)$$

where we have denoted by A the matrix of the system- an ordered array of n^2 elements with a_{ij} laying on the intersection of the ith row and the jth column of A- and by x and b the column vector of unknowns and the column vector of the right hand side of (1), respectively. Therefore we have now a useful notation to treat, at least theoretically, this enormous system of equations. In the following sections we see how a system like (2) can be associated with Google's PageRank and we also will show how to solve this problem.

CONNECTIVITY MATRIX

In order to illustrate how to construct a matrix associated with a set of web pages crawled by Google, let us consider the Figure 1. In this figure four web pages with their outgoing links (outlinks) are shown. The destination of each link is also indicated. For example, in page 3 there are outlinks that go to pages 1 and 2.

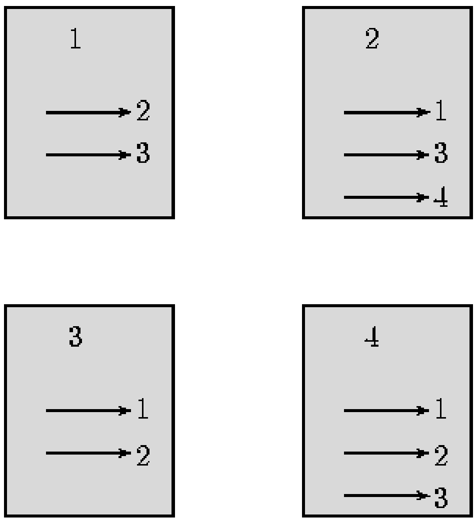

FIGURE 1
A WEB CONSISTING OF FOUR PAGES SHOWING THEIR OUTLINKS.

How can we describe the structure of the links between these four pages? One way to do that is to define what is called the directed graph associated with the set of connections. In fact, this graph is just a schematically way of depicting the same information that is exhibited in Figure 1. In the present case, the directed graph associated with the connections of the web shown in Figure 1 is shown in Figure 2. The pages have been substituted by numbers called vertexes.

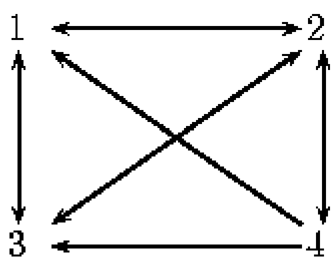

FIGURE 2
DIRECTED GRAPH ASSOCIATED WITH THE SET OF PAGES OF FIGURE 1.

We have gained clarity in our treatment since it is easier to draw Figure 2 than Figure 1. Now we can define a matrix, called connectivity matrix, which summarizes all the information in Figure 2. We define the connectivity (or hyperlink) matrix associated with a directed graph as the square matrix with elements g_{ij} such that they have the value 1 if there is a connection from page j to page i, with $i \neq j$, and 0 otherwise. Therefore the connectivity matrix for the directed graph in Figure 2 is

$$G = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \qquad (3)$$

Now we have a matrix to illustrate the connections between the web pages. Connectivity matrices are nonnegative matrices since their elements are all nonnegative. These matrices are also called sparse matrices since they use to have a great percentage of zeros. There are special procedures to work with sparse matrices; see [3].

In order to show how Google constructs a matrix *A* to form a system of equations like (2) we need first to make some comments about random walks and Markov processes. We do this in the next sections.

RANDOM WALK

In this section we present the classic model of a random walk, also called 'drunkard's walk'. Let us assume that a person is walking on a line with *n* discrete positions according to the following rule: The walker tosses a coin and if it is head he makes a step to the right and otherwise he makes a step to the left; see Figure 3.

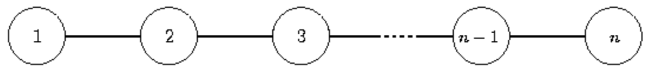

FIGURE 3
A LINE SHOWING DISCRETE POSITIONS.

We also admit that when the person reaches an extreme of the line then in the next movement he turns back, independently of the outcome of the flip. We are interested in describing the movement of this walker. In particular, when the person begins at position marked as '1', what can we say about the position (or state) of the walker when he has taken some steps? We are in front of a random problem. We can not give a deterministic answer to this problem. To fix ideas, let Δt be a time interval and let us describe the time by the expression

$$t = k\,\Delta t, \;\; k = 0, 1, 2, \ldots \qquad (4)$$

and let us denote by $p_i(k)$ the probability of finding the walker in the position i on the line of Figure 3, at instant time given by k. In order to describe the state of the walker in a certain time k we construct the probability distribution vector v_k, which is a nonnegative vector such that the sum of its components is one, and it is given by the expression

$$v_k = \begin{bmatrix} p_1(k) \\ p_2(k) \\ \vdots \\ p_n(k) \end{bmatrix} . \qquad (5)$$

For example, if we start the experiment at $t = 0$, with the walker at position '1' of the line depicted in Figure 3 we have that the initial state of the walker is given by

$$v_0 = \begin{bmatrix} p_1(0) \\ p_2(0) \\ \vdots \\ p_n(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} , \qquad (6)$$

since we know that the walker is at position '1' and no uncertainty is assumed. In the following throwing of the coin we know, since we have imposed this rule, that the walker will go to the position denoted by '2'. Therefore it is clear that

$$v_1 = \begin{bmatrix} p_1(1) \\ p_2(1) \\ \vdots \\ p_n(1) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} . \qquad (7)$$

In the next movement we have uncertainty since the walker can go to the left or to the right with the same probability. Therefore, it is clear that

$$v_2 = \begin{bmatrix} p_1(2) \\ p_2(2) \\ p_3(2) \\ \vdots \\ p_n(2) \end{bmatrix} = \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \\ \vdots \\ 0 \end{bmatrix} . \qquad (8)$$

We remark that the position of the walker given at any time described by k is given by a state vector v_k as in (5). The vector v_0 is called the initial state vector. Observe that in each consecutive movement we are distributing the probabilities and this allows us to compute the state vector v_k at any time. But if we want to describe the long run, e.g. the state given by $k = 1000$, it would be very tedious to compute all the intermediate states to reach this state! How can we manage to solve this situation? We show this in the following section where we introduce Markov processes.

MARKOV CHAINS

The concepts of matrices and product of a matrix by a vector will help us easily describe some computations. First, let us define a matrix *P*, called a transition matrix, associated with the random walk seen in the previous section. Let $P = (p_{ij})$ where p_{ij} is the probability that the walker goes to the position i at time $k+1$ given that he was at position j at time k. In our example it is easy to show that

$$p_{ij} = \begin{cases} 1/2 & if \quad i = j-1,\ j = 2,3,\ldots,n-1 \\ 1/2 & if \quad i = j+1,\ j = 2,3,\ldots,n-1 \\ 1 & if \quad i = 2,\ j = 1 \\ 1 & if \quad i = n-1,\ j = n \\ 0 & otherwise \end{cases} \qquad (9)$$

and this can be written in matrix form as follows

$$P = \begin{bmatrix} 0 & 1/2 & & & & \\ 1 & 0 & 1/2 & & & \\ & 1/2 & 0 & 1/2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1/2 & 0 & 1 \\ & & & & 1/2 & 0 \end{bmatrix} . \qquad (10)$$

Where p_{ij} lies in the position given by row i and column j. Note that this matrix is sparse. Since it is nonnegative and each column sum equals one, it is called a (column) stochastic matrix. Now it is easy to see -and here as in (2) we have a good opportunity to motivate the somewhat strange form of the concept of product of matrices- that we have

$$v_1 = Pv_0, \quad v_2 = Pv_1, \quad v_3 = Pv_2. \qquad (11)$$

We note now that

$$v_3 = Pv_2 = PPv_1 = P^2 v_1 = P^3 v_0. \qquad (12)$$

Here the teacher may explain the notation for the product of matrices and its properties. In our example, it is easy to show that for any instant time given by k the following equations hold

$$v_k = Pv_{k-1} \quad k = 1, 2, \ldots \qquad (13)$$

$$v_k = P^k v_0 \quad k = 1, 2, \dots \tag{14}$$

and (14) implies that the state vector at the instant time k can be computed just using the initial state vector and the rules of our walker, which are given by matrix P. The states given by (13) or (14) are said to form a Markov chain or define a Markov process; see, for example [4] for details.

Let us consider a random walk on a line with only four permitted positions, i.e., $n = 4$, with the notation of Figure 3. In this case the transition matrix is, according to (10)

$$P = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1 \\ 0 & 0 & 1/2 & 0 \end{bmatrix} \quad . \tag{15}$$

Let us assume that the walker begins its walk at the position '1', i.e., v_0 has the form given by (6). We are interested, for example, in the instant of time given by $k = 10$. What can be say about the walker? An easy computation (here we recommend to use some software such as Matlab™, Derive™ or Mathematica™) shows that

$$v_{10} = P^{10}\, v_0 = \begin{bmatrix} 1/3 \\ 0 \\ 2/3 \\ 0 \end{bmatrix}, \tag{16}$$

which means that, for $k = 10$, the probability to find the walker at position '1' is $1/3$, the probability to find him at position '3' is $2/3$, and the rest of probabilities are zero. In conclusion, the location most likely to find the walker is position '3', when $k = 10$; In other words, at this time, the most important position is '3'. In a similar way, for a set of web pages, Google assumes that the most important web page is that with the biggest probability to find a random walker in it [5].

We remark here that matrix P, given by (10), does not depend on the time, i.e., it does not depend on k. This means, in other words, that our walker has always the same rules. In these cases it is said that the Markov chain is homogeneous, since it is not a function of time. We are now interested in the following topics:

1. Whether there exists a state vector, called stationary state vector, such that

$$v_{est} = P\, v_{est}\, . \tag{17}$$

2. Whether there exits a limit state vector v_∞, such that

$$v_\infty = \lim_{k\to\infty} P^k\; v_0 . \tag{18}$$

3. Whether v_{est} and v_∞ are equal.

In the example above it is easy to see that

$$v_{est} = \begin{bmatrix} 1/6 \\ 1/3 \\ 1/3 \\ 1/6 \end{bmatrix}, \tag{19}$$

and it is a stationary state vector since it satisfies (17) where P is given by (15). Here the teacher can suggests the following homework for the students: to show that this system oscillates, when k is large enough (taking k=15 the phenomenon is already observed) between the states

$$v_m = \begin{bmatrix} 1/3 \\ 0 \\ 2/3 \\ 1/6 \end{bmatrix}, \quad v_{m+1} = \begin{bmatrix} 0 \\ 2/3 \\ 0 \\ 1/3 \end{bmatrix} \quad . \tag{20}$$

We remark that (19) means that in our example we have that the vector v_∞ does not exist when v_0 is given by (6). Therefore, this is an example showing that v_∞ and v_0 are not equal. However, the PageRank vector is a vector that satisfies $v_{est} = v_\infty$. Moreover, the matrix P for the Google problem has another interesting property: the equality $v_{est} = v_\infty$ is satisfied for any choice of the initial state vector v_0. The PageRank vector gives the long run probability to find a walker (a web surfer) in each web page. In the next section we focus on how the Google matrix is constructed.

THE GOOGLE MATRIX

We have already seen how to associate a connectivity matrix G to a set of pages. Now, we show how to modify G to obtain a stochastic matrix P. Given a matrix $G = (g_{ij})$ we define the quantities

$$c_j = \sum_{i=1}^{n} g_{ij}, \;\; 1 \le j \le n \tag{21}$$

and the stochastic matrix $P = (p_{ij})$ given by

$$p_{ij} = \begin{cases} g_{ij}/c_j & if \;\; c_j \ne 0 \\ 0 & otherwise \end{cases} \qquad 1 \le i, j \le n \tag{22}$$

In our example, where G is given by (3), we obtain

$$P = \begin{pmatrix} 0 & 1/3 & 1/2 & 1/3 \\ 1/2 & 0 & 1/2 & 1/3 \\ 1/2 & 1/3 & 0 & 1/3 \\ 0 & 1/3 & 0 & 0 \end{pmatrix}. \qquad (23)$$

Matrix P given by (23) is called the Google matrix of the web considered.

GOOGLE'S PAGERANK

We have seen that the PageRank vector is a vector that satisfies $v_{est} = v_\infty$ and this holds for any choice of the initial state vector v_0. Therefore, if we denote by x the PageRank vector we have, from (17), that

$$Px = x \ . \qquad (24)$$

Note that this is a particular case of the eigenvector problem that we study in a first course of university. We recall that given a square matrix A, the nonzero vectors x that satisfy the equation

$$A\,x = \lambda\,x \ , \qquad (25)$$

with λ real (or complex) numbers, are called eigenvectors of A associated with λ. Therefore, the computation of the PageRank vector is an eigenvector problem! In fact, the PageRank vector is a nonnegative eigenvector corresponding to the eigenvalue $\lambda = 1$ and such that the sum of its components is one. In general, the Google matrix P given by (22) will not have the desired properties and some minor modifications have to be made in order to ensure that the matrix admits an eigenvector with the properties mentioned above. The Perron-Frobenius theorem is the base to perform these minor modifications. In more detail, when P is stochastic and primitive (nonnegative, irreducible[1] and $\lambda = 1$ is the only eigenvalue which norm is the spectral radius[2]) then the PageRank vector exists. In this case it is said that the Markov chain is ergodic. See, [6] for details.

A LINEAR SYSTEM FOR GOOGLE

Some authors [5], [7] showed that the PageRank problem can be formulated as the following linear problem

$$(I - \alpha P)y = v \quad . \qquad (26)$$

Where v is a probability distribution vector and $0 < \alpha < 1$. Once the unknown y has been solved the PageRank vector x is then computed using the expression

$$x = \frac{y}{e^T y} \quad , \qquad (27)$$

where $e^T = [1 \ \ 1 \ \ \cdots \ \ 1]$. The linear system (26) is formally a system like (1) with the coefficient matrix $A = I - \alpha P$. This matrix results to be a nonsingular M-matrix [6] which implies that some known iterative methods for linear systems can be applied for solving (26), see e.g. [8]. In the next section we introduce these kinds of methods.

ITERATIVE METHODS

We have shown that the PageRank problem can be formulated either as (24) if we consider the problem as an eigenvalue problem, or as a linear system like (26). When we compute eigenvectors in the classroom we use matrices of order 3 or 4 and we do the computations by hand. Nevertheless, when we have a matrix of a remarkable size (up to billions) we can not use the direct methods that we show in the classroom. Not even with a computer a direct method would be useful! We have to deal with iterative methods. An easy way to introduce the nature of the iterative methods is the following. Let us consider the equation with one unknown

$$3x = 6 \ . \qquad (28)$$

Clearly, the solution is $x = \frac{1}{3} 6 = 2$. Let us imagine that we don't know how to compute the fraction $1/3$ and thus we have to invent a method to solve (28) by inspection, i.e., we may give arbitrary values to x in the hope that we chose the exact value that satisfies (28). This method has the disadvantage that we have infinite values of x to prove. Obviously, in the case of having a system of n linear equations the possibility of finding a solution by inspection is very small. Let us suppose, instead, that we do know how to compute the fraction $1/5$ and we do the following: we split the number 3 as $3 = 5 - 2$ and therefore (28) can be written as

$$(5-2)x = 6 \,, \qquad (29)$$

and this allows to write

$$x = \frac{1}{5} 2x + \frac{1}{5} 6 \,, \qquad (30)$$

[1] A matrix is irreducible if it associated directed graph is such that one can go from any vertex to any other vertex, perhaps in several steps.
[2] The spectral radius is the maximum of the absolute values of the eigenvalues

and now we apply the following: a random value of x would not probably solve this equation and therefore the right hand side of (30) would be different than its left hand side. Let us denote by x_0 the initial guess that we prove on the right hand side of (30) and by x_1 the value that we obtain in the left hand side, i.e.,

$$x_1 = \frac{1}{5} 2x_0 + \frac{1}{5} 6 \quad . \tag{31}$$

We want to fulfill (31) with $x_1 = x_0$. Since we shall prove a random value of x_0 it is not likely that $x_1 = x_0$. Therefore we will have to repeat this method but taking now as a guess the value of x_1. In conclusion we formally have to deal with the successive approximations

$$x_{k+1} = \frac{2}{5} x_k + \frac{6}{5}, \quad k = 0, 1, 2, \ldots \tag{32}$$

This expression is called an iterative scheme. In the Table 1 we show the values for the successive x_k when we begin with x_0 =1.0.

TABLE I

VALUES FOR x_k GIVEN BY (32)

k	x_k
0	1.000
1	1.600
2	1.840
3	1.936
4	1.974
5	1.990
6	1.996
7	1.998
8	1.999

In Table 1 it is shown that x_k converges to 2.0 as k becomes greater. In this point the teacher may explain the difference between convergent and divergent methods and the theorems related. In the case of linear systems of the form $Ax = b$ the convergence of the iterative methods can be studied in terms of the splitting of matrix A in the form $A = M - N$ where M is a nonsingular matrix, and considering the iterative scheme

$$x_{k+1} = M^{-1} N x_k + M^{-1} b, \quad k = 0,1,2,\ldots \tag{33}$$

It is known [6] that when A is nonsingular the convergence of this scheme it is guaranteed when the spectral radius of the iteration matrix $M^{-1}N$ is less than 1. When the coefficient matrix A is a singular matrix (as in the case of Google) the convergence conditions are more complicated and require, among other conditions, that the spectral radius of $M^{-1}N$ be less or equal than 1; see [6], [8].

The classical method of solving the eigenvalue problem (23) in an iterative way is by using the scheme

$$x_{k+1} = P x_k \quad k = 0, 1, 2, \ldots \tag{34}$$

which is called *Power method* and is the method that Google uses to compute the PageRank vector; see [1], [5] and [9] for details.

CONCLUSIONS

Some concepts related with Google's PageRank has been shown. In each section we have motivated the terms introduced mainly relating them with the performing of the inner algorithm of the Google searcher. In our experience, these sorts of explanations help catch the attention of our students increasing their participation in the learning process. These concepts also allow to illustrate that linear algebra is a broadly applicable branch of mathematics and reveal some connections between separate parts of mathematics.

REFERENCES

[1] Page,L., Brin, S., Motwani, R. and Winograd, T. "The PageRank Citation Ranking: Bringing Order to the Web". *Stanford Digital Library Technologies Project*. 1999.

[2] Moler, C. B. "The World's Largest Matrix Computation. Google's PageRank is an eigenvector of a matrix of order 2.7 billion". *MATLAB News \& Notes.* 2002.

[3] Barrett, R., Berry, M. W., Chan, T. F., Demmel, J., Donato, J. *et al*, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, Pennsylvania, USA. 1994.

[4] Nelson, R. *Probability, stochastic processes, and queueing theory*, Springer-Verlag, Ney York, USA. 1995.

[5] Langville, A. N. and Meyer, C. D. "Deeper inside PageRank". *Internet Mathematics,* Vol 1, No. 3, 2005, pp. 335-380.

[6] Berman, A. and Plemmons, R. J. *Nonnegative matrices in the mathematical sciences*, SIAM, Philadelphia, Pennsylvania, USA. 1994.

[7] Del Corso, G. M., Gulli, A. and Romani, F. "Fast PageRank computation via a sparse linear system", *Lecture notes in computer science,*No 3243, 2004, pp. 118-130.

[8] Bru, R., Pedroche, F. and Szyld, D. B. "Cálculo del vector PageRank de Google mediante el método aditivo de Schwarz". *Congreso de Métodos Numéricos en Ingeniería* 2005. J.L. Pérez Aparicio *et al* (ed.), Granada, España. 2005. p. 263.

[9] Pedroche, F. " Métodos de cálculo del vector PageRank". *Bol. Soc. Esp. Mat. Apl.* (2007). To appear.