

# Google matrix

---

A **Google matrix** is a particular stochastic matrix that is used by Google's PageRank algorithm. The matrix represents a graph with edges representing links between pages. The rank of each page can be generated iteratively from the Google matrix using the power method. However, in order for the power method to converge, the matrix must be stochastic, irreducible and aperiodic.

## ***H* matrix**

In order to generate the Google matrix, we must first generate a matrix  $H$  representing the relations between pages or nodes.

Assuming there are  $n$  pages, we can fill out  $H$  by doing the following:

1. Fill in each entry  $(i, j)$  with a 1 if node  $i$  has a link to node  $j$ , and 0 otherwise; this is the adjacency matrix of links.
2. Divide each row by  $k_i$  where  $k_i$  is the total number of links to other pages from node  $i$ . The matrix  $H$  is usually not stochastic, irreducible, or aperiodic, that makes it unsuitable for the PageRank algorithm.

## ***G* matrix**

Given  $H$ , we can generate  $G$  by making  $H$  stochastic, irreducible, and aperiodic.

We can first generate the stochastic matrix  $S$  from  $H$  by adding an edge from every sink state to every other node. In the case where there is only one sink state  $S$ , written as:

$$S = H + \left( \frac{1}{N} e e^T \right)$$

where  $a$  is the number of nodes  $N$ .

Then, by creating a relation between nodes without a relation with a factor of  $\alpha$ , the matrix will become irreducible. By making  $H$  irreducible, we are also making it aperiodic.

The final Google matrix  $G$  can be computed as:

$$G = \alpha S + (1 - \alpha) \frac{1}{N} e e^T \quad (1)$$

By the construction the sum of all non-negative elements inside each matrix column is equal to unit. If combined with the  $H$  computed above and with the assumption of a single sink node, the Google matrix can be computed as:

$$G = \alpha H + (\alpha a + (1 - \alpha)e) \frac{1}{N} e^T.$$

Although  $G$  is a dense matrix, it is computable using  $H$  which is a sparse matrix. Usually for modern directed networks the matrix  $H$  has only about ten nonzero elements in a line, thus only about  $10N$  multiplications are needed to multiply a vector by matrix  $G$  [1,2].

For the actual matrix, Google uses a damping factor  $\alpha$  around 0.85 [1,2,3]. The term  $(1 - \alpha)$  gives a surfer probability to jump randomly on any page. The matrix  $G$  belongs to the class of Perron-Frobenius operators of Markov chains [1].

---

## Spectrum and eigenstates of $G$ matrix

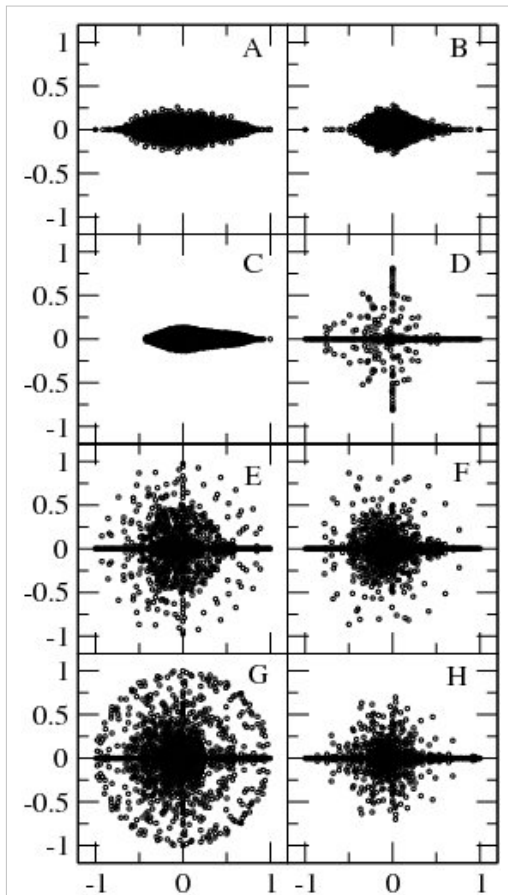


Fig1. Distribution of eigenvalues  $\lambda_i$  of Google matrices in the complex plane at  $\alpha = 1$  for dictionary networks: Roget (A, N=1022), ODLIS (B, N=2909) and FOLDOC (C, N=13356); UK university WWW networks: University of Wales (Cardiff) (D, N=2778), Birmingham City University (E, N=10631), Keele University (Staffordshire) (F, N=11437), Nottingham Trent University (G, N=12660), Liverpool John Moores University (H, N=13578)(data for universities are for 2002) (from [7])

For  $0 < \alpha < 1$  there is only one maximal eigenvalue  $\lambda = 1$  with the corresponding right eigenvector which has non-negative elements  $P_i$  which can be viewed as stationary probability distribution [1]. These probabilities ordered by their decreasing values give the PageRank vector  $P_i$  with the Ragerank  $K_i$  used by Google search to rank webpages. Usually one has for the World Wide Web that  $P \propto 1/K^\beta$  with  $\beta \approx 0.9$ . The number of nodes with a given PageRank value scales as  $N_P \propto 1/P^\nu$  with the exponent  $\nu = 1 + 1/\beta \approx 2.1$ [4,5]. The left eigenvector at  $\lambda = 1$  has constant matrix elements. With  $0 < \alpha$  all eigenvalues move as  $\lambda_i \rightarrow \alpha\lambda_i$  except the maximal eigenvalue  $\lambda = 1$ , which remains unchanged [1]. The PageRank vector varies with  $\alpha$  but other eigenvectors with  $\lambda_i < 1$  remain unchanged due to their orthogonality to the constant left vector at  $\lambda = 1$ . The gap between  $\lambda = 1$  and other eigenvalue is  $1 - \alpha \approx 0.15$  gives a rapid convergence of a random initial vector to the PageRank approximately after 50 multiplications on  $G$  matrix. At  $\alpha = 1$  the matrix  $G$  has generally many degenerate eigenvalues  $\lambda = 1$  (see e.g. [6,7]). An example of the eigenvalue spectrum of the Google matrix of various directed networks is shown in Fig.1 taken from [7].

The Google matrix can be also constructed for the Ulam networks generated by the Ulam method [8] for dynamical maps. The spectral properties of such matrices are discussed in [9,10,11,12,13]. In a number of cases the spectrum is

described by the fractal Weyl law [10,12].

The Google matrix can be constructed also for other directed networks, e.g. for the procedure call network of the Linux Kernel software introduced in [14]. In this case the spectrum of  $\lambda$  is described by the fractal Weyl law with the fractal dimension  $d \approx 1.3$  (see Fig.2 from [15]). Numerical analysis shows that the eigenstates of matrix  $G$  are localized (see Fig.3 from [15]). Arnoldi iteration method allows to compute many eigenvalues and eigenvectors for matrices of rather large size [13,15].

Other examples of  $G$  matrix include the Google matrix of brain [16] and business process management [17].

## Historical notes

The Google matrix with damping factor was described by Sergey Brin and Larry Page in 1998 [18], see also articles PageRank and [19].

## See also

- PageRank, CheiRank
- Arnoldi iteration
- Markov chain, Transfer operator, Perron–Frobenius theorem
- Web search engines

## References

- [1] Langville, Amity N; Carl Meyer (2006). *Google's PageRank and Beyond*. Princeton University Press. ISBN 0-691-12202-4.
- [2] Austin, David (2008). "How Google Finds Your Needle in the Web's Haystack" (<http://www.ams.org/samplings/feature-column/fcarc-pagerank>). AMS Feature Columns
- [3] Law, Edith (2008). "PageRank" ([http://scienceoftheweb.org/15-396/lectures/PageRank\\_Lecture12.pdf](http://scienceoftheweb.org/15-396/lectures/PageRank_Lecture12.pdf)) (PDF)
- [4] Donato D.; Laura L., Leonardi S., Millozzi S. (2004). "Large scale properties of the Webgraph". *Eur. Phys. J. B* v.38, p.239
- [5] Pandurangan G.; Raghavan P., Upfal E. (2005). "Using PageRank to Characterize Web Structure". *Internet Math.* v.3, p. 1
- [6] Serra-Capizzano, Stefano (2005). "Jordan Canonical Form of the Google Matrix: a Potential Contribution to the PageRank Computatin". *SIAM J. Matrix. Anal. Appl.* v.27, p.305
- [7] Georgeot B.; Giraud O., Shepelyansky D.L. (2010). "Spectral properties of the Google matrix of the World Wide Web and other directed networks". *Phys. Rev. E* v.81, p.056109

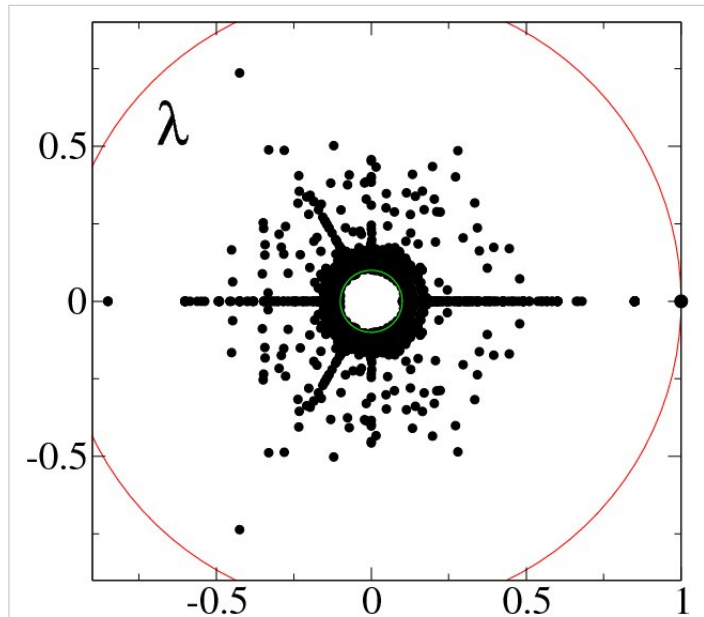


Fig2. Distribution of eigenvalues  $\lambda$  in the complex plane for the Google matrix  $G$  of the Linux Kernel version 2.6.32 with matrix size  $N = 285509$  at  $\alpha = 0.85$ , unit circle is shown by solid curve (from [15])

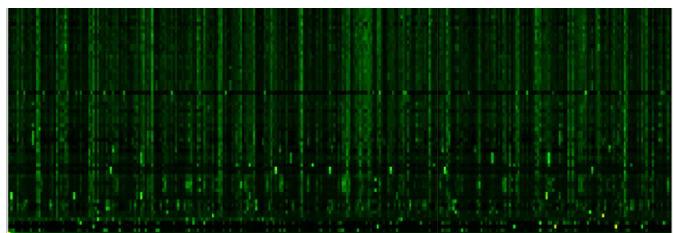


Fig.3 Coarse-grained probability distribution for the eigenstates of the Google matrix for Linux Kernel version 2.6.32. The horizontal lines show the first 64 eigenvectors ordered vertically by  $|\lambda_i|$  (from [15])

- [8] Ulam, Stanislaw (1960). "A Collection of mathematical problems, Interscience tracs in pure and applied mathematics". Interscience, New York p.73
  - [9] Froyland G.; Padberg K. (2009). "Almost-invariant sets and invariant manifolds — Connecting probabilistic and geometric descriptions of coherent structures in flows". *Physica D* v.238, p.1507
  - [10] Shepelyansky D.L.; Zhirov O.V. (2010). "Google matrix, dynamical attractors and Ulam networks". *Phys. Rev. E* v.81, p.036213
  - [11] Ermann L.; Shepelyansky D.L. (2010). "Google matrix and Ulam networks of intermittency maps". *Phys. Rev. E* v.81, p.036221
  - [12] Ermann L.; Shepelyansky D.L. (2010). "Ulam method and fractal Weyl law for Perron-Frobenius operators". *Eur. Phys. J. B* v.75, p.299
  - [13] Frahm K.M.; Shepelyansky D.L. (2010). "Ulam method for the Chirikov standard map". *Eur. Phys. J. B* v.76, p.57
  - [14] Chepelianskii, Alexei D. (2010). "Towards physical laws for software architecture" (<http://arxiv.org/abs/1003.5455>). arXiv:1003.5455
  - [15] Ermann L.; Chepelianskii A.D., Shepelyansky D.L. (2011). "Fractal Weyl law for Linux Kernel Architecture". *Eur. Phys. J. B* v.79, p.115 (arxiv:1005.1395[cs.CE])
  - [16] Shepelyansky D.L.; Zhirov O.V. (2010). "Towards Google matrix of brain". *Phys. Lett. A* v.374, p.3206
  - [17] Abel M.; Shepelyansky D.L. (2011). "Google matrix of business process management". *Eur. Phys. J. B* (to appear) arxiv:1009.2631[cs.CY]
  - [18] Brin S.; Page L. (1998). "The anatomy of a large-scale hypertextual Web search engine". *Computer Networks and ISDN Systems* v.30, p.107
  - [19] Franceschet, Massimo (2010). "PageRank: Standing on the shoulders of giants" (<http://arxiv.org/abs/1002.2858>). arXiv:1002.2858
-

# Article Sources and Contributors

**Google matrix** *Source:* <http://en.wikipedia.org/w/index.php?oldid=409015647> *Contributors:* Bkkbrad, C4andrei, Cmupws, Cunard, DesolateZero, Melcombe, Michael Hardy, NellieBly, Psantora, Rjwilmsi, Rkosuru, Rpyle731, Shepelyansky, Vrwarp, 8 anonymous edits

## Image Sources, Licenses and Contributors

**Image:Googlematrix1.jpg** *Source:* <http://en.wikipedia.org/w/index.php?title=File:Googlematrix1.jpg> *License:* Public Domain *Contributors:* User:Shepelyansky

**Image:Googlematrix2.jpg** *Source:* <http://en.wikipedia.org/w/index.php?title=File:Googlematrix2.jpg> *License:* Creative Commons Zero *Contributors:* L.Ermann, A.D.Chepelianskii and D.L.Shepelyansky

**Image:Googlematrix3.gif** *Source:* <http://en.wikipedia.org/w/index.php?title=File:Googlematrix3.gif> *License:* Creative Commons Zero *Contributors:* User:Shepelyansky

## License

---

Creative Commons Attribution-Share Alike 3.0 Unported  
<http://creativecommons.org/licenses/by-sa/3.0/>

---