PageRank and Markov chains

MVE055 / MSG810

- **Problem**: Given n interlinked webpages, rank them in order of "'importance".
- **Key insight**: many links from important websites pointing to a website indicate importance.

Google invented the PageRank algorithm (named after Larry Page, one of the founders of Google.)

Quote: "PageRank can be thought of as a model of user behaviour. We assume there is a 'random surfer' who is given a web page at random and keeps clicking on links, never hitting 'back' but eventually gets bored and starts on another random page."

We want to use this together with Markov chains to assign an importance weight to each page, just like Google does (for example to give the most important results to a search query first.)

The link relationship of a network of pages can be represented by the "adjacency" matrix, the matrix with entries

$$A_{ij} = \begin{cases} 1 & \text{if site } i \text{ links to site } j \\ 0 & \text{else.} \end{cases}$$

For a web with n pages, construct the $n \times n$ PageRank or Google matrix P without damping as

$$P_{ij} = \left\{ \begin{array}{ll} 1/n_i, & \text{if page } i \text{ links to page } j \\ 0, & \text{otherwise} \end{array} \right. \tag{1}$$

Also define the PageRank or Google matrix P with damping as λ as:

$$P_{ij} = \left\{ \begin{array}{ll} (1-\lambda)/n_i + \lambda/n, & \text{if page } i \text{ links to page } j \\ \lambda/n, & \text{otherwise} \end{array} \right. \tag{2}$$

where n_i is the number of outgoing links on page i. $\lambda \in [0, 1]$ is a constant sometimes called the *damping factor*.

The page rank of each webpage i is the ith entry of the row vector q which solves

$$q = qP.$$

The page rank depends on your damping factor.

1. **Example network.** Find the example internet with for webpages, `http://mschauer.github.io/page1.html`, `http://mschauer.github.io/page2.html`, `http://mschauer.github.io/page3.html`, `http://mschauer.github.io/page4.html`.

   In this exercise, there is no damping ($\lambda = 0$.)

   (a) Use a die (throw again if 5 or 6) to choose one of the 4 web-pages in the network. Record your choice.

   Proceed by clicking on a randomly chosen link on the current webpage to get a new one. Throw a die to choose a random link with uniform probability.

   Record the pages you have visited. Record 15 pages.

   Which page have you visited most?

   (b) Draw a picture with arrows indicating links between the four webpages. Find the adjacency matrix A^{ex}.

   (c) Write the PageRank matrix P^{ex} for the example network assuming $\underline{\lambda = 0}$.

   (d) Show that the rows of P^{ex} in the example sum to 1.

   (e) Describe the Markov process with transition matrix P^{ex} in words.

   (f) Compute: What is the probability that a surfer starting on page 1 ends on page 2 after two clicks?

   (g) Write a program which follows a random surfer in the web P^{ex}. Generate three surf history of a random surfer starting in 1.

   (h) Compute the probability of a random surfer to be in each Webpage after 2, after 10 steps if the starting distribution was $[\frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}; \frac{1}{4}]$.

   (i) Does $q = qP^{ex}$ have a solution?

2. **Markov chains.**

   (a) Show that P defined in (2) is a Markov matrix.

   (b) Explain how $\lambda > 0$ is related to “getting bored” in the quote above.

   (c) Argue that if $\lambda > 0$, there is a solution vector q. Give an interpretation of q. Why is q a good measure for Website importance? (Evidently it was, Google won the search engine market.)

3. **PageRank.**

   (a) Write a program to compute q for the example depending on λ.

   (b) Write a program which takes an adjacency (link) matrix A and λ computes the PageRank vector q.

   (c) Optional: Use this on a data set of your choice.