Math 415 - Lecture 31

Markov matrices and Google

Monday November 9th 2015

Textbook reading: Chapter 5.3

Suggested practice exercises: Chapter 5.3: 8, 9, 12, 14, 10.

Khan Academy video: Finding Eigenvectors and Eigenspaces example

Strang lecture: Lecture 21: Eigenvalues and eigenvectors Lecture 24: Markov Matrices and Fourier Series.

1 Review

1.1 Properties of eigenvectors and eigenvalues

- If $A\mathbf{x} = \lambda\mathbf{x}$ then $\mathbf{x}$ is an **eigenvector** of A with **eigenvalue** λ. All eigenvectors (plus $\mathbf{0}$) with eigenvalue λ form **eigenspace** of λ.
- λ is an eigenvalue of $A \iff \underbrace{\det(A - \lambda I)}_{\text{characteristic polynomial}} = 0$. Why? Because $A\mathbf{x} = \lambda\mathbf{x} \iff (A-\lambda I)\mathbf{x} = \mathbf{0}$. By the way: this means that the eigenspace of λ is just $\mathrm{Nul}(A-\lambda I)$.
- E.g. if $A = \begin{bmatrix} 3 & 2 & 3 \\ 0 & 6 & 10 \\ 0 & 0 & 2 \end{bmatrix}$ then $\det(A - \lambda I) = (3-\lambda)(6-\lambda)(2-\lambda)$.
- Eigenvectors $\mathbf{x}_1, \dots, \mathbf{x}_m$ of A corresponding to different eigenvalues are independent.
- product of eigenvalues = determinant
- sum of eigenvalues = "trace" (sum of diagonal entries)

Example 1. Find the eigenvalues of A as well as a basis for the corresponding eigenspaces, where

$$A = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 3 & 1 \\ -1 & 1 & 3 \end{bmatrix}.$$

Solution.

2 Markov matrices

Definition 2. An $n \times n$ matrices A is **Markov matrix** if has non negative entries, and the entries in each column add to 1.

Theorem 1. *Let A be a Markov matrix. Then*

(i) 1 is an eigenvalue of A and any other eigenvalue λ satisfies $|\lambda| \leq 1$.

(ii) If A has only positive entries, then any other eigenvalue satisfies $|\lambda| < 1$.

Example 3. Let A be

$$\begin{bmatrix} .9 & .2 \\ .1 & .8 \end{bmatrix}.$$

Is A a Markov matrix?

Theorem 2. *Let A be an $n \times n$-Markov matrix with only positive entries and let $\boldsymbol{v} \in \mathbb{R}^n$. Then*

$$\boldsymbol{v}_\infty := \lim_{k \to \infty} A^k \boldsymbol{v} \text{ exists,}$$

*and $A\boldsymbol{v}_\infty = \boldsymbol{v}_\infty$. In this case $\boldsymbol{v}_\infty$ is often called the **steady state**.*

Proof.

□

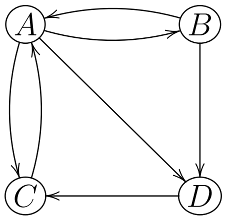
Example 4. Consider a fixed population of people with or without job. Suppose that each year, 50% of those unemployed find a job while 10% of those employed lose their job.What is the unemployment rate in the long term equilibrium?

Solution.

3 Page rank (or: the 25000000000 \$ eigenvector)

Google's success is based on an algorithm to rank webpages, the **Page rank**, named after Google founder Larry Page. (This might be a joke.) The basic idea is to determine how likely it is that a web user randomly gets to a given webpage. The webpages are ranked by these probabilities.

Suppose the internet consisted of the only four webpages A, B, C, D linked as in the following graph.



Imagine a surfer following these links at random. For the probability $PR_n(A)$ that she is at A (after n steps), we need to think about how she could have reached A. We add:

Encode the probabilties at step n in a state vector with four entries.

Definition 5. The **PageRank vector** is the long-term equilibrium. It is an eigenvector of the Markov matrix with eigenvalue 1.

Let's call the Markov matrix with the probabilities T:

- $T - 1I = \begin{bmatrix} -1 & \frac{1}{2} & 1 & 0 \\ \frac{1}{3} & -1 & 0 & 0 \\ \frac{1}{3} & 0 & -1 & 1 \\ \frac{1}{3} & \frac{1}{2} & 0 & -1 \end{bmatrix} \overset{\text{RREF}}{\rightsquigarrow} \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -\frac{2}{3} \\ 0 & 0 & 1 & -\frac{5}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}$

  $\implies$ eigenspace of $\lambda = 1$ is spanned by $\begin{bmatrix} 2 \\ \frac{2}{3} \\ \frac{5}{3} \\ 1 \end{bmatrix}$.

- Now we need to make the entries add up to 1.

$$\begin{bmatrix} PR(A) \\ PR(B) \\ PR(C) \\ PR(D) \end{bmatrix} = \frac{3}{16} \begin{bmatrix} 2 \\ \frac{2}{3} \\ \frac{5}{3} \\ 1 \end{bmatrix} = \begin{bmatrix} 0.375 \\ 0.125 \\ 0.313 \\ 0.188 \end{bmatrix}.$$

  This is the PageRank vector.

- The corresponding ranking of the webpages is A, C, D, B.

Remark. In practical situations the system might be too large for finding the eigenvalues by row operations.

- Google reports having met 60 trillion webpages. Google's search index is over $100,000,000$ gigabytes. Number of Google's servers is secret: about $2,500,000$ More than 1,000,000,000 websites (i.e. hostnames; about 75% not active)
- Thus we have a gigantic but very sparse matrix.

An alternative to row operations is the **power method** (see Theorem 2):
Here:

$$T = \begin{bmatrix} 0 & \frac{1}{2} & 1 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}.$$

Start with an arbitrary state vector, hit it with powers of T.

$$T\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 0.375 \\ 0.083 \\ 0.333 \\ 0.208 \end{bmatrix}, \quad T^2\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 0.375 \\ 0.125 \\ 0.333 \\ 0.167 \end{bmatrix}, \quad T^3\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 0.396 \\ 0.125 \\ 0.292 \\ 0.188 \end{bmatrix}$$

Remark.
- If all entries of T are positive (no zero entries!), then the power method is guaranteed to work.
- In the context of PageRank, we can make sure that this is the case by replacing T with

$$(1-p)\cdot\begin{bmatrix} 0 & \frac{1}{2} & 1 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} + p\cdot\begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}.$$

Just to make sure: still a Markov matrix, now with positive entries Google used to use $p = 0.15$.

4 Practice problems

Problem 6. *Can you see why* 1 *is an eigenvalue for every Markov matrix?*

Problem 7 (just for fun)**.** *The real web contains pages which have no outgoing links. In that case, our random surfer would get "stuck" (the transition matrix is not a Markov matrix). Do you have an idea how to deal with this issue?*