Week 2 Tuesday

- Self-grade due Friday
- Thoughts on "working smart not hard"
  - ↳ focus is a muscle (fast dopamine weakens it)
  - ↳ even 15min is enough to start a hard task

Plan

- Review eigendecomposition
- Page Rank
- Power Method

Eigen decomposition

$v_1, \dots, v_r \in \mathbb{R}^d$ orthonormal eigenvectors

$\langle v_i, v_j \rangle = 0$ if $i \neq j$

$\langle v_i, v_i \rangle = 1$

$\lambda_1 \geq \lambda_2 \dots \geq \lambda_r$ eigenvalues

$$A = \sum_{i=1}^{r} \lambda_i v_i v_i^T$$

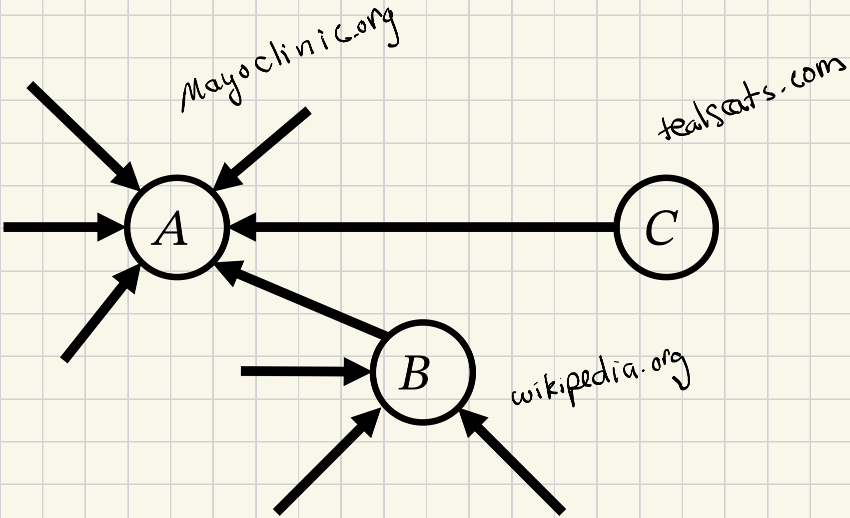
$$A v_j =$$

$$A A =$$

Larry webpages

Page Rank

How do we find <u>important</u> webpages?



Important if:

- lots of incoming links
- incoming links from <u>important</u> pages

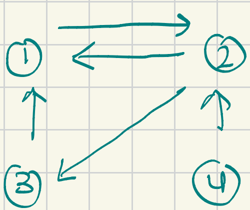
Page Rank Algorithm

iterative process of computing importance

$p^{(0)} \in \mathbb{R}^n$ where $p_i^{(0)} = 1/n$

$$p_i^{(t+1)} = \sum_{j=1}^{n} \mathbb{1}[j \text{ links to } i] \frac{p_j^{(t)}}{d_j}$$

d_j ← # outgoing edges from j



$$p^{(0)} = \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \qquad p^{(1)} =$$

Matrix View

$$p^{(t+1)} = A\, p^{(t)} \qquad A \in \mathbb{R}^{n \times n}$$

$$A_{i,j} = \begin{cases} 1/d_j & \text{if } j \text{ links to } i \\ 0 & \text{otherwise} \end{cases}$$

$$A = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Week 2 Thursday

- self-grade due Friday
- Problem 2 and 3 due Monday

Plan

- Page Rank!

iterative process to compute importance

$p^{(0)} \in \mathbb{R}^n$ where $p_i^{(0)} = 1/n$

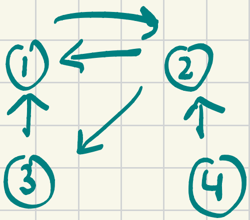
$$p_i^{(t+1)} = \sum_{j=1}^{n} \mathbb{1}[j \text{ links to } i] \frac{p_j^{(t)}}{d_j}$$

d_j ← # outgoing edges from j

Matrix View

$$p^{(t+1)} = A\, p^{(t)} \qquad A \in \mathbb{R}^{n \times n}$$

$$A_{i,j} = \begin{cases} 1/d_j & \text{if } j \text{ links to } i \\ 0 & \text{otherwise} \end{cases}$$



$$A = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Notice that columns sum to 1!

$$\sum_{i=1}^{n} p_i^{(t+1)} = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} [A]_{ij}\, p_j^{(t)} \right)$$

$$= \sum_{j=1}^{n} p_j^{(t)} \sum_{i=1}^{n} [A]_{ij}$$

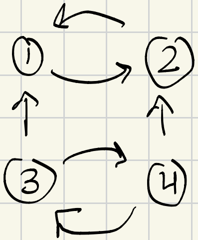
$$= \sum_{j=1}^{n} p_j^{(t)}$$

So, if $\sum_{i=1}^{n} p_i^{(0)} = 1$, then $\sum_{i=1}^{n} p_i^{(t)}$

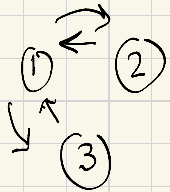
A probability distribution!

$p_i^{(t)} = \Pr($random clicker at page i in time step $t)$

Problem Matrices



Reducible



Periodic

Solution: Teleporting! $0 < \alpha < 1$

$$p^{(t+1)} = \alpha A p^{(t)} + (1-\alpha) \mathbf{1} \frac{1}{n}$$

$$= \left(\alpha A + (1-\alpha) \frac{1}{n} \mathbf{1}\mathbf{1}^T\right) p^{(t)}$$

(since $\mathbf{1}^T p^{(t)} = \sum_{i=1}^{n} p_i^{(t)} = 1$)

$$= M p^{(t)}$$

$$= M M p^{(t-1)}$$

$$= M M \ldots M \; p^{(0)}$$

$$= M^t p^{(0)}$$

Eigendecomposition

Non-symmetric, square matrix $M \in \mathbb{R}^{n \times n}$

$$M = \sum_{i=1}^{r} \lambda_i v_i w_i^T$$

$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r$ eigenvalues

$w_1, w_2, \dots, w_r \in \mathbb{R}^d$ left eigenvectors $\qquad w_i^T M =$

$v_1, v_2, \dots, v_r \in \mathbb{R}^d$ right eigenvectors $\qquad M v_i =$

---

bi-orthonormal

$$\langle v_i, w_j \rangle = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{else} \end{cases}$$

Proof:

$$w_i^T M v_j = w_i^T \lambda_j v_j = w_i^T \lambda_i v_j$$

$$\Leftrightarrow (w_i^T \lambda_j - w_i \lambda_i) v_j = 0$$

$$\Leftrightarrow w_i^T v_j (\lambda_j - \lambda_i) = 0$$

Power Method

$$M = \sum_{i=1}^{r} \lambda_i v_i w_i^T$$

$$MM =$$

$$M^t =$$

$$p^{(t)} = M^t p^{(0)} = \sum_{i=1}^{r} \lambda_i^t v_i w_i^T p^{(0)} = \lambda_1^t \sum_{i=1}^{r} \left(\frac{\lambda_i}{\lambda_1}\right)^t v_i w_i^T p^{(0)}$$

For this matrix,

$\lambda_1 = 1$, $|\lambda_i| < 1 \ \forall \ i \neq 1$

$$\lim_{t \to \infty} p^{(t)} = \lambda_1^t v_1 (w_1^T p^{(0)})$$

$$= v_1 (w_1^T p^{(0)})$$

Next: Supervised Learning

Input: Training data

e.g., weather, image classification, stock prices

$x^{(1)} \in \mathbb{R}^d$ $\quad$ $y^{(1)} \in \mathbb{R}$

$x^{(2)}$ $\quad$ $y^{(2)}$

$\vdots$ $\quad$ $\vdots$

$x^{(n)}$ $\quad$ $y^{(n)}$

Output: $f: \mathbb{R}^d \to \mathbb{R}$ s.t.

$$f(x^{(i)}) \approx y^{(i)}$$

Our Strategy

1. Function class
2. Loss
3. Optimizer