SOLUTIONS

Exercises on Block3:

Link Analysis – PageRank

Advertising

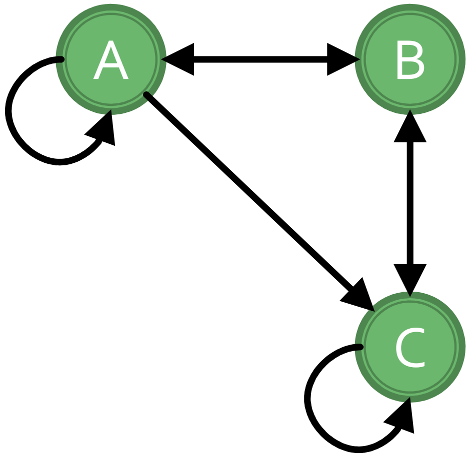
Recommender Systems

Advanced Search Techniques for Large Scale Data Analytics

Pavel Zezula and Jan Sedmidubsky
Masaryk University
http://disa.fi.muni.cz

PageRank (1) – Assignment

- For the following graph

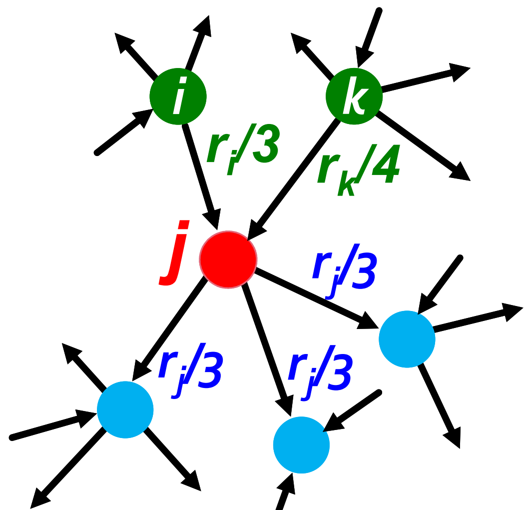


  - Compute the PageRank of each page, assuming no taxation

PageRank (1) – Recap

- Each link's vote is proportional to the **importance** of its source page
- If page j with importance r_j has n out-links, each link gets r_j / n votes
- Page j's own importance is the sum of the votes on its in-links

$$r_j = r_i/3 + r_k/4$$



PageRank (1) – Recap

- **Stochastic adjacency matrix $\boldsymbol{M}$**
  - Let page i has d_i out-links
  - If $i \rightarrow j$, then $M_{ji} = \frac{1}{d_i}$ else $M_{ji} = 0$
    - $\boldsymbol{M}$ is a **column stochastic matrix**
      - Columns sum to 1
- **Rank vector $\boldsymbol{r}$:** vector with an entry per page
  - r_i is the importance score of page i
  - $\sum_i r_i = 1$
- **The flow equations can be written**

$$r_j = \sum_{i \rightarrow j} \frac{r_i}{\mathrm{d_i}}$$

$$\boldsymbol{r} = \boldsymbol{M} \cdot \boldsymbol{r}$$

PageRank (1) – Recap

- **Given a web graph with *n* nodes, where the nodes are pages and edges are hyperlinks**
- **Power iteration:** a simple iterative scheme
  - Suppose there are *N* web pages
  - Initialize: $\mathbf{r}^{(0)} = [1/N,\ldots.,1/N]^T$
  - Iterate: $\mathbf{r}^{(t+1)} = \mathbf{M} \cdot \mathbf{r}^{(t)}$
  - Stop when $|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}|_1 < \varepsilon$

    $|\mathbf{x}|_1 = \sum_{1 \le i \le N} |x_i|$ is the $\mathbf{L_1}$ norm
    Can use any other vector norm, e.g., Euclidean

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

d_i …. out-degree of node i

PageRank (1) – Solution

- The transition matrix for the graph is:

$$M = \begin{pmatrix} 1/3 & 1/2 & 0 \\ 1/3 & 0 & 1/2 \\ 1/3 & 1/2 & 1/2 \end{pmatrix}$$

- By equation method ($M \cdot r = r$), we get the result:

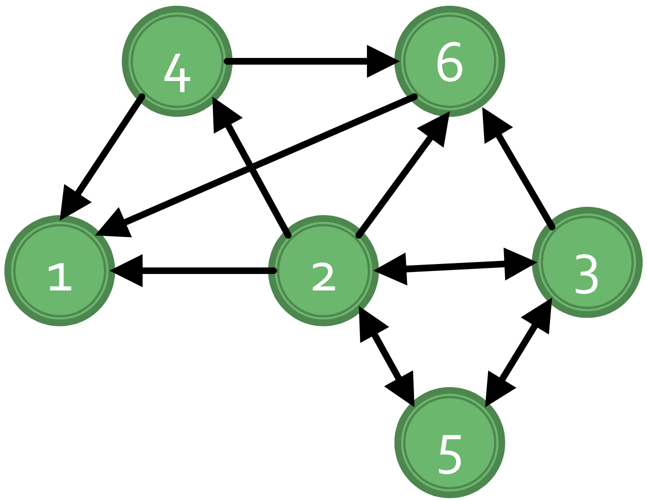
$$\begin{aligned} A &= \frac{1}{3}A + \frac{1}{2}B \\ B &= \frac{1}{3}A + \frac{1}{2}C \\ C &= \frac{1}{3}A + \frac{1}{2}B + \frac{1}{2}C \\ A &+ B + C = 1 \end{aligned} \quad \Rightarrow \quad \begin{aligned} A &= \frac{3}{13} \\ B &= \frac{4}{13} \\ C &= \frac{6}{13} \end{aligned} \qquad \boldsymbol{r} = \begin{pmatrix} \frac{3}{13} & \frac{4}{13} & \frac{6}{13} \end{pmatrix}^T$$

- By iteration method, we get the following list:

$$\begin{pmatrix} 0.3333 \\ 0.3333 \\ 0.3333 \end{pmatrix}, \begin{pmatrix} 0.2777 \\ 0.2777 \\ 0.4444 \end{pmatrix}, \begin{pmatrix} 0.2314 \\ 0.3148 \\ 0.4537 \end{pmatrix}, \begin{pmatrix} 0.2345 \\ 0.3040 \\ 0.4614 \end{pmatrix}, \begin{pmatrix} 0.2301 \\ 0.3088 \\ 0.4609 \end{pmatrix}, \ldots, \begin{pmatrix} 0.2307 \\ 0.3076 \\ 0.4615 \end{pmatrix}$$

PageRank (2) – Assignment

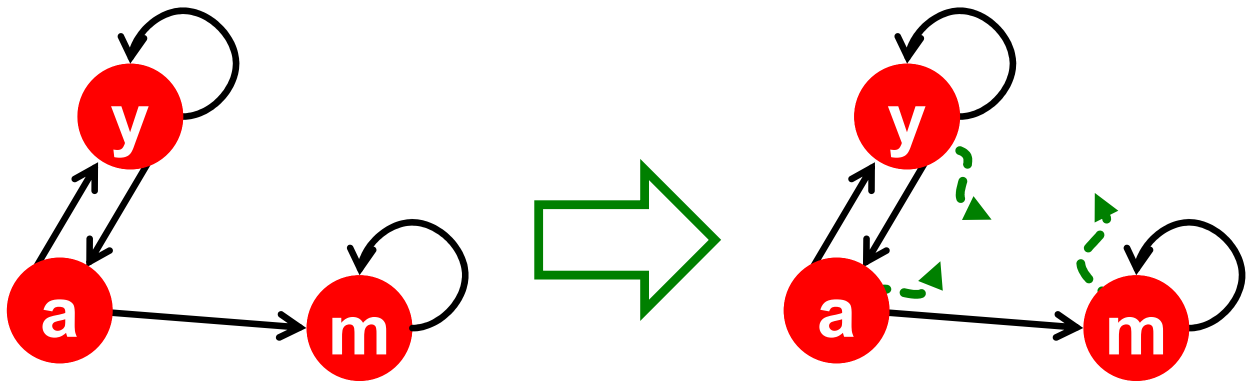
- For the following graph



1) Set up the PageRank equations, assuming β = 0.8
2) Order nodes by PageRank from highest to lowest

PageRank (2) – Recap

- **The Google solution for spider traps: At each time step, the random surfer has two options**
  - With prob. β, follow a link at random
  - With prob. **1-**β, jump to some random page
  - Common values for β are in the range 0.8 to 0.9
- **Surfer will teleport out of spider trap within a few time steps**



PageRank (2) – Recap

- **Google's solution that does it all:**
  At each step, random surfer has two options:
  - With probability β, follow a link at random
  - With probability $1-\beta$, jump to some random page
- **PageRank equation** [Brin-Page, 98]

$$r_j = \sum_{i \rightarrow j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{N}$$

d_i … out-degree of node i

This formulation assumes that M has no dead ends. We can either preprocess matrix M to remove all dead ends or explicitly follow random teleport links with probability 1.0 from dead-ends.

PageRank (2) – Recap

- **PageRank equation** [Brin-Page, '98]

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta) \frac{1}{N}$$

- **The Google Matrix *A*:**

$[1/N]_{NxN}$…N by N matrix where all entries are 1/N

$$A = \beta\, M + (1 - \beta) \left[\frac{1}{N}\right]_{N \times N}$$

- **We have a recursive problem: $\boldsymbol{r} = \boldsymbol{A} \cdot \boldsymbol{r}$**
  And the Power method still works!
- **What is β?**
  - In practice $\beta = 0.8, 0.9$ (make 5 steps on avg., jump)

PageRank (2) – Solution

- Equations:
  - $r_1 = 0.8 \cdot (1/6 \cdot r_1 + 1/2 \cdot r_4 + r_6 + 1/5 \cdot r_2) + 0.2/6$
  - $r_2 = 0.8 \cdot (1/6 \cdot r_1 + 1/3 \cdot r_3 + 1/2 \cdot r_5) + 0.2/6$
  - $r_3 = 0.8 \cdot (1/6 \cdot r_1 + 1/5 \cdot r_2 + 1/2 \cdot r_5) + 0.2/6$
  - $r_4 = 0.8 \cdot (1/6 \cdot r_1 + 1/5 \cdot r_2) + 0.2/6$
  - $r_5 = 0.8 \cdot (1/6 \cdot r_1 + 1/5 \cdot r_2 + 1/3 \cdot r_3) + 0.2/6$
  - $r_6 = 0.8 \cdot (1/6 \cdot r_1 + 1/5 \cdot r_2 + 1/3 \cdot r_3 + 1/2 \cdot r_4) + 0.2/6$

- Without the need of computing the actual importance from the above stated equations, we can derive order between the following pairs of nodes:

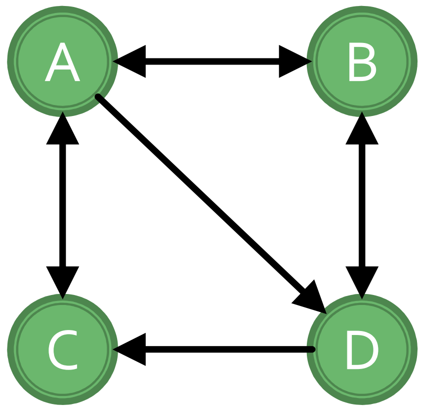
  $r_1 > r_6 \qquad r_4 < r_5 < r_6 \qquad r_2 > r_3 \qquad r_3 > r_5 \qquad r_6 > r_2$

  - This implies final order:

    $$\boldsymbol{r_1 > r_6 > r_2 > r_3 > r_5 > r_4}$$

PageRank (3) – Assignment

- For the following graph



  - Assuming β = 0.8, compute the topic-sensitive PageRank for the following teleport sets:
    1) {A}
    2) {A, C}

PageRank (3) – Recap

- Random walker has a small probability of teleporting at any step
- **Teleport can go to:**
  - **Standard PageRank: Any page with equal probability**
    - To avoid dead-end and spider-trap problems
  - **Topic Specific PageRank: A topic-specific set of "relevant" pages (teleport set)**
- **Idea: Bias the random walk**
  - When walker teleports, she pick a page from a set ***S***
  - ***S*** contains only pages that are relevant to the topic
    - E.g., Open Directory (DMOZ) pages for a given topic/query
  - For each teleport set ***S***, we get a different vector r_S

PageRank (3) – Recap

- **To make this work all we need is to update the teleportation part of the PageRank formulation:**

$$A_{ij} = \begin{cases} \beta M_{ij} + (1-\beta)/|S| & \textbf{if } i \in S \\ \beta M_{ij} + 0 & \textbf{otherwise} \end{cases}$$

  - ***A*** is stochastic!
- We weighted all pages in the teleport set ***S*** equally
  - **Could also assign different weights to pages!**
- **Compute as for regular PageRank:**
  - Multiply by ***M***, then add a vector
  - Maintains sparseness

PageRank (3) – Recap

- $\boldsymbol{r} = \boldsymbol{A} \cdot \boldsymbol{r}$, **where** $\boldsymbol{A_{ji}} = \boldsymbol{\beta}\, \boldsymbol{M_{ji}} + \frac{\boldsymbol{1-\beta}}{\boldsymbol{N}}$
- $r_j = \sum_{\mathrm{i}=1}^{N} A_{ji} \cdot r_i$
- $r_j = \sum_{i=1}^{N} \left[\beta\, M_{ji} + \frac{1-\beta}{N}\right] \cdot r_i$

  $= \sum_{\mathrm{i}=1}^{N} \beta\, M_{ji} \cdot r_i + \frac{1-\beta}{N} \sum_{\mathrm{i}=1}^{N} r_i$

  $= \sum_{\mathrm{i}=1}^{N} \beta\, M_{ji} \cdot r_i + \frac{1-\beta}{N}$ since $\sum r_i = 1$
- **So we get:** $\boldsymbol{r} = \boldsymbol{\beta}\, \boldsymbol{M} \cdot \boldsymbol{r} + \left[\frac{\boldsymbol{1-\beta}}{\boldsymbol{N}}\right]_{\boldsymbol{N}}$

Note: Here we assumed **M** has no dead-ends

$[x]_N$ … a vector of length N with all entries x

PageRank (3) – Recap

- We just rearranged the **PageRank equation**

$$r = \beta M \cdot r + \left[\frac{1-\beta}{N}\right]_N$$

  - where **$[(1-\beta)/N]_N$** is a vector with all N entries **$(1-\beta)/N$**
- ***M*** is a **sparse matrix!** (with no dead-ends)
  - 10 links per node, approx 10N entries
- So in each iteration, we need to:
  - Compute $r^{new} = \beta M \cdot r^{old}$
  - Add a constant value **$(1-\beta)/N$** to each entry in r^{new}
    - **Note if M contains dead-ends then $\sum_j r_j^{new} < 1$ and we also have to renormalize r^{new} so that it sums to 1**

PageRank (3) – Solution 1/4

- The transition matrix for the graph is:

$$M = \begin{pmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix} \qquad \beta \cdot M = \begin{pmatrix} 0 & 2/5 & 4/5 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{pmatrix}$$

1) Computing PageRank for teleport set {A} using **equations**:

$$(1-\beta) \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \begin{aligned} A &= \frac{2}{5}B + \frac{4}{5}C + \mathbf{\frac{1}{5}} \\ B &= \frac{4}{15}A + \frac{2}{5}D \\ C &= \frac{4}{15}A + \frac{2}{5}D \\ D &= \frac{4}{15}A + \frac{2}{5}B \\ A + B &+ C + D = 1 \end{aligned} \Rightarrow \boldsymbol{r} = \begin{pmatrix} \mathbf{\frac{3}{7}} & \mathbf{\frac{4}{21}} & \mathbf{\frac{4}{21}} & \mathbf{\frac{4}{21}} \end{pmatrix}^{\boldsymbol{T}}$$

PageRank (3) – Solution 2/4

- The transition matrix for the graph is:

$$M = \begin{pmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix} \qquad \beta \cdot M = \begin{pmatrix} 0 & 2/5 & 4/5 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{pmatrix}$$

1) Computing PageRank for teleport set {A} using **iterations**:

$$(1-\beta) \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \Rightarrow \quad r^{(1)} = \beta \cdot M \cdot r^{(0)} + \begin{pmatrix} 1/5 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- We can initialize vector *r* in different ways; however, the sum of values must equal to 1, e.g., $r^{(0)} = (1 \quad 0 \quad 0 \quad 0)^T$

$$\Rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.2 \\ 0.2666 \\ 0.2666 \\ 0.2666 \end{pmatrix}, \begin{pmatrix} 0.52 \\ 0.16 \\ 0.16 \\ 0.16 \end{pmatrix}, \dots, \begin{pmatrix} 0.4285 \\ 0.1904 \\ 0.1904 \\ 0.1904 \end{pmatrix}$$

PageRank (3) – Solution 3/4

- The transition matrix for the graph is:

$$M = \begin{pmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix} \qquad \beta \cdot M = \begin{pmatrix} 0 & 2/5 & 4/5 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{pmatrix}$$

2) Computing PageRank for teleport set {A,C} using **equations**:

$$(1-\beta) \cdot \begin{pmatrix} 1/2 \\ 0 \\ 1/2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/10 \\ 0 \\ 1/10 \\ 0 \end{pmatrix} \Rightarrow \begin{aligned} A &= \frac{2}{5}B + \frac{4}{5}C + \mathbf{\frac{1}{10}} \\ B &= \frac{4}{15}A + \frac{2}{5}D \\ C &= \frac{4}{15}A + \frac{2}{5}D + \mathbf{\frac{1}{10}} \\ D &= \frac{4}{15}A + \frac{2}{5}B \\ A + B &+ C + D = 1 \end{aligned} \Rightarrow \boldsymbol{r} = \begin{pmatrix} \mathbf{\frac{27}{70}} & \mathbf{\frac{6}{35}} & \mathbf{\frac{19}{70}} & \mathbf{\frac{6}{35}} \end{pmatrix}^{\boldsymbol{T}}$$

PageRank (3) – Solution 4/4

- The transition matrix for the graph is:

$$M = \begin{pmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix} \qquad \beta \cdot M = \begin{pmatrix} 0 & 2/5 & 4/5 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{pmatrix}$$

2) Computing PageRank for teleport set {A,C} using **iterations**:

$$(1-\beta) \cdot \begin{pmatrix} 1/2 \\ 0 \\ 1/2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/10 \\ 0 \\ 1/10 \\ 0 \end{pmatrix} \Rightarrow \quad r^{(1)} = \beta \cdot M \cdot r^{(0)} + \begin{pmatrix} 1/10 \\ 0 \\ 1/10 \\ 0 \end{pmatrix}$$

- We can initialize vector *r* in different ways; however, the sum of values must equal to 1, e.g., $r^{(0)} = (1 \quad 0 \quad 0 \quad 0)^T$

$$\Rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.1 \\ 0.2666 \\ 0.3666 \\ 0.2666 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 0.1333 \\ 0.2333 \\ 0.1333 \end{pmatrix}, \ldots, \begin{pmatrix} 0.3857 \\ 0.1714 \\ 0.2714 \\ 0.1714 \end{pmatrix}$$

Advertising (1) – Assignment

- Suppose the BALANCE algorithm with bids of 0 or 1 only, to a situation where advertiser
  - A bids on query words x and y
  - B bids on query words x and z
  - Both have a budget of $2. Decide whether the following sequences of queries are certainly handled optimally by the algorithm:
    1) yzyy
    2) xyyz
    3) xyzx

Advertising (1) – Recap

- **BALANCE** Algorithm by Mehta, Saberi, Vazirani, and Vazirani
  - **For each query, pick the advertiser with the largest unspent budget**
    - Break ties arbitrarily (**but in a deterministic way**)

Advertising (1) – Recap

- **Two advertisers A and B**
  - **A** bids on query ***x***, **B** bids on ***x*** and ***y***
  - Both have budgets of **$4**
- **Query stream:** ***x x x x y y y y***
- **BALANCE choice:** **A B A B B B _ _**
  - Optimal: **A A A A B B B B**
- **In general:** For **BALANCE** on **2** advertisers
  Competitive ratio = ¾

Advertising (1) – Solution

1) **Yes** (for input sequence: yzyy)
   - Balance choice: yzy (**$3**) Optimal: yzy (**$3**)
2) **No** (for input sequence: xyyz)
   - If the x is assigned to A, then the second y cannot be satisfied, while the optimum assigns all four queries
   - Balance choice: xyz (**$3**) Optimal: xyyz (**$4**)
3) **Yes** (for input sequence: xyzx)
   - Whichever advertiser is assigned the first x, the other will be assigned the second x, thus using all four queries
   - Balance choice: xyzx (**$4**) Optimal: xyzx (**$4**)

Recomm. Systems (1) – Assignment

- Bookstore has enough ratings to use a more advanced recommendation system
  - Suppose the mean rating of books is 3.4 stars
  - Alice has rated 350 books and her average rating is 0.4 stars higher than average users' ratings
  - Animals Farm, is a book title in the bookstore with 250,000 ratings whose average rating is 0.7 higher than global average
  - What is a baseline estimate of Alice's rating for Animals Farms?

Recomm. Systems (1) – Solution

- Baseline estimate of Alice's rating for Animals Farms:

  $r = 3.4 + 0.7 + 0.4 = \mathbf{4.5}$

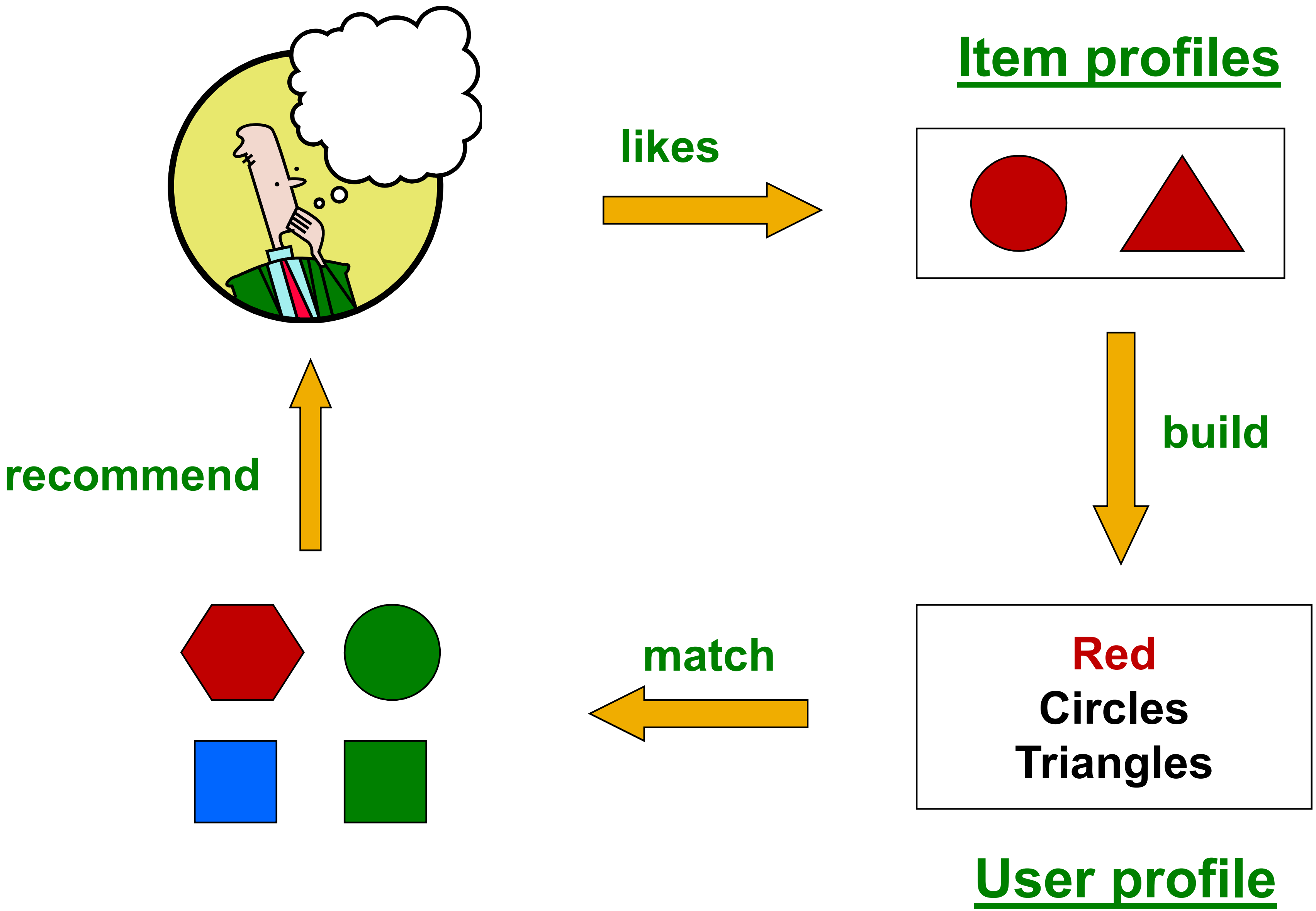
Recomm. Systems (2) – Assignment

- Computers A, B and C have the following features:

Feature	A	B	C
Processor speed	3.06	2.68	2.92
Disk size	500	320	640
Main-memory size	6	4	6

  - Assuming features as a vector for each computer, e.g., A's vector is [3.06, 500, 6], we can compute the cosine distance between any two vectors
  - Scaling dimensions can prefer some components
  - Assume 1 as the scale factor for processor speed, α for the disk size, and β for the main memory size and compute:
    - The cosines of angles between pairs of vectors (in terms of α and β)

Recomm. Systems (2) – Recap



Recomm. Systems (2) – Recap

Recomm. Systems (2) – Solution

- The cosines of angles between pairs of vectors (in terms of α and β)

$$\cos(A,B) = \frac{8.2008 + 160000\alpha^2 + 24\beta^2}{\sqrt{9.3636 + 250000\alpha^2 + 36\beta^2} \cdot \sqrt{7.1824 + 102400\alpha^2 + 16\beta^2}}$$

$$\cos(B,C) = \frac{7.8256 + 204800\alpha^2 + 24\beta^2}{\sqrt{7.1824 + 102400\alpha^2 + 16\beta^2} \cdot \sqrt{8.5264 + 409600\alpha^2 + 36\beta^2}}$$

$$\cos(A,C) = \frac{8.9352 + 320000\alpha^2 + 36\beta^2}{\sqrt{9.3636 + 250000\alpha^2 + 36\beta^2} \cdot \sqrt{8.5264 + 409600\alpha^2 + 36\beta^2}}$$

Recomm. Systems (3) – Assignment

- A user has rated the three computers as follows:
  - A: 4 stars, B: 2 stars, C: 5 stars
- Tasks:
  1) Normalize the ratings for this user
  2) Compute a user profile for the user, with the following features

Feature	A	B	C
Processor speed	3.06	2.68	2.92
Disk size	500	320	640
Main-memory size	6	4	6

Recomm. Systems (3) – Solution

- 1) Normalized ratings:
  - $avg(4+2+5)/3 = 11/3$
  - A: $4 - 11/3 = 1/3$
  - B: $2 - 11/3 = -5/3$
  - C: $5 - 11/3 = 4/3$
- 2) Computed user profile:
  - Processor speed: $3.06 \cdot 1/3 - 2.68 \cdot 5/3 + 2.92 \cdot 4/3 = 0.4467$
  - Disk size: $500 \cdot 1/3 - 320 \cdot 5/3 + 640 \cdot 4/3 = 486.6667$
  - Main-memory size: $6 \cdot 1/3 - 4 \cdot 5/3 + 6 \cdot 4/3 = 3.3333$