CAIM, examen parcial

27 de novembre de 2017. Temps: 1 hora 50 minuts

Exercise 1 (2 points) Suppose a search engine that provides a list of potentially relevant web pages for a given query. For each of these pages, the search engine is able to compute its PageRank weight and its cosine similarity with respect to the query efficiently. Argue how the search engine should sort the pages when presenting them to the user.

Answer: If web pages are sorted only by their cosine similarity then we will be neglecting the PageRank weight, a measure of the importance of a web page. If we sort the documents only by PageRank weights then we will be neglecting the relevance by content that is provided by the cosine similarity. Therefore, we have to combine both kind of weights. We know that c, the cosine similarity, is a number between 0 and 1 (because the document tf-idf vectors have positive components). We also know that the p, the PageRank weight is a number between 0 and 1 because it is actually the probability of visiting the web page following a random walk. As both kind of weights have the same range of variation, a simple way of combining both weights is taking their arithmetic mean, i.e.

$$\frac{c+p}{2}$$

. The problem of that mean is that it is rather easy to produce a rather high value from unbalanced c and p (e.g., very high c but low p or low c and very high p). The way c and p are aggregated should be such that it penalizes pages where p is too small or c is too small (to prevent pages poorly related according to content or with little authority to appear early in the ordering). Therefore, a more convenient way of aggregating c and p is a harmonic mean (F-measure), i.e. $2/(1/c + 1/p)$. The harmonic mean has the virtue of giving a value that is closer to $min(c, p)$ than the arithmetic mean, reducing the risks of the arithmetic mean. As precision (vs recall) is critical for a search engine, we could use an α-F measure to tune the importance of c, i.e.

$$\alpha\text{-}F = \frac{2}{\frac{\alpha}{c} + \frac{1-\alpha}{p}}.$$

We can infer this from the topic Evaluation, where we have addressed a similar problem, namely, the aggregation of precision and recall to produce a single value. Following the idea of topic sensitive PageRank, we could also aggregate c and p as their product cp and introduce some parameter to tune the weight of precision.

Exercise 2 (1.5 points) Suppose a posting list that consists of a sequence of n

docid-frequency pairs, i.e.

$$x_1, y_1, ..., x_i, y_i, ...x_n, y_n$$

where x_i is the i-th docid and y_i is the frequency of occurrence of the term in document x_i. For instance, the sequence of integers

$$1, 7, 5, 3$$

indicates that the term appears seven times in document 1 and 3 times in document 5.

We have compressed a posting list following the format above and obtained the following string of bits

$$00100001110101101111100 1010$$

Decode the bit string to obtain the original posting list assuming that

1. Frequencies have been coded using unary self-delimiting codes as a sequence of 1's ending by a 0.
2. Docids have been coded using gap compression and Elias γ codes (the unary self-delimiting code within the Elias γ code is a sequence of 0's ending by a 1).

Hint: the first element of the bit string is an Elias γ code representing the number 4.

Answer: Segmenting the sequence, one gets a list of pairs

$$(00100, 0), (011, 10), (1, 0), (1, 10), (1, 11110), (010, 10)$$

that encodes the list

$$(4, 1), (3, 2), (1, 1), (1, 2), (1, 5), (2, 2).$$

Undoing gap compression, we finally obtain

$$(4, 1), (7, 2), (8, 1), (9, 2), (10, 5), (12, 2)$$

Exercise 3 (3 points)

1. Explain the difference between a word type and a word token. Include an example.

2. As you know, Zipf's law for word frequencies can be defined as

$$f_i = c_1 i^{-\alpha},$$

where f_i is the frequency of i-th most frequent word type, α is a parameter and c_1 is a constant that depends on α. Another law, Zipf's meaning-frequency law, defines the relationship between f, the frequency of a word, and μ, its number of meanings. The law states that

$$f = c_2 \mu^2,$$

where c_2 is a constant. Suppose a text that contains V types and T tokens. Estimate m, the mean number of meanings of the types of that text based only on the two laws above. Obtain a formula for m that depends only on V, the ratio c_1/c_2 and α.

3. On the previous formula, assume $\alpha = 2$ to obtain an approximate expression for m that depends only on V, T and c_2. Hint:

$$\sum_{i=1}^{V} f_i = T \tag{1}$$

$$\sum_{i=1}^{V} i^{-1} \approx \gamma + \log V \tag{2}$$

$$\sum_{i=1}^{\infty} i^{-2} = \frac{\pi^2}{6}, \tag{3}$$

where γ is Euler's constant.

Answer:

1. A word type is an element of the set of words that have appeared in a text sample. A word token is an occurrence of a word type. The tokenizer produces a sequence of tokens from a text. If our tokenizer lowercases and our sample is "The man loves the woman." we get 5 tokens, i.e. *the*, *man*, *loves*, *the* and *woman*, and four types, i.e. $\{the, man, loves, woman\}$. Altenatively, it could be said that all types occur once except *the* which appears twice.

2. We aim to estimate the mean number of meanings of word types, i.e.

$$m = \frac{1}{V} \sum_{i=1}^{V} \mu_i,$$

where μ_i is the number of meanings of the i-th type ($i = 1, 2, ...V$). Applying Zipf's meaning-frequency law, i.e.

$$\mu = \left(\frac{f}{c_2}\right)^{1/2},$$

m becomes

$$m = \frac{1}{V c_2^{1/2}} \sum_{i=1}^{V} f^{1/2}.$$

Applying Zipf's law for word frequencies, we obtain

$$m = \frac{1}{V} \left(\frac{c_1}{c_2} \right)^{1/2} \sum_{i=1}^{V} i^{-\alpha/2}.$$

3. First, notice that $\alpha = 2$ gives

$$m = \frac{1}{V} \left(\frac{c_1}{c_2} \right)^{1/2} \sum_{i=1}^{V} i^{-1}.$$

Recalling that Eq. 2, we get

$$m \approx \frac{\gamma + \log V}{V} \left(\frac{c_1}{c_2} \right)^{1/2}$$

Second, Eq. 1 and the definition of Zipf's law on ranks above with $\alpha = 2$, give

$$c_1 \sum_{i=1}^{V} i^{-2} = T.$$

Recalling that Eq. 3, we obtain

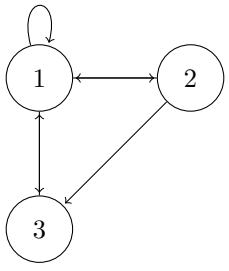
$$c_1 \approx \frac{6T}{\pi^2}$$

for sufficiently large V. Finally, combining the results in the 1st and 2nd step, we get

$$m \approx \frac{\gamma + \log V}{\pi V} \left(\frac{6T}{c_2} \right)^{1/2}.$$

We could neglect γ because it a positive number smaller than 1.

Exercise 4 (3.5 points) Consider a network of web pages defined by the following graph



1. Give the four PageRank equations for that graph assuming a damping factor λ.

2. Solve the system of equations obtaining the PageRank weights of each node as a function of λ.

3. Give the PageRank weights for $\lambda = 1/3$.

4. Suppose that $\vec{p}$ is a vector of PageRank weights and G is a Google matrix. Explain why the definition of the PageRank problem in matrix notation as
$$\vec{p} = G^T \vec{p}$$
is equivalent to
$$p_i = \frac{1-\lambda}{n} + \lambda \sum_{j \to i} \frac{p_j}{k_j^{out}}, \tag{4}$$
where n is the number of vertices and k_j^{out} is the out-degree of the j-th vertex.

Answer:

1. Applying Eq. 4, we get
$$p_1 = \frac{1-\lambda}{n} + \frac{\lambda}{3}p_1 + \frac{\lambda}{2}p_2 + \lambda p_3 \tag{5}$$
$$p_2 = \frac{1-\lambda}{n} + \frac{\lambda}{3}p_1 \tag{6}$$
$$p_3 = \frac{1-\lambda}{n} + \frac{\lambda}{3}p_1 + \frac{\lambda}{2}p_2 \tag{7}$$
Additionally,
$$p_1 + p_2 + p_3 = 1. \tag{8}$$

2. Eqs. 5 and 7 give
$$p_1 - p_3 = \lambda p_3$$
and then
$$p_1 = (\lambda + 1)p_3 \tag{9}$$
Eqs. 6 and 7 give
$$p_3 - p_2 = \frac{\lambda}{2}p_2$$
and then
$$p_3 = \left(\frac{\lambda}{2} + 1\right) p_2. \tag{10}$$
Combining Eqs. 9 and 10, we get
$$p_1 = (\lambda + 1)\left(\frac{\lambda}{2} + 1\right) p_2$$

Applying the definitions of p_1 and p_3 as a function of p_2 obtained above to Eq. 8, we get

$$(\lambda+1)\left(\frac{\lambda}{2}+1\right)p_2 + p_2 + \left(\frac{\lambda}{2}+1\right)p_2 = 1$$

and finally

$$\begin{aligned} p_2 &= \frac{1}{(\lambda+2)\left(\frac{\lambda}{2}+1\right)+1} \\ &= \frac{2}{\lambda^2+4\lambda+6}. \end{aligned} \tag{11}$$

Combining the last result and Eq. 10, we get

$$p_3 = \frac{\lambda+2}{\lambda^2+4\lambda+6}. \tag{12}$$

Applying the definitions of p_2 and p_3 in Eqs. 11 and 12 to $p_1 = 1-p_2-p_3$, we obtain

$$p_1 = \frac{\lambda^2+3\lambda+2}{\lambda^2+4\lambda+6}. \tag{13}$$

3. $p_2 = 18/67 \approx 0.2686$, $p_3 = 21/67 \approx 0.3134$ and $p_1 = 1-p_2-p_3 = 28/67 \approx 0.4179$.

4. $A = \{a_{ij}\}$ is the adjacency matrix. $M = \{m_{ij}\}$ is the transition matrix (obtained normalizing A by row). We use B_{i*} to denote the i-th row vector of matrix B. Knowing that

$$G = \lambda M + \frac{1-\lambda}{n}J,$$

Eq. 4 gives

$$\begin{aligned} p_i &= G_{i*}^T\vec{p} \\ &= \left(\lambda M_{i*}^T + \frac{1-\lambda}{n}J_{i*}\right)\vec{p} \\ &= \sum_{j=1}^{n}\left(\lambda m_{ij}^T + \frac{1-\lambda}{n}\right)p_j \\ &= \sum_{j=1}^{n}\lambda m_{ij}^T p_j + \frac{1-\lambda}{n}\sum_{j=1}^{n}p_j \\ &= \lambda\sum_{j=1}^{n}m_{ij}^T p_j + \frac{1-\lambda}{n} \\ &= \lambda\sum_{j=1}^{n}\frac{a_{ji}}{k_j^{out}}p_j + \frac{1-\lambda}{n} \\ &= \lambda\sum_{j\to i}\frac{p_j}{k_j^{out}} + \frac{1-\lambda}{n} \end{aligned}$$

as we wanted to prove.