

Exam Advances in Data Mining

Wojtek Kowalczyk

w.j.kowalczyk@liacs.leidenuniv.nl

28/10/2019

It is a closed book exam: you are not allowed to use any notes, books, calculators, smartphones, etc. The number of points attached to each question reflects the (subjective) level of question's difficulty. In total you may get 100. The final grade for the exam is the total number of points you receive divided by 10.

In principle, the exam consists of a number of questions with a "single choice answer". It means that for each question you should select exactly one answer. For every correct choice you get some points; for an incorrect choice or no choice you get 0 points.

Mark your choices by crossing the selected option. In case you want to "undo" your choice put a circle around the cross. For example, on the left side the option **b** is selected; on the right side nothing is select – the selection of **b** is "undone":

a) bla bla

~~b~~) ble ble

c) bli ble

a) bla bla

☒ b) ble ble

c) bli ble

If you think that your marking is no longer readable, put your final choice on the left margin (e.g., by writing "a" if you want to select "a").

Additionally, some questions require that you generate your own answer: either a formula or a numeric value. In such cases you get extra space for optional justification of your calculations.

Before starting answering the questions, fill in the following entries:

Name:

Student number:

Study type (ICT, Astronomy, ...):

| | |
|-------------------|--|
| 5+5 points | 1. TF.IDF |
| | <p>Let us suppose that we have a collection of 2^{20} (about 1 million) documents. The word “the” is very popular: in every document it is the most frequent word. On the other hand, the word “idf” is very rare: there is only one document D that contains a single occurrence of this word and no other document contains this word.</p> <p>A) What is the value of TF.IDF of the word “the” in document D?</p> <p>a) 0 $TF(\text{“the”}, D)=1(\text{“the” is most frequent in every document}), \text{ and}$ $IDF(\text{“the”})=\log_2(2^{20}/2^0)=\log_2(1)=0$</p> <p>b) $1/2^{20}=2^{-20}$ c) $1/\log_2(2^{20})=1/20$ d) 1 e) It is impossible to answer this question</p> <p>B) What is the value of TF.IDF of the word “idf” in document D?</p> <p>a) 0 b) $1/2^{20}=2^{-20}$ c) $1/\log_2(2^{20})=1/20$ d) 1 e) It is impossible to answer this question. To calculate $DF(\text{“idf”}, D)$ we need to know the frequency of the most frequent word in D – and we don’t have this information. So although we know $IDF(\text{“idf”}) = \log_2(2^{20}/1)$, we cannot calculate TF.IDF.</p> |

| | |
|-----------------|--|
| 5 points | 2. Hash functions |
| | <p>A hash function that maps documents into 16-bit long integers has been applied to a collection of $N=2^{16}$ documents, returning a list L of 2^{16} hashes. What is the expected number of unique (distinct) values that occur in L?</p> <p>a) About $0.75*N$ b) About $0.63*N$ c) About $0.50*N$ d) About $0.37*N$ e) About $0.25*N$</p> <p>What is the chance that a randomly selected 16-bit long integer k is NOT in L? It is the same as the chance that after selecting at random 2^{16} 16-bit long integers we will never select k. The chance that we don’t select k in a single experiment is $1-1/2^{16}$, the chance that none of the 2^{16} experiments will select k is $(1-1/2^{16})^{2^{16}} = 1/e=1/2.71$ which is close to 0.37. Therefore, the chance that k is IN L is 0.63.</p> |

| 5 points | 3. Two Bloom filters |
|----------|---|
| Question | <p>Consider two Bloom filters, <i>A</i> and <i>B</i>. The filter <i>A</i> uses one buffer with $2N$ bits and two hash functions. The filter <i>B</i> uses a cascade of two buffers with N bits each and two hash functions – one hash function per buffer. Which filter has better (smaller) false-positive rate? Select the right answer:</p> <ul style="list-style-type: none"> a) <i>A</i> is better than <i>B</i>. b) <i>B</i> is better than <i>A</i>. c) Both filters are equally good. d) The answer depends on the value of N. <p>This is a tricky question with several possible answers!</p> <p>What is the false-positive rate of filter <i>A</i>? Using the formulas from slide 18 (AIDM_Streams_1) with $k=2$ and N replaced by $2N$ we have: $fp_A = (1 - ((1 - 1/2N)^{2N})^{2n/2N})^2 \approx (1 - \exp(-2n/2N))^2 = (1 - \exp(-n/N))^2.$</p> <p>And what is the false-positive rate of filter <i>B</i>? A single filter with N bits and a single hash function has the fp rate (check slide 17 Streams_1): $fp = 1 - [(1 - 1/N)^N]^{n/N} \approx 1 - \exp(-n/N)$ so fp for the cascade of two such filters has the fp_B rate $(1 - \exp(-n/N))^2$.</p> <p>Therefore it seems that both filters have the same fp rate. However, in both cases we used an approximation $(1 - 1/m)^m \approx \exp(-1)$ with different values of m! Had we used the actual values of m ($m=2N$ in case <i>A</i> and $m=N$ in case <i>B</i>) we would notice that fp_A is bigger than fp_B. However, the difference would be very small. E.g., for $n=10^6$ (million objects), $N=2 \cdot n$ (number of bits) both filters have fp rate about 15.48%, but $fp_A - fp_B = 2.9778e-08$. Concluding: in theory filter <i>B</i> is better, but in practice the difference is negligible. Therefore answers b) and c) are treated as correct ones.</p> |

| 5 points | 4. Bloom filter with h hash functions |
|----------|--|
| Question | <p>Consider a Bloom filter that uses B bits and h hash functions to “store” n objects. What is the probability that a randomly selected object will pass the filter? Select the right formula:</p> <ul style="list-style-type: none"> a) $e^{-hn/B}$ b) $e^{hn/B}$ c) $(1 - e^{-hn/B})^h$ (see slide 19, Streams_1) d) $(1 - e^{hn/B})^h$ e) None of the above |

| | |
|--------------------|---|
| 5 points | 5. A cascade of 2 Bloom filters |
| Question | Consider a cascade of 2 Bloom filters that are supposed to “store” N objects. The first filter uses h_1 hash functions and n_1 bits; the second filter uses h_2 hash functions and n_2 bits. Write down a formula that estimates the false positive rate: the chance that a randomly chosen object passes both filters of this cascade. |
| Write your answer! | $fp(h_1, n_1, h_2, n_2, N) = \left(1 - \exp\left(-\frac{h_1 N}{n_1}\right)\right)^{h_1} \left(1 - \exp\left(-\frac{h_2 N}{n_2}\right)\right)^{h_2}$ |

| | |
|-------------------------|---|
| 5 + 5 + 5 points | 6. Matrix Factorization and RBM (Restricted Boltzmann Machine) |
| Question | <p>Let us assume that a Matrix Factorization model with $K=50$ factors has been trained for $N=10^7$ users and $M=10^5$ items.</p> <p>A) How many gigabytes of memory are needed for storing model parameters, assuming that every parameter is stored with double precision (i.e., 8 bytes)? Select the closest estimate:</p> <ul style="list-style-type: none"> a) About 1GB b) About 2GB c) About 4GB d) About 8GB e) About 12GB <p>For every user we need to store K parameters and for every item we need to store K parameters so in total we need a little bit more than 4GB: $8 \cdot (N \cdot K + M \cdot K) = 8 \cdot 50 \cdot (10^7 + 10^5) \approx 400 \cdot 10^7 = 4000 \cdot 10^6 = 4000 \text{MB} = 4 \text{GB}$</p> <p>Here we ignored 10^5 as it is only 1% of 10^7.</p> <p>B) How many multiplications are needed to find, for every user, 5 items that (s)he would rate highest? Such recommendations are usually generated in advance, so when a client visits a website, 5 items with the highest predicted rating can be instantly displayed. Select the closest estimate:</p> <ul style="list-style-type: none"> a) $M \cdot N$ b) $K \cdot M \cdot N$ c) $5 \cdot M \cdot N$ d) $5 \cdot K \cdot M \cdot N$ e) None of the above <p>For every user-item combination we have to perform K multiplications, so in total we need $K \cdot M \cdot N$ multiplications.</p> |

| | |
|--|--|
| | <p>C) Suppose that instead of the Matrix Factorization model an RBM is used to model user's preferences. Assuming the same architecture as used during the Netflix Challenge, estimate the number of trainable parameters of such an RBM (select the closest estimate):</p> <ul style="list-style-type: none"> a) $M*N$ b) $K*M*N$ c) $5*M*N$ d) $5*K*M*N$ e) $5*K*N$ f) $5*K*M$ <p>We assume here that K denotes the number of hidden/invisible nodes, N the number of users and M the number of items.</p> <p>The visible layer consists of groups of 5 inputs, one group per item. The invisible layer consist of K nodes. Thus in total we have $5*M*K$ trainable parameters (ignoring biases that would cost another $5*M+K$ parameters).</p> |
|--|--|

| | |
|-------------------|---|
| 5 5 points | 7. Matrix Factorization: time and memory requirements |
| Question | <p>Suppose that training (with help of the SGD algorithm) a matrix factorization model for the original Netflix Challenge data takes h hours of cpu-time and requires m GB of RAM. Let us assume that the same algorithm is applied to an updated dataset which is much bigger than the original one: it contains 3 times more ratings that were given by 4 times more users to 4 times more movies.</p> <p>How many cpu hours H, and GB of RAM M, would be needed to rerun this algorithm on the updated data set? We assume that RAM is used only for storing the model (and not the training data), that the number of factors is not changed, and that the hardware has the same speed.</p> |
| Answer options | <p>CPU-time:</p> <ul style="list-style-type: none"> a) $H=3h$ (the SGD algorithm processes rating after rating; #users and #movies are irrelevant) b) $H=4h$ c) $H=12h$ d) $H=16h$ e) $H=48h$ <p>RAM:</p> <ul style="list-style-type: none"> a) $M=3m$ b) $M=4m$ (the size of the model is #factors*(#users+#movies)) c) $M=12m$ d) $M=16m$ e) $M=48m$ |

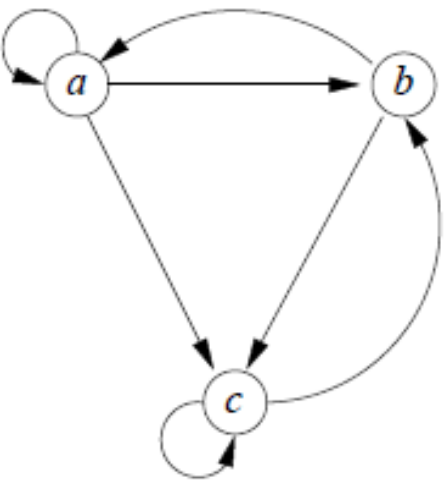
| | |
|-------------------|--|
| 5+5 points | 8. Cosine Similarity and LSH |
| Question | <p>Let us suppose that we have a collection of 2^{20} (about 1 million) documents and that we want to apply LSH with the cosine similarity measure to find pairs of similar documents. First, we remove very frequent and very infrequent words, limiting the set words that occur in the reduced documents to 1024. In this way each document can be represented by a vector of 1024 values of TF.IDFs.</p> <p>A) Assuming that each TF.IDF value is represented by a 32-bit long float, how much memory do we need to store vector representation of all documents? Select the closest estimate:</p> <ul style="list-style-type: none"> a) 1GB b) 2GB c) 4GB d) 8GB e) 16GB <p>The TF.IDF matrix has size $1024 * 1M = 1G$ entries; each entry takes 4 bytes, so the total is 4GB.</p> <p>B) Now suppose that a signature matrix is constructed with help of 512 random projections that are applied to vector representations of all documents. How much memory do we need to represent this matrix? Select the closest estimate:</p> <ul style="list-style-type: none"> a) 64MB b) 128MB c) 0.5GB d) 1GB e) 2GB <p>Hint: what are the elements of the signature matrix? Each element of the signature matrix is 1 or -1, so one bit of information is sufficient to represent it! And 1 byte = 8 bits so $512 * 1M \text{ bits} = 64 * 1M \text{ bytes}$.</p> |

| | |
|---------------------|---|
| 5+5+5 points | 9. Estimation of the Second Moment (Alon-Matias-Szegedy algorithm) |
| Question | <p>a) What is the second moment of the following sequence:</p> <p style="text-align: center;">C, A, D, C, A, A, C, D, B, A</p> <p>There are 4 A's, 1 B, 3 C's and 2 D's so the second moment is: $4^2 + 1^2 + 3^2 + 2^2 = 30$.</p> <p>b) Estimate the second moment of this sequence by applying the AMS-algorithm at the following "random positions": 2, 4, 5 and 8 (we count positions from 1 and not from 0, so the position 2 corresponds to the first occurrence of A). What are the values of these 4 estimates?</p> <p>2: 4 occurrences of A $\Rightarrow 10 * (2 * 4 - 1) = 70$ 4: 2 occurrences of C $\Rightarrow 10 * (2 * 2 - 1) = 30$ 5: 3 occurrences of A $\Rightarrow 10 * (2 * 3 - 1) = 50$ 8: 1 occurrences of D $\Rightarrow 10 * (2 * 1 - 1) = 10$</p> <p>c) What is the final estimate of the second moment? 160/4=40</p> |

| | |
|--------------------------------------|--|
| 5 points | 10. Reservoir Sampling |
| Question | <p>There is an unspecified number (>1000) of documents stored on a big read-only optical disk that is connected to a computer with a small hard disk. Your task is to collect a random sample of 1000 of these documents, and store them on the hard disk, in such a way that each document has the same probability of entering your sample. Unfortunately, you don't know the number of documents stored on the optical disk and you are allowed to read the disk sequentially, document after document, only once. Moreover, the hard disk of the computer on which your sample has to be stored is very small: it can keep exactly 1000 documents. However, during the sampling process you are allowed to delete or to overwrite documents on your hard disk.</p> <p>To solve this problem you decided to use the Reservoir Sampling method. Let us assume that after processing 3000 documents from the optical drive your hard disk is full and you have to decide what to do with document 3001 (let's call it D). How will you proceed?</p> |
| Provide description of the algorithm | <ul style="list-style-type: none"> a) Do nothing – the hard disk is full. b) Replace the oldest document on the hard disk with D. c) Replace a randomly chosen document on the hard disk with D. d) Decide with probability $1/3001$ to save D, overwriting a randomly selected document from the hard disk. e) None of the above – you should do something different! <p>We should decide with probability $1000/3001$ to save D (overwriting a randomly selected document from the hard disk), but this option is not listed!</p> |

| | |
|-----------------|--|
| 5 points | 11. LSH: the key formula |
| Question | Which of the formulas specified below expresses the probability of “being detected”, $p(s, b, r)$, as a function of: the actual similarity s , the number of bands, b , and the number of rows per band, r ? |
| Answer options | <ul style="list-style-type: none"> a) $p(s, b, r) = 1 - (1 - s^r)^b$ b) $p(s, b, r) = 1 - (1 - s^b)^r$ c) $p(s, b, r) = (1 - s^b)^r$ d) $p(s, b, r) = (1 - s^r)^b$ e) None of the above |

| | |
|----------------|---|
| 5 points | 12. LSH: the impact of changing the number of bands |
| Question | Let us suppose that you want to apply LSH technique to 1 million documents to find pairs of documents with Jaccard similarity of at least 0.8. You've already generated a signature matrix with signatures of length 100 and have split it into 4 bands with 25 rows each. Unfortunately, you were not able to find in this way any pair. |
| Answer options | <p>Which action makes most sense:</p> <p>a) Split your signature matrix into more bands with less rows, e.g., 10 bands with 10 rows each. The chance that a pair with similarity 0.8 is detected by a band with 25 rows is $0.8^{25} \approx 0.4\%$, while for 10 rows it is $0.8^{10} \approx 11\%$ so we would highly increase the chance of being detected. Also the number of bands is increased by factor 2.5! So this strategy makes sense.</p> <p>b) Split your signature into less bands with more rows, e.g., 3 bands with 33 rows each. Increasing the number of rows in a band decreases the chance of finding candidate pairs in such a band, so it wouldn't work at all!</p> <p>c) Generate a signature matrix with longer signatures, e.g., 500 and then split it into 20 bands with 25 rows each. Increasing the number of bands by factor 5 shouldn't make a big difference – each band would still have 25 rows, which makes the probability of being detected still very small (although 5 times bigger than in the original setup-now we have 20 bands instead of 4).</p> <p>d) Use brute-force search to find at least some pairs of similar documents. This is completely hopeless: it would take months to complete.</p> |

| | |
|----------|--|
| 5 points | 13. Page Rank: Example |
| Question | <p>Consider the following graph:</p>  <p>Let us assume that we start surfing the graph at a randomly selected node (i.e., all nodes have the same probability of being selected as the starting point), following the links at random. What are the probabilities p_A, p_B, p_C that after making two steps we will end-up in node A, B, C, respectively? Write down the calculated values.</p> |
| | <p>The transition matrix $M = \begin{bmatrix} 1/3 & 1/2 & 0 \\ 1/3 & 0 & 1/2 \\ 1/3 & 1/2 & 1/2 \end{bmatrix}$</p> <p>$v = [1/3, 1/3, 1/3]^T$</p> <p>so $M \cdot v = [5/18, 5/18, 4/9]^T$</p> <p>and</p> <p>$M \cdot (M \cdot v) = [25/108, 17/54, 49/108]^T$</p> <p>Therefore:</p> <p>$p_A = 25/108$ $p_B = 17/54$ $p_C = 49/108$</p> |