Exam Advances in Data Mining

Wojtek Kowalczyk
wojtek@liacs.nl
31/10/2017

It is a closed book exam: you are not allowed to use any notes, books, etc. For each problem you will get some points; additionally you will get 10 points for free. The final grade is the total number of points you receive divided by 10.

1. Recommender Systems (15 = 5+3+2+5)

a) Describe in detail the Matrix Factorization approach to recommendation systems. What error measure is optimized? How? What is regularization and what is its role?

b) Let us assume that a Matrix Factorization model with K factors has been trained for N users and M items. How much memory is needed for storing model parameters, assuming that a single parameter is stored with double precision (i.e., 8 bytes)? Give a formula. In particular, how many gigabytes of memory are needed to store such a model for $N=10^7$, $M=10^5$, $K=50$?

c) Staying with the "$N=10^7$, $M=10^5$, $K=50$" scenario from the previous question: how many multiplications are needed to find, for every user, the item that (s)he would rate highest? Such recommendations are usually generated in advance, so when a client visits a website, an item with the highest predicted rating can be instantly displayed.

d) Let us consider a simple linear regression recommendation system, which maintains for every user u his/her average rating avg_u and for every item i its average rating avg_i. The linear regression model is given by three parameters α, β, γ such that a predicted rating of user u given to item i is:

$$predicted\_rating\_u\_i = \alpha * avg\_u + \beta * avg\_i + \gamma .$$

While the model parameters α, β, γ usually don't change over time, average user and item ratings do change with every new rating. Therefore, it is common to maintain, for every user u, two values: the number of items rated so far, n_u, and the sum of all ratings given so far, s_u, and divide them when required. The same applies to items. In this way the complete model can be stored in RAM and be continuously updated with every new rating. Let us assume that the number of users is always about 100 times bigger than the number of items. How much RAM (approximately) is needed to maintain this simple recommendation system for 1 billion (10^9) users?

2. TF.IDF and LSH (30 = 5+10+3+3+5+4)

a) Given a collection D of documents one can measure the "importance" of a word w that occurs in a document d with help of *"Term Frequency times Inverse Document Frequency"*, TF.IDF. Provide the definition of TF.IDF.

b) Suppose that a huge collection of documents D is stored on a Hadoop system. One term that is involved in the definition of TF.IDF is *Document Frequency*: for any word w, DF of w is the number of documents that contain w, n_w. Provide a pseudo-code of Map and Reduce functions which produce a list of all words that occur in documents from D and their *Document Frequencies*. The Map function should take as input documents from D and the Reduce function should produce a list of pairs: words that occur in documents from D, followed by the number of documents that contain these words.

c) Suppose there is a repository of one million documents, which, in total use 10.000 unique words. Therefore, assuming a fixed ordering of words, each document can be represented by a vector of 10.000 numbers: values of TF.IDFs of all words that might occur in this document. Assuming that TF.IDF values are stored as 32-bit floats, how many bytes are needed to store all these vectors?

d) Suppose now that similarity of documents in our repository is measured by cosine similarity of their vector representations. Provide the definition of cosine similarity. How many pairs of vectors should be investigated in order to find a pair of the most similar documents? How many multiplications would be needed for that?

e) In order to find in our repository pairs of most similar documents (where similarity is measured with cosine similarity), one can use LSH. How many bytes would be needed to store the signatures of all the documents, assuming that 100 random projections are used?

f) Assuming that in our repository there are pairs of documents that are very similar to each other (cosine similarity close to 1), which value of the number of bands would you use in order to find these pairs: small, medium or big? We assume here that we apply LSH technique with signatures of length 100. Justify your answer.

3. Bloom filter (15 = 3+12)

a) Let us suppose that n distinct credit card numbers are "stored" in a Bloom filter A of length N bits (with N>>n) that uses one hash function. What is the probability that a randomly generated card number will pass the filter? Provide a formula.

b) Now suppose that 10% of card numbers (i.e., n/10) have been compromised (e.g., due to skimming) and should be blocked. We know that practically it is impossible to remove these cards from the Bloom filter A. However, we can add another filter B, also of length N and a single hash function (different that the one used by filter A) to store all blocked card numbers. Next, we add the following decision logic: a card number will pass this "tandem" of filters A and B if it passes A and does not pass B. Clearly, no blocked card number will pass this combination. Unfortunately, some valid card numbers (those that are stored in filter A and are not blocked) might now be blocked by filter B: the hash function used by filter B may send a valid card number to the same bin as a blocked card. What is the percentage of valid card numbers that will be blocked by filter B? Write down the formula.

4. Estimating Moments (15 = 3+5+7)

a) Provide a definition of the k-th moment of a sequence of elements $x_1, \ldots, x_n$, for k=0, 1, 2, ...

b)

c) Describe the Alon-Matias-Szegedy algorithm for estimating second moments for streams of a fixed length N. Illustrate working of this algorithm on the sequence:

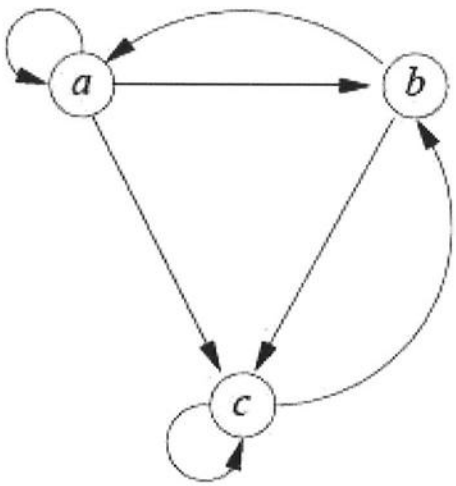
3, 1, 4, 3, 1, 1, 3, 4, 2, 1, 2

considering the following "random positions": 2, 4, 5 and 9 (we count positions from 1 and not from 0, so the position 2 corresponds to the element 1). What are the corresponding $X_i.values$? What is the final estimate of the second moment? What is the true value of the second moment? How does it compare to the "true" second moment of this sequence?

d) Describe the generalization of the Alon-Matias-Szegedy algorithm for estimating second moments over streams of unbounded length.

5. PageRank Algorithm (15 = 9+6)

a) Describe the concept of page rank, the transition matrix, the iterative algorithm for calculating PageRank, the teleporting, and the modified update rule. Specify a transition matrix for the graph below. What is the probability of visiting node C in the second step, assuming that in the first step all nodes have the same probability of being visited, and that β=0.75?



b) An efficient algorithm for computing PageRank that was discussed during the course, stores the transition matrix on a hard disk in a special format. Describe this format. How much RAM and disk space are needed to execute the PageRank algorithm on a graph with 10^9 nodes and 10^{10} links (average out-degree=10). Justify your answer.