Information Retrieval and Recommeder Systems (IRRS)
Master in Data Science (MDS)

Session 6: Architecture, Hadoop, NoSQL

Exercise List, Fall 2022

Exercises for credit. Solving three of these exercises (not solved by the instructors in class) suffice for full credit for this assignment.

Exercise 1

Explain how to solve the following tasks in the MapReduce programming model. Note that they may require more than one phase of map+reduce. Here S is supposed to be a *large* multiset of integers in some *huge* range $1 \dots N$.

1. Given S and a function `bool f(int n)`, say on how many elements in S does `f` evaluate to true.

2. Given S, compute its histogram: For each $i \in 1 \dots N$, give the number of occurrences of i in S.

3. Binning 1: Given S and a number k, partition $1 \dots N$ into k equal-length intervals and say how many elements in S fall into each interval.

4. Compute the cumulative distribution of S: For each $i \in S \cap \{1 \dots N\}$, say how many elements in S are less than or equal to i.

5. Binning 2: Given S and a number k, partition $1 \dots N$ into k equal-weight intervals, where "equal-weight" means that each interval contains roughly the same number of elements in S.

Note that the last two items are more difficult (both in your thinking time and in computation time) than the others.

Exercise 2

Give a solution in the mapreduce model for the following problem: We have a (very big) file formed by lines. Each line contains information on a blog post. More precisely, a line contains one or more names; the first name is the author of the post, and the rest (if any) are the names of the people that commented on the post. Note that a person may comment several times on the same post, and the author him/herself may comment too. For example, in the three lines

```
Jane Joe Zuzana Rosamaria Rosamaria Peter Jane Rosamaria
Jane Xabier Peter Xabier Martha Xabier Martha
Jane
```

the first one represents a post by Jane, which was commented by herself, Joe, Zuzana, Rosamaria (three times), and Peter. The second post is also authored by Jane, and was commented by Xabier (three times), Peter, and Martha (twice). The third post by Jane did not receive any comment at all.

We want to obtain, for each user X that has ever posted something, the name of the user Y that has commented on more *different* posts by X - that is, we want one pair (X,Y) for each such X. If several Y's are tied, return any of them. For example, if Jane only authored the three posts above, then we should return the pair `(Jane,Peter)`, because Peter has commented on the largest number of posts by Jane.

Give pseudocode for map and reduce functions and, if appropriate, combine and partition functions. If it is not obvious, explain what you choose to be the input to a map instance. The efficiency of your solution will be valued. Note that a solution with a single map-reduce phase suffices, but better give one that uses several phases than nothing!

Exercise 3

We say that two consecutive words in a document form a *2-phrase*. A document with L words can thus contain at most $L-1$ different 2-phrases.

Write programs in mapreduce (= map, reduce, and perhaps combine and partition functions) to do the following:

1. Given a set of documents and a parameter k, return the set of 2-phrases that appear at least k times in the set.
2. Given a set of documents and a word w, returns a list of all the pairs (d, v) for each document d and for each v such that (w, v) is a 2-phrase in document d.

Exercise 4

We are given a large set of files written in different languages. We want to choose a subset of files that contains exactly one representative of each language and also compute the frequency (number of files) of each language.

Give a mapreduce program (map, reduce, and perhaps combine and partition functions) for this task. To be precise, assume that each map instance will receive a list of strings, where each string is the name of one of the files to be processed. We want to return, for each language present in the collection, a tuple such as ("English",("The_Martian.epub",2489)), indicating that there are 2489 files in English and that "The_Martian.epub" is one of them. Assume that there is a function `language(string t)` that returns the name of the language in which text `t` is written, such as "English".

Exercise 5

Part 1: Suppose we have two huge vectors $x, y \in R^n$, and assume that only about $k \ll n$ components fit in one machine. Give a MapReduce scheme to compute the inner product of x and y.

Part 2: Given a matrix A and a column vector x, explain how to compute the product Ax in the MapReduce model. Assume that the whole matrix A does not fit in one machine. Let us say that only about k rows (or columns) of A fit in one machine. What if not even a full row, but only a part of a row, fits in one machine?

Exercise 6

The Jacobi method is one iterative way of solving systems of linear equations, similar to the power method described in the course for computing PageRank. Given with $A \in R^{n\times n}$ and $b \in R^n$, we can solve $Ax = b$ approximately as follows:

- We assume that for each i we have $A_{i,i} \neq 0$ (rows can always be reordered so that this is the case).
- (This is easy) Decompose A as $A = D + R$, where D is a diagonal matrix with no 0's in the diagonal and R has only 0's in the diagonal. Compute D^{-1} (say how).
- (Observe: we have now that if x is a solution, we have $(R + D)x = b$, or $b - Rx = Dx$, or $x = D^{-1}(b - Rx)$. This suggests a solution based on the iteration of $x \leftarrow D^{-1}(b - Rx)$.)
- Starting from some initial guess $x^{(0)}$, iterate $x^{(k+1)} = D^{-1}(b - Rx^{(k)})$
- Stop when e.g. $\|x^{(k+1)} - x^{(k)}\|_2 \leq \epsilon$.

Implement the above in the MapReduce model. Appeal to the previous exercise.

Exercise 7

The PageRank algorithm performs the iteration $P \leftarrow G^T P$ starting from some initial guess for P, where G is the Google matrix. Describe how to do this in MapReduce, assuming you have G.

Now assume that we do not have G and we do not want to compute it explicitly, because just storing G requires $O(n^2)$ memory, with n the number of nodes in G. What we have is, for each i, the outdegree of i and a list of the nodes j such that j has an edge to i. If the average outdegree in our graph is k, this uses memory $O(kn)$ instead of $O(n^2)$. Say how to do the PageRank computation in MapReduce from this information, using memory $O(kn)$. It may be easier to do this assuming first damping factor $\lambda = 1$; then, think how to do it with arbitrary λ without increasing the complexity much.

Exercise 8

Suppose we have two (large) tables $T1$ and $T2$ with a common attribute A. Assume that rows of the tables are sufficiently small so that a map function can take as parameters a row of $T1$ and/or a row of $T2$. Think how to implement the basic operations of relational algebra in Mapreduce:

- Say how to compute a *selection* of $T1$: for a predicate P on $T1$'s attributes, produce a table $T1'$ containing the rows of $T1$ that satisfy P.
- Say how to compute a *projection* of $T1$ onto a subset S of the attributes.
- Say how to compute the *join* of $T1$ and $T2$ on K, the table whose rows are $(a, v1, v2)$ for all the pairs such that $(a, v1)$ is a row of $T1$ $(a, v2)$ is a row of $T2$, and a denotes the value of the common attribute A.
- How many intermediate (key,value) pairs are produced by the operations above?
- Remember that the SQL sentence `select <attributes> from <tables> where <conditions>` can be rewritten using selections, projections, and joins. Discuss: Is it then true that MapReduce can simulate select sentences? Is it a good idea to use MapReduce as a basis for a relational DBMS?