Lecture 4: MapReduce: Counting at Scale
Modeling Social Data, Spring 2017
Columbia University

Anant Sharma

February 16, 2017

1 Combining and Reshaping Data

1.1 GroupBy

- Can implement the split/apply/combine framework with group_by
- There are a lot of red herrings while using group_by. We should always ungroup for performance and correctness issues. Examples can be checked in the Jupyter notebook.
  - Ungroup for performance
  - Ungroup for correctness
  - Unintentionally overwriting columns
  - Tricky variable scoping(lazy variable naming)

1.2 Joins

- There are various types of joins which can be used in different scenarios. Fortunately, we don't need to implement these joins, as there are functions in R to use them. We only need to learn which join to use in which situation.
- Inner Join
  - Returns the records which match in both the tables in some specific column
  - We can do an inner join on more than one data frames by nesting the joins
  - Symmetric - The order of the arguments doesn't matter
- Left Join

  - Returns all the records of the left table, even if they might not have matches with the right table
  - Not symmetric

- Right Join
  - Complement of left join; returns all the records of the right table, even if they don't have matches with the left table
  - Not symmetric

- Full Join
  - Combines both left and right join in a way; doesn't drop any records and returns missing values for unmatched values
  - Symmetric

- Anti Join
  - Opposite of inner join; shows the rows in first table which don't have a match with the second table
  - Shows what's being dropped; can be checked to see if any rows were dropped from the left table(number of rows in anti-join should be zero)
  - Not symmetric

1.3 Tidyverse

- Data pre-processing is a very important part of data science and one that which takes up a disproportionate amount of time. Tidyr is one R library which helps us in doing that with two functions - spread and gather. Examples can be seen in the Jupyter notebook on Github.
  - Spread - Can convert a "long" table to a "wide" table, by specifying a column as a key and the value column, which returns one column per key, and value filled in the cells
  - Gather - Does the opposite operation of spread; converts a wide table to a long table

2 MapReduce and Hadoop

2.1 MapReduce

- It helps us to solve the split/apply/combine problem in a distributed manner.

- A distributed solution is required because we can scale vertically only up to a point, after which we'll need multiple machines working on the same problem.
- For example, it takes roughly 4 hours to read 1 TB of data from a commonly used hard disk. Using MapReduce, we can sort 1 TB of data in 62 seconds and a 1 Peta Byte of data in 16.25 hours.
- There are three major parts of Map Reduce.
  - Mapper - Transforms the input records to (key, value) pairs
  - Shuffler - Collects all the intermediates records by key, and assigns them to the reducers by the function hash(key) % num_reducers, where num_reducers is the number of reducers that we have. The only priority of shuffler is to make sure that the records for the correct machines.
  - Reducer - Transforms all the records with given key to final output
- Programmer specifies the map and reduce functions, but he doesn't have to worry about fault tolerance, synchronization and other issues which are taken care of under the hood.

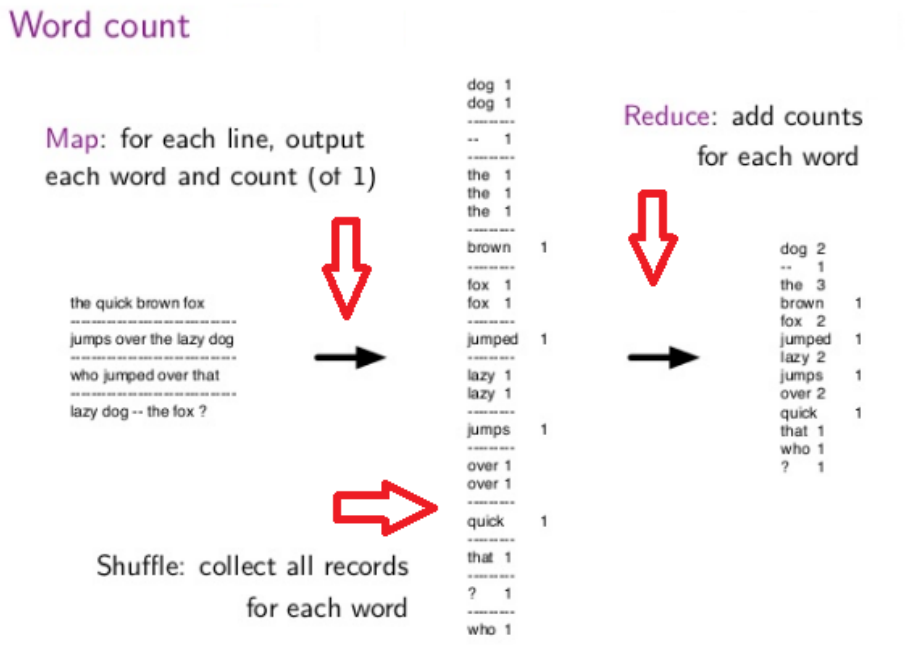

Figure 1: Taken and edited from the slides

2.2 Curse of Last Reducer

- Suppose we're counting the ticket sales of various movies over a weekend, and we make the name of the movie as a key
- There might be one movie which is a blockbuster and has done a great amount of business, whereas the other movies might have done a mild business
- We'll have to wait for a great time until the tickets for the blockbuster movie have been counted, even when all the other movies have already given counts
- In this case, the idea of map reduce breaks down. The reason is the skewness in the data. To rectify this problem, we should have a clever shuffling step in between. We could even give hints to the system by telling it the name of columns which we expect might take more time. Otherwise, we might have to wait for hours.

2.3 Hadoop

- Hadoop is an open-source implementation of the Map Reduce framework. It was named so after a toy of the developer's kid. Hadoop comes along with various utilities/sub-projects like Hadoop Common, Chukwa, HBase, HDFS, Hive, MapReduce, Pig, Zookeeper etc.
- Hadoop streaming is one such utility which allows the user to create and run map/reduce jobs with any executable or script as the mapper and the reducer.
- Higher level languages like Pig and Hive provide robust implementations for many MapReduce operations like filter, sort, join, group_by, along with allowing custom made map and reduce operations.