Bielefeld University
Faculty of Technology

Big Data Analytics

Summer 2023

Number 02, Submission Deadline: May 1 2023, 11:59 PM

IMPORTANT: Exercises 1 and 2 are taken from the last exercise sheet.

1. **LSH:** Consider the following signature matrix:

C_1	C_2	C_3	C_4	C_5	C_6	C_7
1	2	1	1	2	5	4
2	3	4	2	3	2	2
3	1	2	3	1	3	2
4	1	3	1	2	4	4
5	2	5	1	1	5	1
6	1	6	4	1	1	4

   Perform locality-sensitive hashing as introduced in the lecture with three bands of two rows each. Assume we have enough buckets available so that the hash function for each band can be the identity function (i.e. columns hash to the same bucket if and only if they are identical in the band).

   a) Find all candidate pairs and give a textual explanation how you identified the candidates. (4 P)

2. **LS Families:** What is the effect of probability of stating with the family of minhash functions and applying: (8 P)

   1) A 2-way AND construction followed by a 3-way OR construction.
   2) A 3-way OR construction followed by a 2-way AND construction.
   3) A 2-way AND construction followed by a 2-way OR construction, followed by a 2-way AND construction.
   4) A 2-way OR construction followed by a 2-way AND construction, followed by a 2-way OR construction, followed by a 2-way AND construction.

   Show how the probability can be calculated and how the s-curves could look like for the 4 cases.

3. **Set up a functioning Snakemake environment:** (2 P)
   The official Snakemake tutorial [1]contains a walkthrough to setup the library and an environment in various operating systems.

   Make sure to familiarize yourself with this new anaconda/miniconda environment, especially their conda package and environment manager. Setup Snakemake on your machine and also install the libraries `numpy`, `pandas`, and `seaborn` in it. Make sure you install `jupyter` with `conda` as well so you don't have to mix environments if you want to use a notebook for parts of this worksheet.

   Verify that snakemake works in your environment by inspecting the output of snakemake -v on the command line. The version shown should be 5.4.x or above.

4. **Snakemake workflows and targets:**
   In this task we build a small workflow that illustrates how to work with workflow rules and placeholders.[2] The workflow is extended on the next exercise sheet.

   (a) Prepare a working directory as described in the previous task. Create an empty file called `Snakefile` and run `snakemake -n ()`. Please paste the output in your solution. (1 P)

   (b) Place the dataset files `coursesTaken-{A,B}.csv` (available in `coursesTaken.zip` in the Moodle alongside this exercise) in your working directory. Make sure the file is extracted from the ZIP file. (1 P)

   (c) Create a python file `count.py` that: (4 P)

   - accepts an input filename and output filename as command line parameters. (hint: `import sys` and `sys.argv` will be helpful here).
   - iterates over the lines, splits them into columns using the delimiter tab `\t` and counts the number of unique elements in the first column (containing student IDs).
   - Writes the total number of unique students into the output file.

   Verify that your program works as intended by invoking `python3 count.py coursesTaken-B.csv testoutput.txt`. Does the output match the correct value (1000000)?

   (d) Create a Snakemake rule called `count` in your `Snakefile` that (4 P)

[1] `https://snakemake.readthedocs.io/en/stable/tutorial/setup.html`

[2] For another example that has more external dependencies you can also follow the official Snakemake tutorial.

calls this `count.py` script to generate an output called `coursesTaken-A.count` on the input `coursesTaken-A.csv` and `coursesTaken-B.count`. [3]

(e) Dry run the newly created rule to check if it will be applied for each input file using (4 P)
`snakemake -n --cores=2 coursesTaken-A.count coursesTaken-B.count`.
If you are satisfied with the output run `snakemake` without the dry-run flag, paste the content of your `.count` files below:

```
$ cat coursesTaken-A.count

$ cat coursesTaken-B.count
```

What did the parameter `--cores=2` do?

Important:
Please submit your group solution via Moodle. Note that Snakemake will require you to create files such as Snakefile, please ZIP your solution directory (without including the data files). On the next exercise sheet, you will use Snakemake for MapReduce, therefore, try to get familiar with Snakemake now.

[3] Hint: See the tutorial `https://snakemake.readthedocs.io/en/stable/tutorial/basics.html#step-2-generalizing-the-read-mapping-rule` on how these rules are structured and how placeholders can map inputs and outputs.