Artifact: Scalable Distributed Data Anonymization

Sabrina De Capitani di Vimercati*, Dario Facchinetti†, Sara Foresti*,
Gianluca Oldani†, Stefano Paraboschi†, Matthew Rossi†, Pierangela Samarati*
* Università degli Studi di Milano, Italy – Email: *firstname.lastname*@unimi.it
† Università degli Studi di Bergamo, Italy – Email: *firstname.lastname*@unibg.it

I. Introduction

We describe the artifact, publicly available at [1], that implements the proposal in [2], and the reproduction of the experimental results. It is an extended and distributed version of the Mondrian anonymization algorithm. Our solution anonymizes large datasets by partitioning data among workers in a distributed setting. It provides parallel execution on a dynamically chosen number of workers, limiting their interaction and data exchange.

II. Hardware and software requirements

Hardware requirements. The deployment of the artifact requires a machine having:

- a CPU with at least one logical core for each worker;
- at least 2 GB of RAM for each worker.

Software requirements. The deployment of the artifact requires a machine with Linux operating system (the experimental results have been obtained using Ubuntu 20.04 LTS) and the following packages installed:

- *make*, version 4.3;
- *git*, version 2.27.0;
- *zip*, version 3.0, and *gzip*, version 1.10-2;
- *python3*, version 3.8.6;
- *python3-venv*, version 3.8.6;
- *gnuplot*, version 5.2 patchlevel 8.

When these packages are available, the environment set up should be finalized through the following steps:

1) install and set up *docker* and *docker-compose* (for more details on this step, see Section "Prerequisites" at [1]);
2) run `sudo usermod -aG docker <USER>`;
3) reboot the system;
4) check that the following commands run without root privileges:
```
docker run hello-world
docker-compose -version
```

III. Deployment of the artifact

The steps for deploying the artifact are the following:

1) clone the repository through command
```
git clone --depth 1 --branch \
percom2021_artifact \
https://github.com/mosaicrown/mondrian.git
```
2) run `make` to verify that all the software requirements illustrated in Section II, which are needed for the distributed (Spark-based) version of the algorithm, are satisfied by the environment;
3) run `make start` to pull and build a copy of the Docker images necessary to the artifact.

The artifact uses the following Docker containers:

- *Hadoop Namenode* at http://localhost:9870
(the web page available at this url permits to check the status of the *Hadoop Datanode* and to browse the distributed file system);
- *Hadoop Datanode* at http://localhost:9864;
- *Spark History Server* at http://localhost:18080;
- *Spark Cluster Manager* (and thus *Spark worker*s) at http://localhost:8080.

IV. Use of the artifact

The artifact implements the centralized and our distributed version of the Mondrian algorithm. The artifact is complemented with a web UI that can be deployed running command `make ui`. The web UI is available at http://localhost:5000 and can be used to run customized experiments. A complete user guide to the web UI is available at [1]. In the following, we describe the use of the artifact to reproduce the experimental results presented in [2].

IPUMS USA dataset. The experiments in [2] used a sample from the IPUMS USA dataset [3]. The dataset is available at https://ipums.org/, together with a detailed guide for its download. To extract the sample to anonymize from the same dataset used in [2], go to IPUMS website https://usa.ipums.org/usa/ and click on "Get Data". Then, select the attributes of interest (harmonized variables `State FIP Code`, `Age`, `Education Number`, `Occupation`, and `Income` in our experiments) and add them to the cart. For your convenience, you can use the direct links at https://github.com/mosaicrown/mondrian#usa-2018-dataset (each variable name is a link that redirects to the page at ipums.org that permits to add the variable to the cart). Select the sample of interest (among USA samples, *2018 ACS* in our experiments) and create your data extract. To customize the sample size, set parameter *Persons* (in our experiments *Persons* is set to 510, to obtain a dataset with at least 500,000 tuples). Among the formats available for downloading the dataset, select the csv format and save the downloaded gzip archive in the root folder of the project, with name *usa_<extract_number>.csv.gz*.

This work was supported in part by the EC within the H2020 Program under projects MOSAICrOWN and MARSAL, by the Italian Ministry of Research within the PRIN program under project HOPE, and by JPMorgan Chase & Co under project "k-anonymity for AR/VR and IoT/5G".

Note that the sample of the dataset is randomly extracted at each download from the IPUMS USA web site. Hence, it may differ from the one used in our experimental evaluation.

Artifact execution. The procedure to run the experiments has been automated and can be started running command `make artifact_experiments` from the root folder of the project. The procedure operates as follows:

1) it cleans the test environment stopping every Docker container that is still running and removing from HDFS the results produced by the previous runs;
2) it extracts the sample of IPUMS USA dataset to be anonymized from the archive and copies it to the *Spark Driver* volume;
3) it runs the centralized and distributed version of the Mondrian algorithm (see below), and measures the execution time and information loss, storing the results with the following directory structure:

```
mondrian/
 |-- percom_artifact_experiments/
 |-- |-- results/
 |-- |-- |-- runtime_results_<TIMESTAMP>/
 |-- |-- |-- loss_results_<TIMESTAMP>/
```

4) it shuts down all the containers except the *Spark History Server*, which remains available to keep track of the previous runs of the artifact.

Centralized version. The centralized version of Mondrian corresponds to the baseline of the experimental results in [2]. The execution of the algorithm can be monitored through the messages showed on the terminal, which report:

1) the schema and the first few tuples of the input dataset;
2) each decision taken by Mondrian to cut the dataset;
3) the schema and the first few tuples of the anonymized dataset;
4) a summary of the information loss measures and the execution time of the algorithm.

The anonymized dataset is in folder *local/anonymized*.

Distributed version. Given a number n of workers available in the distributed system, the artifact performs the following steps to execute the distributed version of the Mondrian algorithm:

1) start all the Docker services, initialize HDFS, and submit to the *Spark Driver* our Spark Application;
2) recover the dataset from HDFS and show its structure;
3) retrieve the n-quantiles of the best-scoring attribute of the dataset, showing the score used to decide the optimal cut and the size of the partitions;
4) show the first few tuples of the dataset, complemented with a new attribute containing the id of the quantile to which each tuple belongs and hence the worker to which the tuple is assigned;
5) anonymize the dataset;
6) show the first few tuples of the anonymized dataset, with a summary of the execution time.

The anonymized dataset is in folder *distributed/anonymized*.

V. Experimental results

We discuss the experimental results presented in [2], obtained as discussed in the previous section.

Hardware requirements. The artifact has been deployed on a machine equipped with:

- a CPU with 20 logical cores;
- 40 GB of RAM;
- 15 GB of free disk space on an SSD.

Execution time. This experiment measured the execution time when computing a 3-anonymous and 2-diverse version of a sample of the IPUMS USA dataset. The results of the experiments are stored in folder `runtime_results_<TIMESTAMP>`. First, the artifact runs the centralized version of the Mondrian algorithm. The results are saved in file *centralized_results.csv*. Then, the artifact runs the distributed (Spark-based) version of the Mondrian algorithm, varying the number of workers from 2 to 20. The results are saved in file *spark_based_results.csv*. Besides generating the .csv files with the execution time of the centralized and distributed versions of the algorithm, the artifact plots these results generating file *comparison.pdf*. Note that the absolute times obtained running our artifact may slightly differ from the ones in Figure 3 in [2], due to the differences in the hardware of the machine used. We however expect the shape of the curves to be similar, proving the scalability of our distributed version of the algorithm.

Information loss. This experiment measured the information loss when computing a 5-anonymous and 2-diverse version of a sample of the IPUMS USA dataset. The results of the experiments are stored in folder `loss_results_<TIMESTAMP>`. The artifact first runs the centralized version of the Mondrian algorithm, storing the results in file *centralized_results.csv*. Then, it runs the distributed version (with 5, 10, and 20 workers), using a sample including 0.01% of the dataset to determine the most suitable attribute and compute the n-quantiles (with $n = 5$, $n = 10$, and $n = 20$, respectively) for partitioning the dataset among the workers. The results obtained from five runs of the distributed version of the algorithm are stored in file *spark_based_results.csv*. The artifact also generates file *loss_table.csv*, which reports the average and the variance (in the form $\mu \pm \sigma$) of the results in file *spark_based_results.csv*. Note that, since the sample of IPUMS USA dataset is randomly extracted at each download, it may be different from the one used in our experiments and consequently the results might be slightly different from the ones in Figure 4 in [2]. However, we expect the results to have a similar trend, confirming the limited impact of sampling on information loss.

References

[1] https://github.com/mosaicrown/mondrian

[2] S. De Capitani di Vimercati, D. Facchinetti, S. Foresti, G. Oldani, S. Paraboschi, M. Rossi, and P. Samarati, “Scalable distributed data anonymization,” in *Proc. of PerCom 2021*, March 2021.

[3] S. Ruggles *et al.*, “IPUMS USA: Version 10.0 [dataset],” Minneapolis, MN: IPUMS, 2020, https://doi.org/10.18128/D010.V10.0