Finding the most Valuable parts of a Dataset

Data Intensive Systems Course, Utrecht University, 2025-2026

Instructor: Prof. Yannis Velegrakis

DEADLINE: **Monday 3 Nov, 2025 @ 23:59** **Number of Persons: 3**

Deadline for the Presentation files: 26 October @ 23:00

Situation Description:

A typical problem companies are facing nowadays is running out of storage space. The global datasphere (i.e., the total amount of data we have) is expected to surpass the 25 ZBytes of data. As a result, companies are always looking for ways to get rid of data, either because they are running out of space or will soon have to face such a situation. In this project we are looking for ways to identify and remove parts of the data in a dataset.

We assume we have a relational table R (recall that a table is a set of tuples) and for simplicity you can assume that all the columns of the table are integers. We also assume we have a set of queries Q on R of the form:

select * **from** R **where** cond

The cond is a conjunction of equality conditions of the form A=v, where the A is an attribute of R and v is a constant value. For example, if we have the table Person(Id, Name, Age, City, Salary), a query may be the:

select * **from** Person **where** Salary=1000 and Age=45

The results of any query q on the table R is the set of tuples of R that satisfy the conditions in the **where** clause of the query q. Let us denote such a set of tuples as Ans(q).

We denote by |R| the cardinality of R, i.e., the number of tuples it contains. Since the results of a query is a relation itself, we will write |q| and mean the number of tuples that the answer of the query q contains. We also refer to the popularity of a tuple t, and denote it as pop(t) as the number of queries in Q that contain the tuple t in their answer set.

Each tuple has some importance, denoted as imp(t) for a tuple t. Given a subset R' of R, the importance of the set R', which by abuse of notation is also denoted as imp(R'), is a value that depends on the tuples contained and their individual importance.

Assume that we have a threshold T which is a limit on the number of tuples we can have on the disk. If we have more, we have to delete some. In that case we would be interested to identify the set R' of tuples such that $R' \subseteq R$, also $|R'| \leq T$, and the importance or R', denoted as imp(R'), is maximized. In other words, we are interested in identifying and keeping the T tuples that will form the set R' and will be the most important set. The challenging question is to decide how we measure the importance of a tuple, how the importance of a set of tuples and how we can implement a method to find the best R'.

Method 1

One way to measure the importance of a tuple is to assume that it is equal to its popularity, i.e., the more queries a tuple belongs, the more valuable (important) it is, which means:

$$imp(t) = pop(t)$$

For the importance of a set of tuples R', we can assume that it is the sum of the importance of the individual tuples it contains, divided by the total popularity of the tuples in the original set R. In this way the importance of the set is between 0 and 1. In other words:

$$imp(R') = \frac{\sum_{i=1}^{|R'|} imp(t_i)}{\sum_{i=1}^{|R|} imp(t_i)} = \frac{\sum_{i=1}^{|R'|} pop(t_i)}{\sum_{i=1}^{|R|} pop(t_i)}$$

The answer to the problem is the R' that has cardinality $|R'| \leq T$ and has the maximum importance imp(R'). To find that R', one needs to consider all the possible R' sets, compute their importance and select the best.

Method 2

The solution above assumed that the value of each tuple in a set does not depend on the existence of other tuples in the set but only on its own popularity. However, if a tuple is similar to other tuples in the set, even if it is popular, its value is not that high (because there are so many other similar tuples). Thus, we can change the definition of the importance of a tuple in a set R' to be its popularity weighted by the average difference it has to the remaining tuples in the set. (The difference is defined as 1-similarity). This means that the importance of a tuple t∈R' is given by the formula:

$$imp(t) = pop(t) * \frac{\sum_{i=1}^{|R'-\{t\}|}(1 - sim(t, t_i))}{|R'|}$$

With this change, the importance of a set R' is again:

$$imp(R') = \frac{\sum_{i=1}^{|R'|} imp(t_i)}{\sum_{i=1}^{|R|} imp(t_i)} = \frac{\sum_{i=1}^{|R'|} pop(t_i) * \frac{\sum_{k=1}^{|R'-\{t_i\}|}(1 - sim(t_i, t_k))}{|R'|}}{\sum_{i=1}^{|R|} pop(t_i) * \frac{\sum_{k=1}^{|R-\{t_i\}|}(1 - sim(t_i, t_k))}{|R|}}$$

The answer to the problem is the R' that has cardinality |R'|≤T and has the maximum importance imp(R'). To find that specific R', one needs to consider all the possible R' sets and compute their importance.

Method 3

Method 3 adjusted the popularity of a tuple based on its similarity to each individual other tuple in the set. However, a limitation of this approach is that it does not take into consideration the relationship between the tuples already in the system. Thus, we can follow a different approach. We define the importance of a set of tuples to be the average distance they have from their centroid. This means that sets with very similar points will have a low importance (we need sets that are highly heterogeneous). So, given a set R', let $c_{R'}$ be the centroid of the set. The importance of a set R' is given by the formula:

$$imp(R') = \frac{\sum_{i=1}^{|R'|}(1 - sim(t_i, c_{R'}))}{|R'|}$$

To understand the importance of a tuple for a set, we have to see how the tuple changes the value of the whole set. We refer to this as the residual importance for that set. In other words

$$resimp_{R'}(t) = imp(R' \cup \{t\}) - imp(R')$$

The importance of a tuple t in the set R is then defined as the sum of the normalized residual importance over all the possible subsets that can exist from the set R that do not contain t. In particular:

$$resimp(t) = \sum_{S \subseteq R-\{t\}} \frac{|S|! * (|R| - |S| - 1)}{|R|!} resimp_S(t) = \sum_{S \subseteq R-\{t\}} \frac{|S|! * (|R| - |S| - 1)}{|R|!} [\, imp(S \cup \{t\}) - imp(S) \,]$$

Once the residual importance of every tuple is calculated, the answer to the problem at hand is the set of T tuples with the highest residual importance (recall that T is the threshold determining the maximum number of tuples we can keep in our store)

Goal

In this project you are asked to develop a solution in Apache Spark (and any programming language you prefer) that computes the set R'. It takes as input the name of a file that contains the table R, the threshold T (an integer that is smaller than the total number of tuples |R| in the relation R), and the name of the output file, i.e., where the results will ne stored. The program the table R' saved in the output file. Clearly the tuples in the output file constitute a subset of the tuples in R. Since there are 3 methods for computing this (described previously), you need to actually develop 3 independent programs, called method1, method2, and method3.

A table R is provided as a text CSV file where the first row contains the names of the columns separated by comma, and each of the following rows corresponds to a tuple. Each row contains the attribute values of the attributes of the tuple separated by a comma.

You need to also evaluate and compare these methods. To do so, you need to use datasets of different sizes and see how the methods perform, how well they scale to different sizes, what are their limits, and how they compare to each other.

There is no specific dataset that is given to you. But you can create one (this is called synthetic Dataset Generation) in a way that you test your program. You will need to devise a program that generates such a dataset (or use a synthetic data generator). You should be able to provide the number of columns that the dataset will have and the number of rows.

You also need to create a set of queries (randomly) to be able to measure popularity of the different tuples.

Delivery

You need to deliver

1. the code of the program you developed, alongside instructions on how the program runs
2. the dataset you used, and
3. a report in which you describe the solution you have devised and the results of the experiments you have performed to prove the effectiveness and efficiency of your solution.
4. You will also be asked to make a 3-5 min presentation

If you have the group already formed, send the names and student id of the 3 group members to the instructor (i.velegrakis@uu.nl). The instructor will then create a private channel for you on MS Teams which is where you will deliver the project. If you are still looking for partners for the project there is a forum dedicated to that on Brightspace (or you can ask your colleagues in the classroom).

The number of your group is in the name of your MS Teams channel.

In the Files of the teams channel of your group, create the following directories (Pay attention to the capital letters):

1. **Src** (place the code of your project. Have also a README.txt file that explains how it runs.)
2. **Report** (place a file XX.pdf where XX is the number of your group. This will be your final report. See instructions in the next section)
3. **Presentation** (place her a power point presentation (NOT a PDF) called XX.pptx (where XX is the number of your group)).
4. **Data** (place here any datasets you used for the experiments and any output they generated)

PLEASE USE THE NAMING of files and folders EXACTLY AS REQUESTED. Do NOT put any extra information, characters of change something. The files are collected by a script program. If you use a different name, the script will copy nothing and the project will be marked incomplete.

Presentation

On the last week of the lectures (see the lecture schedule for the exact day) there will be a presentation in which a representative of every team will make a 5 min (maximum) presentation of the solution the group is developing. At least one slide is required for each of the following topics.

1. Project/group name and members (**with photos of the members** so that we know who is with whom) (0.3 points)
2. Solution (0.4 points)
3. Datasets + Experiments (what you plan to test and how you created the dataset) (0.3 points)

The presentation should be in line with the final project that will be delivered. Changes are of course allowed but one cannot have a dramatically new solution. The name of the power point file should be X.pptx where X is the number of your group. The number of your group is in the name of your channel.

Evaluation Criteria

The project is evaluated according to the following evaluation criteria:

1. Novelty & sophistication of the idea as well as how well it solves the problem.
2. Technical Depth: Detailed description of the approach and its challenging choices.
3. Presentation: Clarity and Completeness of the report & the presentation.
4. Experimental Evaluation
   4.1. The dataset(s) that have been used in the evaluation
   4.2. The evaluation tests that have been made (what has been tested and how)
   4.3. The comments on the results of the evaluation

Structure of the report

The final report should be written in Latex, using the following template which is available on overleaf: https://www.overleaf.com/read/bgrgzbqhqkjr#412001 It should contain the following sections:

1. **Introduction** (maximum 1 page) in which you introduce the problem you are solving, its importance and the main highlights of your solution (1 paragraph) and the results of your experiments (1 paragraph). Provide a motivation for this work. (Why you think that such a study is important? (you already have an application so it is clear that it is important, but maybe you can think additional applications to make the statement stronger). And why it is challenging (i.e., not trivial) to perform this processing? What were the hard/challenging parts in developing a solution?) Note that a "hard/challenging" part should be generic and not personal to the authors. They should apply to everyone and are challenging due to the nature of the problem at hand. They should not be challenging just because of the capabilities of the author. For example, if the solution is developed in python and the programmer does not know python, then clearly the difficulty is only for the specific author and not for everyone. The goal of the introduction section is to show that you have understood the problem.
2. **Related work** and technologies (maximum 1 page): In case you have used some package in your code, then explain this here. If you only used spark and python, **you do NOT need to have this section**. You do not need to explain what spark is or how important it is. Do not waste space getting into details that everyone else knows already from the lectures or other online sources. Keep it to the basics and to the minimum.
3. **Solution**. In this section you describe in detail what your solution is (or what your solutions are, in case of more than one). The more detailed you are in this section the better the section is. Imagine that you give your report to someone else, and you ask her/him to implement your solution. Will that person be able to do it by looking only at what is written in the document? If yes, then the document is successful. Also explain the reasoning behind every choice you are making. You are free to include some pseudocode because it makes it much easier for people to understand what the text is saying.
4. **Experimental evaluation**. This section contains a detailed description of all the experiments you have done to understand how well your solution works. How does it compare to some baseline? The more things you are testing, the more it helps to understand the performance of the solution, and the better the report is. The size of the section is up to you, since it depends on the complexity of the solution you are proposing and the details you would like to study. Make sure that you also provide a description of the datasets you used as input. After doing the experiments (or where you describe the experiments) you need also to study the results, meaning explain what you observed and how you explain that.

On Brightspace there are the experiments sections of two sample reports from previous years. They are different topics of course but are an indication of how detailed one should go.

Note that given the detailed description of the problem provided here, you do NOT need to have a project statement section in your report