Team project: Discovery of combinations of vectors with low aggregate variance

(Deadline 24 March, 23:59, 50% of the course grade)

Context and learning outcomes: You will use the Spark platform and synopses for discovering interesting combinations of vectors. In the context of this project, a combination of time series is deemed interesting when the element-wise summation of all vectors produces a new vector with variance less than a threshold τ. The learning outcomes of the project include: (1) understanding the practical relevance of big data platforms in real-world problems, (2) being able to quickly master and make practical use of big data platforms, and (3) being able to implement and exploit synopses.

Team formation: Form teams of 6 persons each. You are free to form your own teams.

Submission details: Submit all relevant material for the milestone in Canvas before the deadline. Each team should submit exactly one submission. No late submissions will be accepted.

Dataset: You are given a dataset generator, that will generate vectors of size 10,000 (details provided in canvas). All values in the generated vectors are **non-negative** integers. Remember to **provide the correct group id** when generating the dataset.

Definitions

Aggregate vector. Consider a set of vectors $S = \{X_1, X_2, \dots\}$. By $X[i]$ we refer to the value at the i'th position of vector X. The aggregate vector $\widetilde{X_S}$ of the set S is defined as the component-wise summation of all vectors in S, i.e.,

$$\widetilde{X_S}[i] = \sum_{X \in S} X[i]$$

Variance. The variance of a vector X of length l is: $\sigma^2 = \left(\frac{1}{l}\sum_{i=1}^{l}(X[i])^2\right) - (\mu_X)^2$
where

$$\mu_X = \sum_{j=1}^{l} \frac{X[j]}{l}$$

denotes the mean of all values contained in the vector.

Aggregate variance. Consider a set of vectors $S = \{X_1, X_2, \dots\}$. Let the aggregate vector of S be $\tilde{X}_S$. The aggregate variance of this set is defined as the variance of the aggregate vector $\tilde{X}_S$ of S.

Example. Consider the following example (for simplicity, the vectors are only three-dimensional):
X1=[1,3,5], X2=[5,1,0], X3=[-3, 2, 4]

The aggregate vector of the set S={X1, X2, X3} is then:

$\widetilde{X_S}$=[1+5-3, 3+1+2,5+0+4]=[3,6,9].

The aggregate variance of $\widetilde{X_S}$ is Var{3,6,9}= 6.

Detailed instructions

Q1 (0% of the project grade)

Write code in Spark to import the datasets for the following questions (a) as a data frame/dataset, and, (b) as an RDD. Include the full code in the submission.

Q2 (20% of the project grade)

Generate a dataset of **250 vectors**. Write code **with SparkSQL** to find all triples of vectors <X,Y,Z> from the dataset, with aggregate variance at most τ. You are allowed to use user-defined (aggregate) functions, if you want, but this is not required – alternative approaches might be more efficient.
Report/deliver the following information/code:

a) The SQL query (you can keep parameter τ in the SQL line) [Report and poster].
b) For τ={20,50,310,360,410}, a plot showing the number of results (you do not need to include the actual triples in your report) and the execution time on the provided cluster [Report and poster].
c) Description of how you analyzed the performance of the SQL, optimization methods you have tried to improve the performance, and their impact in performance [Description in the report, summary in the poster].
d) Include the full code in the submission.

Hint: For c), have a look at the EXPLAIN command in SparkSQL, to identify if there exist optimization opportunities, and how you handle them.
The expected length of Q2 in the report is at most ¾ of the page (A4, single column, font size greater than or equal to 10).

Q3 (40% of the project grade)

Generate a dataset of **1000 vectors**. For τ={20, 410}, find all triples of labels of the vectors <X,Y,Z> with aggregate variance at most τ, **without SparkSQL**. Report/deliver the following information/code:

a) A brief description of your system architecture. Present your architecture in a concise way using a diagram/figure that includes the data/processing flow and the series of transformations/actions applied on the input data. The functionality of each transformation/action can be explained with a brief text below the figure. We will use/have used such descriptions in the class. You can also find a short example in Appendix A [Report, and summary in the poster].
b) Optimization tricks that you used, and their impact in performance (i.e., execution time) [Report, and summary in the poster].
c) Also run the code of Q3 on the dataset generated for Q2 (the 250 vectors). Are results of Q3 identical to the results of Q2, for the same τ value? If this is not the case, explain. Also report execution time of the code on the provided cluster [Report and poster].
d) A discussion (at most ½ page) to compare and explain the difference in performance between Q2 and Q3. Be succinct in your explanation [Report].
e) Include the full code in the submission.

Hints:

- Use the web UI of Spark to understand what takes a lot of time.

- You are expected to explore many different optimizations, and possibly different configurations/parameterizations (if applicable) to find the optimal (combination of) configurations. Describe all explored optimizations – even if these did not lead to significant improvement. Some of the possibilities to explore are: (a) the impact of the number of partitions on performance, and ways to reduce the number of tasks, (b) the impact of broadcasting variables, (c) the impact of caching, (d) commutativity or other techniques to avoid redundant computations.

The expected length of Q3 in the report is one page.

Q4 (40% of the project grade)

Generate a dataset of **250 vectors**. Modify your solution of Q3 (or write a solution from scratch) such that all operations inside Spark are performed on sketches, to approximate the result. Precisely, choose one of the sketches taught in this course, and find a way to use this sketch for representing the original vectors, estimating the aggregate vectors, and finally, estimating the variances inside Spark. Your solution is expected to implement the following functionalities:
(functionality 1) find all triples of vector ids with an aggregate variance lower than a threshold,
(functionality 2) find all triples of vector ids with an aggregate variance higher than a threshold.

40% of the grade of this question will be given to the theoretical derivations and concrete explanation on how this will work (pseudocode, description, and short accompanying explanation), whereas the remaining 60% will be awarded for a correct implementation in Spark.

Report/deliver the following information/code:

a) The name of the sketch. Note that the name alone is not sufficient for acquiring any points [Report and poster].
b) A concrete explanation on how this sketch can be exploited to estimate the aggregate variance, including pseudocode. The pseudocode should be sufficient such that an experienced spark coder can implement it, without knowing anything about the sketch [Report and summary in the poster].
c) Concise description of your Spark solution with a diagram and a short explanation (similar to Q3) [Report and summary in the poster].
d) Theoretical results **and** a short explanation that allow a user to choose the correct parameters for the sketch, in order to achieve ε/δ guarantees. The theoretical results should be provided and proved as a single theorem (e.g., similar to Theorem 3 from the Count-min paper[1]) [Report and summary in the poster. Be prepared to explain/repeat the proofs from the report during the poster session].
e) A Spark implementation that works on the provided data. The implementation should accept the filename that contains the vectors, the desired ε/δ parameters, the desired functionality (1 or 2) and the threshold.
f) For functionality 1, execution time of your implementation on the provided cluster for ε={0.001, 0.01}, δ=0.1, and τ={400} [Report and poster].
g) For functionality 2, execution time of your implementation on the provided cluster for ε={0.0001, 0.001, 0.002, 0.01}, δ=0.1, and τ={200000, 1000000} [Report and poster].
h) The corresponding precision/recall values of the results retrieved by your code at sub-questions f and g [Report and poster].
i) A brief explanation for your observations in sub-questions f, g, and h [Report and summary in the poster].
j) A brief discussion about the usefulness of the sketch for optimizing the above functionalities and parameters. In which of the above configurations was the sketch useful, and in which

[1] http://dimacs.rutgers.edu/~graham/pubs/papers/cm-full.pdf

configurations it would instead make sense to compute the exact result, without the sketch? [Report and summary in the poster].

k) A brief discussion about the tightness of the bounds of the sketch [Report and poster].
l) Include the full code in the submission.

The expected length of this section is 1.5 page.

Hints:

- Since the sketch will be built within Spark, it should be implemented in a way that Spark can parallelize its construction. Make sure you include the related explanation in your answer for sub-question b.
- You might want to check method broadcast, in Spark. Is it useful?
- Instead of collecting all answers to the master node, it might be easier/sufficient to count them.

Q5 (0% on the project grade)

Imagine that you are working at a large company, which wants to compute vector combinations with a low aggregate variance on a large dataset of vectors. You have the following options, and you need to suggest a solution to the company. How would you decide between these options? Which are the deciding properties/parameters that you will consider before suggesting a solution?

a) Relational databases
b) A multi-threaded program that runs on a powerful server
c) Spark, on a company-owned cluster
d) Spark, on cloud resources
e) An approximation technique, similar to the solution of Q4, but running on a powerful server
f) An approximation technique, similar to the solution of Q4, executed on cloud resources

Include a compact summary, only in the poster under the section title: "Ethical and other aspects" (a very short paragraph, to help the discussion during your poster presentation). Do not include this part in your report.

Constraints for the whole project:

- You can use Python or Java for your Spark implementation.
- Perform the development/debugging/profiling of the code locally. The capacity of the servers is limited, and you are too many.
- Your solutions should run successfully on the provided server, and with the libraries that are already installed there or are already included in the JVM. You are not allowed to use other third-party libraries that are not installed on the server.
- On the server, you will have 30 minutes per submission, to execute each question. After this time, your solution will time out. **If your solution needs more than this time, you should improve the performance of your solution.**

Deliverables:

You are expected to submit:

a) For java solutions:
   - An executable file app.jar. The file needs to include all dependencies/libraries (a so-called "fat-jar"). It should follow the provided template, such that it can be submitted to the submission system without changes.
   - A zip file containing the code that was used to generate the jar. Make sure that the code compiles normally.

b) For python solutions:
   - A zip containing the source code. It should follow the provided template, such that it can be submitted to the submission system without changes.

c) A pdf report of around 3 pages (not more than 4 pages), containing:
   - The answers to the above questions
   - A small table, detailing the contribution of each team member (approximate percentage of work per member, **and what each one did**). This table will be used to decide on your individual project grades, when there is a significant deviation on the effort of the members.
   - Aim for compactness. Feel free to use tables, in order to summarize the results in a succinct way and save space.
   - References are not included in the page limit. However, references should be from other authors, and not technical reports/extensions of the submission

d) A 1-page poster outlining your contribution and results. This should be the exact same poster that you will present during the poster session. You are not allowed to change/update it after the deadline.

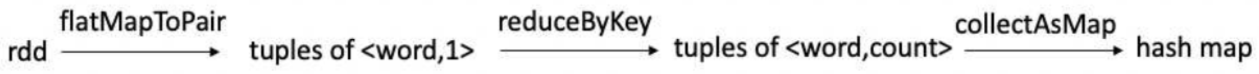
Include all submission files in a single zip. Upload the zip in Canvas before the deadline.

Frequently asked questions

- **Guiding principles for preparing the poster:** The poster and your live presentation (during the poster session) should sufficiently demonstrate your work and insights, as these are described in your report. The idea is to show clearly what you have done so far, and why you deserve a good grade. You may use the same structure as in the report. The poster should be self-standing; it should not require the audience to read the report. The poster will serve as a visual aid for us (and you) when you explain your work to us. It should not include a lot of text and long sentences. Use illustrations/figures, bullets, and tables, to help you organize the discussion and presentation of your work. To prioritize what should be included in a poster, think: i) what do I need to make the presentation clear and understandable, and ii) what information will be appreciated most by the audience, and. It is very rare that the answer for any of these questions is: 'a lot of text'.
- **Preparing for the poster session:** At least one teacher will approach you and ask you to present the poster in three minutes. Your presentation should cover the whole poster. **Practice your presentation well**, such that (a) it makes sense to the audience, (b) it is consistent, from start to end, (c) it takes at most 4 minutes, and (d) it stresses the things you tried and learned, and the reasons you deserve a good grade! After your presentation, the teacher will ask further details and possibly provide feedback. Be prepared to run your solution locally, on one of your laptops. You can run it on a smaller dataset, if needed, such that each question completes in 1-2 minutes.
- **Who needs to attend the poster session**: All team members are expected to join their team's poster presentation. Also, all members are expected to be ready to describe the poster and answer questions, independent of the internal delegation of work inside the team. Inability of a team member (or the whole team) to explain and discuss the solution, results, and poster, will lead to significant deduction of grades for the whole team, even if the delivered solutions and report are correct. In other words, if you cannot convincingly explain and discuss your solution and results, you will not get points, even if the solution works. Make sure that you discuss and explain your solutions inside the team, before joining the poster session.

Appendix A. An example of a brief description of the system architecture

Consider the word-count problem that we discussed repeatedly in the class. The diagram and expected explanation for that code follows:



And the additional required details are:

flatMapToPair: break the text to words. For each word create a tuple <word,1>
reduceByKey: sums up all values for each word. Output is one tuple of <word,count> per distinct word, where count corresponds to the sum
collect: returns a hashmap that includes all discovered words, and the number of occurrences of each word.

The expected details for each operation are 1-2 lines maximum. You do not need to include code in your report and poster, unless this helps!

GOOD LUCK