Advanced Algorithm Project 2023-2024

Matrix Maximum Segment Problem

October 19, 2023

1 Description

The objective of this project is to implement some algorithms for solving the *Matrix Maximum Segment Problem* in 2 Dimensional scenario - known also as the Maximum Subarray Problem - and to provide an experimental study of their running time and quality.

1.1 Problem Description

The Maximum segment sum problem, is the task of finding a contiguous segment with the largest sum, within a given two dimensional array:

1.1.1 Formulation:

Given a 2-D array $\mathbf{A} = (\mathbf{a_{i,j}})_{\mathbf{1 \leq i \leq M, 1 \leq j \leq N}}$ of size MxN with integers, find the indices i_1, i_2, j_1, j_2 where $1 \leq i_1 \leq i_2 \leq M$, $1 \leq j_1 \leq j_2 \leq N$ such that the sum $\sum_{\mathbf{x=i_1}}^{\mathbf{i_2}} \sum_{\mathbf{y=j_1}}^{\mathbf{j_2}} \mathbf{A}[\mathbf{x}, \mathbf{y}]$ is as large as possible.

For example, for the 2-d array A, the maximum contiguous segment with the largest sum is $\begin{pmatrix} -3 & 4 & 2 \\ 8 & 10 & 1 \\ -1 & 1 & 7 \end{pmatrix}$ with $i_1 = 2, i_1 = 4, j_1 = 2, j_2 = 4$ indices.

$$A = \begin{bmatrix} 1 & 2 & -1 & -4 & -20 \\ -8 & -3 & 4 & 2 & 1 \\ 3 & 8 & 10 & 1 & 3 \\ -4 & -1 & 1 & 7 & -6 \end{bmatrix} \tag{1}$$

1.1.2 Constrained Formulation

Given the same 2-D array A of size MxN with integers, and the shape constraints K, L. Find the indices i_1, i_2, j_1, j_2 where $1 \leq i_1 \leq i_2 \leq M$; $1 \leq j_1 \leq j_2 \leq N$; $i_2 - i_1 = K$; $j_2 - j_1 = L$ such that the sum $\sum_{\mathbf{x=i_1}}^{\mathbf{i_2}} \sum_{\mathbf{y=j_1}}^{\mathbf{j_2}} \mathbf{A}[\mathbf{x}, \mathbf{y}]$ is as large as possible. In simple terms, find the sub array of size KxL with the maximum sum possible.

2 Work to do

You must program different algorithms for 2-D Matrix maximum segment problem, including:

- The brute-force approach in exponential and polynomial time, exploring all the possible solutions in a systematic way.
- A Branch-and-Bound version, to be designed on your own.
- A greedy approach.
- A dynamic programming approach.
- A version based on a randomized approach.
- A version based on a genetic programming or ant colony approach.
- Another personal version.

3 Instructions

- Important: At the end of every session on friday, the members of the team must commit their progress to the Github with a summary of the week's work, the commit's will be used to track the progress of the teams and will be part of the grading.
- Note that all the algorithms must be able to output the solution - *i.e. the set of items selected and the value of the proposed solution.*
- A graphical user interface could be proposed to illustrate the behavior of the algorithms(s), but **this is not required** and must be done after the implementation of the algorithms.
- The running time of the algorithms and the quality of the solutions given (in terms of optimality) have to be evaluated on problems of different sizes and different values (*i.e. different variety of array element, positive and negative, a variety of array sizes*). You are free to propose your own benchmarks and tests.
- You have to provide a full and rigorous experimental study to illustrate the different strong and weak points of each algorithm. In particular, it is expected that the efficiency and the quality of the solutions are studied with care.
- The project can be done in a group from 4 to 5 students (exceptionally 3), and each student must program at least one algorithm. You are free to use any programming language among: python, C, C++, java, javascript. It is recommended to maximize the number of students of the same master track in each group to ensure a similar availability between group members.

4 What to send?

4.1 Provisional Planning:

- Each group must upload before **Wednesday October 25, 22:00** a text containing the members of the group on Claroline.
- In the **Group Description 23-24 resource**. This text **must also contain a provisional planning** giving the milestones of the project with the tasks to achieve, the reparation between the group members and the time deserved to each task, if possible the group can mention the procedures used for testing and evaluating the programs.
- The ability to respect the schedule will not be used for evaluation the grade, but during the defence the students must present the real planning and discuss the reasons why the project has run late.
- The group must also include the workload between the members of the group in percentage.
- Finally, the group must include the checklist provided at the end of the document in the report and states for each question what has been done (or not). The questions of the checklist cannot be modified. The quality of the presentation (report, oral, ...) and the quality of the source code will be taken into account.

4.2 Github

- Create a private github repository for the project and include the repository details in the Provisional Planning text file.
- In addition, to the previous step, provide access for the user srikalidindi@outlook.com to your private repository.

Sri Kalidindi will supervise the advancement of the project, you will have 5 meeting for discussing your achievements. **Note that all the questions and mails related to the project must be sent to sri.kalidindi@univ-st-etienne.fr**.

The final version of the project must be uploaded at **Project archive 23 24** resource in Claroline and and also in the GitHub before **Monday the December 4th**, in a .zip or .tar.gz archive, **well-organized**. It must include the source code of the programs, information for installing and using them, and a report in PDF format. A project defense will be scheduled on **8th December** morning or afternoon (the exact schedule will be given later), be careful to come to the defence with a working implementation.

5 Grade scaling:

- at least 7/20 : greedy and dynamic programming versions must work correctly, and be evaluated in a rigorous manner on artificial data, report and source code must be presented neatly.
- at least 10/20 : in addition to the previous point, each member of the group must program a different approach, experimental evaluations on artificial data must be rigorous, source code and report must be presented neatly and the answers to the questions must be correct.
- at least 12/20 : in addition to the previous point, an implementation of the constrained formulation must be done and the number of methods implemented must be higher than the size of the group.
- at least 14/20 : implement another method for both simple and constrained formulations (the number of methods must be higher than the size of the group).
- at least 16/20 : in addition to the previous point, implement two new methods, source code and report must be extremely neat and clear.

Note: The absence of one element can be compensated by the addition of an element of a higher grade. If the full check list is not included in the report and commented, the grade can be divided by 2. **Be careful, the grading scale is given for information purpose.**

6 Checklist

1. Did you proofread your report?
2. Did you present the global objectif of your work?
3. Did you present the principles of all the methods/algorithms used in your project?
4. Did you cite correctly the references to the methods/algorithms that are not from your own? - Did you include all the details of your experimental setup to reproduce the experimental results, and explain the choice of the parameters taken?
5. Did you provide curves, numerical results and error bars when results are run multiple times?
6. Did you comment and interpret the different results presented?
7. Did you include all the data, code, installation and running instructions needed to reproduce the results?
8. Did you engineer the code of all the programs in a unified way to facilitate the addition of new methods/techniques and debugging?
9. Did you make sure that the results different experiments and programs are comparable?
10. Did you comment sufficiently your code?
11. Did you add a thorough documentation on the code provided?
12. Did you provide the additional planning and the final planning in the report and discuss organization aspects in your work?
13. Did you provide the workload percentage between the members of the group in the report?
14. Did you send the work in time?