**ECE 4094**

**Project A**

**Progress Report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **First Name** | **Last name** | **ID** | **Email** |
| **Supervisor 1** | **James** | **Saunderson** |  |  |
| **Supervisor 2** | **Jonathan** | **Li** |  |  |
| **Student 1** | **Simon** | **Teshuva** | **24195537** | **Srtes1@student.monash.edu** |

Contents

[**Introduction** 3](#_Toc505118655)

[**Objectives** 4](#_Toc505118656)

[**Progress to date** 5](#_Toc505118657)

[**Work to be completed** 6](#_Toc505118658)

# **Introduction**

When modelling large datasets, it is useful to evaluate the covariance matrix. As part of the calculations, the logdet of the matrix needs to be computed. As the matrix gets very large, this logdet computation becomes exceedingly time and space inefficeint. In “paper name”, a method to compute an esimate for the logdet of a sparse matrix was proposed which uses a Chebyshev Polynomial function and a Monte-Carlo approach. This method computes the logdet to a very high degree of accuracy, at a much faster rate than the exact method, and uses far less space.

A flaw in this method is that it requires knowledge of the condition number to perform certain elements of the computation. James Saunderson has proposed a new method, which uses a rational function estimator instead of a Chebyshev Polynomial to eliminate the need for the condition number. The primary goals of this project are to implement the method shown in “paper name” and the method proposed by James Saunderson, and compare the effectiveness of the two methods.

# **Objectives**

|  |  |  |  |
| --- | --- | --- | --- |
| Objective Name | Brief Description | Completed | Realistic |
| Generate LSPSDM using the method in “paper name”. | This is diagonally dominant, which is a special case of of the data we will be working with. It is easier to generate than a non-diagonally dominant matrix. | Yes | Yes |
| Implement method in “paper name” | The paper [1] introduces a new algorithm for estimating the log determinant of a LSPSDM. It uses a combination of Monte-Carlo methods and Polynomial funciton estimation for the log function | Almost | Yes |
| Experiment 1 from “paper name” | Run code many times over datasets of varying sizes. Use algorithm form paper and cholesky decomposition and produce graphs similar to paper. | Yes | Yes |
| More advanced analaysis | **Ideas for analysis** | No | No |
| Implement new method | James Saunderson has developed the outline for a new method for estimating the log determinant. This method uses a rational function estimator for the log function and as a result is not reliant on knowing the condition number of the matrix. | In Progress | Yes |
| Progress Report | See introduction to this document | In Progress | Yes |
| Design Specifications | A document detailing the design choices made in this project and the reasons for said choices. | In Progress | Yes |
| Some simple comparisons of old and new | Runtime, space, relative accuracy compared to exact method. | No | Yes |
| Paralellise new | Use MATLAB’s inbuild parallel toolbox to paralellise the new algorithm | No (S1) | Yes |
| Some simple comparisons of old and parallel new | Runtime, space, relative accuracy compared to exact method. | No (S1) | Yes |
| Some advanced comparisons of old and advanced new | **Ideas for analysis** | No (S1) | Yes |
| Real world dataset analysis | Find real world data that can be converted to relevant format and be analysed by parallel new method for final results | No (S1) | Optional |
| More advanced dataset | Generate random dataset that is not diagonally dominant | No (S1) | Optional |
| GUI | Make a GUI that allowes user to get a visual indicator of the performance of the code | No (S1) | Optional |
| Final Report | A detailed report summerising the entire project | No (S1) | Yes |
| Poster | A poster to present at the Spark Night | No (S1) | Yes |
| Video | A short video summerising the project, | No (S1) | Yes |

# **Progress to date**

The initial part of project was to research various parts of the theory behind the project. While I was familiar with much of it beforehand, the refresher was very useful, and some fo the fields were new to me as well. Some of the fields I looked into are:

* Positive Semi-Definite Matrix:
  + A matrix where all the eigenvalues are non-negative.
* Sparse Matrix
  + A matrix where the overwhelming majority of the entries are 0.
* Matrix functions
  + Each of the polynomial functions has an analogue Matrix function.
* Cholesky decomposition
  + An exact method for computing the logdet of a matrox
* Chebyshev polynomials
  + A series of orthogonal polynomials used in “paper name”
* Gershgorin circle theorem
  + A method used to bound a square matrix.
* Gaussian-Legendre quadrature
  + A method of estimating the integral of a function. Uses a series of predefined weights and nodes to estimate the integral.

After ensuring that I had a solid grasp of the theory I began writign code to generate a dataset that could be used in testing and experimentation. The dataset needed to be in the form of a Large (Positive Semi-Definite Matrix (LPSDM) stored in sparse form. The primary MATLAB functions used in this section are kron, rand, randi and sparse.

Once I could reliable generate a LPSDM I began writing code to replicate the algorithm devised in “paper name”. This algorithm iterativley updates a guess for the logdet. It requires knowledge of the condition number of the matrix, which is approximated using Gershgorin Circle Theorem. The codition number is then fed into an equation which is then used to generate a chebyshev approximation for the log function used in the algorithm.

With the algorithm from “paper name” working I began working on replcating the first experiment performed in said paper. The experiment produces the following results;

* Runtime vs Matrix size graph (1e3 to 1e7)
* Relative accuracy compared to Cholesky decompostion, an exact method (1e3 to 3e4)
* Runtime comparison for cholesky decomposition vs algorithm
* Comparison to method used by Zhang & Leithead, 2007 using n = 1000

The first three experiments have been performed, and I am in the process of setting up the fourth one. In addition, I have began implementing the new method. I started by implementing the gauss-legendre quadrature rule on a matrix of size 1x1 and will now begin use that code to generalize the method to a large matrix. After this, I will perform some comparisons on time and space efficeincy of the two methods, as well as comparing their relative accuracy.

# **Work to be completed**

|  |  |  |
| --- | --- | --- |
| **Tasks and expected time allotment** | | |
| **Task** | **Expected hours** | **Start week** |
| Some more advanced alalysis of old | 15 | 1 |
| Some simple comparisons of old and new | 5 | 1 |
| Parallelise new | 20 | 3 |
| Some simple comparisons of old and parallel new | 5 | 3 |
| Some advanced comparisons of old and parallel new (\*) | 15 | 5 |
| Poster | 15 | 7 |
| Final Report | 40 | 7 |
| Video | 10 | 7 |
| Real world dataset analysis (\*) | 20 | 8 |
| More advanced dataset (\*) | 20 | 8 |
| GUI (\*) | 25 | 10 |

\* optional goal for late stages of project

