ECE472 — Methods and tools for big data

Lab 6

Manuel — JI (Summer 2024)

Goals of the lab

- Get familiar with Spark
- Investigate new algorithms
- Run simple data analysis

1 Introduction

In this lab you will implement a simple algorithm in Spark. Please submit your codes as well as the results in your lab report.

With your help to setup the database, Reapor Rich started to further research on the movie market. Krystor, an expert in *Bayesian Analysis* and *Numerical Analysis* developed a *Monte Carlo Simulation Model* to analyze a property of the movies.

At some stage in the analysis procedure, Krystor needs to perform an *LU decomposition* for a large square matrix. He has written some code in *R* with the built in function *lu.decomposition*, which uses *Doolittle Decomposition*, based on solving system of linear equations with *Gaussian Elimination*.

However. since the size of the dataset is too large, the algorithm does not perform well, and this step becomes the bottleneck for the whole algorithm. Now Krystor needs your to improve it.

The documentation of *lu.decomposition* can be found in `https://www.rdocumentation.org/packages/matrixcalc/versions/1.0-3/topics/lu.decomposition`

2 Cholesky Decomposition

In Krystor's algorithm, you found that the matrix which needs to be decomposed can be assumed to be a symmetric positive definite (SPD). The positive definiteness of a square matrix A is defined as:

$$x^T Ax > 0, \qquad \forall x \in \mathbb{R}^n \setminus \mathbf{0},$$

and if A is also symmetric ($A = A^T$), it is an SPD matrix.

After researching on internet, you found an alternative to *Doolittle Decomposition*: *Cholesky Decomposition* can be applied to an SPD matrix, and is roughly twice as efficient as *Doolittle Decomposition*.

If A is an SPD matrix, then there exists a matrix C such that $A = LL^T$, where L is a lower triangular matrix with positive main diagonal values, and $U = L^T$ is an upper triangular matrix with the same main diagonal values. *Cholesky decomposition* can be used to find L in this case.

Reference: `https://en.wikipedia.org/wiki/Cholesky_decomposition`

Algorithm. (*CHOLESKY DECOMPOSITION*)

Input : matrix $[a_{ij}]$ of $n \times n$

Output: matrix $[c_{ij}]$ of $n \times n$

Function `Cholesky`$([a_{ij}])$**:**
  for $i \leftarrow 1$ *to* n **do**
    for $j \leftarrow i+1$ *to* n **do**
      $c_{ij} \leftarrow 0$;
    end for
  end for
  for $j \leftarrow 1$ *to* n **do**
    $c_{jj} \leftarrow \sqrt{a_{jj}}$;
    for $i \leftarrow j+1$ *to* n **do**
      $c_{ij} \leftarrow a_{ij}/c_{jj}$;
      for $k \leftarrow j$ *to* i **do**
        $a_{ik} \leftarrow a_{ik} - c_{ij} \cdot c_{kj}$;
      end for
    end for
  end for
  return $[c_{ij}]$;
end

3 Implementation

In order to solve the Big Data Problem, you recalled a new tool *Spark* which was introduced recently. You therefore decide to give it a try and implement *Cholesky Decomposition* algorithm on *Spark*.

Krystor also provides you with some simple SPD matrices to test your implementation. He would be very grateful if you successfully perform *LU Decomposition* on the matrices.

$$\begin{bmatrix} 25 & -50 \\ -50 & 101 \end{bmatrix}, \quad \begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix}, \quad \begin{bmatrix} 7 & -2 & -3 & 0 & -1 & 0 \\ -2 & 8 & 0 & 0 & -1 & 0 \\ -3 & 0 & 4 & -1 & 0 & 0 \\ 0 & 0 & -1 & 5 & 0 & -2 \\ -1 & -1 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & -2 & 0 & 6 \end{bmatrix}.$$

Please send you implementation and the decomposition result to Krystor before the semester ends because he will not be able to spend much time on the movie analysis project afterwards.