

## VE472 — Methods and tools for big data

### Lab 5

Manuel — UM-JI (Summer 2021)

### 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 the help of the database setup by you, Reapor Rich started to do some 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.

In one step of the analysis procedure, Krystor needs to do *LU decomposition* to a large square matrix. He had 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, and the algorithm is not good enough, this step becomes a bottleneck of the whole algorithm. Now Krystor is finding you 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 needs to be decomposed can be assumed to be a symmetric positive definite (SPD) matrix. The positive definiteness of a square matrix  $A$  is defined as:

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

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

After researching on the Internet, you found that *Cholesky Decomposition* can be an alternative to *Doolittle Decomposition* when the matrix is an SPD matrix, and is roughly twice as efficient as it.

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](https://en.wikipedia.org/wiki/Cholesky_decomposition)

## Algorithm. (*CHOLESKY DECOMPOSITION*)

---

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

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

```
1 Function Cholesky( $[a_{ij}]$ ):  
2   for  $i \leftarrow 1$  to  $n$  do  
3     for  $j \leftarrow i + 1$  to  $n$  do  
4        $c_{ij} \leftarrow 0$ ;  
5     end for  
6   end for  
7   for  $j \leftarrow 1$  to  $n$  do  
8      $c_{jj} \leftarrow \sqrt{a_{jj}}$ ;  
9     for  $i \leftarrow j + 1$  to  $n$  do  
10       $c_{ij} \leftarrow a_{ij} / c_{jj}$ ;  
11      for  $k \leftarrow j$  to  $i$  do  
12         $a_{ik} \leftarrow a_{ik} - c_{ij} \cdot c_{kj}$ ;  
13      end for  
14    end for  
15  end for  
16  return  $[c_{ij}]$ ;  
17 end
```

---

### 3 Implementation

In order to solve the Big Data Problem, you recalled a new tool *Spark* which was introduced recently. You decided to implement the *Cholesky Decomposition* algorithm on *Spark*.

Krystor also provides you with some simple SPD matrices to test your implementation. He'll be happy if you successfully perform *LU Decomposition* on these matrices.

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

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