RESEARCH ARTICLE OPEN ACCESS

Review of Matrix Decomposition Techniques for Signal Processing Applications

Monika Agarwal*, Rajesh Mehra**
*(Department of Electronics and communication, NITTTR, Chandigarh)
** (Department of Electronics and communication, NITTTR, Chandigarh)

ABSTRACT
Decomposition of matrix is a vital part of many scientific and engineering applications. It is a technique that breaks down a square numeric matrix into two different square matrices and is a basis for efficiently solving a system of equations, which in turn is the basis for inverting a matrix. An inverting matrix is a part of many important algorithms. Matrix factorizations have wide applications in numerical linear algebra, in solving linear systems, computing inertia, and rank estimation is an important consideration. This paper presents review of all the matrix decomposition techniques used in signal processing applications on the basis of their computational complexity, advantages and disadvantages. Various Decomposition techniques such as LU Decomposition, QR decomposition , Cholesky decomposition are discussed here.
***Keywords* –** Cholesky Decomposition, LU decomposition, Matrix factorization, QR decomposition.

I. INTRODUCTION

Signal processing is an area of system engineering, electrical engineering and applied mathematics that deals with the operations on or analysis of analog as well as digitized signal, representing time varying or spatially physical quantities. There are various mathematical model applied in signal processing such as linear time invariant system, system identification, optimization, estimation theory, numerical methods [1].In the computational application matrix decomposition is a basic operation. “Matrix decomposition refers to the transformation of the given matrix into a given canonical form”. Matrix factorization is the process of producing the decomposition where the given matrix is transformed to right hand side product of canonical matrices [2]. Matrix decomposition is a fundamental theme in linear algebra and applied statistics which has both scientific and engineering significance. Computational convenience and analytic simplicity are the two main aspects in the purpose of matrix decomposition. In the real world it is not feasible for most of the matrix computation to be calculated in an optimal explicit way such as matrix inversion, matrix determinant, solving linear systems and least square fitting , thus to convert a difficult matrix computation problem into several easier task such as solving triangular or diagonal systems will greatly facilitate the calculations. Data matrices such as proximity matrix or correlation matrix represent numerical observations and it is often huge and hard to analyze them. Therefore to decompose the data matrices into some lower rank canonical forms will reveal the inherent characteristics. Since matrix computation algorithms are expensive computational tasks, hardware implementation of these algorithms require substantial time and effort. There is an increasing demand for domain specific tool for matrix computation algorithms which provide fast and highly efficient hardware production. In this paper there is review of matrix decomposition techniques that are used in signal processing.
The paper is organized as follows: Section II discusses the Matrix decomposition Models. Section III concludes the paper

II. MATRIX DECOMPOSITION MODELS

This paper discusses basically four matrix decomposition models.
QR decomposition
SVD decomposition
LU decomposition
Cholesky Decomposition

2.1 QR decomposition

A matrix A=QR in linear algebra is a decomposition of a matrix A into an orthogonal and right triangular matrix. The Q-R decompositions are generally used to solve the linear least squares problems. If the matrix A is non-singular then the Q-R factorization of A is a unique one if we want that the diagonal elements of R are positive. More generally we can factor a m x n rectangular matrix into an m x m unitary matrix and an m x n upper triangular matrix. There are two standard methods to evaluate Q-R factorization of A such as Graham-

Schmidt triangulizations and Householder triangulizations but Householder triangulizations are much more efficient than Graham-Schmidt triangulizations because of the following statistics involved. The number of operations at the kth step are multiplications=2(n-k+1)2 , additions=(n-k+1)2 + (n-k+1) (n-k) + 2 , division=1and square root=1. Summing all these operations over (n-1) steps we get the complexity as O (n3) but this involves much less computation time than the other triangulation methods [3]. QR factorization is an important tool for solving linear systems of equations because of good error propagation properties and the inevitability of unitary matrices [4]. The main application of QR decomposition is solving the linear least squares problem. There exist systems of linear equations that have no exact solutions. Linear least square is a mathematical optimization technique used to find an approximate solution for these systems. QR decomposition has several advantages that it finds least squares solution as well when no exact solution exists, and if there is exact solution, it finds all.

2.2 SVD DECOMPOSITION

This is also one of the matrix decomposition schemes. This is also applicable for complex rectangular matrices of the dimensions m x n. The singular value decomposition block factors the M-by-N input matrix A such that [4]

$$A=U\times \mathrm{diag}(s)\times V^* \qquad (1)$$

Where U is an M-by-P matrix, V is an N-by-P matrix, S is a length-P , vector P is defined as min(M,N) When M = N, U and V are both M-by-M unitary matrices M > N, V is an N-by-N unitary matrix, and U is an M-by-N matrix whose columns are the first N columns of a unitary matrix N > M, U is an M-by-M unitary matrix, and V is an N-by-M matrix whose columns are the first M columns of a unitary matrix In all cases, S is a 1-D vector of positive singular values having length P. Length-N row inputs are treated as length-N columns. Note that the first (maximum) element of output S is equal to the 2-norm of the matrix A. The output is always sampling based [4].

2.3 LU DECOMPOSITION

LU decomposition is widely used matrix factorization algorithm. it transforms square matrix into lower triangular matrix L and upper matrix U with A=LU. LU decomposition used the Dolittle algorithm for the elimination column by column from left to right. It results in unit lower triangular matrix that can use the storage of the original matrix A [5]. This algorithm requires 2n3/3 floating point operations. Algorithm 1 shows the steps followed for LU decomposition of a matrix. In each iteration , the same steps are applied to a matrix that has one less row and column than in the previous iteration. This algorithms shows the parallelism of each steps while in each iteration, a new sub matrix is constructed and hence efficient use of local storage capacity. It would be preferable to use an algorithm to load a block of data into local storage performs the computation and then transfer new block. The block LU decomposition algorithm partitioned the entire matrix into smaller blocks so that work can be done concurrently to improve the performance.

```
Input:A- n x n matrix with elemnts  a i,p
Output: L, U triangular matrices with elements l i p, u i p
1: for p=1 : n
2.    for q=1 to p-1
3.       for i=q+1 to p-1
4.          A[i,p]=A[i,p]- A[i, q]× A[q,p]
5.       for q=1to p-1
6.       for i=p to n
7.          A(i,p) = A[i,p]- A[i, q]× A[q,p]
8.       for q= p+1:n
9.          A(q,p)=A(q,p)/A(p,p)
```

Algorithm 1.Basic LU decomposition

The algorithm is analyzed as it writes lower and upper triangular matrices onto A matrices then it updates the value of A matrix column by column ((4) and (7)). The final values are computed by the division of each column entry by the diagonal entry of that column. The LU decomposition has various advantages. it is applicable for any matrix. It can be solved by forward substitution as well as backward substitution. The solution of triangular set of equation is trivial to solve. it is direct method and finds all solution. LU decomposition is easy to program. There are several drawbacks that it doesn't find approximate solution (least square). It could easily be unstable. Hence to find the approximate solution Cholesky decomposition is studied.

2.4 CHOLESKY DECOMPOSITION

The Cholesky decomposition factors a positive symmetric matrix to the product of a lower triangular matrix and its transpose, of which two general ways are given by:

$$\boldsymbol{A = U^TU} \qquad \textbf{(2)}$$

$$\boldsymbol{A = LDL^T} \qquad \textbf{(3)}$$

Where A is a positive definite symmetric matrix. U is the upper triangular matrix & UT is Transpose of upper triangular matrix, L is the unit lower triangular matrix & LT is lower triangular matrix and D is the diagonal matrix in which all elements are zeros except the diagonal elements. Cholesky decomposition has 1/3N3 FLOPS complexity with heavy inner data dependency. Introducing a diagonal matrix as shown in Eq. (3) in Cholesky decomposition has many advantages, such as avoiding square roots and alleviating data dependency [5]. In this paper, we design and implement the UTU Cholesky decomposition in Eq. (2) on FPGAs for computation acceleration. Using the fact that A is symmetric [6]:

$$A= LDL^T$$

Where L is lower triangular matrix.

$$\mathbf{L} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & & 0 \\ \vdots & & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \quad \textbf{(4)}$$

With Cholesky decompositions, the elements of L are evaluated as follows:

$$l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} l^2_{kj}} \quad \textbf{(5)}$$

Where k=1, 2 ….n.

$$l_{ki} = \frac{1}{l_{ii}}\left(a_{ki} - \sum_{j}^{i-1} l_{ij}\, l_{kj}\right) \quad \textbf{(6)}$$

Where i=1, 2…..k-1.

First subscript is row index and second one is column index. Cholesky decomposition is evaluated column by column and in each row the elements are evaluated from top to bottom. That is, in each column the diagonal element is evaluated first using equation (4) (the elements above the diagonal are zero) and then the other elements in the same row are evaluated next using equation (5). This is carried out for each column starting from the first one. Note that if the value within the square root in equation (4) is negative, Cholesky decomposition will fail. However, this will not happen for positive semi definite matrices, which are encountered commonly in many engineering systems (e.g., circuits, covariance matrix). Thus, Cholesky decomposition is a good way to test for the positive semi definiteness of symmetric matrices. A general form of the Cholesky algorithm is given in Algorithm 2, where the row-order representation is assumed for stored matrices. In the above algorithm, if one changes the order of the three *for* loops, one can get different variations which give different behavior in terms of memory access patterns and the basic linear Algebraic operation performed in the innermost loop. Out of the six different variations possible, only three are of interest. They are the row-oriented, column-oriented and sub matrixes forms and are described as: In **Row-oriented** L is calculated row by row, with the terms for each row being calculated using terms on the preceding rows that have already been evaluated, **Column-oriented,** the inner loop computes a matrix-vector product. Here, each column is calculated using terms from previously computed columns. And **Sub-matrix t**he inner loops apply the current column as a rank-1 update to the partially reduced sub-matrix.The algorithm is analyzed as the decomposition by transferring the matrix A into memory elements. The diagonal entire of lower triangular matrix, A, are square root of diagonal entries of given matrix (8). It calculate the entries by dividing the corresponding element of given matrix by the belonging column diagonal element (9) .The algorithm works column by column and after the computation of first column of diagonal matrix with the given matrix entries, the elements in the next column are updated (4).

```
1. For p = 1to n
2.        For q=1 to j-1
3.                For i = p to n
4.        A[i,p] = A[i,q] – A[i,q] * A[p,q]
5                 End
6         End
7         A[p,p] = √ A[p,p]
8         For q = p+1 to n
9                 A[q,p]= A[q,p] / A[p,p]
10        End
11   End
```

Algorithm 2: General form of the Cholesky algorithm.

In this paper, the Cholesky factorization method has several advantages over the other decomposition techniques. It has faster execution because only one factor needs to be calculated as the other factor is simply the transpose of the first one. Thus, the number of operations for dense matrices is approximately n3/3 under Cholesky. It is highly adaptive to parallel architectures hence pivoting is not required. This makes the Cholesky algorithm especially suitable for parallel architectures as it reduces inter-processor communications. A lot of work [8] has been devoted to parallelizing the Cholesky factorization method. it has significantly lower memory requirements. In the Cholesky algorithm, only the lower triangular matrix is used for factorization, and the intermediate and final results (the Cholesky factor L) are overwritten in the original matrix. Thus, only the lower triangular matrix must be stored, resulting in significant savings in terms of memory requirements. it eliminates the divider operation dependency. It improves the data throughput.

III. CONCLUSIONS

The proposed paper review all the matrix decomposition techniques used in signal processing applications. QR decomposition has complicated algorithm and also a slower algorithm. LU decomposition doesn’t find approximate solution and have stability issues. Cholesky decomposition can find approximate solution of the problem. Among all the matrix decomposition techniques Cholesky

Decomposition is the well suited for signal processing application. It is efficient in terms of memory storage capacity, computational cost, speed, data alleviating and throughput.

REFERENCES

[1] http://en.wikipedia.org/wiki/Signal_processing

[2] E. W. Weisstein. Matrix decomposition. MathWorld–A Wolfram Web Resource. [Online]. Available:

a. http://mathworld.wolfram.com/MatrixDecomposition.html

[3] Steven Chapra and Raymond Canale, Numerical methods for Engineers, Fourth Edition,Mcgraw Hill, 2002.

[4] Mathworks, "Users Guide Filter Design Toolbox", March-2007.

[5] R.Barett,templates for the solution of linear systems: building blocks for iterative method, second ed.SIAM,1994.

[6] DepengYang, G.D.Peterson, H.Li and junquing Sun," *An FPGA implementation for solving least square problem*" IEEE Symposium on Field-Programmable Custom Computing Machines, pp.no.306-309, 2009.

[7] X. Wang and S.G. Ziavras, "Parallel LU Factorization of Sparse Matrices on FPGA-Based Configurable Computing Engines," Concurrency and Computation: Practice and Experience, 16.4:319-343, April 2004.

[8] Guiming Wu,Yong Dou, *" A High Performance and Memory Efficient LU decomposer on FPGAs*" IEEE Transactions on Computers, Vol. 61, No. 3, pp. no. 366-378, 2012.