Some Important Matrix Factorizations

Signal Processing Graduate Seminar III

Lucio Azzari

lucio.azzari@tut.fi

Tampere University of Technology

November 28, 2012

Introduction

- Factorization is fundamental in order to simplify and speed up matrices operations.

Example

- Let us consider the general system of linear equations in the matrix form:

$$A\mathbf{x} = \mathbf{b}. \tag{1}$$

- Solving (1) for $\mathbf{x}$ with the *canonical procedure* (compute A^{-1} and multiply by $\mathbf{b}$) requires $O(m^3)$ operations.
- If A does not have any specific structure (e.g., Toeplitz, Vandermonde, etc.) we do not have special algorithm for a faster computation ($O(m^2)$ operations).
- It reduces the amount of operations needed for the computation of $\mathbf{x}$: it can be found without the explicit inversion of A^{-1}.

- Use of factorization permits to decompose A in such a way that simplifies the further operations.

Outline

The LU Factorization

- When A is a square $m \times m$ matrix, it is possible to compute its LU factorization as
$$PA = LU, \tag{2}$$

where:

$$L = \left[\begin{array}{ccccc} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ \vdots & & & & \\ l_{m1} & l_{m2} & l_{m3} & \cdots & 1 \end{array} \right], \quad U = \left[\begin{array}{ccccc} u_{11} & u_{13} & u_{13} & \cdots & u_{1m} \\ 0 & u_{22} & u_{23} & \cdots & u_{2m} \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & u_{mm} \end{array} \right].$$

- L is a lower triangular matrix with ones on the diagonal.
- U is a upper triangular matrix.
- P is a permutation matrix, here used for *pivoting*.

Computing the LU factorization

- Gaussian elimination method in order to obtain an upper triangular matrix.
- Considering A row-wise,

$$A = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix},$$

  we modify the matrix using *row operations*.

Row operation

Replacement of a row of a matrix with a linear combination of other rows:

$$\mathbf{a}_i^T \leftarrow \mathbf{a}_i^T - \alpha \mathbf{a}_j^T,$$

with α (multiplier) being a scalar.

- In order to ensure that L is lower triangular $\alpha = a_{ij}/a_{jj}$, with a_{ij} being the elements that we want to zero.

Computing the LU factorization-U computation

Using successively row operations leads to compute the matrix U.

Procedure:

1. Let us consider a 3×3 matrix

$$A = \begin{bmatrix} 2 & 4 & -5 \\ 6 & 8 & 1 \\ 4 & -8 & -3 \end{bmatrix}.$$

2. We must zero: $6, 4, -8$.
3. $\mathbf{a}_2^T \leftarrow \mathbf{a}_2^T - 3\mathbf{a}_1^T$:

$$\begin{bmatrix} 2 & 4 & -5 \\ 0 & -4 & 16 \\ 4 & -8 & -3 \end{bmatrix}.$$

4. $\mathbf{a}_3^T \leftarrow \mathbf{a}_3^T - 2\mathbf{a}_1^T$:

$$\begin{bmatrix} 2 & 4 & -5 \\ 0 & -4 & 16 \\ 0 & -16 & 7 \end{bmatrix}.$$

5. $\mathbf{a}_3^T \leftarrow \mathbf{a}_3^T - 4\mathbf{a}_2^T$:

$$U = \begin{bmatrix} 2 & 4 & -5 \\ 0 & -4 & 16 \\ 0 & 0 & -57 \end{bmatrix}.$$

The diagonal elements of U are referred as pivots.

Computing the LU factorization-L computation

L can be found expressing the elimination steps in terms of matrix multiplications.

1. The first elimination:

$$\begin{bmatrix} 2 & 4 & -5 \\ 0 & -4 & 16 \\ 4 & -8 & -3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -5 \\ 6 & 8 & 1 \\ 4 & -8 & -3 \end{bmatrix} = E_1 A,$$

   where E_1 is denoted as *elementary matrix*.
2. E_1: identity matrix except for a off-diagonal element (-3).
3. The coordinate (2,1) of the off-diagonal element indicates position of the eliminated element, the row in which it belongs (row 2) and even the row used for the elimination (in this case row 1).

Computing the LU factorization-L computation

4. The second elimination:

$$\begin{bmatrix} 2 & 4 & -5 \\ 0 & -4 & 16 \\ 0 & -16 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -5 \\ 0 & -4 & 16 \\ 4 & -8 & -3 \end{bmatrix} = E_2 E_1 A.$$

5. The third elimination:

$$U = \begin{bmatrix} 2 & 4 & -5 \\ 0 & -4 & 16 \\ 0 & 0 & -57 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -5 \\ 0 & -4 & 16 \\ 0 & -16 & 7 \end{bmatrix} = E_3 E_2 E_1 A.$$

6. We finally have:

$$A = E_1^{-1} E_2^{-1} E_3^{-1} U = LU,$$

with

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 4 & 1 \end{bmatrix}.$$

Computing the LU factorization-Discussions

- L is composed by the α coefficients, thus, there is no necessity to compute it directly.
- Including the computation of U, the factorization can be computed in $O(2/3m^3)$ operations.
- The algorithm described so far is correct, but could lead to numerically poor solutions.
- The coefficients α, resulting by divisions, could be very large numbers, and there is even the possibility of a division by 0.
- We want to choose the more appropriate solution: use of the matrix P.

Pivoting

Solution

Pivoting: permuting the rows of the matrix so that the pivot is the largest (in absolute value) element in the unreduced part of the k-th column.

1. Let us consider the 3×3 matrix

$$A = \begin{bmatrix} 2 & 4 & -5 \\ 6 & 8 & 1 \\ 4 & -8 & -3 \end{bmatrix}.$$

2. We must change the second row and put it to the first:

$$P_{12}A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -5 \\ 6 & 8 & 1 \\ 4 & -8 & -3 \end{bmatrix}.$$

Pivoting

3. The new matrix is:
$$\begin{bmatrix} 6 & 8 & 1 \\ 2 & 4 & -5 \\ 4 & -8 & -3 \end{bmatrix},$$
and the first 2 eliminations are:
$$\mathbf{a}_2^T \leftarrow \mathbf{a}_2^T - \frac{1}{3}\mathbf{a}_1^T,$$
$$\mathbf{a}_3^T \leftarrow \mathbf{a}_3^T - \frac{2}{3}\mathbf{a}_1^T.$$

4. That is
$$E_2 E_1 P_{12} A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{2}{3} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} P_{12} A = \begin{bmatrix} 6 & 8 & 1 \\ 0 & \frac{4}{3} & -\frac{16}{3} \\ 0 & -\frac{40}{3} & -\frac{11}{3} \end{bmatrix}.$$

Pivoting

5. Now we have to minimize the element on the second column. Thus, we have to change the third row (largest value on the second column) with the second row (pivot position), using the matrix P_{23}.
6. We can now zero the last coefficient (generating the third elementary matrix E_3).
7. We have a solution of the form

$$E_3 P_{23} E_2 E_1 P_{12} A = U.$$

8. Solving for A:

$$A = P_{12}^{-1} {E_1}^{-1} {E_2}^{-1} {P_{23}}^{-1} {E_3}^{-1} U = VU.$$

9. V is not lower triangular, but can we can define the matrix $L = (P_{23} P_{12}) V$, that is lower triangular.
10. Finally we can define $P = P_{23} P_{12}$, and write the relation $PA = LU$.

Solving the system

Reminder

LU factorization:

$$PA = LU.$$

System of equations:

$$A\mathbf{x} = \mathbf{b}.$$

- Since the permutation matrices are orthogonal:

$$A = P^T LU.$$

- The system (1) can be written as:

$$LU\mathbf{x} = P\mathbf{b}.$$

- Finally:

$$L\mathbf{y} = \mathbf{c}, \tag{3}$$

with $U\mathbf{x} = \mathbf{y}$, and $P\mathbf{b} = \mathbf{c}$.

Solving the system-*Forward substitution*

- Exploding the matrices we have

$$\begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ \vdots & & & & \\ l_{m1} & l_{m2} & l_{m3} & \cdots & l_{mm} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \mathbf{c},$$

  with $l_{ii} = 1,\ i = 1, \ldots, m$.

- The first row can easily be solved:

$$y_1 = \frac{c_1}{l_{11}} = c_1.$$

- Knowing $y_1 = c_1$ we can compute y_2:

$$l_{21}y_1 + l_{22}y_2 = c_2 \Rightarrow y_2 = c_2 - l_{21}y_1.$$

- In general:

$$y_j = \left(c_j - \sum_{i=1}^{j-1} l_{ji} y_i \right).$$

Solving the system-*Back substitution*

- Now we can solve $U\mathbf{x} = \mathbf{y}$.

$$\begin{bmatrix} u_{11} & u_{13} & u_{13} & \cdots & u_{1m} \\ 0 & u_{22} & u_{23} & \cdots & u_{2m} \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & u_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \mathbf{y}.$$

- The last row can easily be solved:

$$x_m = \frac{1}{u_{mm}} y_m.$$

- Analogously to the forward substitution:

$$x_{m-1} = \frac{1}{u_{(m-1)(m-1)}} \left(y_{m-1} - x_m u_{(m-1)m} \right).$$

- In general:

$$x_j = \frac{1}{u_{jj}} \left(y_j - \sum_{i=j+1}^{m} u_{jk} x_k \right).$$

Solving the system-Discussions

- For the forward and back substitutions are required approximately $O(m^2/2)$ operations each ($O(m^2)$ in total).
- The back substitution requires that the elements on the diagonal of U are non-zero.
- Even considering different systems of equations

$$\begin{aligned} A\mathbf{x}_1 &= \mathbf{b}_1 \\ A\mathbf{x}_2 &= \mathbf{b}_2 \end{aligned},$$

  the use of LU is always more efficient (it is the same as computing $A^{-1}\mathbf{b}_2$ knowing A^{-1}).
- Never explicitly invert a matrix numerically!

Computing the determinant using LU factorization

- If we want to compute $\det(A)$:

$$\det(A) = \det(P^T LU) = \det(P)\det(L)\det(U) = \pm \prod_{i=1}^{m} u_{ii}.$$

- By the fact that P is a permutation matrix, we have

$$\det(P^T) = \det(P) = (-1)^S,$$

where S is the number of permutations (number of P_{ij} matrices).

The Cholesky factorization

- Let us consider a positive-definite $m \times m$ Hermitian matrix B. It is possible to compute its Cholesky factorization as

$$B = LL^H, \tag{4}$$

  where L is a lower triangular matrix, and L^H is its Hermitian $(L^H = \left(L^T\right)^*)$.
- L is the *"square root"* of the matrix B.
- Special case of LU factorization, where $U = L^H$.

Positive definite matrix

The matrix B is called positive definite if

$$\mathbf{x}^T B \mathbf{x} > 0, \quad \forall \mathbf{x} \neq \mathbf{0} \ .$$

Equivalently: all the principal minors have positive determinant / all the eigenvalues are positive.

Computing the Cholesky factorization

1. Let us denote B as

$$B = \begin{bmatrix} \alpha & \mathbf{v}^H \\ \mathbf{v} & B_1 \end{bmatrix} = LL^H = \begin{bmatrix} l_{11} & \mathbf{0}^\mathbf{T} \\ \mathbf{l}_{21} & L_{22} \end{bmatrix} \begin{bmatrix} l_{11} & \mathbf{l}_{21}^T \\ \mathbf{0} & L_{22}^T \end{bmatrix}.$$

2. It is trivial

$$l_{11} = \sqrt{\alpha}, \quad \mathbf{l}_{21} = \frac{1}{l_{11}}\mathbf{v}.$$

3. We also have

$$B_1 = \mathbf{l}_{21}\mathbf{l}_{21}^T + L_{22}L_{22}^T \Rightarrow B_1 - \mathbf{l}_{21}\mathbf{l}_{21}^T = L_{22}L_{22}^T,$$

   that is a Cholesky factorization of order $m-1$.
4. We can apply the same algorithm in order to find the first row and first column of $L_{22}L_{22}^T$, and we will find a Cholesky factorization of order $m-2$.
5. Iterating the process we can find the solution.
6. Requires $O(m^3/3)$ operations.

Generating Gaussian random vectors with covariance R

- Let us suppose that we have a Gaussian random number generator $\mathcal{N}(0,1)$.
- We can generate random vectors of the type $\mathcal{N}(0,1)$ with covariance R using the Cholesky factorization.
- Factorize $R = LL^H$.
- Generate a vector $\mathbf{x}$ with a set of $\mathcal{N}(0,1)$ independent random variables.
- Let $\mathbf{z} = L\mathbf{x}$.
- Since $E\left[\mathbf{x}\mathbf{x}^T\right] = I$,

$$E\left[\mathbf{z}\mathbf{z}^T\right] = LE\left[\mathbf{x}\mathbf{x}^T\right]L^T = LL^T = R,$$

thus $\mathbf{z}$ has the desired covariance R.

Solving a system of equations

- Let us consider a system
$$A\mathbf{x} = \mathbf{b},$$
where A is Hermitian positive definite.
- We can write $A = LL^H$.
- We can solve for the two systems
$$\begin{aligned} L\mathbf{y} &= \mathbf{b}, \\ L^H\mathbf{x} &= \mathbf{y}, \end{aligned}$$
like for the LU factorization (but with less operations for L).

Least-square solution

Matrix form of the least square problem

When the basis vector are finite dimensional, the solution $\mathbf{x}$ is the minimizer of

$$\|\mathbf{e}\|^2 = \|\mathbf{b} - A\mathbf{x}\|^2, \tag{5}$$

where $\mathbf{b}$ is the ground truth, and A is the coefficients matrix.
The matrix form of the approximated solution to (5) is

$$\hat{\mathbf{x}} = \left(A^H A\right)^{-1} A^H \mathbf{b},$$

or equivalently

$$A^H A \hat{\mathbf{x}} = A^H \mathbf{b}. \tag{6}$$

- Denoting $A^H A = LL^H$ and $A^H \mathbf{b} = \mathbf{p}$ we can solve (6) using Cholesky factorization:

$$\begin{aligned} L\mathbf{y} = \mathbf{p} \rightarrow &\quad \mathrm{forward-substitution}, \\ L^H\hat{\mathbf{x}} = \mathbf{y} \rightarrow &\quad \mathrm{back-substitution}. \end{aligned}$$

QR Factorization

- Considering an $m \times n$ matrix A, with $m > n$, the QR factorization is defined as
$$A = QR. \tag{7}$$
- Q is an unitary $m \times m$ matrix, and R is an $m \times n$ upper triangular matrix of the form
$$R = \begin{bmatrix} R_1 \\ \mathbf{0} \end{bmatrix},$$
where R_1 is an $n \times n$ upper matrix, and $\mathbf{0}$ an $(m-n) \times n$ matrix of zeros.

Definition

An $m \times m$ matrix Q is said unitary if $QQ^H = I$.
If Q has only real elements, and $QQ^T = I$, Q said orthogonal.

Lemma

If $\mathbf{y} = Q\mathbf{x}$, with Q being an $m \times m$ matrix, then $\|\mathbf{y}\| = \|\mathbf{x}\|$ if and only if Q is unitary.

Computing the QR factorization

We investigate 3 different methods for the QR factorization:

1. GramSchmidt method: exploit the GramSchmidt orthonormalization.
2. Householder method: exploit the vector reflection via Householder transformation.
3. Givens rotation method: exploit the Givens rotation matrix.

Computing the QR factorization-Gram-Schmidt method

GramSchmidt method

It is used to orthonormalize a set of linerly independent vectors $S = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$.

Let us denote the orthonormalized set as $S' = \{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$, then the Gram-Schmidt method consists in

$$
\begin{aligned}
\mathbf{u}_1 &= \mathbf{v}_1, & \mathbf{e}_1 &= \tfrac{\mathbf{u}_1}{\|\mathbf{u}_1\|}, \\
\mathbf{u}_2 &= \mathbf{v}_2 - \langle \mathbf{v}_2, \mathbf{u}_1 \rangle \mathbf{e}_1, & \mathbf{e}_2 &= \tfrac{\mathbf{u}_2}{\|\mathbf{u}_2\|}, \\
&\vdots & &\vdots \\
\mathbf{u}_k &= \mathbf{v}_k - \sum_{j=1}^{k-1} \langle \mathbf{v}_k, \mathbf{u}_j \rangle \mathbf{e}_j, & \mathbf{e}_k &= \tfrac{\mathbf{u}_k}{\|\mathbf{u}_k\|}.
\end{aligned}
$$

Computing the QR factorization-Gram-Schmidt method

1. Let us consider the $m \times n$ full column rank matrix A with its set of column vectors $S = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$.
2. Using Gram-Schmidt we can find the orthonormal set of A $S' = \{\mathbf{e}_1, \dots, \mathbf{e}_n\}$.
3. With S' we can express the elements of S as

$$\mathbf{a}_k = \sum_{j=1}^{k} \langle \mathbf{a}_k, \mathbf{e}_j \rangle \, \mathbf{e}_j, \quad \mathbf{e}_j = \frac{\mathbf{u}_j}{\|\mathbf{u}_j\|} \ ,$$

   that expressed in the matrix form is

$$A = [\mathbf{a}_1 | \dots | \mathbf{a}_n] = [\mathbf{e}_1 | \dots | \mathbf{e}_n] \begin{bmatrix} \mathbf{a}_1 \cdot \mathbf{e}_1 & \mathbf{a}_2 \cdot \mathbf{e}_1 & \cdots & \mathbf{a}_n \cdot \mathbf{e}_1 \\ 0 & \mathbf{a}_2 \cdot \mathbf{e}_2 & \cdots & \mathbf{a}_n \cdot \mathbf{e}_2 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{a}_n \cdot \mathbf{e}_n \end{bmatrix} .$$

4. In this case Q is an $m \times n$ unitary matrix, while R is an $n \times n$ upper triangular matrix.

Computing the QR factorization-Householder transformation

- Permits to factorize a matrix in $O(m^3)$ operations top.
- Let us consider a column vector $\mathbf{x}$.
- It is possible to compute the reflection of $\mathbf{x}$ across the space perpendicular to a vector $\mathbf{v}$

$$H_v \mathbf{x} = \mathbf{x}_r, \quad H_v = I - 2\frac{\mathbf{v}\mathbf{v}^H}{\mathbf{v}^H\mathbf{v}}.$$

- If we project twice, we have again the point $\mathbf{x}$, thus H_v is unitary.
- If we want to zero all the elements of $\mathbf{x}$ but the first

$$\left(I - 2\frac{\mathbf{v}\mathbf{v}^H}{\mathbf{v}^H\mathbf{v}}\right)\mathbf{x} = \alpha \mathbf{e}_1,$$

  with $\mathbf{e}_1 = [1, 0, ..., 0]^T$, and $\alpha = \pm\|\mathbf{x}\|_2$ (H_v is unitary).
- We obtain $\mathbf{v} = \mathbf{x} \pm \|\mathbf{x}\|_2 \mathbf{e}_1$.

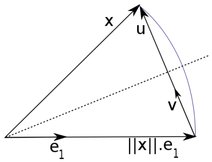

Figure: Graphical representation of the Houselord transformation. $\mathbf{x}$ is reflected respect to the line orthogonal to the vector $\mathbf{v}$, in order to be collinear to $\mathbf{e_1}$.

Computing the QR factorization-Householder transformation

1. We convert A to an upper triangular form.
2. Using the Householder transformation we can zero all the elements of the first column of A

$$Q_1 A = \begin{bmatrix} \alpha_1 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix},$$

   with $\|Q_1 \mathbf{a}_1\| = \|\alpha_1\|$.
3. Now we can zero the last two elements of the second column

$$\begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & H_2 \end{bmatrix} \begin{bmatrix} \alpha_1 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix} = Q_2 Q_1 A = \begin{bmatrix} \alpha_1 & \times & \times \\ 0 & \alpha_2 & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix}.$$

Computing the QR factorization-Householder transformation

4. Finally,

$$\begin{bmatrix} 1 & 0 & \mathbf{0}^T \\ 0 & 1 & 0 \\ \mathbf{0} & 0 & H_3 \end{bmatrix} \begin{bmatrix} \alpha_1 & \times & \times \\ 0 & \alpha_2 & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix} = Q_3 Q_2 Q_1 A = \begin{bmatrix} \alpha_1 & \times & \times \\ 0 & \alpha_2 & \times \\ 0 & 0 & \alpha_3 \\ 0 & 0 & 0 \end{bmatrix} = R,$$

   and

$$Q_3 Q_2 Q_1 = Q^H.$$

5. Since Q_1, Q_2, and Q_3 are hermitian, $Q = Q_1 Q_2 Q_3$.
6. For a general $m \times n$ matrix

$$Q = Q_1 Q_2 \ldots Q_n, \quad Q_j = I - 2\frac{\tilde{\mathbf{v}}_j \tilde{\mathbf{v}}_j^H}{\tilde{\mathbf{v}}_j^H \tilde{\mathbf{v}}_j},$$

   where

$$\tilde{\mathbf{v}}_j = [\underbrace{0, 0, \ldots 0,}_{j-1} \ \mathbf{v}_j^T]^T, \quad \mathbf{v}_j = \mathbf{x}_j + \operatorname{sign}(\mathbf{x}_j(1)) \|\mathbf{x}_j\|_2 \mathbf{e1} \quad .$$

Computing the QR factorization-Givens rotation

Matrix rotation

- A two-dimensional point ($\mathbf{x} = [x, y]^T$) rotation by an angle θ can be obtained by matrix multiplication

$$\mathbf{x}' = \; G_\theta \mathbf{x}, \quad G_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} .$$

- The matrix G_θ is orthogonal

$$G_\theta^T G_\theta = I,$$

where G_θ^T is the rotation of the point by the angle θ in the oposite direction.

- Considering a generic point $\mathbf{x}$, it can be rotated on the x-axis in such a way that the second coordinate is zero

$$\theta = -\tan^{-1}\left(\frac{y}{x}\right).$$

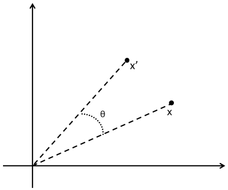

Figure: Example of point rotation form $\mathbf{x}$ to $\mathbf{x}'$.

Computing the QR factorization-Givens rotation

1. Considering an $m \times n$ matrix A, the Givens rotation method exploits matrix rotation in order to zero the elements of A, and obtain an upper matrix R.
2. Similar to the Houselord method, but instead of zeroing the elements of one column in a single step, it zeros one single element per time.
3. Used when we have to zero just few elements.
4. It starts from the bottom of the first column, and works up the columns.
5. Let us consider the case of zeroing the element a_{ik} of A.
6. We choose one element a_{jk} on the same column (usually $j = i - 1$), and we set
$$x = a_{jk},$$
$$y = a_{ik},$$
in order to zero the y component.

Computing the QR factorization-Givens rotation

7. The Givens matrix $G_\theta(i,k,j)$ can be easily found

$$G_\theta(i,k,j) = \begin{array}{cc} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} & \begin{array}{c} \\ \\ j \\ \\ i \\ \\ \\ \end{array} \\ \begin{array}{ccccccc} & & j & & i & & \end{array} & \end{array} ,$$

where $c = \cos\theta$, and $s = \sin\theta$.

8. The new matrix $A' = G_\theta(i,k,j)A$ has a zero in the position (i,k).
9. Iterating the process we can zero all the elements of A

$$G_{\theta_N} G_{\theta_{N-1}} \ldots G_{\theta_1} A = R, \tag{8}$$

with

$$Q = (G_{\theta_N} G_{\theta_{N-1}} \ldots G_{\theta_1})^T = G_{\theta_1}^T G_{\theta_2}^T \ldots G_{\theta_N}^T. \tag{9}$$

Full-rank least-square problem

- Let us consider the system

$$A\mathbf{x} \approx \mathbf{b}, \tag{10}$$

  where A is an $m \times n$ full-column rank matrix.
- The solution $\hat{\mathbf{x}}$ that minimizes $\|A\mathbf{x} - \mathbf{b}\|_2$ is

$$\hat{\mathbf{x}} = \left(A^H A\right)^{-1} A^H \mathbf{b}.$$

- We can factorize A as

$$A = QR = Q \begin{bmatrix} R_1 \\ \mathbf{0} \end{bmatrix},$$

  where R_1 is an $n \times n$ matrix, and $\mathbf{0}$ is $(m-n) \times n$ matrix of zeros.
- Let us denote

$$Q^H \mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix},$$

  where $\mathbf{c}$ and $\mathbf{d}$ are, respectively, a $n \times 1$ and a $(m-n) \times 1$ vectors.

Full-rank least-square problem

- We have to minimize

$$\begin{aligned} \|A\mathbf{x}-\mathbf{b}\|_2^2 = \|QR\mathbf{x}-\mathbf{b}\|_2^2 = \left\|Q(R\mathbf{x}-Q^H\mathbf{b})\right\|_2^2 = \\ \left\|\begin{bmatrix} R_1 \\ \mathbf{0} \end{bmatrix}\mathbf{x} - \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}\right\|_2^2 = \|R_1\mathbf{x}-\mathbf{c}\|_2^2 + \|\mathbf{d}\|_2^2 . \end{aligned} \tag{11}$$

- Minimizing (11) is thus reduced to the system

$$R_1\hat{\mathbf{x}} = \mathbf{c},$$

that can be solved easily by *back-substitution*, since R_1 is upper triangular.

Least-square problem using Givens rotation

- We can express the system $A\mathbf{x} \approx \mathbf{b}$ as

$$[A|\mathbf{b}] \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = B\mathbf{h} \approx 0.$$

- Let use the Givens rotation in such a way to obtain

$$\begin{bmatrix} \hat{a}_{11} & \hat{a}_{12} & \hat{a}_{13} & \cdots & \hat{a}_{1n} & \hat{b}_1 \\ & \hat{a}_{22} & \hat{a}_{23} & \cdots & \hat{a}_{2n} & \hat{b}_2 \\ & & \hat{a}_{33} & \cdots & \hat{a}_{3n} & \hat{b}_3 \\ \vdots & & & & \vdots & \\ & & & & \hat{a}_{nn} & \hat{b}_n \\ \times & \times & \times & \times & \times & \hat{b}_{n+1} \\ \times & \times & \times & \times & \times & \hat{b}_{n+2} \\ \vdots & & & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \\ -1 \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \end{bmatrix},$$

by multiplication $(Q_p \ldots Q_2 Q_1) B$.

Least-square problem using Givens rotation

- We do not need to zeros the rows from $n+1$ to m.
- By *back-substitution* we can find

$$x_n = \frac{\hat{b}_n}{\hat{a}_{nn}},$$
$$x_{n-1} = \frac{\hat{b}_{n-1} - \hat{a}_{(n-1)n} x_n}{\hat{a}_{(n-1)(n-1)}},$$
$$\vdots$$
$$x_i = \frac{\hat{b}_i - \sum_{j=i+1}^{n} a_{ij} x_j}{\hat{a}_{ii}},$$
$$\vdots$$
$$x_1 = \frac{\hat{b}_1 - \sum_{j=2}^{n} a_{1j} x_j}{\hat{a}_{11}}.$$

Summary

- We studied 3 factorization methods:
  - the LU factorization, that factorizes a general square matrix A in two matrices, a lower triangular L and an upper triangular U;
  - the Cholesky factorization, that factorizes a square, positive-definite Hermitian matrix A in two matrices L and L^H;
  - the QR factorization, that factorizes a rectangular matrix A in two matrices, an unitary matrix Q and an upper triangular matrix R.
- For each method, we investigated how to compute the factorization.
- Furthermore, practical applications have been presented.