Rank Revealing QR Factorizations

Zane Perry, Luke Sellmayer, Maedée Trank-Greene

December 19, 2023

Table of Contents

Project Summary

- Rank-revealing factorizations (RRFs) have emerged as a powerful tool across various fields, from machine learning to reduced order modeling

Project Summary

- Rank-revealing factorizations (RRFs) have emerged as a powerful tool across various fields, from machine learning to reduced order modeling
- RRFs are used to compute the numerical rank of a matrix to build a low-rank approximation

Project Summary

- Rank-revealing factorizations (RRFs) have emerged as a powerful tool across various fields, from machine learning to reduced order modeling
- RRFs are used to compute the numerical rank of a matrix to build a low-rank approximation
- Low-rank approximations are useful as they keep the information that is necessary while saving space

Project Summary

- Rank-revealing factorizations (RRFs) have emerged as a powerful tool across various fields, from machine learning to reduced order modeling
- RRFs are used to compute the numerical rank of a matrix to build a low-rank approximation
- Low-rank approximations are useful as they keep the information that is necessary while saving space
- Traditional rank-revealing methods such as SVD are computationally expensive for large matrices

Project Summary

- Rank-revealing factorizations (RRFs) have emerged as a powerful tool across various fields, from machine learning to reduced order modeling
- RRFs are used to compute the numerical rank of a matrix to build a low-rank approximation
- Low-rank approximations are useful as they keep the information that is necessary while saving space
- Traditional rank-revealing methods such as SVD are computationally expensive for large matrices
- We are motivated to implement the RRQRF since it should be a computationally less expensive method

Project Summary

- Rank-revealing factorizations (RRFs) have emerged as a powerful tool across various fields, from machine learning to reduced order modeling
- RRFs are used to compute the numerical rank of a matrix to build a low-rank approximation
- Low-rank approximations are useful as they keep the information that is necessary while saving space
- Traditional rank-revealing methods such as SVD are computationally expensive for large matrices
- We are motivated to implement the RRQRF since it should be a computationally less expensive method

Table of Contents

Background Information

Let $A \in \mathbb{R}^{m\times n}$, $r \in \mathbb{N}$, and $k \in \mathbb{N}$

Definition: Rank

The $rank(A) = r$ of a matrix A is the number of linearly independent rows or columns of A.

Background Information

Let $A \in \mathbb{R}^{m \times n}$, $r \in \mathbb{N}$, and $k \in \mathbb{N}$

Definition: Rank

The $rank(A) = r$ of a matrix A is the number of linearly independent rows or columns of A.

Definition: Numerical Rank

The $num\_rank(A) = k$ of a matrix A is the number of singular values, σ_i for $i = 1, 2, \ldots, n$, that are greater than a given $\epsilon > 0$.

Background Information

Let $A \in \mathbb{R}^{m \times n}$, $r \in \mathbb{N}$, and $k \in \mathbb{N}$

Definition: Rank

The $rank(A) = r$ of a matrix A is the number of linearly independent rows or columns of A.

Definition: Numerical Rank

The $num\_rank(A) = k$ of a matrix A is the number of singular values, σ_i for $i = 1, 2, \ldots, n$, that are greater than a given $\epsilon > 0$.

Definition: Numerically Rank Deficient

A matrix is numerically rank deficient if there are $k < n$ singular values of A that are larger than $\epsilon > 0$, or equivalently $n - k$ singular values that are smaller than $\epsilon > 0$.

What is a Rank-Revealing Factorization (RRF)?

- RRFs are a common method used to compute the **numerical rank** of a matrix A to build a low-rank approximation of A

What is a Rank-Revealing Factorization (RRF)?

- RRFs are a common method used to compute the **numerical rank** of a matrix A to build a low-rank approximation of A
- Several common RRFs used in practice
  1. SVD
  2. QR Factorization

What is a Rank-Revealing Factorization (RRF)?

- RRFs are a common method used to compute the **numerical rank** of a matrix A to build a low-rank approximation of A
- Several common RRFs used in practice
  1. SVD
  2. QR Factorization
- The RRF of a matrix $A \in \mathbb{R}^{m \times n}$ can be defined as

Definition: RRF

$A = XDY^T, X \in \mathbb{R}^{m \times p}, D \in \mathbb{R}^{p \times p}, Y \in \mathbb{R}^{n \times p}$ where

1. $p < min\{m, n\}$
2. D is diagonal and nonsingular
3. X and Y are well conditioned

QR Factorization as an RRF

Definition: RRF

$A = XDY^T, X \in \mathbb{R}^{m \times p}, D \in \mathbb{R}^{p \times p}, Y \in \mathbb{R}^{n \times p}$

- The QR factorization of a matrix A is

$$A = QR, Q \in \mathbb{R}^{m \times m}, R \in \mathbb{R}^{m \times n} \tag{1}$$

QR Factorization as an RRF

Definition: RRF

$A = XDY^T, X \in \mathbb{R}^{m\times p}, D \in \mathbb{R}^{p\times p}, Y \in \mathbb{R}^{n\times p}$

- The QR factorization of a matrix A is

$$A = QR, Q \in \mathbb{R}^{m\times m}, R \in \mathbb{R}^{m\times n} \tag{1}$$

- QR as an RRF

$$X = Q, D = diag(R), Y = (D^{-1}R)^T \tag{2}$$

QR Factorization as an RRF

Definition: RRF

$A = XDY^T, X \in \mathbb{R}^{m \times p}, D \in \mathbb{R}^{p \times p}, Y \in \mathbb{R}^{n \times p}$

- The QR factorization of a matrix A is

$$A = QR, Q \in \mathbb{R}^{m \times m}, R \in \mathbb{R}^{m \times n} \tag{1}$$

- QR as an RRF

$$X = Q, D = diag(R), Y = (D^{-1}R)^T \tag{2}$$

- This basic definition of the RRF with the QR factorization can lead to potential problems

Numerical Example

- Consider the following matrix and its QR factorization

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Numerical Example

- Consider the following matrix and its QR factorization

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; A = QR = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Numerical Example

- Consider the following matrix and its QR factorization

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; A = QR = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- What is the RRF of this matrix?

Numerical Example

- Consider the following matrix and its QR factorization

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A = QR = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- What is the RRF of this matrix?

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Numerical Example

- Consider the following matrix and its QR factorization

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A = QR = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- What is the RRF of this matrix?

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- What about Y?

Numerical Example

- Consider the following matrix and its QR factorization

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A = QR = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- What is the RRF of this matrix?

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- What about Y?
  - D is a non-invertible matrix, so we cannot find Y

QR Factorization with Column Pivoting

The QR factorization with column pivoting for a matrix $A \in \mathbb{R}^{m \times n}$ is

Definition: QR with Pivoting

$AP = QR$ where

1. $P \in \mathbb{R}^{m \times m}$ is a permutation matrix
2. $Q \in \mathbb{R}^{m \times m}$ has orthonormal columns
3. $R \in \mathbb{R}^{m \times n}$ is upper triangular where $|r_{11}| \geq |r_{22}| \geq \cdots \geq |r_{nn}|$

Numerical Rank with Column Pivoting

- Previously, to find the low-rank approximation of a matrix, we examined the **singular values**

Numerical Rank with Column Pivoting

- Previously, to find the low-rank approximation of a matrix, we examined the **singular values**
- Now, we examine the **diagonal entries** of R

Numerical Rank with Column Pivoting

- Previously, to find the low-rank approximation of a matrix, we examined the **singular values**
- Now, we examine the **diagonal entries** of R
- For a given $\epsilon > 0$, if $|r_{kk}| \geq \epsilon \geq |r_{k+1,k+1}|$, then we can write $AP = QR$ as the following block matrix

Numerical Rank with Column Pivoting

- Previously, to find the low-rank approximation of a matrix, we examined the **singular values**
- Now, we examine the **diagonal entries** of R
- For a given $\epsilon > 0$, if $|r_{kk}| \geq \epsilon \geq |r_{k+1,k+1}|$, then we can write $AP = QR$ as the following block matrix

Definition: R Block Matrix

$$AP = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} = Q_1 \begin{bmatrix} R_{11} & R_{12} \end{bmatrix} + \begin{bmatrix} 0 & Q_2 R_{22} \end{bmatrix}$$

Numerical Rank with Column Pivoting

Definition: R Block Matrix

$$AP = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} = Q_1 \begin{bmatrix} R_{11} & R_{12} \end{bmatrix} + \begin{bmatrix} 0 & Q_2 R_{22} \end{bmatrix} \text{ where}$$

1. $R_{11} \in \mathbb{R}^{k \times k}$, $R_{12} \in \mathbb{R}^{k \times (n-k)}$, and $R_{22} \in \mathbb{R}^{(n-k) \times (n-k)}$
2. $Q_1 \in \mathbb{R}^{m \times k}$ and $Q_2 \in \mathbb{R}^{m \times (n-k)}$

- If we can show that R_{22} is numerically negligible, then

Numerical Rank with Column Pivoting

Definition: R Block Matrix

$$AP = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} = Q_1 \begin{bmatrix} R_{11} & R_{12} \end{bmatrix} + \begin{bmatrix} 0 & Q_2 R_{22} \end{bmatrix} \text{ where}$$

1. $R_{11} \in \mathbb{R}^{k \times k}$, $R_{12} \in \mathbb{R}^{k \times (n-k)}$, and $R_{22} \in \mathbb{R}^{(n-k) \times (n-k)}$
2. $Q_1 \in \mathbb{R}^{m \times k}$ and $Q_2 \in \mathbb{R}^{m \times (n-k)}$

- If we can show that R_{22} is numerically negligible, then
  1. Q_1 will give a $\mathcal{O}(\epsilon)$ approximation to the range of A
  2. $Q_1 \begin{bmatrix} R_{11} & R_{12} \end{bmatrix}$ is our low-rank approximation of A with $num\_rank = k$

Numerical Rank with Column Pivoting

Definition: R Block Matrix

$$AP = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} = Q_1 \begin{bmatrix} R_{11} & R_{12} \end{bmatrix} + \begin{bmatrix} 0 & Q_2 R_{22} \end{bmatrix} \text{ where}$$

1. $R_{11} \in \mathbb{R}^{k \times k}$, $R_{12} \in \mathbb{R}^{k \times (n-k)}$, and $R_{22} \in \mathbb{R}^{(n-k) \times (n-k)}$
2. $Q_1 \in \mathbb{R}^{m \times k}$ and $Q_2 \in \mathbb{R}^{m \times (n-k)}$

- If we can show that R_{22} is numerically negligible, then
  1. Q_1 will give a $\mathcal{O}(\epsilon)$ approximation to the range of A
  2. $Q_1 \begin{bmatrix} R_{11} & R_{12} \end{bmatrix}$ is our low-rank approximation of A with $num\_rank = k$
- R_{22} is numerically negligible if the following inequality holds, where F is the Frobenius norm

$$||R_{22}||_2 \leq ||R_{22}||_F \leq 2^{-\frac{1}{2}}(n-k+1)\epsilon \tag{3}$$

Development of the QR Algorithm

- There are 3 primary ways that the standard QR factorization algorithm can be implemented

Development of the QR Algorithm

- There are 3 primary ways that the standard QR factorization algorithm can be implemented
    1. The Gram-Schmidt Process
    2. Householder Reflection Transformations
    3. Given's Rotations

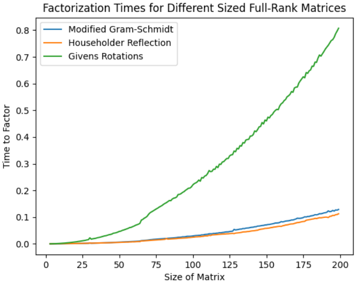
Development of the QR Algorithm

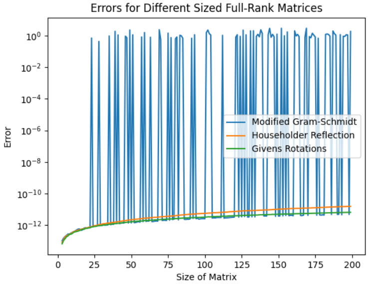
- There are 3 primary ways that the standard QR factorization algorithm can be implemented
    1. The Gram-Schmidt Process
    2. Householder Reflection Transformations
    3. Given's Rotations
- To test our QR factorization with column pivoting algorithm, we generated random rank-deficient matrices by forcing a numerical rank

Development of the QR Algorithm

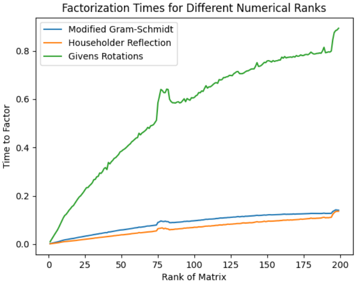
- There are 3 primary ways that the standard QR factorization algorithm can be implemented
  1. The Gram-Schmidt Process
  2. Householder Reflection Transformations
  3. Given's Rotations
- To test our QR factorization with column pivoting algorithm, we generated random rank-deficient matrices by forcing a numerical rank

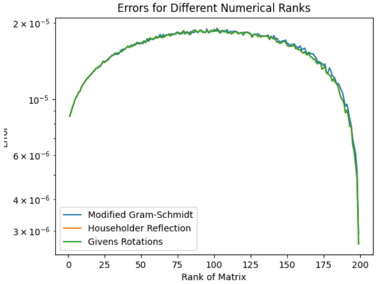
Comparing QR Implementations



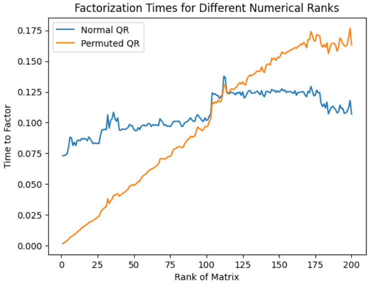


Comparing QR Implementations





Comparing QR Implementations



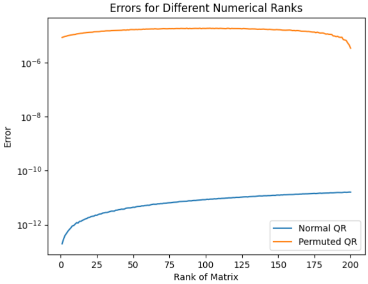


Table of Contents

Background Information

- Applied the RRQRF to a signal processing problem developed by Per Christian Hansen and Soren Holdt Jensen [1]

Background Information

- Applied the RRQRF to a signal processing problem developed by Per Christian Hansen and Soren Holdt Jensen [1]
- Estimates a clean signal from a noisy one

Background Information

- Applied the RRQRF to a signal processing problem developed by Per Christian Hansen and Soren Holdt Jensen [1]
- Estimates a clean signal from a noisy one
- Why signal processing?

Background Information

- Applied the RRQRF to a signal processing problem developed by Per Christian Hansen and Soren Holdt Jensen [1]
- Estimates a clean signal from a noisy one
- Why signal processing?
  - Inspired by recent machine learning techniques to produce the song *Now and Then* by the Beatles

Background Information: Hankel Matrix

- What is a Hankel matrix?

Background Information: Hankel Matrix

- What is a Hankel matrix?
  - Consider a signal $s \in \mathbb{R}^N$ where $s = \begin{bmatrix} s_1 & s_2 & \dots & s_N \end{bmatrix}^T$
  - The corresponding Hankel matrix is

$$\mathbb{H}(s) = \begin{bmatrix} s_1 & s_2 & s_3 & \dots & s_n \\ s_2 & s_3 & s_4 & \dots & s_{n+1} \\ s_3 & s_4 & s_5 & \dots & s_{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_m & s_{m+1} & s_{m+2} & \dots & s_N \end{bmatrix}$$

Background Information: Hankel Matrix

- What is a Hankel matrix?
  - Consider a signal $s \in \mathbb{R}^N$ where $s = \begin{bmatrix} s_1 & s_2 & \dots & s_N \end{bmatrix}^T$
  - The corresponding Hankel matrix is

$$\mathbb{H}(s) = \begin{bmatrix} s_1 & s_2 & s_3 & \dots & s_n \\ s_2 & s_3 & s_4 & \dots & s_{n+1} \\ s_3 & s_4 & s_5 & \dots & s_{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_m & s_{m+1} & s_{m+2} & \dots & s_N \end{bmatrix}$$

- What is a covariance matrix?

Background Information: Hankel Matrix

- What is a Hankel matrix?
  - Consider a signal $s \in \mathbb{R}^N$ where $s = \begin{bmatrix} s_1 & s_2 & \dots & s_N \end{bmatrix}^T$
  - The corresponding Hankel matrix is

$$\mathbb{H}(s) = \begin{bmatrix} s_1 & s_2 & s_3 & \dots & s_n \\ s_2 & s_3 & s_4 & \dots & s_{n+1} \\ s_3 & s_4 & s_5 & \dots & s_{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_m & s_{m+1} & s_{m+2} & \dots & s_N \end{bmatrix}$$

- What is a covariance matrix?
  - A square matrix C where the diagonal entries are the variance and the remaining entries are the covariance.

The Algorithm

- Below are the steps of the algorithm implemented from the paper

The Algorithm

- Below are the steps of the algorithm implemented from the paper
  1. Obtain pure and noisy signals and combine the two, $s = \bar{s} + e$

The Algorithm

- Below are the steps of the algorithm implemented from the paper
  1. Obtain pure and noisy signals and combine the two, $s = \bar{s} + e$
  2. Build corresponding Hankel matrix, $\mathbb{H}(s)$

The Algorithm

- Below are the steps of the algorithm implemented from the paper

  1. Obtain pure and noisy signals and combine the two, $s = \bar{s} + e$
  2. Build corresponding Hankel matrix, $\mathbb{H}(s)$
  3. Build corresponding covariance matrix from the Hankel matrix, $C(\mathbb{H}(s))$

The Algorithm

- Below are the steps of the algorithm implemented from the paper
  1. Obtain pure and noisy signals and combine the two, $s = \bar{s} + e$
  2. Build corresponding Hankel matrix, $\mathbb{H}(s)$
  3. Build corresponding covariance matrix from the Hankel matrix, $C(\mathbb{H}(s))$
  4. Put $C(\mathbb{H}(s))$ through RRQRF to find numerical rank k

The Algorithm

- Below are the steps of the algorithm implemented from the paper

1. Obtain pure and noisy signals and combine the two, $s = \bar{s} + e$
2. Build corresponding Hankel matrix, $\mathbb{H}(s)$
3. Build corresponding covariance matrix from the Hankel matrix, $C(\mathbb{H}(s))$
4. Put $C(\mathbb{H}(s))$ through RRQRF to find numerical rank k
5. Force rank k on Hankel matrix, which gives us an estimate of the pure signal

The Algorithm

- Below are the steps of the algorithm implemented from the paper
  1. Obtain pure and noisy signals and combine the two, $s = \bar{s} + e$
  2. Build corresponding Hankel matrix, $\mathbb{H}(s)$
  3. Build corresponding covariance matrix from the Hankel matrix, $C(\mathbb{H}(s))$
  4. Put $C(\mathbb{H}(s))$ through RRQRF to find numerical rank k
  5. Force rank k on Hankel matrix, which gives us an estimate of the pure signal
  6. Extract pure signal from rank k Hankel approximation through summing the anti-diagonal elements

Pure Sine Signal

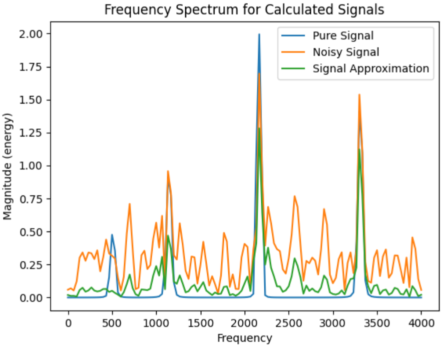
- Before using a speech signal, we consider an artificial sine signal, $\bar{s}$, and Gaussian white noise, e, for a combined noisy signal, $s = \bar{s} + e$

Pure Sine Signal

- Before using a speech signal, we consider an artificial sine signal, $\bar{s}$, and Gaussian white noise, e, for a combined noisy signal, $s = \bar{s} + e$
- With the steps outlined in the previous slide, we obtained the following results

Pure Sine Signal

- Before using a speech signal, we consider an artificial sine signal, $\bar{s}$, and Gaussian white noise, e, for a combined noisy signal, $s = \bar{s} + e$
- With the steps outlined in the previous slide, we obtained the following results



Speech Signal

- Our speech signal is a 3-second recording of someone speaking and adding Gaussian white noise.

Speech Signal

- Our speech signal is a 3-second recording of someone speaking and adding Gaussian white noise.
- Before implementing the steps previously mentioned, we needed to
    1. Estimate variance of the noise from the gaps in the speech patterns
    2. Split 3-second recording into disjoint sub-intervals of 30 milliseconds

Speech Signal

- Our speech signal is a 3-second recording of someone speaking and adding Gaussian white noise.
- Before implementing the steps previously mentioned, we needed to
  1. Estimate variance of the noise from the gaps in the speech patterns
  2. Split 3-second recording into disjoint sub-intervals of 30 milliseconds
- Then for each interval, we complete steps 2 - 6 previously mentioned

Speech Signal

- Our speech signal is a 3-second recording of someone speaking and adding Gaussian white noise.
- Before implementing the steps previously mentioned, we needed to
  1. Estimate variance of the noise from the gaps in the speech patterns
  2. Split 3-second recording into disjoint sub-intervals of 30 milliseconds
- Then for each interval, we complete steps 2 - 6 previously mentioned
- Recombine estimated signal segments

Speech Signal Results

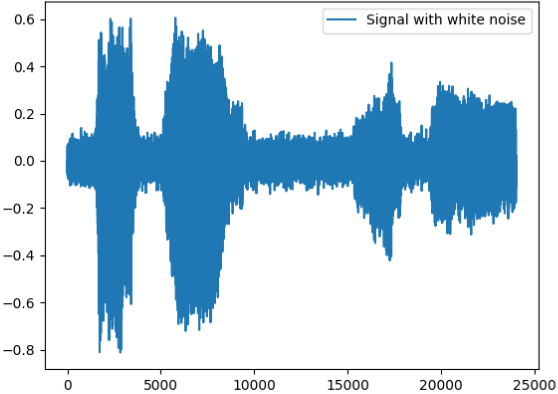


Figure: Noisy Speech Signal

Speech Signal Results

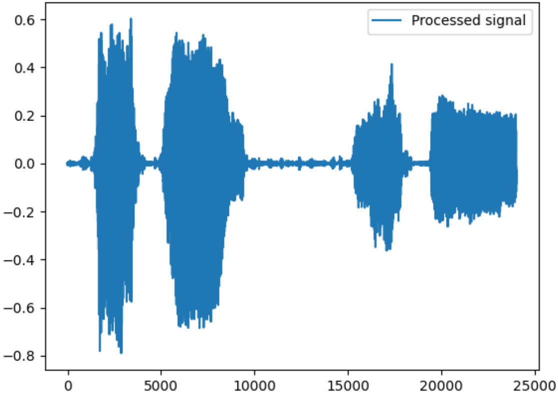


Figure: Processed Signal

Speech Signal Results

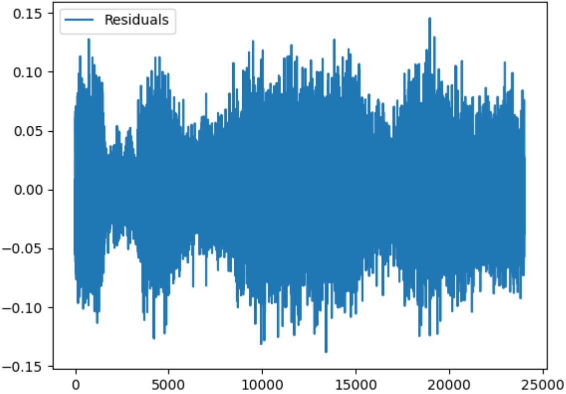


Figure: Residuals

Speech Signal Results

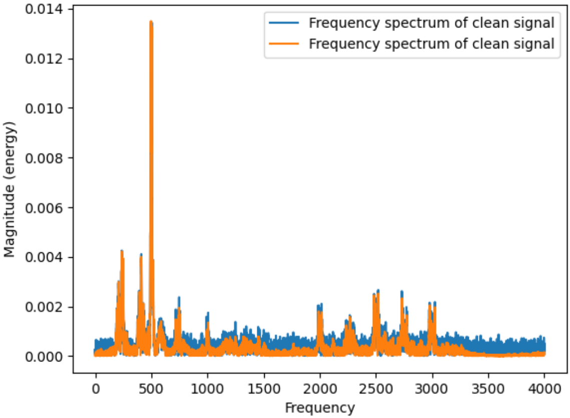


Figure: Spectra Error

Table of Contents

Conclusion

- For low-rank matrices approximations, QR factorization with pivoting is faster

Conclusion

- For low-rank matrices approximations, QR factorization with pivoting is faster
- Householder reflection matrices provided the best balance between speed of factorization and stability when compared against both matrix size and matrix rank

Conclusion

- For low-rank matrices approximations, QR factorization with pivoting is faster
- Householder reflection matrices provided the best balance between speed of factorization and stability when compared against both matrix size and matrix rank
- The RRQRF can extract pure and noisy signals from pure sine waves and speech signals

Conclusion

- For low-rank matrices approximations, QR factorization with pivoting is faster
- Householder reflection matrices provided the best balance between speed of factorization and stability when compared against both matrix size and matrix rank
- The RRQRF can extract pure and noisy signals from pure sine waves and speech signals
- Can form good estimates without having any knowledge about the noise in the signal itself.

Future Work

- Continuing to increase the efficiency of the QR algorithm

Future Work

- Continuing to increase the efficiency of the QR algorithm
- Further explore non-white noise methods mentioned in the paper
  - General noise
  - Rank-deficient noise

References

[1] Per Christian Hansen and Søren Holdt Jensen. "Subspace-Based Noise Reduction for Speech Signals via Diagonal and Triangular Matrix Decompositions: Survey and Analysis". In: *EURASIP Journal on Advances in Signal Processing* 2007.1 (2007), p. 092953. DOI: 10.1155/2007/92953. URL: https://doi.org/10.1155/2007/92953.