QR Factorization Using Householder Transformation

Sakshi Reddy , Nishita Singam , Surya Kiran

Department of Science and Electrical Engineering
University of Maryland, Baltimore County

ABSTRACT:

Earlier, the QR decomposition of a non-singular matrix was calculated using a variety of algorithms. Examples include the Schmidt algorithm, the modified Gram-Schmidt algorithm, re-orthogonalization, etc. However, "Householder transformations" produced superior outcomes because the matrix's orthogonality was kept and they were comparatively more stable. In linear algebra, a QR decomposition, also known as a QR factorization or QU factorization, is a decomposition of a matrix A into a product A = QR of an orthogonal matrix Q and an upper triangular matrix R. We do implementation of Handwritten Digit Classification. To improve the project and minimize the cost of calculation, we will use the QR decomposition algorithm for categorization. We collect the data needed to put the classification method for handwritten digits into practice, Testing the project and accuracy of the algorithm.

INTRODUCTION:

Alston Scott Householder applied the Householder transformation in a 1958 publication in linear algebra, the division of a matrix A into an orthogonal matrix Q and an upper triangular matrix R is known as a QR factorization or QU factorization. The resolution of linear systems, eigenvalue issues, and least squares approximations can all be aided by QR-decompositions. Numerous methods, such as the Gram-Schmidt process, Householder transformations, and Givens rotations, can be used to compute the QR decomposition. Each offers a number of advantages and disadvantages. The constructor will never fail because the QR decomposition exists even if the matrix does not have complete rank. The least squares solution of non-square systems of concurrent linear equations is where the QR decomposition is most frequently applied. The decomposition is distinct if R's diagonal entries must be positive.

In order to solve linear least squares problems, compute eigenvalues, and solve linear systems of equations, QR factorization plays a vital role. Identical issues can occur with wireless cellular systems and geo-location. For a matrix A of size i x j QR factorization is given by

$$A = QR$$

Where, Q is an orthogonal matrix of size i x i and R is a matrix of size i x j which is an upper triangular matrix. Using a set of manually entered data, train the algorithm to classify each numerical value into a matrix, then organize the values in ascending order with labels for the quickest response. For the process we use QR-decomposition using Householder transformation.

BACKGROUND AND MOTIVATION:-

Character recognition in handwriting has emerged since the 1980s. The problem of handwritten digit identification using a classifier is indeed very important and has numerous applications, including online handwriting recognition on computer tablets, analysing bank check amounts, sorting postal mail by postal code, and numeric inputs in details worked out by hand (such as tax filings). While attempting to address this problem, many difficulties are encountered. The thickness, size alignment, and positioning of the handwritten digits in relationship to the borders are not always consistent.

A Householder transformation, often referred to as a Householder reflection or an elementary reflector, is a type of linear transformation used to represent a reflection around a plane or hyperplane that contains the origin. The Householder operator is similar to it when applied to general inner product spaces.

In order to efficiently parametrize unitary operators, it was demonstrated that the Householder transformation has a one-to-one relationship with the canonical coset decomposition of unitary matrices as specified in group theory.

Last but not least, we point out that a single Householder transform, as opposed to a single Givens transform, can work on all columns of a matrix and, as a result, has the lowest computing cost for QR decomposition and tridiagonalization.

RELATED WORK:

In 1958, Alston S. Householder developed the Householder Transformation, which outperformed the conventional Givens Rotation in terms of computational efficiency. Since then, numerous technical and computational advances have led to the development of numerous distinct types of Householder Transformation implementations. The handwritten digits can also be produced using neural networks and other sophisticated machine learning techniques. With the householder transformation and the QR decomposition. The matrix created as a result of this transformation can be defined as the outer product of two coordinate vectors.

Systems of linear equations can also be solved using the Householder method. Compared to the Gaussian elimination approach, the method is substantially more stable. However, compared to the Gaussian approach, this method requires more time to reach the solution.

METHODOLOGY:

We have first divided the dataset into 4 parts, train dataset, test dataset, correct labels for test dataset and labels for train dataset.

```
# Load files containing handwritten digit image data

# Training data
Training_data = scipy.io.loadmat('D:/NC/Project/azip.mat')['azip']

# Testing data
Testing_data = scipy.io.loadmat('D:/NC/Project/testzip.mat')['testzip']

# Correct labels for testing data
Correct_labels = scipy.io.loadmat('D:/NC/Project/dtest.mat')['dtest']

# Labels for training data
Labels_training_data = scipy.io.loadmat('D:/NC/Project/dzip.mat')['dzip']
```

Fig: Loading files containing handwritten digit image data for training, testing of data

The dimensions of each dataset are as follows:
Dimensions of train dataset: (256, 1707)
Dimensions of test dataset: (256, 2007)
Dimensions of test labels dataset: (1, 2007)
Dimensions of train labels dataset: (1, 1707)

For classification of handwritten digits, we are first sorting the data according to the labels, in ascending order. I.e. images for digit 0 will come first and for digit 9 will be last in the train dataset, train dataset labels and after this we aggregated both dataset to get proper aligned digits and their labels to train our model.

For each digit in train dataset, we have following numbers of digit-label pairs
DigitSetSizes: [319, 252, 202, 131, 122, 88, 151, 166, 144, 132]
The first entry of each number in sorted train dataset were shown as following –

Then the train dataset was partitioned into separate arrays for each digit from 0 to 9.

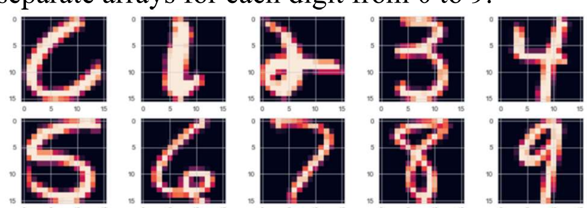

Fig.: Each image of the digit is in the form of a 16X16 matrix of pixel values.

For the project, we have used SVD and QR decomposition to reduce the computational expenses and complexity of computation. When multiple solutions need to be computed with only minor changes in the underlying data, knowledge of the difference between the old data set and the new can be used to update an existing factorization at reduced computational cost

SVD is a Singular Value Decomposition, it is used to compress the high resolution data into just the key features that are necessary for analysis of the data

We found the singular values for each digit and they are significantly decreasing after 20. So we can eliminate the data after 20.

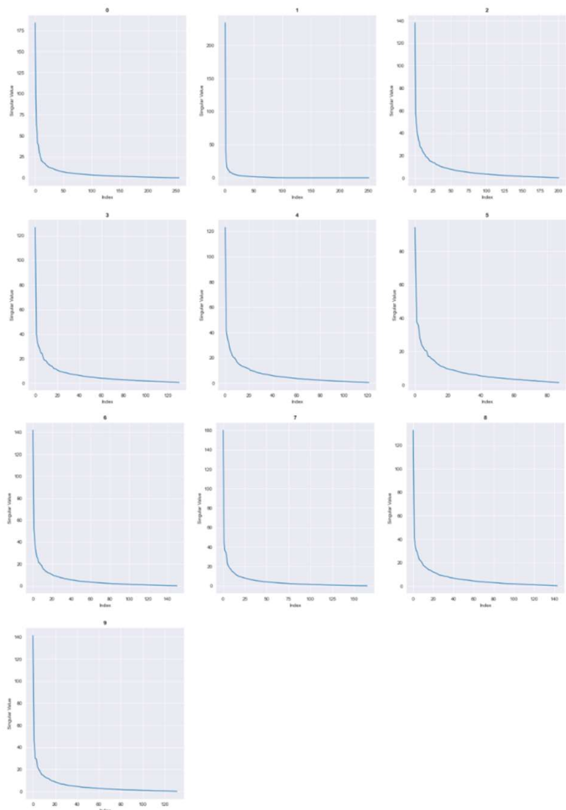

Fig: Graph of Singular value for each digit

Here we are using the key as 15 because, when we tested it with rank value as 5, the accuracy of the algorithm was reduced to 90% and when we selected rank value as 20, the accuracy of the algorithm was the same i.e. 94%.

This shows us that we don't have to do computation for huge matrices, we can reduce them and still get the same results.

```
# Compress each digit set using a rank 15 SVD approximation and store results into list
SVD_Rank_15 = [np.zeros((1,1))[0]]*10
Plot_S = [np.zeros(1)]*10
k=15
for j in range(10):
    U,S,V = np.linalg.svd(Digits_partitioned[j],full_matrices=False)
    U = U[:,0:k] # Grab first 15 dominant singular column vectors for each digit set
    Plot_S[j] = S
    SVD_Rank_15[j] = U # Store dominant singular column vectors
```

The Householder reflection matrix is an nxn orthogonal matrix:

$$P = I - \frac{2}{v^T v} v v^T, \qquad 0 \neq v \in \mathbb{R}^n.$$

Multiplying this with a vector of matrices, gives us an upper triangular matrix.
To implement the household transformation, we have implemented 2 functions, the householder_vectorized() function calculates the householder vector value for each column of the SVD output matrix. The output of this function is the household vector and the tau value.
The qr_decomposition() function, initializes the value of Q to an identity matrix and R is a copy of the SVD_Rank_15. The qr_decomposition() function, calls the householder_vectorized() function for each column of the matrix. This function also calculates the final value of Q and R using the householder vector and tau value by the above given function. We are storing the final values of Q and R for each digit are stored in the QR array to use for further computation.

```
#Householder transformation algorithm to implement QR decomposition on each digit set in SVD_Rank_15
def householder_vectorized(a):

    v = a / (a[0] + np.copysign(np.linalg.norm(a), a[0]))
    v[0] = 1
    tau = 2 / (v.T @ v)

    return v,tau

def qr_decomposition(A: np.ndarray) -> Union[np.ndarray, np.ndarray]:
    m,n = A.shape
    R = A.copy()
    Q = np.identity(m)

    for j in range(0, n):
        # Apply Householder transformation.
        v, tau = householder_vectorized(R[j:, j, np.newaxis])

        H = np.identity(m)
        H[j:, j:] -= tau * (v @ v.T)
        R = H @ R
        Q = H @ Q

    return Q[:n].T, np.triu(R[:n])

    return Q[:n].T, R[:n]
```

```
#calculating QR decomposition and storing it in an array
QR = [[np.zeros((1,1))[0],np.zeros((1,1))[0]]]*10
for i in range(10):
    Q,R = qr_decomposition(SVD_Rank_15[i])
    QR[i] = [Q,R]
```

```
# Solve system of equations for each digit set in Digits_partitioned and each Testing_data digit in Testing_data
Residuals = [0]*10
Predicts = [0]*len(Correct_labels[0])
for i in range(len(Correct_labels[0])):
    for j in range(10):
        x = np.matmul(np.matmul(np.linalg.inv(QR[j][1]),np.transpose(QR[j][0])),Testing_data[:,i])
        Residuals[j] = np.linalg.norm(np.matmul(SVD_Rank_15[j],x)-Testing_data[:,i],ord=2)
    Predicts[i] = np.argmin(np.array(Residuals))
    Residuals = [0]*10
```

To predict the label of test dataset entries, we are using a system of equation, that uses the Q and R value of the train dataset and calculates the residual value for each digit entry in test data. The final predicted digit is considered based on the residual value, the digit having minimum residual value is selected as a predicted label for the test dataset entry.

RESULTS:

Following confusion matrix shows the classification done by the model for each digit, rows represent the predicted value and column represents the correct labels for the test dataset.

	0	1	2	3	4	5	6	7	8	9
0	355	0	9	3	1	4	3	0	3	0
1	0	260	1	0	1	1	1	1	0	3
2	2	0	179	3	0	2	0	1	1	0
3	0	0	3	148	0	9	0	0	4	0
4	1	3	2	1	186	0	2	3	0	3
5	0	0	1	8	1	140	1	0	2	0
6	0	1	0	0	3	0	162	0	0	0
7	0	0	1	1	3	0	0	141	1	4
8	0	0	2	2	0	1	1	0	153	2
9	1	0	0	0	5	3	0	1	2	165

Fig: Confusion matrix

As we can see in the confusion matrix, the 0 was predicted as 0 for 355 test entries and for 2 times, it was identified as 2 and 1 time it was identified as 4 and respectively.

The confusion matrix gives a clear idea about the predicted digits and which digits are getting predicted wrong. As 2 was predicted as 0 for 9 test entries.

For each digit, the accuracy is shown in the following table and the plotted graph.

	Accuracy
0	0.988858
1	0.984848
2	0.904040
3	0.891566
4	0.930000
5	0.875000
6	0.952941
7	0.959184
8	0.921687
9	0.932203

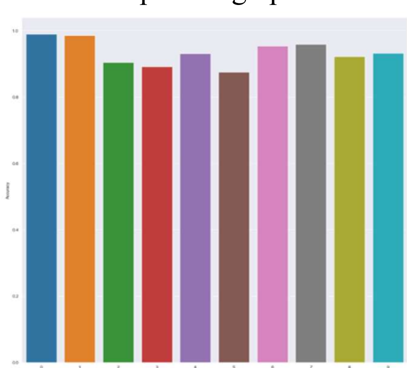

Fig: Accuracy of each digit for handwritten digits

As we can see that the accuracy of digit 0 is highest, and the accuracy of the digits 2,3 and 5 is least.

Our model was able to classify the handwritten digits with an overall accuracy of 94%. The computation expense required for 16X16 matrix computation was reduced and the complexity of the matrix multiplication was decreased due to the use of Singular Value Decomposition and QR decomposition using Householder Transformation.

APPLICATIONS:

The fundamental aspect of handwritten recognition research has been provided in our current study. Consequently, this work will inspire potential readers to pursue careers in pattern recognitions. The use of handwriting recognition technology includes such things like Postal office automation with code number recognition on Envelope, Automatic license plate recognition, Bank automation, National ID number recognition system.

(a) Postal office automation with code number recognition on Envelope:

It facilitates accurate data entry into a system quickly and minimizes manual data redundancy errors. It is often used in the postal service since postal IDs tend to be written in numbers, making it easier to recognise using the handwritten digits technique.

Fig: Postal card sample

(b) Automatic license plate recognition:

Automated license plate readers (ALPRs) adopt a technology called handwritten digit recognition. The lightning-fast, computer-controlled surveillance systems which are commonly mounted to police vehicles, streetlights, flyovers, and street poles. All visible license plate numbers, as well as the date, time and place, are automatically recorded by ALPRs.

Police may use the data they've gathered to track down earlier positions for license plates, assess if a vehicle was located at a scene of crime, and detect cars that might be associated in the crime.

Fig: License Plate sample

(c) Bank automation:

The bank account number, amount of cash, and identity of the account owner are filled out on to the bank paper during a conventional cash deposit process. The account number and amount to be deposited are then manually entered into the computer. When there are plenty of customers at once, the procedure will take a little while, and occasionally the banker might miss things when reading or typing the data. The account number and the sum are read by the system and accurately recorded in this case, making advantage of the handwritten digit recognition.

(d) National ID number recognition system:

The national ID, which represents the identity of the user in the country, is very important. .In a numerous applications, the national ID would be a mandatory field. In situations when the written numerals of the national ID are difficult for individuals to read and comprehend, the handwritten digits could be used to identify the digits of the person's identity Card.

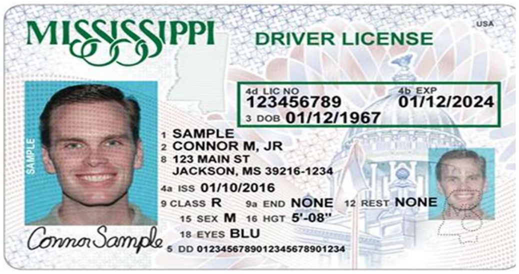

Fig: National ID sample

CHALLENGES AND DIFFICULTIES:

1. The Python libraries that we intend to utilize for this project also have a significant expectation of learning and adaptation requirements to get good hands-on.
2. Strokes have such a significant amount of difference and uncertainty from individual to individual.
3. An individual's writing pattern obviously varies and evolves throughout period.
4. Due to the deterioration with time, the actual material or image seems to be of poor quality.
5. Although people need not absolutely have to write all the numbers in a linear fashion on paper, content in printed material usually sits in a straight line.
6. In distinction to printed material, generally sits bolt upright throughout, written text may have a variable tilt towards the side.
7. Comparing overall value of accurate named dataset collecting against simulated data

FUTUREWORK AND IMPROVEMENTS:

The handwritten digits those are [0-9] numerical can be detected by this project as of the current stage. The detection of textual content, encompassing alphabet letters, special characters, recognition for the character in different languages, Music notes and other characters as well also so forth, is also another area we would really want to expand our project on. As this has a broad array of applications. Some of them are like prescriptions automatically converted to typed forms, forensics and biometrics, composing and transmitting Messages in one's native language, fully automated notation reader for music, Using one's own handwriting and native script while creating computer apps, filling out forms online.

CONCLUSION:

One among the most difficult aspects of pattern recognition is handwriting detection. The project's objective is to characterize numerical data, the majority of which have been stored as captured images. On the basis of unique combinations, a variety of techniques for pattern recognition have been employed to identify writing either online and offline.

In order to accurately interpret handwritten digits, this study that looked at a representation of individual digits. In order to identify handwritten numbers, various machine learning approaches were applied in this research. The major challenge in any recognition procedure is to deal with accurate feature classification and extraction methods. The suggested approach makes an effort to deal with accuracy-related variables. Tested written digits were accurately recognized by our system 94% of the times. Overall, the findings were great.

REFERENCES:

[1] Gill, Phillip E., Walter Murray, and Margaret H. Wright. Numerical Linear Algebra and Optimization. New York: Addison-Wesely, 1991.

[2] Kerl, John. "The Householder Transformation in Numerical Analysis." January 22, 2007

[3] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammerling, J. Demmel, C. Bischof, and D. Sorensen, "Lapack: A portable linear algebra library for high-performance computers," in Proceedings of the 1990 ACM/IEEE Conference on Supercomputing, ser. Supercomputing '90. Los Alamitos, CA, USA: IEEE Computer Society Press, 1990, pp. 2–11. [Online]. Available: http://dl.acm.org/citation.cfm?id=110382.110385

[4] Taboga, Marco (2021). "Householder matrix", Lectures on matrix algebra.https://www.statlect.com/matrix-algebra/Householder-matrix.

[5] H. Drucker, R. Schapire, and P. Simard, Boosting Performance in Neural Networks, International Journal of Pattern Recognition and Artificial Intelligence 7 705-720 (1993).

[6] John S. Denker and Christopher C. J. Burges, Image Segmentation and Recognition, in The Mathematics of Induction, D. H. Wolpert (ed.), Addison-Wesley (1994).

[7] S. Jaeger, C. L. Liu, M. Nakagawa, "The state of art in Japanese online handwriting recognition compared to techniques in Western handwriting recognition", International Journal on Document Analysis and Recognition, vol. 6, pp. 75-88, 2003.

[8] https://arxiv.org/pdf/1612.04470.pdf

[9] F. Merchant, T. Vatwani, A. Chattopadhyay, S. Raha, S. K. Nandy and R. Narayan, "Efficient Realization of Householder Transform Through Algorithm-Architecture Co-Design for Acceleration of QR Factorization," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 8, pp. 1707-1720, 1 Aug. 2018, doi: 10.1109/TPDS.2018.2803820.

[10] Lloyd N. Trefethen, Householder triangularization of a quasimatrix, IMA Journal of Numerical Analysis, Volume 30, Issue 4, October 2010, Pages 887–897, https://doi.org/10.1093/imanum/drp018

[11] Ahmed, P. and Suen, C.Y., \Computer recognition of totally unconstrained handwritten ZIP codes," Inter. J. Pattern Recog. and Artif. Intell., 1, 1-15, 1987.

[12] Amit, Y. and Kong, A., \Graphical templates for image matching," Technical report no. 373, Dept. of Statistics, University of Chicago, 1993.

[13] Amit, Y., \Graphical shape templates for deformable model registration with application to MRI brain scans," Technical Report, Department of Statistics, University of Chicago, 1994.

[14] Arkin, E., Meijer, H., Mitchell, J., Rappaport, D., and Skiena, S., \Decision trees for geometric models," Proc. Ninth ACM Symp. on Computational Geometry, 1993.