FEDERATED MATRIX COMPLETION TECHNIQUE

Ansh Goyal

Shubh Shankar

Vishwas Kaushik

Abstract

This paper introduces a novel approach that combines Federated Learning (FL) with Matrix Completion (MC) using the Nystrom approximation algorithm. Matrix refers to a collection of large-scale data in the form of a 2D array. Data is generally stored in the form of a matrix. The problem of missing data arises when the dimensions of the matrices become extremely large and the matrix is sparsely populated. In such cases, Nystrom approximation is used as one of the methods for matrix completion [1]. However, when matrix completion algorithms are applied in real-life scenarios, the privacy of the clients is often jeopardized [2]. Moreover, the computation time for such a large amount of data is very high [3]. To prevent such a thing from happening, we have proposed a decentralized approach to the given problem, which also reduces the overall computation time and load on a server. In this paper, two approaches of Federated Nystrom approximation have been devised and implemented. The results have been recorded and analyzed thoroughly. The methodology and the results will be discussed in this paper.

1 Introduction

This paper investigates the incorporation of matrix completion techniques with federated learning to address collaborative data privacy challenges. Matrix completion is a powerful tool for imputing missing entries in matrices, finding applications in recommendation systems, and collaborative filtering The key principle of federated learning is training a machine learning model without needing to know each user's raw private data [4]. It is a method for recovering lost information. It originates from machine learning and usually deals with highly sparse matrices. Missing or unknown data is estimated using the low-rank matrix of the known data [5].The Nyström method is an efficient technique to generate low-rank matrix approximations and is used in several large-scale learning applications. This method's key aspect is the procedure for which columns are sampled from the original matrix [6].

Federated Nyström Approximation

Nyström approximation is a technique used in linear algebra for approximating missing information in large matrices. It's particularly useful when dealing with kernel matrices, which arise in machine learning and other computational applications. In the context of kernel matrices, Nyström approximation works by selecting a subset of rows and columns from the original matrix and using them to construct a low-rank approximation. The key idea is that by choosing a representative subset, often referred to as pivot points, one can approximate the entire matrix with much lower computational cost [7] [8].

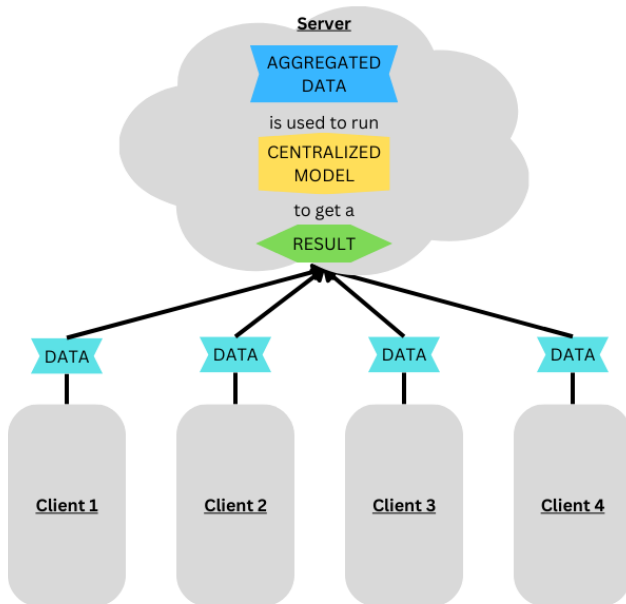


Figure 1: Traditional Machine Learning

In the traditional machine learning models, it is customary for clients to send their actual data to the server. The server collects the data from all the clients, aggregates it, and uses that data to train and test the accuracy of the models. In such cases the data is centralized and hence data privacy and security is compromised [9]. Moreover, the central server needs to collect and process all data, leading to high communication overhead, especially when dealing with large datasets. The figure 1, shows a pictorial representation of how a traditional Machine Learning Model works.

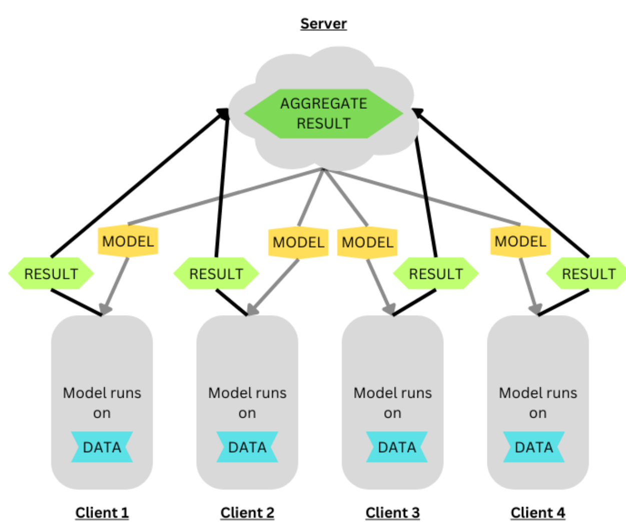


Figure 2: Federated Learning

This is why Federated Learning is considered an alternative. Federated Learning works on the concept of decentralized training models by distributing the training process across multiple devices or servers while keeping raw data localized [10]. It means that rather than sending the actual data of the client, the resultant updates (or gradients) are sent to the server after applying the model on the device itself. This reduces privacy risks associated with centralizing sensitive information and results in reduced communication costs hence being advantageous in scenarios where bandwidth is limited or expensive. The decentralized and privacy-preserving approach makes Federated Learning well-suited for certain applications, especially in contexts where data privacy and security are paramount [2]. The figure 2, shows a pictorial representation of how a traditional Machine Learning Model works.

2 Approach

The traditional Nystrom algorithm involves the approximation of missing data entries with the help of certain pivots and user data. Consider a kernel matrix K consisting of four components,

K = $\begin{bmatrix} A & B \\ B^t & Q \end{bmatrix}$

then Q is approximated by the formula

$$Q = B^t A^{-1} B$$

where,
A is the matrix containing pivot points
B is the matrix with user"s data
B^t is the transpose of B
A^{-1} is the inverse of A
Q is the resultant Matrix after applying Nystrom Approximation
Two main approaches have been considered for implementing federated Nyström completion.

- In the first method, a single kernel matrix is taken into account.
  1. The clients are free to choose how much information they share, based on the number of rows and columns they agree to share for the approximation.
  2. The pivot points are accordingly sent by the server and approximation is carried out.
  3. The final result is then aggregated into a single matrix and compared with the original to find errors.
- The second method is almost similar to the first one except that:
  1. all clients now possess their individual kernel matrix.
  2. The set of pivot points with the same dimensions are sent by the server and Nyström approximation is carried out.
  3. Then the aggregated matrix is compared to the average of respective parts of all kernel matrices to find out the error.

The proposed approach for Federated Nystrom is shown mathematically as follows

$$Q_i = B_i^t A_i^{-1} B_i$$

where,

A_i is the matrix containing pivot points for *ith* client

B_i is the matrix with user"s data for *ith* client

B_i^t is the transpose of B

A_i^{-1} is the inverse of A

Q_i is the resultant Matrix for *ith* client after applying Nystrom Approximation

The different Q_i are then aggregated in the form of $Q_a g$ by using simple averages for overlapping terms. The basic difference between the two approaches is the selection procedure of the B matrix. The data used for the analysis was generated independently. Low-rank positive semi-definite matrices having various dimensions have been used for the approximations.

Algorithm

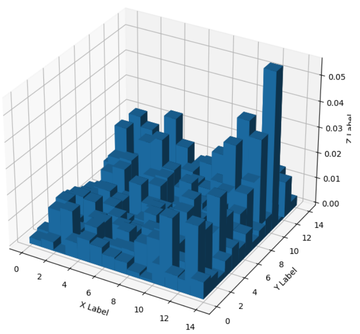
- func computePositiveDifferences original from client data:
  - return difference
- func Calculate the percentage error to the client from the original:
  - return percentageErrorMatrix
- func calculate Average Percentage Error:
  - return totalError / numElements
- func calculate FrobeniusNormError:
  - return FrobeniusNorm
- Procedure:
  - for Nyström approximation:
    - read kernel matrix from file
    - extract submatrices A, Qorg
    - extract client data B
    - compute Q using formula
    - calculate errors(%) and display results

- for federated Nyström :
  - read kernel matrix from file
  - extract client data
  - loop for each client:
    - read dimensions row and col
    - extract client original data subB
    - compute subQ for each client using subA sent by the server
    - calculate errors for each client and display the results
  - consolidate client data in the server then aggregate
  - calculate errors for aggregate data and display results
- for federated Nyström using diff kernel:
  - read multiple kernel matrices from files
  - loop for each client:
    - read dimensions row and col
    - extract clients data subB
    - compute subQ for each client using subA sent by the server
    - calculate errors for each client and display results
  - consolidate clients original data in QF and the resultant data in Qag
  - calculate errors for aggregated values and display results

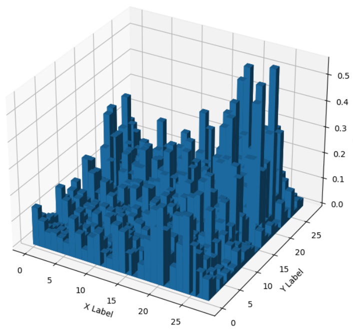
- end Procedure

3 Results and Analysis

A large amount of data was analysed during the course of the research work. Firstly, Nyström approximation algorithm was executed on low rank positive semi-definite matrices of varied sizes. Matrices of sizes 16X16, 32X32, 64X64, 100X100 and 128X128 were taken. the rank of the matrix was assumed to be 1:8 ratio of its size. The results are shown in the figures.



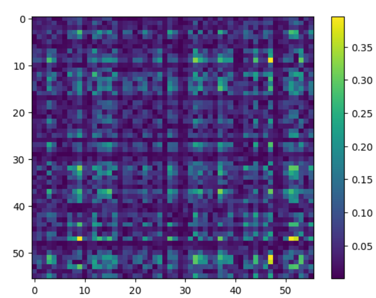
(a) size 16X16



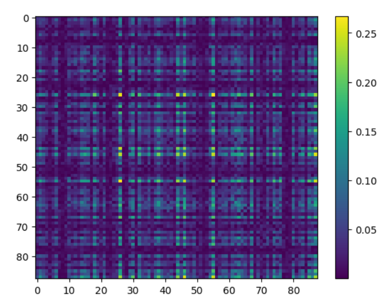
(b) size 32X32

Figure 3: Column Charts of Difference Matrix

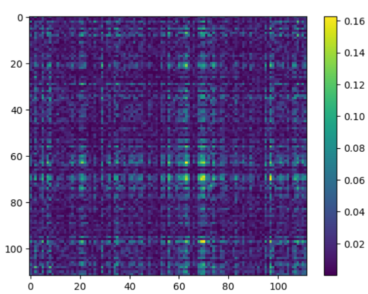
The figure 3, shows a column chart for the difference matrix obtained when the results of Nyström approximation were compared with the original matrix. The average percentage errors were reported to be 0.59603% and 0.268479% and Frobenius norm errors were 0.157801% and 3.07449% respectively.



(a) size 64X64



(b) size 100X100



(c) size 128X128

Figure 4: Heat Maps of Difference Matrix

The figure 4, shows heat maps for the difference matrix obtained when the results of Nyström approximation were compared with the original matrix. The average percentage errors were reported to be 0.0267757%, 0.00380605% and 0.00111726% and Frobenius norm errors were 4.98121%, 4.49045% and 3.28442% respectively.

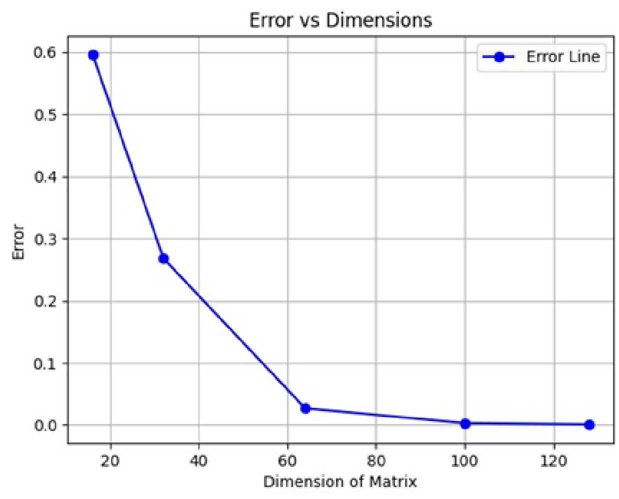
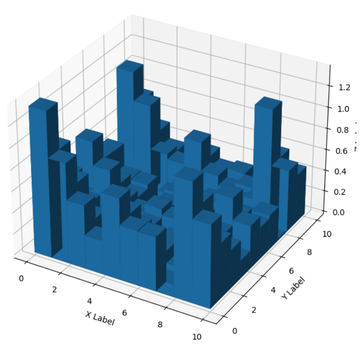
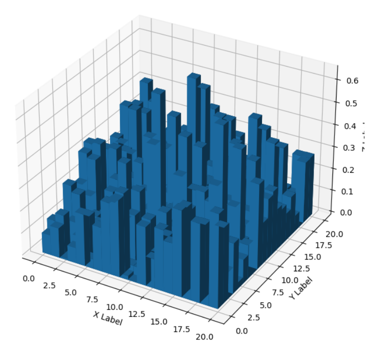


Figure 5: Line chart depicting Errors vs Size of a matrix

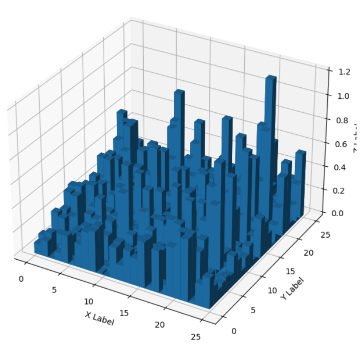
The figure 5, shows a line chart for a graph depicting total error vs size of the matrix. It is observed that the difference matrix has overall small individual values. This indicates that the Nyström approximation algorithm is fairly accurate and gives almost the same values with minimal error. This is considered as the base for future observations regarding federated Nyström approximation.



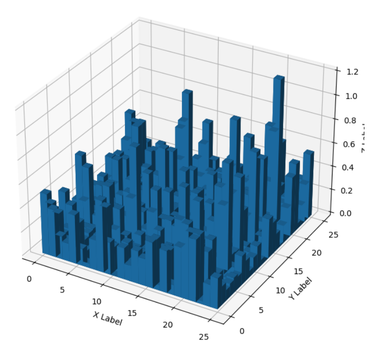
(a) Client 1



(b) Client 2



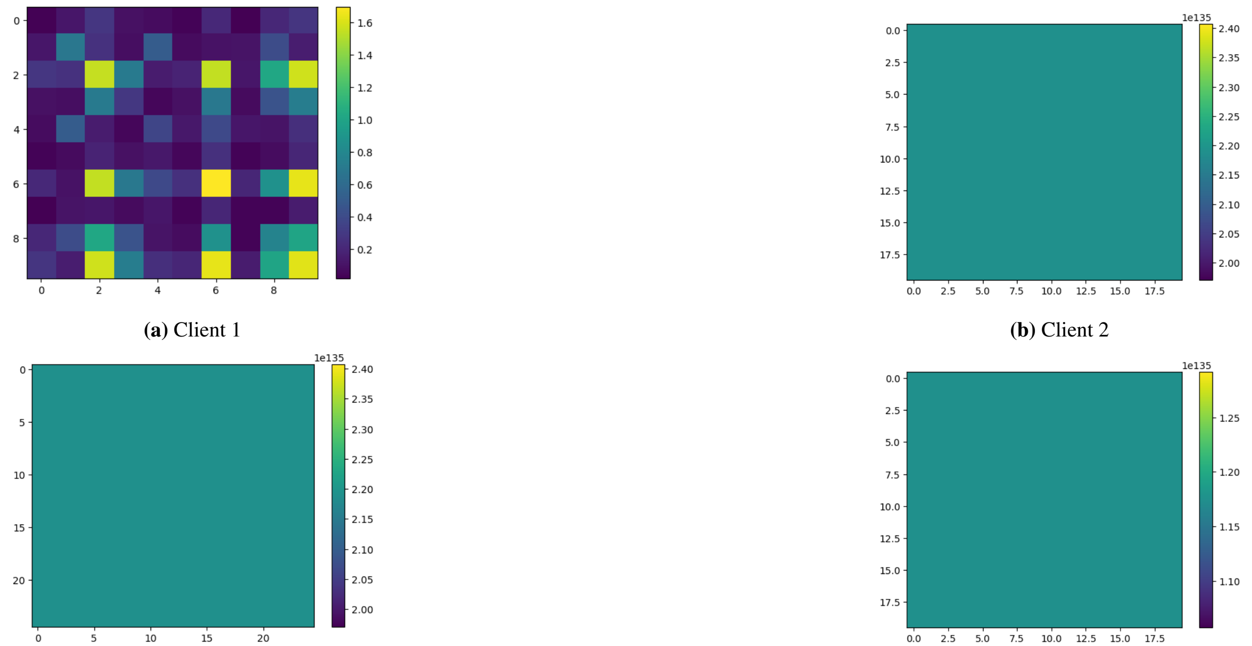
(c) Client 3



(d) Aggregated Matrix

Figure 6: Column Charts of Difference Matrix for clients having similar data

The figure 6, shows column charts for the difference matrix obtained for the first approach of Federated Nystrom approximation when the obtained results were compared with the original matrix. It is observed that clients having similar data have low individual values of difference matrix, and hence a high overall accuracy. The overall percentage error of the aggregate matrix was found to be 0.571% and 1.332%, 0.438%, and 0.533% for the respective clients.



(a) Client 1

(b) Client 2

(c) Client 3

(d) Client 4

Figure 7: Heat Maps of Difference Matrix for clients having different data

The figure 7, shows heat maps for the difference matrix obtained for the second approach of Federated Nystrom approximation, when the results of Nyström approximation were compared with the original matrix. It is observed that clients having different data have very high individual values of difference matrix, and hence an extremely low overall accuracy. The overall percentage error of the aggregate matrix was found to be 100% and 1.519%, 3.48695e+70%, 3.48695e+70%, and 1.87107e+70% for the respective clients. It is observed that client 1 has a low value of percentage error. This is because The pivot points were chosen from the matrix of client 1 and hence the data is more relevant for client 1 than any other client. Since the data of pivot points is not relevant for other clients, the percentage error of individual entries is very high (in the order of e35).

4 Conclusion

This paper was aimed to find methods to incorporate Nyström approximation for matrix completion with federated learning. A novel methodology was developed, and an operational algorithm was formulated to execute this task. It was observed that clients having similar data can be approximated with a high accuracy as compared to clients having completely different sets of data. All the theoretical aspects have been understood and worked upon and all the necessary information has been gathered. It is intended to implement this research on hardware devices using services such as Raspberry Pi.

Acknowledgment

The authors would like to place on record their deep sense of gratitude to Dr. Neeraj Jain, Assistant Professor (Senior Grade), Jaypee Institute of Information Technology, India for his generous guidance, continuous encouragement, and supervision. The authors also express their sincere gratitude to the university and faculty members for their stimulating guidance, insightful comments, and constructive suggestions throughout the present work to improve the quality of this paper.

References

[1] L. T. Nguyen, J. Kim, and B. Shim, "Low-rank matrix completion: A contemporary survey," *IEEE Access*, vol. 7, pp. 94215–94237, 2019.

[2] Z. Li, B. Ding, C. Zhang, N. Li, and J. Zhou, "Federated matrix factorization with privacy guarantee," *Proceedings of the VLDB Endowment*, vol. 15, no. 4, 2021.

[3] A. A. Abbasi, S. Moothedath, and N. Vaswani, "Fast federated low rank matrix completion," in *2023 59th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1–6, IEEE, 2023.

[4] D. Chai, L. Wang, K. Chen, and Q. Yang, "Secure federated matrix factorization," *IEEE Intelligent Systems*, vol. 36, no. 5, pp. 11–20, 2020.

[5] M. Qiao, Z. Shan, F. Liu, and W. Sun, "A fast matrix-completion-based approach for recommendation systems," *arXiv preprint arXiv:1912.00600*, 2019.

[6] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling techniques for the nystrom method," in *Artificial intelligence and statistics*, pp. 304–311, PMLR, 2009.

[7] J. Jafarov, "Survey of matrix completion algorithms," *arXiv preprint arXiv:2204.01532*, 2022.

[8] C. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," *Advances in neural information processing systems*, vol. 13, 2000.

[9] A. Flanagan, W. Oyomno, A. Grigorievskiy, K. E. Tan, S. A. Khan, and M. Ammad-Ud-Din, "Federated multi-view matrix factorization for personalized recommendations," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part II*, pp. 324–347, Springer, 2021.

[10] S. Caldas, V. Smith, and A. Talwalkar, "Federated kernelized multi-task learning," in *Proc. SysML Conf*, pp. 1–3, 2018.