Privacy Preserving Federated Matrix Factorization for Recommendation Systems

Course: Privacy Preserving Machine Learning, Lecturer: Dr. Adi Akavia ,Written by: Shir Sneh

August 2023

Abstract

This report describes a privacy-preserving federated learning algorithm for matrix factorization in recommendation systems. With this system, user privacy is protected while accurate recommendations are enabled through the use of the homomorphic properties of the Paillier encryption scheme.

1 Introduction

Recommendation systems are used to provide users with personalized suggestions and recommendations that help them discover relevant items, content, products, and services. These systems are widely used in many industries and applications, including e-commerce, streaming platforms, social media, and content websites.
As those systems become more popular and used more frequently, security concerns arise.

Privacy-preserving recommendation systems strive to balance the need for personalized recommendations with user privacy and security. By adopting a decentralized approach, federated recommendation systems address these concerns. In this context, "federated" refers to the collaboration of multiple entities, such as devices, servers, or users, to collectively generate recommendations without sharing raw data.

To accomplish this objective, we employ the technique of Matrix factorization.
Matrix factorization is a mathematical technique used to break down a matrix into multiple smaller matrices whose combination approximates the original matrix. In simpler terms, it's a method to decompose a large data matrix into smaller, more interpretable matrices that capture the underlying patterns and relationships in the data. In recommendation systems, for instance, it can be used to uncover latent factors or features that explain the observed interactions between users and items.

However, it often involves sharing user data with a central server, presenting privacy risks. To tackle this issue, a privacy-preserving federated machine learning algorithm, tailored for matrix factorization, is being introduced. This methodology facilitates model training on client devices, transmitting only the model updates to the server, thereby reducing the need to exchange raw user data.

The proposed system harnesses the homomorphic properties found in the Paillier encryption scheme. This enables computations to be performed on encrypted data, safeguarding the privacy of user data throughout the process. This report provides an in-depth overview of the system's design and implementation, its commitments to privacy and accuracy, and includes a practical usage example.

1.1 My Contribution

In the scope of this project, I've constructed an application focused on song recommendations. The recommendation system proposed here adopts a privacy-preserving federated approach. It incorporates matrix factorization and Paillier encryption to provide users with personalized song recommendations while maintaining their privacy. The primary aim is to deliver song suggestions that align with the user's preferences while upholding the security of their private data.

1.2 Related Work

Within this segment, I explore the work of others and provide an overview of my findings and insights gained from their research.

Firstly, I looked at the work of DataSourcer in the GitHub project, where a Song Recommendation system was implemented based on the "Taste profile subset" dataset that is auxiliary to the popular million songs dataset. This dataset is a freely available collection of audio features and metadata for a million contemporary popular music tracks. The protocol suggested and implemented in this project is a matrix factorization methodology based on SVD (Singular Value Decomposition). While the implementation was pretty good, it doesn’t provide privacy of the user data. when trying to integrate MPC (Secure multiparty computation) to this approach, I got low performances.

Next, I examined the research by Chai, Wang, Chen, and Yang presented in their paper titled "Secure Federated Matrix Factorization." In their study, they introduced an approach to secure matrix factorization within the context of federated machine learning, referred to as FedMF (Federated Matrix Factorization). They began by demonstrating that a distributed matrix factorization system, where users transmit gradients to the server as plaintext, can inadvertently expose users' rating details. Consequently, they developed a secure matrix factorization framework founded on homomorphic encryption principles. Their protocol served as a solid foundation for my project. I seamlessly integrated their federated matrix factorization technique into my work, implementing and employing it to compute user ratings. These ratings constitute the foundational data on which my model generates personalized song recommendations for users.

2 Preliminaries

2.1 Federated Learning

Federated learning is a unique machine learning paradigm where multiple clients, such as users or devices, collaborate to train a model while retaining their local datasets. The central server aggregates updates from individual datasets to create a global model, subsequently shared with clients for local refinements. This approach ensures data privacy as raw data remains confined to clients' devices, eliminating the need for transmission.

By adopting this method, organizations can tap into distributed data's potential while maintaining privacy. Each client contributes to model enhancement without exposing raw data to external parties, safeguarding user privacy and enabling robust model creation through diverse data sources.

2.2 Matrix Factorization

Matrix Factorization stands as a widely used method within recommendation systems. It involves breaking down a matrix that represents user-item interactions into two matrices of lower rank. These are often called the user and item matrices.

In more detail, the user matrix encapsulates latent features associated with users, while the item matrix does the same for items. These latent features could encompass various attributes or characteristics that are not explicitly captured in the original data.

The true power of matrix factorization emerges when it comes to making predictions. By taking the dot product of the latent feature vectors from the user and item matrices, it becomes possible to predict the interaction or preference of a user for a particular item. This means that even if explicit interactions between certain users and items are missing in the original data, the model can make educated guesses based on the inferred latent features.

2.3 Homomorphic Encryption

Homomorphic encryption represents an encryption method that possesses a unique capability: it enables computations to be executed on encrypted data, producing an encrypted outcome. Once decrypted, this outcome matches the result of the corresponding operations conducted on the original, unencrypted data.

Essentially, homomorphic encryption enables us to work with sensitive data while maintaining its confidentiality. It's a crucial aspect of our approach, allowing us to perform intricate computations without the need to decrypt the data at any point during the process.

2.4 Paillier Encryption

The Paillier cryptosystem operates as a public-key encryption method that possesses distinctive homomorphic attributes. In particular, it facilitates the addition of two encrypted numbers, a concept referred to as homomorphic addition. Additionally, it enables the multiplication of an encrypted number with a non-encrypted counterpart, known as homomorphic multiplication.

These homomorphic properties hold significant implications. They grant us the ability to carry out computations on encrypted data without the need for decryption. To elaborate, consider the scenario of homomorphic addition: we can add two encrypted numbers together, and upon decryption, the result matches the sum of the respective original numbers. Similarly, in the case of homomorphic multiplication, the product of an encrypted number and a non-encrypted number, when decrypted, corresponds to what would be obtained by multiplying the two original numbers.

This capability holds immense significance for our purpose of providing recommendations based on matrix factorization. It empowers us to conduct complex operations on sensitive, encrypted data while maintaining the confidentiality of the actual information. In essence, we can manipulate encrypted data to extract valuable insights without compromising the privacy and security of the raw data.

3 Algorithm

3.2 Description

The architecture of the privacy-preserving federated matrix factorization recommendation system consists of various integral components: the User, Server, GUIApp classes, and the calculation functions.

The User class embodies system users, encompassing both their actual and predicted ratings. It includes a binary mask that signifies their rated items, user identification, encrypted password, learning rate, as well as private and public keys. This class also features a method for updating user ratings.

The Server class functions as the central server within the federated learning framework. It encompasses item profiles, a learning rate, user count, and a public key. This class includes a method for updating these profiles based on user gradients.

The GUIApp class takes charge of the user interface, offering functions for user management such as login and signup. It also provides functions for data management, including loading data onto the screen and updating it as necessary.

The calculation functions consist of essential matrix factorization calculations, such as update_matrices, compute_gradient, and compute_loss. The pivotal function, update_matrices, orchestrates the iterative process involving gradient computation and updates. This process continues until convergence, as determined by a threshold reflecting the change in total loss.

3.3 Pseudocode

```
Update_matrices(users, server):
    Initialize variables and convergence threshold.
    while true do:
        initialize start time and total loss.
        for each user in users:
            compute gradient
            update song profile
        if |total loss – previous loss| < loss threshold do:
            increase local convergence.
            if we reached convergence threshold do:
                Print success message and break
        else:
            local convergence ← 0, previous loss ← total loss
        add total loss to loss array and increase iteration’s index.
```

3.4 Leakage Profile

User privacy: The actual ratings given by users remain confidential through encryption using the Paillier homomorphic encryption scheme.

Model Convergence Details: Information about whether the model has reached convergence and the number of iterations required might inadvertently hint at the complexity and diversity of user data. This information is typically not considered sensitive.

3.5 Correctness Guarantees

The algorithm's reliability stems from the inherent traits of the matrix factorization technique, a firmly established methodology within recommendation systems. This technique is known for its ability to provide accurate results in such systems. Furthermore, the attributes of the Paillier cryptosystem play a crucial role in preserving correctness. This cryptographic scheme ensures that computations carried out on encrypted data yield precise outcomes once they are decrypted.

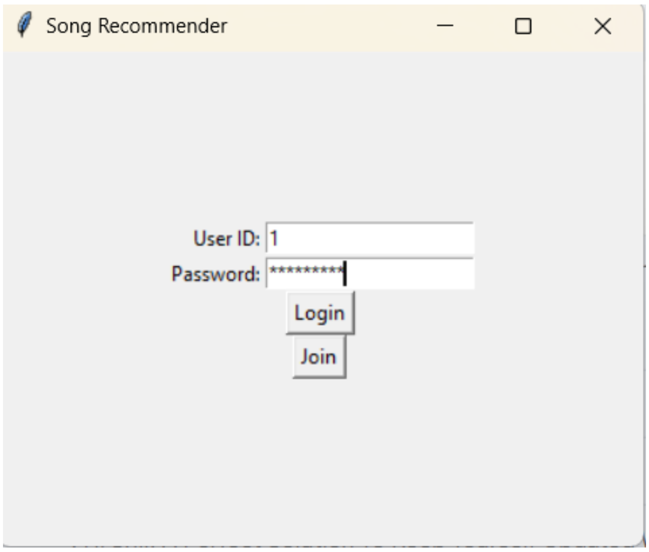
3.6 Privacy Guarantees

The algorithm employs federated learning to keep raw user data confidential while allowing model improvements to be collaboratively integrated. The Paillier cryptosystem enhances this protection by enabling encrypted computations, making the algorithm a robust choice for privacy-conscious applications.

4 System evaluation and use case example

The executed system presents itself as a straightforward application featuring a clean and user-friendly graphical user interface (GUI). Initially, there's a login screen offering the option to either log in as a registered user or sign up as a new user. Upon successful login, each user gains access to a curated list of recommended songs tailored to their preferences.

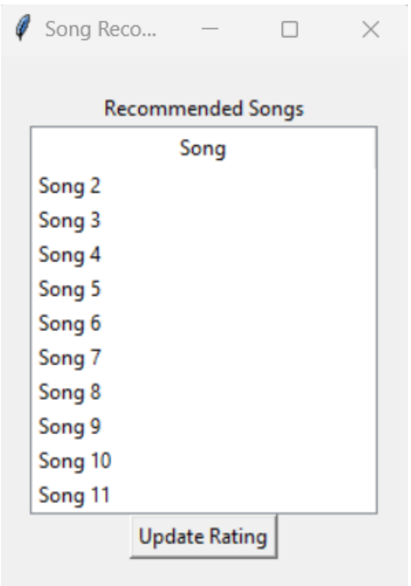
For a visual walkthrough, please refer to Figure 1, Figure 2, and figure 3 showcasing a demonstration run for a registered user. Additionally, Figure 4 provides insight into the process for a new user during the demo run.



User ID: 1

Password: *********

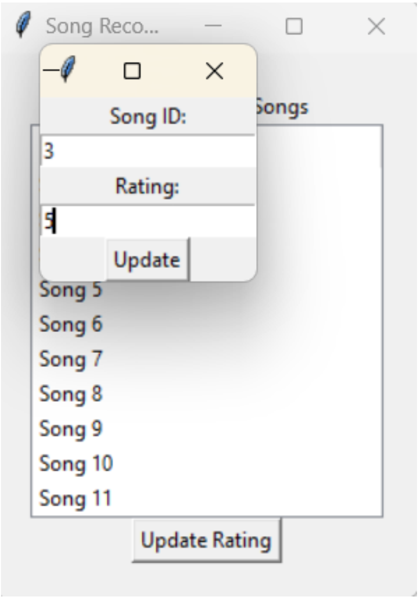
Figure 1: Login screen



Recommended Songs

Song
Song 2
Song 3
Song 4
Song 5
Song 6
Song 7
Song 8
Song 9
Song 10
Song 11

Figure 2: Recommendation Screen

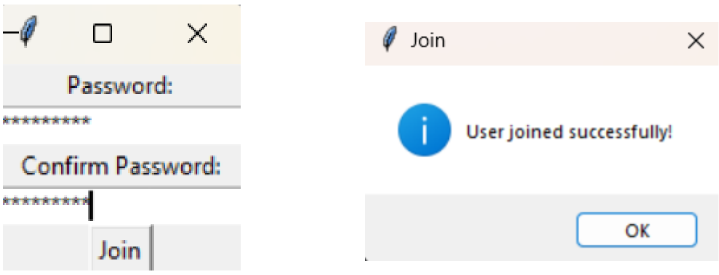


Song ID: 3

Rating: 5

Song 5
Song 6
Song 7
Song 8
Song 9
Song 10
Song 11

Figure 3: Updating Rating



Password: *********

Confirm Password: *********

User joined successfully!

Figure 4: Adding a new user

5 Conclusion

This paper introduced an adept federated learning algorithm for recommendation systems, which effectively merges federated learning with the Paillier cryptosystem to establish robust user privacy. The approach revolves around conducting computations on encrypted data, thereby upholding the confidentiality of raw user data.

My executed model effectively showcased the algorithm's prowess in furnishing precise recommendations without compromising user privacy. Nevertheless, there exist opportunities for refinement, including the exploration of alternative privacy methods to tackle the issue of information leakage associated with the disclosure of Model Convergence Details.

6 References

1. Di Chai, Leye Wang, Kai Chen and Qiang Yang. Secure Federated Matrix Factorization. 2019.
2. DataSorcerer, GitHub project: Music-Recommendation-System. 2018.
3. Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. 2018.
4. Muhammad Ammad-ud-din, Elena Ivannikova, Suleiman A. Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. Federated collaborative filtering for privacy-preserving personalized recommendation system. 2019.
5. Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. Applying differential privacy to matrix factorization. 2015.