Universidad Autónoma de Madrid

Escuela Politécnica Superior

25|26

Trabajo fin de grado

Privacidad Diferencial en Sistemas de Recomendación Federados basados en Factorización de Matrices

Álvaro de Santos de la Pedraja

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C\Francisco Tomás y Valiente nº 11

excelencia Campus Internacional UAM CSIC

www.uam.es

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Privacidad Diferencial en Sistemas de Recomendación Federados basados en Factorización de Matrices

Autor: Álvaro de Santos de la Pedraja
Tutor: Alejandro Bellogin Kouki

mayo 2026

Todos los derechos reservados.
Queda prohibida, salvo excepción prevista en la Ley, cualquier
forma de reproducción, distribución comunicación pública
y transformación de esta obra sin contar con la autorización
de los titulares de la propiedad intelectual.
La infracción de los derechos mencionados puede ser constitutiva de
delito contra la propiedad intelectual *(arts. 270 y sgts. del Código Penal)*.

DERECHOS RESERVADOS
© 19 de Mayo de 2026 por UNIVERSIDAD AUTÓNOMA DE MADRID
Francisco Tomás y Valiente, n^o 1
Madrid, 28049
Spain

Álvaro de Santos de la Pedraja
Privacidad Diferencial en Sistemas de Recomendación Federados basados en Factorización de Matrices

Álvaro de Santos de la Pedraja
C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

Agradecimientos

Agradecido a mi familia, por apoyarme siempre, y a mi tutor Alejandro Bellogin Kouki, por su dirección y atención a lo largo de todo el trabajo.

Resumen

Los sistemas de recomendación tienen como objetivo la personalización de las recomendaciones a nivel de usuario mediante la extracción de patrones comunes en su comportamiento, desempeñando un papel fundamental en los servicios digitales actuales. En particular, los modelos basados en factorización de matrices se han consolidado como una técnica estándar en el campo, impulsados por sus ventajas en robustez frente a la dispersión de los datos, inherente al ámbito de la recomendación, y su escalabilidad. Sin embargo, el auge de los sistemas de recomendación también plantea graves problemas de privacidad: manejan datos altamente sensibles que describen comportamientos, hábitos y preferencias de los usuarios. Se han propuesto diversos paradigmas como contramedida a esta dificultad. Por un lado, el aprendizaje federado constituye un esquema que pretende distribuir el entrenamiento del sistema, de modo que este se realiza de forma descentralizada en los dispositivos de los clientes. Por otro lado, la privacidad diferencial establece un marco matemático que proporciona garantías cuantificables respecto a la información que un atacante puede inferir del modelo.

En este trabajo de fin de grado se lleva a cabo el diseño, implementación y evaluación de un sistema de recomendación federado basado en factorización de matrices con privacidad diferencial. Para ello, se realiza un estudio exhaustivo de los conceptos requeridos para fundamentar el modelo propuesto en lo concerniente a sistemas de recomendación, aprendizaje federado y privacidad diferencial. El diseño e implementación del modelo se plantea en tres fases incrementales, que transitan desde un esquema de factorización de matrices centralizado hasta un modelo federado con garantías cuantificables de privacidad. Estas se obtienen mediante un mecanismo gaussiano bajo un esquema de privacidad diferencial central, cuyo presupuesto se contabiliza a través del formalismo de Rényi. El rendimiento del modelo se evalúa sobre dos conjuntos de datos de la familia *MovieLens*, analizando empíricamente el compromiso entre la federación del sistema y las garantías de privacidad introducidas, por un lado, y la utilidad final del modelo, por otro. Los resultados obtenidos muestran que el modelo propuesto preserva niveles de utilidad aceptables bajo garantías de privacidad razonables, lo que lo configura como una solución viable para el despliegue de sistemas de recomendación con garantías de privacidad cuantificables.

Palabras clave

Sistema de recomendación, factorización de matrices, aprendizaje federado, privacidad diferencial, ruido gaussiano, Rényi, *MovieLens*

Abstract

Recommender systems aim to personalize recommendations at the user level by extracting common patterns from user behavior, playing a fundamental role in today's digital services. In particular, matrix factorization models have become established as a standard technique in the field, driven by their robustness against data sparsity, inherent to the recommendation domain, and their scalability. However, the rise of recommender systems also raises serious privacy concerns: they handle highly sensitive data describing user behaviors, habits, and preferences. Several paradigms have been proposed to address this challenge. On the one hand, federated learning constitutes a framework that distributes the training of the system, such that it is carried out in a decentralized manner on the clients' devices. On the other hand, differential privacy establishes a mathematical framework that provides quantifiable guarantees on the information an attacker can infer from the model.

This bachelor's thesis presents the design, implementation, and evaluation of a federated recommender system based on matrix factorization with differential privacy. To this end, a comprehensive study is carried out of the concepts required to underpin the proposed model, covering recommender systems, federated learning, and differential privacy. The design and implementation of the model is structured in three incremental phases, transitioning from a centralized matrix factorization scheme to a federated model with quantifiable privacy guarantees. These guarantees are obtained through a Gaussian mechanism under a central differential privacy scheme, whose privacy budget is accounted for via the Rényi differential privacy formalism. The performance of the model is evaluated on two datasets from the *MovieLens* family, empirically analyzing the trade-off between the federation of the system and the privacy guarantees introduced, on the one hand, and the final utility of the model, on the other. The results obtained show that the proposed model preserves acceptable utility levels under reasonable privacy guarantees, establishing it as a viable solution for deploying recommender systems with quantifiable privacy guarantees.

Keywords

Recommender systems, matrix factorization, federated learning, differential privacy, Gaussian noise, Rényi, *MovieLens*

Índice

1 Motivación y Objetivos **1**

2 Estado del Arte **3**

2.1 Sistemas de Recomendación . . . 3

2.1.1 Tipología . . . 4

2.1.2 Formulación matemática y factorización de matrices . . . 4

2.1.3 El problema de la privacidad . . . 5

2.1.4 Implementación de sistemas de recomendación . . . 5

2.2 Aprendizaje federado . . . 6

2.2.1 Fundamentos del aprendizaje federado . . . 6

2.2.2 Sistemas de recomendación en entornos federados . . . 7

2.2.3 Modelos de amenaza en aprendizaje federado . . . 8

2.3 Privacidad diferencial . . . 9

2.3.1 Fundamentos de la privacidad diferencial . . . 9

2.3.2 Privacidad diferencial en sistemas de recomendación federados . . . 15

2.3.3 Implementación de privacidad diferencial . . . 15

2.4 Evaluación . . . 16

3 Diseño e Implementación **17**

3.1 Diseño general . . . 17

3.1.1 Objetivos . . . 17

3.1.2 Desarrollo . . . 18

3.1.3 Arquitectura . . . 18

3.2 Fase I: Factorización de matrices centralizada . . . 18

3.2.1 Formulación básica y optimizaciones . . . 19

3.2.2 Algoritmo de entrenamiento . . . 20

3.3 Fase II: Extensión al aprendizaje federado . . . 21

3.3.1 Configuración del sistema federado . . . 21

3.3.2 Algoritmo federado propuesto . . . 22

3.4 Fase III: Incorporación de privacidad diferencial central . . . 24

3.4.1 Modelo de amenaza . . . 24

3.4.2 Adición de privacidad diferencial . . . 25

3.4.3 Cálculo del presupuesto de privacidad . . . 27

4 Pruebas y Resultados **29**
4.1 Configuración Experimental 29
4.1.1 Conjuntos de datos 30
4.1.2 Factorización de matrices centralizado 30
4.2 Efecto de la federación 32
4.2.1 Selección de épocas locales 32
4.2.2 Comparación con el modelo centralizado 33
4.2.3 Efecto del número de clientes 33
4.3 Efecto de la privacidad diferencial 34
4.3.1 Comparación de utilidad 35
4.3.2 Compromiso privacidad–utilidad 36
4.4 Discusión 37
5 Conclusiones y Trabajo Futuro **39**
Bibliografía **43**
Acrónimos **45**
Apéndices **47**
A Tablas Complementarias **49**
A.1 Estadísticas de los datos 49
A.2 Hiperparámetros 49
A.3 Resultados completos 49
B Figuras Complementarias **55**

Listas

Lista de algoritmos

3.1 Entrenamiento de MF mediante SGD. 21
3.2 Entrenamiento local en el cliente. 22
3.3 Coordinación y agregación en el servidor. 24
3.4 Muestreo probabilístico de clientes. 25
3.5 Agregación federada con ruido gaussiano. 26

Lista de ecuaciones

2.1 Factorización de matrices base 5
2.2 Agregación federada 6
2.3 Factorización de matrices federada 8
2.4 Variable aleatoria de pérdida de privacidad 9
2.5 Privacidad diferencial 10
2.6 Divergencia de Rényi 10
2.7 Privacidad diferencial de Rényi 10
2.8 Conversión de privacidad diferencial de Rényi a privacidad diferencial 10
2.9 Composición de mecanismos aleatorios 11
2.10 Sensibilidad de una función 12
2.11 Mecanismo gaussiano 12
2.12 Privacidad diferencial de Rényi para el mecanismo gaussiano 13
2.13 Invarianza a traslaciones de la divergencia de Rényi 13
2.14 Amplificación de la privacidad diferencial de Rényi por submuestreo 14
2.15 Raíz del error cuadrático medio 16

3.1 Factorización matricial con mejoras 20
3.2 Función de pérdida de factorización matricial 20
3.3 Actualización generada por un cliente 23
3.4 Función determinista de agregación federada 23
3.5 Actualización federada del modelo 23
3.6 Cota de la sensibilidad en agregación federada 26
3.7 Privacidad diferencial del mecanismo de agregación 27

3.8 Privacidad diferencial del sistema federado 27
3.9 Presupuesto de privacidad en función del multiplicador de ruido 28

Lista de figuras

2.1 Esquema del proceso de FL 7

3.1 Arquitectura general del sistema implementado 19

4.1 Convergencia del modelo de factorización de matrices en el entrenamiento 31
4.2 Efecto del número de clientes en la federación 34
4.3 Efecto del nivel de ruido sobre la utilidad del sistema 36
4.4 Compromiso privacidad-utilidad 36

B.1 Convergencia del modelo federado en el entrenamiento 55

Lista de tablas

4.1 Estadísticas de los datasets de la familia *Movielens* 30
4.2 Resultados para el modelo de factorización de matrices 31
4.3 Resultados para el modelo de factorización de matrices federado 33
4.4 Configuraciones según el escenario 38

A.1 Tamaño de los splits para los datasets ml-100k y ml-1m 49
A.2 Búsqueda de hiperparámetros para el modelo de factorización de matrices centralizado 50
A.3 Búsqueda de épocas locales para el modelo federado 51
A.4 Efecto del número de clientes de la federación 51
A.5 Resultados para el modelo federado con privacidad diferencial en ml-100k 52
A.6 Resultados para el modelo federado con privacidad diferencial en ml-1m 53

1

Motivación y Objetivos

En las últimas décadas, los sistemas de recomendación (*Recommender Systems*, RS) se han convertido en una pieza fundamental de los servicios digitales, mediando prácticamente toda interacción de los usuarios. Estos sistemas desempeñan un papel central en la personalización de las recomendaciones para cada usuario, lo cual efectúan mediante la extracción de patrones comunes en su comportamiento. Un punto de inflexión en la popularización de los RS fue el *Netflix Prize* [1], una competición organizada por *Netflix* entre 2006 y 2009 en la que los participantes trataban de superar la precisión del sistema de recomendación de la compañía. Esta competición supuso la consolidación de los modelos basados en factorización de matrices (*Matrix Factorization*, MF) como una técnica estándar en el campo. En la actualidad, los RS se han establecido como un componente clave de la economía digital, con aplicaciones que abarcan desde el *streaming* hasta el comercio electrónico y las redes sociales.

No obstante, la consolidación de los RS ha planteado problemas críticos de privacidad. Los datos sobre los que operan tienen un carácter inherentemente sensible, ya que describen el comportamiento, los hábitos y las preferencias de los usuarios. En 2008 se demostró [2] que era posible inferir la identidad de los usuarios en el conjunto de datos perturbado del *Netflix Prize*, lo que llevó a la cancelación de la segunda edición de la competición en 2010. Posteriormente, diversas regulaciones como el Reglamento General de Protección de Datos (*General Data Protection Regulation*, GDPR) [3] han establecido marcos formales para la protección de la privacidad de los usuarios. Estos factores han motivado el desarrollo de técnicas que permitan construir RS sin comprometer la privacidad de los usuarios.

En respuesta a esta necesidad han emergido principalmente dos paradigmas. Por un lado, el aprendizaje federado (*Federated Learning*, FL), popularizado por *Google* [4], plantea el entrenamiento del modelo como un esquema distribuido en el que el entrenamiento ocurre de manera local en los dispositivos, sin necesidad de centralizar los datos. Por otro lado, la privacidad diferencial (*Differential Privacy*, DP), formalizada por [5], proporciona garantías cuantificables sobre la información que un atacante puede inferir del modelo, mediante la adición de ruido controlado durante el entrenamiento. La combinación de ambos paradigmas aplicada a sistemas de recomendación constituye el foco de este trabajo de fin de grado.

El presente trabajo tiene como objetivo principal el diseño, implementación y evaluación de un sistema de recomendación federado con privacidad diferencial, con los siguientes objetivos específicos:

O1 Implementar un modelo de factorización de matrices centralizado que sirva como referencia para la evaluación de los modelos posteriores.

O2 Extender el modelo centralizado al paradigma de aprendizaje federado, adaptando el proceso de entrenamiento a un esquema distribuido en el que los datos permanecen localmente en cada cliente.

O3 Incorporar mecanismos de privacidad diferencial al sistema federado, proporcionando garantías cuantificables sobre la privacidad de las contribuciones de cada cliente.

O4. Evaluar el rendimiento del sistema propuesto sobre conjuntos de datos estándar de la literatura, comparando los distintos modelos implementados en términos de utilidad.

O5 Analizar empíricamente el efecto del número de clientes de la federación sobre la utilidad del sistema, tanto en ausencia como en presencia de privacidad diferencial.

O6 Estudiar el compromiso existente entre privacidad y utilidad en función del nivel de ruido introducido por el mecanismo diferencial.

Con el fin de alcanzar los objetivos descritos, el presente documento se estructura de la siguiente manera. El Capítulo 1 introduce la motivación, los objetivos y la estructura del trabajo. El Capítulo 2 presenta los conceptos necesarios para el desarrollo de los capítulos posteriores. En primer lugar, establece el marco de los sistemas de recomendación y ubica los modelos de factorización de matrices dentro de él. En segundo lugar, explora el esquema federado: desde una perspectiva general hasta su aplicación a los RS, y en particular a la MF. En tercer lugar, desarrolla el marco teórico de la privacidad diferencial, con un carácter exhaustivo demostrando los resultados presentados, y situándola finalmente en el contexto de los sistemas de recomendación federados. Por último, introduce los conceptos de evaluación en aprendizaje supervisado y los RS necesarios para los experimentos del presente trabajo.

El Capítulo 3 presenta el diseño e implementación del sistema propuesto. El capítulo comienza estableciendo los objetivos y la arquitectura del modelo, así como la estructura incremental de la implementación en tres fases. La primera fase construye el modelo de MF centralizado como referencia, incorporando algunas mejoras respecto al MF básico. La segunda fase extiende el entrenamiento del modelo de MF a un esquema federado. La tercera fase introduce la DP y detalla el mecanismo para el cálculo del presupuesto de privacidad. El Capítulo 4 presenta los resultados obtenidos al ejecutar diversas pruebas de evaluación sobre el sistema propuesto. En primer lugar, describe los conjuntos de datos sobre los que se realiza la evaluación y fija los hiperparámetros del modelo de referencia MF. A continuación, analiza el efecto sobre la utilidad de incorporar el esquema federado y estudia el impacto de la variación del número de clientes. Finalmente, examina el compromiso privacidad-utilidad en función del nivel de ruido introducido por el mecanismo diferencial y el presupuesto de privacidad. El Capítulo 5 recoge las conclusiones del trabajo y propone algunas líneas de investigación futura.

2

Estado del Arte

Este capítulo presenta el estado del arte relativo a los sistemas de recomendación, con especial atención a los enfoques basados en la factorización de matrices, el aprendizaje federado y las técnicas de privacidad diferencial. El objetivo es situar el trabajo en el contexto de las principales aportaciones existentes, analizando los modelos y metodologías más relevantes propuestos en la literatura.

El análisis conjunto de estos enfoques permite identificar tanto sus fortalezas como sus limitaciones, especialmente en lo relativo a las garantías de privacidad. Este estudio justifica la necesidad de integrar técnicas de FL y DP en modelos de recomendación basados en la MF, sentando las bases teóricas para el desarrollo presentado en los capítulos posteriores.

2.1. Sistemas de Recomendación

Los sistemas de recomendación corresponden a una clase de sistemas de información que facilitan la toma de decisiones de un usuario frente a un conjunto de objetos denominados ítems [6]. Para ello, el sistema realiza una predicción sobre las preferencias del usuario por un determinado elemento a partir de un conjunto de información establecido previamente.

Formalmente, el sistema trata de estimar una función que, dados un usuario y un ítem, predice una valoración explícita (por ejemplo, una puntuación) o implícita (como una interacción). Desde este punto de vista, los RS se sitúan dentro del aprendizaje automático, y particularmente dentro del aprendizaje supervisado. Debido a la cantidad de información disponible y la necesidad de personalización en múltiples ámbitos, su relevancia ha crecido de manera exponencial en las ultimas décadas. No obstante, el problema de recomendación plantea diversos problemas significativos como la dispersión de los datos, la escalabilidad, y la gestión de la información sensible.

Consecuentemente, la investigación en este contexto ha dado lugar a una amplia variedad de enfoques y modelos. En las siguientes secciones se revisan los principales enfoques existentes, con especial hincapié en el filtrado colaborativo, que supone el marco del presente trabajo.

2.1.1. Tipología

La clasificación de los RS se puede enfocar desde distintos puntos de vista. En este caso, se han diferenciado en función del tipo de información que emplean y de las hipótesis que realizan sobre los usuarios. La clasificación es la siguiente:

- Por un lado, los RS **basados en contenido** [6, pp. 251–299] modelan las preferencias de un usuario a partir de las características explícitas de los ítems. La comparación se realiza entre el perfil del usuario y la descripción de los elementos. Estos métodos permiten recomendar ítems de manera personalizada incluso con información inexistente de otros usuarios, sin embargo, requieren la disponibilidad de descriptores de calidad, lo que limita su aplicabilidad.
- Por otro lado, los RS **basados en filtrado colaborativo** [6, pp. 39–141] utilizan las interacciones entre usuarios e ítems, sin necesidad de información adicional sobre el contenido. Estos métodos operan bajo la hipótesis de que usuarios con comportamientos similares tienden a poseer preferencias similares.
- Finalmente, los **sistemas híbridos** [6, pp. 1–37] combinan información, integrando enfoques basados en contenido y filtrado colaborativo para mitigar las limitaciones de cada uno de ellos. Estos modelos se suelen aplicar en escenarios complejos con mejores resultados, pero resultan mas costosos de implementar y mantener.

En el presente trabajo se adopta el paradigma del filtrado colaborativo. En concreto, se presta atención a los modelos que representan usuarios e ítems mediante factores latentes, los cuales se introducen en la siguiente sección.

2.1.2. Formulación matemática y factorización de matrices

El problema de recomendación se plantea como la estimación de preferencias a partir de una matriz de interacciones parcialmente observada. Sea $R \in \mathbb{R}^{m \times n}$ la matriz usuario-ítem, donde cada entrada r_{ui} representa la valoración del usuario u respecto al ítem i. En la práctica, cada usuario solo interactúa con un número limitado de ítems, por lo que la matriz es altamente dispersa. El objetivo del sistema consiste en predecir las entradas no observadas de R. Este problema puede interpretarse como una tarea de *matrix completion* en la que se busca reconstruir la estructura de la matriz a partir de un subconjunto limitado de observaciones.

El filtrado colaborativo engloba diversos métodos. Los métodos basados en vecinos explotan la similitud directa entre usuarios o ítems, pero presentan limitaciones en escalabilidad. Los métodos basados en factores latentes tratan de paliar algunas de estas dificultades. En particular, la **factorización de matrices** constituye una de las técnicas más eficaces en RS. La idea básica del procedimiento trata de aproximar la matriz de interacciones R mediante el producto de dos matrices de menor rango,

$P \in \mathbb{R}^{m \times k}$ y $Q \in \mathbb{R}^{n \times k}$, de forma que:

$$R \approx PQ^{\top}. \tag{2.1}$$

Cada fila de las matrices representa una proyección latente del usuario o del ítem en un espacio de dimensión reducida $k \ll \min(m, n)$. Las representaciones permiten al modelo capturar patrones globales a partir de observaciones locales. Desde el punto de vista del aprendizaje, el modelo trata de minimizar una función de pérdida sobre las entradas observadas de R. Los modelos de MF exhiben múltiples ventajas en cuanto a mayor robustez frente a la dispersión y mejor escalabilidad. En consecuencia, la MF constituye el modelo base considerado para este trabajo.

2.1.3. El problema de la privacidad

Los RS se basan en datos que reflejan directamente las preferencias y comportamientos de los usuarios, lo que plantea desafíos de privacidad. La centralización de esta información durante el entrenamiento de modelos conlleva riesgos asociados a filtraciones, accesos no autorizados y ataques de inferencia sobre los datos subyacentes. Más allá de la exposición directa de los datos, los modelos entrenados pueden revelar información sensible a través de sus parámetros o predicciones, dando lugar a ataques específicos como los de inferencia de pertenencia [7], en los que un atacante trata de determinar si un determinado usuario formó parte del conjunto de entrenamiento.

Consecuentemente, se requieren enfoques que limiten la exposición de los datos y que ofrezcan garantías formales de privacidad. Estas consideraciones han motivado el uso de paradigmas como el FL y la DP [8–12]. En las siguientes secciones se introducen ambos enfoques en el contexto de modelos basados en la MF.

2.1.4. Implementación de sistemas de recomendación

La implementación práctica de modelos de RS basados en factorización de matrices requiere el uso de librerías especializadas. En la actualidad, los principales marcos de desarrollo incluyen `TensorFlow` [13], `PyTorch` [14] y `JAX` [15], los cuales proporcionan herramientas eficientes para la definición y entrenamiento de modelos de aprendizaje. Existen además librerías específicas para sistemas de recomendación, como `Surprise` [16] o `LensKit` [17], orientadas a la implementación rápida de modelos de filtrado colaborativo, aunque su flexibilidad es limitada cuando se requiere extender los modelos a esquemas distribuidos o incorporar mecanismos de privacidad. En particular, `PyTorch` se ha consolidado como uno de los entornos más utilizados en investigación, debido a su flexibilidad, y por tanto, constituye la elección en el presente trabajo para la implementación.

2.2. Aprendizaje federado

El aprendizaje federado redefine el proceso de entrenamiento distribuido al eliminar la necesidad de centralizar los datos, trasladando el cómputo a los dispositivos que los generan. En esta sección se analizan sus fundamentos, los principales desafíos asociados, su aplicación al entrenamiento de sistemas de recomendación y el concepto de modelo de amenaza en su contexto.

2.2.1. Fundamentos del aprendizaje federado

El FL es un paradigma de aprendizaje automático distribuido que consiste en el entrenamiento de modelos globales a partir de los datos de entidades (o dispositivos) que permanecen locales en los mismos [18]. El FL se basa, por tanto, en una arquitectura **cliente-servidor** en la que el proceso de entrenamiento se desarrolla de forma iterativa siguiendo los siguientes pasos:

(a) El servidor mantiene un modelo global.

(b) El servidor distribuye el modelo global a los clientes.

(c) Cada cliente utiliza sus datos locales para realizar actualizaciones del modelo recibido.

(d) Los clientes envían sus actualizaciones al servidor y el servidor combina las contribuciones de todos los clientes, obteniendo una versión actualizada del modelo global.

Este proceso se repite hasta obtener el modelo entrenado. La Figura 2.1 muestra una representación del proceso iterativo.

Uno de los algoritmos de combinación de las contribuciones mas comunes es el **agregación federada** (*Federated Averaging*, FedAvg) [4]. Este consiste en combinar los modelos entrenados localmente por los clientes mediante una agregación ponderada de sus parámetros. Denotando por $\Delta \mathbf{w}^{(c)}$ las actualizaciones aprendidas por el cliente c, el modelo global se actualiza según:

$$\Delta \mathbf{w}_{t+1} = \sum_{c \in \mathcal{C}} \frac{n_c}{n} \Delta \mathbf{w}^{(c)}, \tag{2.2}$$

donde n_c representa el número de muestras utilizadas en el entrenamiento local del cliente c y $n = \sum_{c \in \mathcal{C}} n_c$, $\mathcal{C}$ siendo el conjunto de clientes participantes en la actualización de los parámetros.

No obstante, el FL también presenta múltiples problemas que no ocurren en el aprendizaje centralizado. En primer lugar, los datos locales de cada cliente reflejan únicamente el comportamiento de un subconjunto reducido de usuarios, y por tanto no son representativos de la distribución de la población. Por otro lado, la cantidad de datos locales es altamente variable. Además, aunque el FL reduce la exposición directa de los datos, no proporciona en sí mismo garantías de privacidad, ya que las actualizaciones compartidas resultan vulnerables a ataques de inferencia y, por lo tanto, pueden filtrar información sensible.

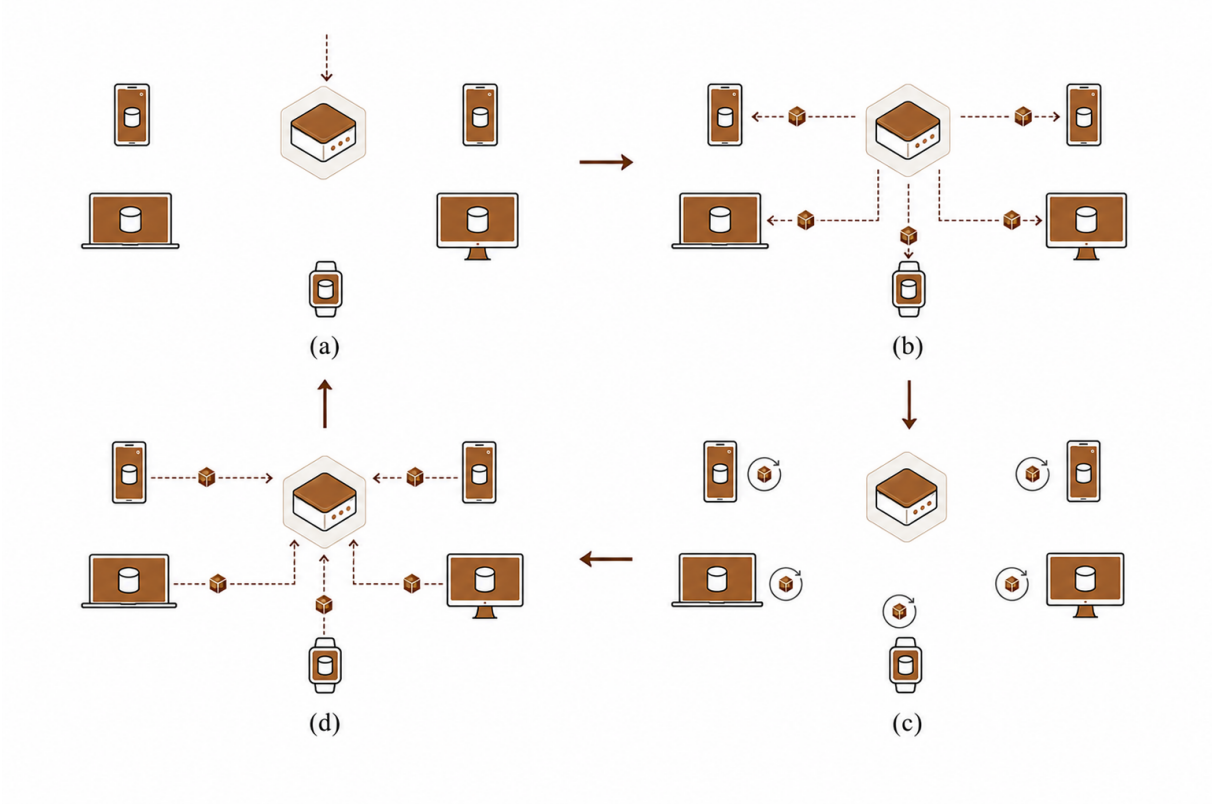


Figura 2.1: Esquema iterativo del proceso de FL: (a) se inicializa el modelo global, (b) el modelo se distribuye a los clientes (c) cada cliente entrena localmente con sus datos privados, (d) los clientes envían sus actualizaciones al servidor y el servidor agrega las contribuciones. Figura adaptada de [19].

Para paliar estos riesgos, se han propuesto algunas técnicas dentro del marco del FL. Por un lado, la encriptación homomórfica permite realizar operaciones sobre datos o actualizaciones cifradas sin necesidad de descifrarlas [18]. No obstante, este enfoque implica un elevado coste computacional y de comunicación. Por otro lado, la DP introduce un marco formal para acotar la información que un atacante es capaz de inferir sobre las contribuciones de los clientes. Este trabajo se enfoca en este último.

2.2.2. Sistemas de recomendación en entornos federados

La aplicación del FL a RS es natural debido a la estructura del problema de recomendación: las interacciones usuario-ítem se encuentran inherentemente distribuidas y contienen información altamente sensible. Los sistemas de recomendación federados (*Federated Recommender Systems*, FRS) suelen asumir una partición de los datos a nivel de usuario o grupo de usuarios, donde cada cliente dispone únicamente de las interacciones de los usuarios que tiene asignados. En el caso de la MF las preferencias de cada usuario pueden representarse mediante factores latentes que se aprenden localmente. Por otro lado, los ítems requieren de una dependencia compartida por lo que se debe realizar un aco-

plamiento entre los modelos locales de los clientes. En este contexto, la matriz de interacciones global R puede interpretarse como la agregación de los vectores de interacción individuales $\{\mathbf{r}_u\}_{u=1}^{m}$, donde cada $\mathbf{r}_u$ recoge las interacciones del usuario u con el conjunto de ítems. De forma análoga, la matriz de factores latentes de usuarios P se construye como la unión de los vectores latentes individuales $\{\mathbf{p}_u\}_{u=1}^{m}$. Bajo este planteamiento, cada cliente modela localmente las interacciones de los usuarios que tiene asignados mediante una aproximación de la forma:

$$\mathbf{r}_u \approx \mathbf{p}_u Q^\top, \tag{2.3}$$

donde $\mathbf{p}_u$ es aprendido de manera local por el cliente, mientras que la matriz Q, que contiene la información latente asociada a los ítems, se aprende y comparte de forma agregada a nivel global.

Diversos trabajos [20, 21] muestran que, aunque los FRS pueden aproximar el rendimiento de los sistemas centralizados, la descentralización en el entrenamiento conlleva una pérdida en la precisión de las predicciones. Como consecuencia, conviene analizar el compromiso que existe entre utilidad y privacidad.

2.2.3. Modelos de amenaza en aprendizaje federado

En el contexto del FL, la especificación del modelo de amenaza concierne la identificación de las entidades maliciosas en el ámbito en el que se despliega el modelo, y qué información tienen a su disposición. La literatura distingue principalmente entre dos tipos de atacantes [18].

El primero es el atacante **honesto-pero-curioso** o pasivo. Este atacante ejecuta el protocolo correctamente pero intenta inferir información privada a partir de los mensajes que observa durante el proceso de entrenamiento. En el contexto del FL, como ejemplo, un servidor honesto-pero-curioso agrega correctamente las actualizaciones de los clientes pero podría intentar extraer información sobre los datos locales de cada usuario observando dichas actualizaciones.

El segundo es el atacante **malicioso** o activo. Este atacante puede desviarse arbitrariamente del protocolo, modificar las actualizaciones recibidas, enviar mensajes falsos o coordinar ataques con otros participantes. Este modelo de amenaza es considerablemente más fuerte y queda fuera del alcance de este trabajo.

Concretar el modelo de amenaza de un sistema es crítico para distinguir qué garantías de seguridad es capaz de ofrecer dicho sistema y, consecuentemente, entender sus limitaciones. La especificación concreta del modelo de amenaza adoptado en este trabajo se detalla en el siguiente capítulo.

2.3. Privacidad diferencial

La privacidad diferencial proporciona un marco matemático [22,23] para controlar de forma explícita la información que un algoritmo puede revelar sobre contribuciones individuales. Su formulación resulta especialmente relevante en algoritmos iterativos donde la pérdida de privacidad se acumula a lo largo del tiempo. En esta sección se desarrollan sus fundamentos y se estudia su integración en sistemas de recomendación federados.

2.3.1. Fundamentos de la privacidad diferencial

La propiedad esencial de la DP es que la presencia o ausencia de una única contribución individual no debe modificar de manera significativa la distribución de salida del algoritmo.

Definición 2.1 (Mecanismo aleatorio)**.** Un *mecanismo* es una aplicación aleatoria $\mathcal{M} : \mathcal{D} \to \mathcal{Y}$ donde $\mathcal{D}$ es el espacio de conjuntos de datos y $\mathcal{Y}$ el espacio de salidas. Para cada $D \in \mathcal{D}$, el mecanismo induce una distribución de probabilidad sobre $\mathcal{Y}$.

En la práctica, los mecanismos de DP suelen adoptar la forma $\mathcal{M}(D) = f(D) + \mathcal{N}$ donde f es una función determinista del conjunto de datos y $\mathcal{N}$ es una variable aleatoria de ruido calibrado. En particular, el mecanismo aleatorio definido en este trabajo se define de tal manera.

Definición 2.2 (Conjuntos de datos adyacentes)**.** Dos conjuntos de datos $D, D' \in \mathcal{D}$ se dicen *adyacentes*, y se denota $D \sim D'$, si difieren en una única contribución individual.

La noción de DP se basa en comparar las distribuciones inducidas por el mecanismo sobre conjuntos de datos adyacentes. Para formalizar esta comparación se introduce la variable aleatoria de pérdida de privacidad.

Definición 2.3 (Variable aleatoria de pérdida de privacidad)**.** Sea $\mathcal{M}$ un mecanismo y $D \sim D'$. Para una salida $y \in \mathcal{Y}$, se define la *variable aleatoria de pérdida de privacidad* como:

$$Z(y) = \log \frac{\mathbb{P}[\mathcal{M}(D) = y]}{\mathbb{P}[\mathcal{M}(D') = y]}, \tag{2.4}$$

donde $\mathbb{P}[\mathcal{M}(D) = y]$ denota la densidad de la distribución inducida por el mecanismo aplicado al conjunto de datos D, evaluada en y.

La variable cuantifica cuánto cambia la verosimilitud de una salida concreta cuando se modifica una única contribución individual en el conjunto de datos. El valor de Z será mas próximo a cero cuando la proporción entre ambas distribuciones sea mas cercana a uno, es decir, cuando las densidades tomen valores similares.

Definición 2.4 (Privacidad diferencial)**.** Un mecanismo $\mathcal{M}$ satisface (ε, δ)-*privacidad diferencial* si para todo par de conjuntos de datos adyacentes $D \sim D'$ se cumple:

$$\mathbb{P}_{y\sim\mathcal{M}(D)}\big[Z(y) > \varepsilon\big] \leq \delta. \tag{2.5}$$

Esta formulación probabilística expresa que, salvo con probabilidad δ, la pérdida de privacidad está acotada por ε. El análisis directo de esta definición bajo composiciones repetidas de mecanismos resulta complejo. Por este motivo, en este trabajo se emplea la privacidad diferencial de Rényi (*Rényi Differential Privacy*, RDP) como medida principal de privacidad, ya que permite una contabilidad precisa y analíticamente manejable de la pérdida de privacidad en algoritmos iterativos. Esta medida de privacidad alternativa a la medida común fue introducida por [24].

Definición 2.5 (Divergencia de Rényi)**.** Sean P y Q dos distribuciones de probabilidad definidas sobre el mismo espacio y sea $\alpha > 1$. La *divergencia de Rényi de orden* α se define como:

$$D_\alpha(P \,\|\, Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_{y\sim Q}\left[\left(\frac{\mathbb{P}[P = y]}{\mathbb{P}[Q = y]}\right)^\alpha\right], \tag{2.6}$$

donde $\mathbb{P}[P = y]$ y $\mathbb{P}[Q = y]$ son las densidades de P y Q en y, respectivamente.

La divergencia de Rényi constituye una medida de distancia entre dos distribuciones establecida fuertemente en el ámbito de la probabilidad. En el contexto de la DP permite definir el siguiente concepto de privacidad.

Definición 2.6 (Privacidad diferencial de Rényi)**.** Un mecanismo $\mathcal{M}$ satisface ε-*privacidad diferencial de Rényi* de orden $\alpha > 1$ si para todo par de conjuntos de datos adyacentes $D \sim D'$ se cumple:

$$D_\alpha\big(\mathcal{M}(D) \,\|\, \mathcal{M}(D')\big) \leq \varepsilon. \tag{2.7}$$

Esta definición impone una cota uniforme sobre los momentos de la variable de pérdida de privacidad. La razón por la que se desarrolló esta medida de privacidad diferencial particularmente es por la facilidad de conversión que posee con la privacidad diferencial usual.

Teorema 2.7. *Sea $\mathcal{M}$ un mecanismo aleatorio que satisface $\varepsilon(\alpha)$-RDP de orden $\alpha > 1$. Entonces, para todo $\delta \in (0, 1)$, el mecanismo $\mathcal{M}$ satisface $(\varepsilon(\delta), \delta)$-DP, donde:*

$$\varepsilon(\delta) = \varepsilon(\alpha) + \frac{1}{\alpha - 1} \log \frac{1}{\delta}. \tag{2.8}$$

Demostración. La siguiente demostración es una adaptación de [24] y [25]. Supóngase que $\mathcal{M}$ satisface $\varepsilon(\alpha)$-RDP de orden $\alpha > 1$. Introduciendo la variable aleatoria de pérdida de privacidad y usando que $\big(\mathbb{P}[\mathcal{M}(D) = y]\big)/\big(\mathbb{P}[\mathcal{M}(D') = y]\big) = e^{Z(y)}$, la definición de divergencia de Rényi implica:

$$
\begin{aligned}
D_\alpha\big(\mathcal{M}(D)\,\|\,\mathcal{M}(D')\big) &= \frac{1}{\alpha-1}\log\int_{\mathcal{Y}}\left(\frac{\mathbb{P}[\mathcal{M}(D)=y]}{\mathbb{P}[\mathcal{M}(D')=y]}\right)^{\alpha}\mathbb{P}[\mathcal{M}(D')=y]\,dy \\
&= \frac{1}{\alpha-1}\log\int_{\mathcal{Y}}\left(\frac{\mathbb{P}[\mathcal{M}(D)=y]}{\mathbb{P}[\mathcal{M}(D')=y]}\right)^{\alpha-1}\mathbb{P}[\mathcal{M}(D)=y]\,dy \\
&= \frac{1}{\alpha-1}\log\int_{\mathcal{Y}}\exp\big((\alpha-1)Z(y)\big)\,\mathbb{P}[\mathcal{M}(D)=y]\,dy \\
&= \frac{1}{\alpha-1}\log\mathbb{E}_{y\sim\mathcal{M}(D)}\left[\exp\big((\alpha-1)Z(y)\big)\right] \\
&\leq \varepsilon(\alpha).
\end{aligned}
$$

Equivalente y reordenando, se obtiene una cota sobre el momento exponencial de la pérdida de privacidad:

$$
\mathbb{E}_{y\sim\mathcal{M}(D)}\left[\exp\big((\alpha-1)Z(y)\big)\right] \leq \exp\big((\alpha-1)\varepsilon(\alpha)\big).
$$

Sea ahora $\varepsilon' > 0$ arbitrario. Aplicando la desigualdad de Markov a la variable aleatoria no negativa $\exp\big((\alpha-1)Z\big)$ y la cota proporcionada por la cota anterior, se obtiene:

$$
\begin{aligned}
\mathbb{P}_{y\sim\mathcal{M}(D)}\big[Z(y) > \varepsilon'\big] &\leq \exp\big(-(\alpha-1)\varepsilon'\big)\,\mathbb{E}_{y\sim\mathcal{M}(D)}\left[\exp\big((\alpha-1)Z(y)\big)\right] \\
&\leq \exp\big((\alpha-1)(\varepsilon(\alpha)-\varepsilon')\big).
\end{aligned}
$$

Para un nivel de fallo $\delta \in (0,1)$, eligiendo ε' tal que:

$$
\varepsilon' = \varepsilon(\alpha) + \frac{1}{\alpha-1}\log\frac{1}{\delta}
$$

se obtiene

$$
\mathbb{P}_{y\sim\mathcal{M}(D)}\big[Z(y) > \varepsilon(\delta)\big] \leq \delta,
$$

con $\varepsilon(\delta)$ dado por la expresión anterior. Por tanto, el mecanismo $\mathcal{M}$ satisface $(\varepsilon(\delta),\delta)$-DP. □

En consecuencia, el análisis de privacidad a lo largo de este trabajo se realizará en términos de RDP, convirtiendo posteriormente las garantías obtenidas a parámetros (ε,δ) cuando sea necesario.

Teorema 2.8. *Sean $\mathcal{M}_1$ y $\mathcal{M}_2$ dos mecanismos aleatorios aplicados secuencialmente sobre un mismo conjunto de datos. Si $\mathcal{M}_1$ satisface $\varepsilon_1(\alpha)$-RDP y $\mathcal{M}_2$ satisface $\varepsilon_2(\alpha)$-RDP, ambos de orden $\alpha > 1$, entonces el mecanismo compuesto:*

$$
\mathcal{M}(D) = \big(\mathcal{M}_1(D), \mathcal{M}_2(D)\big) \tag{2.9}
$$

satisface $\big(\varepsilon_1(\alpha)+\varepsilon_2(\alpha)\big)$-RDP de orden α.

Demostración. La siguiente demostración es una adaptación de [24] al caso de mecanismos independientes. Sea $D \sim D'$. Usando la independencia de los mecanismos:

$$\begin{aligned}
\exp\big[(\alpha-1)D_\alpha(\mathcal{M}(D)\|\mathcal{M}(D'))\big] &= \int_{\mathcal{Y}_1\times\mathcal{Y}_2} \big(\mathbb{P}[\mathcal{M}_1(D)=x]\mathbb{P}[\mathcal{M}_2(D)=y]\big)^\alpha \\
&\qquad \cdot \big(\mathbb{P}[\mathcal{M}_1(D')=x]\mathbb{P}[\mathcal{M}_2(D')=y]\big)^{1-\alpha}\,dx\,dy \\
&= \int_{\mathcal{Y}_1\times\mathcal{Y}_2} \mathbb{P}[\mathcal{M}_1(D)=x]^\alpha \mathbb{P}[\mathcal{M}_1(D')=x]^{1-\alpha} \\
&= \mathbb{P}[\mathcal{M}_2(D)=y]^\alpha \mathbb{P}[\mathcal{M}_2(D')=y]^{1-\alpha}\,dx\,dy \\
&= \left(\int_{\mathcal{Y}_1} \mathbb{P}[\mathcal{M}_1(D)=x]^\alpha \mathbb{P}[\mathcal{M}_1(D')=x]^{1-\alpha}\,dx\right) \\
&\qquad \cdot \left(\int_{\mathcal{Y}_2} \mathbb{P}[\mathcal{M}_2(D)=y]^\alpha \mathbb{P}[\mathcal{M}_2(D')=y]^{1-\alpha}\,dy\right) \\
&\le \exp\big((\alpha-1)\varepsilon_1(\alpha)\big)\,\exp\big((\alpha-1)\varepsilon_2(\alpha)\big) \\
&= \exp\big((\alpha-1)(\varepsilon_1(\alpha)+\varepsilon_2(\alpha))\big).
\end{aligned}$$

Tomando logaritmos y dividiendo por $\alpha-1$ se concluye que:

$$D_\alpha\big(\mathcal{M}(D)\|\mathcal{M}(D')\big) \le \varepsilon_1(\alpha)+\varepsilon_2(\alpha),$$

lo que demuestra la aditividad de la RDP bajo composición de mecanismos independientes. □

El resultado anterior permite llevar una contabilidad exacta de la pérdida de privacidad cuando se aplican mecanismos de forma secuencial, lo que resulta especialmente relevante en algoritmos iterativos. Para instanciar mecanismos concretos con garantías de privacidad, es necesaria la siguiente definición.

Definición 2.9 (Sensibilidad de una función)**.** Sea $f : \mathcal{D} \to \mathbb{R}^d$ una función determinista. La *sensibilidad* de f respecto a la norma euclídea $\|\cdot\|_2$ se define como:

$$\Delta f = \sup_{D\sim D'} \big\|f(D)-f(D')\big\|_2. \tag{2.10}$$

La sensibilidad mide la máxima variación posible de la salida de f cuando se modifica una única contribución individual en el conjunto de datos. Uno de los mecanismos más utilizados en este contexto es el mecanismo gaussiano, que consiste en perturbar la salida de una función determinista mediante ruido gaussiano isotrópico.

Definición 2.10 (Mecanismo gaussiano)**.** Sea $f : \mathcal{D} \to \mathbb{R}^d$ una función. El *mecanismo gaussiano* se define como

$$\mathcal{M}(D) = f(D) + \mathcal{N}(0, \sigma^2 I_d), \tag{2.11}$$

donde $\mathcal{N}(0, \sigma^2 I_d)$ denota una distribución normal multivariante con media nula y covarianza $\sigma^2 I_d$.

El parámetro σ controla el nivel de ruido añadido y se elige en función de las garantías de privacidad deseadas. En particular, el mecanismo gaussiano admite una caracterización especialmente sencilla en términos de RDP.

Teorema 2.11. *El mecanismo gaussiano $\mathcal{M}(D) = f(D) + \mathcal{N}(0, \sigma^2 I_d)$ satisface $\varepsilon(\alpha)$-RDP de orden $\alpha > 1$, donde:*

$$\varepsilon(\alpha) = \frac{\alpha(\Delta f)^2}{2\sigma^2}, \tag{2.12}$$

y Δf es la sensibilidad de f.

Demostración. La siguiente demostración es una adaptación de [23]. Sea $D \sim D'$ un par de conjuntos de datos adyacentes. Comenzamos estableciendo un lema auxiliar.

Lema 2.12 (Invarianza a traslaciones)**.** *La divergencia de Rényi entre dos gaussianas con la misma covarianza es invariante a traslaciones:*

$$D_\alpha\big(\mathcal{N}(\mu_1, \sigma^2 I_d) \| \mathcal{N}(\mu_2, \sigma^2 I_d)\big) = D_\alpha\big(\mathcal{N}(\mu_1 - \mu_2, \sigma^2 I_d) \| \mathcal{N}(0, \sigma^2 I_d)\big). \tag{2.13}$$

Demostración. Por definición de divergencia de Rényi, el cambio de variable $z' = z - \mu_2$ en la integral:

$$\int_{\mathbb{R}^d} \frac{p(z)^\alpha}{q(z)^{\alpha-1}} dz = \int_{\mathbb{R}^d} \frac{p(z'+\mu_2)^\alpha}{q(z'+\mu_2)^{\alpha-1}} dz',$$

donde $p(z'+\mu_2) \propto \exp\big(-\|z' - (\mu_1 - \mu_2)\|^2/(2\sigma^2)\big)$ y $q(z'+\mu_2) \propto \exp\big(-\|z'\|^2/(2\sigma^2)\big)$, que es exactamente la integral correspondiente a medias $\mu_1 - \mu_2$ y 0. □

Por el lema anterior, sin pérdida de generalidad se puede tomar $f(D') = 0$ y $f(D) = \mu$ con $\mu = f(D) - f(D')$, obteniendo:

$$\begin{aligned} D_\alpha\big(\mathcal{M}(D) \| \mathcal{M}(D')\big) &= \frac{1}{\alpha - 1} \log \mathbb{E}_{z \sim \mathcal{M}(0)} \left(\frac{\mathbb{P}[\mathcal{M}(\mu) = z]}{\mathbb{P}[\mathcal{M}(0) = z]} \right)^\alpha \\ &= \frac{1}{\alpha - 1} \log \int_{\mathbb{R}^d} \mathbb{P}[\mathcal{M}(\mu) = z]^\alpha \cdot \mathbb{P}[\mathcal{M}(0) = z]^{1-\alpha} \, dz. \end{aligned}$$

Dado que el mecanismo gaussiano es isótropo, el integrando depende de z únicamente a través de su proyección escalar sobre la dirección de μ. Mediante una rotación que lleva μ al eje coordenado e_1, la integral sobre $\mathbb{R}^d$ se reduce a una integral gaussiana unidimensional que puede evaluarse analíticamente completando el cuadrado. Omitiendo el desarrollo algebraico por brevedad, se obtiene:

$$D_\alpha\big(\mathcal{M}(D) \| \mathcal{M}(D')\big) = \frac{\alpha \|\mu\|_2^2}{2\sigma^2} \leq \frac{\alpha(\Delta f)^2}{2\sigma^2},$$

donde la desigualdad se obtiene por definición de sensibilidad:

$$\|\mu\|_2 = \|f(D) - f(D')\| \leq \Delta f.$$

Por tanto, el mecanismo gaussiano satisface $\varepsilon(\alpha)$-RDP de orden α con $\varepsilon(\alpha)$ como en la Ecuación (2.12). □

En muchos escenarios prácticos el mecanismo privado no se aplica sobre el conjunto completo de datos, sino únicamente sobre una fracción aleatoria de este. Esta aleatoriedad en la participación introduce una fuente adicional de privacidad conocida como amplificación por submuestreo [26].

Teorema 2.13. *Sea $\mathcal{M}$ un mecanismo aleatorio que satisface $(\alpha, \varepsilon(\alpha))$-RDP cuando se aplica sobre un conjunto de datos D. Sea $\mathcal{S}_q$ un esquema de submuestreo de Poisson que incluye cada elemento de D de manera independiente con probabilidad $q \in (0, 1)$. Se define el mecanismo submuestreado como*

$$\mathcal{M}_q(D) = \mathcal{M}(\mathcal{S}_q(D)).$$

Entonces $\mathcal{M}_q$ satisface $(\alpha, \varepsilon_q(\alpha))$-RDP, donde $\varepsilon_q(\alpha)$ admite la siguiente cota explícita

$$\varepsilon_q(\alpha) \leq \frac{1}{\alpha - 1} \log \left(1 + \sum_{j=2}^{\alpha} \binom{\alpha}{j} q^j \exp\big((j-1)\varepsilon(j)\big) \right). \tag{2.14}$$

En particular, para el caso en que $\mathcal{M}$ es el mecanismo gaussiano con varianza σ^2 y sensibilidad Δf, se cumple

$$\varepsilon(j) = \frac{j(\Delta f)^2}{2\sigma^2},$$

lo que proporciona una cota cerrada y computable para la pérdida de privacidad en RDP bajo submuestreo de Poisson.

Este resultado presenta una versión simplificada. Si bien es posible obtener cotas más estrictas sin dicha aproximación, su derivación excede el alcance de este trabajo. Para un tratamiento exhaustivo se remite al lector a [26].

Los resultados anteriores establecen un marco unificado para analizar la privacidad de mecanismos iterativos mediante RDP, permitiendo cuantificar de forma precisa la pérdida de privacidad asociada a la adición de ruido gaussiano y su composición a lo largo de múltiples iteraciones. En la siguiente sección se aplicará este marco al estudio de la DP en los FRS.

2.3.2. Privacidad diferencial en sistemas de recomendación federados

Existen dos enfoques principales para aplicar privacidad diferencial en los sistemas de recomendación federados [27], que difieren en el punto del sistema en el que se introduce el mecanismo de privacidad.

En el enfoque de **privacidad diferencial local** (*Local Differential Privacy*, LDP) [28], cada cliente perturba sus actualizaciones locales antes de enviarlas al servidor. De este modo, las garantías de privacidad se mantienen frente a cualquier observador externo, incluido el propio servidor, ya que las actualizaciones transmitidas no contienen información exacta sobre los datos individuales. No obstante, este enfoque requiere la adición de una cantidad considerable de ruido en cada cliente, lo que suele degradar la calidad del modelo aprendido.

Por otro lado, el enfoque de **privacidad diferencial central** (*Central Differential Privacy*, CDP) introduce el mecanismo de privacidad en el servidor, tras la agregación de las actualizaciones locales. En este caso, se asume que el servidor es una entidad de confianza para ejecutar correctamente el protocolo de agregación y aplicar el mecanismo de DP, pero no para acceder directamente a los datos individuales de los clientes. Bajo este supuesto, el ruido se añade únicamente al resultado agregado, lo que permite reducir la varianza introducida por el mecanismo y obtener un mejor compromiso entre privacidad y rendimiento.

En este trabajo se adopta el enfoque de CDP, el cual, en el contexto de FL, resulta especialmente adecuado cuando el objetivo es garantizar privacidad a nivel de usuario, protegiendo la contribución completa de cada cliente al modelo global. Además, la CDP se integra de forma natural con esquemas de agregación como FedAvg (véase Sección 2.2.2) el cual puede considerarse como la función determinista de un mecanismo aleatorio.

2.3.3. Implementación de privacidad diferencial

La implementación práctica de mecanismos de DP en modelos de aprendizaje automático requiere el uso de librerías especializadas que integren de forma eficiente tanto la adición de ruido calibrado como la contabilidad de privacidad. Un contador de privacidad es un módulo que acumula y registra la pérdida de privacidad incurrida, permitiendo obtener en todo momento el presupuesto de privacidad consumido. Entre las principales herramientas disponibles se encuentran `TensorFlow Privacy` [29], `Opacus` [30] y `Diffprivlib` [31]. En particular, `Opacus` proporciona soporte directo para contadores de privacidad basados en RDP. Dado que el análisis de privacidad se realiza en términos de RDP, se ha seleccionado `Opacus` como herramienta principal para la implementación experimental de los mecanismos de privacidad diferencial.

2.4. Evaluación

En el análisis de modelos de aprendizaje supervisado es necesario evaluar su rendimiento. El procedimiento estándar se basa en la partición del conjunto de datos disponible en tres subconjuntos disjuntos: un conjunto de entrenamiento, utilizado para ajustar los parámetros del modelo; un conjunto de validación, empleado en la selección de hiperparámetros mediante validación empírica; y un conjunto de test, reservado para la evaluación final del modelo.

La validación empírica consiste en entrenar un conjunto de configuraciones candidatas sobre el conjunto de entrenamiento, evaluarlas sobre el conjunto de validación y seleccionar aquella con mayor utilidad. De este modo, el rendimiento se mide sobre ejemplos no observados durante el aprendizaje, proporcionando una estimación empírica de su capacidad de generalización y asegurando que la selección de hiperparámetros no introduce un sesgo en los resultados finales.

La elección de las métricas de evaluación depende de la naturaleza de la variable objetivo y del tipo de tarea abordada. En particular, para tareas de regresión supervisada, una métrica habitual es la raíz del error cuadrático medio (*Root Mean Squared Error*, RMSE), definida como:

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} (y - \hat{y}(x))^2}, \tag{2.15}$$

donde Ω denota el conjunto de instancias del conjunto de evaluación, y representa el valor real de la variable objetivo y $\hat{y}$ la predicción generada por el modelo. Esta métrica será la utilizada para validar la utilidad de todos los modelos entrenados en este trabajo.

Cabe recalcar que en el contexto específico de los sistemas de recomendación existen métricas adicionales orientadas a evaluar la calidad del *ranking* de las recomendaciones, como *Recall@k* o *Precision@k* [6]. Su análisis queda no obstante fuera del alcance del presente trabajo.

3

Diseño e Implementación

En este capítulo se presenta el diseño y la implementación del sistema propuesto, estructurado de forma incremental en distintas fases. El objetivo es describir cómo el sistema evoluciona desde un modelo de recomendación centralizado hasta una arquitectura de aprendizaje federado que incorpora mecanismos de privacidad diferencial. La organización del capítulo sigue esta evolución progresiva.

3.1. Diseño general

Esta sección describe los objetivos, desarrollo y arquitectura del sistema de manera general.

3.1.1. Objetivos

El diseño del sistema se ha guiado por una serie de objetivos principales que condicionan la arquitectura empleada:

1. Implementar un sistema de recomendación basado en la factorización de matrices, capaz de modelar interacciones usuario–ítem de forma eficiente y con buen rendimiento en un entorno centralizado.
2. Evolucionar el sistema desde un enfoque centralizado a uno federado, en el que los datos de los usuarios no se agregan en un único repositorio, sino que permanecen distribuidos.
3. Incorporar mecanismos de privacidad diferencial central que proporcionen garantías formales de privacidad y controlen explícitamente el presupuesto de privacidad.
4. Mantener una arquitectura modular y extensible, que permita incorporar futuras mejoras o extensiones, como privacidad local o encriptación homomórfica.

Estos objetivos aseguran que el sistema no solo sea funcional, sino también adecuado como plataforma experimental para el análisis de técnicas de FL y DP en sistemas de recomendación.

3.1.2. Desarrollo

Con el objetivo de abordar de forma sistemática los objetivos, el desarrollo del sistema se ha estructurado en tres fases claramente diferenciadas:

Fase I Modelo centralizado. Implementación de un sistema de recomendación basado en la factorización de matrices en un entorno centralizado, que sirve como línea base para la evaluación del rendimiento.

Fase II Aprendizaje federado. Extensión del modelo centralizado al paradigma de aprendizaje federado, en el que los datos permanecen distribuidos entre los usuarios y el entrenamiento se realiza de forma colaborativa mediante la agregación de actualizaciones locales.

Fase III Privacidad diferencial. Incorporación de mecanismos de privacidad diferencial central en el proceso de agregación del servidor.

Esta estructura incremental permite mantener una arquitectura modular y extensible, distinguiendo las distintas funcionalidades del sistema.

3.1.3. Arquitectura

A alto nivel, el sistema final se compone de dos entidades principales: los clientes, que representan a los usuarios y almacenan localmente sus datos de interacción, y un servidor central, responsable de coordinar el proceso de entrenamiento y de mantener los parámetros globales del modelo. En la fase centralizada, el servidor dispone de acceso directo a todos los datos de entrenamiento y realiza el aprendizaje de forma clásica. En las fases posteriores, el rol del servidor evoluciona hacia un coordinador que agrega actualizaciones procedentes de los clientes, sin acceder directamente a sus datos.

La arquitectura general del sistema se resume de forma esquemática en la Figura 3.1, siguiendo el esquema iterativo cliente-servidor del FL descrito en la Sección 2.2. En ella se representan los distintos clientes, que mantienen localmente sus datos de interacción, y el servidor central, encargado de coordinar el proceso de entrenamiento. Asimismo, se ilustra el flujo de parámetros entre clientes y servidor, así como el punto en el que se aplica el mecanismo de DP. La implementación completa del sistema está disponible públicamente en [32].

3.2. Fase I: Factorización de matrices centralizada

En esta primera fase se desarrolla un modelo de factorización de matrices en un entorno centralizado, que sirve como punto de partida para el resto del trabajo. El objetivo es establecer una base

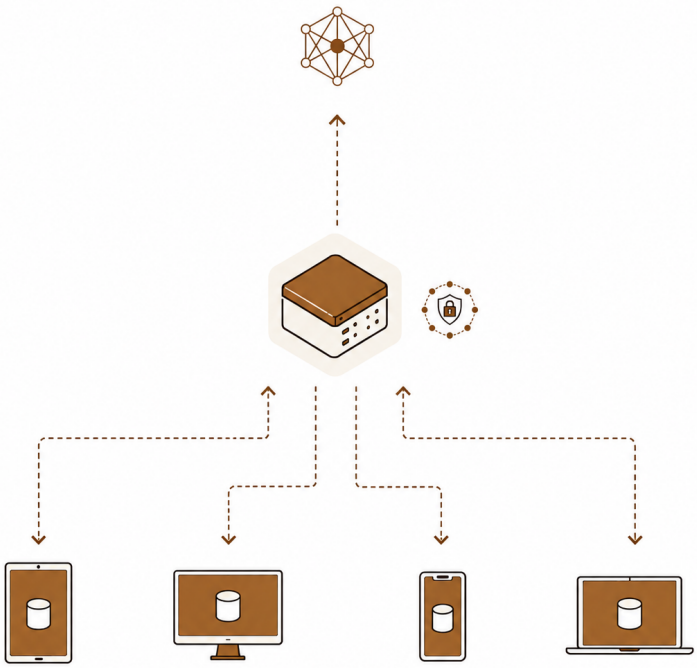


Figura 3.1: Arquitectura propuesta, mostrando el flujo de parámetros entre clientes y servidor central y la aplicación del mecanismo de privacidad diferencial. Figura adaptada de [19].

sólida tanto teórica como experimental, definiendo la formulación matemática del problema, las mejoras introducidas sobre el modelo clásico y el procedimiento de entrenamiento utilizado.

3.2.1. Formulación básica y optimizaciones

Bajo el enfoque de MF descrito en la Sección 2.1.2, la predicción para un par usuario–ítem es:

$$r_{ui} = \mathbf{p}_u^\top \mathbf{q}_i,$$

donde $\mathbf{p}_u$ y $\mathbf{q}_i$ representan las filas correspondientes de P y Q. Aunque la formulación básica de MF permite capturar interacciones latentes entre usuarios e ítems, en escenarios reales resulta habitual introducir mejoras adicionales que modelen efectos sistemáticos presentes en los datos y mejoren la estabilidad del entrenamiento.

En esta fase se incorporan tres extensiones fundamentales [33]. En primer lugar, se introduce la **media global** μ, definida como el valor medio de las valoraciones del conjunto de entrenamiento. Este término captura la tendencia general del sistema. Adicionalmente, se incorporan **sesgos** específicos de usuario b_u y de ítem b_i. Algunos usuarios tienden a otorgar valoraciones más altas o más bajas

de forma consistente, mientras que ciertos ítems reciben valoraciones sistemáticamente superiores o inferiores. La inclusión de estos términos permite capturar dichos patrones. Con estas mejoras, la expresión final del modelo de predicción queda definida como:

$$r_{ui} = \mu + b_u + b_i + \mathbf{p}_u^\top \mathbf{q}_i. \tag{3.1}$$

Se denotarán por $\mathbf{b}_{user}$ y $\mathbf{b}_{item}$ los vectores de sesgos y por b_u y b_i los sesgos correspondientes al usuario u y al ítem i, respectivamente. El aprendizaje de los parámetros se formula como un problema de optimización que minimiza el error cuadrático medio sobre el conjunto de valoraciones observadas, incorporando términos de regularización ℓ_2 sobre todos los parámetros a entrenar del modelo:

$$\min_{P,Q,b_u,b_i} \sum_{(u,i)\in\Omega} (\hat{r}_{ui} - r_{ui})^2 + \lambda(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + b_u^2 + b_i^2), \tag{3.2}$$

donde $\lambda > 0$ es el coeficiente de regularización. Este término penaliza valores elevados de los factores latentes y de los sesgos, mejorando la capacidad de generalización del modelo. La media global se calcula una única vez a partir del conjunto de entrenamiento y se mantiene fija, mientras que los factores latentes y los sesgos se optimizan conjuntamente mediante el algoritmo de aprendizaje. Esta formulación extendida constituye la versión final del modelo centralizado que se emplea como referencia en las fases posteriores del sistema.

3.2.2. Algoritmo de entrenamiento

El entrenamiento del modelo de MF se realiza mediante descenso por gradiente estocástico (*Stochastic Gradient Descent*, SGD) [34], una técnica ampliamente utilizada en los RS debido a su eficiencia computacional y su buen comportamiento en problemas de gran escala.

A partir de la función objetivo definida en la Ecuación 3.2, los parámetros del modelo se actualizan iterativamente minimizando el error cuadrático medio. En cada iteración, el algoritmo procesa subconjuntos de datos (*minibatch*), lo que introduce aleatoriedad en el proceso de optimización y favorece una convergencia más estable. La aleatoriedad en las actualizaciones permite al algoritmo SGD escapar de mínimos locales en la función de pérdida, cuya estructura suele ser altamente compleja y no convexa. El resultado son predicciones puntuales potencialmente mejores.

El procedimiento completo de entrenamiento se describe en el Algoritmo 3.1. El algoritmo se implementa de forma directa utilizando `PyTorch`. `PyTorch` permite integrar de forma eficiente un optimizador basado en SGD, lo que facilita tanto la reproducibilidad como la extensión del modelo a las fases posteriores.

input : Conjunto de valoraciones R; dimensión latente k; número de *epochs* E; tasa de aprendizaje η; coeficiente de regularización λ
output: Matrices de factores latentes P, Q y vectores de sesgo $\mathbf{b}_u$, $\mathbf{b}_i$

Inicializar $P \in \mathbb{R}^{n \times k}$ y $Q \in \mathbb{R}^{m \times k}$ con valores aleatorios pequeños;
Inicializar $\mathbf{b}_u \leftarrow \mathbf{0}$ y $\mathbf{b}_i \leftarrow \mathbf{0}$;
Calcular la media global μ a partir del conjunto de entrenamiento;
for $e \leftarrow 1$ **to** E **do**
  foreach *minibatch* $B \subset R$ **do**
    foreach $(u, i, r_{ui}) \in B$ **do**
      $\hat{r}_{ui} \leftarrow \mu + b_u + b_i + \mathbf{p}_u^\top \mathbf{q}_i$;
      $err \leftarrow \hat{r}_{ui} - r_{ui}$;
      Actualizar $\mathbf{p}_u \leftarrow \mathbf{p}_u - \eta(err\, \mathbf{q}_i + \lambda \mathbf{p}_u)$;
      Actualizar $\mathbf{q}_i \leftarrow \mathbf{q}_i - \eta(err\, \mathbf{p}_u + \lambda \mathbf{q}_i)$;
      Actualizar $b_u \leftarrow b_u - \eta(err + \lambda b_u)$;
      Actualizar $b_i \leftarrow b_i - \eta(err + \lambda b_i)$;
    end
  end
end

Algoritmo 3.1: Algoritmo de entrenamiento centralizado para MF con media global, sesgos y regularización, optimizado mediante SGD.

3.3. Fase II: Extensión al aprendizaje federado

En esta segunda fase se extiende el modelo de MF centralizado al paradigma de FL, con el objetivo de eliminar la necesidad de centralizar los datos de los usuarios durante el proceso de entrenamiento. El propósito de esta fase es adaptar la formulación y el algoritmo de entrenamiento presentados anteriormente a un entorno distribuido, manteniendo la estructura fundamental del modelo y permitiendo una comparación coherente con el enfoque centralizado.

3.3.1. Configuración del sistema federado

En el escenario de FL considerado, el sistema se compone de un conjunto de clientes $\mathcal{C}$, donde cada cliente $c \in \mathcal{C}$ agrupa a uno o más usuarios $\mathcal{U}_c$, y de un servidor central encargado de coordinar el proceso de entrenamiento. Cada cliente dispone únicamente de las interacciones de los usuarios asignados a él y no comparte directamente los datos brutos con el servidor ni con otros clientes.

Como se vio en la Sección 2.2.2, en el contexto de MF los parámetros del modelo se dividen en función de su naturaleza y del rol que desempeñan en el FL. En particular, los factores latentes de usuario $\mathbf{p}_u$ y el sesgo de usuario b_u se consideran parámetros locales para cada usuario $u \in \mathcal{U}_c$. En particular, el cliente c gestiona localmente el conjunto de parámetros de usuario $\{\mathbf{p}_u, b_u\}_{u \in \mathcal{U}_c}$. Por el contrario, los factores latentes de ítem q_i, el sesgo de ítem b_i y la media global μ se tratan como parámetros globales, compartidos y actualizados de forma colaborativa a través del servidor central

durante el proceso de entrenamiento federado. Esta configuración federada constituye la base sobre la que se define el algoritmo de entrenamiento distribuido presentado a continuación.

3.3.2. Algoritmo federado propuesto

El entrenamiento federado del modelo de MF se articula mediante una interacción iterativa entre clientes y servidor, organizada en rondas de aprendizaje colaborativo. En cada ronda, el servidor coordina la distribución de los parámetros globales del modelo y la agregación de las actualizaciones generadas localmente por los clientes, sin acceder a los datos de interacción individuales.

El Algoritmo 3.2 detalla el procedimiento ejecutado en cada cliente. A partir de los parámetros globales recibidos y de sus parámetros locales persistentes, cada cliente c entrena el modelo sobre su conjunto de datos privados $\mathbf{r}_c = \{\mathbf{r}_u\}_{u \in \mathcal{U}_c}$ utilizando SGD. El entrenamiento local se realiza durante un número fijo de épocas y sobre *minibatches* de datos, siguiendo una función de pérdida análoga a la del caso centralizado (véase Ecuación (3.2)). Al finalizar el entrenamiento local, cada cliente calcula explícitamente las actualizaciones de los parámetros globales en forma de incrementos $\Delta\mathbf{w}^{(c)}$, que se envían al servidor para su posterior agregación.

input : Datos locales del cliente $r_c = \{r_u\}_{u \in \mathcal{U}_c}$; parámetros globales Q, $\mathbf{b}_{item}$, μ; parámetros locales actuales $\{\mathbf{p}_u, b_u\}_{u \in \mathcal{U}_c}$; número de *epochs* locales E; tasa de aprendizaje η; coeficiente de regularización λ
output: Actualizaciones locales $\Delta Q^{(c)}$, $\Delta \mathbf{b}_{item}^{(c)}$

Guardar copias iniciales $Q^{(0)} \leftarrow Q$, $\mathbf{b}_{item}^{(0)} \leftarrow \mathbf{b}_{item}$;
for $e \leftarrow 1$ **to** E **do**
  foreach *minibatch* $B \subset r_c$ **do**
    foreach $(u, i, r_{ui}) \in B$ **do**
      $\hat{r}_{ui} \leftarrow \mu + b_u + b_i + \mathbf{p}_u^\top \mathbf{q}_i$;
      $err \leftarrow \hat{r}_{ui} - r_{ui}$;
      Actualizar $\mathbf{p}_u \leftarrow \mathbf{p}_u - \eta(err\, \mathbf{q}_i + \lambda \mathbf{p}_u)$;
      Actualizar $b_u \leftarrow b_u - \eta(err + \lambda b_u)$;
      Actualizar $\mathbf{q}_i \leftarrow \mathbf{q}_i - \eta(err\, \mathbf{p}_u + \lambda \mathbf{q}_i)$;
      Actualizar $b_i \leftarrow b_i - \eta(err + \lambda b_i)$;
    end
  end
end
Calcular $\Delta Q^{(c)} \leftarrow Q - Q^{(0)}$;
Calcular $\Delta \mathbf{b}_{item}^{(c)} \leftarrow \mathbf{b}_{item} - \mathbf{b}_{item}^{(0)}$;
Enviar $\Delta Q^{(c)}$ y $\Delta \mathbf{b}_{item}^{(c)}$ al servidor;

Algoritmo 3.2: Algoritmo ejecutado por cada cliente para el entrenamiento local mediante SGD y el cálculo de actualizaciones federadas.

Para la fase de agregación en el servidor se adopta el esquema FedAvg (véase Ecuación (2.2)). En el caso de este trabajo, no se pondera por el número de contribuciones de cada cliente. En concreto, el servidor combina las actualizaciones locales mediante una media aritmética simple, tratando a todos

los clientes de forma homogénea. Esta elección se realiza en el contexto de DP a nivel de usuario, ya que evita que clientes con un mayor volumen de interacciones ejerzan una influencia desproporcionada sobre el modelo agregado, lo cual complicaría el análisis de la sensibilidad del mecanismo.

Con el fin de formalizar el proceso de agregación y facilitar el análisis posterior desde el punto de vista de la DP, resulta conveniente introducir una notación matemática que abstraiga las actualizaciones locales generadas por los clientes y la operación de combinación realizada por el servidor. Sea:

$$\Delta \mathbf{w}^{(c)} := \left(\Delta \mathbf{q}_1^{(c)},\ \ldots,\ \Delta \mathbf{q}_n^{(c)},\ \Delta \mathbf{b}_{item}^{(c)} \right)^\top \in \mathbb{R}^d, \tag{3.3}$$

la actualización local generada por el cliente c, donde $\{\Delta \mathbf{q}_j^{(c)}\}_{j=1}^n$ representa las variaciones de los parámetros latentes asociados a los ítems y $\Delta \mathbf{b}_{item}^{(c)}$ corresponde a las actualizaciones de los sesgos de ítem. La dimensión total del vector viene dada por $d = n(k+1)$. Sea $\mathcal{D}$ el espacio de datasets, donde cada $D = \{\Delta \mathbf{w}^{(c)}\}_{c \in \mathcal{C}}$ es un conjunto finito de actualizaciones $\Delta \mathbf{w}^{(c)} \in \mathbb{R}^d$. La operación de agregación realizada por el servidor puede expresarse mediante la función determinista:

$$f : \mathbb{R}^{|\mathcal{C}| \times d} \to \mathbb{R}^d, \qquad f(D) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \Delta \mathbf{w}^{(c)}, \tag{3.4}$$

que corresponde a una media aritmética no ponderada de las actualizaciones locales recibidas. En consecuencia, la actualización de los parámetros globales en la ronda t puede escribirse como:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \Delta \mathbf{w}_t, \qquad \Delta \mathbf{w}_t = f\left(\{\Delta \mathbf{w}^{(c)}\}_{c \in \mathcal{C}} \right), \tag{3.5}$$

donde $\mathcal{C}$ denota el subconjunto de clientes participantes en dicha ronda y $\mathbf{w}$ se construye de la misma manera que en la Ecuación (3.3) con los Q y $\mathbf{b}_{item}$ del modelo antes de la agregación. Esta formulación permite analizar el mecanismo de agregación de forma independiente del procedimiento de optimización local.

El Algoritmo 3.3 describe el procedimiento ejecutado en el servidor central. En cada ronda federada, el servidor selecciona un subconjunto de clientes participantes, distribuye los parámetros globales actuales y agrega las actualizaciones locales recibidas siguiendo la formulación de FedAvg definidas en las Ecuaciones (3.3)– (3.5). El resultado de esta agregación define una nueva estimación de los parámetros globales que se emplea en la siguiente ronda federada. El Algoritmo 3.4 describe el procedimiento de selección de clientes mediante muestreo independiente con probabilidad fija por ronda federada. Este esquema es equivalente a un muestreo de Poisson y esta directamente relacionado con el análisis del presupuesto de privacidad que se efectuará en la siguiente sección.

En conjunto, el algoritmo federado propuesto mantiene la estructura fundamental del modelo centralizado, al tiempo que elimina la necesidad de centralizar los datos de los usuarios. Una vez mas,

input : Conjunto de clientes $\mathcal{C}$; número de rondas federadas T
output: Parámetros globales actualizados Q, $\mathbf{b}_{item}$, μ

Inicializar los parámetros globales Q, $\mathbf{b}_{item}$ y μ;
for $t \leftarrow 1$ **to** T **do**
  Muestrear subconjunto de clientes $\mathcal{C}_t \subseteq \mathcal{C}$;
  Enviar Q, $\mathbf{b}_{item}$ y μ a los clientes de $\mathcal{C}_t$;
  Construir $\mathbf{w} \leftarrow (\mathbf{q}_1, \ldots, \mathbf{q}_n, \mathbf{b}_{item})^\top$
  foreach *cliente* $c \in \mathcal{C}_t$ (en paralelo) **do**
    Recibir $\Delta Q^{(c)}$ y $\Delta \mathbf{b}_i^{(c)}$;
    Construir $\Delta \mathbf{w}^{(c)} \leftarrow \left(\Delta \mathbf{q}_1^{(c)}, \ldots, \Delta \mathbf{q}_n^{(c)}, \Delta \mathbf{b}_{item}^{(c)}\right)^\top$
  end
  Agregar $\Delta \mathbf{w}_t \leftarrow \frac{1}{|\mathcal{C}_t|} \sum_{c \in \mathcal{C}_t} \Delta \mathbf{w}^{(c)}$
  Actualizar $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \Delta \mathbf{w}_t$
  Reconstruir $(Q, \mathbf{b}_{item}) \leftarrow \textsc{Unflatten}(\mathbf{w}_{t+1})$
end

Algoritmo 3.3: Algoritmo ejecutado por el servidor central para coordinar el entrenamiento federado y agregar actualizaciones locales mediante FedAvg.

la implementación práctica de los Algoritmos 3.2 y 3.3 se realiza utilizando `PyTorch`. La estructura modular del código facilita la separación entre la lógica de cliente y de servidor. Esta aproximación no solo simplifica la implementación, sino que resulta especialmente adecuada para la extensión del sistema a la fase de DP.

3.4. Fase III: Incorporación de privacidad diferencial central

El objetivo de esta fase es proteger a los usuarios frente a posibles ataques de inferencia, limitando la cantidad de información que puede extraerse de las actualizaciones agregadas. La sección comienza definiendo el modelo de amenaza considerado y los supuestos de seguridad adoptados. A continuación, se describe la implementación del mecanismo de CDP aplicado al proceso de agregación federada. Finalmente, se detalla el cálculo del presupuesto de privacidad.

3.4.1. Modelo de amenaza

El modelo de amenaza adoptado en este trabajo es el del servidor honesto-pero-curioso, introducido en la Sección 2.2.3. Concretamente, se asume que el servidor ejecuta correctamente el protocolo de agregación y aplica el mecanismo de DP especificado, pero podría intentar inferir información sobre los datos privados de los usuarios a partir de los mensajes que observa durante el entrenamiento.

Con el fin de reforzar esta garantía, se asume la existencia de un mecanismo de encriptación homomórfica a lo largo del proceso de agregación, de forma que las actualizaciones individuales de cada

input : Conjunto de clientes disponibles $\mathcal{C}$; tasa de muestreo $q \in (0, 1)$
output: Subconjunto de clientes seleccionados $\mathcal{C}_t$

Inicializar $\mathcal{C}_t \leftarrow \emptyset$;
foreach *cliente* $c \in \mathcal{C}$ **do**
  Generar $z \sim \mathcal{U}(0, 1)$;
  if $z < q$ **then**
    Añadir c a $\mathcal{C}_t$;
  end
end
return $\mathcal{C}_t$;

Algoritmo 3.4: Algoritmo de selección de clientes mediante muestreo independiente con probabilidad fija por ronda federada.

cliente permanecen cifradas durante su transmisión y agregación en el servidor. En consecuencia, el servidor únicamente tiene acceso a los parámetros globales del modelo Q y $\mathbf{b}_i$ tras la aplicación del mecanismo de DP, sin poder observar las contribuciones individuales de cada cliente. El principal objetivo de protección es garantizar que los parámetros globales del modelo resultantes no revelen información sensible sobre la contribución de ningún cliente concreto. En este sentido, la privacidad se define a nivel de cliente, protegiendo el conjunto de interacciones $\mathbf{r}_c = \{\mathbf{r}_u\}_{u \in \mathcal{U}_c}$ de todos los usuarios asignados al cliente c.

Respecto a los clientes, se adopta un modelo de amenaza completamente honesto (véase Sección 2.2.3), asumiendo que ejecutan el protocolo local correctamente y que no presentan ataques activos. Esto implica que los parámetros locales $\{\mathbf{p}_u, b_u\}_{u \in \mathcal{U}_c}$ permanecen protegidos en todo momento.

3.4.2. Adición de privacidad diferencial

Con el fin de garantizar CDP, se introduce ruido gaussiano directamente sobre el resultado de la agregación federada. Esto da lugar a un mecanismo gaussiano $\mathcal{M}$ (véase Definición 2.10), aplicado a la salida de la función de agregación del servidor (véase Ecuación (3.4)). Para que el mecanismo gaussiano satisfaga una garantía de DP acotada por un parámetro ε, es necesario que la sensibilidad de la función determinista f esté acotada (véase Teorema 2.11). Considérense dos conjuntos de datos adyacentes $D \sim D'$, con $D = \{\Delta\mathbf{w}^{(c)}\}_{c \in \mathcal{C}}$ y $D' = \{\Delta\mathbf{w}^{(c)}\}_{c \in \mathcal{C}'}$ que difieren únicamente en la contribución de un único cliente, es decir:

$$\mathcal{C} = \{c_1, \ldots, c_j, \ldots, c_n\}, \qquad \mathcal{C}' = \{c_1, \ldots, c'_j, \ldots, c_n\},$$

donde $|\mathcal{C}| = |\mathcal{C}'|$. La sensibilidad de f viene dada por:

$$\Delta f = \sup_{D \sim D'} \left\| f(D) - f(D') \right\|_2 = \left\| \frac{1}{|\mathcal{C}|} \left(\Delta\mathbf{w}^{(c_j)} - \Delta\mathbf{w}^{(c'_j)} \right) \right\|_2 \leq \frac{1}{|\mathcal{C}|} \left(\|\Delta\mathbf{w}^{(c_j)}\|_2 + \|\Delta\mathbf{w}^{(c'_j)}\|_2 \right).$$

En ausencia de restricciones adicionales, la sensibilidad de f puede ser arbitrariamente grande, ya que las actualizaciones locales no están acotadas en norma.

Para garantizar una sensibilidad finita y controlable, se aplica un procedimiento de *clipping* que proyecta cada actualización local sobre la bola euclídea de radio $C > 0$, esto es, $\|\tilde{\Delta}\mathbf{w}^{(c)}\|_2 \leq C$ para $c \in \mathcal{C}$. Bajo este supuesto, la sensibilidad de la función de agregación queda acotada por:

$$\Delta f \leq \frac{2C}{|\mathcal{C}|}. \tag{3.6}$$

El Algoritmo 3.5 describe el procedimiento ejecutado en el servidor central para llevar a cabo la agregación federada con DP. La ejecución es muy similar a la del Algoritmo 3.3: antes de proceder a la agregación, el servidor aplica el operador de clipping a cada actualización individual, garantizando que todas las contribuciones quedan acotadas en norma euclídea; sobre el resultado de esta agregación se añade ruido gaussiano isotrópico proporcional a σ^2, obteniéndose así la salida final del mecanismo gaussiano central; finalmente, el vector agregado ruidoso se utiliza para actualizar los parámetros globales del modelo.

input : Conjunto de clientes $\mathcal{C}$; número de rondas T; tasa de muestreo q; límite de clipping C; desviación estándar del ruido σ
output: Parámetros globales actualizados Q, $\mathbf{b}_{item}$

Inicializar los parámetros globales Q y $\mathbf{b}_{item}$;
for $t \leftarrow 1$ **to** T **do**
  Muestrear subconjunto de clientes $\mathcal{C}_t \subseteq \mathcal{C}$;
  Enviar Q, $\mathbf{b}_{item}$ y μ a los clientes de $\mathcal{C}_t$;
  Construir $\mathbf{w} \leftarrow (\mathbf{q}_1, \ldots, \mathbf{q}_n, \mathbf{b}_{item})^\top$
  foreach *cliente* $c \in \mathcal{C}_t$ (en paralelo) **do**
    Recibir $\Delta Q^{(c)}$ y $\Delta \mathbf{b}_{item}^{(c)}$;
    Construir $\Delta \mathbf{w}^{(c)} \leftarrow \left(\Delta \mathbf{q}_1^{(c)}, \ldots, \Delta \mathbf{q}_n^{(c)}, \Delta \mathbf{b}_{item}^{(c)}\right)^\top$
    Aplicar $\Delta \mathbf{w}^{(c)} \leftarrow \Delta \mathbf{w}^{(c)} \cdot \min\left(1, \frac{C}{\|\Delta \mathbf{w}^{(c)}\|_2}\right)$
  end
  Agregar $\Delta \mathbf{w}_t \leftarrow \frac{1}{|\mathcal{C}_t|} \sum_{c \in \mathcal{C}_t} \Delta \mathbf{w}^{(c)}$
  Generar $Z \sim \mathcal{N}(0, \sigma^2 I_d)$
  Actualizar $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \Delta \mathbf{w}_t + Z$
  Reconstruir $(Q, \mathbf{b}_{item}) \leftarrow$ Unflatten$(\mathbf{w}_{t+1})$
end

Algoritmo 3.5: Algoritmo ejecutado por el servidor central que agrega actualizaciones locales mediante un mecanismo de CDP.

La implementación práctica del Algoritmo 3.5 se ha realizado utilizando `PyTorch` nuevamente. De este modo, el algoritmo propuesto constituye una extensión natural de FedAvg que incorpora garantías formales de CDP, y establece una base adecuada para el análisis del presupuesto de privacidad acumulado que se aborda en la siguiente sección.

3.4.3. Cálculo del presupuesto de privacidad

En cada ronda federada, el servidor aplica un mecanismo gaussiano sobre la salida de la función de agregación definida en la Ecuación (3.4), con sensibilidad global $\Delta f = 2C/|\mathcal{C}|$. De acuerdo con la expresión de la privacidad diferencial de Rényi del mecanismo gaussiano (véase Teorema 2.11), una única aplicación del mecanismo sin tener en cuenta el muestreo de clientes satisface, para todo $\alpha > 1$:

$$\varepsilon(\alpha) = \frac{2\alpha C^2}{|\mathcal{C}|^2\sigma^2}. \tag{3.7}$$

Sin embargo, en el escenario considerado no todos los clientes participan en cada ronda federada, sino que el subconjunto de clientes $\mathcal{C}_t$ se selecciona mediante muestreo independiente con probabilidad fija q, equivalente a un muestreo de Poisson. Este esquema permite beneficiarse del fenómeno de *amplificación de la privacidad por subsampling*. En particular, aplicando el resultado de amplificación por muestreo para RDP (véase Teorema 2.13), se obtiene una cota más ajustada para la pérdida de privacidad por ronda, que depende explícitamente de la tasa de muestreo q, del nivel de ruido σ y del orden α:

$$\varepsilon_t(\alpha) = \frac{1}{\alpha - 1} \log \left(1 + q^2 \binom{\alpha}{2} \exp\left(\frac{2C^2}{|\mathcal{C}_t|^2\sigma^2} \right) + \sum_{j=3}^{\alpha} q^j \binom{\alpha}{j} \exp\left(\frac{j(j-1)C^2}{|\mathcal{C}_t|^2\sigma^2} \right) \right),$$

donde $\mathcal{C}_t$ es el conjunto de clientes muestreados en la iteración t.

El entrenamiento del sistema de recomendación federado se realiza de forma iterativa, aplicando este mecanismo de agregación ruidosa en cada ronda. Gracias a la propiedad de composición aditiva de la RDP (Teorema 2.8), la pérdida total de privacidad tras T rondas federadas viene dada por:

$$\varepsilon_T(\alpha) = \sum_{t=1}^{T} \varepsilon_t(\alpha) = T\,\varepsilon_t(\alpha),$$

donde $\varepsilon_t(\alpha)$ denota la cota por ronda tras tener en cuenta el efecto del subsampling. Finalmente, para obtener una garantía de DP en el sentido clásico, la cota acumulada en términos de RDP se transforma en parámetros (ε, δ). Fijado un nivel de fallo $\delta \in (0, 1)$, el parámetro ε se obtiene aplicando el resultado de conversión de RDP a DP (véase Teorema 2.7):

$$\varepsilon(\delta) = \min_{\alpha > 1} \left(\varepsilon_T(\alpha) + \frac{1}{\alpha - 1} \log \frac{1}{\delta} \right), \tag{3.8}$$

donde la minimización respecto al orden α permite obtener la cota más ajustada posible para el nivel de privacidad deseado.

En este trabajo, el cálculo del presupuesto de privacidad acumulado se ha realizado utilizando el contador de privacidad basado en RDP proporcionado por la librería `Opacus`. Dicho contador permite

integrar de forma automática los efectos del subsampling, la composición a lo largo de múltiples rondas y la conversión final a parámetros (ε, δ), garantizando la correcta correspondencia entre la formulación teórica presentada y su implementación práctica.

Es importante aclarar la convención utilizada para el parámetro σ en este trabajo. En la formulación teórica anterior, σ representa la desviación estándar absoluta del ruido añadido por el mecanismo. En la implementación, no obstante, resulta más conveniente parametrizar el ruido como un *multiplicador* adimensional σ_{mult}, de forma que el ruido efectivo se calcula como $\sigma_{\text{abs}} = \sigma_{\text{mult}} \cdot \mathbb{E}[\Delta f]$, donde:

$$\mathbb{E}[\Delta f] = \frac{2C}{\mathbb{E}[|\mathcal{C}_t|]} = \frac{2C}{q \cdot |\mathcal{C}|},$$

es la sensibilidad media del mecanismo. Esta convención es la habitual en bibliotecas estándar de DP como `Opacus`, que suponen que la sensibilidad es 1. De esta manera, sustituyendo esta relación en la Ecuación (3.7), el presupuesto de privacidad por aplicación del mecanismo que calcula `Opacus` se aproxima como:

$$\mathbb{E}[\varepsilon(\alpha)] = \frac{2\alpha C^2}{\mathbb{E}[|\mathcal{C}_t|^2] \cdot \sigma_{\text{abs}}^2} \approx \frac{2\alpha C^2}{(q \cdot |\mathcal{C}|)^2 \cdot (\sigma_{\text{mult}} \cdot 2C/q \cdot |\mathcal{C}|)^2} = \frac{\alpha}{2\sigma_{\text{mult}}^2}. \tag{3.9}$$

Esta expresión muestra que ε depende únicamente del multiplicador σ_{mult} cuando $|\mathcal{C}|$ es suficientemente grande, independientemente de la sensibilidad. En el resto del trabajo, los resultados experimentales se reportan en términos de σ_{abs}, dado que es la magnitud que aparece directamente en la expresión teórica de $\varepsilon(\alpha)$ y permite interpretar el efecto del ruido sobre la utilidad de manera más directa.

4

Pruebas y Resultados

Este capítulo presenta los experimentos realizados para evaluar el sistema propuesto. Todos los experimentos presentados en este capítulo son reproducibles a partir del repositorio público [32], que incluye los scripts de entrenamiento, evaluación y generación de resultados utilizados.

Los experimentos se articulan en torno a preguntas de investigación, que actúan como hipótesis:

P1. ¿Cómo afecta el tamaño del dataset a la utilidad del modelo de factorización de matrices?

P2. ¿Cuál es el coste en utilidad de adoptar un esquema de aprendizaje federado a nivel de usuario frente al modelo centralizado?

P3. ¿Cómo afecta el número de clientes de la federación a la utilidad del sistema?

P4. ¿En qué medida degrada la privacidad diferencial la utilidad del sistema federado?

P5. ¿Qué garantías de privacidad se pueden alcanzar manteniendo una utilidad razonable?

La pregunta **P1** se aborda en la Sección 4.1. Las preguntas **P2** y **P3** se responden en la Sección 4.2. Las preguntas **P4** y **P5** se argumentan en la Sección 4.3.

La medida de utilidad empleada es el RMSE, definida en la Ecuación (2.15). Todos los experimentos realizados se promedian sobre 5 semillas con el objetivo de aumentar la fiabilidad de los resultados. Los experimentos que implican una selección de hiperparámetros (véase Sección 4.1.2 y Sección 4.2.1) se llevan a cabo siguiendo el procedimiento de validación empírica descrito en la Sección 2.4.

4.1. Configuración Experimental

Esta sección describe la configuración experimental adoptada para el modelo de factorización de matrices centralizado, permitiendo establecer una base comparativa para los experimentos llevados a cabo en las secciones subsiguientes. En primer lugar se presentan los conjuntos de datos utilizados y el proceso de partición de los datos.

4.1.1. Conjuntos de datos

Los experimentos se llevan a cabo sobre dos conjuntos de datos de la familia *MovieLens* [35, 36], ampliamente utilizados como referencia en la evaluación de sistemas de recomendación. *MovieLens* es una colección de conjuntos de datos de valoraciones explícitas de películas recopilados por el *GroupLens Research Lab* de la Universidad de Minnesota, en los que los usuarios asignan puntuaciones numéricas a películas que han visto. Gracias a su disponibilidad pública y su variedad de tamaños constituyen una referencia habitual para la comparación de modelos de recomendación.

El primer conjunto de datos utilizado es *MovieLens 100K* (ml-100k), que contiene $100\ 000$ valoraciones realizadas por 943 usuarios sobre $1\ 682$ películas, en una escala entera de 1 a 5. Es el conjunto de datos de referencia más clásico de la familia *Movielens*, y presenta una densidad del $6{,}30\,\%$, la más alta de los dos conjuntos utilizados. El conjunto ml-100k garantiza una cantidad mínima de 20 interacciones por usuario en el conjunto de entrenamiento.

El segundo conjunto de datos utilizado es *MovieLens 1M* (ml-1m), que contiene $1\ 000\ 209$ valoraciones realizadas por $6\ 040$ usuarios sobre $3\ 706$ películas, también en una escala entera de 1 a 5. Su mayor volumen de datos lo convierten en un conjunto de datos más exigente computacionalmente, pero también más representativo de escenarios reales de recomendación a escala.

La Tabla 4.1 resume las principales características de ambos conjuntos de datos, incluyendo la densidad de la matriz de interacciones, definida como el cociente entre el número de valoraciones observadas y el total de interacciones posibles entre usuarios e ítems.

	Usuarios	Ítems	Interacciones	Densidad	Media interacciones por usuario
ml-100k	943	1 682	100 000	6,30 %	106
ml-1m	6 040	3 706	1 000 209	4,47 %	166

Tabla 4.1: Principales características de los conjuntos de datos ml-100k y ml-1m.

Los datos de cada conjunto se dividen en tres subconjuntos: entrenamiento, validación y test, siguiendo una estrategia de partición por usuario. Concretamente, para cada usuario se barajan aleatoriamente sus interacciones y se asigna el $70\,\%$ al conjunto de entrenamiento, el $15\,\%$ al de validación y el $15\,\%$ restante al de test. Esta estrategia garantiza que todos los usuarios están representados en los tres subconjuntos, lo cual es importante en la transición al sistema federado. La Tabla A.1 del apéndice muestra el número de interacciones resultante en cada subconjunto para ambos datasets.

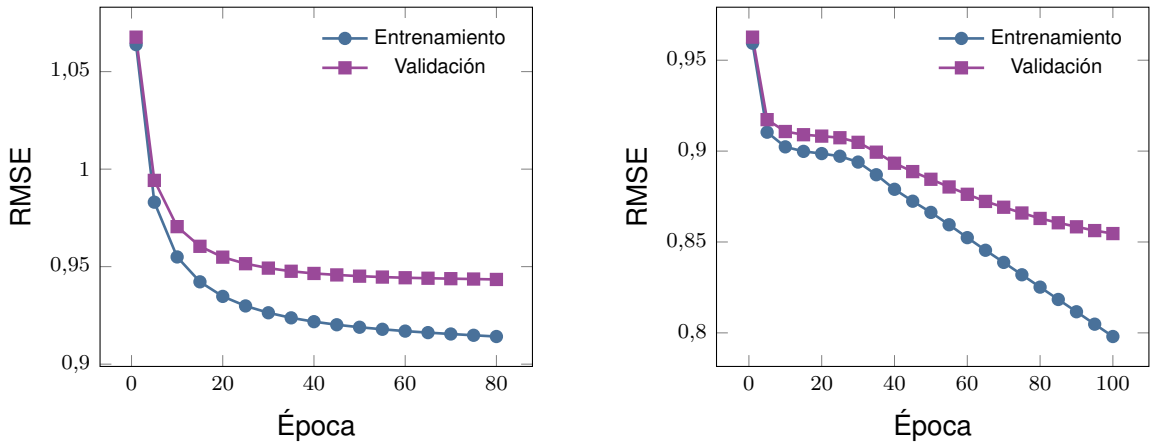
4.1.2. Factorización de matrices centralizado

Los hiperparámetros sujetos a selección en el modelo de factorización de matrices centralizado son la dimensión del espacio latente k, la tasa de aprendizaje η, el coeficiente de regularización λ, el tama-

ño de *minibatch* B y el número de épocas de entrenamiento E. Estos hiperparámetros se seleccionan optimizando el RMSE de validación, y se fijarán posteriormente para los modelos federados estudiados en secciones posteriores con el objetivo de garantizar una comparación justa entre modelos.

La Tabla A.2(a) del apéndice recoge los resultados para ml-100k. Se selecciona la configuración $k = 32$, $\eta = 0{,}05$, $\lambda = 0{,}001$, $B = 64$, $E = 80$, que alcanza el menor RMSE de validación entre las configuraciones con una brecha entrenamiento-validación moderada, evitando el sobreajuste observado en configuraciones con menor tamaño de *minibatch*. La Tabla A.2(b) del apéndice recoge los resultados para ml-1m, donde se selecciona la configuración $k = 64$, $\eta = 0{,}1$, $\lambda = 0{,}001$, $B = 64$, $E = 100$, que obtiene el menor RMSE de validación con una brecha entrenamiento-validación razonable.

La Figura 4.1 muestra la evolución del RMSE de entrenamiento y validación a lo largo de las épocas para la configuración seleccionada en cada dataset. En ambos casos se observa una convergencia estable sin signos de sobreajuste pronunciado, lo que valida la elección de los hiperparámetros.



(a) ml-100k

(b) ml-1m

Figura 4.1: Evolución del RMSE de entrenamiento y validación del modelo factorización de matrices centralizado para la configuración seleccionada en cada dataset.

La Tabla 4.2 recoge los resultados finales sobre el conjunto de test para el modelo de factorización de matrices centralizado con la configuración seleccionada en cada dataset.

	ml-100k	ml-1m
MF	0,9379	0,8572

Tabla 4.2: Resultados finales sobre el conjunto de test para el modelo factorización de matrices centralizado.

Los resultados muestran un RMSE notablemente superior en ml-100k respecto a ml-1m. Esta diferencia se atribuye principalmente a la escasez de interacciones por usuario en ml-100k: aunque este conjunto presenta una densidad ligeramente superior, cada usuario solo dispone de aproximadamente

74 interacciones de entrenamiento, lo que constituye una señal insuficiente para estimar con precisión los factores latentes de usuario. Este efecto se ve agravado por el ruido inherente al problema de predicción de valoraciones. En contraste, ml-1m ofrece una media de interacciones por usuario considerablemente mayor, lo que proporciona mayor señal al entrenar el modelo.

4.2. Efecto de la federación

En esta sección se analiza la extensión del modelo centralizado al paradigma de aprendizaje federado. Los experimentos se llevan a cabo sobre la configuración de hiperparámetros seleccionada en la Sección 4.1.2. En primer lugar se estudia la selección del número de épocas locales por ronda, hiperparámetro específico del modelo federado. A continuación se compara la utilidad del modelo federado frente al modelo centralizado. En ambas subsecciones, la federación se configura a nivel de usuario: cada cliente del esquema federado corresponde a un único usuario, de forma que el número de clientes es igual al número de usuarios del dataset. Por último, se analiza el efecto de variar el número de clientes de la federación agrupando múltiples usuarios por cliente.

4.2.1. Selección de épocas locales

En el modelo federado se introduce un hiperparámetro adicional, el número de épocas locales por ronda LE, que controla el grado de optimización local realizada por cada cliente antes de enviar sus actualizaciones al servidor. El ratio de muestreo de clientes por ronda se fija a $q = 0{,}1$, valor habitual en la literatura [4], que implica que en cada ronda se selecciona aproximadamente el $10\,\%$ de los clientes disponibles.

La Tabla A.3 del apéndice recoge los resultados de la búsqueda del número de épocas locales por ronda LE para el modelo federado en ambos datasets. Se sigue el mismo procedimiento que en la Sección 4.1.2. En ambos casos se selecciona $LE = 16$, valor que minimiza el RMSE de validación. Cabe señalar que la tendencia sugiere que valores de LE superiores a 16 podrían mejorar adicionalmente el RMSE. Sin embargo, el coste computacional de cada ronda de entrenamiento crece linealmente con LE, lo que propone un compromiso entre utilidad y coste, cuya exploración escapa el marco de este trabajo.

La Figura B.1 del apéndice muestra la evolución del RMSE de entrenamiento y validación a lo largo de las rondas para la configuración federada seleccionada en cada dataset.

4.2.2. Comparación con el modelo centralizado

La Tabla 4.3 recoge los resultados finales sobre el conjunto de test para el modelo de factorización de matrices federado con la configuración seleccionada en cada dataset.

	ml-100k	ml-1m
FL	$1{,}0023$	$0{,}9614$

Tabla 4.3: Resultados finales sobre el conjunto de test para el modelo factorización de matrices federado.

La transición del modelo factorización de matrices centralizado al modelo federado conlleva un incremento del RMSE en ambos datasets. Esta degradación es inherente al paradigma federado y se debe a dos factores principales. En primer lugar, el servidor únicamente tiene acceso a los factores de ítem $\mathbf{Q}$ y los sesgos de ítem $\mathbf{b}_{item}$, mientras que los factores de usuario $\mathbf{p}_u$ y los sesgos de usuario $\mathbf{b}_{user}$ permanecen en el dispositivo local de cada cliente y no son optimizados de forma global. En segundo lugar, el proceso de agregación introduce ruido en las actualizaciones de los factores de ítem, ya que cada cliente optimiza localmente sobre un subconjunto reducido de interacciones antes de enviar sus actualizaciones al servidor. Esta degradación en el RMSE es consistente con los resultados reportados en la literatura para modelos de factorización de matrices federados [11, 12].

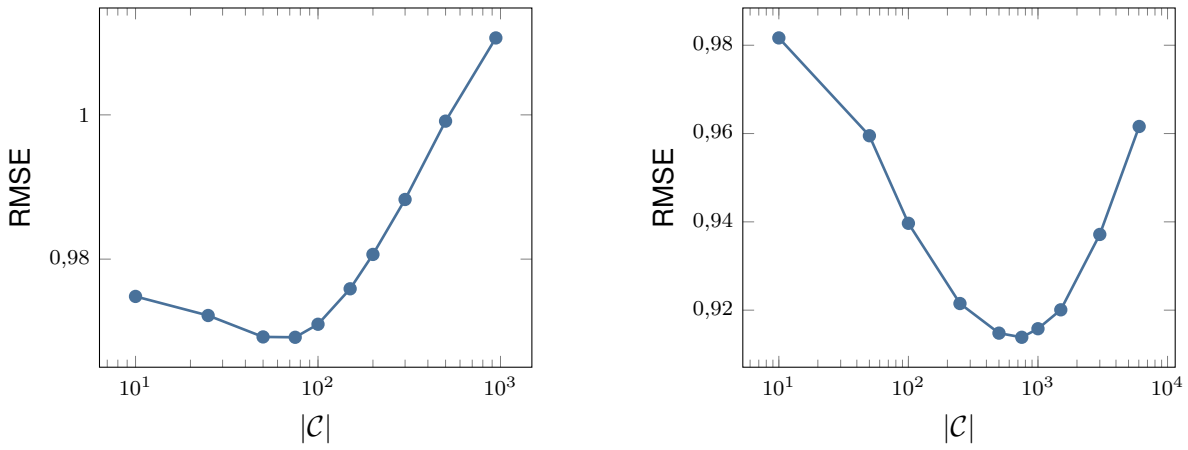
4.2.3. Efecto del número de clientes

En las subsecciones anteriores, el esquema federado se ha configurado con un cliente por usuario. Sin embargo, en escenarios reales es habitual que cada cliente agrupe las interacciones de múltiples usuarios. Para estudiar el efecto de esta configuración sobre la utilidad del sistema, se varía el número de clientes de la federación, manteniendo el resto de hiperparámetros a los valores seleccionados anteriormente. La asignación de usuarios a clientes se realiza de forma aleatoria y uniforme.

La Tabla A.4 del apéndice recoge el RMSE sobre el conjunto de test para cada valor del número de clientes explorado en ambos datasets. La Figura 4.2 muestra la evolución del RMSE sobre el conjunto de test en función del número de clientes.

En ambos datasets el RMSE alcanza un mínimo para un valor del número de clientes. En ml-100k, dicho mínimo se obtiene con $|\mathcal{C}| = 50$ clientes (aproximadamente 19 usuarios por cliente), con un RMSE de $0{,}9692$. En ml-1m, el mínimo se sitúa en $|\mathcal{C}| = 750$ clientes (aproximadamente 8 usuarios por cliente), con un RMSE de $0{,}9139$.

Este comportamiento corresponde al compromiso entre dos efectos opuestos. Un número elevado de clientes implica que cada uno dispone de un volumen reducido de datos locales, lo que limita la calidad de las actualizaciones generadas. Un número reducido de clientes, por el contrario, conlleva



(a) ml-100k

(b) ml-1m

Figura 4.2: Evolución del RMSE en función del número de clientes para ambos datasets.

un menor número de oportunidades de agregación a lo largo del entrenamiento, dado que la tasa de muestreo por ronda q permanece fija. El punto de equilibrio entre ambos efectos depende del número de usuarios disponibles: con $6\ 040$ usuarios en ml-1m frente a 943 en ml-100k, es posible mantener un mayor número de clientes sin reducir excesivamente el volumen de datos locales por cliente, lo que desplaza el óptimo hacia configuraciones con más clientes.

La comparación con el modelo factorización de matrices centralizado muestra que el coste de federación se reduce si se selecciona el número adecuado de clientes. En ml-100k, la diferencia disminuye de $0{,}064$ puntos en la configuración con un cliente por usuario a $0{,}031$ puntos con $|\mathcal{C}| = 50$. En ml-1m, la reducción es de $0{,}104$ a $0{,}057$ puntos con $|\mathcal{C}| = 750$.

No obstante, esta mejora en utilidad conlleva una contrapartida en términos de privacidad. Agrupar los datos de varios usuarios en un mismo cliente implica que dicho cliente tiene acceso al conjunto completo de interacciones de los usuarios asignados, de forma que un ataque activo sobre un único cliente comprometería los datos de todos ellos simultáneamente. Adicionalmente, en escenarios con un volumen de datos elevado, el almacenamiento local puede resultar inviable, lo que requiere recurrir a infraestructuras compartidas, exponiendo potencialmente los datos de los usuarios a accesos no autorizados.

4.3. Efecto de la privacidad diferencial

En esta sección se analiza el efecto de la incorporación de privacidad diferencial sobre la utilidad del sistema de recomendación federado. Los experimentos estudian el efecto conjunto de dos dimensiones: la magnitud del ruido gaussiano añadido, parametrizado por su desviación estándar absoluta σ_{abs}, y el número de clientes de la federación $|\mathcal{C}|$.

El umbral de recorte se establece en $C = 1{,}0$, valor estándar en la literatura [25] que determina la sensibilidad global del mecanismo. El presupuesto de privacidad ε se calcula mediante el contador de privacidad de `Opacus` (véase Sección 3.4.3) para un valor de δ fijado siguiendo el criterio estándar $\delta < 1/n$, donde n es el número de usuarios del dataset, de forma que la probabilidad de fallo del mecanismo sea despreciable respecto al tamaño de la población. Aplicando este criterio resulta $\delta = 10^{-3}$ para ml-100k y $\delta = 10^{-4}$ para ml-1m.

Las configuraciones de $|\mathcal{C}|$ exploradas se seleccionaron para cubrir el rango completo de valores estudiado en la Sección 4.2. Para ml-100k se consideran $|\mathcal{C}| \in \{50, 200, 500, 943\}$ y para ml-1m, $|\mathcal{C}| \in \{500, 1\,000, 3\,000, 6\,040\}$.

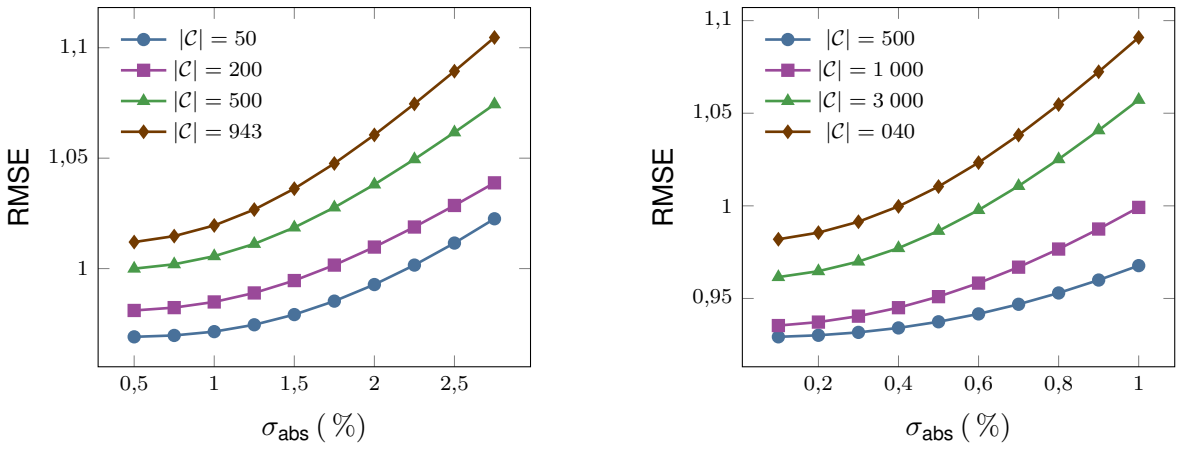
Para cada configuración de $|\mathcal{C}|$, el parámetro σ_{abs} (véase Sección 3.4.3) se mantiene constante entre configuraciones, mientras que el multiplicador de ruido σ_{mult} se ajusta en función de $|\mathcal{C}|$ según la relación $\sigma_{\text{mult}} = \sigma_{\text{abs}} \cdot |\mathcal{C}| \cdot q/(2C)$. Los valores de σ_{abs} explorados se han seleccionado de forma que el multiplicador de ruido σ_{mult} no explotara la utilidad del modelo para ninguna configuración de $|\mathcal{C}|$. Los valores concretos de σ_{abs} explorados y los resultados completos pueden consultarse en las Tablas A.5 y A.6 del apéndice.

4.3.1. Comparación de utilidad

La Figura 4.3 muestra la evolución del RMSE sobre el conjunto de test en función de σ_{abs} para cada configuración de $|\mathcal{C}|$ en ambos datasets. Para todas las configuraciones y ambos datasets, el RMSE aumenta de forma monótona con σ_{abs}: a mayor ruido, mayor degradación de la utilidad. Esta degradación en el RMSE es consistente con los resultados reportados en la literatura para sistemas de recomendación federados con privacidad diferencial [8, 11, 12].

En ml-1m, la degradación media del RMSE sobre el rango de σ_{abs} explorado es de $0{,}077$ puntos, frente a $0{,}070$ puntos en ml-100k, a pesar de que el rango de σ_{abs} es aproximadamente cinco veces más estrecho. Esto se debe a que ml-1m permite configuraciones con un número de clientes significativamente mayor. Para estas configuraciones las contribuciones agregadas ya constituyen una señal débil de por sí, con lo cual presentan mayor sensibilidad al nivel de ruido absoluto.

Respecto al efecto de $|\mathcal{C}|$, las curvas muestran que un mayor número de clientes implica un RMSE más elevado para el mismo σ_{abs}, en línea con lo observado en la Sección 4.2: más clientes conlleva menos datos locales por cliente y actualizaciones de menor calidad, efecto que se ve agravado bajo DP al actuar el ruido sobre una señal más ruidosa.

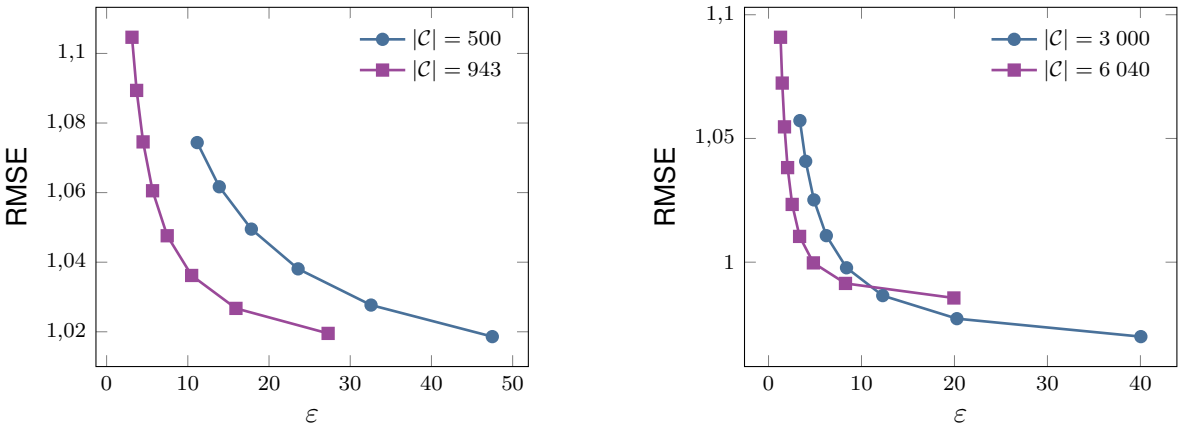


(a) ml-100k (b) ml-1m

Figura 4.3: Evolución del RMSE sobre el conjunto de test en función de σ_{abs} para cada configuración de $|\mathcal{C}|$.

4.3.2. Compromiso privacidad–utilidad

La Figura 4.4 muestra el RMSE sobre el conjunto de test en función del presupuesto de privacidad ε para las configuraciones de $|\mathcal{C}|$ que alcanzan $\varepsilon \leq 50$. A partir de este umbral las garantías de privacidad tienen poco significado práctico [37]. En el eje horizontal, valores menores de ε corresponden a garantías de privacidad más fuertes.



(a) ml-100k (b) ml-1m

Figura 4.4: Evolución del RMSE sobre el conjunto de test en función del presupuesto de privacidad ε para cada configuración de $|\mathcal{C}|$.

Para todas las configuraciones representadas, el RMSE decrece al aumentar ε, a mayor presupuesto de privacidad (menor protección), mayor utilidad del sistema. El trade-off es cuantificable y consistente en ambos datasets. La no linealidad de la curva se explica por la relación entre σ y ε derivada

del mecanismo gaussiano. Como muestra la Ecuación (2.12), el presupuesto de privacidad $\varepsilon(\alpha)$ es inversamente proporcional a σ^2_{abs}. En consecuencia, en la región de ε elevado, pequeños incrementos de σ permiten obtener garantías de privacidad significativamente más estrictas a un coste moderado en utilidad. Sin embargo, en la región de ε pequeño, reducir ε adicionalmente requiere valores de σ cada vez mayores.

En la literatura de DP aplicada al aprendizaje automático, se consideran razonables presupuestos de privacidad en el rango $\varepsilon \leq 10$ [37], siendo $\varepsilon \leq 1$ un régimen de privacidad fuerte. Dentro de este rango, en ml-100k únicamente la configuración $|\mathcal{C}| = 943$ satisface garantías razonables, con RMSE entre $1{,}036$ ($\varepsilon = 10{,}47$) y $1{,}105$ ($\varepsilon = 3{,}13$). En ml-1m, las configuraciones $|\mathcal{C}| = 3\,000$ y $|\mathcal{C}| = 6\,040$ alcanzan $\varepsilon \leq 10$, con RMSE entre $0{,}991$ ($\varepsilon = 8{,}28$) y $1{,}091$ ($\varepsilon = 1{,}29$). En ambos datasets, las configuraciones con menor número de clientes no son capaces de alcanzar garantías de privacidad razonables dentro del rango de σ_{abs} explorado.

Este fenómeno está justificado matemáticamente por la Ecuación 3.7, que muestra que el presupuesto de privacidad $\varepsilon(\alpha)$ es inversamente proporcional a $|\mathcal{C}|^2$ y a σ^2_{abs}. En consecuencia, reducir el número de clientes obliga a incrementar el ruido proporcionalmente para mantener el mismo presupuesto de privacidad, con el consiguiente deterioro de la utilidad del modelo.

Esta relación tiene además una interpretación intuitiva. El objetivo de la DP es limitar la variabilidad del mecanismo ante datasets adyacentes (véase Definición 2.2). La sensibilidad, recogida en la Ecuación 3.6, decrece con el número de clientes: con más clientes, cada contribución individual tiene un peso relativo menor en la agregación, lo que reduce la influencia de un único cliente sobre la salida del mecanismo. Por tanto, se requiere menos ruido para alcanzar el mismo nivel de protección, mejorando el compromiso privacidad-utilidad.

Cabe destacar que las garantías de privacidad se aplican a nivel de cliente, no de usuario. La DP garantiza que la salida del mecanismo no permite distinguir si un cliente completo participó o no en el entrenamiento, pero no protege la información de los usuarios individuales que ese cliente agrupa. Intuitivamente, el mecanismo oculta la presencia de un cliente en particular en el conjunto de datos: si dicho cliente fuese expuesto, revelaría la participación de todos los individuos que agrupa. En consecuencia, el mismo presupuesto ε a nivel de cliente corresponde a una garantía más débil cuanto más usuarios agrupan los clientes.

4.4. Discusión

Los resultados obtenidos permiten responder de forma directa a las preguntas de investigación planteadas al inicio del capítulo. En respuesta a **P1**, los resultados muestran que el tamaño del dataset influye significativamente en la utilidad del modelo: ml-1m, con una media de 166 interacciones por usuario, obtiene un RMSE de $0{,}857$ sobre el conjunto de test, frente a $0{,}938$ de ml-100k, que dispone

de apenas 106 interacciones por usuario.

En respuesta a **P2**, la transición al esquema federado conlleva un incremento del RMSE de $0{,}064$ puntos en ml-100k y de $0{,}104$ puntos en ml-1m, atribuible a las limitaciones inherentes del paradigma federado descritas anteriormente. En respuesta a **P3**, el número de clientes de la federación constituye un parámetro determinante en la utilidad del sistema. En ml-100k, el óptimo se alcanza con $|\mathcal{C}| = 50$ clientes, reduciendo la brecha con el modelo centralizado de $0{,}064$ a $0{,}031$ puntos. En ml-1m, el óptimo se sitúa en $|\mathcal{C}| = 750$ clientes, reduciendo la brecha de $0{,}104$ a $0{,}057$ puntos.

En respuesta a **P4**, la incorporación de DP introduce una degradación monótona del RMSE con el nivel de ruido σ_{abs}, cuyo impacto depende del número de clientes. En ml-100k, la degradación máxima sobre las configuraciones exploradas es de $0{,}093$ puntos para $|\mathcal{C}| = 943$; en ml-1m alcanza $0{,}109$ puntos para $|\mathcal{C}| = 6\,040$. En respuesta a **P5**, únicamente las configuraciones con un número elevado de clientes son capaces de alcanzar presupuestos de privacidad razonables ($\varepsilon \leq 10$) manteniendo una utilidad aceptable. En ml-100k, la configuración $|\mathcal{C}| = 943$ logra $\varepsilon = 10{,}47$ con RMSE de $1{,}0362$. En ml-1m, la configuración $|\mathcal{C}| = 6\,040$ alcanza $\varepsilon = 8{,}28$ con RMSE de $0{,}9914$.

La Tabla 4.4 sintetiza las configuraciones para cada escenario considerado en este capítulo.

Escenario	Dataset	$\lvert\mathcal{C}\rvert$	RMSE	ε
Máxima utilidad (centralizado)	ml-100k	–	$0{,}938$	–
	ml-1m	–	$0{,}857$	–
Federado sin privacidad	ml-100k	50	$0{,}969$	–
	ml-1m	750	$0{,}914$	–
Federado con privacidad razonable	ml-100k	943	$1{,}0362$	$10{,}47$
	ml-1m	$6\,040$	$0{,}9914$	$8{,}28$
Federado con privacidad fuerte	ml-100k	943	$1{,}105$	$3{,}13$
	ml-1m	$6\,040$	$1{,}091$	$1{,}29$

Tabla 4.4: Configuraciones para cada escenario de uso, con el RMSE y el presupuesto de privacidad ε asociados.

5

Conclusiones y Trabajo Futuro

Este trabajo ha presentado el diseño, la implementación y evaluación de un sistema de recomendación federado basado en factorización de matrices con privacidad diferencial. El Capítulo 2 ha comenzado estudiando en detalle los fundamentos teóricos y prácticos que conciernen los sistemas de recomendación (*Recommender Systems*, RS), y en particular la factorización de matrices (*Matrix Factorization*, MF); el aprendizaje federado (*Federated Learning*, FL), y en concreto su aplicación a los sistemas de recomendación; y la privacidad diferencial (*Differential Privacy*, DP), estableciendo formalmente su marco matemático y situándolo en los sistemas de recomendación federados. A continuación, el Capítulo 3 ha establecido el diseño e implementación del modelo propuesto en tres fases incrementales. En la primera fase se desarrollaba el modelo centralizado de MF, incorporando mejoras adicionales respecto al modelo estándar. En la segunda fase se adaptaba la arquitectura del sistema para implementar una estructura de cliente-servidor que permitiese la transición al sistema con FL. La adición de la DP se llevaba a cabo en la tercera fase, junto con el cálculo explícito del presupuesto de privacidad final del modelo. Finalmente, el Capítulo 4 ha evaluado el modelo propuesto sobre los conjuntos de datos *MovieLens 100K* y *MovieLens 1M*, estándar en la literatura sobre la evaluación de RS, que ha permitido analizar su comportamiento en datasets con estructuras variadas. Se ha estudiado el efecto de la federación, representada como la variación del número de clientes en el esquema federado, y el efecto de la privacidad diferencial, a través de los parámetros σ_{abs} y ε.

El esquema de aprendizaje federado introduce un coste en utilidad respecto al modelo centralizado. El detrimento en utilidad es inherente a la estructura del FL ya que la agregación de las actualizaciones locales de los clientes introduce ruido, ocasionado por la potencial convergencia de las actualizaciones a mínimos distintos. Los experimentos exhiben que el número de clientes $|\mathcal{C}|$ constituye un factor determinante en el efecto de federar el sistema, con un óptimo no trivial para cada conjunto de datos que representa un equilibrio entre dos factores. Con pocos clientes, el número esperado de clientes seleccionados por ronda $\mathbb{E}[|\mathcal{C}_t|] = |\mathcal{C}| \cdot q$ es reducido. Por el contrario, cuando el número de clientes crece, los datos locales que agrupa cada cliente son menores, y consecuentemente las actualizaciones pierden calidad.

La incorporación de privacidad diferencial introduce una degradación monótona de la utilidad con el nivel de ruido σ_{abs}. Los experimentos han demostrado que para ambos conjuntos de datos únicamente

las configuraciones con un número elevado de clientes son capaces de obtener garantías de privacidad fuertes manteniendo una utilidad razonable. Esto se debe a que el aumento del número de clientes reduce la sensibilidad del mecanismo, ofreciendo mayor DP para un mismo σ_{abs}. Por otro lado, las garantías ofrecidas por el sistema propuesto son a nivel de cliente, no de usuario. Cuando un cliente agrupa varios usuarios, el ε reportado garantiza privacidad más débil a nivel de usuario, limitando la interpretabilidad directa de los resultados para clientes con varios usuarios.

Como se ha visto en este trabajo, los sistemas de recomendación se basan en datos que reflejan directamente las preferencias y comportamientos de los usuarios, cuya naturaleza intrínsecamente sensible requiere tratamiento por modelos capaces de combinar utilidad y privacidad. Los resultados obtenidos demuestran que la combinación de aprendizaje federado y privacidad diferencial constituye una solución viable para el despliegue de sistemas de recomendación con garantías de privacidad, requiriendo de un cuidado especial en la selección de la configuración. Asimismo, los experimentos revelan que el compromiso entre privacidad y utilidad es cuantificable y por lo tanto es posible equilibrar ambas tomando decisiones informadas en función del contexto de despliegue.

El presente trabajo ofrece varias líneas de investigación que pueden extender los resultados obtenidos. En primer lugar, la evaluación de la utilidad del modelo se efectúa sobre el error cuadrático medio, métrica estándar en el aprendizaje supervisado. Sin embargo, en el contexto de RS existen métricas más relevantes en la literatura actual (véase Sección 2.4), que añadirían un grado mayor de robustez a los resultados. En esta misma línea, resultaría natural extender la evaluación a otros conjuntos de datos relevantes en la literatura, como *Amazon Reviews* [38]. En segundo lugar, los experimentos de federación asumen una independencia y distribución idéntica de los datos de los usuarios, seleccionando los usuarios correspondientes a cada cliente aleatoriamente. En la práctica los usuarios se agrupan por razones geográficas o por dispositivo [18], por lo que puede resultar de interés estudiar el efecto de datos no independientes e idénticamente distribuidos sobre la convergencia y utilidad del modelo. En tercer lugar, se podrían incorporar mejoras en el mecanismo de DP, comparando mecanismos alternativos al mecanismo gaussiano [23] o estudiar el efecto del umbral de recorte C como hiperparámetro en lugar de valor fijo. En cuarto lugar, los experimentos se podrían extender para evaluar la privacidad del modelo en un escenario práctico, sometiéndolo a ataques de inferencia de presencia [7]. Se han llevado a cabo algunas pruebas preliminares en esta dirección. Finalmente, el paradigma desarrollado en este trabajo se podría extender a modelos de RS más expresivos, relevantes en la literatura moderna, como el filtrado colaborativo neuronal [39] o los modelos basados en grafos [40].

Bibliografía

[1] J. Bennett and S. Lanning, "The Netflix prize," in *Proceedings of KDD Cup and Workshop*, 2007.

[2] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy*, pp. 111–125, 2008.

[3] European Parliament and Council of the European Union, "Regulation (EU) 2016/679 (General Data Protection Regulation)." Official Journal of the European Union, L 119, 4 May 2016, 2016.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, PMLR, 2017.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference (TCC)*, pp. 265–284, 2006.

[6] F. Ricci, L. Rokach, and B. Shapira, eds., *Recommender Systems Handbook*. Springer US, 2022.

[7] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017.

[8] Z. Yang and T. Mbodza, "Enhancing privacy and security in recommender systems through federated learning and differential privacy," in *2024 12th International Conference on Information Systems and Computing Technology (ISCTech)*, pp. 1–6, IEEE, 2024.

[9] T. Li, L. Song, and C. Fragouli, "Federated recommendation system via differential privacy," *arXiv preprint arXiv:2005.06670*, 2020.

[10] R. Liu, Y. Cao, Y. Wang, L. Lyu, Y. Chen, and H. Chen, "Privaterec: Differentially private model training and online serving for federated news recommendation," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, (Long Beach, CA, USA), pp. 4539–4549, Association for Computing Machinery, 2023.

[11] Q. Wang, H. Yin, T. Chen, J. Yu, A. Zhou, and X. Zhang, "Fast-adapting and privacy-preserving federated recommender system," *The VLDB Journal*, vol. 31, no. 4, pp. 877–896, 2022.

[12] Z. Xu, C. Chu, and S. Song, "An effective federated recommendation framework with differential privacy," *Electronics*, vol. 13, no. 8, 2024.

[13] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous systems," in *OSDI*, 2016.

[14] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[15] J. Bradbury *et al.*, "Jax: composable transformations of python+numpy programs," 2018. Google Research.

[16] N. Hug, "Surprise: A Python library for recommender systems," vol. 5, p. 2174, 2020.

[17] M. D. Ekstrand, “LensKit for Python: Next-generation software for recommender systems experiments,” in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 2999–3006, 2020.

[18] P. Kairouz, H. B. McMahan, *et al.*, “Advances and open problems in federated learning,” *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[19] Flower AI Documentation, “Differential privacy,” 2026. Consultado el 19 de mayo de 2026.

[20] S. Wang and T.-H. Chang, “Federated matrix factorization: Algorithm design and application to data clustering,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 1625–1640, 2022.

[21] Z. Liu, L. Yang, Z. Fan, H. Peng, and P. S. Yu, “Federated social recommendation with graph neural network,” *ACM Transactions on Intelligent Systems and Technology*, vol. 13, pp. 1–24, Aug. 2022.

[22] N. Ponomareva, H. Hazimeh, A. Kurakin, Z. Xu, C. Denison, H. B. McMahan, S. Vassilvitskii, S. Chien, and A. Thakurta, “How to dp-fy ml: A practical guide to machine learning with differential privacy,” *arXiv preprint arXiv:2303.00654*, 2023.

[23] E. Cowan, M. Shoemate, and M. Pereira, *Hands-On Differential Privacy: Introduction to the Theory and Practice Using OpenDP*. O’Reilly Media, 2024. Consultado el 19 de mayo de 2026.

[24] I. Mironov, “Rényi differential privacy,” in *Proceedings of the 30th IEEE Computer Security Foundations Symposium (CSF)*, pp. 263–275, IEEE, 2017.

[25] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in *ACM Conference on Computer and Communications Security (CCS)*, pp. 308–318, 2016.

[26] Y. Zhu and Y.-X. Wang, “Poisson subsampled Rényi differential privacy,” in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 7634–7642, PMLR, June 2019.

[27] P. Müllner, E. Lex, M. Schedl, and D. Kowald, “Differential privacy in collaborative filtering recommender systems: A review,” *Frontiers in Big Data*, vol. 6, p. 1249997, 2023.

[28] X. Li, W. Zhou, A. Ul Haq, and S. Khan, “Ldpmf: Local differential privacy enhanced matrix factorization for advanced recommendation,” *Knowledge-Based Systems*, vol. 309, p. 112892, 2025.

[29] Google AI, “Tensorflow privacy,” 2019. Available at https://github.com/tensorflow/privacy.

[30] Meta AI, “Opacus: User-friendly differential privacy library in pytorch,” 2021. Available at https://opacus.ai.

[31] N. Holohan *et al.*, “Diffprivlib: The ibm differential privacy library,” *arXiv preprint arXiv:1907.02444*, 2019.

[32] A. de Santos de la Pedraja, “Federated Matrix Factorization with Differential Privacy.” `https://github.com/Alvarixxs/fedmf`, 2026.

[33] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[34] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT)*, pp. 177–186, Springer,

2010.

[35] G. Research, "MovieLens datasets." https://grouplens.org/datasets/movielens/, 2023.

[36] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 1–19, 2015.

[37] A. Blanco-Justicia, D. Sánchez, J. Domingo-Ferrer, and K. Muralidhar, "A critical review on the use (and misuse) of differential privacy in machine learning," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–16, 2022.

[38] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 507–517, 2016.

[39] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International World Wide Web Conference*, pp. 173–182, 2017.

[40] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference*, pp. 639–648, 2020.

Acrónimos

CDP privacidad diferencial central.

DP privacidad diferencial.

FedAvg agregación federada.

FL aprendizaje federado.

FRS sistemas de recomendación federados.

GDPR Reglamento General de Protección de Datos.

LDP privacidad diferencial local.

MF factorización de matrices.

ml-100k *MovieLens 100K*.

ml-1m *MovieLens 1M*.

RDP privacidad diferencial de Rényi.

RMSE raíz del error cuadrático medio.

RS sistemas de recomendación.

SGD descenso por gradiente estocástico.

Apéndices

A

Tablas Complementarias

A.1. Estadísticas de los datos

La Tabla A.1 recoge el número de interacciones en cada subconjunto tras aplicar la partición 70/15/15.

	ml-100k	ml-1m
Entrenamiento	69 572	697 378
Validación	14 577	147 221
Test	15 851	155 610

Tabla A.1: Número de interacciones en cada subconjunto tras aplicar la partición 70/15/15 por usuario.

A.2. Hiperparámetros

La Tabla A.2 recoge los resultados completos de la búsqueda de hiperparámetros del modelo factorización de matrices centralizado. La Tabla A.3 recoge los resultados de la búsqueda del número de épocas locales para el modelo federado.

A.3. Resultados completos

La Tabla A.4 recoge los resultados del experimento de efecto de la federación. Las Tablas A.5 y A.6 recogen los resultados del experimento de efecto de la privacidad diferencial.

Dim. latente	Tasa aprend.	Regulariz.	*Minibatch*	Épocas	Entrenamiento	Validación
16	0,05	0,01	64	80	0,9500	0,9695
32	**0,05**	**0,001**	**64**	**80**	**0,9141**	**0,9434**
32	0,05	0,001	32	80	0,8170	0,9197
32	0,05	0,001	64	100	0,9111	0,9424
32	0,1	0,001	64	80	0,8770	0,9277
64	0,05	0,001	32	80	0,7801	0,9173
64	0,05	0,001	64	80	0,9131	0,9432
64	0,05	0,001	64	100	0,9088	0,9418
64	0,05	0,0001	32	80	0,6436	0,9307
64	0,01	0,001	64	80	0,9403	0,9591
128	0,05	0,001	32	80	0,7329	0,9147
128	0,05	0,001	32	100	0,6030	0,9190
128	0,05	0,001	64	80	0,9111	0,9428
32	0,01	0,01	32	80	0,9389	0,9587

(a) ml-100k

Dim. latente	Tasa aprend.	Regulariz.	*Minibatch*	Épocas	Entrenamiento	Validación
64	0,05	0,001	16	100	0,4467	0,9512
64	0,05	0,001	32	100	0,6217	0,8577
64	0,05	0,001	64	100	0,8660	0,8844
64	0,05	0,001	128	100	0,9009	0,9101
64	0,05	0,001	64	150	0,8307	0,8652
64	0,05	0,01	16	100	0,8830	0,8968
64	0,05	0,01	64	100	0,9332	0,9399
64	0,01	0,001	32	100	0,8958	0,9064
64	**0,1**	**0,001**	**64**	**100**	**0,7976**	**0,8544**
128	0,05	0,001	16	100	0,2964	0,9339
128	0,05	0,001	64	100	0,8601	0,8811
128	0,05	0,01	32	100	0,9114	0,9199
128	0,05	0,001	64	150	0,8211	0,8621
256	0,05	0,001	64	100	0,8530	0,8777
256	0,05	0,01	32	100	0,9114	0,9199

(b) ml-1m

Tabla A.2: Resultados de la búsqueda de hiperparámetros del modelo factorización de matrices centralizado.

LE	Entrenamiento	Validación
1	1,0333	1,0436
2	1,0244	1,0373
4	1,0169	1,0317
8	1,0066	1,0222
16	**0,9908**	**1,0094**

(a) ml-100k

LE	Entrenamiento	Validación
1	1,0216	1,0291
2	1,0152	1,0236
4	1,0049	1,0140
8	0,9868	0,9990
16	**0,9276**	**0,9592**

(b) ml-1m

Tabla A.3: Resultados de la búsqueda del número de épocas locales por ronda LE para el modelo federado.

$\|\mathcal{C}\|$	RMSE
10	0,9748
25	0,9722
50	**0,9692**
75	0,9692
100	0,9710
150	0,9759
200	0,9807
300	0,9883
500	0,9991
943	1,0107

(a) ml-100k

$\|\mathcal{C}\|$	RMSE
10	0,9817
50	0,9595
100	0,9397
250	0,9215
500	0,9148
750	**0,9139**
1 000	0,9158
1 500	0,9201
3 000	0,9371
6 040	0,9616

(b) ml-1m

Tabla A.4: Resultados del modelo federado variando el número de clientes.

σ_{abs} (%)	RMSE	ε
0,50	0,9691	278 241,58
0,75	0,9697	122 579,36
1,00	0,9715	68 097,58
1,25	0,9746	42 880,30
1,50	0,9792	29 182,03
1,75	0,9853	20 922,40
2,00	0,9928	15 561,58
2,25	1,0016	11 886,23
2,50	1,0116	9 257,27
2,75	1,0226	7 312,21

(a) $|\mathcal{C}| = 50$

σ_{abs} (%)	RMSE	ε
0,50	0,9810	15 639,45
0,75	0,9823	5 862,18
1,00	0,9849	2 475,29
1,25	0,9890	1 084,71
1,50	0,9946	538,63
1,75	1,0016	317,28
2,00	1,0097	215,38
2,25	1,0188	162,06
2,50	1,0286	131,27
2,75	1,0388	104,56

(b) $|\mathcal{C}| = 200$

σ_{abs} (%)	RMSE	ε
0,50	1,0000	1 084,71
0,75	1,0019	257,49
1,00	1,0056	131,27
1,25	1,0112	74,09
1,50	1,0186	47,50
1,75	1,0277	32,56
2,00	1,0381	23,58
2,25	1,0495	17,83
2,50	1,0617	13,88
2,75	1,0744	11,15

(c) $|\mathcal{C}| = 500$

σ_{abs} (%)	RMSE	ε
0,50	1,0120	146,89
0,75	1,0147	55,63
1,00	1,0195	27,28
1,25	1,0267	15,93
1,50	1,0362	10,47
1,75	1,0476	7,46
2,00	1,0605	5,66
2,25	1,0746	4,49
2,50	1,0894	3,70
2,75	1,1046	3,13

(d) $|\mathcal{C}| = 943$

Tabla A.5: Resultados para cada configuración de $|\mathcal{C}|$ y σ_{abs} en ml-100k.

σ_{abs} (%)	RMSE	ε
0,1	0,9293	85 555,91
0,2	0,9301	19 555,91
0,3	0,9317	7 334,32
0,4	0,9341	3 100,71
0,5	0,9374	1 362,48
0,6	0,9417	679,89
0,7	0,9469	403,20
0,8	0,9530	275,82
0,9	0,9599	209,16
1,0	0,9677	170,68

(a) $|\mathcal{C}| = 500$

σ_{abs} (%)	RMSE	ε
0,1	0,9354	19 555,91
0,2	0,9372	3 100,71
0,3	0,9404	679,89
0,4	0,9450	275,82
0,5	0,9510	170,68
0,6	0,9583	106,03
0,7	0,9669	74,15
0,8	0,9767	53,59
0,9	0,9875	40,05
1,0	0,9992	31,10

(b) $|\mathcal{C}| = 1\,000$

σ_{abs} (%)	RMSE	ε
0,1	0,9615	679,89
0,2	0,9647	106,03
0,3	0,9699	40,05
0,4	0,9772	20,26
0,5	0,9865	12,28
0,6	0,9977	8,39
0,7	1,0107	6,22
0,8	1,0252	4,88
0,9	1,0408	4,00
1,0	1,0572	3,38

(c) $|\mathcal{C}| = 000$

σ_{abs} (%)	RMSE	ε
0,1	0,9820	104,30
0,2	0,9855	19,94
0,3	0,9914	8,28
0,4	0,9997	4,82
0,5	1,0104	3,34
0,6	1,0233	2,54
0,7	1,0382	2,05
0,8	1,0547	1,72
0,9	1,0724	1,48
1,0	1,0909	1,29

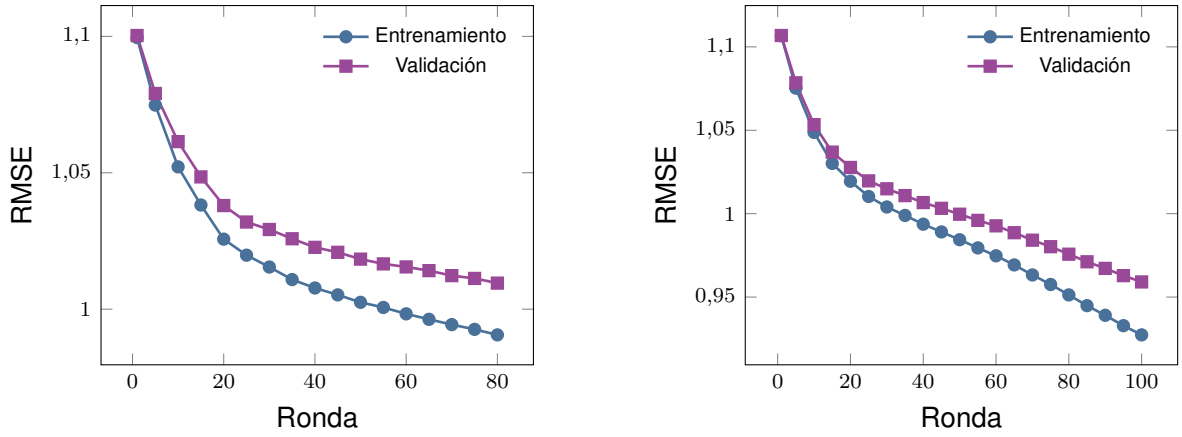
(d) $|\mathcal{C}| = 6\,040$

Tabla A.6: Resultados para cada configuración de $|\mathcal{C}|$ y σ_{abs} en ml-1m.

B

Figuras Complementarias

Las Figura B.1 muestra la evolución del RMSE de entrenamiento y validación a lo largo de las rondas para la configuración federada seleccionada en cada dataset.



(a) ml-100k

(b) ml-1m

Figura B.1: Evolución del RMSE de entrenamiento y validación del modelo federado para la configuración seleccionada en cada dataset.

UÁM
Universidad Autónoma
de Madrid