




User-controlled federated matrix factorization for recommender systems

Vito Walter Anelli¹ · Yashar Deldjoo¹ · Tommaso Di Noia¹ · Antonio Ferrara¹  · Fedelucio Narducci¹

Received: 20 April 2021 / Revised: 1 November 2021 / Accepted: 1 November 2021 /

Published online: 31 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Recommendation services have been extensively adopted in various user-centered applications to help users navigate a vast space of possible choices. In such scenarios, data ownership is a crucial concern since users may not be willing to share their *sensitive* preferences (e.g., visited locations, read books, bought items) with a central server. Unfortunately, data collection is at the basis of modern approaches to the recommendation problem. Decreased users' willingness to share personal information and data protection policies can result in the "data scarcity" dilemma affecting applications such as recommender systems. In the work at hand, we thoroughly study and extend FPL (Federated Pair-wise Learning), a recommendation approach that follows the *Federated Learning* principles. In FPL, users collaborate in training a pair-wise learning to rank factorization model while controlling the amount of sensitive data that leaves their devices. An extensive experimental evaluation reveals the effectiveness of the proposed architecture concerning the accuracy and beyond-accuracy objectives and the impact of disclosed users' information on the quality of the final model. The paper also analyzes the impact of communication costs with the central server on the system's performance by varying local computation and training parallelism. Furthermore, the study investigates the injection of additional biases in the final recommendation that could affect the fairness of the system. The public implementation is available at <https://split.to/sisinflab-fpl>.

✉ Antonio Ferrara
antonio.ferrara@poliba.it

Vito Walter Anelli
vitowalter.anelli@poliba.it

Yashar Deldjoo
yashar.deldjoo@poliba.it

Tommaso Di Noia
tommaso.dinoia@poliba.it

Fedelucio Narducci
fedelucio.narducci@poliba.it

¹ Politecnico di Bari, Bari, Italy

Keywords Federated learning · Collaborative filtering · Pair-wise learning · Matrix factorization

1 Introduction

Recommender Systems (RSs) have emerged as a solution to better support users' decision-making and promote business by recommending novel and personalized items. These models are generally hosted on *centralized* servers and train their models by exploiting massive proprietary and sensitive data. For instance, Collaborative Filtering (CF) models, which have been the mainstream research line in the RS community over the last two decades (McFee et al., 2012; Yuan et al., 2016), need sufficient in-domain interaction data to discover similar behavioral/preference patterns in a user community. In principle, this could result in a grave threat to users' privacy. Moreover, the European Union, the US Congress, and other jurisdictions legislated new disclosure laws in recent years. As an example, in 2018, GDPR (General data protection regulation, 2020) was proposed by the EU that removes the default option for collecting, storing, and harnessing individual data and requires explicit authorization from the users to use their data. Although the fundamental role played by these laws is to protect users' privacy, the consequent data scarcity dilemma can thereby jeopardize the training of high-quality models.

In this context, Federated Learning (FL) has been proposed by Google in recent years as a means to offer a *privacy-by-design* solution for machine-learned models (Konečný et al. 2015, 2016, McMahan et al. 2017). Federated learning aims to meet ML-privacy shortcomings by horizontally distributing the model's training over user devices; thus, clients locally train the global model exploiting private data without sharing it (McMahan et al., 2017). Federated Learning differs from distributed computing, since in the latter we witness a well-balanced computational effort among devices. Instead, with Federated Learning, the overall data is supposed to be massive in amount and unbalanced between personal devices. Recently, the benefits of federated learning in recommender systems have led to advantages for the privacy of the users of those systems (Anelli et al., 2021a). FPL (short for Federated Pair-wise Learning) (Anelli et al., 2021b) is an example of how the outstanding performance of Learning-to-Rank models for recommendation can be exploited in a federated scenario, giving users greater control of their data. Indeed, a disruptive effect of employing FPL is that users participating in the federation process can decide how much they are willing to disclose their private sensitive preferences.

This work presents an extensive analysis of FPL considering a large number of dimensions. In Anelli et al. (2021b), the authors analyzed how FPL, allowing users not to share a portion of private data for privacy concerns, impacts the performance of the system itself. This work extends the previous by a large margin, analyzing to what extent this portion impacts accuracy, diversity, and bias disparity of recommendation and communication costs in the whole system. Finally, the work investigates how local computation, i.e., user-wise training of a recommendation model, affects the overall performance.

To summarize, our research tackles the following questions:

- RQ1.** Is it possible to integrate Pair-wise Learning with Federated Learning principles to build a federated version of factorization models? What is the impact of federated parameters (i.e., *computation parallelism*, and *local computation amount*) on the quality of recommendation?

- RQ2.** The protection of the user's feedback can put the recommendation service in jeopardy. Can users receive a high-quality recommendation while limiting the amount of disclosed sensitive data?
- RQ3.** The sequentiality of the original pair-wise algorithms can be replicated at the price of increased communication costs. What is the optimal (or sub-optimal) trade-off between communication costs and recommendation utility?
- RQ4.** With limited training information, the recommendation algorithm might learn differently and unexpectedly. Does the federated recommendation (and the possible reduced information budget) inject additional biases in the final recommendation?

Our contributions are an in-depth investigation that aims to answer the above Research Questions thoroughly. To this extent, we have carried out extensive experiments on real-world datasets (Yang et al., 2016) in the Point of Interest (PoI) domain by considering the accuracy of recommendation and diversity metrics (Item Coverage and Gini Index). Afterward, we analyzed communication cost and accuracy in a multi-objective perspective and fairness (i.e., the Bias Disparity) of FPL recommendations. The experimental evaluation shows that FPL provides high-quality recommendations, putting the user in control of the amount of sensitive data disclosed.

2 Related work

2.1 Collaborative filtering methods for recommendation

Academia and industry have proposed several competitive recommendation algorithms. Algorithms based on Nearest Neighbors, Latent Factor Models and Artificial Neural Networks are undoubtedly the most representative examples of the Collaborative Filtering systems, that extract user preference patterns in a collaborative fashion.

The Nearest Neighbors scheme has shown its competitiveness for quite a long time. The user-based scheme and the item-based scheme find the k -nearest user neighbors and the k -nearest item neighbors based on a similarity function. It then exploits them to predict a score for each user-item pair.

Although they use the same logic behind the scenes, user-based and item-based schemes show their effectiveness in different contexts.

After these models, the most innovative idea to implement Collaborative Filtering has been decomposing the user-item rating matrix and exploiting the dot product to reconstruct the matrix and compute similarities. This idea led to the Matrix Factorization (MF) technique, which is probably the most representative of the factorization-based recommendation family. Nevertheless, several generalized/specialized variants have been proposed, such as FM (Rendle, 2010), SVD++ (Koren, 2008), PITF (Rendle & Schmidt-Thieme, 2010), FPMC (Rendle et al., 2010).

Unfortunately, rating-prediction-oriented optimization, like SVD, has shown its limits in the recommendation research (McNee et al., 2006). Consequently, a new class of *Learning to Rank* algorithms has been developed in the last decade, mainly ranging from point-wise (Koren & Sill, 2011) to pair-wise (Rendle et al., 2009), through List-wise (Shi et al., 2010) approaches. Among pair-wise methods, BPR (Rendle et al., 2009) is one of the most broadly adopted, thanks to its outstanding capabilities to correctly rank preserving an acceptable computational complexity. It exploits a stochastic gradient descent algorithm to learn the relative order between positive and negative items.

In the last years, methods that exploit side information have emerged (Anelli et al., 2021; Anelli et al., 2019a; Anelli et al., 2017). Finally, various architectures of deep neural networks have established themselves either in search and recommendation research. For each kind of recommendation task, one or more neural architectures have emerged that show competitive performance. Some examples are Recurrent Neural Networks for the sequential recommendation, variational autoencoders for general purpose collaborative filtering, and deep reinforcement learning methods for the interactive recommendation.

2.2 Motivations for federated learning

RSs need to collect user information related to attributes, demands, and preferences to work properly (Jeckmans et al., 2013). As a rule of thumb, the accuracy of recommendations is directly proportional to the level of detail of the gathered information (Huang et al., 2004). Regrettably, the more detailed the knowledge about users is the more significant the threat to the user's privacy becomes (Bilge et al., 2013).

In contexts like this, Federated Learning was introduced to learn models from a population while learning as little as possible about individuals. It meets the privacy shortcomings by horizontally distributing the model's training over user devices (McMahan et al., 2017). Beyond privacy, Federated Learning has posed several other challenges and opened new research directions (Kairouz et al. 2019; Anelli et al. 2019, 2020). Recently, Federated learning has extended to a more comprehensive idea of privacy-preserving decentralized collaborative ML approaches (Yang et al., 2019). These methods include horizontal federations, where different local datasets share the same feature space, and vertical federations, where devices share the training samples; however, they differ in feature space. Yang et al. (2019) identified some recent Federated Learning challenges and open research directions.

2.3 Federated recommender systems

Some researchers focused the attention on the decentralized and distributed matrix-factorization approaches (Duriakova et al., 2019; Fierimonte et al., 2017). However, in this work, we focus on federated learning principles theoretically and practically different from classical decentralized and distributed approaches, since Federated Learning assumes the presence of a coordinating server and the use of private and self-produced data on each node.

A federated implementation of collaborative filtering has been proposed in Ammad-uddin et al. (2019), which uses the SVD-MF method for implicit feedback (Hu et al., 2008). Here, the training is a mixture of Alternating Least Squares (ALS) and Stochastic Gradient Descent (SGD) for preserving users' privacy. However, its security limits have been analyzed in Chai et al. (5555aug). Recently, the federated learning paradigm spread to the recommendation tasks, thanks to its capability of dealing with sensitive data. As an example, DeepRec (Han et al., 2021) proposes a sequential recommender system where on-device training is performed to fine-tune with user sensitive data a deep learning model trained with data collected before the GDPR regulations. The recent model PREFER (Guo et al., 2021) is a sequence-based matrix factorization recommender system designed for the POI domain: the training data is enriched with time and distance information and, similarly to our model, does not share sensitive parameters about user profiles. Although sequential models employ the same metrics as the other recommender systems, their evaluation protocol is different. Indeed, they usually exploit a temporal leave-one-out splitting protocol and evaluate one recommendation per user, while other recommender systems mainly rely on temporal hold-out, k-folds cross-validation, or fixed timestamp splitting and evaluate

top- n recommendations per user. Moreover, each sequential model comes with a definition of *session* (necessary to compute the sequences), while this notion is absent in the other recommendation systems. For these reasons, the comparison of our methods with those recommender systems remains beyond the scope of the current work. Nevertheless, incomprehensibly, almost no work addressed top- N recommendation exploiting the “Learning to rank” paradigm. In this sense, one rare example is the work by Kharitonov (2019), who recently proposed to combine evolution strategy optimization with a privatization procedure based on differential privacy. The FPL framework, introduced by Anelli et al. (2021b), tackles some of the Federated learning challenges for a recommendation scenario and introduces a federated pair-wise recommender system where users are in control of their sensitive data. While the impact of incomplete data on its performance has been studied, other important research dimensions still need to be studied and are investigated in the work at hand.

3 Background

In this section, we introduce the fundamentals of the federated learning paradigm, the pair-wise learning to rank approach, and the factorization models. In detail, we designed the section to provide (i) a brief motivation of the technologies, (ii) the essential mathematical background, (iii) the formal definition of the main concepts, and (iv) the notation that is adopted in the following.

3.1 Federated learning

Federated learning (FL) is a paradigm initially envisioned by Google (Konečný et al., 2016; McMahan et al., 2017) to train a machine-learning model from data distributed among a loose federation of users’ devices (e.g., personal mobile phones). The rationale is to face the increasing issues of ownership and locality of data to mitigate the privacy risks resulting from centralized machine learning (Kairouz et al., 2019) while improving personalization (Jalalirad et al., 2019). In particular, given Θ denoting the parameters of a machine learning model, we consider a learning scenario where the objective is to minimize a generic loss function $G(\Theta)$. FL is a learning paradigm in which the users $u \in \mathcal{U}$ of a federation, who are owners of the data useful to train the model, collaborate to solve the learning problem under the coordination of a central server S without sharing or exchanging their raw data with S . From an algorithmic point of view, we start with S sharing Θ with the federation of devices. Then, specific methods solve a local optimization problem on the single device. The client shares the parameters of its local model with S . The parameters provided by the clients are then used to update Θ , which is sent back to the devices in a new iteration step.

Federated Learning poses some critical challenges. The first one is the feasibility of the adoption of decentralized machine learning schemes in real-world scenarios, due to client availability and communication potential issues. Another important challenge regards the learning convergence. Indeed, the federated approach realizes a certain number of parallel local steps before aggregating the parameters in the central model. This approach shows some similarities with batches (without the distributional guarantees) and therefore several dissimilarities with classic SGD (Stochastic Gradient Descent) steps. Moreover, hurdles related to data distribution among devices remain at the core of FL research. Finally, preventing the server or any other user from reconstructing a user’s dataset cannot be guaranteed by the only FL. In fact, it is achieved by juxtaposing schemes like encryption or

differential privacy. All these challenges and many others are active topics in current FL research. However, they remain out of our work scope since we focus on a more fundamental problem: the user's choice not to share a piece of their preference.

3.2 Factorization models and pair-wise recommendation

A recommendation problem is usually conceived as the activity of finding the items of a catalog a particular user might be interested in. Formally, let $\mathbf{X} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ be the user-item matrix where each entry x_{ui} represents an explicit or binary implicit feedback (e.g., explicit rating or check-in, respectively) of user $u \in \mathcal{U}$ for item $i \in \mathcal{I}$.

Definition 1 (Recommendation Problem) A recommendation problem over a set of users \mathcal{U} and a set of items \mathcal{I} is defined as the activity of finding for each user $u \in \mathcal{U}$ an item $i \in \mathcal{I}$ not rated by u that maximizes a utility function $g : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}$.

In the work at hand, an implicit feedback scenario is considered — i.e., feedback is, e.g., purchases, visits, clicks, views, check-ins —, with \mathbf{X} containing binary values. Therefore, $x_{ui} = 1$ and $x_{ui} = 0$ denote either user u has consumed or not item i , respectively.

In FPL, the underlying data model is a Factorization model, inspired by MF (Koren et al., 2009), a recommendation model that became popular in the last decade thanks to its state-of-the-art recommendation accuracy (kumar Bokde et al., 2015).

Definition 2 (Matrix Factorization) Given a set of users \mathcal{U} , a set of items \mathcal{I} , and a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$, Matrix Factorization builds a model Θ in which each user u and each item i is represented by the embedding vectors \mathbf{p}_u and \mathbf{q}_i , respectively, in the shared latent space \mathbb{R}^F . The core of the algorithm relies on the assumption that \mathbf{X} can be factorized such that the dot product between \mathbf{p}_u and \mathbf{q}_i can explain any observed user-item interaction x_{ui} , and that any non-observed interaction can be estimated as:

$$\hat{x}_{ui}(\Theta) = b_i(\Theta) + \mathbf{p}_u^T(\Theta)\mathbf{q}_i(\Theta), \quad (1)$$

where b_i is a term denoting the bias of the item i .

Among pair-wise approaches for learning-to-rank the items of a catalog, Bayesian Personalized Ranking (BPR) (Rendle et al., 2009) is one of the most broadly adopted, thanks to its capabilities to correctly rank with *acceptable* computational complexity.

Definition 3 (Bayesian Personalized Ranking) Let $\mathcal{K} : \mathcal{U} \times \mathcal{I} \times \mathcal{I}$ be a training set defined by $\mathcal{K} = \{(u, i, j) \mid x_{ui} = 1 \wedge x_{uj} = 0\}$. Bayesian Personalized Ranking is an optimization approach aiming to learn a model Θ that solves the personalized ranking task according to the following optimization criterion:

$$\max_{\Theta} \sum_{(u,i,j) \in \mathcal{K}} \ln \sigma(\hat{x}_{uij}(\Theta)) - \lambda \|\Theta\|^2, \quad (2)$$

where $\hat{x}_{uij}(\Theta) = \hat{x}_{ui}(\Theta) - \hat{x}_{uj}(\Theta)$ is a real value modeling the relation between user u , item i and item j , $\sigma(\cdot)$ is the sigmoid function, and λ is a model-specific regularization parameter to prevent overfitting.

Pair-wise optimization can be applied to a wide range of recommendation models, included factorization. Hereafter, we denote the model $\Theta = (\mathbf{P}, \mathbf{Q}, \mathbf{b})$, where $\mathbf{P} \in \mathbb{R}^{|\mathcal{U}| \times F}$ is a matrix whose u -th row corresponds to the vector \mathbf{p}_u , and $\mathbf{Q} \in \mathbb{R}^{|\mathcal{I}| \times F}$ is a matrix in

which the i -th row corresponds to the vector \mathbf{q}_i . Finally, $\mathbf{b} \in \mathbb{R}^{|\mathcal{I}|}$ is a vector whose i -th element corresponds to the value b_i .

4 Federated Pair-wise Learning

In this section, we introduce the fundamental concepts regarding the Collaborative Filtering recommendation using a Federated Learning scheme. Along with the problem definition, the notation we adopt is presented. Hereby, we want to make the reader aware that FPL is a tool for putting users in control of their data. In detail, here we focus on analyzing how different levels of data disclosure affect the recommendation. Providing privacy guarantees, e.g., by incorporating FPL in dedicated frameworks (Chai et al., 2019; Bonawitz et al., 2017; Abadi et al., 2016), remains out of the scope of this work.

4.1 Architecture

Following the FL principles, let \mathcal{U} be the set of users (clients) with a server S coordinating them. Assume users consume items from a catalog \mathcal{I} and give feedback about them (as in the recommendation problem of Section 3.2). S is aware of the catalog \mathcal{I} , while exclusively user u knows her own set of consumed items.

To setup the federation for FPL, a shared global model is built on the server S , while different private local models are built on each user's device.

Definition 4 (FPL Global Model) In FPL, the server S builds a global model $\Theta_S = \langle \mathbf{Q}, \mathbf{b} \rangle$, where $\mathbf{Q} \in \mathbb{R}^{|\mathcal{I}| \times F}$ and $\mathbf{b} \in \mathbb{R}^{|\mathcal{I}|}$ are the item-factor matrix and the bias vector introduced in Section 3.2.

Definition 5 (FPL Local Model) On each user u 's device FPL builds a model $\Theta_u = \langle \mathbf{p}_u \rangle$, which corresponds to the representation of user u in the latent space of dimensionality F .

Hence, in FPL, Θ_u and Θ_S are privately combined together. The client produces tailored recommendations by scalar multiplying local \mathbf{p}_u and \mathbf{q}_i . Each user u holds her own private dataset $\mathbf{x}_u \in \mathbb{R}^{\mathcal{I}}$, which, analogously to a centralized recommender system, corresponds to the u -th row of matrix \mathbf{X} . Each FPL client u hosts a user-specific training set $\mathcal{K}_u : \mathcal{U} \times \mathcal{I} \times \mathcal{I}$ defined by $\mathcal{K}_u = \{(u, i, j) \mid x_{ui} = 1 \wedge x_{uj} = 0\}$, where x_{ui} represents the i -th element of x_u . Please note that we refer to $X^+ = \sum_{u \in \mathcal{U}} |\{x_{ui} \mid x_{ui} = 1\}|$ as the total number of positive interactions in the system.

4.2 Training procedure

The classic BPR-MF learning procedure (Rendle et al., 2009) for model training can not be directly applied to the FPL model, since we have decoupled the representation of users and items respectively on the local devices and the server. In the following, we show the FPL learning procedure that is executed for a number R of rounds of communication and envisages **Distribution to Devices** \rightarrow **Federated Optimization** \rightarrow **Transmission to Server** \rightarrow **Global Aggregation** sequences between the server and the clients (Fig. 1).

1. **Distribution to Devices.** S randomly selects a subset of users $\mathcal{U}^{-t} \subseteq \mathcal{U}$ and delivers to them the current model Θ_S^{t-1} . The set \mathcal{U}^{-t} can be either defined by S , or the result of a request for availability sent by S to clients in \mathcal{U} .

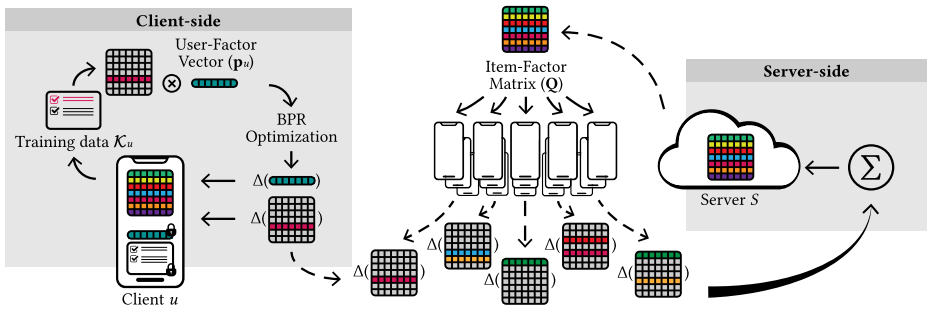


Fig. 1 Training protocol of FPL. In the middle, Item-Factor Matrix is sent by the server to the federation of devices. On the left, local training phase is represented. The local output, together with the output of the other devices, is sent to the server. On the right, server-side, aggregation of received updates is performed

2. **Federated Optimization.** Each user $u \in \mathcal{U}^t$ generates T random triples (u, i, j) from her dataset \mathcal{K}_u and for each of them performs BPR stochastic optimization to compute the updates for the local \mathbf{p}_u vector of Θ_u^{t-1} , and for $\mathbf{q}_i, b_i, \mathbf{q}_j,$ and b_j of the received Θ_S^{t-1} , following:

$$\Delta\theta^t = \frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \theta} \hat{x}_{uij} - \lambda\theta^{t-1}, \tag{3}$$

$$\text{with } \hat{x}_{uij} = [b_i^{t-1} + (\mathbf{p}_u^{t-1})^T \cdot \mathbf{q}_i^{t-1}] - [b_j^{t-1} + (\mathbf{p}_u^{t-1})^T \cdot \mathbf{q}_j^{t-1}], \tag{4}$$

$$\text{and } \frac{\partial}{\partial \theta} \hat{x}_{uij} = \begin{cases} (\mathbf{q}_i^{t-1} - \mathbf{q}_j^{t-1}) & \text{if } \theta = \mathbf{p}_u, \\ \mathbf{p}_u^{t-1} & \text{if } \theta = \mathbf{q}_i, \\ -\mathbf{p}_u^{t-1} & \text{if } \theta = \mathbf{q}_j, \\ 1 & \text{if } \theta = b_i, \\ -1 & \text{if } \theta = b_j. \end{cases} \tag{5}$$

It is worth noticing that Rendle et al. (2009) suggests, in a centralized scenario, to adopt a uniform distribution (over \mathcal{K}) to choose the training triples randomly. The purpose is to avoid data traversed item-wise or user-wise, since this may lead to slow convergence. Conversely, in a federated approach, we are required to train the model user-wise since the training of each round of communication is performed separately on each client u knowing only data in \mathcal{K}_u . This is the reason why, in FPL, the designer can control the number of triples T used for training, to tune the degree of local computation — i.e., how much the sampling is user-wise traversing. Moreover, it is worth noticing that, in FPL, the users in the round t compute the gradients based on the same model parameters Θ_S^{t-1} . As a consequence, their updates are independent and computed in parallel, unlike in stochastic centralized learning.

At the end of the federated computation, given a shared learning rate α , each client can update its local model Θ_u — containing the user profile \mathbf{p}_u — by aggregating the computed update:

$$\Theta_u^t := \Theta_u^{t-1} + \alpha \Delta\Theta_u^t. \tag{6}$$

3. **Transmission to Server.** In a purely distributed architecture, each user in \mathcal{U}^{-t} returns to S the computed update. Instead, in FPL, the clients in \mathcal{U}^{-t} send back to S a portion of the updates $(\Delta\Theta_{S,u}^t)$ for the computed item factor vector and item bias. In detail, sharing all the updates may lead to a significant users' private data disclosure that may lead to a privacy issue if the server S is curious. On the one hand, each pair of updates for a consumed item i and for a non-consumed item j contains equal but opposite gradients. Thus, if the user u sends all of them to S , they may reveal patterns of like/dislike user tastes. On the other hand, items rated by a user are more likely to be sampled and their corresponding vectors to be updated, thus allowing the server S to reconstruct, after some epochs, part of the user dataset \mathcal{K}_u . Since our primary goal is to put users in control of their data, FPL proposes a solution to overcome these vulnerabilities. By sending the sole update $(\Delta\mathbf{q}_j, \Delta b_j)$ of each training triple (u, i, j) , user u would share with S indistinguishably negative or missing values, which are assumed to be *non-sensitive* data. Furthermore, in FPL we introduce the parameter π , which allows users to control of the number of consumed items to share with the central server S . In detail, π works as a probability that the update $\Delta\Theta_{S,u}^t$ contains a specific positive item update $(\Delta\mathbf{q}_i, \Delta b_i)$ in addition to $(\Delta\mathbf{q}_j, \Delta b_j)$.
4. **Global Aggregation.** S aggregates the received updates in \mathbf{Q} and \mathbf{b} to build the new model:

$$\Theta_S^t := \Theta_S^{t-1} + \alpha \sum_{u \in \mathcal{U}^{-t}} \Delta\Theta_{S,u}^t, \quad (7)$$

with α being the learning rate (each row of \mathbf{Q} and each element of \mathbf{b} are updated by summing up the contribution of all clients in \mathcal{U}^{-t} for the corresponding item).

FPL reshapes the training scheme of centralized BPR-MF. However, it does not affect the computation for the model optimization, thus FPL has the same computational complexity of BPR-MF. Nonetheless, it is important to consider that some hyperparameters, analyzed in Section 4.3, can affect the convergence of FPL, increasing/decreasing the computation and communication costs.

4.3 Convergence analysis of FPL

Unlike other learning paradigms, in federated learning, the training data is not independent and identically distributed (non-iid). The user's local data is not representative of the overall data distribution. Therefore, one cannot replace them with samples drawn from the overall distribution. In 2020, Li et al. (2020) has shown that, given L -smooth and μ -strongly convex local losses like BPR, a federated optimization based on averaging of local parameters converges to the global optimum with a convergence rate of $\mathcal{O}(\frac{1}{RT})$. FPL may converge to a sub-optimal solution at least $\Omega(\alpha(T-1))$ away from the optimal one if weight decay is not considered. The number of rounds needed to reach a target performance is a function of the number of local epochs T , both linearly and inversely dependent on it (Li et al., 2020). Therefore, over-small and over-large values of T may lead to a large number of rounds of communication. In particular, if data is non-iid and T exceeds $\mathcal{O}(RT)$, convergence is not guaranteed, since the sum of local minima may not correspond to the global minimum. If sampling probabilities are highly non-uniform across the users, convergence may be slower (Zhao et al., 2018). However, some novel schemes have been recently proposed to address this issue (Li et al., 2020), and we will test them in future investigations.

Finally, under the non-iid setting, the convergence rate has a weak dependence on the size of \mathcal{U}^- . In practice, the participation ratio can be set small or large, according to the communication requirements and without affecting FPL convergence.

4.4 Privacy analysis of FPL

Section 4 starts by stating that FPL has not been conceived to be a privacy-preserving framework. Rather, it is a tool to control the trade-off between (potentially) exposed sensitive data and the recommendation quality. Federated learning hides, by design, users' raw data to the server: the updates sent by clients are anonymously aggregated, and only the aggregated information is deployed. Nevertheless, some *malicious* actors might still try to learn sensitive information if they have access to parts of the system, as already discussed in Section 4.2. For this reason, federated learning alone is not considered to provide privacy guarantees to users. FPL is a federated recommender system fed by implicit feedback. Consequently, providing privacy guarantees implies that the existence of each transaction in the user's history must be kept secret. With reference to (3) and (4), suppose a pair of positive and negative items i and j . The notation of $\Delta \mathbf{q}_i^t$ and $\Delta \mathbf{q}_j^t$ could be extended by focusing on a single latent factor f :

$$\Delta \mathbf{q}_{i,f}^t = \mathbf{p}_{u,f}^{t-1} \sigma(\mathbf{p}_{u,f}^{t-1}(\mathbf{q}_{i,f}^{t-1} - \mathbf{q}_{j,f}^{t-1})), \quad (8)$$

$$\Delta \mathbf{q}_{j,f}^t = -\mathbf{p}_{u,f}^{t-1} \sigma(\mathbf{p}_{u,f}^{t-1}(\mathbf{q}_{i,f}^{t-1} - \mathbf{q}_{j,f}^{t-1})), \quad (9)$$

where $\sigma(\cdot)$ returns values in the range $(0, 1)$. These equations show that the modules of $\Delta \mathbf{q}_{i,f}^t$ and $\Delta \mathbf{q}_{j,f}^t$ (that have to be sent to the server) are identical, while their signs are opposite. Moreover, the sign of the update depends on both the existence/absence of a transaction for k and on $\text{sgn}(\mathbf{p}_{u,f}^{t-1})$. Therefore, the sign of a gradient does not directly reveal the presence or absence of an item in the user's training set, but the pairs of positive and negative gradients disclose user preference patterns. In a round of communication, all the updates for the consumed items share the same sign, as well as all the updates for the non-consumed items have the same positive or negative sign, depending on $\text{sgn}(\mathbf{p}_{u,f}^{t-1})$. Suppose the server S is a honest-but-curious agent, i.e., it may try to inspect the updates to obtain some user information. Let us assume that, as soon as it obtains enough information adequate to identify one or more consumed/non-consumed items, the entire user dataset will be exposed. To avoid this problem, FPL puts users in control of their data. If the users adopt the *privacy-oriented* masking procedure discussed in Section 4.2, they can decide the fraction of updates for positive items to send. In the case of exposure of the user transactions, only a fraction is given up. This work studies and analyzes the recommendation performance in this data scarcity scenario. While we do not explicitly define a user-specific protocol for privacy level tuning, the system allows both possibilities: the system designer defines a fixed portion of data users should share, or users actively decide the fraction of data to share. For instance, the users might choose among a set of privacy/accuracy trade-off levels, as already happens with location data in some commercial products. If a user is not satisfied with the accuracy performance, she might modify the privacy/accuracy trade-off level at any moment.

Other possible privacy issues, like active reconstruction of the user profile, are not considered here and are out of the scope of this work. However, federated learning literature already provides privacy protocols like differential privacy and cryptographic methods. They have been proven to guarantee user privacy, so FPL architecture has been explicitly designed to work with them.

Table 1 Characteristics of the datasets used for experiments: $|\mathcal{U}|$, $|\mathcal{I}|$, and X^+ are the number of users, items, and records

Dataset	$ \mathcal{U} $	$ \mathcal{I} $	X^+	$\frac{X^+}{ \mathcal{U} }$	$\frac{X^+}{ \mathcal{I} }$	$\frac{X^+}{ \mathcal{I} \cdot \mathcal{U} }$ %
Brazil	17,473	47,270	599,958	34.34	12.69	0.00073%
Canada	1,340	29,518	63,514	47.40	2.15	0.00161%
Italy	1,353	25,522	54,088	39.98	2.20	0.00157%

5 Experimental setup

In this section, we introduce the experimental setting designed to answer the research questions.

5.1 Datasets

The evaluation of FPL needs to meet some particular constraints: the availability of transaction data to obtain a reliable experimental setting and a domain that guarantees the presence of data the user may prefer to protect. Following these constraints, we believe that the Point-of-Interest (PoI) domain would be optimal to test FPL, since it concerns data that users usually perceive as sensitive. Among the many available datasets, a very good candidate is the *Foursquare* dataset (Yang et al., 2016). In fact, it is often considered as a reference for evaluating PoI recommendation models. To mimic a federation of devices in a single country, we have extracted check-ins for three countries, namely Brazil, Canada, and Italy. While selecting the different countries, our only constraint was to obtain datasets with different size/sparsity characteristics. Hence, we choose three countries in three different regions of the world. To fairly evaluate FPL against the baselines, we have kept users with more than 20 interactions¹. Moreover, we have split the datasets by adopting a realistic temporal hold-out 80-20 splitting on a per-user basis (Gunawardana & Shani, 2015; Anelli et al., 2019b). The resulting training and test sets have been used with all the methods in comparison, including the state-of-the-art algorithms. Table 1 shows the characteristics of the resulting training sets adopted in the experiments.

5.2 Collaborative filtering baselines

To evaluate the efficacy of FPL, we have conducted the experiments by considering non-personalized methods (random and most popular recommendation), and different recommendation approaches, including the centralized **BPR-MF** implementation (Rendle et al., 2009), **User-kNN** and **Item-kNN** (Koren, 2010), **VAE** (Liang et al., 2018), and **FCF** (Ammad-ud-din et al., 2019), which is, to date, the only federated recommendation approach based on MF². Following Dacrema et al. (2019), we considered only **VAE** as representative of the neural approaches.

To evaluate the impact of exploiting only a partial user feedback on recommendation accuracy, we have evaluated different values of π in $[0.0, 1.0]$ with step 0.1, with $\pi = 0.0$ meaning that u is not sharing any positive feedback with the server, and $\pi = 1.0$ meaning

¹The limitations of the Collaborative Filtering in a cold-start user setting are well-known in literature. However, they are beyond the scope of this work.

²Since no source code is available, we implemented it from scratch and considered it in the reader's interest.

that u is sharing the updates on all positive items. Hence, we have considered four different configurations regarding computation and communication:

- **sFPL**: it reproduces the centralized stochastic learning, where the central model is updated sequentially; thus, we set $|\mathcal{U}^-| = 1$ to involve just one random client per round, and it extracts solely one triple (u, i, j) from its dataset ($T = 1$) for the training phase;
- **sFPL+**: we increase client local computation by raising to $\frac{X^+}{|\mathcal{U}^-|}$ the number of triples T extracted from \mathcal{K}_u by each client involved in the round of communication;
- **pFPL**: we enable parallelism by involving all clients in each round of communication ($\mathcal{U}^- = \mathcal{U}$) and we keep $T = 1$;
- **pFPL+**: we extend pFPL by letting each client sample $T = \frac{X^+}{|\mathcal{U}^-|}$ triples from \mathcal{K}_u ; the rationale is that the overall training samples are exactly X^+ , as in centralized BPR-MF.

Rendle et al. (2009) suggest to set the number of triples used for training in one epoch of BPR to X^+ . This corresponds to the number of total number of positive interactions in the system. Therefore, the federated training is comparable to BPR when X^+ optimization steps are performed. To this extent, we introduce the number of rounds of communication per epoch (*rpe*). Consequently, FPL computation after *rpe* rounds is comparable to one epoch of centralized BPR when $|\mathcal{U}^-| \cdot T \cdot rpe = X^+$. This results in $rpe = X^+$ for sFPL, $rpe = |\mathcal{U}^-|$ for sFPL+, $rpe = \frac{X^+}{|\mathcal{U}^-|}$ for pFPL, and $rpe = 1$ for pFPL+.

5.3 Reproducibility

For what regards the splitting strategy, we have adopted a **temporal hold-out 80/20** to separate our datasets in training and test set. Moreover, to find the most promising learning rate α , we have further split the training set, adopting a temporal hold-out 80-20 strategy on a user basis to extract her validation set. **User-kNN** and **Item-kNN** have been experimented for $k \in \{10, 20, \dots, 10\}$ considering Cosine Vector Similarity. **VAE** has been trained by considering three autoencoder topologies, with the following number of neurons per layer: 200-100-200, 300-100-300, 600-200-600. We have chosen candidate models by considering the best models after training for 50, 100, and 200 epochs, respectively. For the **factorization models**, we have performed a grid search in BPR-MF for $\alpha \in \{0.005, 0.05, 0.5\}$ varying the number of latent factors in $\{10, 20, 50\}$. Then, to ensure a fair comparison, we have exploited the same learning rate and number of latent factors to train **FPL** and **FCF**, and we explored the models in the range of $\{10, \dots, 50\}$ iterations. We have set *user-* and *positive item-*regularization parameter to $\frac{1}{20}$ of the learning rate. The *negative item-*regularization parameter is $\frac{1}{200}$ of the learning rate, as suggested in *mymedialite*³ implementation as well as by Anelli et al. (2019). We made the implementation of FPL publicly available⁴. Moreover, it will be soon integrated into the reproducibility framework Elliot (Anelli et al., 2021).

5.4 Evaluation metrics

The RQs (see Section 1) cover a broad spectrum of different recommendation dimensions. To this end, we have decided to measure several metrics to evaluate the approaches under the different perspectives.

³<http://www.mymedialite.net/>

⁴<https://split.to/sisinflab-fpl>

Accuracy The accuracy of the models is measured by exploiting Precision ($P@N$) and Recall ($R@N$). They respectively represent, for each user, the proportion of relevant recommended items in the recommendation list, and the fraction of relevant items that have been altogether suggested. We have assessed the statistical significance of results by adopting Student's paired T-test considering p-values $< 0.05^5$.

Beyond-Accuracy To measure the diversity of recommendations, we have measured the Item Coverage ($IC@N$), and the Gini Index ($G@N$). IC provides the number of diverse items recommended to users. It also conveys the sense of the degree of personalization (Adomavicius & Kwon, 2012). Gini (G) is a metric about distributional inequality. It measures how unequally different items a RS provides users with Castells et al. (2015). In the formulation adopted (Gunawardana & Shani, 2015), a higher value of G corresponds to higher personalization.

Fairness The problem of unfair outputs in machine learning applications is well studied (Bozdag, 2013; Dwork et al., 2011) and also it has been extended to recommender systems (Mansoury et al., 2019). In detail, in this work, we check whether items belonging to specific groups have equal chance to be shown in the recommended lists. In order to do that, we measure Bias Disparity ($BD@N$) (Mansoury et al., 2019) for groups of items. With this metric we quantify, for each category of items, the deviation of the proposed recommendations from the initial dataset bias.

6 Results and discussion

In this Section, we focus on the different experiments conducted to explore the dimensions covered by the Research Questions (see Section 1). First, to position FPL with respect to the baselines, we analyze the accuracy, beyond-accuracy, and bias disparity of the recommendations. Once the analysis is completed, we investigate the impact of communication costs, and we study the multi-objective optimization of maximizing the accuracy while minimizing the communication costs. To this extent, we have explored the Pareto frontier, considering the two different dimensions.

6.1 Recommendation accuracy

To answer RQ1, we want to assess whether it is possible to obtain a recommendation performance comparable to a centralized pair-wise learning approach while allowing the users to control their data. In this respect, Table 2 shows the accuracy and diversity results of the comparison between the state-of-the-art baselines and the four configurations of FPL presented in Section 5. By focusing on accuracy metrics, we may notice that VAE outperforms the other approaches in the three datasets. However, who is familiar with VAE knows that, since it restricts training data by applying k-core, it does not always produce recommendations for all the users. With regards to User-kNN, we notice that it outperforms all the other approaches in the three datasets, while the performance of Item-kNN and BPR-MF approximately settle in the same range of values. This is possibly due to the user-item ratio (Adomavicius & Zhang, 2012), that favors the user-based schemes (see Table 1).

Moreover, it is important to investigate the differences of FPL with respect to BPR-MF, which is a pair-wise centralized approach, being FPL the first federated pair-wise recommender based on a factorization model. The performance of BPR-MF against FPL, in

⁵The complete results are available in the implementation repository.

Table 2 Results of accuracy metrics for baselines and FPL on the three datasets. For each configuration of FPL and for each dataset, the experiment with the best π is shown (see the bottom part for details). For all metrics, the greater the better. Among federated algorithms, the best performance is in **boldface**

		Brazil		Canada		Italy	
		P@10	R@10	P@10	R@10	P@10	R@10
Centralized	Random	0.00013	0.00015	0.00030	0.00035	0.00030	0.00029
	Top-Pop	0.01909	0.02375	0.04239	0.04679	0.04634	0.05506
	User-kNN	0.10600	0.13480	0.07639	0.07533	0.06881	0.07833
	Item-kNN	0.07716	0.09607	0.04006	0.03881	0.04663	0.05356
	VAE *	0.10320	0.13153	0.06060	0.06317	0.10421	0.21324
	BPR-MF	0.07702	0.09494	0.03694	0.03650	0.04560	0.05458
Federated	FCF	0.03089	0.03749	0.03724	0.03836	0.03126	0.03708
	sFPL **	0.07757	0.09581	0.04515	0.04550	0.04701	0.05600
	sFPL+ **	0.08682	0.11004	0.05701	0.05665	0.05595	0.06229
	pFPL **	0.07771	0.09582	0.04582	0.04637	0.04642	0.05465
	pFPL+ **	0.08733	0.11085	0.05761	0.05755	0.05565	0.06291

*For Italy, VAE does not produce recommendations for all the users; thus, we followed the weighting scheme proposed in prior literature (Mesas & Bellogín, 2017)

**Best π obtained for each of the proposed FPL variations across three countries (Brazil, Canada, and Italy) are: sFPL = (0.5, 0.1, 0.4), sFPL+ = (0.9, 0.4, 0.2), pFPL = (0.8, 0.1, 1), pFPL+ = (0.8, 0.3, 0.1)

the configuration sFPL, shows how precision and recall in sFPL are slightly outperforming BPR-MF, while achieving very similar diversity values. The consideration that the performance is comparable is surprising since the two methods share the sequential training, but sFPL exploits a π reduced to 0.5, 0.1, and 0.4, respectively, for Brazil, Canada, and Italy. This behavior is more evident in Fig. 2, where the harmonic mean between Precision and Recall (F1) is plotted for different values of π . If we look at the dark blue line with squares, we may observe how the best result does not correspond to $\pi = 1$. Compared to FCF, FPL generally behaves better and preserves privacy to a greater extent, since sharing gradients of all rated items in FCF can result in a data leak (Chai et al., 2019).

In the last three rows of Table 2, we explore an increasing of the local computation (sFPL+), or an increased parallelism (pFPL), or a combination of both (pFPL+). In detail, we observe that sFPL+ takes advantage of the increased local computation, and FPL significantly outperforms BPR-MF for the three datasets; for instance, for Canada, we

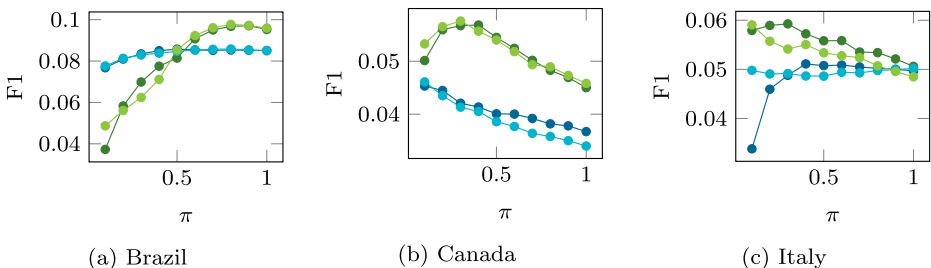


Fig. 2 F1 performance at different values of π in the range [0.1, 1]. The colors represent the four configurations: blue squares refer to sFPL, green squares to sFPL+, blue circles to pFPL, and green circles to pFPL+

observe an interesting increase in precision. Instead, when comparing p FPL with s FPL, we observe that the increased parallelism does not affect the performance significantly. Even then, the increased local computation boosts the Precision and Recall performance, up to 24% for precision in the Italy dataset. The results confirm RQ1, since “*the proposed system can generate recommendations with a quality that is comparable with the centralized pair-wise learning approach. Moreover, the increased local computation causes a considerable improvement in the accuracy of recommendations. On the other side, the training parallelism does not significantly affects results. Finally, when the local computation is combined with parallelism, the results show a further improvement*”.

To answer RQ2, we varied π in the range $[0.1, \dots, 1.0]$ to assess how removal of the updates for consumed items affects the final recommendation accuracy, and we plotted the accuracy performance by considering F1 in Fig. 2. As previously observed, the best performance rarely corresponds to $\pi = 1$. On the contrary, a general trend can be observed: the training reaches a peak for a certain value of π — depending on the dataset —, and then the system performance decays in accuracy when increasing the value of π . In rare cases, e.g., s FPL, and p FPL for Brazil dataset, the decay is absent, but results that are very close for different values of π . The general behavior suggests that the system learning exploits the updates of positive items to absorb information about popularity. This consideration is coherent with the mathematical formulation of the learning procedure, and it is also supported by the observation that for Canada and Italy FPL reaches the peak before with respect to Brazil. Indeed, Canada and Italy datasets are less sparse than Brazil, and the increase of information about positive items may lead to push up too much the popular items (this is a characteristic of pair-wise learning), while the same behavior in Brazil can be observed for values of π very close to 1. The same mathematical background, for s FPL+ and p FPL+ with Brazil dataset, which is very sparse, explains the higher value of π needed to reach good performance. Here, the lack of positive information with a vast catalog of items, confuses the training that cannot exploit item popularity. Now, we can positively answer to RQ2: *user can receive high-quality recommendations also when she decides to disclose a small amount of her sensitive data. However, it should be noted that the more the dataset is sparse, the more the amount of sensitive data should be large.*

6.2 Accuracy or diversity: exploring the trade-off between precision and item coverage

In Table 3, we have depicted the diversity metrics results of each experiment, i.e., item coverage, and Gini Index. What immediately catches our attention is an increase in IC and Gini in accord with the increase of local computation. In this sense, FPL shows a consistent prominence on BPR-MF. This performance is motivated by mere observation of the algorithm. By increasing local computation, each client compares each positive item with a significantly larger number of negative samples (i.e., wider spread). We have also explored the values of IC against the values of precision for each dataset and for each configuration while varying the parameter π . In Fig. 3, we plot these values by considering increasing π in the direction of the arrows. The plots unveil that, for Canada and Italy, by increasing the local computation (s FPL+ and p FPL+), the plots develop rightwards, i.e., a significant IC increase. Although such an increase may lead to low precision (as in the random recommender), we observe that the same configurations also push up the value of precision, so that the green points are positioned at the top of the plots. We also note that the value of π affects more IC in configurations with high computation than those with low computation. However, while IC seems to increase when increasing π , precision follows the

Table 3 Results of beyond-accuracy metrics for baselines and FPL on the three datasets. For each configuration of FPL and for each dataset, the experiment with the best π is shown (see the bottom part for details). For all metrics, the greater the better. Among federated algorithms, the best performance is in **boldface**.

		Brazil		Canada		Italy	
		IC@10	G@10	IC@10	G@10	IC@10	G@10
Centralized	Random	46120	0.70946	10815	0.26809	10478	0.28914
	Top-Pop	19	0.00020	18	0.00030	19	0.00035
	User-kNN	3083	0.01159	609	0.00321	577	0.00282
	Item-kNN	16535	0.07449	4393	0.05404	3241	0.03293
	VAE *	5503	0.02117	1044	0.00652	165	0.02336
	BPR-MF	2552	0.00756	1216	0.00998	19	0.00036
Federated	FCF	911	0.00095	504	0.00174	403	0.00158
	sFPL **	1581	0.00561	451	0.00243	18	0.00036
	sFPL+ **	5200	0.01449	1510	0.01259	932	0.00789
	pFPL **	2114	0.00638	425	0.00213	96	0.00056
	pFPL+ **	3820	0.01106	1214	0.00981	936	0.00725

* For Italy, VAE does not produce recommendations for all the users; thus, we followed the weighting scheme proposed in prior literature (Mesas & Bellogín, 2017)

** Best π obtained for each the proposed FPL variations across three countries (Brazil, Canada, and Italy) are: sFPL = (0.5, 0.1, 0.4), sFPL+ = (0.9, 0.4, 0.2), pFPL = (0.8, 0.1, 1), pFPL+ = (0.8, 0.3, 0.1)

previously described behavior. At first glance, Brazil seems to behave differently from the other datasets. Even here, we may discern a better combination of IC and precision for configurations with high computation. FPL needs to reach a higher value of π to witness a high precision and high IC. This behavior was also evident in the accuracy analysis, considering the different values of π .

6.3 Accuracy vs communication cost: a multi-objective analysis

In a FL setting, communication rounds between clients and server play a crucial role. In fact, a large amount of information exchanged might hinder the effectiveness of the overall approach as it requires high network costs. This perspective has led us to define a metric, the *Communication Cost per Epoch (CCE)*, which calculates communication costs that each

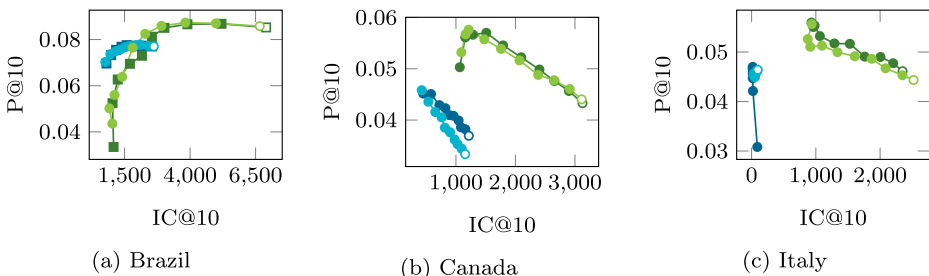


Fig. 3 Item Coverage (IC@10) versus Precision (P@10) with cutoff 10. The colors represent the four configurations: blue squares refer to sFPL, green squares to sFPL+, blue circles to pFPL, and green circles to pFPL+. The white points denote $\pi = 1.0$ to specify the direction of increasing π

particular FPL configuration requires as the number of bidirectionally exchanged vectors. Let T be the number of sent updates for non-consumed items and πT the number of sent updates for consumed items. For rpe rounds the server establishes a communication with $|\mathcal{U}^-|$ clients, sending to each of them $|\mathcal{I}|$ vectors and receiving from each of them $T(1 + \pi)$ update vectors. Therefore, CCE is estimated as $CCE = rpe \cdot |\mathcal{U}^-| \cdot (|\mathcal{I}| + T(1 + \pi))$. Given these definitions, in this section, we want to analyze the effects of the different configurations and π values on the communication cost.

For convenience, we focus the analysis on Brazil, the biggest and sparsest dataset. In Table 4, we show the values of precision and communication cost for each FPL configuration and each value of π . The Total Communication Cost (TCC) is computed as the product between CCE and the number of epochs needed to reach such precision value. At a first glance, it is noteworthy how, within a specific configuration, the value of π does not affect significantly the total communication cost, while it highly impacts on the best precision value. For each configuration, we plot in Fig 4 the best values of precision (in boldface in Table 4) against their TCC . Here, the total communication cost should be minimized, while the precision should be maximized. The optimal solution in terms of multi-objective optimization corresponds to the $pFPL+$ configuration. Instead, in absence of parallelism, we witness a much higher communication cost for reaching the best precision. Moreover, it is interesting how $sFPL$ and $pFPL$ are perfectly overlapping both in terms of accuracy (as also confirmed by the previous analyses) and in terms of communication costs. However, increasing the local computation in a parallel setting make FPL to reach the best performance with the minimum overall communication cost.

The multi-objective analysis between communication cost and accuracy may help the designer in providing the best setup for the federation of clients. Here, the analysis suggests holding high parallelism configurations with high local computation as the set of optimal settings. The experiment shows that in FPL there is no need for sacrificing accuracy for communication costs. Instead, the user can freely choose the value of π without affecting the communication costs. In order to answer the RQ3 we can state that *deciding to limit the communication costs does not particularly affect the recommendation accuracy*.

Table 4 Total Communication Cost ($\times 10^{-12}$) (TCC) versus Precision (P@10) on Brazil dataset

π	sFPL		sFPL+		pFPL		pFPL+	
	TCC	P@10	TCC	P@10	TCC	P@10	TCC	P@10
0.1	1.27623	0.06961	1.41913	0.03347	1.27623	0.07026	0.99339	0.04358
0.2	1.27623	0.07327	1.41924	0.05241	1.27623	0.07366	0.99347	0.05022
0.3	1.27624	0.07551	1.41934	0.06269	1.27624	0.07497	0.99354	0.05598
0.4	1.27624	0.07686	1.41944	0.06949	1.27624	0.07582	0.99361	0.06382
0.5	1.27624	0.07757	1.41955	0.07298	1.27624	0.07671	0.99368	0.07648
0.6	1.27624	0.07733	1.41965	0.08121	1.27624	0.07723	0.99375	0.08247
0.7	1.27625	0.07714	1.41975	0.08506	1.27625	0.07758	0.99383	0.08590
0.8	1.27625	0.07730	1.41985	0.08660	1.27625	0.07771	0.99390	0.08733
0.9	1.27625	0.07724	1.41996	0.08682	1.27625	0.07726	0.99397	0.08699
1.0	1.27625	0.07702	1.42006	0.08523	1.27625	0.07703	0.99404	0.08582

TCC is the product between the value of CCE and the actual number of epochs needed to obtain the best accuracy value. For each configuration, the best precision value is in **boldface** and reported in the summary graph in Fig. 4

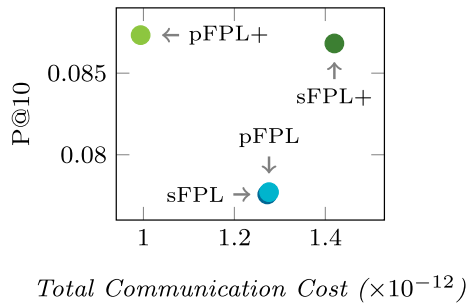


Fig. 4 Total Communication Cost ($\times 10^{-12}$) compared to Precision (P@10) on Brazil dataset. The colors represents the four configurations: dark blue is sFPL, dark green is sFPL+, light blue is pFPL, light green is pFPL+. For each configuration, the best accuracy performance is shown. The top-left corner of the plot is the best trade-off between accuracy and communication costs

Overall, FPL shows its best trade-off between communication costs and accuracy when both parallelism and high local computation are set.

6.4 Bias disparity in FPL

When depriving the recommender of a part of the user's feedback, one of the biggest concerns is the potential bias shift (Burke et al., 2017). Bias analysis, and fairness are gaining momentum in the last years (Deldjoo et al., 2021), they unveil several essential aspects of the recommenders' behavior. To explore what happens the category biases in the different configurations and values of π , we measure the bias disparity (BD) in recommendation lists for the categories of the venues. This metric analyzes how much the output of a recommendation algorithm deviates the natural propensity of the users for particular categories of items towards other categories. Notably, for a category of items C , BD is defined as it follows:

$$BD(C) = \frac{B_R(C) - B_T(C)}{B_T(C)}, \quad (10)$$

where $B_T(C)$ is the source bias on category C , i.e., how much users were biased towards category C in the training set, and $B_R(C)$ is the bias on C in recommendation lists.

Table 5 shows the source bias value B_T (Mansoury et al., 2019) for the different categories in training data, with a value above 1 denoting a higher susceptibility to choose the category items. Table 6 shows the results in terms of Bias Disparity (BD) for FPL and the other baselines. Here, the closer to 0, the closer to the initial bias. As expected, Top-Pop

Table 5 Bias values B_T (Mansoury et al., 2019) of population on the different categories in training data (A&E: Arts & Entertainment, C&U: College & University, NS: Nightlife Spot, O&R: Outdoors & Recreation, P&OP: Professional & Other Places, S&S: Shop & Service, T&T: Travel & Transport)

Dataset	A&E	C&U	Food	NS	O&R	P&OP	Res.	S&S	T&T
Brazil	1.4949	0.6289	1.1024	1.3286	1.1340	0.6424	0.5202	0.8314	1.3699
Canada	1.7224	0.8310	1.0879	1.6594	0.9719	0.6610	0.4134	0.8087	1.2328
Italy	1.4130	0.8221	0.9317	1.3559	1.3292	0.7868	0.4171	0.8482	1.2678

Table 6 Results of recommendation bias disparity for each category in Brazil dataset (see Table 1) for baselines and FPL

		A&E	C&U	Food	NS	O&R	P&OP	Res.	S&S	T&T
Centralized	Random	-0.325	0.559	-0.080	-0.245	-0.143	0.549	0.911	0.203	-0.273
	Top-Pop	-1.000	-1.000	-0.302	-1.000	-0.999	-1.000	-1.000	-1.000	6.660
	User-kNN	0.445	-0.832	0.261	-0.213	0.162	-0.842	-0.969	-0.431	0.621
	Item-kNN	0.346	0.140	0.068	-0.102	0.153	-0.347	-0.381	-0.227	0.083
	VAE	0.393	-0.723	0.223	-0.315	0.194	-0.776	-0.911	-0.310	0.572
	BPR-MF	0.301	-0.712	0.232	-0.710	0.142	-0.758	-0.992	-0.434	1.165
Federated	FCF	-0.464	-0.968	0.687	-0.910	0.135	-0.960	-0.994	-0.946	1.239
	sFPL	0.272	-0.738	0.263	-0.756	0.161	-0.814	-0.997	-0.368	1.072
	sFPL+	0.311	-0.675	0.160	-0.396	0.278	-0.806	-0.903	-0.291	0.812
	pFPL	0.253	-0.813	0.218	-0.613	0.143	-0.739	-0.992	-0.479	1.239
	pFPL+	0.154	-0.566	0.190	-0.351	0.345	-0.764	-0.913	-0.410	0.778

For each configuration of FPL and for each dataset, the experiment with the best π is shown. The closer to 0 the better. Among federated algorithms, the best performance is in **boldface**

changed the recommendation towards T&T, which is the most popular category in the training set. By focusing on FPL, we may notice that it bias positively and negatively the same categories of the other state-of-the-art algorithms. Notably, it particularly pushes the bias of recommendation towards popular categories (e.g., A&E, Food, T&T), while it emphasizes the unpopularity of specific categories — above all C&U , P&OP, Residence —. This is probably due to the pair-wise nature of the approach, which works by iteratively increasing the difference values between enjoyed items and the others (the same behavior is evident for BPR-MF). The Bias Disparity analysis helps to answer RQ4. Hence, we draw the following consideration: “*the proposed system generates recommendations that are biased to the initial user preferences since it emphasizes the differences between consumed and non-consumed items. This behavior is also coherent with the recommendations of the other state-of-the-art algorithms*”.

7 Conclusion and future work

This work proposes Federated Pair-wise Learning (FPL), a novel federated learning framework that exploits pair-wise learning for factorization models in a recommendation scenario. The model leaves the user-specific information of the original factorization model in the clients’ devices so that a user may be entirely in control of her sensitive data and could share no positive feedback with the server. The framework can be envisioned as a general factorization model in which clients can tune the amount of information shared among devices. To analyze the degree of accuracy, the diversity of the recommendation results, we have conducted an extensive experimental evaluation. However, even a vast evaluation is not enough to gain a more in-depth understanding of how FPL operates. Therefore, we have extended the evaluation to investigate the optimal trade-off between accuracy, and amount of shared transactions. Afterwards, the study provides a theoretical analysis of the privacy issues of FPL, the details of computational complexity, and an investigation on communication costs considering the different operational modes. Finally, the work analyzes the shift of the

original data bias when the system is fed with partial information. To the best of our knowledge, it is one of the first attempts to understand how a federated learning approach impacts the fairness of the overall system. The proposed model shows performance comparable with several state-of-the-art baselines and the classic centralized factorization model with pairwise learning. Interestingly, indeed, clients can share a small portion of their data with the server and still receive high-performance recommendations. We believe that the proposed approach represents the joining link between federated matrix factorization and the modern recommendation systems that optimize the item ranking instead of the prediction error. In the near future, it would be interesting to investigate the behavior of FPL in new privacy settings, examine the effects of each user freely choosing which specific data to keep private, and extend the experimental analysis to other datasets and domains. Finally, we think that federated learning to rank approach, along with a rigorous analysis of the dimensions involved in the recommendation process, may open the doors to a new class of ubiquitous recommendation engines.

Declarations

Competing interests The authors declare that they have no conflict of interest.

References

- Abadi, M., Chu, A., Goodfellow, I.J., McMahan, H.B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In E.R. Weippl, S. Katzenbeisser, C. Kruegel, A.C. Myers, & S. Halevi (Eds.) *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 308–318). Vienna: ACM. <https://doi.org/10.1145/2976749.2978318>.
- Adomavicius, G., & Kwon, Y. (2012). Improving aggregate recommendation diversity using ranking-based techniques. *IEEE TKDE*, 24(5), 896–911.
- Adomavicius, G., & Zhang, J. (2012). Impact of data characteristics on recommender systems performance. *ACM Trans Manag Inf Syst*, 3(1), 3:1–3:17. <https://doi.org/10.1145/2151163.2151166>.
- Ammad-ud-din, M., Ivannikova, E., Khan, S.A., Oyomno, W., Fu, Q., Tan, K.E., & Flanagan, A. (2019). Federated collaborative filtering for privacy-preserving personalized recommendation system. CoRR arXiv:1901.09888.
- Anelli, V.W., Bellogín, A., Ferrara, A., Malitesta, D., Merra, F.A., Pomo, C., Donini, F.M., & Noia, T.D. (2021). Elliot: A comprehensive and rigorous framework for reproducible recommender systems evaluation. In: SIGIR. ACM, pp 2405–2414.
- Anelli, V.W., Deldjoo, Y., Di Noia, T., & Ferrara, A. (2019). Towards effective device-aware federated learning. In: Int. Conf. of the Italian Association for Artificial Intelligence. Springer, pp 477–491.
- Anelli, V.W., Deldjoo, Y., Noia, T.D., & Ferrara, A. (2020). Prioritized multi-criteria federated learning. *Intell Artif*, 14(2), 183–200.
- Anelli, V.W., Deldjoo, Y., Noia, T.D., Ferrara, A., & Narducci, F. (2021). Federank: User controlled feedback with federated recommender systems. In: ECIR (1), Lecture Notes in Computer Science, vol 12656. Springer, pp 32–47.
- Anelli, V.W., Deldjoo, Y., Noia, T.D., Ferrara, A., & Narducci, F. (2021). How to put users in control of their data in federated top-n recommendation with learning to rank. In: SAC '21: The 36th ACM/SIGAPP symposium on applied computing, virtual event. ACM, Republic of Korea, pp 1359–1362. <https://doi.org/10.1145/3412841.3442010>.
- Anelli, V.W., Noia, T.D., Lops, P., & Sciascio, E.D. (2017). Feature factorization for top-n recommendation: From item rating to features relevance. In *RecSysKTL, CEUR Workshop Proceedings*, (Vol. 1887 pp. 16–21). CEUR-WS.org.
- Anelli, V.W., Noia, T.D., Sciascio, E.D., Ferrara, A., & Mancino, A. (2021). Sparse feature factorization for recommender systems with knowledge graphs. In *RecSys 2021: Fifteenth ACM Conference on Recommender Systems (RecSys '21)*. Amsterdam: ACM. <https://doi.org/10.1145/3460231.3474243>.

- Anelli, V.W., Noia, T.D., Sciascio, E.D., Pomo, C., & Ragone, A (2019). On the discriminative power of hyper-parameters in cross-validation and how to choose them. In *Proc. of the 13th ACM Conf. on Recommender Systems* (pp. 447–451).
- Anelli, V.W., Noia, T.D., Sciascio, E.D., Ragone, A., & Trotta, J (2019). How to make latent factors interpretable by feeding factorization machines with knowledge graphs. In *ISWC (1), Lecture Notes in Computer Science*, (Vol. 11778 pp. 38–56). Springer.
- Anelli, V.W., Noia, T.D., Sciascio, E.D., Ragone, A., & Trotta, J (2019). Local popularity and time in top-n recommendation. In *European Conf. on Information Retrieval*, (Vol. 11437 pp. 861–868). Springer.
- Bilge, A., Kaleli, C., Yakut, I., Gunes, I., & Polat, H (2013). A survey of privacy-preserving collaborative filtering schemes. *Int J Softw Eng Knowl Eng*, 23(8), 1085–1108.
- Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., & Seth, K (2017). Practical secure aggregation for privacy preserving machine learning. *IACR Cryptol ePrint Arch*, 2017, 281. <http://eprint.iacr.org/2017/281>.
- Bozdag, E. (2013). Bias in algorithmic filtering and personalization. *Ethics Inf Technol*, 15(3), 209–227. <https://doi.org/10.1007/s10676-013-9321-6>.
- Burke, R., Sonboli, N., Mansoury, M., & Ordo nez-Gauger, A (2017). Balanced neighborhoods for fairness-aware collaborative recommendation.
- Castells, P., Hurley, N.J., & Vargas, S (2015). *Novelty and diversity in recommender systems*: Springer.
- Chai, D., Wang, L., Chen, K., & Yang, Q (2020). Secure federated matrix factorization. *IEEE Intell Syst*, 01, 1–1. <https://doi.org/10.1109/MIS.2020.3014880>.
- Chai, D., Wang, L., Chen, K., & Yang, Q (2019). Secure federated matrix factorization. CoRR arXiv:1906.05108.
- Dacrema, M.F., Cremonesi, P., & Jannach, D (2019). Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In T. Bogers, A. Said, P. Brusilovsky, & D. Tikk (Eds.) *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019* (pp. 101–109). Copenhagen: ACM. <https://doi.org/10.1145/3298689.3347058>.
- Deldjoo, Y., Anelli, V.W., Zamani, H., Bellogín, A., & Noia, TD (2021). A flexible framework for evaluating user and item fairness in recommender systems. *User Model User Adapt Interact*, 31(3), 457–511. <https://doi.org/10.1007/s11257-020-09285-1>.
- Duriakova, E., Tragos, E.Z., Smyth, B., Hurley, N., Pe na, F.J., Symeonidis, P., Geraci, J., & Lawlor, A (2019). Pdmfrec: a decentralised matrix factorisation with tunable user-centric privacy. In *Proc. of the 13th ACM Conf. on Recommender Systems, RecSys 2019* (pp. 457–461). Copenhagen.
- Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, RS (2011). Fairness through awareness. CoRR arXiv:1104.3913.
- Fierimonte, R., Scardapane, S., Uncini, A., & Panella, M (2017). Fully decentralized semi-supervised learning via privacy-preserving matrix completion. *IEEE Trans Neural Netw Learn Syst*, 28(11), 2699–2711. <https://doi.org/10.1109/TNNLS.2016.2597444>.
- General data protection regulation (2020). (accessed May 31, 2020). <https://gdpr-info.eu/>.
- Gunawardana, A., & Shani, G. (2015). Evaluating recommender systems. In *Recommender Systems Handbook* (pp. 265–308). Springer.
- Guo, Y., Liu, F., Cai, Z., Zeng, H., Chen, L., Zhou, T., & Xiao, N (2021). PREFER: point-of-interest recommendation with efficiency and privacy-preservation via federated edge learning. *Proc ACM Interact Mob Wearable Ubiquit Technol*, 5(1), 13:1–13:25. <https://doi.org/10.1145/3448099>.
- Han, J., Ma, Y., Mei, Q., & Liu, X (2021). Deeprec: On-device deep learning for privacy-preserving sequential recommendation in mobile commerce. In J. Leskovec, M. Grobelnik, M. Najork, J. Tang, & L. Zia (Eds.) *WWW '21: The web conference 2021, virtual event / ljubljana* (pp. 900–911). Slovenia: ACM / IW3C2. <https://doi.org/10.1145/3442381.3449942>.
- Hu, Y., Koren, Y., & Volinsky, C (2008). Collaborative filtering for implicit feedback datasets. In *Proc. of the 8th IEEE Int. Conf. on Data Mining (ICDM 2008)* (pp. 263–272). Pisa: IEEE Computer Society.
- Huang, Z., Chen, H., & Zeng, DD (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans Inf Syst*, 22(1), 116–142.
- Jalalirad, A., Scavuzzo, M., Capota, C., & Sprague, MR (2019). A simple and efficient federated recommender system. In *Proc. of the 6th IEEE/ACM Int. Conf. on Big Data Computing, Applications and Technologies* (pp. 53–58). <https://doi.org/10.1145/3365109.3368788>.
- Jeckmans, A.J.P., Beye, M., Erkin, Z., Hartel, P.H., Lagendijk, R.L., & Tang, Q (2013). Privacy in recommender systems. In *Social Media Retrieval, Computer Communications and Networks* (pp. 263–281). Springer.
- Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., & et al (2019). Advances and open problems in federated learning. arXiv:1912.04977.

- Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K.A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R.G.L., Rouayheb, S.E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B., Gruteser, M., ... Zhao, S (2019). Advances and open problems in federated learning. arXiv:1912.04977.
- Kharitonov, E. (2019). Federated online learning to rank with evolution strategies. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (pp. 249–257).
- Konečný, J., McMahan, B., & Ramage, D (2015). Federated optimization: Distributed optimization beyond the datacenter. CoRR arXiv:1511.03575.
- Konečný, J., McMahan, H.B., Ramage, D., & Richtárik, P (2016). Federated optimization: Distributed machine learning for on-device intelligence. CoRR arXiv:1610.02527.
- Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 426–434). Las Vegas. <https://doi.org/10.1145/1401890.1401944>.
- Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans Knowl Discov Data (TKDD)*, 4(1), 1–24.
- Koren, Y., Bell, R.M., & Volinsky, C (2009). Matrix factorization techniques for recommender systems. *IEEE Comput*, 42(8), 30–37.
- Koren, Y., & Sill, J. (2011). Ordrec: an ordinal model for predicting personalized item rating distributions. In B. Mobasher, R.D. Burke, D. Jannach, & G. Adomavicius (Eds.) *Proc. of the 2011 ACM Conf. on Recommender Systems, RecSys 2011* (pp. 117–124). Chicago: ACM.
- kumar Bokde, D., Girase, S., & Mukhopadhyay, D (2015). Role of matrix factorization model in collaborative filtering algorithm: A survey. CoRR arXiv:1503.07475.
- Li, X., Huang, K., Yang, W., Wang, S., & Zhang, Z (2020). On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020*. Addis Ababa: OpenReview.net. <https://openreview.net/forum?id=HJxNANVtDS>.
- Liang, D., Krishnan, R.G., Hoffman, M.D., & Jebara, T (2018). Variational autoencoders for collaborative filtering. In *Proceedings of 2018 WWW Conference* (pp. 689–698).
- Mansoury, M., Mobasher, B., Burke, R., & Pechenizkiy, M (2019). Bias disparity in collaborative recommendation: Algorithmic evaluation and comparison. In R. Burke, H. Abdollahpouri, E.C. Malthouse, K.P. Thai, & Y. Zhang (Eds.) *Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th ACM Conference on Recommender Systems (RecSys 2019)*, CEUR Workshop Proceedings, Vol. 2440. Copenhagen: CEUR-WS.org. <http://ceur-ws.org/Vol-2440/paper6.pdf>.
- McFee, B., Barrington, L., & Lanckriet, G.R.G (2012). Learning content similarity for music recommendation. *IEEE Trans Audio Speech Lang Process*, 20(8), 2207–2218.
- McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, BA (2017). Communication-efficient learning of deep networks from decentralized data. In *Proc. of 20th Int. Conf. on Artificial Intelligence and Stat.* (pp. 1273–1282). <http://proceedings.mlr.press/v54/mcmahan17a.html>.
- McNee, S.M., Riedl, J., & Konstan, JA (2006). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems* (pp. 1097–1101).
- Mesas, R.M., & Bellogín, A. (2017). Evaluating decision-aware recommender systems. In *Proc. of the 11th ACM Conf. on Recommender Systems* (pp. 74–78). <https://doi.org/10.1145/3109859.3109888>.
- Rendle, S. (2010). Factorization machines. In *ICDM 2010, the 10th IEEE international conference on data mining* (pp. 995–1000). Sydney. <https://doi.org/10.1109/ICDM.2010.127>.
- Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L (2009). BPR: bayesian personalized ranking from implicit feedback. In *Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence* (pp. 452–461).
- Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L (2010). Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010* (pp. 811–820). Raleigh. <https://doi.org/10.1145/1772690.1772773>.
- Rendle, S., & Schmidt-Thieme, L. (2010). Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010* (pp. 81–90). New York. <https://doi.org/10.1145/1718487.1718498>.
- Shi, Y., Larson, M., & Hanjalic, A (2010). List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems* (pp. 269–272).
- Yang, D., Zhang, D., & Qu, B (2016). Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM TIST*, 7(3), 30:1–30:23.
- Yang, Q., Liu, Y., Chen, T., & Tong, Y (2019). Federated machine learning: Concept and applications. *ACM TIST*, 10(2), 12:1–12:19.

- Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., & Yu, H (2019). Federated learning. Morgan & Claypool Publishers.
- Yuan, J., Shalaby, W., Korayem, M., Lin, D., AlJadda, K., & Luo, J (2016). Solving cold-start problem in large-scale recommendation engines: A deep learning approach. In *2016 IEEE Int. Conf. on Big Data, BigData 2016* (pp. 1901–1910). Washington: IEEE Computer Society.
- Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V (2018). Federated learning with non-iid data. CoRR arXiv:[1806.00582](https://arxiv.org/abs/1806.00582).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.