PDMFRec: A Decentralised Matrix Factorisation with Tunable User-centric Privacy

immediate

October 30, 2019

Erika Duriakova[1] Elias Z. Tragos[1] Barry Smyth[1] Neil Hurley[1] Francisco J. Peña[1] Panagiotis Symeonidis[1] James Geraci[2] Aonghus Lawlor[1]

[1]Insight Centre for Data Analytics, University College Dublin
[2]Samsung Electronics Co., Ltd.

Abstract

Conventional approaches to matrix factorisation (MF) typically rely on a centralised collection of user data for building a MF model. This approach introduces an increased risk when it comes to user privacy. In this short paper we propose an alternative, user-centric, privacy enhanced, decentralised approach to MF. Our method pushes the computation of the recommendation model to the user's device, and eliminates the need to exchange sensitive personal information; instead only the loss gradients of local (device-based) MF models need to be shared. Moreover, users can select the amount and type of information to be shared, for enhanced privacy. We demonstrate the effectiveness of this approach by considering different levels of user privacy in comparison with state-of-the-art alternatives.

Key Words: matrix factorisation; decentralised matrix factorisation; privacy aware; rating prediction;

1 Introduction

While the sensitive issue of personal data and user privacy has often informed recommender systems (RS) research, recent initiatives, such as the European Union's General Data Protection Regulation (GDPR) [29], have created a strong imperative for a new generation of RS, which prioritise *privacy by design*. GDPR creates a new set of regulations within which companies must operate, constraining the collection, storage, and use of personal data. The requirements for "data minimisation" and "collection limitation" mandate that only a minimum

amount of personal information should be collected and processed for a specific purpose. This makes it harder for companies to harness the type of user data that has driven RS success of late with the explicit and informed consent of the user.

Central collection of user data has a major privacy risk as evidenced by the frequent privacy breaches that have impacted as many as 1.1 billion users in recent years [14]. Previous attempts to improve the privacy characteristics of RS include data obfuscation ([2, 22]), anonymisation ([18, 24]), differential privacy ([1, 20]) or encryption ([8, 34]). These methods are typically applied to centralised systems and even today most of the state of the art recommendation algorithms offer limited if any privacy assurances [12].

One way to enhance the privacy characteristics of RS has been to develop distributed approaches to recommendation, thereby decentralising the storage of personal information and the computation that is required to generate recommendations. This has been enabled by the improved processing and storage capabilities of modern devices, from phones to TVs; see for example [3, 7, 15, 17] for some recent examples of distributed, device-based approaches.

In this short paper we propose *PDMFRec*, a novel, decentralised approach to MF, which can operate on a user's device, eliminating the need for a central server or for sharing sensitive personal information. Users can set their own privacy levels, selecting subsets of item gradients to share, for the purpose of training a decentralised MF model. We discuss the privacy benefits of this approach with respect to "untraceability" and "anonymity" ([23]) and evaluate different privacy levels for their relationship to recommendation performance and communication cost. Finally, we evaluate the effectiveness of our approach on example datasets, showing that it outperforms the state of the art even with increased privacy levels.

2 Related Work

Distributed and decentralised RS architectures aim to avoid the need to collect data and train RS algorithms centrally, as shown in Table 1a. Distributed methods use a server for the computation and enable the remote devices to learn a global model. Decentralised methods allow the remote nodes to work collaboratively to learn personalised models. Data shared in the network can be the user ratings/preferences (considered as raw data), the model weights or the gradients (Table 1b). Actually, data don't always have the same effect on user privacy. Sharing raw data (ratings, coordinates and interactions) is considered as high privacy risk, because it allows to extract user profiles or locations and link the data back to individual users. Sharing model weights or gradients makes the extraction of meaningful information for user profiles harder, decreasing the privacy risk to medium or low. Thus, shifting from a centralised to a decentralised approach might solve some issues of central collection of information, but also opens an entirely new spectrum of privacy risks when sharing information with fellow nodes [27, 32].

(a)

2-4 **Storage**	**Centralised**	**Computation Distributed**	**Decentralised**
Centralised	[13]	[10, 15]	-
Distributed	-	[7, 11, 17, 19, 27]	[3, 21, 28, 31],`PDMFRec`

(b)

2-4 **Privacy Risk**	**Raw data**	**Data shared Weights**	**Gradients**
High	[13, 21]		
Medium	[10, 17]	[11, 19, 28, 31]	[7]
Low	-	-	[3, 27],`PDMFRec`

Table 1: Comparison of distributed vs decentralised RS approaches and type of data shared vs privacy risk.

Past attempts towards non-centralised learning include distributed deep learning [5, 6], decentralised SGD [9, 33] and parallel SGD [16, 35]. Most works focused on optimising the distributed consensus problem to speed up training and achieve similar accuracy with centralised models, while attempts to consider privacy have also been made [26, 27]. Interestingly though, there is only a handful of works that have focused on decentralised RS, using decentralised MF [3, 17], distributed Factorisation Machines [7, 15] or Federated Learning [4]. Chen et. al. [3] proposed a decentralised MF framework for point of interest recommendations, creating a network of users who learn their model by sharing item gradients with their neighbours. Liao et. al. [17] proposed a decentralised MF algorithm focusing on network distance prediction. Both these approaches require users exchanging location information (in the form of coordinates or geographic location) which is sensitive information according to GDPR. Li et. al. [15] proposed a distributed factorisation machines algorithm, having remote workers sharing their gradients, but assuming that the workers have access to a global repository of data, effectively compromising on user data privacy.

In contrast with past approaches, we provide enhanced user privacy protection and we use different update rules for the distributed consensus problem. In our proposed `PDMFRec`, the neighbourhoods are created using the number of co-rated items between the users, with the users having the option to limit this information for increased privacy. We argue that this is a first step towards limiting the amount and type of non-private information that users share for creating the neighbourhoods and is more privacy preserving than sharing location information as in the past approaches. We can also control the density of the neighbourhood graph increasing the privacy of the users. Additionally, we aim to decrease the memory footprint of our model together with the number of tunable parameters and we demonstrate that this does not affect accuracy.

3 Privacy-enhanced decentralised RS

3.1 `PDMFRec` Overview

The goal of `PDMFRec` is to improve the privacy of users, while maintaining a good level of accuracy. The process of local matrix factorisation computation

performed on user's device (UD) is described in Algorithm 1. Apart from standard inputs into this algorithm such as the ratings matrix $\mathbf{R}$, learning rate η and regulariser term λ, the algorithm also requires the pre-computed weighted network of trusted UDs stored in the form of adjacency matrix $\mathbf{W}$. In the process of decentralised computation, each UD executes its own copy of the algorithm using its own ratings matrix $\mathbf{R}$.

Algorithm 1 `PDMFRec`

Require: set of user-item ratings R, learning rate η, regulariser λ, nearest neighbours N_u of target user u, user-user weighted matrix $\mathbf{W}$

1: **for** e epochs and for each user u **do**
2: **for** $r_{u,i} \in R_u$ **do**
3: $\frac{\partial \mathcal{L}}{\partial \mathbf{p}_u} = -2(r_{u,i} - \mathbf{p}_u \mathbf{q}_i^\top)\mathbf{q}_i - \lambda \mathbf{p}_u$ ▷ Local MF phase
4: $\frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} = -2(r_{u,i} - \mathbf{p}_u \mathbf{q}_i^\top)\mathbf{p}_u - \lambda \mathbf{q}_i$
5: $\mathbf{p}_u = \mathbf{p}_u - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{p}_u}$, and $\mathbf{q}_i = \mathbf{q}_i - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{q}_i}$
6: **for** $u' \in N_u$ **do** ▷ Share phase
7: $\frac{\partial \mathcal{L}}{\partial \mathbf{q}_i}^{u'} = \mathbf{W}_{u,u'} \frac{\partial \mathcal{L}}{\partial \mathbf{q}_i}$
8: send $\frac{\partial \mathcal{L}}{\partial \mathbf{q}_i}^{u'}$ to u'
9: **end for**
10: **for** $u' \in N_u$ **do** ▷ Collect phase
11: Collect $\frac{\partial \mathcal{L}}{\partial \mathbf{q}_{i'}}^{u'}$ from user u'
12: $\mathbf{q}_{i'} = \mathbf{q}_{i'} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{q}_{i'}}^{u'}$
13: **end for**
14: **end for**
15: **end for**

Ensure: user and item latent factors $\mathbf{p}$ and $\mathbf{q}$

Overall, `PDMFRec` consists of four key phases:

(1) Neighbourhood Creation: To build the neighbourhood of a UD, we use the information of co-rated items between each pair of UDs and we assign the weight using the cosine similarity as follows: $w = \frac{|I_u \cap I_{u'}|}{\sqrt{|I_u| \cdot |I_{u'}|}}$, where I_u represents the set of rated items by UD u. Further to control the overall density of the network of neighbours, we define a threshold value as follows: $|I_u \cap I_{u'}| \geq threshold$, which disables links between users that have less than *threshold* number of co-rated items, limiting the overall density of this network.

(2) Local MF phase: Each UD proceeds by first calculating user and item gradients, which are then in turn used to update the user and item latent factors (Algorithm 1 lines 3-5).

(3) Share phase: Each UD exchanges the newly computed item gradients with their set of neighbours. Before the send operation, a weight stored in $\mathbf{W}$ is applied to each gradient (Algorithm 1, lines 6-9). This step ensures that higher weight is placed on gradients sent to neighbours whose taste is more similar to the current UD.

(4) Collect phase: Once all UDs have exchanged the newly computed item gradients, each UD updates the local item latent factors for which it has received the gradients. Note that it is possible for a single UD to receive more

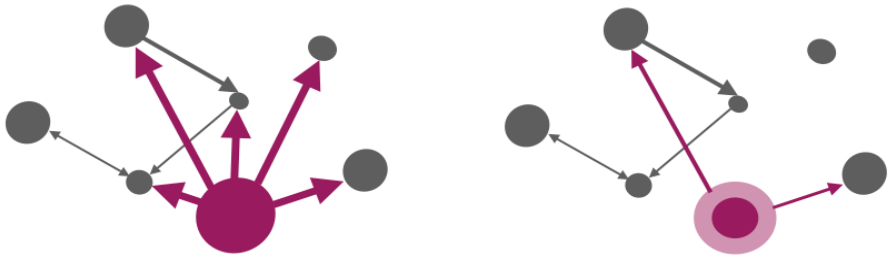


Figure 1: The effect of user privacy level on network density.

than one item gradient for any particular item. In such case we simply calculate the average between all such gradients (see Algorithm 1 lines 10-13).

3.2 Complexity Analysis

The **time complexity** for each user u is composed of three parts for calculating, sharing and collecting the gradients from the neighbours, (Lines 3-5, 6-9 and 10-13 of Algorithm 1). For a single epoch the time complexity of calculating the gradients is $\mathcal{O}(|R_u| \cdot d)$, the time complexity of sharing the data is $\mathcal{O}(|R_u| \cdot |N_u|)$ and the time complexity of collecting the gradients is $\mathcal{O}(|I| \cdot |N_u| \cdot d)$. Given that $|I| \geq |R_u|$, the total time complexity for an epoch is $\mathcal{O}(|I| \cdot |N_u| \cdot d)$. Each user u will need $\mathcal{O}(|I| \cdot d + d) = \mathcal{O}(|I| \cdot d)$ **storage space** in her device for the item latent factors matrix $\mathbf{Q}$ and her own latent factors vector $\mathbf{p}_u$. The **communication complexity** for each user u is determined by the share phase (Lines 6-9 of Algorithm 1), and the collect phase, (Lines 10-13). The communication complexity for the share phase is $\mathcal{O}(|N_u| \cdot |R_u| \cdot d)$ and the communication complexity for the collect phase is $\mathcal{O}(|N_u| \cdot |I| \cdot d)$. Given that $|I| \geq |R_u|$, the total communication complexity is $\mathcal{O}(|N_u| \cdot |I| \cdot d)$.

3.3 Privacy analysis

Sharing item gradients in decentralised systems, as opposed to item latent factors, has been proved to be privacy preserving, although it may reveal some information about user interactions [32]. `PDMFRec` goes one step further, and allows users to hide a fraction of their rating history. This way, user's full rating profile remains protected, since not all item identifiers are shared among neighbours. We prove, that with this approach, we can still achieve very competitive accuracy. With this in mind, `PDMFRec` consists of two extra privacy levels as opposed to past approaches. In the first privacy level (L1) a subset of randomly chosen items is hidden for each UD only during the neighbourhood creation, while in the second level (L2) the same subset is also hidden during the training phase.

Hiding increasing fraction of items per UD affects the overall dataset statistics making it harder to find other UDs based on co-rated items. Figure 1 illustrates this effect, where on the left side, a user (purple node) shares all item

identifiers for the neighbourhood creation and on the right side, the same user decides to keep a large part of his profile private (illustrated by smaller node size), thus effectively decreasing the number of similar neighbours.

By setting the threshold for neighbourhood creation we can also control the density of the overall neighbourhood network. According to [23], "anonymity is the state of being not identifiable within a set of subjects, the anonymity set", so the larger the "anonymity set size" the higher the privacy of the users ([30]). However, on a flip side, this also leads to increased communication costs. Our results show that we can find the right balance between very good accuracy with high "anonymous set sizes" (or density) and low communication cost. Further, during the training process it is not possible to identify any particular UD in the `PDMFRec` framework through the gradient exchange phase, since the weights identifying the UD-UD similarity are applied by the sender and the messages have no identifiers, such as user id, profile data, or IP addresses. This ensures "sender anonymity", which is also linked with "untraceability" [23], since the receiver can not identify the sender. Any sender identifiers such as IP can be anonymised in intermediate servers using redactable signatures that remove parts of the data without breaking the data integrity [25].

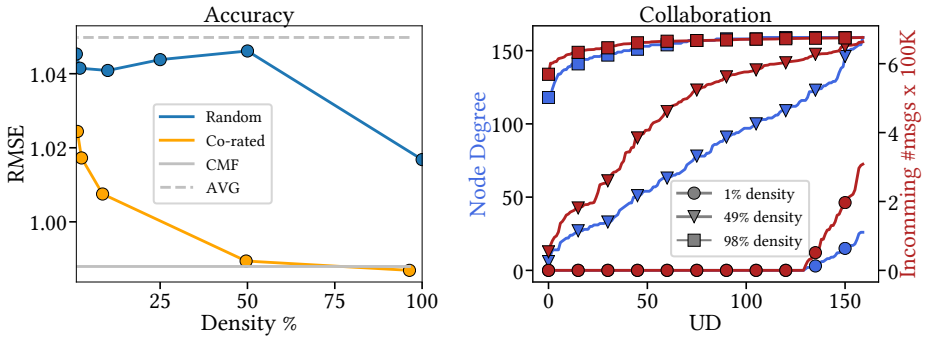
Additional steps for privacy will be considered in future versions of the algorithm, considering differential privacy as in [1], adding noise to the gradients that are shared and also sharing random gradients for items, with which the user has not interacted. Additionally, a more privacy enhanced method for creating the neighbourhoods without any prior knowledge about user preferences is pursued.

4 Experimental Evaluation

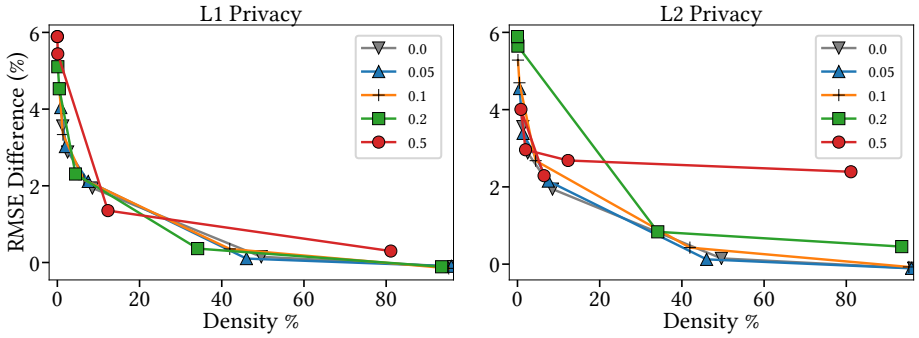
We begin this section by first explaining the datasets and evaluation protocols that will be used in our experiments.

Datasets: We run our experimentsd on a single server using Message Passing Interface (MPI). Since this is a mock-up of decentralised approach, the overall computation computes $|U|$ independent local MF models, each storing ($\mathbb{R}^{d\times|I|}$) latent factor matrices. Therefore, to allow the simulation to run on single server, we use randomly sampled subsets of the following two datasets: **ML100K** with 160 users, 1470 items, 18465 ratings and 18.9 co-rated items between any pair of users on average; **Epinions** with 311 users, 14584 items, 21296 ratings and 1.7 co-rated items on average. We use a 80-20 split such that 80% of data is used during training and the remaining 20% is used during testing.

Evaluation Protocol: When evaluating the effectiveness of the decentralisation process in `PDMFRec` we compare against basic MF model using SGD approach. We argue that more sophisticated MF models would not provide a fair comparison, since `PDMFRec` does not contain any further enhancements yet (this will be considered in future work). However, we do compare `PDMFRec` against the state-of-the-art decentralised RS such as [3].



(a) The impact of neighbourhood density showing (i) the change in model accuracy and (ii) the overall collaboration across all UDs.



(b) The impact of increasing fraction of hidden items in two levels of privacy showing the change in RMSE as compared to Centralised MF using SGD.

Figure 2: Accuracy trade-offs with respect to (a)neighbourhood density and (b) fraction of hidden data in two levels of privacy.

Hyper-parameters: All models used in the evaluation process are dynamically reducing the learning rate when the algorithm reaches local plateau in the loss function. If the number of epochs without loss improvement is greater then the predefined *patience*, the learning rate is updated as follows: $\eta = \eta * factor$. In our experiments we set $\eta = 0.01$, $\lambda = 0.001$, $patience = 2$ and $factor = 0.5$.

4.1 Neighbourhood Density Analysis

Here, we provide insights into `PDMFRec` in terms of accuracy, incoming communication per UD and neighbourhood network degree distribution when the overall network density changes. The intuition is that, in decentralised MF scenarios, we naturally aim to minimise the communication cost. However, on the flip side, a very sparse adjacency matrix results in sparse collaboration between users during the training process affecting the overall model accuracy. Moreover, note that decreasing density also affects the "anonymity" (see Section 3.3). We compare our approach against the centralised MF with basic stochastic gradient

descent, a base case using the average model (which recommends the average rating per user) and the PDMFRec with neighbourhoods of different densities using a random graph generator.

The accuracy of PDMFRec with randomly selected neighbours outperforms the average recommender (Figure 2(a)). More interestingly PDMFRec with neighbours selected based on the information from co-rated items can reach the accuracy of the centralised model when the overall neighbourhood density is just over 50%. This is because with 50% density neighbourhood all UDs receive incoming communication allowing them to learn their models collaboratively.

	ML100K			Epinions		
Density	1%	49%	97%	1%	11%	30%
PDMFRec	**1.0244**	**0.9894**	**0.9869**	**1.1761**	**1.1616**	1.1609
Chen et al. [3]	1.0337	1.5496	1.6040	1.1800	1.167	**1.1583**

Table 2: Performance of PDMFRec vs. state-of-the-art.

4.2 User Controlled Privacy Settings

Here we demonstrate the behaviour of PDMFRec under different user-set privacy settings, where users can decide to keep a fraction of their rating profile private. For this purpose PDMFRec randomly selects a subset of item identifiers that will be excluded from the training process. We study the effect of increasingly higher fraction of hidden item identifiers on the accuracy of PDMFRec. We divide this experiment into two parts, with respect to the two levels of privacy discussed in Section 3.3.

L1 Privacy: The overall accuracy of PDMFRec does not suffer much from an increasing fraction of hidden items per UD (Figure 2(b)). Up to 50% of items can be hidden with a very little impact on accuracy, as the difference in accuracy between 0% and 50% L1 Privacy on highest density neighbourhood network is just 1%. Hiding increasing fraction of items per UD changes the overall dataset statistics. As we increase the fraction of removed items the average number of co-rated items decreases, making it harder to construct a dense neighbourhood network (see Section 3.3).

L2 Privacy: When applying the second level of privacy on PDMFRec, increasing the fraction of hidden items per UD during the model computation has a greater impact on the overall model accuracy. However, it can be observed that each user can hide up to 20% of their item gradients without significantly affecting the overall accuracy. Note that as discussed in Section 3.3, when L2 privacy is applied, the training process complies more with the GDPR requirement for "data minimisation" and "collection limitation", since users can hide a large fraction of their profile and only share parts of their gradients, with still a very high accuracy.

4.3 Comparison with State-of-the-art

We compare PDMFRec against the state-of-the-art decentralised point-of-interest recommender by Chen et al.[3]. Table 2 shows that PDMFRec significantly out-

performs Chen et al. on the **ML100K** dataset. Moreover, the approach of Chen at al. seems to suffer from large neighbourhoods, making this method unsuitable for models with neighbourhoods of higher density and therefore decreasing the "anonymity set size", which lowers the level of privacy. In the case of **Epinions** dataset both models have comparable accuracy with increasing neighbourhood density. Recall that the model of Chen et al. shares the full set of item gradients and requires two item latent factors (one local and one global) per UD. Moreover both latent factors require their own regularises effectively increasing the overall number of parameters that need to be tuned and also increasing the overall memory foot print per UD. This demonstrates that comparable performance can be reached without the need for extra item latent space and additional tuning parameters.

5 Conclusions

The main contribution of this paper is a novel user-centric, privacy enhanced, decentralised MF method for RS, providing state-of-the-art levels of recommendation performance while giving users full control over their data and privacy settings. The proposed method allows the MF model to be trained in a decentralised way, without the need for users to share their raw data, avoiding the need for a central server and thereby eliminating many of the traditional privacy risks associated with conventional recommender systems. We also show how this increased level of privacy does not come at any cost to recommendation performance: the proposed method can achieve accuracy levels that are comparable to those available from centralised, non-privacy preserving approaches.

The work is supported by the Insight Centre for Data Analytics under Grant Number SFI/12/RC/2289_P2 and Samsung Research, Samsung Electronics Co., Seoul, Republic of Korea.

References

[1] Armita Afsharinejad and Neil Hurley. 2018. Performance Analysis of a Privacy Constrained kNN Recommendation Using Data Sketches. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, New York, NY, USA, 10–18. https://doi.org/10.1145/3159652.3159673

[2] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2012. The impact of data obfuscation on the accuracy of collaborative filtering. *Expert Systems with Applications* 39, 5 (2012), 5033–5042.

[3] Chaochao Chen, Ziqi Liu, Peilin Zhao, Jun Zhou, and Xiaolong Li. 2018. Privacy Preserving Point-of-Interest Recommendation Using Decentralized Matrix Factorization. In *Proceedings of the Thirty-Second AAAI Confer-*

ence on Artificial Intelligence. https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16123

[4] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated meta-learning for recommendation. *arXiv preprint arXiv:1802.07876* (2018).

[5] Dipankar Das, Sasikanth Avancha, Dheevatsa Mudigere, Karthikeyan Vaidynathan, Srinivas Sridharan, Dhiraj Kalamkar, Bharat Kaul, and Pradeep Dubey. 2016. Distributed deep learning using synchronous stochastic gradient descent. *arXiv preprint arXiv:1602.06709* (2016).

[6] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. 2012. Large scale distributed deep networks. In *Advances in neural information processing systems*. 1223–1231.

[7] Yue Ding, Dong Wang, Xin Xin, Guoqiang Li, Daniel Sun, Xuezhi Zeng, and Rajiv Ranjan. 2018. SCFM: Social and crowdsourcing factorization machines for recommendation. *Applied Soft Computing* 66 (2018), 548–556.

[8] Zekeriya Erkin, Thijs Veugen, Tomas Toft, and Reginald L Lagendijk. 2012. Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE transactions on information forensics and security* 7, 3 (2012), 1053–1066.

[9] Rainer Gemulla, Erik Nijkamp, Peter J Haas, and Yannis Sismanis. 2011. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 69–77.

[10] Peng Han, Bo Xie, Fan Yang, and Ruimin Shen. 2004. A scalable P2P recommender system based on distributed collaborative filtering. *Expert Systems with Applications* 27, 2 (2004), 203 – 210. https://doi.org/10.1016/j.eswa.2004.01.003

[11] Yaochen Hu, Di Niu, and Jianming Yang. 2018. Stochastic Distributed Optimization for Machine Learning from Decentralized Features. *CoRR* abs/1812.06415 (2018). arXiv:1812.06415 http://arxiv.org/abs/1812.06415

[12] Sarika Jain, Anjali Grover, Praveen Singh Thakur, and Sourabh Kumar Choudhary. 2015. Trends, problems and solutions of recommender system. In *International Conference on Computing, Communication & Automation*. IEEE, 955–958.

[13] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug. 2009), 30–37. https://doi.org/10.1109/MC.2009.263

[14] Paige Leskin. 2018. The 21 scariest data breaches of 2018. (2018). https://www.businessinsider.com/data-hacks-breaches-biggest-of-2018-2018-12

[15] Mu Li, Ziqi Liu, Alexander J. Smola, and Yu-Xiang Wang. 2016. DiFacto: Distributed Factorization Machines. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM '16)*. ACM, New York, NY, USA, 377–386. https://doi.org/10.1145/2835776.2835781

[16] Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. 2015. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Advances in Neural Information Processing Systems*. 2737–2745.

[17] Yongjun Liao, Wei Du, Pierre Geurts, and Guy Leduc. 2013. DMFSGD: A Decentralized Matrix Factorization Algorithm for Network Distance Prediction. *IEEE/ACM Transactions on Networking* 21, 5 (Oct. 2013), 1511–1524.

[18] Zhifeng Luo, Shuhong Chen, and Yutian Li. 2013. A distributed anonymization scheme for privacy-preserving recommendation systems. In *2013 IEEE 4th International Conference on Software Engineering and Service Science*. IEEE, 491–494.

[19] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*. 1273–1282. http://proceedings.mlr.press/v54/mcmahan17a.html

[20] Frank McSherry and Ilya Mironov. 2009. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 627–636.

[21] Thanasis G Papaioannou, Jean-Eudes Ranvier, Alexandra Olteanu, and Karl Aberer. 2012. A decentralized recommender system for effective web credibility assessment. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 704–713.

[22] Rupa Parameswaran and Douglas M Blough. 2007. Privacy preserving collaborative filtering using data obfuscation. In *2007 IEEE International Conference on Granular Computing (GRC 2007)*. IEEE, 380–380.

[23] Andreas Pfitzmann and Marit Köhntopp. 2001. Anonymity, unobservability, and pseudonymity, a proposal for terminology. In *Designing privacy enhancing technologies*. Springer, 1–9.

[24] Juan E Rubio, Cristina Alcaraz, and Javier Lopez. 2017. Recommender system for privacy-preserving solutions in smart metering. *Pervasive and Mobile Computing* 41 (2017), 205–218.

[25] Kai Samelin, Henrich C. Pöhls, Arne Bilzhause, Joachim Posegga, and Hermann de Meer. 2012. Redactable Signatures for Independent Removal of Structure and Content. In *Information Security Practice and Experience*, Mark D. Ryan, Ben Smyth, and Guilin Wang (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 17–33.

[26] Yu-Bo Sheng and Lan Zhou. 2017. Distributed secure quantum machine learning. *Science Bulletin* 62, 14 (2017), 1025–1029.

[27] Reza Shokri and Vitaly Shmatikov. 2015. Privacy-Preserving Deep Learning. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*. ACM, New York, NY, USA, 1310–1321. https://doi.org/10.1145/2810103.2813687

[28] Hanlin Tang, Xiangru Lian, Ming Yan, and Jin Liu. 2018. D^2: Decentralized Training over Decentralized Data. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, 4848 – 4856. 35th International Conference on Machine Learning (ICML 2018); Conference Location: Stockholm, Sweden; Conference Date: July 10-15, 2018.

[29] European Union. 2016. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). (2016). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX\%3A32016R0679

[30] Isabel Wagner and David Eckhoff. 2018. Technical privacy metrics: a systematic survey. *ACM Computing Surveys (CSUR)* 51, 3 (2018), 57.

[31] Zhangyang Wang, Xianming Liu, Shiyu Chang, Jiayu Zhou, Guo-Jun Qi, and Thomas S Huang. 2015. Decentralized recommender systems. *arXiv preprint arXiv:1503.01647* (2015).

[32] Feng Yan, Shreyas Sundaram, S V N Vishwanathan, and Yuan Qi. 2013. Distributed Autonomous Online Learning: Regrets and Intrinsic Privacy-Preserving Properties. *IEEE Transactions on Knowledge and Data Engineering* 25, 11 (Sept. 2013), 2483–2493.

[33] Kun Yuan, Qing Ling, and Wotao Yin. 2016. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization* 26, 3 (2016), 1835–1854.

[34] Justin Zhan, Chia-Lung Hsieh, I-Cheng Wang, Tsan-Sheng Hsu, Churn-Jung Liau, and Da-Wei Wang. 2010. Privacy-preserving collaborative recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40, 4 (2010), 472–476.

[35] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. 2010. Parallelized stochastic gradient descent. In *Advances in neural information processing systems*. 2595–2603.