Data Masking for Recommender Systems: Prediction Performance and Rating Hiding

Manel Slokom
TU Delft
Netherlands
m.slokom@tudelft.nl

Martha Larson
Radboud University and TU Delft
Netherlands
m.larson@cs.ru.nl

Alan Hanjalic
TU Delft
Netherlands
A.Hanjalic@tudelft.nl

ABSTRACT

Data science challenges allow companies, and other data holders, to collaborate with the wider research community. In the area of recommender systems, the potential of such challenges to move forward the state of the art is limited due to concerns about releasing user interaction data. This paper investigates the potential of privacy-preserving data publishing for supporting recommender system challenges. We propose a data masking algorithm, Shuffle-NNN, with two steps: Neighborhood selection and value swapping. Neighborhood selection preserves valuable item similarity information. The data shuffling technique hides (i.e., changes) ratings of users for individual items. Our experimental results demonstrate that the relative performance of algorithms, which is the key property that a data science challenge must measure, is comparable between the original data and the data masked with Shuffle-NNN.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

ACM RecSys 2019 Late-breaking Results, 16th-20th September 2019, Copenhagen, Denmark

KEYWORDS

Recommender systems, privacy-preserving data publishing, data masking

INTRODUCTION TO DATA MASKING WITH SHUFFLE-NNN

We propose a masking approach, *Shuffling Non-Nearest-Neighbors* (Shuffle-NNN), which changes (i.e., hides) a large proportion of the values of the ratings in the original user-item matrix, $\mathcal{R}$, to create a masked data set, $\mathcal{R}'$. Our work is a step towards masked data that can be publicly released for data science challenges. For this reason, we adopt a relative-rank success criterion: given a set of recommender algorithms to be compared, the relative ranking of the algorithms trained and tested on $\mathcal{R}$ and on $\mathcal{R}'$ must be maintained. Shuffle-NNN is motivated by the observation that privacy-preserving techniques [13] (privacy-preserving data publishing [3], privacy-preserving data mining [1]) have not yet been systematically applied to the data released for recommender system challenges.

Shuffle-NNN generates a masked data by changing a large portion of values of the ratings in a user's profile. We chose a data shuffling technique because of its previous success in masking numerical data in other domains [11].

Shuffle-NNN works as follows. Our overall strategy is to shuffle ratings in a way that maintains key item-item similarities in the data set. First, we determine the item neighborhoods, i.e., the most similar item for every item, and join them, giving us an overall set of critical items. Then, we shuffle ratings for items not in this set (non-critical items, i.e., the rest of the items). Shuffle-NNN has two parameters. We fix the neighbhorhood size $k = 40$, since this value has been shown to be effective in practice[1]. Given k, we set θ (a minimum threshold on item-similarity that must be met for inclusion in a neighborhood) by running some test rankings. Our exploratory experiments showed that a range of θ values can be effective, and that θ can be determined using a subset of the algorithms to be ranked (meaning that it can be set in-house before releasing data).

Recall that the goal of this paper is not to demonstrate the absolute performance of the algorithms, but rather to evaluate if the relative performance of algorithms is the same on the original and on the masked data. For this purpose, we need a selection of classic recommender algorithms, ranging from baselines that are known not to yield state-of-the-art performance, to current algorithms. We carry out experiments on both ranking prediction, and on classic rating prediction tasks. The algorithms for ranking prediction are: *Most Popular*, *KNN* is user/item-based K-Nearest Neighbor. *(BPRFM)* [5] and *BPRMF*. We followed the hyperparameters tested in [9].The algorithms for rating prediction are: *ItemKNN* [12], *UserKNN*, *SlopeOne* [8],*Co-clustering* [4]. matrix factorization *(MF)*, Biased Matrix Factorization *(BMF)* [7] and *(SVD++)* [6]. We also test baseline algorithms: *Average-Item* and *Average-User* which use the average rating value of a user or item for predictions.

[1] http://www.mymedialite.net/examples/datasets.html

Table 1: Statistics of the data sets used in our experiments and analysis.

data set	#Users	#Items	#Ratings	Range	Av.rating	Density(%)	Variance
MovieLens 100k/ Random empirical /Masked	943/-/-	1682 /-/-	100.000/-/-	[1,5] /-/-	3.529/2.997/3.529	6.30 /-/-	1.1256/ 1.415/ 1.125
GoodBook/ Random empirical /Masked	53424/-/-	10000/-/-	981.756/-/-	[1,5]/-/-	3.8565/ 3.858/3.856	0.18/-/-	0.9839/ 0.9837/0.9838

Table 2: The ranking prediction performance on the original data, on the masked data and on the empirical data. For the masked MovieLens data we used $\theta = 0.4$ and for the masked GoodBook data we used $\theta = 0.15$.

		R@5 / R@10 for MovieLens data set		R@5 / R@10 for GoodBook data	
		Original	*Masked*	*Original*	*Masked*
The relative ranking of algorithms	*UserKNN*	0.093/0.177	0.093/0.178	0.434/0.595	0.433/0.594
	BPRMF	0.084/0.165	0.085/0.166	0.398/0.575	0.396/0.571
	BPRFM	0.082/ 0.159	0.082/0.159	0.387/0.567	0.382/0.563
	ItemKNN	0.079/ 0.156	0.079/0.155	0.315/0.452	0.314/0.451
	Popular	0.058/ 0.099	0.064/0.108	0.033/0.053	0.028/0.059

EXPERIMENTAL FRAMEWORK

The experiments are implemented using WrapRec [10]. We test on two publicly available data sets (cf. Table 1). We choose MovieLens 100k [2], since it is well understood, and Goodbooks-10K [3], since it is larger and sparser.

[2]https://grouplens.org/datasets/movielens/

[3]https://www.kaggle.com/philippsp/book-recommender-collaborative-filtering-shiny/data

COMPARATIVE ALGORITHM RANKING

Ranking Prediction Performance

We train and test the algorithms for ranking prediction both on the original data and and the masked data. Results are given in Table 2. It can be seen that the comparative algorithm ranking is maintained. In other words, for all cases, the best algorithm on the masked data is also the best algorithm on the original data and the worst algorithm on the masked data is also the worst algorithm on the original data. These results demonstrate the success of Shuffle-NNN. In Table 2, we report results for specific values of θ. However, relative ranking is actually maintained for a range of values of θ (not shown here). In addition to Recall@5 and Recall@10, we found also that Precision@5/ Precision@10 maintains the same relative ranking of algorithms.

Interestingly, when we tested our algorithms on *empirical data* (which we generate by replacing individual values with values drawn randomly from the global distribution of ratings in the original data), the comparative ranking was also preserved. This means that Shuffle-NNN is sufficient, but

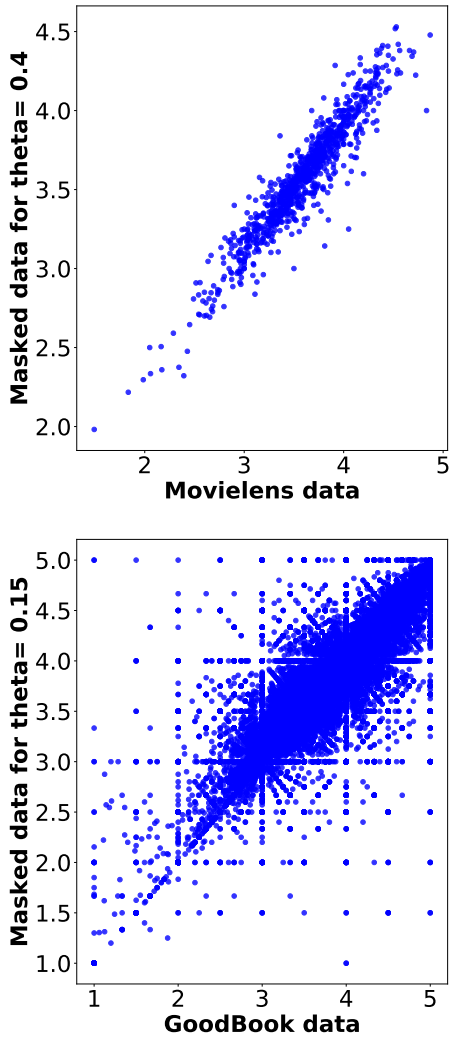

Figure 1: Average user ratings original vs. masked data (masked using θ= 0.4 for MovieLens; $\theta = 0.15$ for GoodBook).

Table 3: The prediction performance on the original data, on the masked data and on the empirical data. Note: Lower RMSE is better. The colors show the ranking of algorithms. For the masked MovieLens data we used $\theta = 0.4$ and for the masked GoodBook data we used $\theta = 0.15$.

	RMSE for MovieLens data set		RMSE for GoodBook data set	
	Original	*Masked*	*Original*	*Masked*
The relative ranking of algorithms	SVD++ (0.902)	SVD++ (0.957)	BMF (0.822)	BMF (0.85)
	BMF (0.911)	BMF (0.962)	SVD++ (0.825)	SVD++ (0.854)
	ItemKNN (0.918)	MF (0.968)	ItemKNN (0.826)	ItemKNN (0.861)
	MF (0.929)	ItemKNN (0.972)	MF (0.856)	MF (0.873)
	UserKNN (0.935)	UserKNN (0.979)	SlopeOne (0.865)	UserKNN (0.894)
	SlopeOne (0.937)	SlopeOne (0.986)	UserKNN (0.868)	SlopeOne (0.903)
	Co-Clustering (0.974)	Co-Clustering (1.026)	Average-User (0.883)	Average-User (0.913)
	Average-Item (1.023)	Average-Item (1.030)	Co-Clustering (0.891)	Co-Clustering (0.924)
	Average-User (1.042)	Average-User (1.081)	Average-Item (0.948)	Average-Item (0.947)

not actually necessary in the case of Top-N recommendation. Our conclusion from these results is that (at least for these data sets) the values of the ratings are not important if the goal is a relative ranking among algorithms. What is important is that our masked data sets maintain information on which items were rated.

Rating Prediction Performance

To dig deeper, we carry out rating prediction experiments. Results are reported in Table 3. Here, we observe that the relative ranking is well maintained between the original data and masked data (although not perfectly). Our focus here is on relative performance, but it is interesting to note that the absolute RMSE on the masked data remains within 5% of its value on the original data. In contrast to the ranking-prediction results, we found that the empirical distribution does less well in maintaining the ranking than Shuffle-NNN.

RATING HIDING

Next, we discuss the ability of Shuffle-NNN to hide ratings. A rating is considered hidden if its value changes between the masked data and the original data [2]. First, we look at the global proportion of ratings hidden in the masked data. We find that at our operating point, Shuffle-NNN achieved a rating hiding percentage of 0.7 for MovieLens data and 0.68 for the GoodBook data.

Then, we look at the impact of masking at the user level. Figure 1 illustrates the relationship between average user ratings before and after masking. Figure 2 shows, for different level of hidden ratings, for how many users that level was achieved. The average user rating of the masked data is

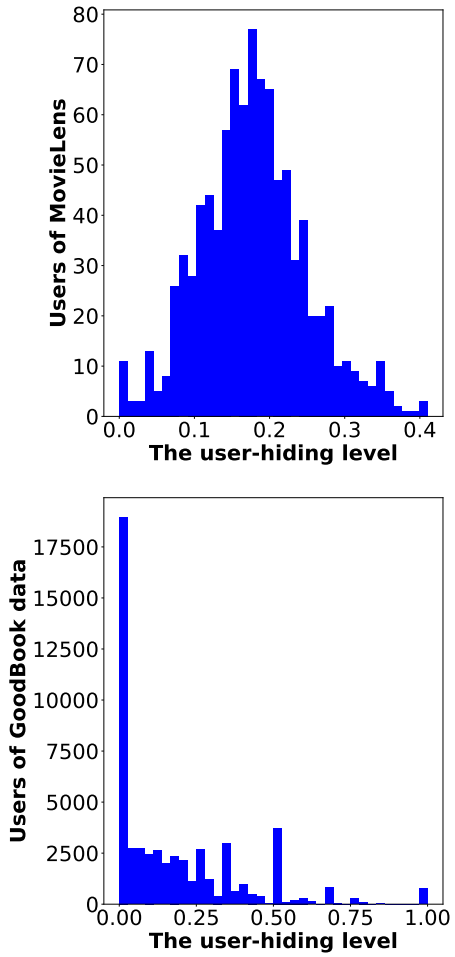

Figure 2: Average user ratings original vs. masked data (masked using θ= 0.4 for MovieLens; $\theta = 0.15$ for GoodBook).

impacted, but still correlated with the original values. The protection varies per user, but is relatively high. For the GoodBook data, there is a large number of users have a hiding level equal to zero. On average these users have rated less than 30 books.

CONCLUSIONS AND OUTLOOK

Our overall conclusion is that data masking has great potential for data science challenges. It is possible to develop a masking approach, such that masked data can be used to train and test algorithms with little impact on the *relative* performance of algorithms. Shuffle-NNN provides valuable evidence about what information can be removed from the user-item matrix and what information should be maintained. When the critical items list is larger than the list of items that is shuffled, it is easier to reconstruct the original values. Future work will investigate the difficulty of reconstructing the original values from the shuffled data, which is an issue important to consider in cases where the critical items are in the majority. We note that even modest levels of rating hiding can support deniability.

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. 2000. Privacy-preserving Data Mining. *SIGMOD Rec.* 29, 2 (2000), 439–450.
[2] Elisa Bertino, Igor Nai Fovino, and Loredana Parasiliti Provenza. 2005. A framework for evaluating privacy preserving data mining algorithms. *Data Mining and Knowledge Discovery* 11, 2 (2005), 121–154.
[3] Benjamin CM Fung, Ke Wang, Ada Wai-Chee Fu, and S Yu Philip. 2010. *Introduction to privacy-preserving data publishing: Concepts and techniques.* Chapman and Hall/CRC.
[4] T. George and S. Merugu. 2005. A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the Fifth IEEE Conference on Data Mining (ICDM'05)*. 625–628.
[5] Weiyu Guo, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Personalized Ranking with Pairwise Factorization Machines. *Neurocomput* 214 (2016), 191–200.
[6] Yehuda Koren. 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 (KDD '08)*. ACM.
[7] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer Society Press* 42, 8 (2009), 30–37.
[8] Daniel Lemire and Anna Maclachlan. 2005. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM, 471–475.
[9] Babak Loni, Roberto Pagano, Martha Larson, and Alan Hanjalic. 2019. Top-N Recommendation with Multi-Channel Positive Feedback Using Factorization Machines. *ACM Trans. Inf. Syst.* 37, 2 (2019), 15:1–15:23.
[10] Babak Loni and Alan Said. 2014. WrapRec: An Easy Extension of Recommender System Libraries. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys '14)*. ACM, 377–378.
[11] Krishnamurty Muralidhar and Rathindra Sarathy. 2006. Data shuffling: A new masking approach for numerical data. *Management Science* 52, 5 (2006), 658–670.
[12] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM.
[13] Vicenç Torra. 2017. *Masking Methods.* Springer International Publishing, Cham, 191–238.