BlurM(or)e: Revisiting Gender Obfuscation in the User-Item Matrix*

Christopher Strucks
Radboud University
Netherlands
chr.strucks@gmail.com

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

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

ABSTRACT

Past research has demonstrated that removing implicit gender information from the user-item matrix does not result in substantial performance losses. Such results point towards promising solutions for protecting users' privacy without compromising prediction performance, which are of particular interest in multistakeholder environments. Here, we investigate BlurMe, a gender obfuscation technique that has been shown to block classifiers from inferring binary gender from users' profiles. We first point out a serious shortcoming of BlurMe: Simple data visualizations can reveal that BlurMe has been applied to a data set, including which items have been impacted. We then propose an extension to BlurMe, called BlurM(or)e, that addresses this issue. We reproduce the original BlurMe experiments with the MovieLens data set, and point out the relative advantages of BlurM(or)e.

CCS CONCEPTS

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

KEYWORDS

Recommender Systems, Privacy, Data Obfuscation

1 INTRODUCTION

When users rate, or otherwise interact with items, they may be aware that they are providing a recommender system with preference information. Less likely, is, however, that users know that interaction information can implicitly hold sensitive personal information. In this paper, we focus on the problem of binary gender information in the user-item matrix, which can be inferred by using a gender classifier. The state of the art in gender obfuscation for recommender system data, is to our knowledge, represented by Weinsberg et. al. [11], who propose a gender obfuscation approach for a user-item matrix of movie ratings, called BlurMe. Successful obfuscation means that a user's gender cannot be correctly inferred by a classifier that has been previously trained on other users' rating data. BlurMe accomplishes this obfuscation without a substantial impact on the prediction performance of the recommender system that is trained on the obfuscated data. Our study of BlurMe has revealed that it has a serious shortcoming. In this paper, we discuss this issue, and propose an extension to BlurMe, called BlurM(or)e, that addresses it. We test BlurM(or)e against a reimplementation of BlurMe, reproducing experiments from [11].

Obfuscation is an important tool to maintaining user privacy, alongside other tools such as encryption. Obfuscation is widely studied in other areas, but does not receive a great amount of attention in the area of recommender systems, exceptions are [2, 8]. Obfuscation can be added to the user-item matrix by users themselves, freeing them from an absolute dependency on the service provider to secure their data and use it properly. In [1, 2] the user can decide what data to reveal and how much protection is put on the data. Even trusted service providers can have issues, such as breaches, or data being acquired and used inappropriately [7].

The main contributions of this paper are:

- A discussion of a flaw we discovered in BlurMe.
- An extension to BlurMe, called BlurM(or)e, that addresses this issue.
- A set of experiments, whose results demonstrate the ability of BlurM(or)e to obfuscate binary gender in the user-item matrix with minimal impact on recommendation performance.

The paper is organized as follows. In Section 2, we cover the related work, before going on to present the shortcoming of BlurMe and our proposed improvement BlurM(or)e in Section 3. Next, we present our experiments and results in Section 4, and in Section 5 we discuss our reproduction of BlurMe[1]. We finish in section 6 with a discussion and conclusion.

2 BACKGROUND AND RELATED WORK

In this section, we discuss work most closely related to our own.

2.1 Obfuscating the User-Item matrix

In order to protect user demographic information in the user-item matrix, researchers have suggested data obfuscation. Data obfuscation (a.k.a. data masking) describes the process of hiding the original, possibly sensitive data with modified or even fictional data [10]. The goal is to protect the privacy of users, while maintaining the utility of the data. Data obfuscation can be done in several ways, e.g., [9] used lexical substitution as obfuscation mechanism for text or [5] used user groups instead of individual users to hide personal information from the recommender system. In BlurMe [11], the authors found that it is possible to infer the gender of users from their rating histories via basic machine learning classifiers. They proposed an algorithm, BlurMe, which successfully obfuscates the gender of a user, thereby blocking gender inference. BlurMe basically adds ratings to every user profile that are typical for the opposite gender, and is currently state of the art. The best performing BlurMe obfuscation strategy, the *greedy* strategy, decreases the

Presented at the RMSE workshop held in conjunction with the 13th ACM Conference on Recommender Systems (RecSys), 2019, in Copenhagen, Denmark.

[1]The code for the reproduction as well as for BlurM(or)e and the exploratory analysis that we carried out is available at https://github.com/STrucks/BlurMore

accuracy of a logistic regression inference model from 80.2% on the original data to 2.5% on the obfuscated data (adding 10% extra ratings). The other proposed strategies have a smaller impact on the classification accuracy. For this reason, in this work, we focus on, and extend, the greedy strategy. Details and more explanations about the gender inference and obfuscation process can be found in section 5.

2.2 Inference on the User-Item matrix

The goal of BlurMe obfuscation is to protect against gender inference. The BlurMe [11] authors use basic machine learning models that can successfully infer users' gender from the user-item matrix. The most recent work on inference on the user-item matrix is, to our knowledge, that of [6], who developed a deep retentive learning framework that beats the conventional, standard machine learning approaches in the task of inferring user demographic information. For gender inference, [6] achieves a classification accuracy of 82%. However, this is only 2% better than the standard logistic regression model used in [11]. We adopt the model from [11] here since it is sufficiently close to the state of the art for our purposes.

3 BUILDING A BETTER BLURME

3.1 The Issue with BlurMe

BlurMe [11] proposes a powerful algorithm that can obfuscate the gender of a user. However, BlurMe has an important flaw: If the rating frequency of the movies are visualized, it is possible to determine that BlurMe has been applied to the data set, and to identify the movies for which ratings have been added. In figure 1(A), the rating frequency is shown for 20 items from the MovieLens data set before obfuscation. In figure 1(B), the rating frequency is shown for the same 20 items after obfuscation with BlurMe. BlurMe exhibits sharp spikes of items; here, it is item ID 27 (called *Persuasion*), which is marked in red. These spikes indicate that BlurMe has been applied, and point to the movies for which ratings have been added. There are two dangers associated with these spikes. First, if BlurMe is running at an operating point of 10% extra ratings using the greedy strategy, as mentioned above, then the gender inference accuracy is 2.5%. This means that if the information is known that BlurMe has been applied, it is simple to reverse the decision of the classifier, and gender can be known with an accuracy of 97.5%. Second, if we do not know the operating point of BlurMe (<10% extra ratings will not guarantee us a gender classification accuracy that we can reverse), we still can find the spikes in the rating histogram, and attempt to reverse BlurMe. In order to find a BlurMe spike we would look for movies that are known not to be particularly popular, but still have a lot of ratings in the BlurMe data. In this paper, we focus on addressing the first danger, and leave the second to future work.

3.2 The Definition of BlurM(or)e

BlurM(or)e was inspired by an exploratory analysis that we carried out, which revealed that a large number of movies are indicative of a gender. For this reason, it is not necessary to restrict the algorithm to add ratings only to the most correlated movies (like the greedy strategy of BlurMe does). This means that we can mask the data without heavily relying on a small set of movies indicative of gender.

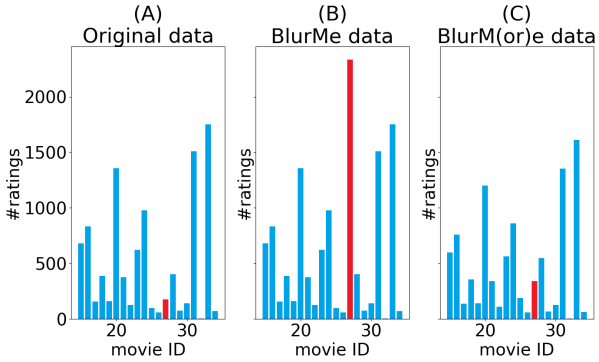

Figure 1: *#ratings* per movie for the movies 15 to 35. The red bar indicates an example of obvious data obfuscation after BlurMe is applied. The BlurMe data was created with the greedy strategy and with 10% extra ratings. The BlurM(or)e data contains also 10% extra ratings.

Based on these insights, we designed BlurM(or)e, which works as follows: We create, just like BlurMe, two lists of movies, L_f *and* L_m, that correlate most strongly with females and males respectively. After that, we alter every user profile by adding movies from the opposite gender list with the greedy strategy proposed in BlurMe [11]. However, if a movie has already doubled its initial rating count, it will be removed from the list. (We use $\times 2$, i.e., doubling, in this paper because it works well, and leave exploration of other possible values to future work). Also, we keep track of the number of added ratings, so that we can remove the same number later on. After every user has received extra ratings up to a fixed percentages of their original ratings, we remove ratings from users that have rated a lot of movies (here we choose $\geqslant$ 200 movies, although future work could investigate other values). The idea is that these users provide already enough data for the gender classifier, so removing some of their ratings would not impact the classifier. This idea is also inspired by our exploratory analysis, which revealed that the gender classifier does not benefit from additional data once a user has already provided 200 ratings. This removal would be more difficult to diagnose in the user-item matrix, since exact information of the rating rates about users would need to be available.

4 EXPERIMENTS AND MAIN RESULTS

4.1 Data

This study uses the publicly available MovieLens data set[2]. We chose MovieLens 1M, which is also used by BlurMe [11], whose work we are reproducing and extending. MovieLens 1M contains 3.7K movies and about 1M ratings of 6K different users, and also information on binary user gender. It is important to note that the distribution in the data set is unbalanced: there are 4331 males that produced 750K ratings and 1709 females that produced 250K ratings. Statistics of the original and the obfuscated data sets, are summarized in Table 1. We note that the number of items decreases for BlurM(or)e data sets due to the fact that the algorithm might remove all ratings of a certain movies by accident.

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

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

data set	#Users	#Items	#Ratings	Range	Av.rating	Density(%)	Variance
MovieLens 1m	6040	3706	1.000.209	[1,5]	3.58	4.47	1.25
BlurMe 1% extra ratings	6040	3706	1.013.416	[1,5]	3.58	4.53	1.25
BlurMe 5% extra ratings	6040	3706	1.052.886	[1,5]	3.58	4.70	1.20
BlurMe 10% extra ratings	6040	3706	1.099.545	[1,5]	3.57	4.91	1.16
BlurM(or)e 1% extra ratings	6040	3705	1.000.797	[1,5]	3.57	4.47	1.24
BlurM(or)e 5% extra ratings	6040	3700	1.000.773	[1,5]	3.55	4.48	1.22
BlurM(or)e 10% extra ratings	6040	3699	1.000.395	[1,5]	3.57	4.48	1.16

4.2 Comparison of BlurMe and BlurM(or)e

We compare the performance of our new obfuscation mechanism, BlurM(or)e, with the original obfuscation mechanism BlurMe. The performance is measured, in line with the experiments in BlurMe, by the classification accuracy of a logistic regression model that is trained on unaltered data, and tested on obfuscated data. The performance is cross-validated using 10-fold cross-validation. Table 2 shows that BlurM(or)e performs similarly to BlurMe. The more obfuscation is applied to the data set, the lower the classification accuracy is. Note that Table 2 contains the reproduction of BlurMe that is discussed in detail in section 5. A big advantage of BlurM(or)e

		Extra ratings			
Classifier	**Data set**	0%	1%	5%	10%
Logistic Regression	BlurMe	0.76	0.54	0.15	0.02
Logistic Regression	BlurM(or)e	0.76	0.64	0.36	0.19
Random Classifier	Original	0.50	0.50	0.50	0.50

Table 2: Gender inference results measured in accuracy on BlurMe (reproduction) and BlurM(or)e

is that an attacker cannot easily see that the data set is obfuscated. Figure 1 on the previous page shows the number of ratings per movie for 20 different movies in the MovieLens 1m data set. The red bar corresponds to the number of ratings for the movie with ID 27. After the BlurMe obfuscation is applied, the red bar spans approximately ten times its original size. This makes the attacker suspicious and indicates that the data set is obfuscated. However, if the BlurM(or)e obfuscation is applied, the red bar only doubles its size, which is less noticeable. Also, BlurM(or)e has more similar statistics to the original data. Table 1 shows that BlurM(or)e keeps the number of interactions as well as the density similar to the original MovieLens data set, while BlurMe produces a more dense data set with more interactions.

The reduction part of BlurM(or)e has a less noticeable effect on the data set. Since the ratings are removed randomly from users with an extreme number of ratings, the number of ratings per movies distribution does not change dramatically (the bar with ID 20 shrinks $\approx$ 10% of its original size in the BlurM(or)e data set).

4.3 Recommendation Performance

Using a well known collaborative filtering technique, Matrix Factorization [4], similar to BlurMe, we notice that the change in RMSE is not substantial. The change has a maximum of 0.0298 for Movielens with BlurM(or)e and 0.0381 for BlurMe (with greedy strategy and 10% extra ratings). We can see in Table 3 that the RMSE is decreasing with an increase in obfuscation. BlurMe [11] discovered the same effect and explained that this might be due to the density of the obfuscated data. Since BlurM(or)e does not increase the overall density of the data, an alternative explanation can be found. The reason, lies perhaps, in increasing the density of users with few ratings.

	Extra ratings			
Obfuscation	0%	1%	5%	10%
Original	0.8766	—	—	—
BlurMe	0.8766	0.8686	0.8553	0.8385
BlurM(or)e	0.8766	0.8711	0.8640	0.8468

Table 3: The RMSE performance with Matrix Factorization on the original data, BlurMe data and on BlurM(or)e data.

5 BLURME REPRODUCTION IN DETAIL

Since we did not have the code of the original BlurMe [11], we reimplemented it in order to carry out the comparison in this paper. Because the paper was not specific about the settings of all parameters, it is not possible to create an exact replication. For completeness, we discuss our reimplementation here, so that authors building on our work have the complete details.

5.1 Gender Inference

This section describes our reimplementation of the gender inference models. We create the user-item matrix by associating every user with a vector of ratings: x_i with i being the index of the user and $x_{i,j}$ being the rating of user i for movie j. If the movie is not rated, we set $x_{i,j} = 0$. This results in a $U\,x\,I$ matrix, where U is the number of users and I is the number of items. Every user vector is associated with a gender, that will serve as target label for the classifier.

Following the experiments of [11], all classifiers are trained and tested on this user-item matrix with 10-fold cross-validation. We do not have information about the splits that were used, so we use our own splits. The ROC area under the curve as well as precision and recall are reported as performance measures. A comparison of the results can be seen in Table 4. The SVM uses a linear kernel and a C value of 1. For the Bernoulli classifier, the user-item matrix is transformed, so that every rating $x_{i,j}$ that is greater than 0, is set to 1. This means that the Bernoulli Bayes classifier ignores the value of the rating and only uses information about whether a user i rated the movie j or not. All remaining parameters for the other classifiers are set to the default values.

There is about a 4% difference between the scores reported in the original BlurMe paper [11], and those we measured with our

reproduction. Further exploration revealed that normalization, in terms of scaling all ratings from values in $[0, 5]$ to values in $[0, 1]$, can have a large impact on scores. We do not focus on normalization further here, but point out its impact because it suggests that there are parameters that could have been adjusted that are not explicitly recorded in [11]. In this paper, we have chosen to focus on the logistic regression model, since it is the fastest and achieves the best performance.

	BlurMe results		**Reproduction Results**	
Classifier	AUC	P/R	AUC	P/R
Bernoulli	0.81	0.79/0.76	0.77	0.88/0.48
Multinomial	0.84	0.80/0.76	0.81	0.89/0.77
SVM	0.86	0.78/0.77	0.79	0.83/0.82
Logistic Regression	0.85	0.80/0.80	0.81	0.84/0.83

Table 4: Gender inference results for both, BlurMe and the reproduction thereof. The performance is measured in ROC AUC, precision and recall.

Note that Table 4 uses ROC AUC as performance metric and Table 2 uses classification accuracy. This choice was made by BlurMe and for the sake of comparing the models, we did the same.

5.2 Gender Obfuscation

This section describes our reimplementation of the obfuscation approach of BlurMe [11]. Recall that the basic idea of BlurMe is to add fictional ratings to every user that are atypical for the their gender. BlurMe [11] creates two lists, L_f and L_m, of atypical movies for each gender by training and cross-validating a logistic regression model on the training set. The movies in L_f and L_m are ranked according to their average rank across the folds. The rank of a movie within a fold is determined by its coefficient that is learned by the logistic regression model. The lists L_f and L_m also include the average coefficient over all folds for each movie that serve as correlation metric between the movie and the user's gender.

After these lists are created, BlurMe takes every user profile and adds k fictive ratings to the profile for movies from the opposite gender list. The parameter k limits the number of extra ratings and is set to 1%, 5% or 10% in the original experiments. A male user with 100 ratings in the original data set would be obfuscated by adding 5 (for $k = 5\%$) fictive ratings from the female list.

There are some design choices left: Which movies should be selected from the lists and what should the fictive rating be? The authors of BlurMe [11] proposed three different selection strategies for the first problem: the *Random Strategy*, the *Sampled Strategy* and the *Greedy Strategy*. The *Random Strategy* chooses k movies uniformly at random from the list, the *Sampled Strategy* chooses k movies randomly, but in line with the score distribution of the movies. Thus, a movie that has a high coefficient is more likely to be added. Finally, the *Greedy Strategy* chooses the movie with the highest score. The authors do not mention the length of the lists, thus we chose to include all movies with a positive coefficient in the L_f list, and all movies with a negative coefficient in the L_m list.

For the fictive rating of a user A for a movie B, BlurMe suggests using either the average rating for movie B or the predicted rating for user A for movie B. Since [11] reports that there is almost no difference between these approaches, we chose to set the extra ratings for a movie according to its respective overall average rating. This average is rounded, because only integer ratings are valid.

The authors of BlurMe take the following attack protocol into account: A gender inference model is trained on real, non-obfuscated data and tested on the obfuscated data. For this reason, the gender inference model is trained on unaltered data and tested on obfuscated data. They use 10-fold cross-validation and report the average classification accuracy of the model.

We report results achieved by our BlurMe reproduction in Table 5. The reproduction is generally congruent with the original. The difference is negligible, we can see that the classification accuracy decreases if the obfuscation increases.

		Extra ratings			
	Strategy	0%	1%	5%	10%
BlurMe	*Random*	0.802	0.776	0.715	0.611
	Sampled	0.802	0.752	0.586	0.355
	Greedy	0.802	0.577	0.173	0.025
Reproduction	*Random*	0.76	0.74	0.69	0.62
	Sampled	0.76	0.71	0.58	0.33
	Greedy	0.76	0.54	0.15	0.02
Reproduction, Normalized	*Random*	0.81	0.80	0.78	0.76
	Sampled	0.81	0.80	0.78	0.75
	Greedy	0.81	0.78	0.74	0.70

Table 5: Performance of BlurMe's and the reproduction's obfuscation algorithm measured by classification accuracy.

6 DISCUSSION & CONCLUSION

In conclusion, this work points to a weakness in a state-of-the-art gender obfuscation algorithm, BlurMe [11], and presents an improved algorithm, BlurM(or)e, that addresses the issue. BlurM(or)e is shown to be able to obfuscate gender in the user-item matrix without substantial increase in RMSE. In other words, it keeps the utility of the data set intact. This work has shed light on some of the challenges of gender obfuscation.

We finish with a discussion of points from [11] that should be taken into account in future research. As mentioned before, normalization of the data set can have an enormous impact on the classification performance. In Table 5, we see that when our reproduction incorporates normalization the accuracy of gender inference still decreases with increasing obfuscation, but at a much slower rate.

In addition, BlurMe used the ROC area under the curve metric for the first gender inference experiments, yet changed to classification accuracy for the gender inference on the obfuscated data set. Using accuracy as a performance metric on imbalanced data sets is a practice that should be avoided. It is advised to report the ROC AUC, precision-recall AUC and ROC AUC on skew-normalized data when dealing with imbalanced data sets [3].

Finally, BlurMe declares (in [11]) the classification accuracy of 2.5% as a success. One can argue that the gender is only truly obfuscated if an attacking model achieves the same performance as a random classifier (i.e., exactly 50% accuracy, in the case of binary classification). This point should be taken into account in deciding the operational settings for BlurMe or BlurM(or)e. The decision also needs to consider the ease with which it is possible to detect whether a user's data has been obfuscated. Future work will study possibilities for obfuscating obfuscation.

REFERENCES

[1] Shlomo Berkovsky, Yaniv Eytani, Tsvi Kuflik, and Francesco Ricci. 2007. Enhancing Privacy and Preserving Accuracy of a Distributed Collaborative Filtering. In *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys '07)*. ACM, 9–16.

[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] László A Jeni, Jeffrey F Cohn, and Fernando De La Torre. 2013. Facing Imbalanced Data—Recommendations for the Use of Performance Metrics. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. IEEE, 245–251.

[4] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer Society Press* 42, 8 (2009), 30–37.

[5] Dongsheng Li, Qin Lv, Li Shang, and Ning Gu. 2017. Efficient Privacy-Preserving Content Recommendation for Online Social Communities. *Neurocomputing* 219 (2017), 440–454.

[6] Yongsheng Liu, Hong Qu, Wenyu Chen, and SM Hasan Mahmud. 2019. An Efficient Deep Learning Model to Infer User Demographic Information From Ratings. *IEEE Access* 7 (2019), 53125–53135.

[7] Roger McNamee and Sandy Parakilas. 2018. The Facebook breach makes it clear: data must be regulated, The Guardian. https://www.theguardian.com/commentisfree/2018/mar/19/facebook-data-cambridge-analytica-privacy-breach, Online; accessed 05-July-2019.

[8] Rupa Parameswara and Douglas M. Blough. 2007. Privacy Preserving Collaborative Filtering Using Data Obfuscation. In *2007 IEEE International Conference on Granular Computing (GRC '07)*. IEEE, 380–380.

[9] Sravana Reddy and Kevin Knight. 2016. Obfuscating Gender in Social Media Writing. In *Proceedings of the 2016 EMNLP Workshop on NLP and Computational Social Science*. ACL, 17–26.

[10] Vicenç Torra. 2017. *Data Privacy: Foundations, New Developments and the Big Data Challenge*. Springer International Publishing, Cham, 191–238.

[11] Udi Weinsberg, Smriti Bhagat, Stratis Ioannidis, and Nina Taft. 2012. BlurMe: Inferring and Obfuscating User Gender Based on Ratings. In *Proceedings of the 2012 ACM Conference on Recommender Systems (RecSys '12)*. ACM, 195–202.