On the impact of Differential Privacy in Collaborative Recommendation

Lucas Caetano Lopes Rodrigues
lucasclopesr@dcc.ufmg.com.br
Federal University of Minas Gerais
Belo Horizonte, Minas Gerais, Brasil

Lucas Starling de Paula Salles
lucastarling@dcc.ufmg.br
Federal University of Minas Gerais
Belo Horizonte, Minas Gerais, Brasil

Paulo Henrique Maciel Fraga
paulo.maciel@dcc.ufmg.br
Federal University of Minas Gerais
Belo Horizonte, Minas Gerais, Brasil

Rafael Augusto Botelho Perez
rafael.perez@dcc.ufmg.br
Federal University of Minas Gerais
Belo Horizonte, Minas Gerais, Brasil

ABSTRACT

The past decades discoveries and breakthroughs in computer systems and, in particular, the fields of Recommender Systems, Information Retrieval, Machine Learning and Data Science have shown how these systems perform an important role in the evolution of society. However, in order to work correctly and generate useful insights and products, these systems require great amounts of individual data, collected from several distinct sources, in many cases without user consent or awareness. Therefore, techniques are being researched and developed to guarantee users' privacy in systems that use large amounts of granular user data.

In this project we evaluate how ϵ-differential privacy affects the quality of recommender systems, namely Item-based Collaborative Filtering, User-based Collaborative Filtering and Latent Factors Collaborative Filtering, by adding noise to the data used in the training stage of the models and comparing the nDCG score before and after the perturbation. We show that inserting noise in the input data directly affects the quality of the recommenders, but it is possible to find an adequate privacy parameter that guarantees an acceptable inferior bound on the recommenders quality while preserving users' privacy.

CCS CONCEPTS

• **Recommender Systems**; • **Differential Privacy**; • **Collabortive Filtering**; • **Item-based Collabortive Filtering**; • **User-based Collabortive Filtering**; • **Latent Factors Collabortive Filtering**; • **Matrix Factorization**;

KEYWORDS

Recommender Systems, Differential Privacy, Collaborative Filtering

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the past decades we have witnessed the emergence of computer systems with great capacity for storing and processing enormous amounts of data, alongside with the popularization of large-scale software that collects user data in every level of granularity. The ability to collect and process such amounts of data has raised concerns about these system's users privacy. Users' data is subject not only to system exploits and data leaks, but also unethical exploitation by the technology companies that hold the data.

Given the increasing capacity of storing and processing user data, new challenges have emerged in the context of moral and ethical use of computer systems, challenges that previous generations had never faced. Therefore, new approaches to deal with such challenges are being thoroughly studied and developed.

One of these newborn challenges, faced by governments and big technology companies that collect large amounts of user data is how to guarantee users' privacy without completely stopping collecting individual's granular data. This particular challenge is not trivial, since the quantity and quality of the data directly affects the utility of the systems that are build upon that data.

In this project, we analyze how the state-of-the-art techniques to guarantee individual privacy in databases that contain user data affects the quality of recommender systems, namely Item-based Collaborative Filtering, User-based Collaborative Filtering and Latent Factors Collaborative Filtering. Following the approach in [6], we compare the quality of the recommenders before and after we add noise to the users' data and use it to train the recommenders. Our goal is to pragmatically measure how well the recommenders perform when trained over noisy data compared to training them over the original, unperturbed data. We seek to visualize the trade-off between privacy and utility, which is well-known in the privacy research literature.

2 DIFFERENTIAL PRIVACY

In order to prevent improper usage of users' data, great efforts have been directed to create privacy definitions and mathematical models that can pragmatically measure users privacy in modern systems that collect huge amounts of data. One of the most accepted and widespread modern privacy definition is *Differential Privacy* [1–4].

2.1 Privacy

Differential Privacy is intuitively based on the concept of *Plausible Deniability*, i.e., every user may plausibly deny the presence of a tuple in a database that can be linked back to them. Therefore, the output of a computation over the database should not allow inference about any particular record in the input [5]. To achieve such guarantees, it is required that the probability of an output of any computation over the database is roughly the same given any small input changes, such as adding or removing a record from the database.

Formally, we denote D and D' as two identical datasets except for one row. We say that any couple of datasets with such property are *adjacent*. Let $\mathcal{M} : D \rightarrow Range(\mathcal{M})$ be a *randomized function* that takes as input the rows of D and D' and outputs a computation over the rows (e.g., $\mathcal{M}$ can be a *counting query*, a *statistical calculation* or a *recommender system*).

Definition 2.1. A randomized function or **mechanism** $\mathcal{M} : D \rightarrow$ Range($\mathcal{M}$) is said to maintain ϵ-differential privacy if, for any two *adjacent* datasets D and D' and any output $S \subseteq Range(\mathcal{M})$,

$$P(\mathcal{M}(D) \in S) \leq e^{\epsilon} \cdot P(\mathcal{M}(D') \in S)$$

The privacy guarantee is measured through the parameter ϵ. Greater values of ϵ imply lower privacy, while small values of ϵ imply higher privacy. The choice for this parameter is not trivial and, in most cases, the decision is made experimentally by leveraging the level of privacy guarantees and the underlying loss of quality in the data resulting from applying privacy-preserving techniques.

2.2 Utility

The employment of an ϵ-differentially private Mechanism $\mathcal{M}$ affects the data's utility, since the mechanism is designed to perform non-deterministic computations that satisfy the output constraints stated in Definition 2.1.

For example, suppose the mechanism $\mathcal{M}$ is a recommender system that takes as input a database containing users' ratings for a set of items and a set of user-item pairs it will generate predictions for. The system outputs a prediction score for each user-item pair in the input. In order to protect users' privacy, noise is added to the input data scores individually, following some probability distribution. Naturally, the quality of the recommender system is going to decrease, given that the input does not correspond directly to the users' true ratings.

Thus, the noise added to the input data must be carefully calibrated in order to balance users privacy and recommendation quality. In particular, differential privacy is guaranteed by adding noise calibrated to the L_1-sensitivity of the input data. In the case of ratings, the L_1-sensitivity is given by $\Delta r = r_{\max} - r_{\min}$ [2, 6], which is the maximum value a single user's rating can contribute to the system's output.

2.3 The Laplace Mechanism

In this project, the Laplace Mechanism calibrated to the L_1-sensitivity was used to add noise to the users' ratings data individually. The Laplace Mechanism is a noise-adding mechanism that takes as input a real value x and adds to it a real value following the Laplace distribution:

Definition 2.2. The Laplace Distribution centered at 0 with scale b is the distribution with probability density function

$$\text{Lap}(x|b) = \frac{1}{2b} e^{-\frac{|x|}{b}}$$

In order to guarantee ϵ-differential privacy, we take $b = \frac{\Delta r}{\epsilon}$ in Definition 2.2 [5].

3 PRIVACY PRESERVING RECOMMENDER SYSTEMS

Recommender systems face particular challenges in the privacy field: they need to consume a lot of granular user data to generate useful recommendations. Usually, the more data the system has, more useful recommendations it can generate. However, collecting too much data might concern the individuals that use the system and is prone to create privacy policy compliance difficulties for the company.

An adequate approach to mitigate users' privacy risks is to develop and use privacy-preserving recommender systems. To create such systems, the inherent tradeoff between privacy and recommendation quality must be leveraged.

Following the process described in [6], a privacy-preserving recommender system can be implemented by adding noise to the input data before training any algorithms or building any recommender, a technique called *Input Perturbation*.

The input perturbation is done by adding noise through a Laplace Mechanism, calibrated to the L_1-sensitivity of the data, given by $\Delta r = r_{\max} - r_{\min}$, where r is a user rating in the recommender system input database.

4 EXPERIMENTS AND RESULTS

Taking [6] as our main reference, we carry out an analysis of the quality of three different recommender systems, comparing the results of systems trained on original data[1] versus systems trained on perturbed data, in which we inserted noise using the Laplace Mechanism.

For the analysis, we developed three different Collaborative Filtering recommenders: Item-based Collaborative Filtering, User-based Collaborative Filtering and Latent Factors Collaborative Filtering using Matrix Factorization.

The three models were trained in the original data and generated predictions for a set of user-item pairs. We evaluated the quality of the recommenders using the *Normalized Discounted Cumulative Gain* (nDCG) metric, which creates our baseline. Afterwards, we recreated the models using as input the perturbed data, and evaluated them using the same metric. The perturbed input databases were generated by inserting noise in the ratings column according to the Laplace distribution with parameter $b = \frac{\Delta r}{(\epsilon / \text{size of database})}$, varying the values $\epsilon = 3, \epsilon = 10, \epsilon = 100, \epsilon = 1000, \epsilon = 10\,000, \epsilon =$

[1] In the evaluation we used the Amazon Electronic Product Reviews dataset with the rating scale of 1 to 5 stars (https://www.kaggle.com/saurav9786/amazon-product-reviews).

100 000, $\epsilon = 1\,000\,000$. We discuss the results in the following sections, including plots that show how the perturbed input affects the quality of the recommenders and a table that displays the numerical values for the experiments results.

4.1 Item-based Collaborative Filtering

In Item-based Collaborative Filtering a user-item ratings matrix is taken as input and the predictions are generated by calculating the similarity between the target item and the other items the target user has rated and liked. Similarity between item vectors is calculated using cosine similarity.

For the experiments, recommendations were generated based on the original data rating values and the quality of the resulting recommender was measured using the nDCG metric. Then, the input data was perturbed according to the Laplace distribution using the method described in Section 2.3 and the process was repeated. We present the results in Figure 1 for the different values of ϵ.

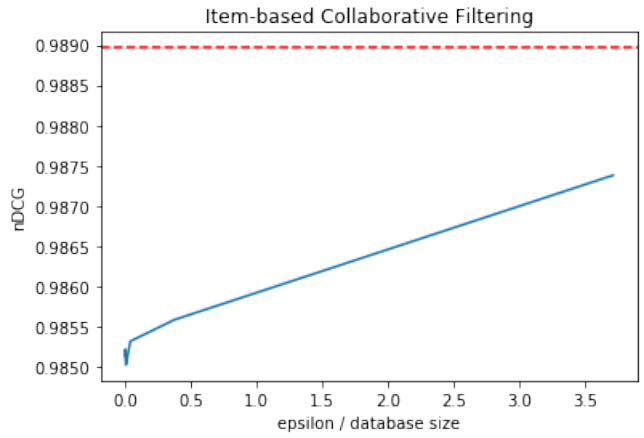

Figure 1: Item-based Collaborative Filtering: the red dashed line indicates the baseline nDCG, measured over the model created from the original data. The blue line shows how the nDCG metric evolves while changing the value of ϵ. nDCG is a ranking quality metric in $[0, 1]$; the closest to 1 the better.

4.2 User-based Collaborative Filtering

In User-based Collaborative Filtering a user-item ratings matrix is taken as input and the predictions are generated by calculating the similarity between the target user and the other users that rated and liked the target item. Similarity between user vectors is calculated using cosine similarity.

For the experiments, recommendations were generated based on the original data rating values and the quality of the resulting recommender was measured using the nDCG metric. Then, the input data was perturbed according to the Laplace distribution using the method described in Section 2.3 and the process was repeated. We present the results in Figure 2 for the different values of ϵ.

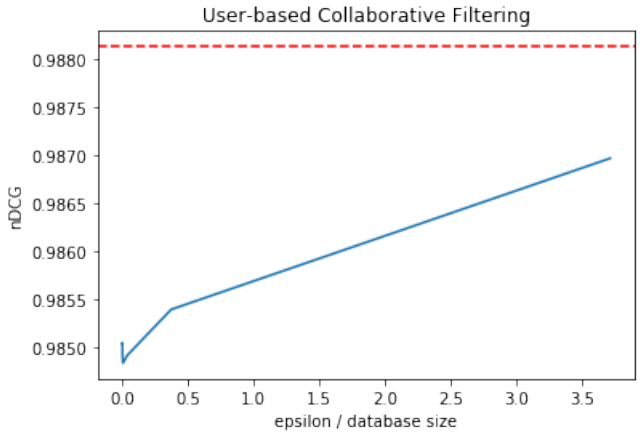

Figure 2: User-based Collaborative Filtering: the red dashed line indicates the baseline nDCG, measured over the model created from the original data. The blue line shows how the nDCG metric evolves while changing the value of ϵ. nDCG is a ranking quality metric in $[0, 1]$; the closest to 1 the better.

4.3 Latent Factors Collaborative Filtering

In Latent Factors Collaborative Filtering, the user-item ratings matrix is also generated before building the recommender. Latent factors are found by factorizing the user-item matrix into two smaller matrices, user-factor and factor-item, achieving dimensionality reduction while also treating the problem of big sparse user-item matrices.

The Matrix Factorization method uses a Gradient Descent algorithm to find the entries of two matrices that decompose the original user-item ratings matrix. In this project, we used Python's *matrix factorization library*[2] to generate the intermediate latent factors matrices and create the recommender model, before and after adding noise to the input data. We present the results in Figure 3 for the different values of ϵ.

4.4 Results

The data collected from the experiments confirm the previously known results from the literature around differential privacy: smaller values of ϵ yield less accurate computations over the data and greater privacy, while greater values of ϵ yield more accuracy and less privacy. This unavoidable tradeoff must be taken in consideration by companies and governments who utilize granular user data to produce useful products.

5 CONCLUSIONS

As shown in this project, the usage of *Input Perturbation* method to provide privacy in the context of recommender systems directly affects the resulting quality of the system designed. However, adding noise to the input data is a necessary step in the development of a recommender system that uses great amounts of granular user

[2]https://pypi.org/project/matrix-factorization/

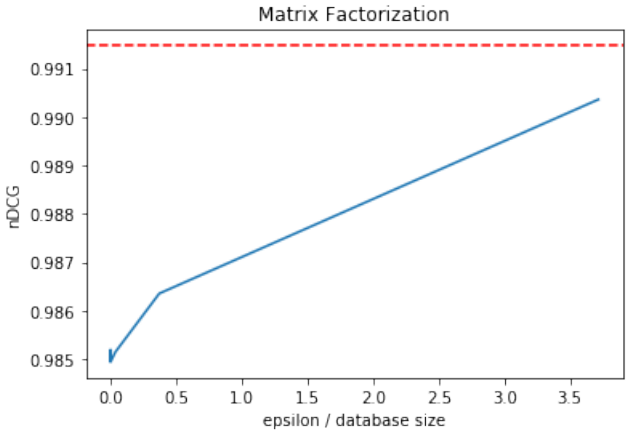

Figure 3: Latent Factors Collaborative Filtering: the red dashed line indicates the baseline nDCG, measured over the model created from the original data. The blue line shows how the nDCG metric evolves while changing the value of ϵ. nDCG is a ranking quality metric in $[0, 1]$; the closest to 1 the better.

Input	Item-based CF	User-based CF	LF CF
Original data	0.98896	0.98812	0.99147
$\epsilon = \frac{3}{269043}$	0.98513	0.98503	0.98517
$\epsilon = \frac{10}{269043}$	0.98522	0.98504	0.98493
$\epsilon = \frac{10^2}{269043}$	0.98514	0.98491	0.98519
$\epsilon = \frac{10^3}{269043}$	0.98503	0.98483	0.98495
$\epsilon = \frac{10^4}{269043}$	0.98532	0.98491	0.98514
$\epsilon = \frac{10^5}{269043}$	0.98559	0.98538	0.98635
$\epsilon = \frac{10^6}{269043}$	0.98738	0.98696	0.99036

Table 1: nDCG results for each Collaborative Filtering model. The first row contains the baseline nDCG measurement, calculated for the models trained over the original data. The following rows contain the nDCG measurement for models trained over the noisy data, perturbed according to the method described in Section 2.3, using an ϵ-differentially private Laplace Mechanism with parameters ϵ and $\Delta r = r_{\max} - r_{\min}$.

data, even more so if it deals with sensitive information, such as geolocation or medical data.

We have shown that increasing the value of the privacy parameter ϵ makes the quality of the recommender approach the baseline - a model trained over the original, unperturbed data - while harming individual privacy guarantees. The choice for the ϵ parameter is not trivial, and should be considered carefully in order to correctly comply to data privacy regulations and policies.

As the experiments carried out on this project indicates, the Latent Factors Collaborative Filtering model was the one that performed the best given input perturbation, however, both Item-based and User-based Collaborative Filtering models could achieve similar results by adjusting the ϵ parameter.

REFERENCES

[1] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[2] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3âĂŞ4):211âĂŞ407, August 2014.

[3] Cynthia Dwork. Differential privacy: A survey of results. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation*, TAMC'08, page 1âĂŞ19, Berlin, Heidelberg, 2008. Springer-Verlag.

[4] Cynthia Dwork and Moni Naor. On the diculties of disclosure prevention in statistical databases or the case for dierential privacy. *Journal of Privacy and Confidentiality*, 2, 09 2010.

[5] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. volume Vol. 3876, pages 265–284, 01 2006.

[6] Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. Applying differential privacy to matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, page 107âĂŞ114, New York, NY, USA, 2015. Association for Computing Machinery.